

默默的点滴

智障儿童欢乐多

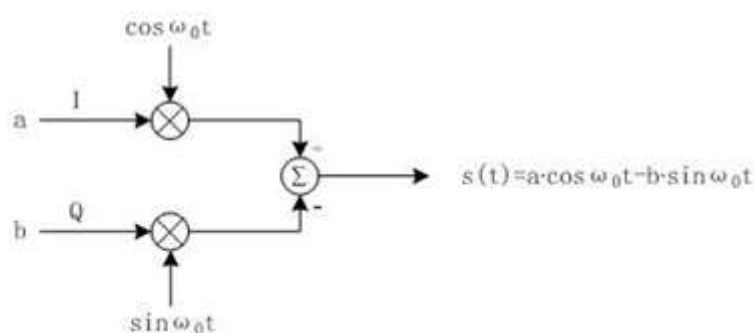
无线通信中的IQ调制，BPSK调制，QPSK调制，16QAM调制的理解

先从IQ调制说起：

IQ调制：

我们先来看看什么是IQ信号。↵

IQ信号与IQ调制有关，IQ调制也叫正交调制，其调制原理如下：↵



I路和Q路分别输入两个数据a、b，I路信号与 $\cos \omega_0 t$ 相乘，Q路信号与 $\sin \omega_0 t$ 相乘，之后再叠加（通常Q路在叠加时会乘以-1），输出信号为： $s(t) = a \cos \omega_0 t - b \sin \omega_0 t$ 。这个过程我们称之为IQ调制，也叫正交调制。↵

输入正交调制器的信号一般被称为IQ信号，经常用复数来表示： $a+jb$ ，对应复平面上的一个点，因此IQ信号通常被大家称为“复信号”。如果再将I路数据相乘的 $\cos \omega_0 t$ 和与Q路数据相乘的 $\sin \omega_0 t$ 表示为： $e^{j\omega_0 t} = \cos \omega_0 t + j \sin \omega_0 t$ ，这样IQ调制就可以利用复数乘法来实现：

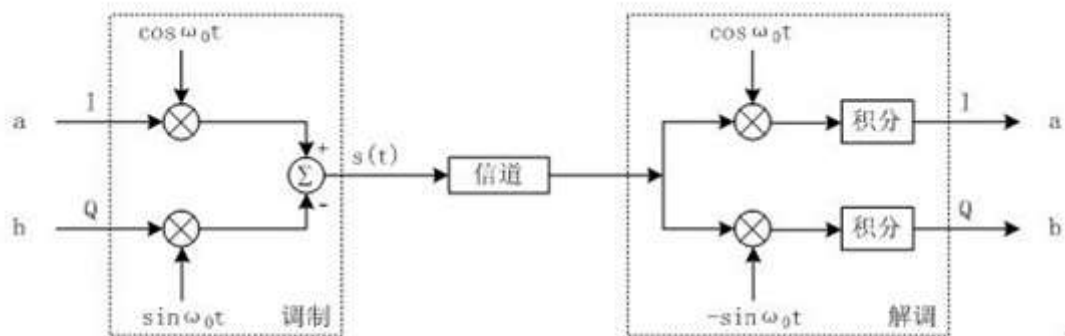


$$\begin{aligned} & \text{Re}\{(a + jb)e^{j\omega_0 t}\} \\ &= \text{Re}\{(a + jb)(\cos \omega_0 t + j \sin \omega_0 t)\} \\ &= \text{Re}\{(a \cos \omega_0 t - b \sin \omega_0 t) + j(b \cos \omega_0 t + a \sin \omega_0 t)\} \\ &= a \cos \omega_0 t - b \sin \omega_0 t \end{aligned}$$

值得注意的是：在IQ调制过程中出现的信号 a 、 b 、 $\cos \omega_0 t$ 、 $\sin \omega_0 t$ 以及最终输出的信号 $s(t) = a \cos \omega_0 t - b \sin \omega_0 t$ 全部都是实信号，只是在实现过程中我们把相关的信号表示为复数而已。

IQ解调原理：

IQ解调原理如下图所示：



接收端收到 $s(t)$ 后，分为两路：

一路乘以 $\cos \omega_0 t$ 再积分，就可以得到 a ：

$$\begin{aligned} & \frac{2}{T} \int_{-T/2}^{T/2} s(t) \cos \omega_0 t dt \\ &= \frac{2}{T} \int_{-T/2}^{T/2} (a \cos \omega_0 t - b \sin \omega_0 t) \cos \omega_0 t dt \\ &= \frac{2}{T} \int_{-T/2}^{T/2} (a \cos^2 \omega_0 t - b \sin \omega_0 t \cos \omega_0 t) dt \\ &= \frac{2}{T} \int_{-T/2}^{T/2} \left[\frac{a}{2} (1 + \cos 2\omega_0 t) - \frac{b}{2} \sin 2\omega_0 t \right] dt \\ &= \frac{2}{T} \cdot \frac{a}{2} \cdot T = a \end{aligned}$$

