文档说明

项目地址：https://github.com/LvGitHub-9/SpreadSpectrumCommunication
名称：码片时间与扩频增益
说明：研究扩频中的码片时间与扩频带宽的关系，说明扩频增益计算原理
版本：V1.0
作者: 小吕同学
修改记录：

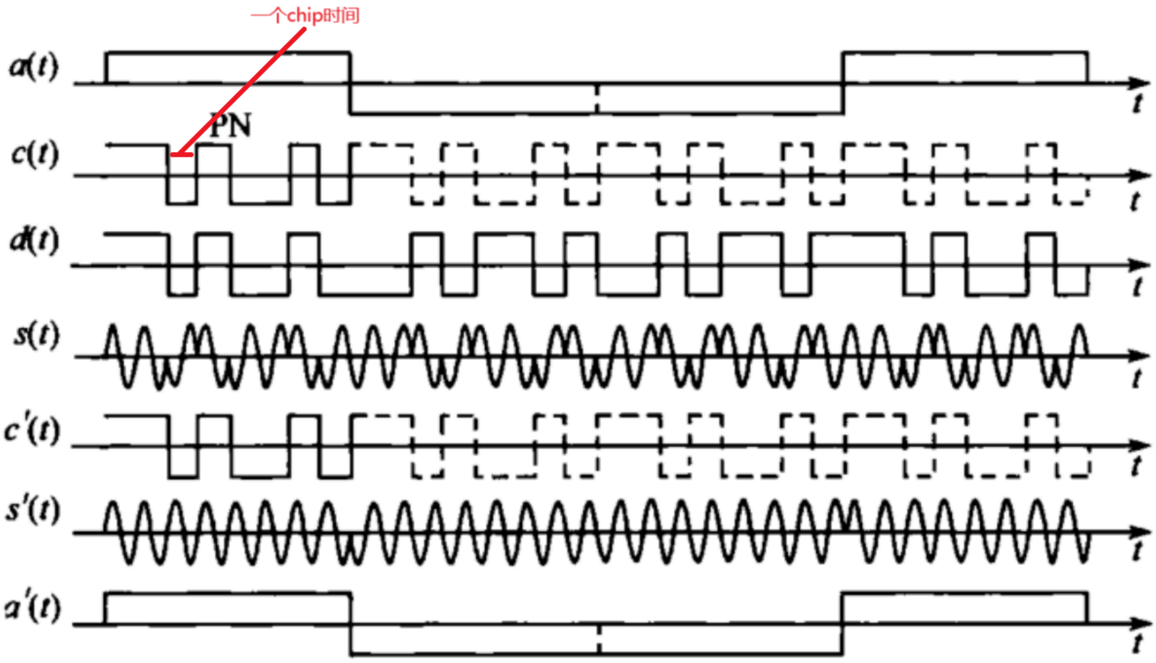
版本号	日期	作者	说明
V1.0	2025-1-3	小吕同学	首次发布

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

扩频中的码片时间

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。

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构成的二元码流，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）。



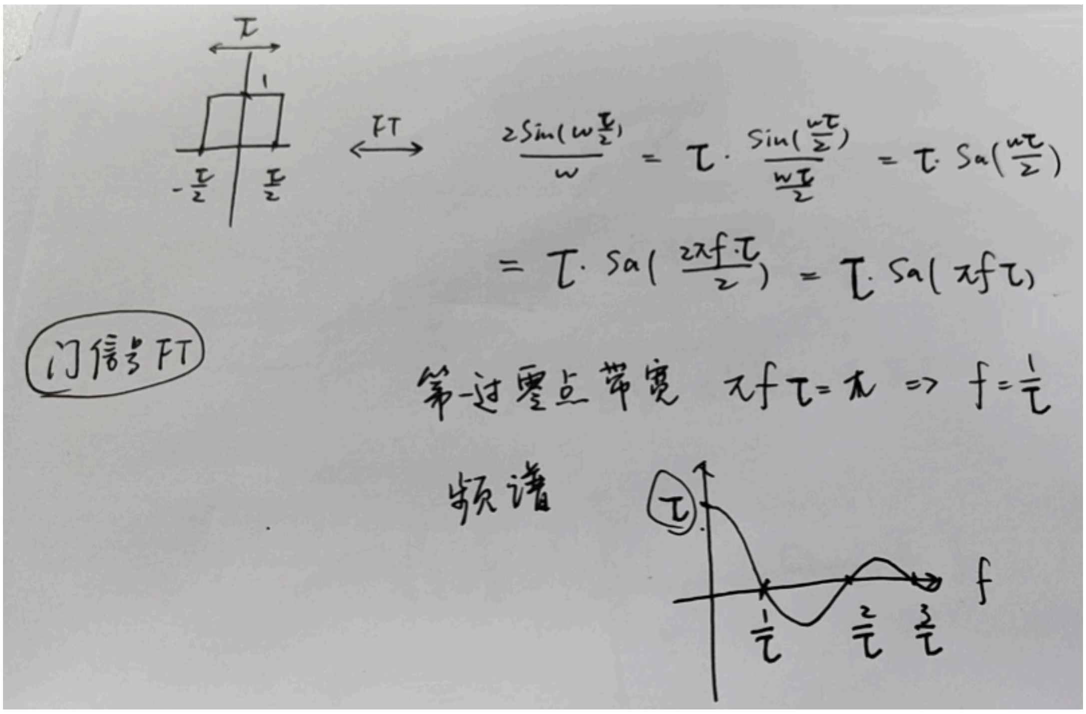
码片时间有什么用?

