数字通信 seminar5 报告

第六组

GPS 简介

GPS 分为 3 个部分，如下所示[1]：

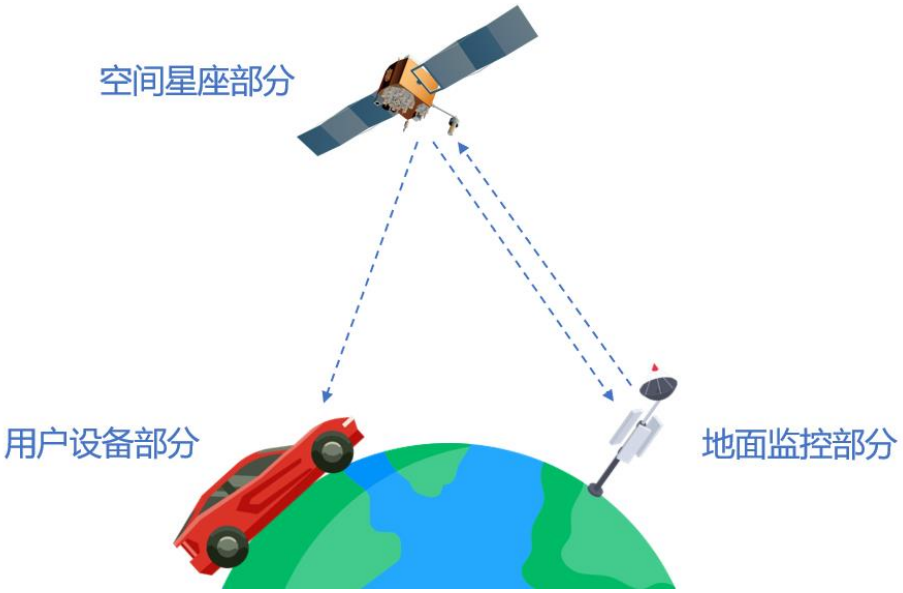

图1 GPS 组成示意

其中，卫星与地面站双向通信；而与用户设备只能卫星广播给用户设备。

GPS 通过三遍测量法计算位置，然而由于钟差（即时钟不同步），需要 4 颗卫星才能准确定位。

---

[1] 谢钢, GPS 原理与接收机设计[M], 2017 年 1 月第 1 版, 电子工业出版社，2017.1:2.