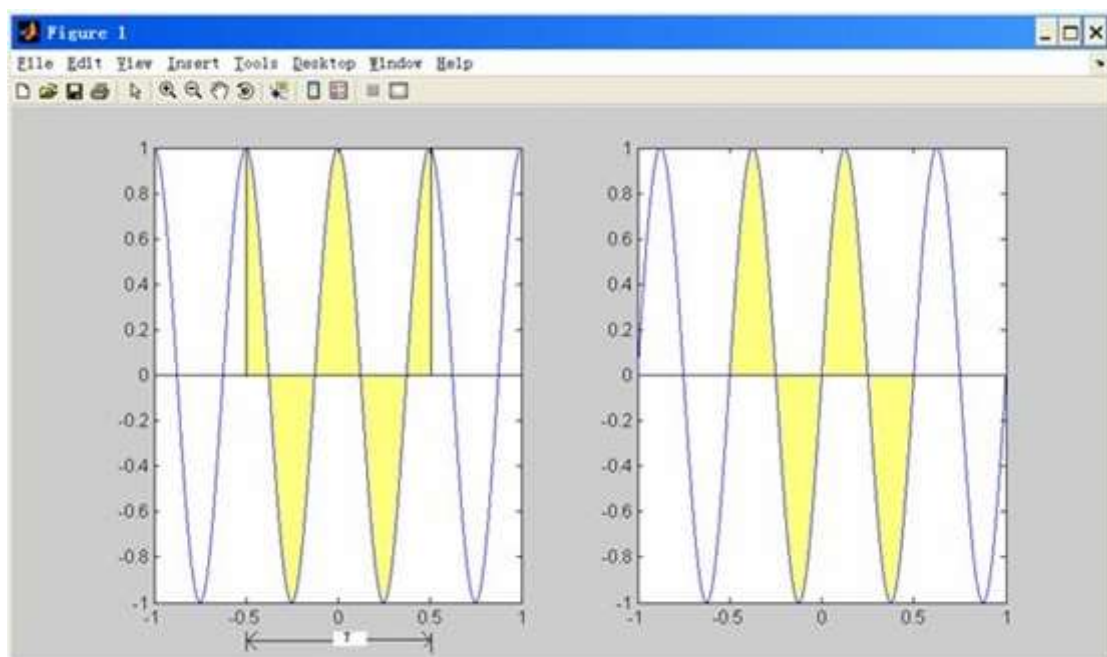
另一路乘以 $-\sin \omega_0 t$ 再积分，就可以得到b：

$$\begin{aligned} & \frac{2}{T} \int_{-T/2}^{T/2} s(t)(-\sin \omega_0 t) dt \\ &= \frac{2}{T} \int_{-T/2}^{T/2} (-a \cos \omega_0 t + b \sin \omega_0 t) \sin \omega_0 t dt \\ &= \frac{2}{T} \int_{-T/2}^{T/2} (-a \sin \omega_0 t \cos \omega_0 t + b \sin^2 \omega_0 t) dt \\ &= \frac{2}{T} \int_{-T/2}^{T/2} \left[\frac{a}{2} (-\sin 2\omega_0 t) + \frac{b}{2} (1 - \cos 2\omega_0 t) \right] dt \\ &= \frac{2}{T} \cdot \frac{b}{2} \cdot T = b \end{aligned}$$

其中T是 $T_0=2\pi/\omega_0$ 的整数倍即可。

注意：上面用到了 $\sin 2\omega_0 t$ 和 $\cos 2\omega_0 t$ 在 $[-T/2, T/2]$ 区间内积分为0。这是很显然的，如下

图所示：黄色部分面积（横轴上方的面积取正，横轴下方的面积取负）的代数和为0。



Linux 下使用 **GNU Octave** 运行下面的代码：

```
1 t=-1:0.001:1;
2 f=1;
3 y=cos(2*pi*f*t);
4 subplot(1,2,1);plot(t,y);
5 y=sin(2*pi*f*t);
6 subplot(1,2,2);plot(t,y);
```

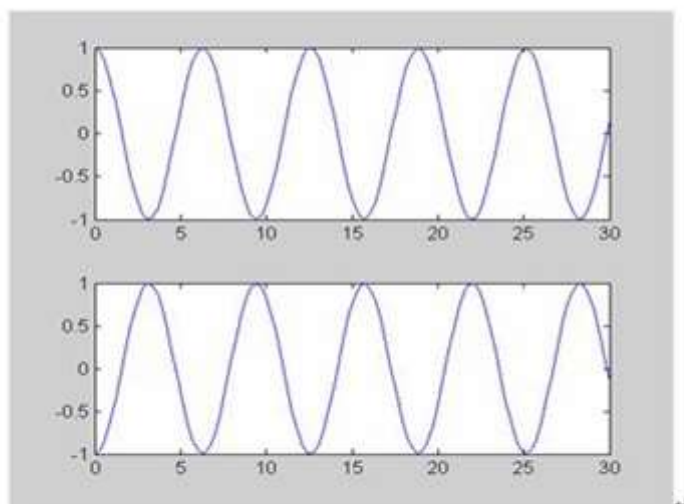
MATLAB

前面我们讲了IQ调制和解调的原理，下面我们看一下如何应用IQ调制来实现MPSK调制（QPSK、8PSK等）、MQAM调制（16QAM、64QAM等）。

先来了解一下BPSK（Binary Phase Shift Keying，二相相移键控）

先来了解一下BPSK (Binary Phase Shift Keying, 二相相移键控)。

两个信号源：一个相位为0: $\cos(\omega t)$ ，一个相位为 π : $\cos(\omega t + \pi)$



根据输入，通过按键控制输出哪个信号(Phase Shift Keying)，例如：

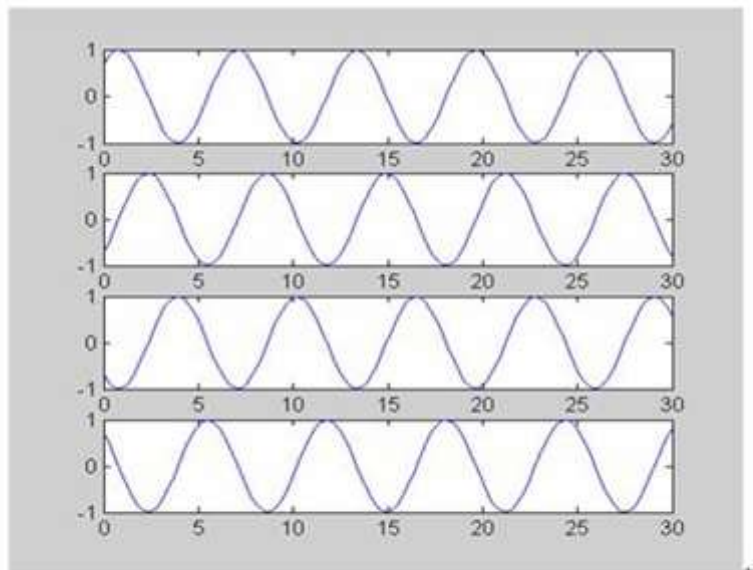
输入是0时输出 $\cos(\omega t)$ ，相位为0。

输入是1时输出 $\cos(\omega t + \pi)$ ，相位为 π 。

这就是BPSK的最初含义。

BPSK：用2个相位分别表示0和1。

QPSK：用4个相位分别表示00、01、10、11。



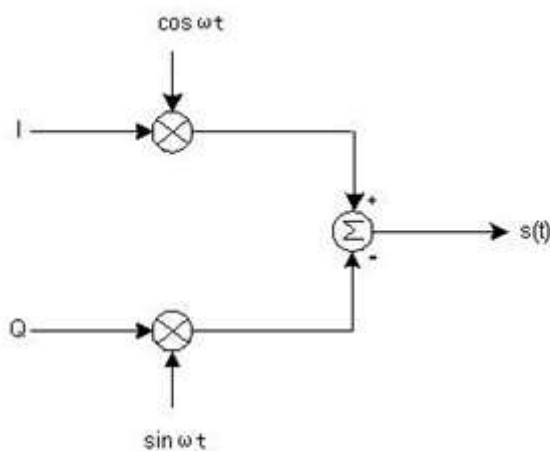
8PSK：用8个相位分别表示000、001、010、011、100、101、110、111。

