

# 文档说明

---

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

名称: M元CSK扩频

说明: M元CSK扩频系统的扩频解扩以及检测方法研究

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	2025-1-2	小吕同学	首次发布

FindMe: [https://space.bilibili.com/10179894?spm\\_id\\_from=333.1007.0.0](https://space.bilibili.com/10179894?spm_id_from=333.1007.0.0)

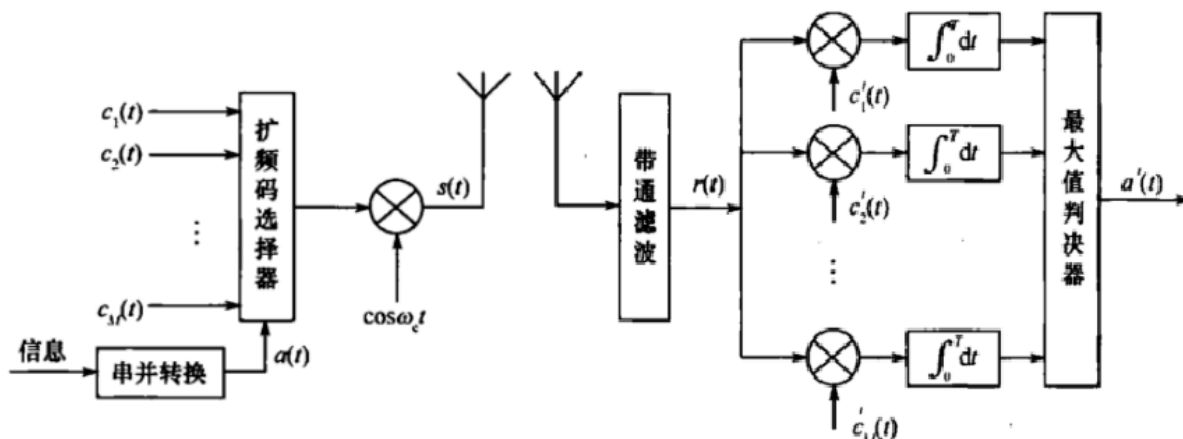
Copyright 2024 Lv. All Rights Reserved.

Distributed under MIT license.

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

# M元扩频原理

传统直接序列扩频采用一条扩频码进行扩频，而M元扩频采用M个扩频码进行扩频，利用扩频码准正交的互相关性质，用扩频码序列携带信息，每条消息可携带信息量从1bit提升到了 $\log_2(M)$ bit。假设选用M=32的M元扩频，每条消息可携带的5bit信息，相对应的，接收端需要做32次相关计算，会提高解调的复杂程度。



## M元扩频仿真

利用扩频码数量携带更多的消息，实际应用中会选取M=32个扩频码，这里为了简化仿真过程，仅采用M=4个扩频码实现。

```
%% 消息生成
bits=60; % 消息个数
mes=randi([0,1],1,bits);
mary=2; % mary进制信源
p=rem(length(mes),mary); % bit补齐
if(p~=0)
    mes=[mes zeros(1,mary-p)];
end
rmes=reshape(mes,mary,[])'; % 串并转换
bimes=bi2de(rmes); % 4进制，2比特

%% Gold序列生成
initial=[1 0 1 1 0 0]; % 6阶
feedback1=103;
feedback2=147;

m1 = mseq(initial,feedback1,0);
m2 = mseq(initial,feedback2,0);
L=length(m2); % 扩频码长

gold=[]; % Gold扩频码矩阵
gold(1,:) = 2*(mod(m1+m2,2)) - 1;
gold(2,:) = 2*(mod(m1+circshift(m2,15),2)) - 1;
gold(3,:) = 2*(mod(m1+circshift(m2,30),2)) - 1;
gold(4,:) = 2*(mod(m1+circshift(m2,45),2)) - 1;

%% 扩频码选择
ssmes=zeros(length(bimes),L);
for i=1:length(bimes)
```

```

ssmes(i,:)=gold(bimes(i,1)+1,:);
end
ssmes=ssmes';
ssmes=reshape(ssmes,1,[]); % 并串转换

%% 脉冲成型、上变频、信道、下变频、低通滤波

%% 相关解码
N=length(ssmes)/L; % 消息个数
M=size(gold,1); % 扩频码个数
en=zeros(M,N); % 存储解码
cor=[]; % 存储互相关函数作图

for i=1:N
    sym=ssmes(1,1+(i-1)*L:i*L); % 选取窗口
    corbuf=zeros(M,2*L-1); % 自相关矩阵缓存
    for j=1:M
        temp=xcorr(sym,gold(j,:)); % 跟M个扩频码做互相关，判断峰值
        if(max(abs(temp))>=40)
            en(j,i)=1;
        end
        corbuf(j,:)=temp;
    end
    cor=[cor corbuf]; % 存储相关矩阵
end

figure % 作图相关峰矩阵
hold on
for i=1:M
    subplot(4,1,i)
    plot(cor(i,:));
    axis([1 length(cor) -20 L]);
end

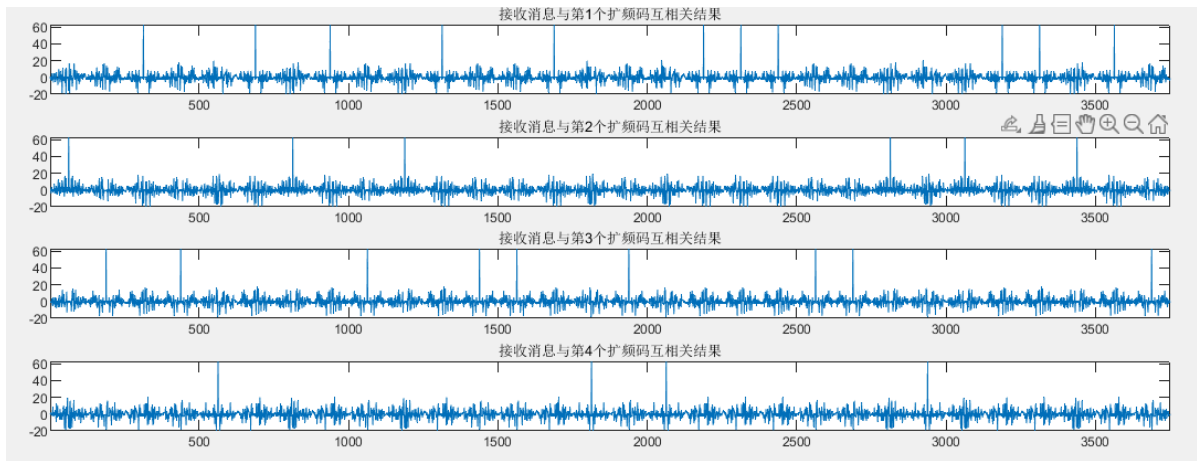
encode=zeros(1,N); % 峰值解码
for i=1:N
    for j=1:M
        if(en(j,i)==1)
            encode(1,i)=j-1;
        end
    end
end
encode=encode';
encode=de2bi(encode)';
dec=reshape(encode,1,[]); % 并串转换

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

```

程序输出：

```
> 解码误码率: 0
```



## 程序分析

4元扩频，每条扩频码携带2bit消息。将消息每两个分成一组

```
% 消息生成
bits=60; % 消息个数
mes=randi([0,1],1,bits);
mary=2; % mary进制信源
p=rem(length(mes),mary); % bit补齐
if(p~=0)
    mes=[mes zeros(1,mary-p)];
end
rmes=reshape(mes,mary,[])'; % 串并转换
bimes=bi2de(rmes); % 4进制，2比特
```