GPS 向用户设备发送的消息如下：

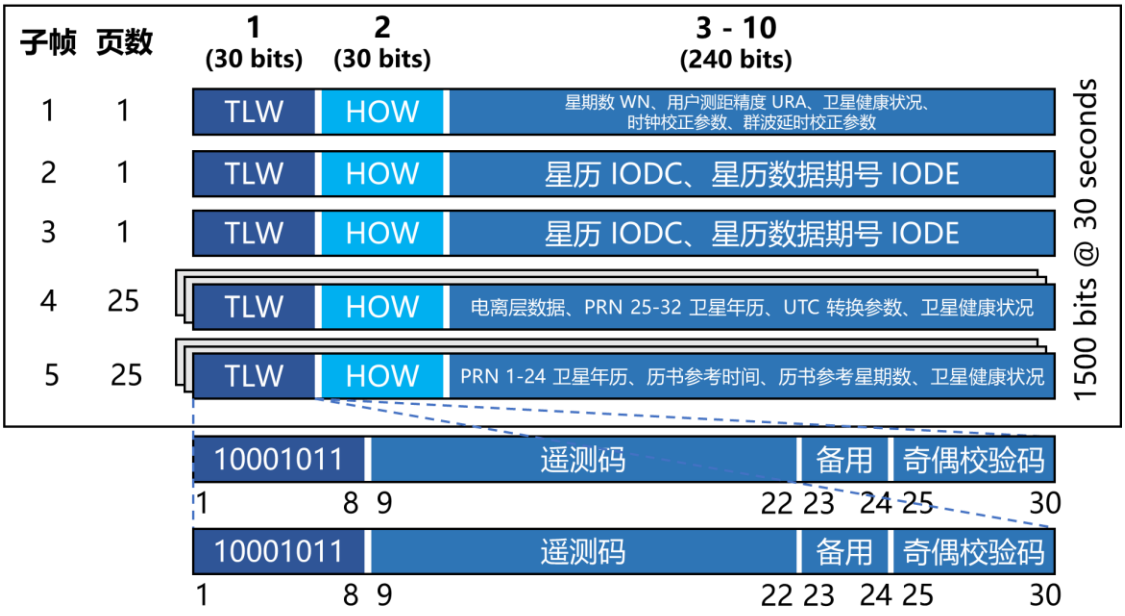

图2 GPS 导航报文内容

发送时，每次发送 5 帧。

GPS 信号发送

简单来讲，GPS 数据的发送与接收分为以下几个步骤：

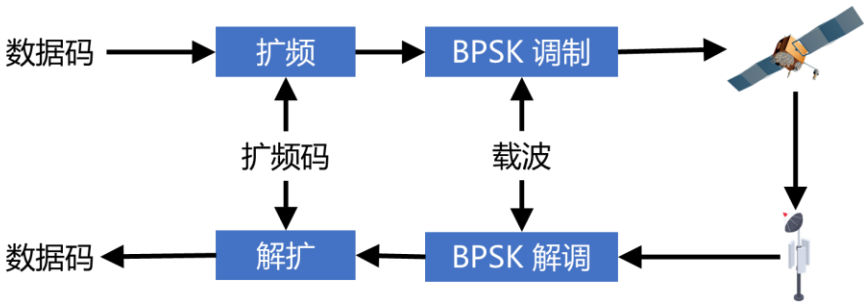

图3 GPS 收发步骤

如图所示，在发送时首先进行扩频，然后调制（通常是 BPSK）。接收时正好相反，先解调再解扩。

调制

在调制中主要使用了三种载波（L1、L2、L5），如图所示[2]。

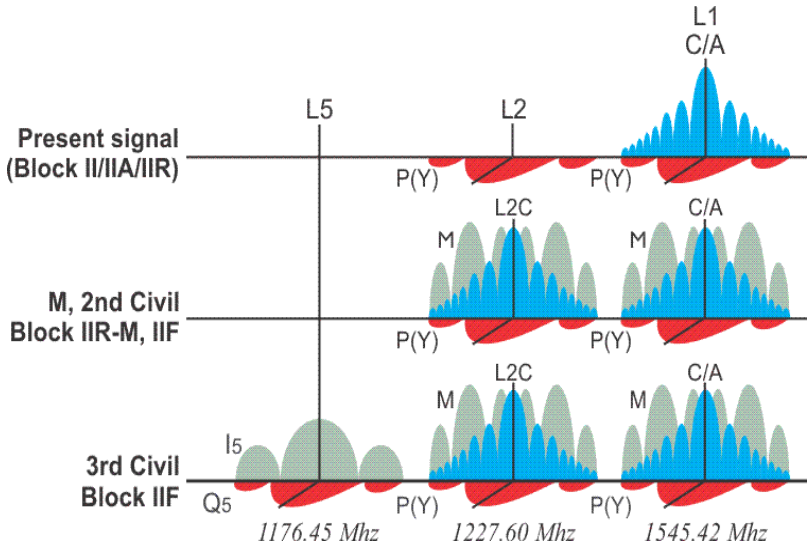

图4 GPS 占用带宽

L1、L2、L5 的中心频率均为原子钟核心频率 $f_0 = 10.23$ MHz 的整数倍[3][4]。具体的，有

- $f_1 = 154 f_0 = 1575.42$ MHz
- $f_2 = 120 f_0 = 1227.60$ MHz
- $f_5 = 115 f_0 = 1176.45$ MHz

这样做的好处在于更加容易生成载波信号。

扩频

GPS 主要使用了三种扩频码：P 码、C/A 码、M 码。其中，M 码为军用，缺少相关资料。

---

[2] Steve Lazar, GPS for Land Surveyors[M], Aerospace Corporation, Summer 2002.

[3] DCarlson A.B., Communication Systems: An Introduction to Signals and Noise in Electrical Communication,Third Edition, McGraw-Hill Inc., 1986.

[4] Erickson D., Taylor C., “Pacify the Power: GPS Harness for Large-Area Electrical Grid,” GPS World, April2005.

P 码（Precise code）频率为 10.23 Mbps，运行在 L1 和 L2 上。P 码总长度较长，共有 2×10^{14} bit，发送需要 37 周。每个卫星（1-32）和地面站（33-37）都被分配了其中的一部分，该部分周期为 7 天，对应的是星历更新的周期[5678]。

P 码采用的是相位调制，如图所示：

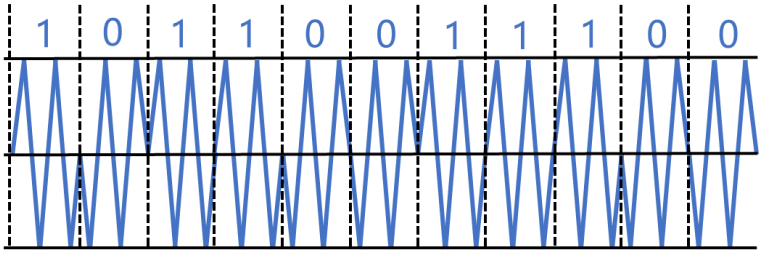

图5 P 码调制方式示意

加密过后的 P 码被称为 P(Y) 码。其区别是，P(Y) 码在第四个子帧中有一个标志告诉接收器扩频码已被加密，一次来防止欺骗。

C/A 码（Civilian Acquisition code）顾名思义为民用码。他的速率为 P 码的十分之一，即 1.023 Mbps。每颗卫星的 C/A 码有 1023 位，每毫秒即可广播一次。它运行在 L1 上，旨在提供标准定位服务（SPS）。作为对比，P(Y) 码则提供的是精准定位服务（PPS）。

在设计之初，美国低估了 C/A 码的定位精准度，于是通过选择可用性（SA）故意降低到 95% 的时间为水平 ±100 米、垂直 ±175 米精度。这一措施直到 21 世纪初才取消。

P 码和 C/A 码的生成都利用的是线性移位反馈寄存器（LSFR）。我们以 C/A 码为例。

---

[5] Maxim Integrated Products Inc., “An Introduction to Direct-Sequence Spread-Spectrum Communications, ”Application Note 1890, February 18, 2003.

[6] Pacific Crest Corporation, “The Guide to Wireless GPS Data Links, ” Santa Clara, CA, September 2000.

[7] Swider R., “GPS Policy Update, ” Civil GPS Service Interface Committee, Department of Defense, Salt LakeCity, UT, September 9-11, 2001.

[8] Peterson R., Ziemer R., Borth D., Introduction to Spread Spectrum Communications, Prentice Hall, 1995.