如何用IQ调制实现QPSK调制？

除了上面的“键控”方法来实现PSK之外，有没有其它方法来实现呢？

有！那就是IQ正交调制，这是目前普遍使用的方法。

我们看一下：在IQ调制器的输入端分别输入 $(+1, +1)$ 、 $(-1, +1)$ 、 $(-1, -1)$ 、 $(+1, -1)$ ，输出将得到什么？



$$s(t) = I \cdot \cos(\omega t) - Q \cdot \sin(\omega t) = A \cos(\omega t + \theta)$$

将 $(+1, +1)$ 、 $(-1, +1)$ 、 $(-1, -1)$ 、 $(+1, -1)$ 作为 (I, Q) 分别代入，经过简单的三角函数运算，可得： $A = \sqrt{2}$ ， θ 如下图所示：

输入信号		输出相位
I	Q	θ
+1	+1	$\pi/4$
-1	+1	$3\pi/4$
-1	-1	$5\pi/4$
+1	-1	$7\pi/4$

推导过程：

$$I=+1, Q=+1, s(t) = \cos \omega t - \sin \omega t = \sqrt{2} \cos(\omega t + \pi/4)$$

$$I=-1, Q=+1, s(t) = -\cos \omega t - \sin \omega t = \sqrt{2} \cos(\omega t + 3\pi/4)$$

$$I=-1, Q=-1, s(t) = -\cos \omega t + \sin \omega t = \sqrt{2} \cos(\omega t + 5\pi/4)$$

$$I=+1, Q=-1, s(t) = \cos \omega t + \sin \omega t = \sqrt{2} \cos(\omega t + 7\pi/4)$$

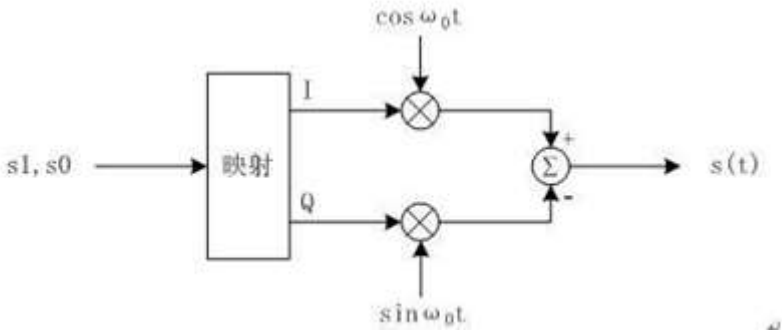
注：其中主要用到了 $\cos(\alpha + \beta) = \cos\alpha\cos\beta - \sin\alpha\sin\beta$

为了使输出信号 $s(t)$ 的幅值 $A=1$ ，调整输入信号的幅度为 $1/\sqrt{2}$ 即可。

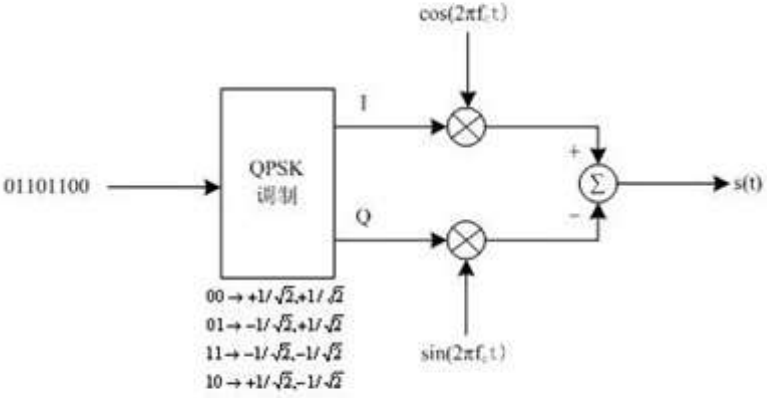
将上述4个相位及对应的IQ信号和输入的00、01、11、10建立映射关系：

输入信号：s1s0	IQ信号	输出信号相位
00	$+1/\sqrt{2}, +1/\sqrt{2}$	$\pi/4$
01	$-1/\sqrt{2}, +1/\sqrt{2}$	$3\pi/4$
11	$-1/\sqrt{2}, -1/\sqrt{2}$	$5\pi/4$
10	$+1/\sqrt{2}, -1/\sqrt{2}$	$7\pi/4$