生成四个gold序列，依次编号0-3;

```
% Gold序列生成
initial=[1 0 1 1 0 0]; % 6阶
feedback1=103;
feedback2=147;

m1 = mseq(initial,feedback1,0);
m2 = mseq(initial,feedback2,0);
L=length(m2); % 扩频码长

gold=[]; % Gold扩频码矩阵
gold(1,:) = 2*(mod(m1+m2,2)) - 1;
gold(2,:) = 2*(mod(m1+circshift(m2,15),2)) - 1;
gold(3,:) = 2*(mod(m1+circshift(m2,30),2)) - 1;
gold(4,:) = 2*(mod(m1+circshift(m2,45),2)) - 1;
```

bimes里每一行对应的数字用扩频码补充，最后并串转换为发送信号;

```
% 扩频码选择
ssmes=zeros(length(bimes),L);
for i=1:length(bimes)
    ssmes(i,:)=gold(bimes(i,1)+1,:);
end
ssmes=ssmes';
ssmes=reshape(ssmes,1,[]); % 并串转换
```

接收端选取一个扩频码长度的窗口，分别跟4个扩频码进行相关运算，计算结果中如果有超出阈值的数，就可以判断解码；

```
for i=1:N
    sym=ssmes(1,1+(i-1)*L:i*L); % 选取窗口
    corbuf=zeros(M,2*L-1); % 自相关矩阵缓存
    for j=1:M
        temp=xcorr(sym,gold(j,:)); % 跟M个扩频码做互相关，判断峰值
        if(max(abs(temp))>=40)
            en(j,i)=1;
        end
        corbuf(j,:)=temp;
    end
    cor=[cor corbuf]; % 存储相关矩阵
end
```

最后根据解码峰值的位置，依次将原始消息序列解出来。

```
encode=zeros(1,N); % 峰值解码
for i=1:N
    for j=1:M
        if(en(j,i)==1)
            encode(1,i)=j-1;
        end
    end
end
encode=encode';
encode=de2bi(encode)';
dec=reshape(encode,1,[]); % 并串转换
```

## 双极性M元扩频仿真

通过双极性变换，每个符号可以多携带一个bit消息。

```
%% 消息生成
bits=60; % 消息个数
mes=randi([0,1],1,bits);
mary=3; % mary进制信源
p=rem(length(mes),mary); % bit补齐
if(p~=0)
    mes=[mes zeros(1,mary-p)];
end
rmes=reshape(mes,mary,[]);
bimes=bi2de(rmes); % 8进制，3比特

%% Gold序列生成
initial=[1 0 1 1 0 0]; % 6阶
feedback1=103;
feedback2=147;

m1 = mseq(initial,feedback1,0);
m2 = mseq(initial,feedback2,0);
L=length(m2); % 扩频码长
```

```

gold1=mod(m1+m2,2); % 移位模2加
gold2=mod(m1+circshift(m2,15),2);
gold3=mod(m1+circshift(m2,30),2);
gold4=mod(m1+circshift(m2,45),2);

gold=[]; % Gold扩频码矩阵
gold(1,:) = 2*gold1 - 1; % BPSK, 正负携带1比特消息
gold(2,:) = 2*gold2 - 1;
gold(3,:) = 2*gold3 - 1;
gold(4,:) = 2*gold4 - 1;

gold(5,:) = -1*gold(1,:);
gold(6,:) = -1*gold(2,:);
gold(7,:) = -1*gold(3,:);
gold(8,:) = -1*gold(4,:);

%% 选择扩频码
ssmes=zeros(length(bimes),L);
for i=1:length(bimes)
    ssmes(i,:)=gold(bimes(i,1)+1,:);
end
ssmes=ssmes';
ssmes=reshape(ssmes,1,[]);

%% 脉冲成型、上变频、信道、下变频、低通滤波

%% 相关解码
N=length(ssmes)/L; % 消息个数
M=size(gold,1)/2; % 扩频码个数
en=zeros(M,N); % 存储解码
cor=[]; % 存储互相关函数作图

for i=1:N
    sym=ssmes(1,1+(i-1)*L:i*L); % 选取窗口
    corbuf=zeros(M,2*L-1); % 自相关矩阵缓存
    for j=1:M % 跟M个扩频码做互相关, 判断峰值
        temp=xcorr(sym,gold(j,:));
        if(max(abs(temp))>=40) % 判断峰值
            if(max(temp)>40)
                en(j,i)=1;
            elseif(min(temp)<=-40)
                en(j,i)=-1;
            end
        end
        corbuf(j,:)=temp;
    end
    cor=[cor corbuf]; % 存储相关矩阵
end

figure % 作图相关峰矩阵
hold on
for i=1:M
    subplot(4,1,i)
    plot(cor(i,:));
    axis([1 length(cor) -L L]);
    title(['接收消息与第' num2str(i) '个扩频码互相关结果'])

```

```

end

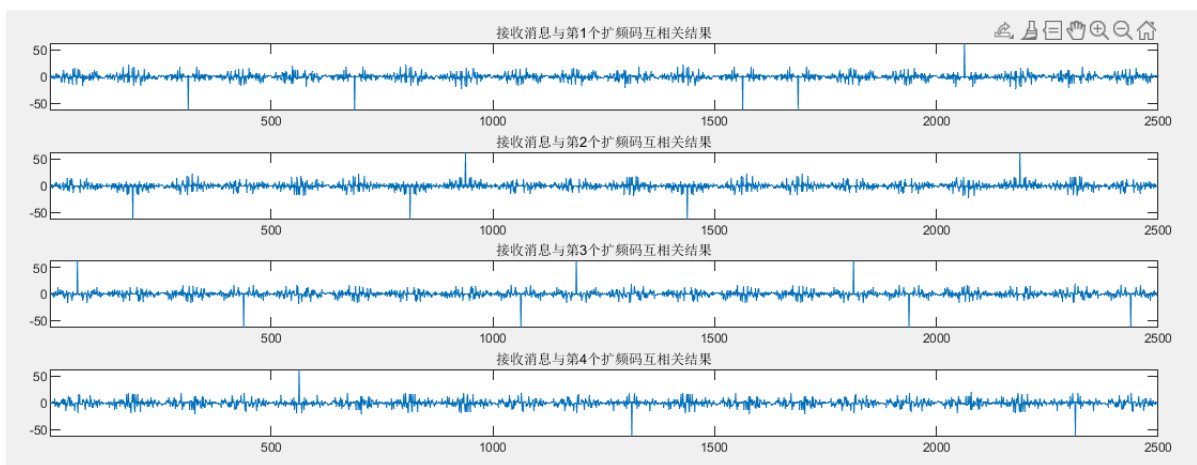
encode=zeros(1,N);           % 峰值解码
for i=1:N
    for j=1:M
        if(en(j,i)==1)
            encode(1,i)=j-1;
        elseif (en(j,i)==-1)
            encode(1,i)=j+3;
        end
    end
end
encode=encode';
encode=de2bi(encode)';
dec=reshape(encode,1,[]);     % 并串转换

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

```

程序输出：

> 解码误码率： 0



## 程序分析

4元扩频，每条扩频码携带2bit消息；双极性码携带1bit消息。每个符号携带3bit消息，设定最高位为极性，剩下两位为扩频码，将消息每三个分成一组；

```

%% 消息生成
bits=60;           % 消息个数
mes=randi([0,1],1,bits);
mary=3;           % mary进制信源
p=rem(length(mes),mary); % bit补齐
if(p~=0)
    mes=[mes zeros(1,mary-p)];
end
rmes=reshape(mes,mary,[])';
bimes=bi2de(rmes); % 8进制，3比特

