

Error Control Coding

From Theory to Practice

差错控制 编码

(英) Peter Sweeney 著
俞越 张丹 译



差错控制编码

Error Control Coding

From Theory to Practice

Peter Sweeney 著

俞越 张丹 译

清华大学出版社

北京

Error Control Coding From Theory to Practice

Peter Sweeney

Copyright 2002 John Wiley & Sons, Ltd., Baffins Lane, Chichester,
West Sussex PO19 1UD, England

All Rights Reserved. Authorized translation from the English language edition published by Jossey-Bass, Inc. a John Wiley & Sons, Inc. company.

本书中文简体字翻译版由 John Wiley & Sons, Ltd. 授权清华大学出版社独家出版发行。未经许可之出口视为违反著作权法,将受法律之制裁。未经出版者预先书面许可,不得以任何方式复制或抄袭本书的任何部分。

北京市版权局著作权合同登记号 图字: 01-2004-0380 号

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

图书在版编目(CIP)数据

差错控制编码 / 斯维尼(Sweeney,P.)著;俞越,张丹译. —北京:清华大学出版社,2004.6
书名原文: Error Control Coding From Theory to Practice
ISBN 7-302-08459-9

I. 差… II. ①斯…②俞…③张… III. 信道编码—编码理论 IV. TN911.22

中国版本图书馆 CIP 数据核字(2004)第 030718 号

出版者: 清华大学出版社

<http://www.tup.com.cn>

社总机: 010-62770175

地 址: 北京清华大学学研大厦

邮 编: 100084

客户服务: 010-62776969

组稿编辑: 徐培忠

文稿编辑: 孙建春

封面设计: 付建飞

印刷者: 世界知识印刷厂

装订者: 北京鑫海金澳胶印有限公司

发行者: 新华书店总店北京发行所

开 本: 185×260 **印张:** 12.25 **字数:** 305 千字

版 次: 2004 年 6 月第 1 版 2004 年 6 月第 1 次印刷

书 号: ISBN 7-302-08459-9/TP·6079

印 数: 1~4000

定 价: 24.00 元

本书如存在文字不清、漏印以及缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:(010)62770175-3103 或(010)62795704

目 录

第 1 章 数字通信中的编码原理	1
1.1 差错控制方案	1
1.2 数字通信系统的组成	1
1.3 信源编码	2
1.4 差错控制编码	3
1.5 调制	4
1.6 信道	5
1.7 解调	6
1.7.1 相干解调.....	6
1.7.2 差分解调.....	7
1.7.3 软判决解调.....	8
1.8 解码	9
1.8.1 编码和解码示例.....	9
1.8.2 软判决解码	11
1.8.3 其他解码方法	12
1.9 码性能和编码增益.....	12
1.10 信息论对码性能的限制	14
1.11 用于多电平调制的编码	16
1.12 用于突发错误信道的编码	17
1.13 多级编码	18
1.14 错误检测基本方法	18
1.14.1 ARQ 策略.....	19
1.14.2 差错隐藏	20
1.14.3 分组码的检错和纠错能力	20
1.15 编码方案的选择	21
1.15.1 概述	21
1.15.2 数据结构	22
1.15.3 信息类型	22
1.15.4 数据速率	23
1.15.5 实时数据处理	23
1.15.6 功率和带宽的限制	24
1.15.7 信道差错机理	24
1.15.8 成本	24
1.16 结论	25

1.17	练习	25
1.18	参考文献	26
第2章	卷积码	27
2.1	引子	27
2.2	卷积码的主要特性	27
2.3	生成子多项式	28
2.4	术语	28
2.5	编码状态图	29
2.6	卷积码的距离特性	30
2.7	计算距离和重量结构	30
2.8	最大似然解码	31
2.9	维特比算法	32
2.9.1	主要原理	32
2.9.2	维特比解码示例	33
2.9.3	出现的问题	34
2.10	维特比解码的应用实现方法	35
2.11	卷积码的性能	39
2.12	性能较好的卷积码	41
2.13	删除卷积码	42
2.14	卷积码的应用	43
2.15	多级调制码	44
2.15.1	Ungerboeck 编码 8-PSK	45
2.15.2	QAM 星座图	46
2.16	贯序解码	47
2.17	小结	49
2.18	练习	49
2.19	参考文献	50
第3章	线性分组码	52
3.1	引子	52
3.2	二进制码数理	52
3.3	奇偶校验	53
3.4	系统码	54
3.5	线性分组码的最小汉明距	54
3.6	如何编码——生成子矩阵	54
3.7	用奇偶校验矩阵编码	55
3.8	用奇偶校验矩阵解码	57
3.9	利用标准阵列解码	58