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 $x(t)=\begin{cases}1, & \lvert t\rvert<T_1\\ 0, & T_1<\lvert t\rvert\leqslant\frac{T}{2}\end{cases}$ 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
$x(t)\begin{cases}1, & \lvert t\rvert<T_1\\ 0, & \lvert t\rvert>T_1\end{cases}$	$\frac{2\sin\omega T_1}{\omega}$	—
$\frac{\sin Wt}{\pi t}$	$X(j\omega)=\begin{cases}1, & \lvert\omega\rvert<W\\ 0, & \lvert\omega\rvert>W\end{cases}$	—
$\delta(t)$	1	—
$u(t)$	$\frac{1}{j\omega}+\pi\delta(\omega)$	—
$\delta(t-t_0)$	$e^{-j\omega t_0}$	—
$e^{-at}u(t)$，$\mathcal{Re}\{a\}>0$	$\frac{1}{a+j\omega}$	—
$te^{-at}u(t)$，$\mathcal{Re}\{a\}>0$	$\frac{1}{(a+j\omega)^2}$	—
$\frac{t^{n-1}}{(n-1)!}e^{-at}u(t)$，$\mathcal{Re}\{a\}>0$	$\frac{1}{(a+j\omega)^n}$	—

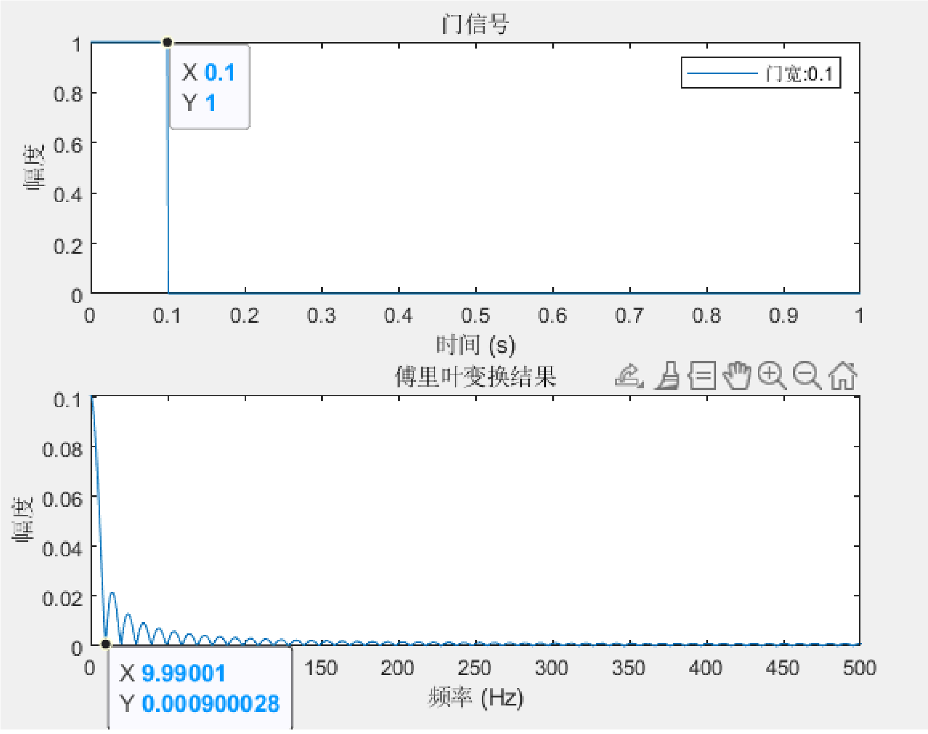
在这里说明，门信号的傅里叶变换带宽是第一过零点带宽（以下称为带宽），通过公式推导可以发现，门信号时域的门宽和带宽成倒数关系。