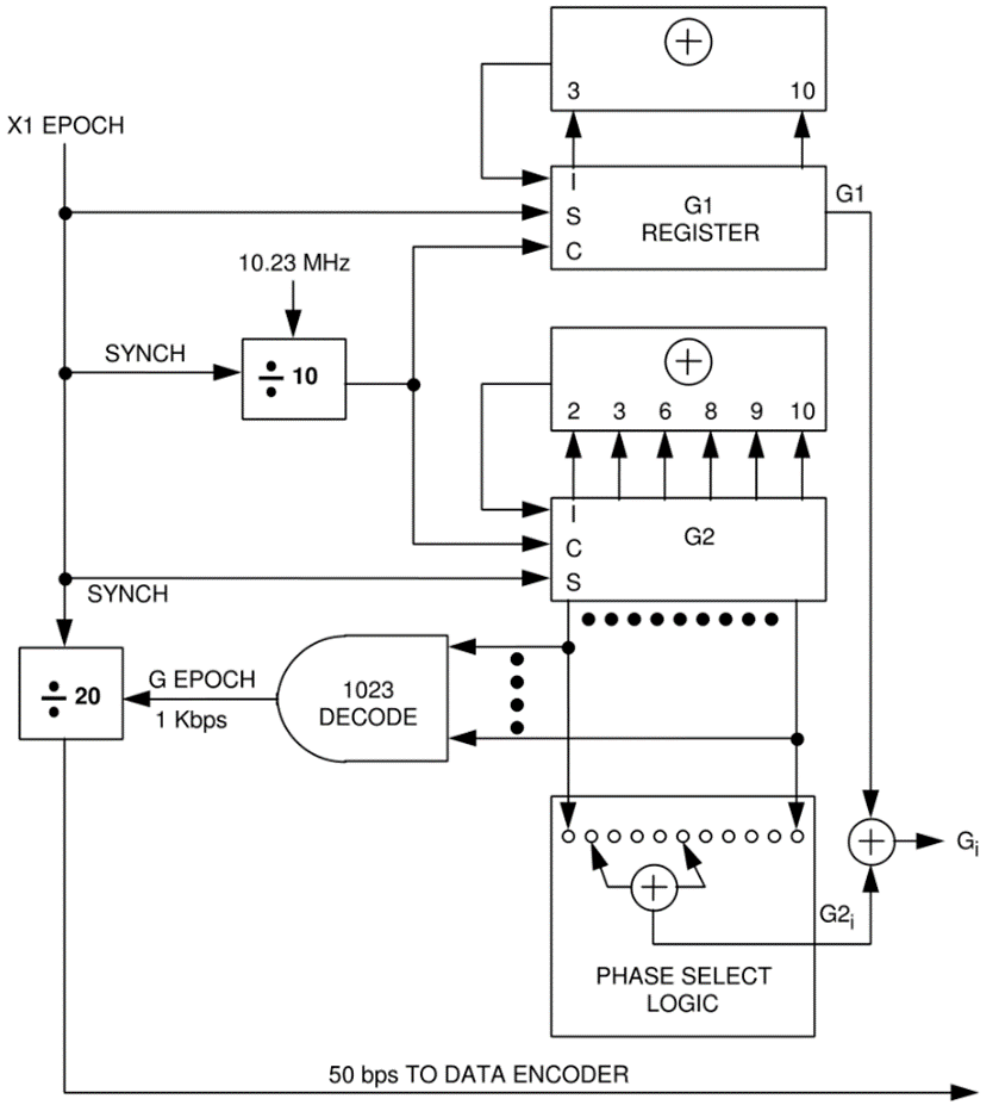

图6 C/A 码生成实现

如图是一个典型的 C/A 码生成器结构。其中，G1 和 G2 为 LSFR。输出时，通过相位选择器选择 G2 中的两个位置做异或，所得结果再与 G1 最后一位异或，即可得到扩频码。相位选择器选择的位置共有 37 种，每个卫星或地面站均不相同。

G1 和 G2 的具体原理如下所示：

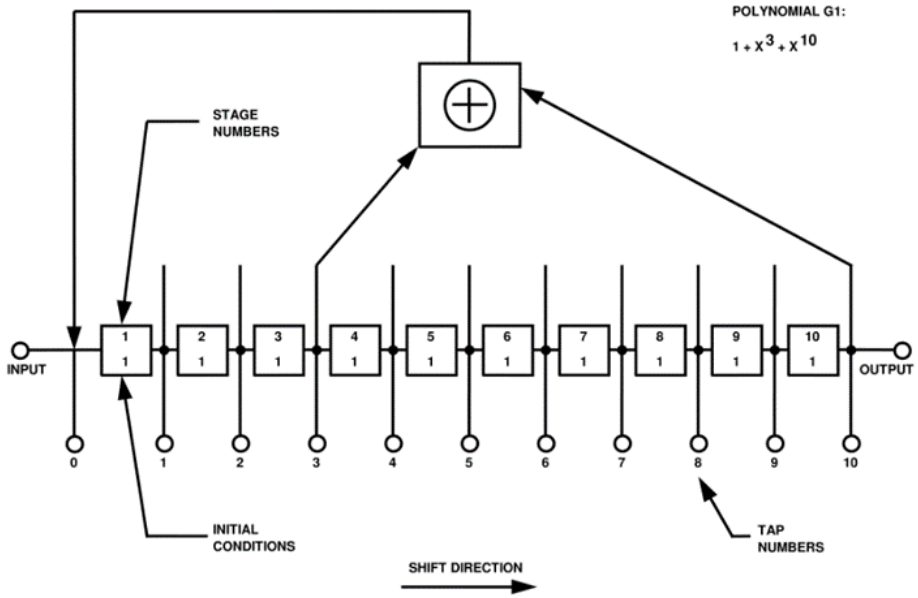

图7 C/A 码G1 LSFR

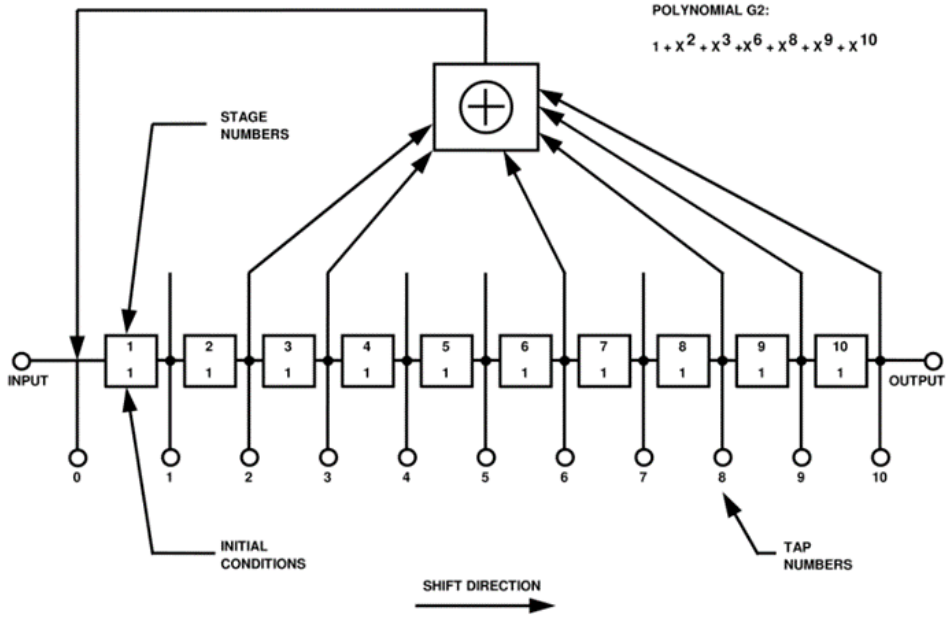

图8 C/A 码G2 LSFR

他们都利用的是有限域中的运算。其中，

- G1 为 $1 + x^3 + x^{10}$
- G2 为 $1 + x^2 + x^3 + x^6 + x^8 + x^9 + x^{10}$

根据文档，G2 寄存器设置有初始值和时延。具体的值如下所示：

Table 3-Ia. Code Phase Assignments (sheet 1 of 2)