3.10	线性分组码的编码解码器设计	59
3.11	对分组码的修正	61
3.11.1	扩展码	61
3.11.2	缩短码	61
3.11.3	通过缩短码长增加最小距离	62
3.12	Dorsch 解码算法	63
3.12.1	Dorsch 解调算法举例	63
3.13	小结	65
3.14	练习	65
3.15	参考文献	66
第 4 章	循环码	68
4.1	引子	68
4.2	循环码的定义	68
4.3	循环码示例	68
4.4	多项式表示	69
4.5	卷积编码	69
4.6	建立循环性	70
4.7	推导循环码的特性	71
4.8	本原多项式	72
4.9	循环码编码的系统编码	73
4.10	循环码的校正子	73
4.11	编码的实现	74
4.12	解码	76
4.13	解码操作	78
4.14	多个误码的纠错	79
4.15	多比特误码纠错示例	79
4.16	缩短循环码	80
4.17	删除循环码	81
4.18	BCH 码	83
4.19	纠正突发错误的循环码	83
4.19.1	法尔码	84
4.20	小结	86
4.21	练习	86
4.22	参考文献	87
第 5 章	有限域算术	88
5.1	引子	88
5.2	有限域的定义	88

5.3	素数大小的有限域 $GF(p)$	89
5.4	二进制域的扩展——有限域 $GF(2^m)$	90
5.5	有限域元素的多项式表示	91
5.6	多项式和有限域元素的性质	92
5.6.1	多项式的根	93
5.6.2	最小多项式	93
5.6.3	元素的阶数	93
5.6.4	有限域元素作为多项式的根	94
5.6.5	不可约多项式的根	94
5.6.6	多项式的分解	94
5.7	有限域上的傅立叶变换	95
5.8	有限域傅立叶变换的另一种理解	96
5.9	根和谱分量	97
5.10	快速傅立叶变换	97
5.11	使用多项式基的硬件乘法器	99
5.12	使用对偶基的硬件乘法器	101
5.12.1	基的转换	102
5.12.2	乘法逆元素	102
5.13	基于正规基的硬件乘法器	102
5.14	有限域算术的软件实现	104
5.15	小结	105
5.16	练习	106
5.17	参考文献	106
第 6 章	BCH 码	107
6.1	引子	107
6.2	通过多项式的根确定循环码	107
6.3	BCH 码的定义	107
6.4	二进制 BCH 码的结构	108
6.4.1	纠正单个错误的码	108
6.4.2	纠正两个错误的码	109
6.4.3	纠正三个错误的码	109
6.5	根和奇偶校验矩阵	109
6.5.1	汉明码	110
6.5.2	二进制 BCH 码	111
6.6	代数解码	112
6.7	BCH 解码和 BCH 限	113
6.8	频域上的解码	114

6.9	二进制 BCH 码的解码示例	115
6.10	关键方程的多项式形式	116
6.11	欧几里得算法	117
6.12	伯利坎普-梅西算法	118
6.13	小结	120
6.14	练习	120
6.15	参考文献	121
第 7 章	里德-索罗门码	122
7.1	引子	122
7.2	里德-索罗门码的生成系数多项式	122
7.3	里德-索罗门码的时域编码	123
7.4	里德-索罗门码的解码	124
7.5	里德-索罗门码解码示例	125
7.6	频域编码的里德-索罗门码	126
7.7	里德-索罗门码的进一步示例	127
7.8	纠删解码	129
7.9	里德-索罗门码纠错纠删解码示例	130
7.10	通用最小距离解码	131
7.11	韦尔奇-伯利坎普算法	132
7.12	单扩展里德-索罗门码	133
7.12.1	单扩展码示例	134
7.13	双扩展里德-索罗门码	135
7.13.1	双扩展码示例	136
7.14	小结	137
7.15	练习	137
7.16	参考文献	138
第 8 章	分组码性能的计算	139
8.1	引子	139
8.2	汉明界	139
8.3	普洛特金界	140
8.4	格瑞兹莫界	140
8.5	辛格里顿界	141
8.6	吉尔伯特-沃尔沙莫夫界	141
8.7	错误检测	142
8.8	分组码的随机错误检测性能	142