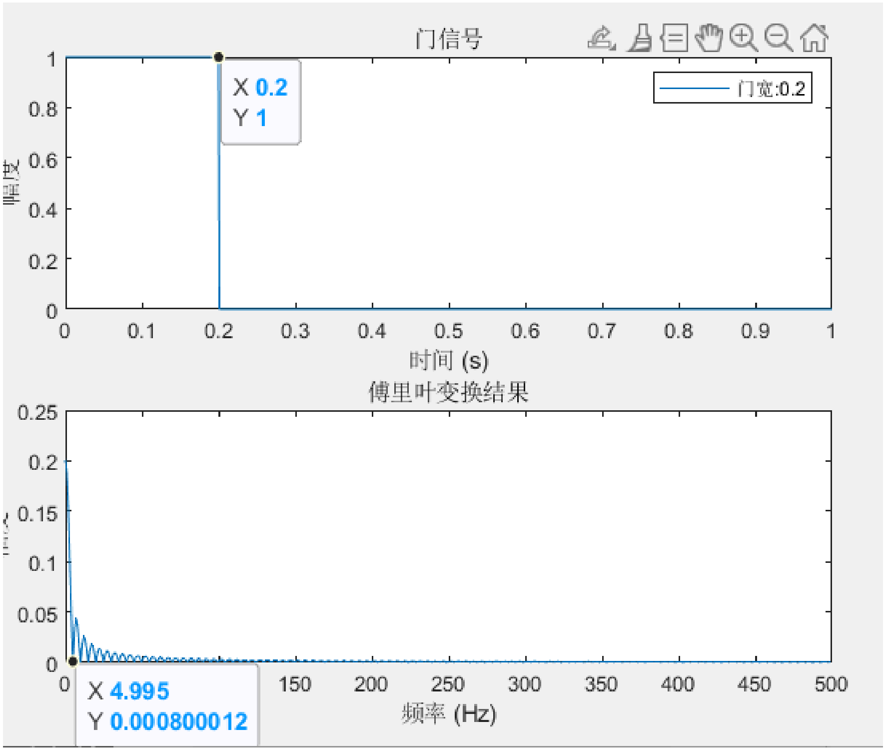
我们也可以从Matlab上写程序来查看这一关系。

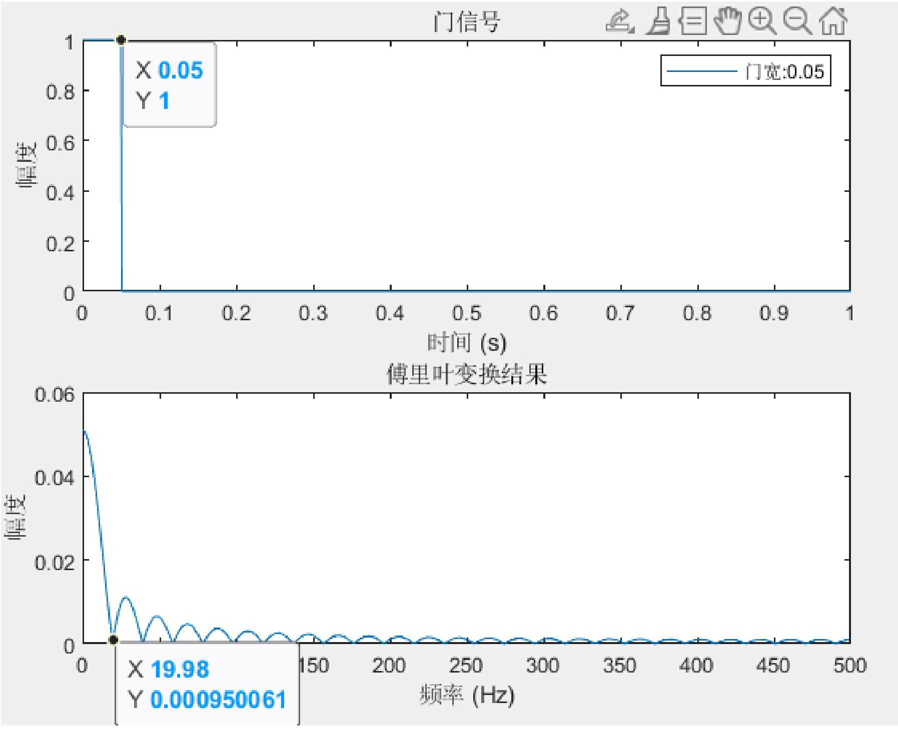
```
%% chip（Kimi.ai生成的代码）
close all; clear; clc;

%% 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]);
```