```

生成四个gold序列，依次编号0-3，将每个gold取负后，编号4-7；

```
%% Gold序列生成
initial=[1 0 1 1 0 0];           % 6阶
feedback1=103;
feedback2=147;

m1 = mseq(initial,feedback1,0);
m2 = mseq(initial,feedback2,0);
L=length(m2);                    % 扩频码长

gold1=mod(m1+m2,2);              % 移位模2加
gold2=mod(m1+circshift(m2,15),2);
gold3=mod(m1+circshift(m2,30),2);
gold4=mod(m1+circshift(m2,45),2);

gold=[];                         % Gold扩频码矩阵
gold(1,:) = 2*gold1 - 1;         % BPSK, 正负携带1比特消息
gold(2,:) = 2*gold2 - 1;
gold(3,:) = 2*gold3 - 1;
gold(4,:) = 2*gold4 - 1;

gold(5,:) = -1*gold(1,:);
gold(6,:) = -1*gold(2,:);
gold(7,:) = -1*gold(3,:);
gold(8,:) = -1*gold(4,:);
```

bimes里每一行对应的数字用扩频码补充，最后并串转换为发送信号；

```
%% 选择扩频码
ssmes=zeros(length(bimes),L);
for i=1:length(bimes)
    ssmes(i,:)=gold(bimes(i,1)+1,:);
end
ssmes=ssmes';
ssmes=reshape(ssmes,1,[]);
```

接收端选取一个扩频码长度的窗口，分别跟4个扩频码进行相关运算，计算结果中设定两个阈值，一正一负，判断解码峰值位置与极性；

```
%% 相关解码
N=length(ssmes)/L;              % 消息个数
M=size(gold,1)/2;              % 扩频码个数
en=zeros(M,N);                 % 存储解码
cor=[];                         % 存储互相关函数作图

for i=1:N
    sym=ssmes(1,1+(i-1)*L:i*L); % 选取窗口
    corbuf=zeros(M,2*L-1);      % 自相关矩阵缓存
    for j=1:M                   % 跟M个扩频码做互相关，判断峰值
        temp=xcorr(sym,gold(j,:));
        if(max(abs(temp))>=40) % 判断峰值
            if(max(temp)>40)
                en(j,i)=1;
            end
        end
    end
end
```



```

        elseif(min(temp)<=-40)
            en(j,i)=-1;
        end
    end
    corbuf(j,:)=temp;
end
cor=[cor corbuf];          % 存储相关矩阵
end

```

最后根据解码峰值的位置与极性，依次将原始消息序列解出来。

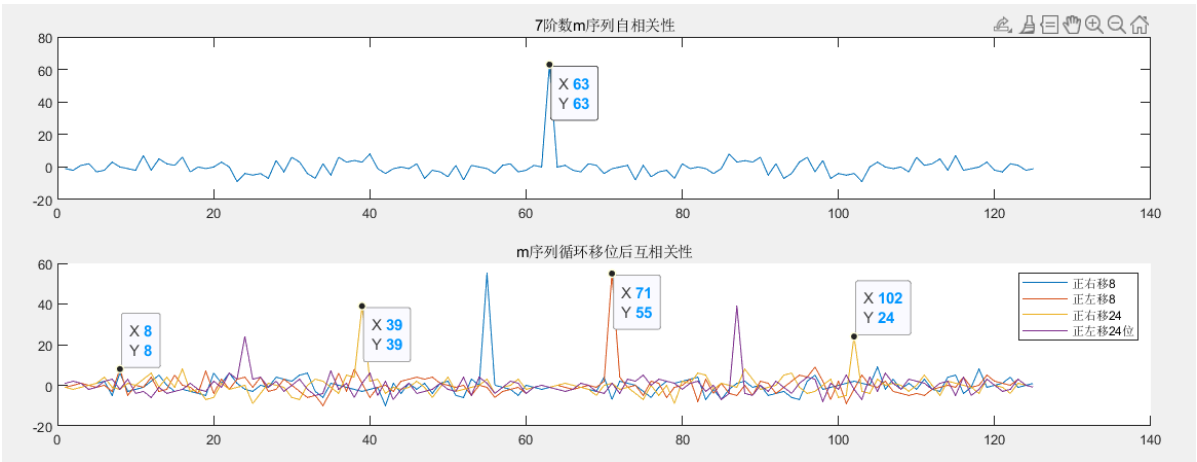
```

encode=zeros(1,N);          % 峰值解码
for i=1:N
    for j=1:M
        if(en(j,i)==1)
            encode(1,i)=j-1;
        elseif (en(j,i)==-1)
            encode(1,i)=j+3;
        end
    end
end
encode=encode';
encode=de2bi(encode)';
dec=reshape(encode,1,[]);    % 并串转换

```

# 循环移位键控扩频原理

循环移位键控（Cyclic Shift Keying, CSK）是一种广义M元扩频方式，通过对伪随机序列进行循环移位，利用相位携带信息。



可以看出，阶数为7，长度为63的m序列，循环移位n位后，互相关峰移位n位，峰值下降n，同时会出现一个峰值位n的第二相关峰，随着循环移位的位数增加，第二相关峰的位置和峰值也会逐渐变化。