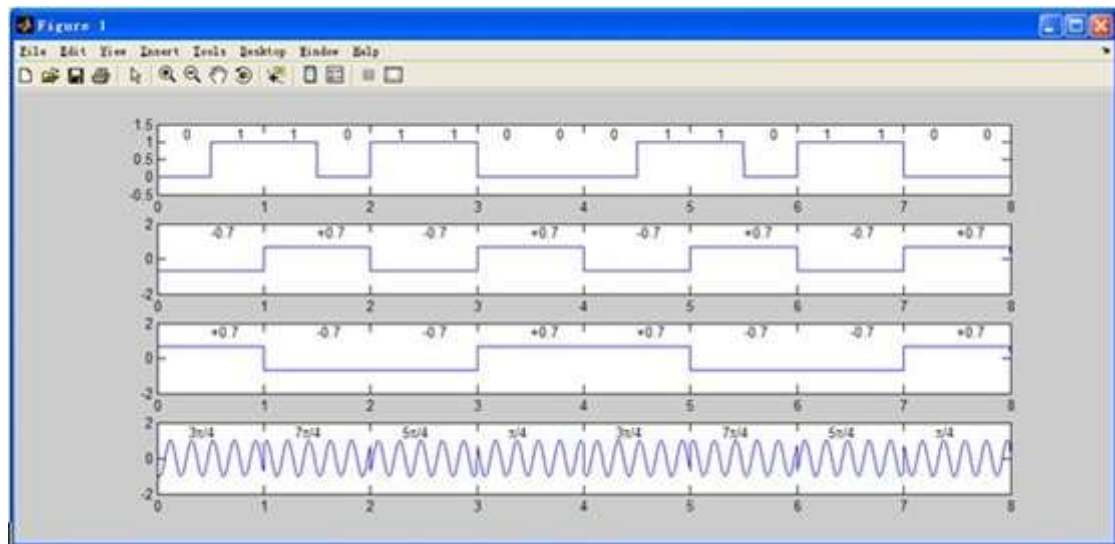
这实际上就实现了QPSK调制：将输入的数据每两个比特划分为一组：s1s0，再根据上面的映射关系转换为对应的IQ信号，最终对应一个输出信号的相位。



假定输入QPSK调制器的数据为：0110110001101100（假定左边的数据先进入调制器），如下图所示：



经QPSK调制后的信号s(t)的时域波形如下图所示：从上至下依次为输入数据信号、I路信号、Q路信号、输出QPSK调制后的信号。



Linux 下使用 **GNU Octave** 运行下面的代码:

	MATLAB
1	%输入信号
2	
3	>> subplot(4,1,1);
4	>> t=0:0.001:8;
5	>> d=[0 0 ;0.5 1;1 1;1.5 0;2 1 ;2.5 1;3 0;3.5 0;4 0;4.5 1 ;5 1 ;5.5 0 ;6 1 ;6.5
6	>> s=pulstran(t-0.25,d,'rectpuls',0.5);plot(t,s) ;
7	>> axis([0 8 -0.5 1.5]);
8	>> text(0.25,1.2,'0') ; text(0.75,1.2,'1') ; text(1.25,1.2,'1') ; text(1.75,1.2,
9	>> text(2.25,1.2,'1') ; text(2.75,1.2,'1') ; text(3.25,1.2,'0') ; text(3.75,1.2,
10	>> text(4.25,1.2,'0') ; text(4.75,1.2,'1') ; text(5.25,1.2,'1') ; text(5.75,1.2,
11	>> text(6.25,1.2,'1') ; text(6.75,1.2,'1') ; text(7.25,1.2,'0') ; text(7.75,1.2,

	MATLAB
1	% I路信号
2	
3	>> subplot(4,1,2);
4	>> t=0:0.001:8;
5	>> a=1/sqrt(2);
6	>> d=[0 -a ;1 +a;2 -a;3 +a; 4 -a ;5 +a;6 -a;7 +a];
7	>> s=pulstran(t-0.5,d,'rectpuls');plot(t,s) ;
8	>> axis([0 8 -2 2]);
9	>> text(0.5,1.5,'-0.7') ; text(1.5,1.5,'+0.7') ;text(2.5,1.5,'-0.7') ;text(3.5,1
10	>> text(4.5,1.5,'-0.7') ; text(5.5,1.5,'+0.7') ;text(6.5,1.5,'-0.7') ;text(7.5,1

	MATLAB
1	% Q路信号
2	
3	>> subplot(4,1,3);
4	>> t=0:0.001:8;
5	>> d=[0 +a;1 -a;2 -a;3 +a; 4 +a;5 -a;6 -a;7 +a];
6	>> s=pulstran(t-0.5,d,'rectpuls');plot(t,s) ;
7	>> axis([0 8 -2 2]);
8	>> text(0.5,1.5,'+0.7') ; text(1.5,1.5,'-0.7') ; text(2.5,1.5,'-0.7') ; text(3.5,
9	>> text(4.5,1.5,'+0.7') ; text(5.5,1.5,'-0.7') ; text(6.5,1.5,'-0.7') ; text(7.5,

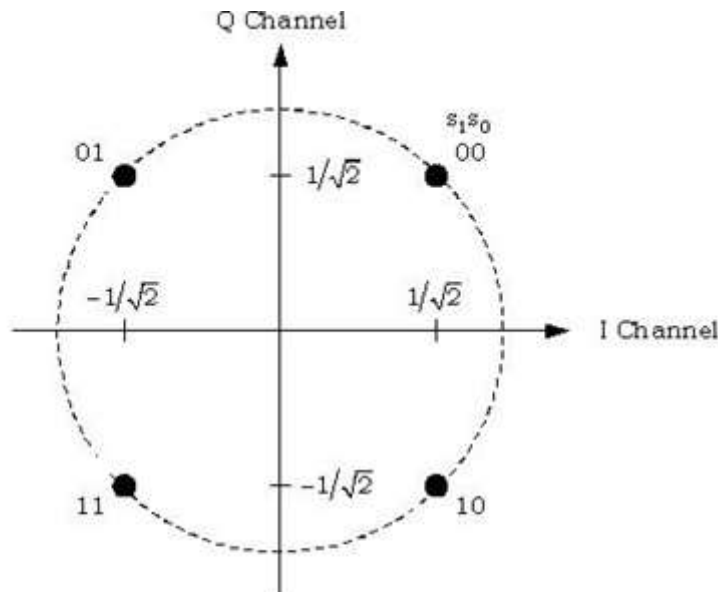
	MATLAB

```

1 %QPSK调制信号
2
3 >> subplot(4,1,4);
4 >> t=0:0.001:8;
5 >> d1=[0 -a ;1 +a;2 -a;3 +a; 4 -a ;5 +a;6 -a;7 +a];
6 >> s1=pulstran(t-0.5,d1,'rectpuls').*cos(2*pi*5*t) ;
7 >> d2=[0 +a;1 -a;2 -a;3 +a; 4 +a;5 -a;6 -a;7 +a];
8 >> s2=pulstran(t-0.5,d2,'rectpuls').*sin(2*pi*5*t);
9 >> plot(t,s1-s2) ;
10 >> axis([0 8 -2 2]);
11 >> text(0.3,1.5,'3\pi/4') ; text(1.3,1.5, '7\pi/4') ; text(2.3,1.5,'5\pi/4') ; t
12 >> text(4.3,1.5, '3\pi/4') ; text(5.3,1.5, '7\pi/4') ; text(6.3,1.5,'5\pi/4') ;