SV ID No.	GPS PRN Signal No.	Code Phase Selection		Code Delay Chips		First 10 Chips Octal* C/A	First 12 Chips Octal P
		C/A($G2_i$)***	($X2_i$)	C/A	P		
1	1	2 ⊕ 6	1	5	1	1440	4444
2	2	3 ⊕ 7	2	6	2	1620	4000
3	3	4 ⊕ 8	3	7	3	1710	4222
4	4	5 ⊕ 9	4	8	4	1744	4333
5	5	1 ⊕ 9	5	17	5	1133	4377
6	6	2 ⊕ 10	6	18	6	1455	4355
7	7	1 ⊕ 8	7	139	7	1131	4344
8	8	2 ⊕ 9	8	140	8	1454	4340
9	9	3 ⊕ 10	9	141	9	1626	4342
10	10	2 ⊕ 3	10	251	10	1504	4343
11	11	3 ⊕ 4	11	252	11	1642	|
12	12	5 ⊕ 6	12	254	12	1750	|
13	13	6 ⊕ 7	13	255	13	1764	|
14	14	7 ⊕ 8	14	256	14	1772	|
15	15	8 ⊕ 9	15	257	15	1775	|
16	16	9 ⊕ 10	16	258	16	1776	|
17	17	1 ⊕ 4	17	469	17	1156	|
18	18	2 ⊕ 5	18	470	18	1467	|
19	19	3 ⊕ 6	19	471	19	1633	4343
20	20	4 ⊕ 7	20	472	20	1715	4343
21	21	5 ⊕ 8	21	473	21	1746	|
22	22	6 ⊕ 9	22	474	22	1763	|
23	23	1 ⊕ 3	23	509	23	1063	|
24	24	4 ⊕ 6	24	512	24	1706	|
25	25	5 ⊕ 7	25	513	25	1743	|
26	26	6 ⊕ 8	26	514	26	1761	|
27	27	7 ⊕ 9	27	515	27	1770	|
28	28	8 ⊕ 10	28	516	28	1774	|
29	29	1 ⊕ 6	29	859	29	1127	|
30	30	2 ⊕ 7	30	860	30	1453	|
31	31	3 ⊕ 8	31	861	31	1625	|
32	32	4 ⊕ 9	32	862	32	1712	|
65	33***	5 ⊕ 10	33	863	33	1745	|
66	34**	4 ⊕ 10	34	950	34	1713	|
67	35	1 ⊕ 7	35	947	35	1134	|
68	36	2 ⊕ 8	36	948	36	1456	|
69	37**	4 ⊕ 10	37	950	37	1713	4343

图9 C/A 码部分参数

不过，初始值和时延不是必须的。在没有时延的情况下，如果将初始值全部置 1，最终的结果相同。

MATLAB 仿真代码如下：

```
function g = C_A_Code(num)
    taps = [2 ,6 ; 3 ,7 ; 4 ,8 ; 5 ,9 ; 1 ,9 ;
```

```
         2 ,10; 1 ,8 ; 2 ,9 ; 3 ,10; 2 ,3 ;
         3 ,4 ; 5 ,6 ; 6 ,7 ; 7 ,8 ; 8 ,9 ;
         9 ,10; 1 ,4 ; 2 ,5 ; 3 ,6 ; 4 ,7 ;
         5 ,8 ; 6 ,9 ; 1 ,3 ; 4 ,6 ; 5 ,7 ;
         6 ,8 ; 7 ,9 ; 8 ,10; 1 ,6 ; 2 ,7 ;
         3 ,8 ; 4 ,9 ; 5 ,10; 4 ,10; 1 ,7 ;
         2 ,8 ; 4 ,10
    ];
g2s = [1, 1, 0, 0, 1, 0, 0, 0, 0, 0; % 1440
       1, 1, 1, 0, 0, 1, 0, 0, 0, 0; % 1620
       1, 1, 1, 1, 0, 0, 1, 0, 0, 0; % 1710
       1, 1, 1, 1, 1, 0, 0, 1, 0, 0; % 1744
       1, 0, 0, 1, 0, 1, 1, 0, 1, 1; % 1133
       1, 1, 0, 0, 1, 0, 1, 1, 0, 1; % 1455
       1, 0, 0, 1, 0, 1, 1, 0, 0, 1; % 1131
       1, 1, 0, 0, 1, 0, 1, 1, 0, 0; % 1454
       1, 1, 1, 0, 0, 1, 0, 1, 1, 0; % 1626
       1, 1, 0, 1, 0, 0, 0, 1, 0, 0; % 1504
       1, 1, 1, 0, 1, 0, 0, 0, 1, 0; % 1642
       1, 1, 1, 1, 1, 0, 1, 0, 0, 0; % 1750
       1, 1, 1, 1, 1, 1, 0, 1, 0, 0; % 1764
       1, 1, 1, 1, 1, 1, 1, 0, 1, 0; % 1772
       1, 1, 1, 1, 1, 1, 1, 1, 0, 1; % 1775
       1, 1, 1, 1, 1, 1, 1, 1, 1, 0; % 1776
       1, 0, 0, 1, 1, 0, 1, 1, 1, 0; % 1156
       1, 1, 0, 0, 1, 1, 0, 1, 1, 1; % 1467
       1, 1, 1, 0, 0, 1, 1, 0, 1, 1; % 1633
       1, 1, 1, 1, 0, 0, 1, 1, 0, 1; % 1715
       1, 1, 1, 1, 1, 0, 0, 1, 1, 0; % 1746
       1, 1, 1, 1, 1, 1, 0, 0, 1, 1; % 1763
       1, 0, 0, 0, 1, 1, 0, 0, 1, 1; % 1063
       1, 1, 1, 1, 0, 0, 0, 1, 1, 0; % 1706
       1, 1, 1, 1, 1, 0, 0, 0, 1, 1; % 1743
       1, 1, 1, 1, 1, 1, 0, 0, 0, 1; % 1761
       1, 1, 1, 1, 1, 1, 1, 0, 0, 0; % 1770
       1, 1, 1, 1, 1, 1, 1, 1, 0, 0; % 1774
       1, 0, 0, 1, 0, 1, 0, 1, 1, 1; % 1127
       1, 1, 0, 0, 1, 0, 1, 0, 1, 1; % 1453
       1, 1, 1, 0, 0, 1, 0, 1, 0, 1; % 1625
       1, 1, 1, 1, 0, 0, 1, 0, 1, 0; % 1712
       1, 1, 1, 1, 1, 0, 0, 1, 0, 1; % 1745
       1, 1, 1, 1, 0, 0, 1, 0, 1, 1; % 1713
       1, 0, 0, 1, 0, 1, 1, 1, 0, 0; % 1134
       1, 1, 0, 0, 1, 0, 1, 1, 1, 0; % 1456
       1, 1, 1, 1, 0, 0, 1, 0, 1, 1; % 1713
    ];
delay = [5  , 6  , 7  , 8  , 17 , ...
         18 , 139, 140, 141, 251, ...
         252, 254, 255, 256, 257, ...
         258, 469, 470, 471, 472, ...
```

```
            473, 474, 509, 512, 513, ...
            514, 515, 516, 859, 860, ...
            861, 862, 863, 950, 947, ...
            948, 950
        ];
n = 10; % size of LFSR
L = 2^n - 1; % number of chips in a code

g1_sel = [0 0 1 0 0 0 0 0 0 1]; % G1 generator
g1 = ones(1, n); % G1 vector
g2_sel = [0 1 1 0 0 1 0 1 1 1]; % G2 generator
g2 = g2s(num, :); % G2 vector

tap = taps(num, :);

for i = 1 : (delay(num) - 1)
    g2 = [mod(sum(g2 .* g2_sel), 2), g2(1:n - 1)];
end

for i = 1 : L
    g(i) = mod(g1(n) + mod(sum(g2(tap)), 2), 2);
    g1 = [mod(sum(g1 .* g1_sel), 2), g1(1 : n - 1)];
    g2 = [mod(sum(g2 .* g2_sel), 2), g2(1 : n - 1)];
end
```