## 循环移位键控扩频仿真

```
%% 消息生成
bits=20; % 消息个数
mes=randi([0,1],1,bits);
mary=2; % mary进制信源
p=rem(length(mes),mary); % bit补齐
if(p~=0)
    mes=[mes zeros(1,mary-p)];
end
rmes=reshape(mes,mary,[])'; % 串并转换
bimes=bi2de(rmes); % 4进制，2比特

%% Gold序列生成
initial=[1 0 1 1 0 0]; % 6阶
feedback1=103;
feedback2=147;

m1 = mseq(initial,feedback1,0);
m2 = mseq(initial,feedback2,0);
L=length(m2); % 扩频码长
gold=2*(mod(m1+m2,2)) - 1; % 原始扩频码

csk=[]; % csk扩频码矩阵
csk(1,:)=circshift(gold,24);
csk(2,:)=circshift(gold,8);
csk(3,:)=circshift(gold,-8);
csk(4,:)=circshift(gold,-24);

%% 扩频码选择
ssmes=zeros(length(bimes),L);
for i=1:length(bimes)
    ssmes(i,:)=csk(bimes(i,1)+1,:);
```

```

end
ssmes=ssmes';
ssmes=reshape(ssmes,1,[]);          % 并串转换

%% 脉冲成型、上变频、信道、下变频、低通滤波

%% 相关解码
N=length(ssmes)/L;                  % 消息个数
en=zeros(1,N);                      % 存储解码
cor=[];                             % 存储互相关函数作图

xxx=[];                             % 存储互相关函数峰值的位置和峰值
for i=1:N
    sym=ssmes(1,1+(i-1)*L:i*L);      % 选取窗口
    corbuf=zeros(1,2*L-1);          % 自相关矩阵缓存
    temp=xcorr(gold,sym);            % 跟原始扩频码做互相关(注意顺序)

    [max_val, linear_idx] = max(temp(:));
    [row, col] = ind2sub(size(temp), linear_idx);
    xxx(i,:)=[row,col,max_val];

    cor=[cor temp];                 % 存储相关矩阵
end

encode=[];                          % 根据峰值位置解码
for i=1:N
    if(xxx(i,2)<47)
        encode(i,1)=0;
    elseif(xxx(i,2)<63)
        encode(i,1)=1;
    elseif(xxx(i,2)<79)
        encode(i,1)=2;
    else
        encode(i,1)=3;
    end
end

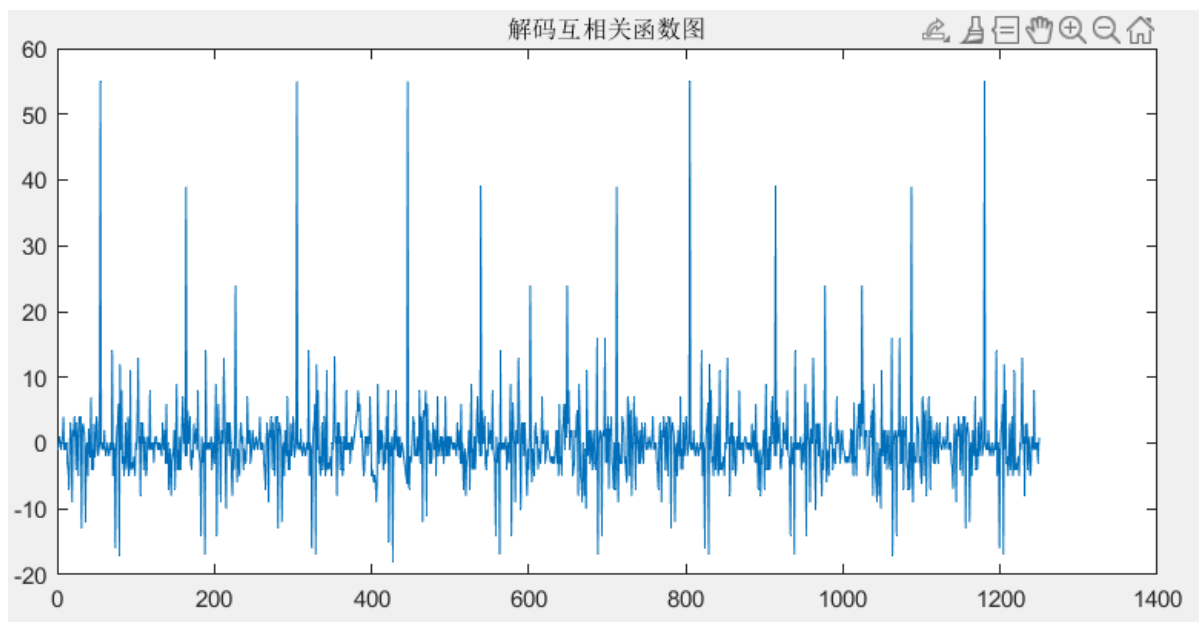
encode=encode';
encode=de2bi(encode)';
dec=reshape(encode,1,[]);           % 并串转换

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

```

程序输出:

```
> 解码误码率: 0
```



## 程序分析

循环移位的噪声容限跟码长选取有很大关系，本次采用7阶gold序列，循环移位4次，可携2bit消息

```

%% 消息生成
bits=20; % 消息个数
mes=randi([0,1],1,bits);
mary=2; % mary进制信源
p=rem(length(mes),mary); % bit补齐
if(p~=0)
    mes=[mes zeros(1,mary-p)];
end
rmes=reshape(mes,mary,[])'; % 串并转换
bimes=bi2de(rmes); % 4进制，2比特

```

生成循环移位矩阵

```

%% CSK序列生成
initial=[1 0 1 1 0 0]; % 6阶
feedback1=103;
feedback2=147;

m1 = mseq(initial,feedback1,0);
m2 = mseq(initial,feedback2,0);
L=length(m2); % 扩频码长
gold=2*(mod(m1+m2,2)) - 1; % 原始扩频码

csk=[]; % csk扩频码矩阵
csk(1,:)=circshift(gold,24);
csk(2,:)=circshift(gold,8);
csk(3,:)=circshift(gold,-8);
csk(4,:)=circshift(gold,-24);

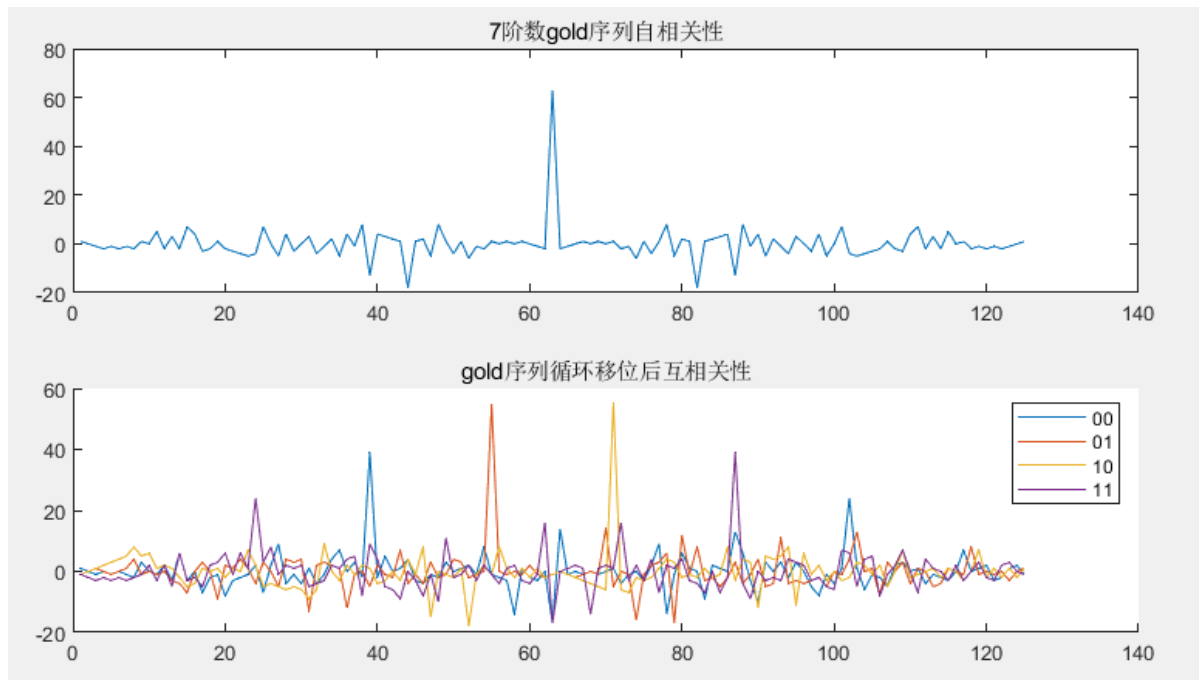
```

可以作图查看互相关性

```

% 作互相关函数图
figure
subplot(2,1,1)
plot(xcorr(gold))
title('7阶数gold序列自相关性');
subplot(2,1,2)
hold on
for i=1:4
    plot(xcorr(gold,csk(i,:)));
end
title('gold序列循环移位后互相关性');
legend('00','01','10','11')

```



选码

```

%% 扩频码选择
ssmes=zeros(length(bimes),L);
for i=1:length(bimes)
    ssmes(i,:)=csk(bimes(i,1)+1,:);
end
ssmes=ssmes';
ssmes=reshape(ssmes,1,[]); % 并串转换

```

重点在解码，首先找出一个buf，跟原始gold序列做互相关（注意，由于是相位携带信息，如果互相关顺序颠倒，整个相位信息会颠倒，误码率为1，不信你可以试试），在互相关函数里找出峰值的坐标，进行存储

```

%% 相关解码
N=length(ssmes)/L; % 消息个数
en=zeros(1,N); % 存储解码
cor=[]; % 存储互相关函数作图

xxx=[]; % 存储互相关函数峰值的位置和峰值
for i=1:N
    sym=ssmes(1,1+(i-1)*L:i*L); % 选取窗口