8.9	码重分布	143
8.9.1	汉明码	143
8.9.2	里德-索罗门码	143
8.9.3	已知码重分布的对偶码	144
8.10	最坏情况下未检测到的错误率	144
8.11	突发错误的检测	145
8.12	检错码举例	145
8.13	使用分组码的输出错误率	146
8.14	检测到的不可纠正的错误	147
8.15	应用举例——光通信	149
8.16	小结	150
8.17	练习	150
第9章	多级编码	152
9.1	引子	152
9.2	串行级联	152
9.3	采用分组码作为内码的串行级联	153
9.3.1	极长码	153
9.3.2	正交码	153
9.3.3	里德-马勒码	154
9.3.4	高码率码的软判决解码	154
9.4	内码为卷积码的串行级联系统	155
9.5	乘积码(product code)	155
9.6	一般阵列码	158
9.7	多级编码的应用	160
9.8	小结	162
9.9	练习	162
9.10	参考文献	163
第10章	迭代解码	164
10.1	引子	164
10.2	BCJR 算法	164
10.3	BCJR 乘积码示例	165
10.4	利用外信息	166
10.5	递归系统卷积码	167
10.6	RSC 码的 MAP 解码	169
10.7	交织及网格终止	171

10.8	软输出维特比算法·····	172
10.9	GALLAGER 码·····	176
10.10	迭代解码的串行级联·····	181
10.11	性能及复杂度问题·····	182
10.12	移动通信系统中的应用·····	183
10.13	turbo 网格编码调制·····	183
10.14	小结·····	184
10.15	练习·····	185
10.16	参考文献·····	185

第 1 章 数字通信中的编码原理

1.1 差错控制方案

差错控制编码的目的是:在信息从信源向信宿的传输过程中尽最大可能减少误码。它可以看做是信息论的一个分支,起源可追溯到 20 世纪 40 年代后期香农(Shannon)的工作。早期的理论工作指明了后来的发展方向,也为差错控制编码的基本原理提供了一些启示。然而,在探索 and 实现编码过程中出现的问题使人们意识到,应用编码的实际效果和最初预想的效果总会有些许差别。

香农的工作表明,任意信道都可以用无误传输的信息容量来表征。在传输速率小于信道容量的情况下,总可以找到一种方法,使误码率减小到想要达到的任意水平。差错控制可以通过在传输中加入冗余信息来实现。这意味着在传递消息本身所需的最少信息量中加入了更多的符号,于是,接收端上只有某些信息图样符合有效传输。一旦引入足够程度的差错控制,误码率就可以低至扩展码的长度所需的水平,这样噪声的影响就会平均分布到更长的时间段中。

经验表明,使用可行的解码方案找到好的长码,说起来容易,做起来难。所以,与非编码的通信方式相比,编码的实际操作主要集中在那些能够获得的改进上。因此,编码的使用会扩大通信系统的操作范围,减小误码率,降低发射功率要求,或者获得所有这些优点的综合。

除了现有的很多编码外,还有几种普通技术也可用于差错控制,这要根据数据的性质和用户对无误接收的要求来进行选择。最复杂的技术属于前向纠错类型,它假设使用一种能够纠正各种错误的码。其他的方法还有:检测到误码,然后请求重传,即重传差错控制;使用固有的冗余信息来处理错误数据,使得差错在主观上不易被用户察觉,这种方法叫差错隐蔽。