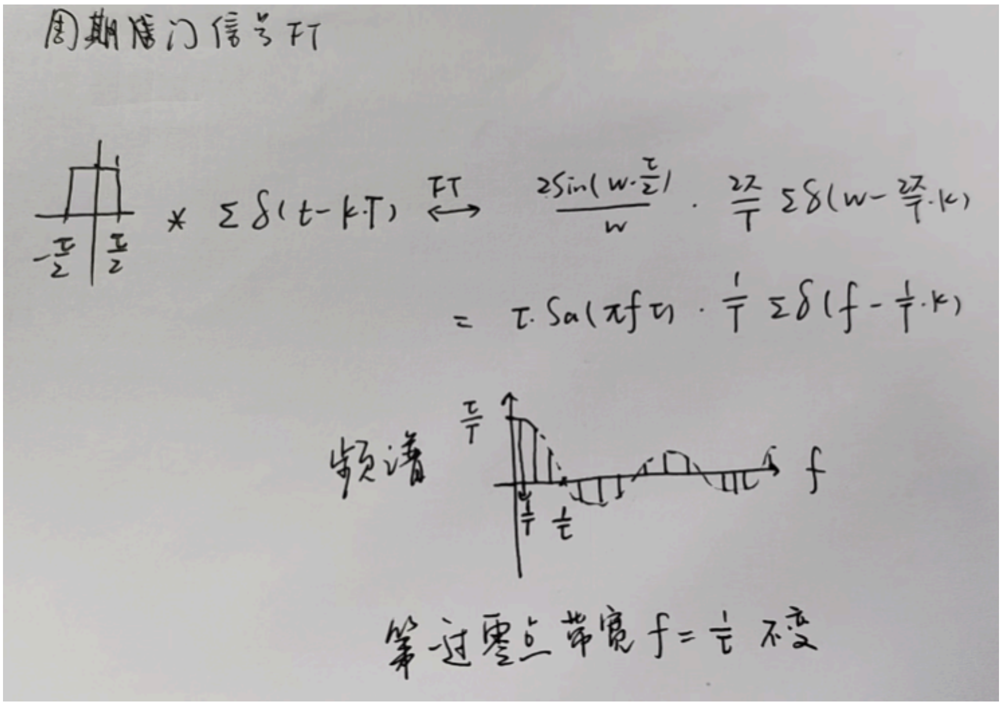


修改门宽的大小，观察带宽变化。





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。

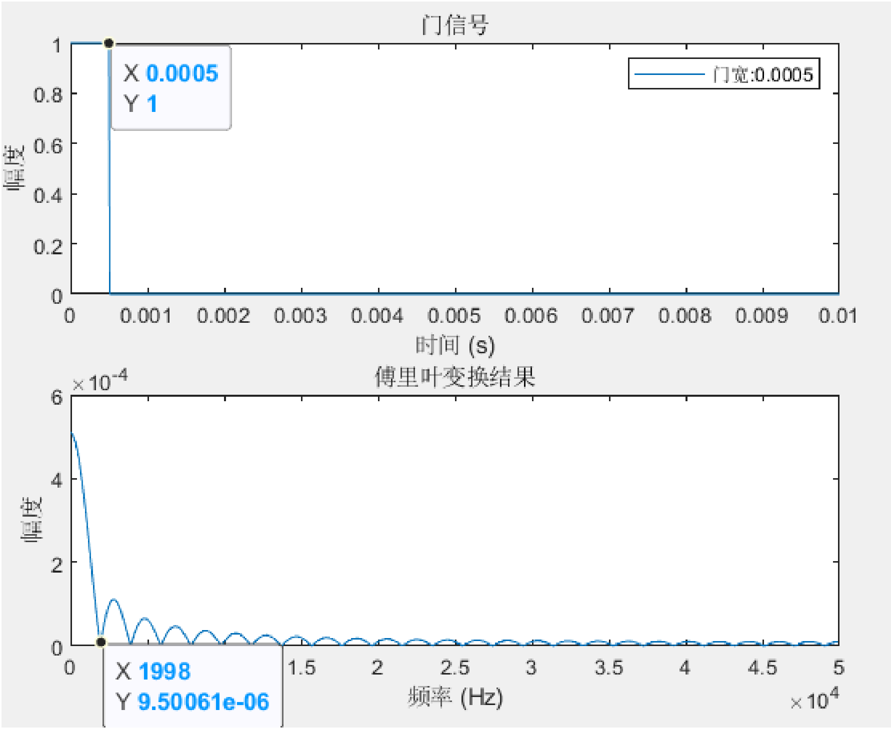


如何选取扩频的码片时间?

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