```

```

corbuf=zeros(1,2*L-1);           % 自相关矩阵缓存
temp=xcorr(gold,sym);            % 跟原始扩频码做互相关(***)注意顺序不能颠倒**)

[max_val, linear_idx] = max(temp(:));
[row, col] = ind2sub(size(temp), linear_idx);
xxx(i,:)=[row,col,max_val];

cor=[cor temp];                  % 存储相关矩阵
end

figure
plot(cor);
title('解码互相关函数图');

```

CSK解码，根据峰值位置解码

```

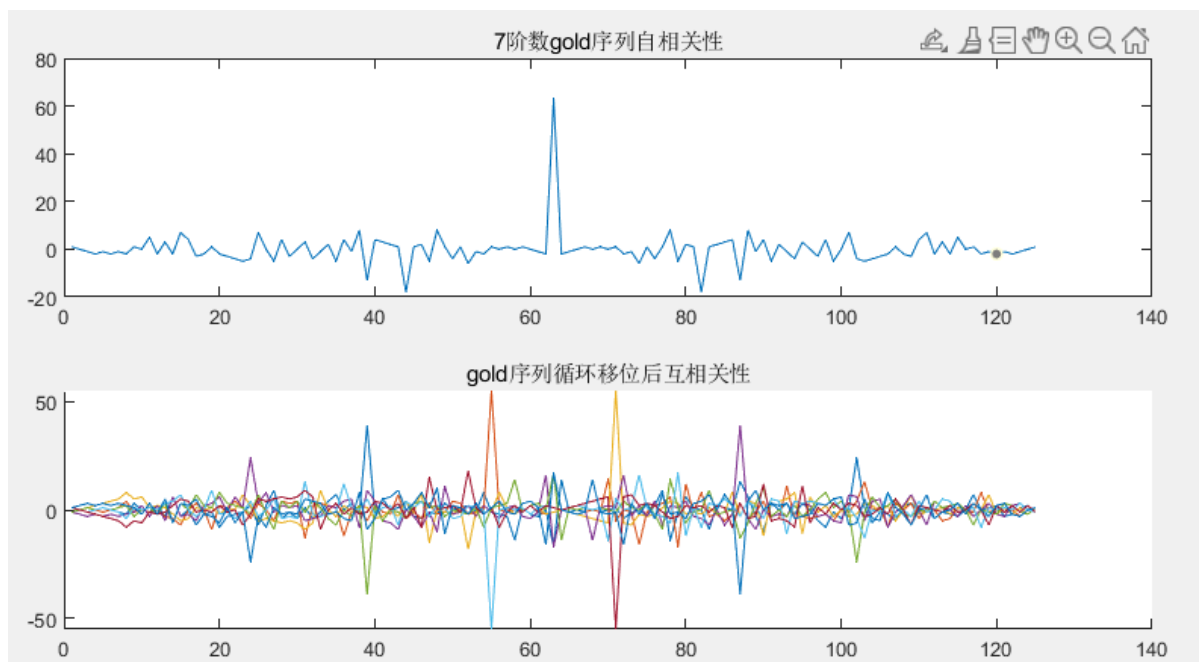
encode=[];                       % 根据峰值位置解码
for i=1:N
    if(xxx(i,2)<47)
        encode(i,1)=0;
    elseif(xxx(i,2)<63)
        encode(i,1)=1;
    elseif(xxx(i,2)<79)
        encode(i,1)=2;
    else
        encode(i,1)=3;
    end
end

encode=encode';
encode=de2bi(encode)';
dec=reshape(encode,1,[]);        % 并串转换

```

## 双极性循环移位键控扩频仿真

双极性循环移位的互相关图



通过双极性变换，每个符号可以多携带一个bit消息。

```
%% 消息生成
bits=20; % 消息个数
mes=randi([0,1],1,bits);
mary=3; % mary进制信源
p=rem(length(mes),mary); % bit补齐
if(p~=0)
    mes=[mes zeros(1,mary-p)];
end
rmes=reshape(mes,mary,[])'; % 串并转换
bimes=bi2de(rmes); % 8进制，3比特

%% CSK序列生成
initial=[1 0 1 1 0 0]; % 6阶
feedback1=103;
feedback2=147;

m1 = mseq(initial,feedback1,0);
m2 = mseq(initial,feedback2,0);
L=length(m2); % 扩频码长
gold=2*(mod(m1+m2,2)) - 1; % 原始扩频码

csk=[]; % csk扩频码矩阵
csk(1,:)=circshift(gold,24);
csk(2,:)=circshift(gold,8);
csk(3,:)=circshift(gold,-8);
csk(4,:)=circshift(gold,-24);

csk(5,:)=-csk(1,:);
csk(6,:)=-csk(2,:);
csk(7,:)=-csk(3,:);
csk(8,:)=-csk(4,:);

%% 扩频码选择
ssmes=zeros(length(bimes),L);
for i=1:length(bimes)
    ssmes(i,:)=csk(bimes(i,1)+1,:);
end
ssmes=ssmes';
ssmes=reshape(ssmes,1,[]); % 并串转换

%% 脉冲成型、上变频、信道、下变频、低通滤波

%% 相关解码
N=length(ssmes)/L; % 消息个数
en=zeros(1,N); % 存储解码
cor=[]; % 存储互相关函数作图

xxx=[]; % 存储互相关函数峰值的位置和峰值
for i=1:N
    sym=ssmes(1,1+(i-1)*L:i*L); % 选取窗口
    corbuf=zeros(1,2*L-1); % 自相关矩阵缓存
    temp=xcorr(gold,sym); % 跟原始扩频码做互相关(注意顺序)
```

```

        if(max(temp(:))<abs(min(temp(:))))           % 判断极性，如果负极性找负峰值
            [max_val, linear_idx] = min(temp(:));
        else                                           % 反之，正极性找正峰值
            [max_val, linear_idx] = max(temp(:));
        end
        [row, col] = ind2sub(size(temp), linear_idx);
        xxx(i,:)=[row,col,max_val];
        cor=[cor temp];                               % 存储相关矩阵
    end

figure
plot(cor);
title('解码互相关函数图');

encode=[];
for i=1:N
    if(xxx(i,2)<47)
        shift=1;
    elseif(xxx(i,2)<63)
        shift=2;
    elseif(xxx(i,2)<79)
        shift=3;
    else
        shift=4;
    end
    if(xxx(i,3)<0)
        sig=4;
    else
        sig=0;
    end
    encode(i,1)=(xxx(i,1)-1)*8+sig+shift-1;
end

encode=encode';
encode=de2bi(encode)';
dec=reshape(encode,1,[]);                            % 并串转换

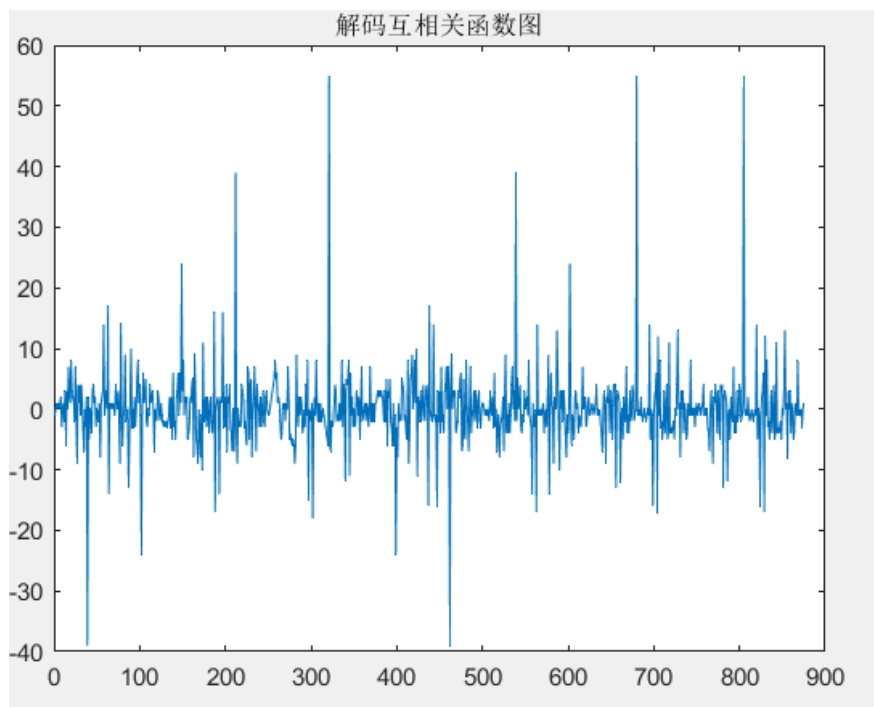
%% 误码率
A=find(dec~=mes);
BER=length(A)/length(mes);
disp(['解码误码率: ' num2str(BER)])