利用程序，我们可以生成第一个和第二个 C/A 码：

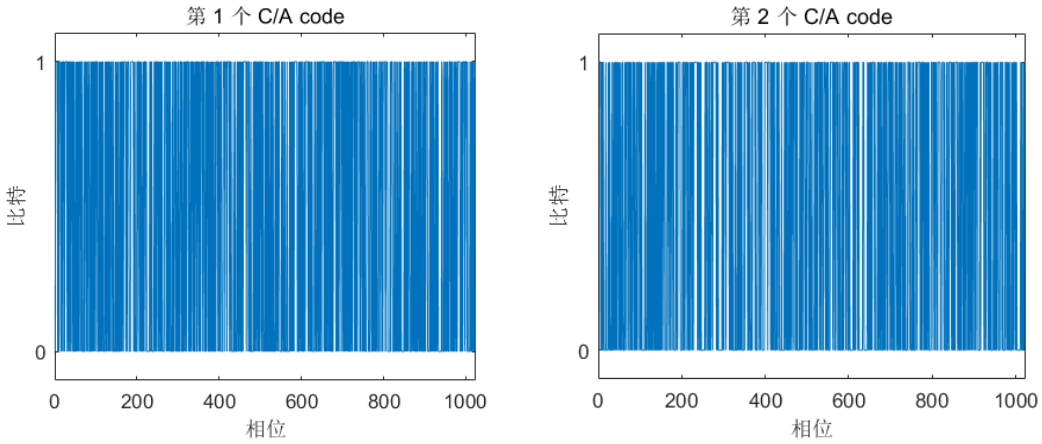

图10 第1 和第2 个 C/A 码

当然，C/A 码具体的值不是我们所关心的。我们需要关心的是它的相关性。

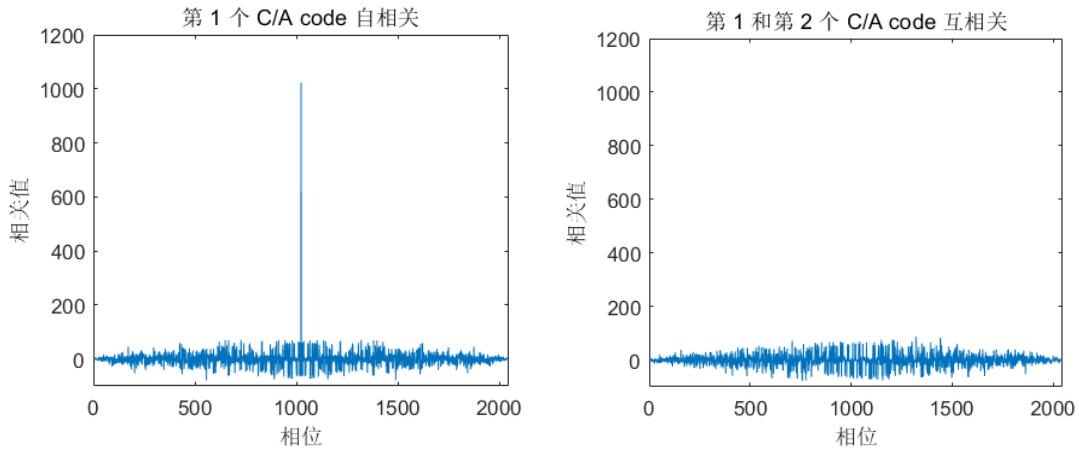

图 11 C/A 码自相关与互相关比较

左图为第 1 个 C/A code 自相关，右图为第 1 和第 2 个 C/A code 互相关。它们的明显区别在于：自相关有着非常突出的峰值，而互相关则没有。因此，我们可以通过识别峰值来找到正确的 C/A 码。