```
%% chip（Kimi.ai生成的代码）
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

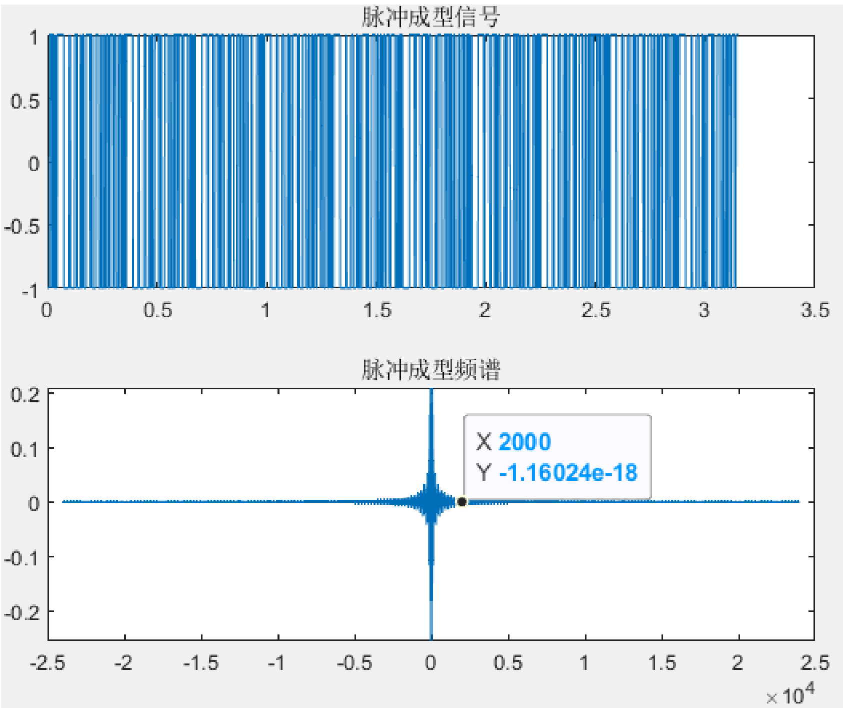
参考修改过后的DSSS.m

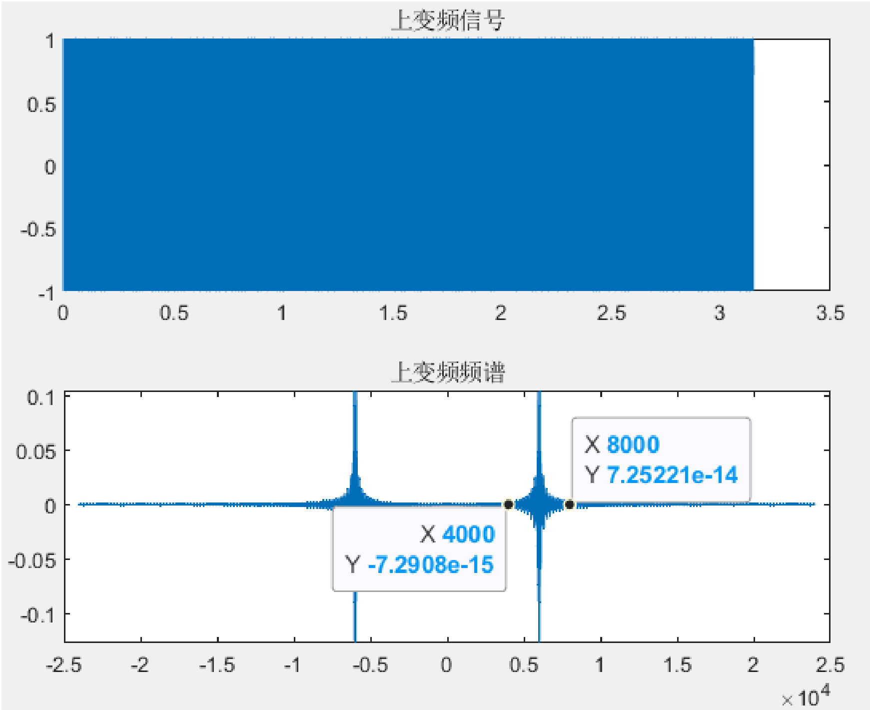
```
%% 直接序列扩频
close all; clear; clc;