```

程序输出：

```
> 解码误码率: 0
```





## 程序分析

每个符号3bit消息

```
%% 消息生成
bits=20; % 消息个数
mes=randi([0,1],1,bits);
mary=3; % mary进制信源
p=rem(length(mes),mary); % bit补齐
if(p~=0)
    mes=[mes zeros(1,mary-p)];
end
rmes=reshape(mes,mary,[])'; % 串并转换
bimes=bi2de(rmes); % 8进制, 3比特
```

双极性CSK序列生成

```
%% CSK序列生成
initial=[1 0 1 1 0 0]; % 6阶
feedback1=103;
feedback2=147;

m1 = mseq(initial,feedback1,0);
m2 = mseq(initial,feedback2,0);
L=length(m2); % 扩频码长
gold=2*(mod(m1+m2,2))-1; % 原始扩频码

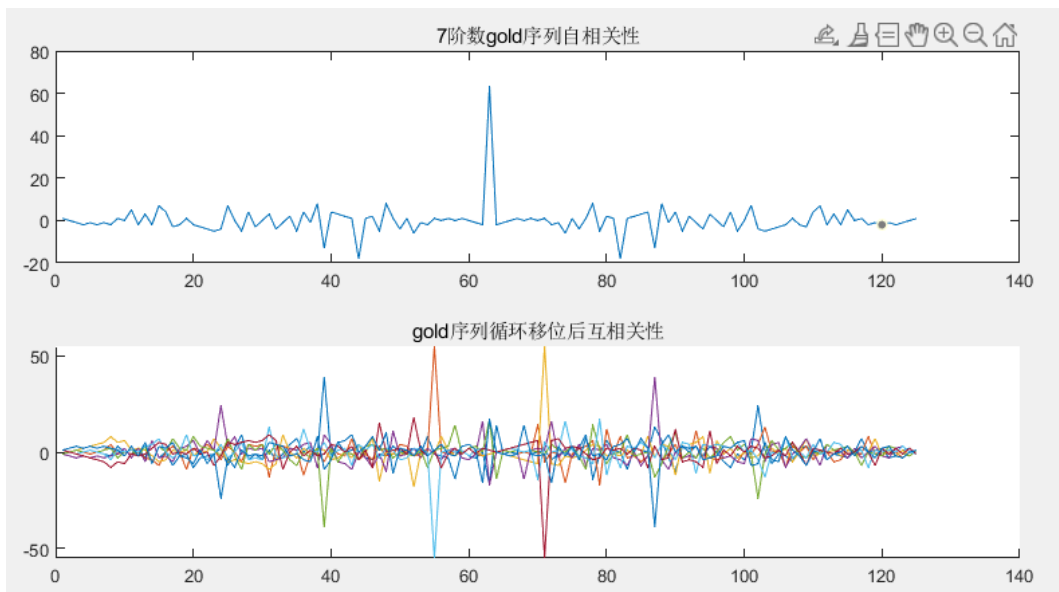
csk=[]; % csk扩频码矩阵
csk(1,:)=circshift(gold,24);
csk(2,:)=circshift(gold,8);
csk(3,:)=circshift(gold,-8);
csk(4,:)=circshift(gold,-24);

csk(5,:)=-csk(1,:);
csk(6,:)=-csk(2,:);
```

```
csk(7,:)=-csk(3,:);
csk(8,:)=-csk(4,:);
```

可以查看互相关图

```
% 作互相关函数图
figure
subplot(2,1,1)
plot(xcorr(gold))
title('7阶数gold序列自相关性');
subplot(2,1,2)
hold on
for i=1:8
    plot(xcorr(gold,csk(i,:)));
end
title('gold序列循环移位后互相关性');
```



选码

```
%% 扩频码选择
ssmes=zeros(length(bimes),L);
for i=1:length(bimes)
    ssmes(i,:)=csk(bimes(i,1)+1,:);
end
ssmes=ssmes';
ssmes=reshape(ssmes,1,[]); % 并串转换
```

重点在解码，首先找出一个buf，跟原始gold序列做互相关（注意，由于是相位携带信息，如果互相关顺序颠倒，整个相位信息会颠倒，误码率为1，不信你可以试试），在互相关函数里，首先判断峰值正负，如果正，就找正峰值最大值的位置，如果为负，就找负峰值最大值的位置并存储

```
%% 相关解码
N=length(ssmes)/L; % 消息个数
en=zeros(1,N); % 存储解码
cor=[]; % 存储互相关函数作图

xxx=[]; % 存储互相关函数峰值的位置和峰值
```

```

for i=1:N
    sym=ssmes(1,1+(i-1)*L:i*L); % 选取窗口
    corbuf=zeros(1,2*L-1); % 自相关矩阵缓存
    temp=xcorr(gold,sym); % 跟原始扩频码做互相关(注意顺序)

    if(max(temp(:))<abs(min(temp(:)))) % 判断极性, 如果负极性找负峰值
        [max_val, linear_idx] = min(temp(:));
    else % 反之, 正极性找正峰值
        [max_val, linear_idx] = max(temp(:));
    end
    [row, col] = ind2sub(size(temp), linear_idx);
    xxx(i,:)=[row,col,max_val];
    cor=[cor temp]; % 存储相关矩阵
end

figure
plot(cor);
title('解码互相关函数图');

```

CSK解码, 根据峰值位置和正负, 解码

```

encode=[];
for i=1:N
    if(xxx(i,2)<47)
        shift=1;
    elseif(xxx(i,2)<63)
        shift=2;
    elseif(xxx(i,2)<79)
        shift=3;
    else
        shift=4;
    end
    if(xxx(i,3)<0)
        sig=4;
    else
        sig=0;
    end
    encode(i,1)=(xxx(i,1)-1)*8+sig+shift-1;
end

encode=encode';
encode=de2bi(encode)';
dec=reshape(encode,1,[]); % 并串转换