我们将第 1 个 C/A code 做相移（即，将尾部的一部分移动到头部），再与原来的 C/A 码做互相关，得到下图：

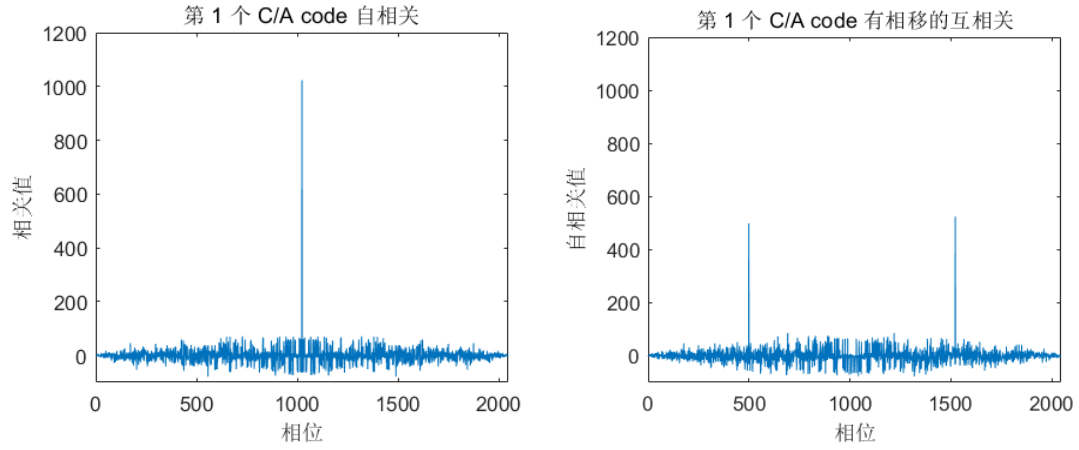

图 12 C/A 码相移后相关性比较

通过对比可以发现，尽管相移后仍然有明显的峰值，但峰值处的相关值降低了。我们做出了峰值与相移大小的关系图：

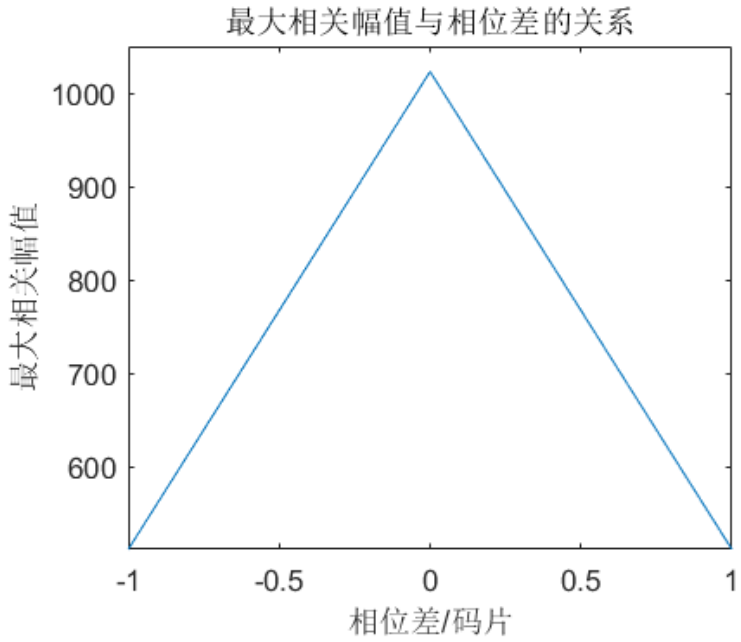

图13 C/A 码最大幅度值与相位差关系

由图可以看出，它是一个完美的等腰三角形。在没有相移的情况下，相关值最大。

GPS 信号接收

接收机信号处理分为 4 个步骤[9]，如下所示：

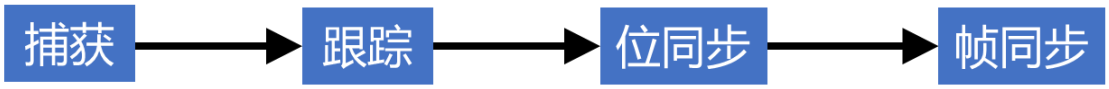

图14 GPS 信号接收流程

其中，跟踪分为载波环（用于解调）和码环（用于解扩）。这里，我们只讨论与主题有关的码环。

根据上文得到的结论，在解扩时，我们尝试使用不同的 C/A 码解扩，如果看到明显的峰值，则表明选对了 C/A 码。选择正确的 C/A

[9] Abraham C., Fuchs D., “Method and Apparatus for Computing Signal Correlation at Multiple Resolutions, ”US Patent 6704348, March 9, 2004.

码后，需要使用峰值与相移大小的关系来使得相位对齐。具体做法如下图所示[10][11][12][13]：

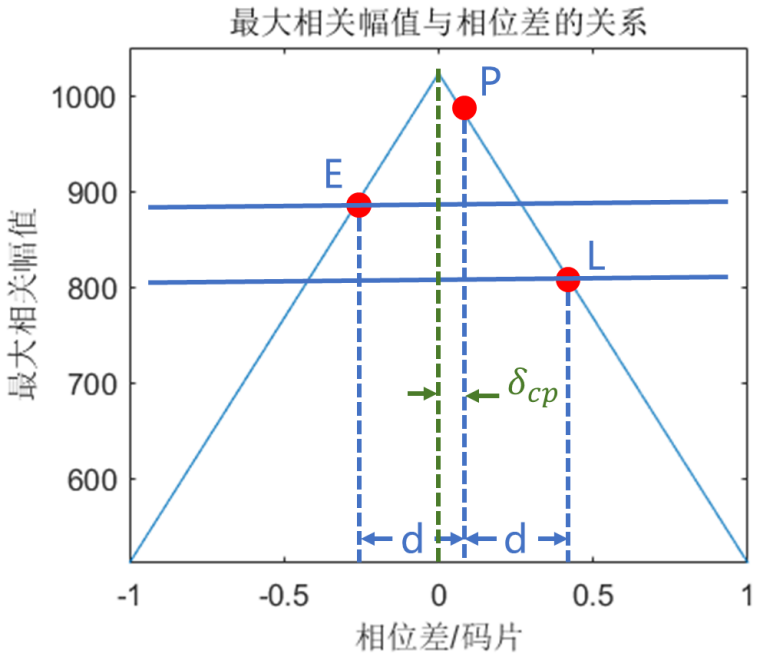

图 15 确定最大幅度的方法

假设当前随意选取的相位为 P，其与最大值对应的相位差为 δ_{cp}。我们将其向前移动 d 个相位到点 E，再向后移动 d 个相位到点 L。根据等腰三角形的性质可以得出，如果 E 和 L 的值相等，则 P 点在最大值，如下图所示。

---

[10] Agarwal N., Basch J., Beckman P., Bharti P., Bloebaum S., Casadei S., Chou A., Enge P., Fong W., Hathi N.,Mann W., Sahai A., Stone J., Tsitsiklis J., Van Roy B., “Algorithms for GPS Operation Indoors and

[11] Alban S., Akos D., Rock S., Gebre-Egziabher D., “Performance Analysis and Architectures for INS-Aided GPS Tracking Loops, ” The ION National Technical Meeting, Anaheim, CA, January 2003.

[12] Chiou T-Y., Gebre-Egziabher D., Walter T., Enge P., “Model Analysis on the Performance for an Inertial Aided FLL-Assisted-PLL Carrier-Tracking Loop in the Presence of Ionospheric Scintillation, ” Proceedings of the 2007 National Technical Meeting of the ION, San Diego, CA, January 22-24, 2007.

[13] McGraw G., Braasch M., “GNSS Multipath Mitigation Using Gated and High Resolution CorrelatorConcepts, ” Proceedings of the National Technical Meeting of the ION, San Diego, CA, January 25-27, 1999.