%% 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
% 假设信号使用带宽为4-8kHz，基带信号带宽为2kHz，chip长度为1/2kHz=0.5ms
% 信号发送频率为48kHz，0.5ms能够发送0.5ms*48kHz=240个符号
% 即一个码片(chip)长度为240
rect=240;
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
dmmes=mmes.*cos(2*pi*fc*t);     % 解调

%% 低通滤波
```

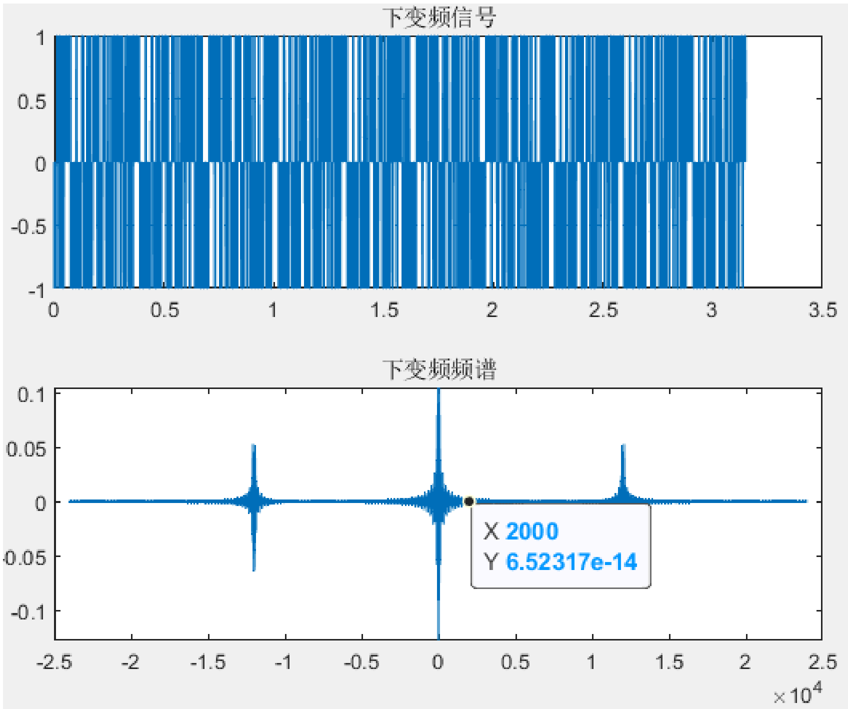
```
Delay = 32;                     % 32阶滤波器
fircoef = fir1(2*Delay,fb/fs);
lpf = filter(fircoef,1,[dmmes zeros(1,Delay)]);
fmes = lpf(Delay+1:end);

%% 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]);