```

QPSK调制的星座图



4个点分别对应4个相位： $\pi/4$ 、 $3\pi/4$ 、 $5\pi/4$ 、 $7\pi/4$ 。

不只是QPSK调制，其它数字调制，包括PSK、QAM调制，都有对应的星座图。

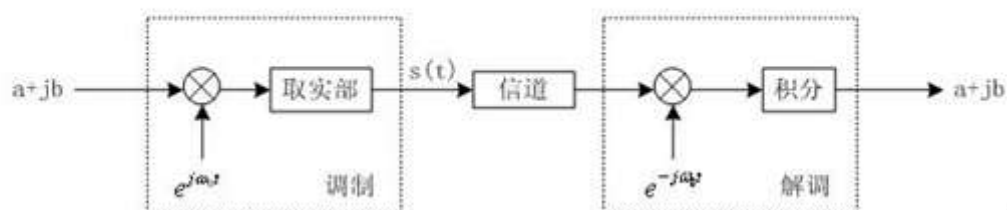
由于星座图完整、清晰地表达了数字调制的映射关系，因此很多书中提到数字调制时经常只是画个星座图完事，不做过多描述。

数字调制也因此而经常被称为“星座调制”。

星座图，就是说一个坐标，如高中的单位圆，横坐标是I，纵坐标是Q，相应于投影到I轴的，叫同相分量，同理投影到Q轴的叫正交分量。由于信号幅度有差别，那么就有可能落在单位圆之内。具体地说，64QAM，符号有64个，等于2的6次方，因此每个符号需要6个二进制来代表才够用。这64个符号就落在单位圆内，根据幅度和相位的不同 落的地方也不同。从其中一个点跳到另一个点，就意味着相位调制和幅度调制同时完成了。”

上面是从映射的角度来引出星座图的，下面我们再换个角度来理解。

这是前面已经讲过的利用复数运算来实现IQ调制解调的原理框图：



发送端：输入 $a+jb$ （I路输入 a ，Q路输入 b ）。

接收端：输出 $a+jb$ （I路输出 a ，Q路输出 b ）。

对于QPSK来讲， a 、 b 只有两种取值 $+1/\sqrt{2}$ 、 $-1/\sqrt{2}$ ， $a+jb$ 有四种取值：

$$\frac{1}{\sqrt{2}} + j\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}} + j\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}} - j\frac{1}{\sqrt{2}}, +\frac{1}{\sqrt{2}} - j\frac{1}{\sqrt{2}}$$

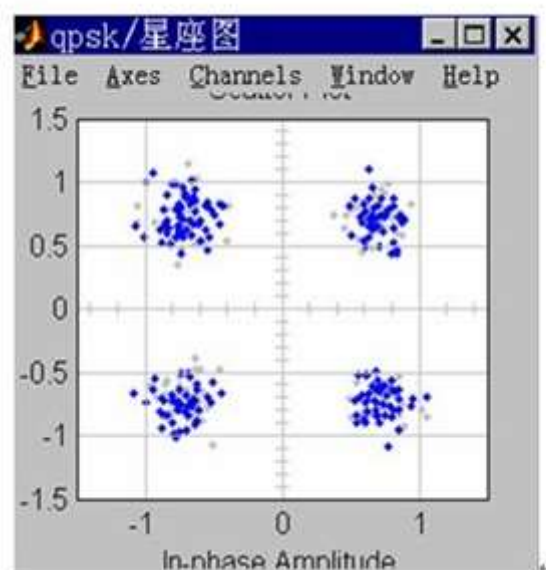
将这4个复数画到I、Q平面上（横轴为实轴，对应I路；纵轴为虚轴，对应Q路）就是QPSK的星座图。

QPSK的映射关系可以随意定吗？

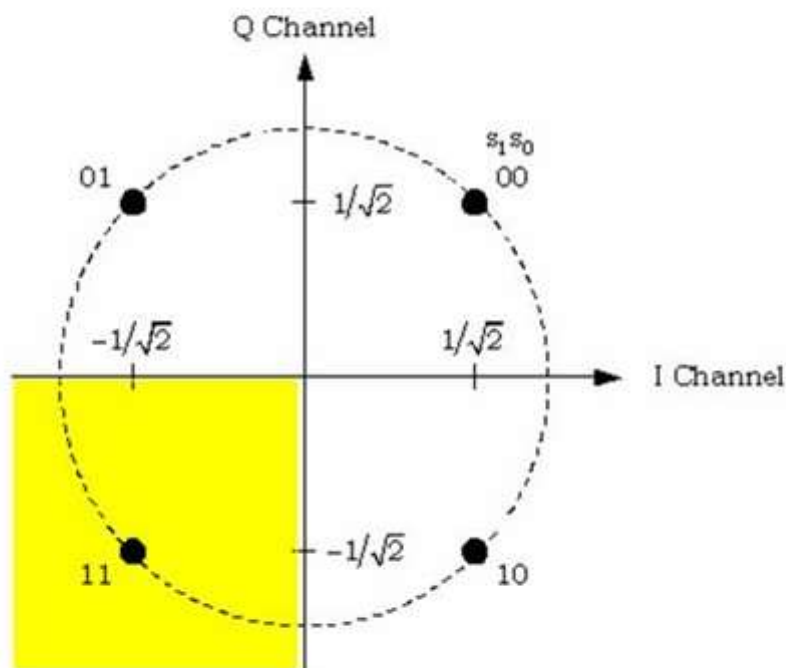
为什么映射时取00、01、11、10顺序与 $\pi/4$ 、 $3\pi/4$ 、 $5\pi/4$ 、 $7\pi/4$ 进行一一映射呢？为什么不取00、01、10、11的顺序呢？