```

# 双极性M元CSK扩频原理

结合以上两种方法，4元扩频码携带2bit信息，4相循环移位携带2bit信息，双极性携带1bit消息

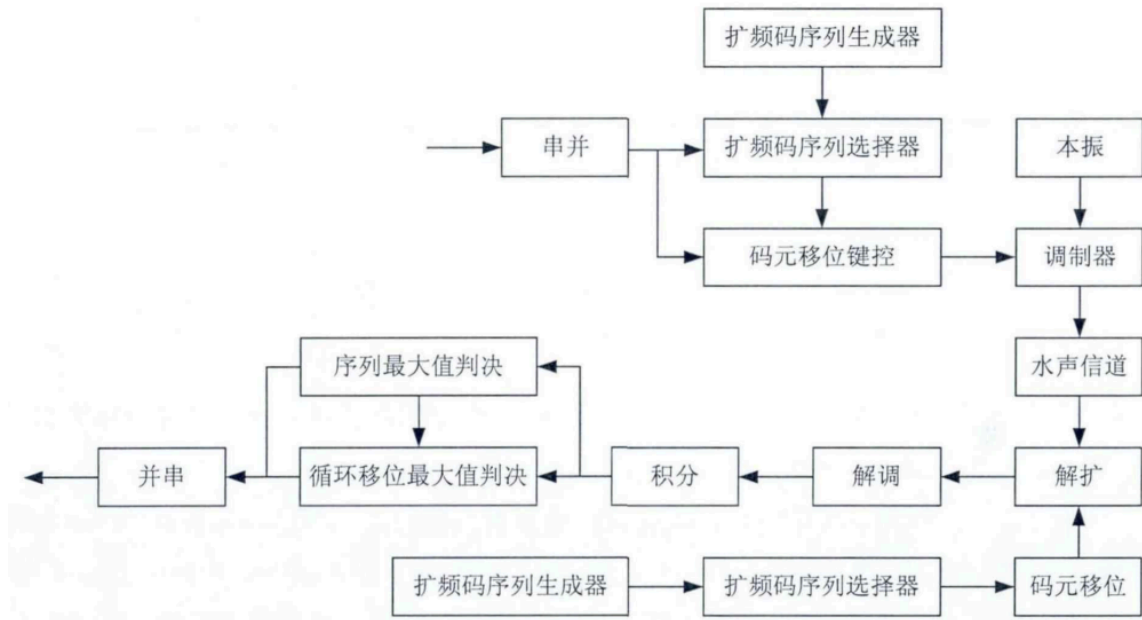


图 2.3 M 元 CSK 水声通信系统

## 双极性M元CSK扩频仿真

```
%% M元扩频-CSK-BPSK-Gold序列
close all; clear; clc;

%% 消息生成
bits=120; % 消息个数
mes=randi([0,1],1,bits);
mary=5; % mary进制信源
p=rem(length(mes),mary); % bit补齐
if(p~=0)
    mes=[mes zeros(1,mary-p)];
end
rmes=reshape(mes,mary,[])'; % 串并转换
bimes=bi2de(rmes); % 8进制，3比特

%% 4元Gold序列4相CSK序列生成
initial=[1 0 1 1 0 0]; % 6阶
feedback1=103;
feedback2=147;
m1 = mseq(initial,feedback1,0);
m2 = mseq(initial,feedback2,0);
L=length(m2); % 扩频码长

gold=[]; % 4元扩频
gold(1,:) = 2*(mod(m1+m2,2)) - 1;
gold(2,:) = 2*(mod(m1+circshift(m2,15),2)) - 1;
gold(3,:) = 2*(mod(m1+circshift(m2,30),2)) - 1;
gold(4,:) = 2*(mod(m1+circshift(m2,45),2)) - 1;

n=[24 8 -8 -24]; % 4相循环移位
```

```

for i=1:length(n)
    goldcdsk1(i,:)=circshift(gold(1,:),n(i));
    goldcdsk2(i,:)=circshift(gold(2,:),n(i));
    goldcdsk3(i,:)=circshift(gold(3,:),n(i));
    goldcdsk4(i,:)=circshift(gold(4,:),n(i));
end

for i=1:length(n) % BPSK
    goldcdsk1(i+4,:)=-goldcdsk1(i,:);
    goldcdsk2(i+4,:)=-goldcdsk2(i,:);
    goldcdsk3(i+4,:)=-goldcdsk3(i,:);
    goldcdsk4(i+4,:)=-goldcdsk4(i,:);
end
gold4cdsk=[goldcdsk1;goldcdsk2;goldcdsk3;goldcdsk4];

%% 选择扩频码
ssmes=zeros(length(bimes),length(gold(1,:)));
for i=1:length(bimes)
    smes(i,:)=gold4cdsk(bimes(i,1)+1,:);
end
ssmes=ssmes';
ssmes=reshape(ssmes,1,[]);

%% 脉冲成型、上变频、信道、下变频、低通滤波

%% 相关解码
N=length(ssmes)/L; % 消息个数
en=zeros(1,N); % 存储解码
cor=[]; % 存储互相关函数作图
M=size(gold,1); % M元扩频码个数

for i=1:N % 求互相关矩阵
    sym=ssmes(1,1+(i-1)*L:i*L); % 选取窗口
    corbuf=zeros(M,2*L-1); % 自相关矩阵缓存
    for j=1:M % 跟原始扩频码做互相关(注意顺序)
        corbuf(j,:)=xcorr(gold(j,:),sym);
    end
    cor=[cor corbuf]; % 存储相关矩阵
end

xxx=[]; % 存储互相关函数峰值的位置和峰值
for i=1:N % 寻找互相关峰值位置
    buf=cor(1:4,1+(i-1)*125:i*125); % 选取窗口
    if(max(buf(:))<abs(min(buf(:)))) % 判断极性, 如果负极性找负峰值
        [max_val, linear_idx] = min(buf(:));
    else % 反之, 正极性找正峰值
        [max_val, linear_idx] = max(buf(:));
    end
    [row, col] = ind2sub(size(buf), linear_idx); % 转成行列形式
    xxx(i,:)=[row,col,max_val];
end

encode=[];
for i=1:N % 根据扩频码和互相关峰值位置与极性, 解码
    if(xxx(i,2)<47)
        shift=0;
    end
end