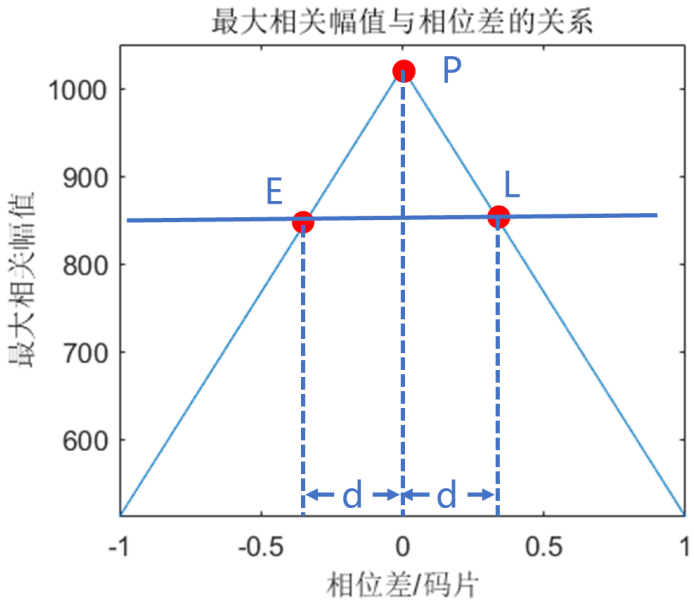

图 16 最大的幅度

而如果 E 和 L 不等，则需要进行调整，向左或向右偏移。然而，由于噪声等因素的存在，实际情况并不是完美的三角形（如下图所示），故无法直接计算出需要偏移的量，只能一点点尝试。

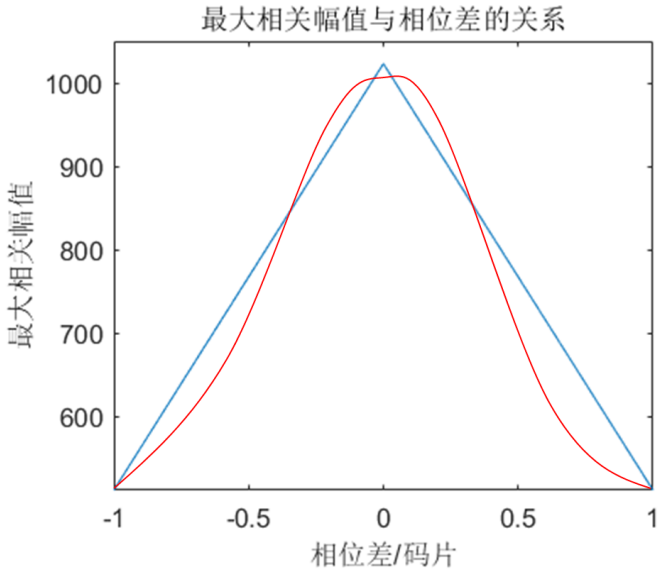

图 17 理论值与真实值比较

具体的流程如下所示：

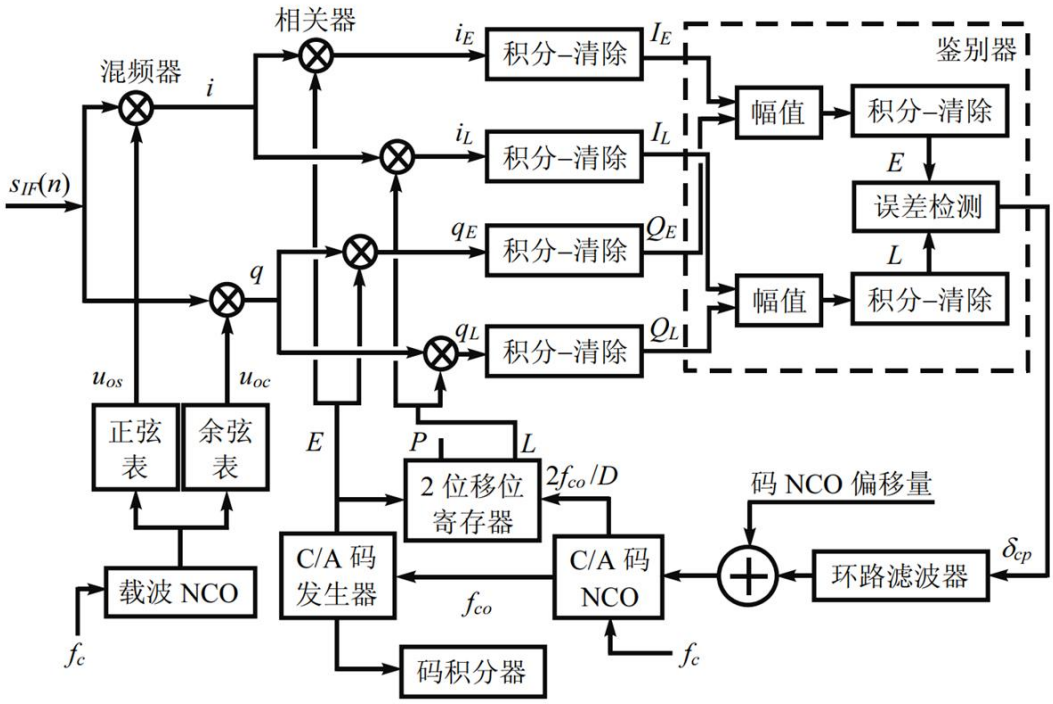

图 18 码环流程图

右下角的环路滤波器接收计算出的偏移量，并生成实际的偏移量。具体的，计算方法为

$$\widehat{\delta_{cp}}(n) = (1-\alpha)\widehat{\delta_{cp}}(n-1) + \alpha\delta_{cp}(n)$$

其中，$\widehat{\delta_{cp}}(n)$ 为实际采用的偏移量，α 为滤波系数，通常取 0.05[14][15]。

经过滤波器后，C/A 码晶体振荡器根据偏移量生成 C/A 码，并通过移位寄存器生成上文提到的 E 和 L。E 和 L 分别与原信号的 I 路和 Q 路相乘，并计算 1 ms 内的相关值，分别得到 I_E, I_L, Q_E, Q_L。然后根据

---

[14] Abolmaesumi P., Sirouspour M.R., “An Interacting Multiple Model Probabilistic Data Association Filter forCavity Boundary Extraction from Ultrasound Images, ” IEEE Transactions on Medical Imaging, Vol. 23, No.6, pp. 772-784, June 2004.