这个需要从接收端的QPSK解调来理解。

由于信道条件不是理想的，当QPSK调制后的数据 $a+jb$ 通过信道到达接收端解调时，得到的数据 $a'+jb'$ 不会正好位于星座图中4个点中某个点正中央的位置，而是分布在4个点周边一定范围内：



接收端如何判决收到的数据 $a'+jb'$ 是星座图中的哪个点呢？最简单的办法就是看距离4个点中的哪个点最近。假定接收到的数据 $a'+jb'$ 位于IQ平面的第三象限，则判决接收到的数据为11，如下图所示：

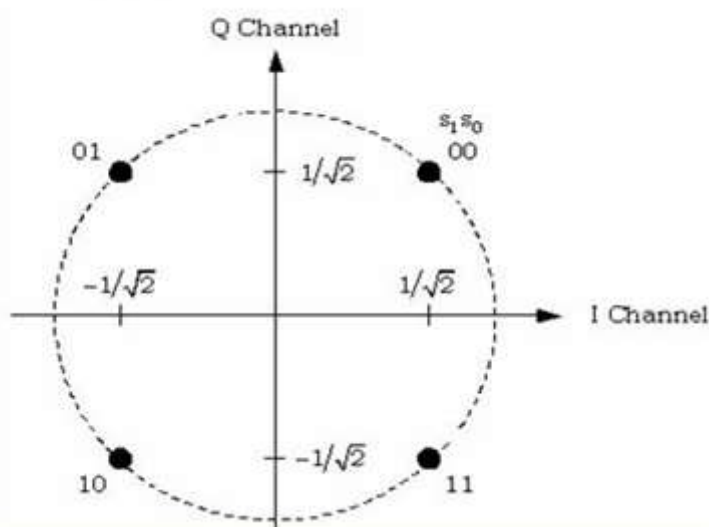


判决有可能出错：从概率的角度讲，误判为相邻点的概率要高于非相邻点。

以发送数据是11为例，接收数据如果没出现在第三象限，则其出现在二、四象限的概率要高于第一象限。

按上述映射关系，接收数据误判为01和10的概率要高于误判为00的概率。11误判为01、11误判为10都只是错了1个比特。

如果我们将映射关系改为：按00、01、10、11顺序与 $\pi/4$ 、 $3\pi/4$ 、 $5\pi/4$ 、 $7\pi/4$ 一一映射，如下图所示：



还以发送数据是11为例，接收数据误判为10和00的概率要高于误判为01的概率。11误判为10错了1个比特，但11误判为00却错了2个比特。

综上所述，在相同的信道条件下，采用 $00 \leftrightarrow \pi/4$ 、 $01 \leftrightarrow 3\pi/4$ 、 $10 \leftrightarrow 5\pi/4$ 、 $11 \leftrightarrow 7\pi/4$ 映射关系的QPSK调制的误比特率要高于采用 $00 \leftrightarrow \pi/4$ 、 $01 \leftrightarrow 3\pi/4$ 、 $11 \leftrightarrow 5\pi/4$ 、 $10 \leftrightarrow 7\pi/4$ 映射关系。

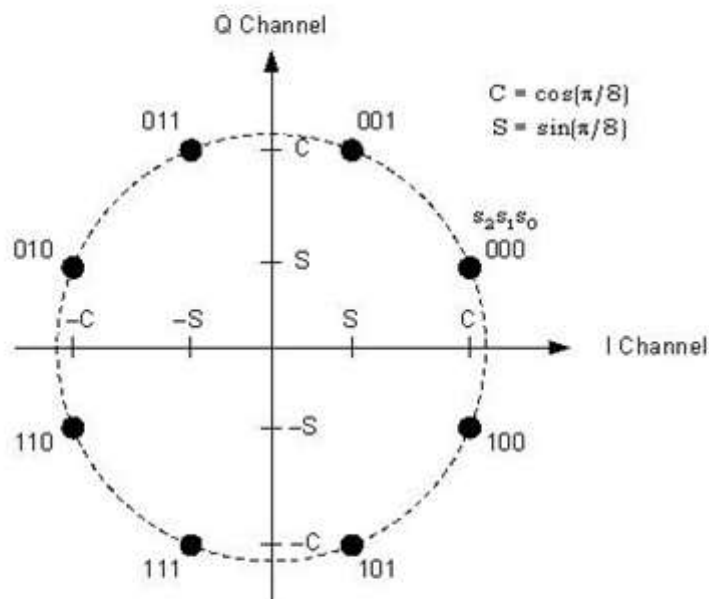
象00、01、11、10这样，相邻的两个码之间只有1位数字不同的编码叫做**格雷码**。QPSK调制中使用的就是格雷码。