```

```

elseif(xxx(i,2)<63)
    shift=1;
elseif(xxx(i,2)<79)
    shift=2;
else
    shift=3;
end
if(xxx(i,3)<0)
    sig=4;
else
    sig=0;
end
encode(i,1)=(xxx(i,1)-1)*8+sig+shift;
end
encode=encode';
ec=de2bi(encode)';
dec=reshape(ec,1,[]);

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

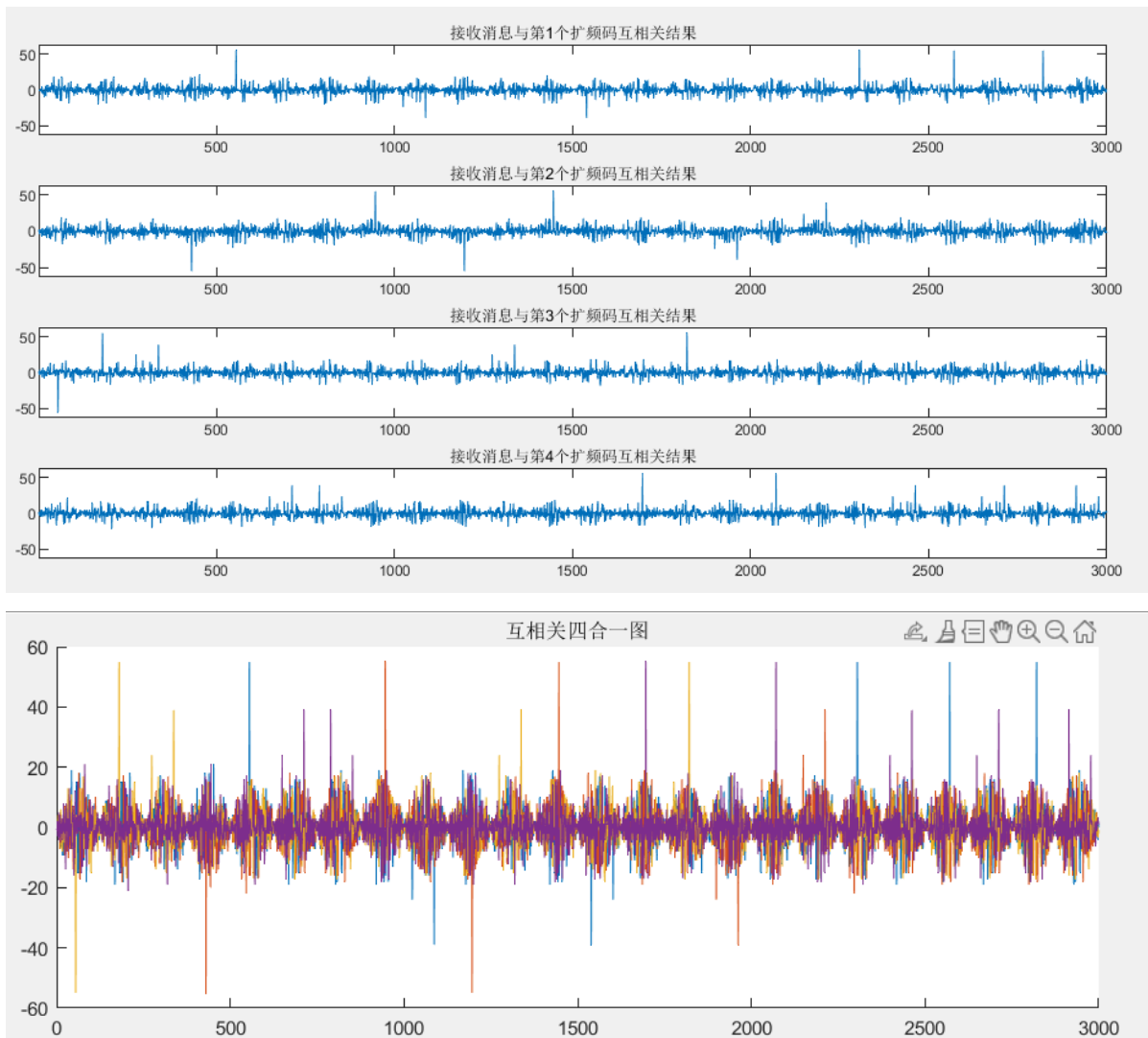
%% 画图
figure
hold on
for i=1:M
    subplot(4,1,i)
    plot(cor(i,:));
    title(['接收消息与第' num2str(i) '个扩频码互相关结果'])
end

figure
hold on
for i=1:M
    plot(cor(i,:));
    title('互相关四合一图')
end

```

程序输出：

```
> 解码误码率: 0
```



## 程序分析

每个符号5bit消息

```
% 消息生成
bits=120; % 消息个数
mes=randi([0,1],1,bits);
mary=5; % mary进制信源
p=rem(length(mes),mary); % bit补齐
if(p~=0)
    mes=[mes zeros(1,mary-p)];
end
rmes=reshape(mes,mary,[])'; % 串并转换
bimes=bi2de(rmes); % 8进制, 3比特
```

双极性4元CSK序列生成

```
% 4元Gold序列4相CSK序列生成
initial=[1 0 1 1 0 0]; % 6阶
feedback1=103;
feedback2=147;
m1 = mseq(initial,feedback1,0);
m2 = mseq(initial,feedback2,0);
L=length(m2); % 扩频码长
```

```

gold=[]; % 4元扩频
gold(1,:) = 2*(mod(m1+m2,2)) - 1;
gold(2,:) = 2*(mod(m1+circshift(m2,15),2)) - 1;
gold(3,:) = 2*(mod(m1+circshift(m2,30),2)) - 1;
gold(4,:) = 2*(mod(m1+circshift(m2,45),2)) - 1;

n=[24 8 -8 -24]; % 4相循环移位
for i=1:length(n)
    goldcdsk1(i,:)=circshift(gold(1,:),n(i));
    goldcdsk2(i,:)=circshift(gold(2,:),n(i));
    goldcdsk3(i,:)=circshift(gold(3,:),n(i));
    goldcdsk4(i,:)=circshift(gold(4,:),n(i));
end

for i=1:length(n) % BPSK
    goldcdsk1(i+4,:)= -goldcdsk1(i,:);
    goldcdsk2(i+4,:)= -goldcdsk2(i,:);
    goldcdsk3(i+4,:)= -goldcdsk3(i,:);
    goldcdsk4(i+4,:)= -goldcdsk4(i,:);
end

gold4cdsk=[goldcdsk1;goldcdsk2;goldcdsk3;goldcdsk4];

```

## 选码

```

%% 选择扩频码
ssmes=zeros(length(bimes),length(gold(1,:)));
for i=1:length(bimes)
    ssmes(i,:)=gold4cdsk(bimes(i,1)+1,:);
end
ssmes=ssmes';
ssmes=reshape(ssmes,1,[]);

```

重点在解码，首先找出一个buf，跟原始gold序列做互相关（注意，由于是相位携带信息，如果互相关顺序颠倒，整个相位信息会颠倒，误码率为1，不信你可以试试）

```

%% 相关解码
N=length(ssmes)/L; % 消息个数
en=zeros(1,N); % 存储解码
cor=[]; % 存储互相关函数作图
M=size(gold,1); % M元扩频码个数

for i=1:N
    buf=ssmes(1,1+(i-1)*L:i*L); % 选取窗口
    corbuf=zeros(M,2*L-1); % 自相关矩阵缓存
    for j=1:M % 跟原始扩频码做互相关(注意顺序)
        corbuf(j,:)=xcorr(gold(j,:),buf);
    end
    cor=[cor corbuf]; % 存储相关矩阵
end

```

在互相关矩阵里找出正峰值或负峰值的位置



```

xxx=[]; % 存储互相关函数峰值的位置和峰值
for i=1:N
    buf=cor(1:4,1+(i-1)*125:i*125); % 选取窗口
    if(max(buf(:))<abs(min(buf(:)))) % 判断极性，如果负极性找负峰值
        [max_val, linear_idx] = min(buf(:));
    else % 反之，正极性找正峰值
        [max_val, linear_idx] = max(buf(:));
    end
    [row, col] = ind2sub(size(buf), linear_idx); % 转成行列形式
    xxx(i,:)=[row,col,max_val];
end

```

根据扩频码和互相关峰值位置与极性，解码

```

encode=[];
for i=1:N % 根据扩频码和互相关峰值位置与极性，解码
    if(xxx(i,2)<47)
        shift=0;
    elseif(xxx(i,2)<63)
        shift=1;
    elseif(xxx(i,2)<79)
        shift=2;
    else
        shift=3;
    end
    if(xxx(i,3)<0)
        sig=4;
    else
        sig=0;
    end
    encode(i,1)=(xxx(i,1)-1)*8+sig+shift;
end
encode=encode';
ec=de2bi(encode)';
dec=reshape(ec,1,[]);

```

## 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] 于洋.高效率水声扩频通信技术研究[D].哈尔滨工程大学,2014.