[15] Welch G., Bishop G., “An Introduction to the Kalman Filter, ” University of North Carolina, April 5, 2004.

$$E = \sqrt{{I_E}^2 + {Q_E}^2}$$

$$L = \sqrt{{I_L}^2 + {Q_L}^2}$$

得到幅值。

当然，前面的步骤是相干的。我们也可以使用非相干的办法，这在低信噪比的情况下表现会更好。

再计算出幅值后，进入鉴别器，计算期望的相位调整值 δ_{cp}。具体的计算方法有三种：

- 幅值法

$$\delta_{cp} = \frac{1}{2}\frac{E - L}{E + L}$$

- 功率法

$$\delta_{cp} = \frac{1}{2}\frac{E^2 - L^2}{E^2 + L^2}$$

- 点积功率法

$$\delta_{cp} = \frac{1}{2}\frac{E - L}{P}$$

我们分别作出了这三种方法的图像：

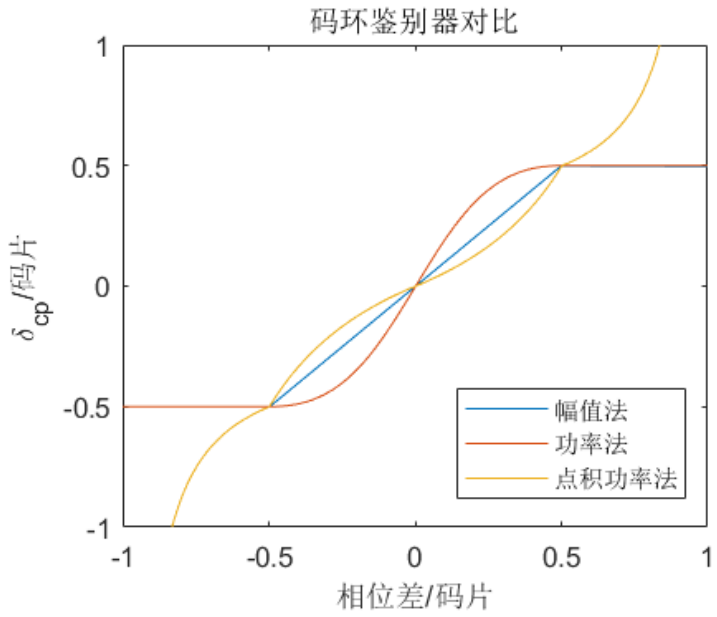

图 19 码环鉴别器对比

可以看到，三条线共有三个交点。如果当前在最左侧的交点左侧，则说明 E、P、L 三点均在等腰三角形的左侧，无法继续计算；最右侧交点的右侧同理。

同时，我们可以看到幅值法是线性的——这是非常好的特性。因此，幅值法也是用得最多的方法。

附录

- 作图代码

```
%% 第 1 个 C/A code
ca1 = C_A_Code(1);
x = [1:1023];
plot(x, ca1)
axis([0, 1024, -0.1, 1.1]);
title('第 1 个 C/A code');
set(gca, 'YTick', [0, 1]);
xlabel('相位');
ylabel('比特');

%% 第 2 个 C/A code
ca2 = C_A_Code(2);
x = [1:1023];
plot(x, ca2)
axis([0, 1024, -0.1, 1.1]);
```

```
title('第 2 个 C/A code');
set(gca, 'YTick', [0, 1]);
xlabel('相位');
ylabel('比特');

%% 第 1 个 C/A code 自相关
ca1 = C_A_Code(1);
ca1 = (ca1 * 2) - 1;
x = [1:2045];
y = xcorr(ca1);
plot(x, y)
axis([0, 2046, -100, 1200]);
xlabel('相位');
ylabel('相关值');
title('第 1 个 C/A code 自相关')

%% 第 1 和第 2 个 C/A code 互相关
ca1 = C_A_Code(1);
ca1 = (ca1 * 2) - 1;
ca2 = C_A_Code(2);
ca2 = (ca2 * 2) - 1;
x = [1:2045];
y = xcorr(ca1, ca2);
plot(x, y)
axis([0, 2046, -100, 1200]);
xlabel('相位');
ylabel('相关值');
title('第 1 和第 2 个 C/A code 互相关')

%% 第 1 个 C/A code 有相移的互相关
ca1 = C_A_Code(1);
ca1 = (ca1 * 2) - 1;
ca2 = [ca1(500:end), ca1(1:499)];
x = [1:2045];
y = xcorr(ca1, ca2);
plot(x, y)
axis([0, 2046, -100, 1200]);
xlabel('相位');
ylabel('自相关值');
title('第 1 个 C/A code 有相移的互相关')

%% 最大相关幅值与相位差的关系
ca1 = C_A_Code(1);
ca1 = (ca1 * 2) - 1;
y = zeros(1, 1023);
for i = [2:1023]
    ca2 = [ca1(i:end), ca1(1:i-1)];
    [y(i), p] = max(xcorr(ca1, ca2));
```

```
end
[y(1), p]= max(xcorr(g1));
y = [y(512:end), y(1:511)];
x = [-1:(1 / 511):1];
plot(x, y)
axis([-1, 1, 512, 1050]);
xlabel('相位差/码片');
ylabel('最大相关幅值');
title('最大相关幅值与相位差的关系')

%% 码环鉴别器对比
ca1 = C_A_Code(1);
ca1 = (ca1 * 2) - 1;
y = zeros(1, 1023);
for i = [2:1023]
    ca2 = [ca1(i:end), ca1(1:i-1)];
    [y(i), p] = max(xcorr(ca1, ca2));
end
[y(1), p]= max(xcorr(ca1));
y = [y(512:end), y(1:511)];

delta1 = zeros(1, 1023);
delta2 = zeros(1, 1023);
delta3 = zeros(1, 1023);
E = 0;
L = 0;
for i = [1:1023]
    if i > 256
        E = y(i - 256) - 512;
    end
    if i < 512 + 256
        L = y(i + 256) - 512;
    end
    P = y(i) - 512;
    delta1(i) = (E - L) / (E + L) / 2;
    delta2(i) = (E * E - L * L) / (E * E + L * L) / 2;
    delta3(i) = (E - L) / P / 4;
end
x = [-1:(1 / 511):1];
plot(x, delta1)
hold on;
plot(x, delta2)
hold on;
plot(x, delta3)
legend('幅值法','功率法','点积功率法');
axis([-1, 1, -1, 1]);
xlabel('相位差/码片');
ylabel('\delta_{cp}/码片');
title('码环鉴别器对比')
```