十进制数	自然二进制数	格雷码
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

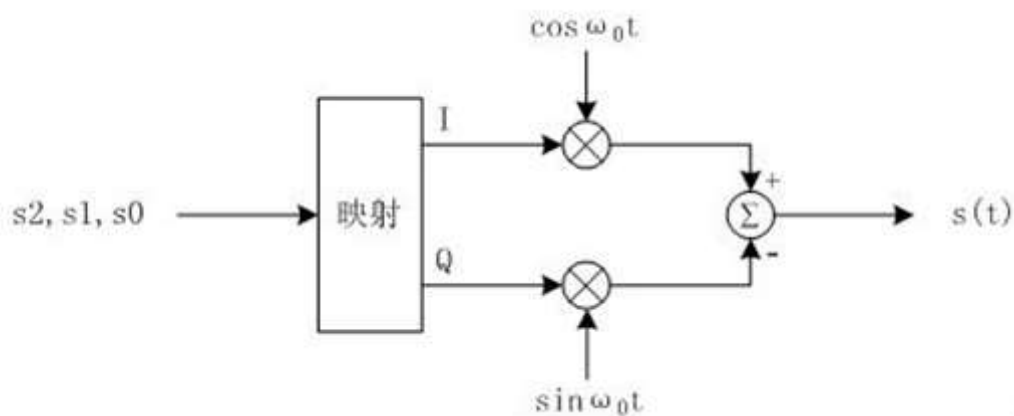
如何使用IQ调制实现8PSK？

上面讲了QPSK，下面再来看看：如何使用IQ调制实现8PSK？

1、先照葫芦画瓢，把星座图画出来：



2、画原理框图：根据上面QPSK的实现，很容易想到：将输入的数据每三个比特划分为一组： $s_2 s_1 s_0$ ，共有8种组合，对应8个输出信号的相位。

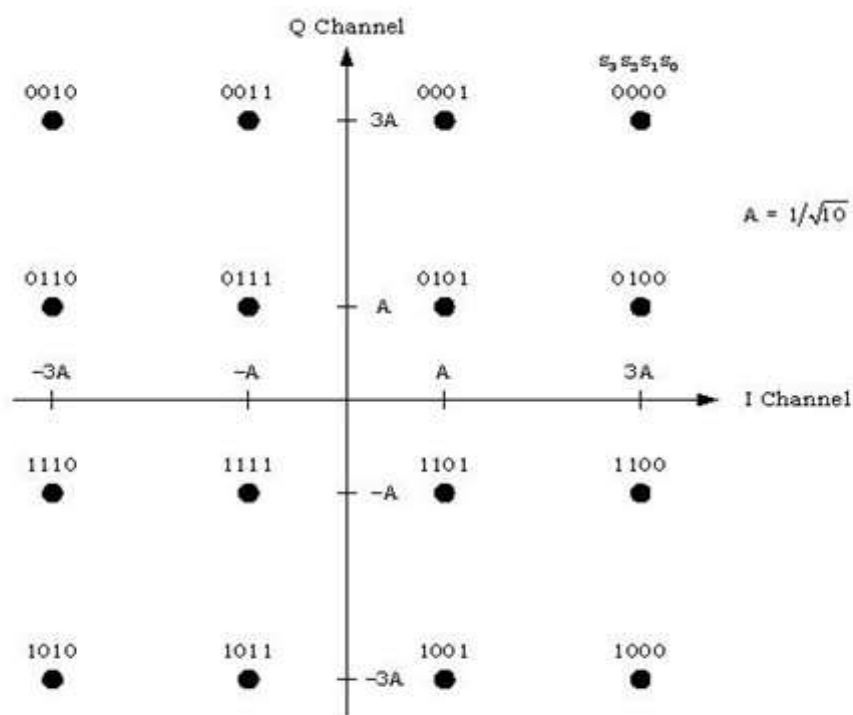


3、确定映射关系：随着输出相位的递增，输入的 $s_2s_1s_0$ 采用格雷码即可。

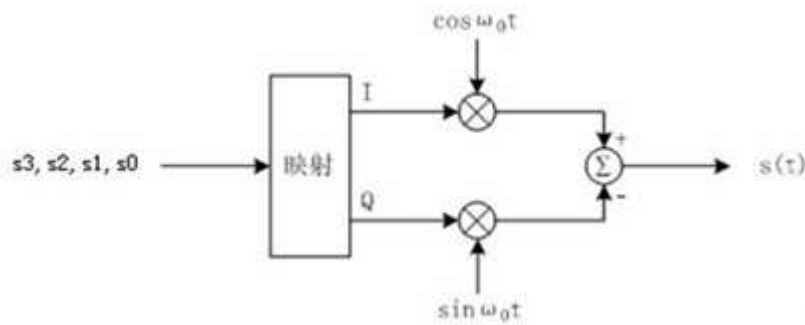
输入信号： $s_2s_1s_0$	IQ信号	输出信号相位
000	$+C, +S$	$\pi/8$
001	$+S, +C$	$3\pi/8$
011	$-S, +C$	$5\pi/8$
010	$-C, +S$	$7\pi/8$
110	$-C, -S$	$9\pi/8$
111	$-S, -C$	$11\pi/8$
101	$+S, -C$	$13\pi/8$
100	$+C, -S$	$15\pi/8$

如何使用IQ调制实现16QAM?

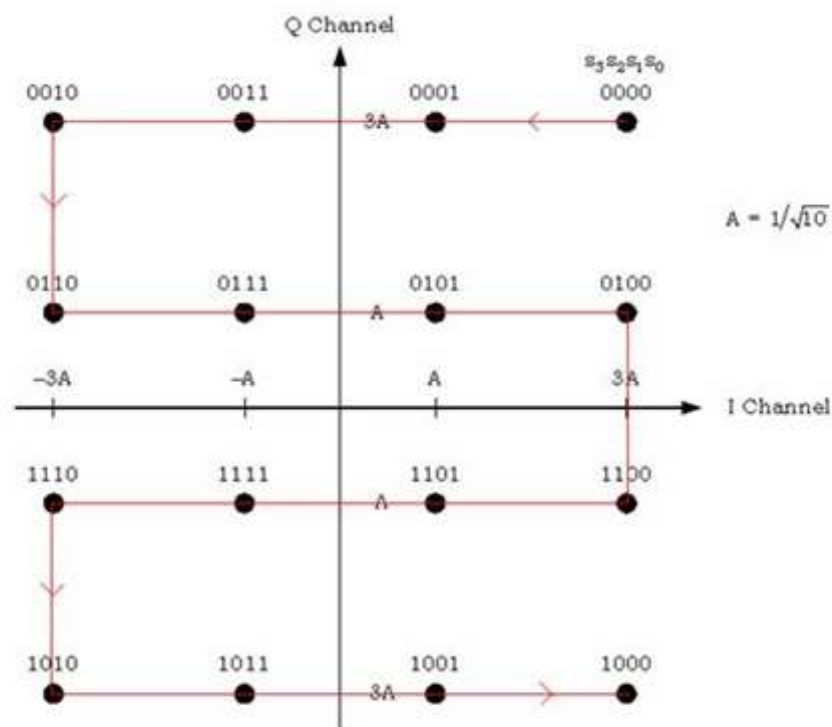
1、星座图：下面是最常见的一种16QAM星座图



2、原理图：实际上不需要再画了，和前面没什么差别，主要是输入的比特被分成了4个一组： $s_3s_2s_1s_0$ 。



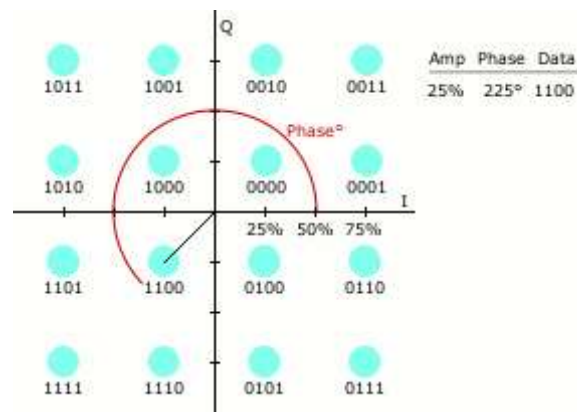
3、映射关系：注意格雷码与各个点之间的映射关系，非常巧妙的是：任何一个点和相邻的左、右、上、下四个点（不一定都有）对应的格雷码都只有1位不同，这样带来的好处是：在出现误码时错1个比特的概率高，而错多个比特的概率相对要低一些（道理与连载38中讲的类似）。