%% 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);
```

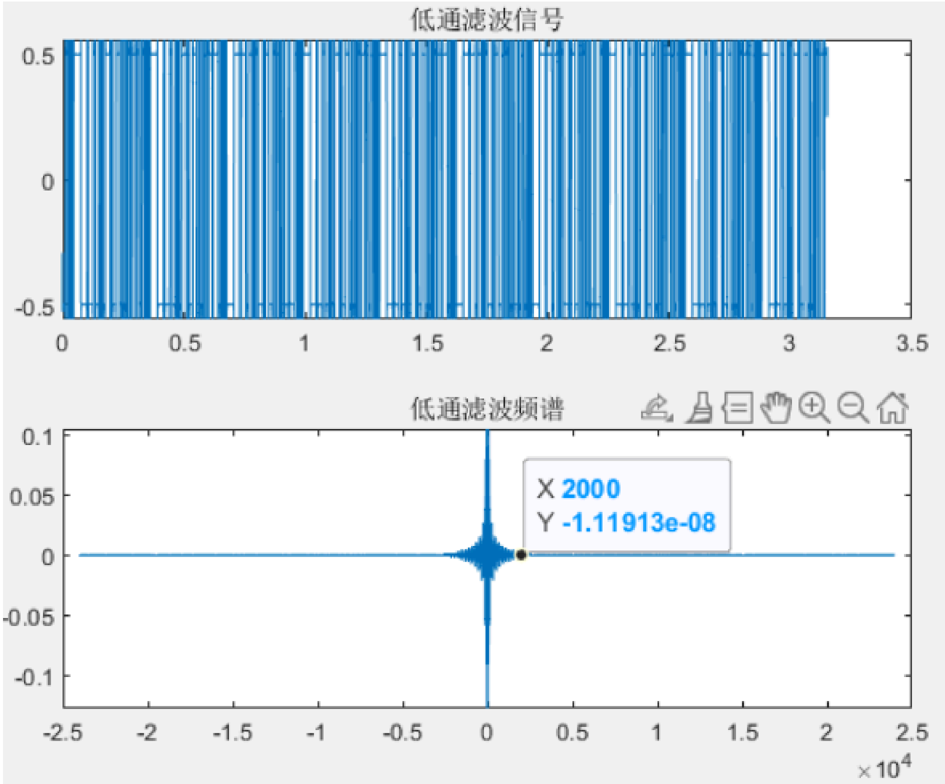
```
plot(f,fftshift(rmes_fft));title('rmes_fft');
title('脉冲成型频谱');

%% 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(dmmes)/fs;
figure
subplot(2,1,1);
plot(t,dmmes);title('dmmes');
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:




上变频信号
上变频频谱
X 8000
Y 7.25221e-14
X 4000
Y -7.2908e-15
$\times 10^4$


下变频信号
下变频频谱
X 2000
Y 6.52317e-14
$\times 10^4$


低通滤波信号
0.5
0
-0.5
0
0.5
1
1.5
2
2.5
3
3.5
低通滤波频谱
0.1
0.05
0
-0.05
-0.1
X 2000
Y -1.11913e-08
-2.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5
$\times 10^4$

扩频增益计算

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Tb，信号带宽Fb=1/Tb。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Tb会被分为63个小的码片时间Tc，Tc=Tb/63，那么扩频之后的带宽就是1/Tc=63/Tb，带宽自然就是原来的63倍了，扩频增益为63。

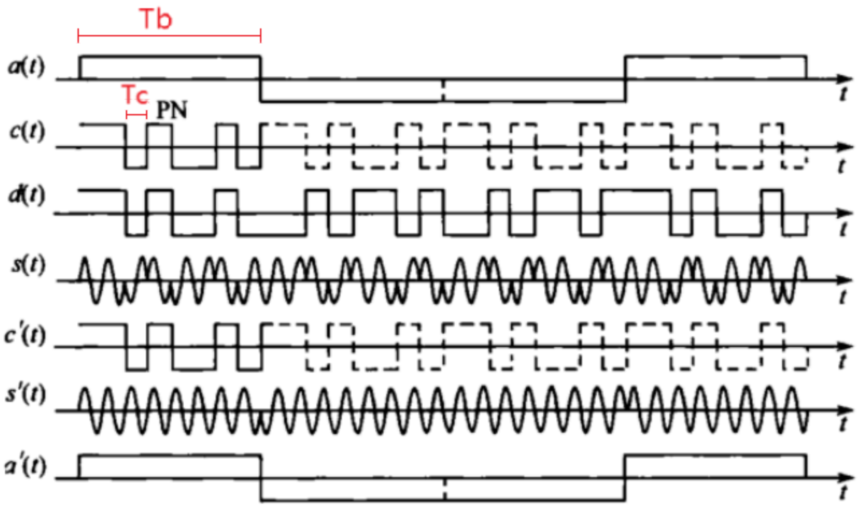


图 4.2 DS/BPSK 调制系统示例图

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

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

Gp是扩频增益，B是扩频信号带宽，Bm是原始信号带宽。当扩频码长为63，Gp=10*lg（63）=17.99dB。

参考资料

[1] Alan V. Oppenheim, Alan S. Willsky, S. Hamid Nawab著.信号与系统.电子工业出版社.2013