输入的格雷码和IQ信号之间的映射关系表（事实上这张图也没必要画了，一切都在上面的星座图中）：

$s_3s_2s_1s_0$	IQ信号
0000	$+3A, +3A$
0001	$+A, +3A$
0011	$-A, +3A$
0010	$-3A, +3A$
0110	$-3A, +A$
0111	$-A, +A$
0101	$+A, +A$
0100	$+3A, +A$
1100	$+3A, -A$
1101	$+A, -A$
1111	$-A, -A$
1110	$-3A, -A$
1010	$-3A, -3A$
1011	$-A, -3A$
1001	$+A, -3A$
1000	$+3A, -3A$

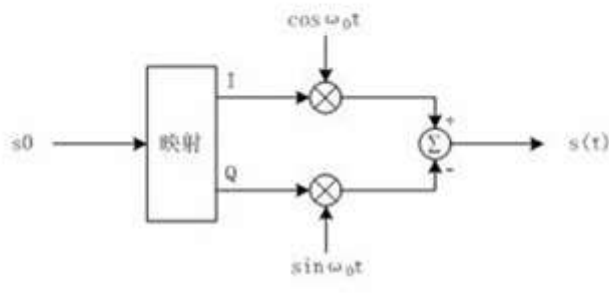
注：前面讲的PSK调制（QPSK、8PSK），星座图中的点都位于单位圆上，模相同（都为1），只有相位不同。而QAM调制星座图中的点不再位于单位圆上，而是分布在复平面的一定范围内，各点如果模相同，则相位必不相同，如果相位相同则模必不相同。星座图中点的分布是有讲究的，不同的分布和映射关系对应的调制方案的误码性能是不一样的，这里不再展开去讲。



利用IQ调制实现BPSK调制

一般的系统中经常会同时用到BPSK、QPSK、QAM调制等多种调制方式，前面讲了QPSK、8PSK和QAM调制可以利用IQ调制来实现，那BPSK能不能用IQ调制来实现呢？

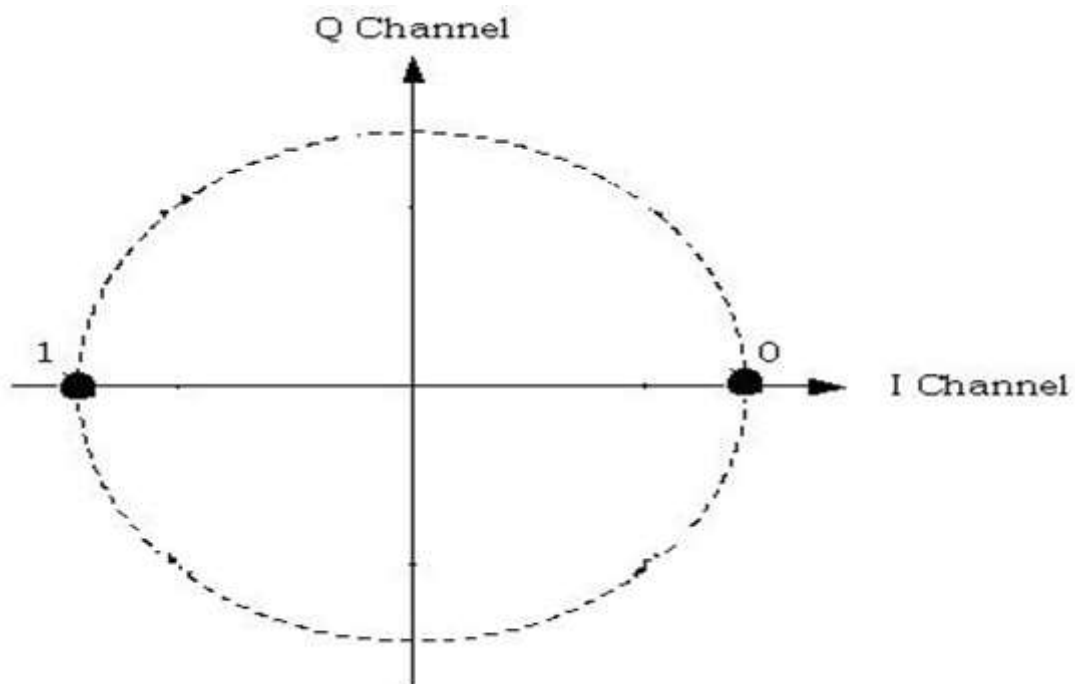
答案是肯定的—只要令Q路数据为0即可。



输入信号和IQ信号、输出信号相位之间的映射关系为：

输入信号: s_0	IQ信号	输出信号相位
0	+1, 0	0
1	-1, 0	π

BPSK的星座图如下：



参考链接

- [通信里 星座图 到底是什么意思啊?](#)
- [正交幅度调制](#)

发布者

默默

默默码农 [查看默默的所有文章](#) →

📅 2018 年 4 月 20 日 👤 默默 📁 LTE、Software defined Radio 26,729 浏览

《无线通信中的IQ调制，BPSK调制，QPSK调制，16QAM调制的理解》上有5条评论

syg

2019 年 7 月 5 日 下午 3:26

写的很好哦～

5GNR

2019 年 8 月 8 日 上午 11:49

受益非浅

钩划

2019 年 8 月 26 日 下午 4:28

我爱这篇文章

是是是

2019 年 9 月 18 日 下午 5:22

太厉害了吧

wjx

2019 年 9 月 22 日 下午 9:32

写的太棒了

浙ICP备12020288号-1