本章首先看一下数字通信系统的组成。第 1.3~1.8 节详细介绍了各个组成部分。第 1.8 节将要给出一个简单的示例码,用来表明差错检测和纠正在理论上是怎样实现的。第 1.9 节将讨论纠错码的性能。第 1.10 节将要介绍现有的理论性能。大量更加深入的主题将在第 1.11~1.14 节中讨论,它们分别是带宽限制条件下的编码、针对突发错误的编码、多级编码(称为级联,concatenation)和能够替代前向纠错的其他方法。最后,第 1.15 节总结了选择编码方式时需要考虑的各种因素。

1.2 数字通信系统的组成

典型的编码通信系统的组成如图 1.1 所示。在信源信息被信源编码器转换成数字格式后要应用差错控制编码技术。图 1.1 中,编码和调制按照惯例是分开,不过我们稍后

便会发现,有些情况下,这两部分必须放在一起设计。在接收端,操作顺序与发送端的操作相反。

各部分的功能将在以后的几节里详细介绍。

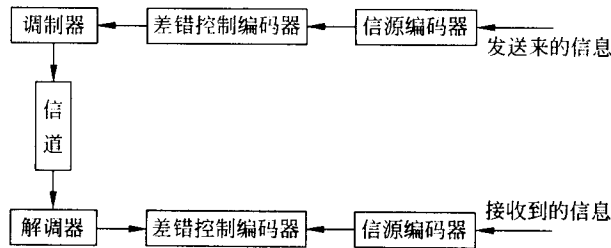


图 1.1 编码通信系统

1.3 信源编码

信息采用数字形式表达,去除数据内部固有冗余的技术也可能需要同时运用。任意消息中包含的信息量都由这条消息被选择进行传送的概率 p 来定义。信息含量 H 的单位为比特,可用下式计算:

$$H = \log_2(1/p)$$

例如,如果一条消息有 1% 的机会被选择传送,那么它所含的信息量近似为 6.64 比特。

如果一共有 M 条消息可供选择,而消息 m 被选择的概率设为 p_m ,那么一条消息中所传送的平均信息量为:

$$\bar{H} = \sum_{m=0}^{M-1} p_m \log_2 \left(\frac{1}{p_m} \right)$$

对于 p_m , 有 $\sum_{m=0}^{M-1} p_m = 1$ 。

如果各条消息是等概率的,即: $p_m = 1/M$,那么所传送的平均信息量正好是 $\log_2(M)$ 。这正是每条消息以定长编码方案表示时所需的比特数。以 256 条消息为例,8 比特的码可以用来表示其中任意一条消息,如果它们以等概率传输,那么任意消息的信息含量也是 8 比特。

如果消息不以等概率进行传输,那么它的平均信息量就小于 $\log_2(M)$ 比特。这时人们便希望寻找一种使用更少比特数的数字表示法,且越接近平均信息量越好。这可以通过使用可变字长码(譬如霍夫曼(Huffman)码或算术码,其中,被传送的序列的长度可以尽可能地接近消息的信息含量)来实现。或者,在一些主观应用中,例如语音、图像、视频,有损的压缩技术既可以用来产生固定字长格式,又可以用来产生可变字长格式。这样做的目的是,允许接收端将发送来的信息重建成某种与信源信息不完全匹配的形式,但是这些差别在主观上不容易被用户察觉。

1.4 差错控制编码

差错控制编码技术原则上是各种数字信号处理技术的综合,目的是使信道噪声影响平均分配到多个传输信号中。单个传输符号所承载的噪声量远比较长时间间隔中经历的噪声量更难预测,所以,编码符号的噪声容限与非编码符号所需要的噪声容限相比要小一些。

把冗余信息加入传输序列中是差错控制编码的重要组成部分。因此,纠错码传输的比特数肯定大于表示信息所需的比特数。如果不满足这一点,码甚至不让我们检测到差错的存在,就更谈不上差错控制了。也就是说,从理论上讲,信源编码器所完成的任何不完全压缩都可以被认为是具有差错控制能力。然而,事实上,最好先将信源信息尽可能完全地压缩,然后再由纠错解码器以一种可以用来获得最佳效果的方式重新引入冗余信息。

编码器的结构如图 1.2 所示。信息总是以帧的形式输入编码器,每一帧由固定数目的符号组成。在大多数情况下,编码器输入端的符号都是单个比特,只有在极少数情况下,编码器要求输入多个比特。这里的“符号”一词并没有特殊含义。

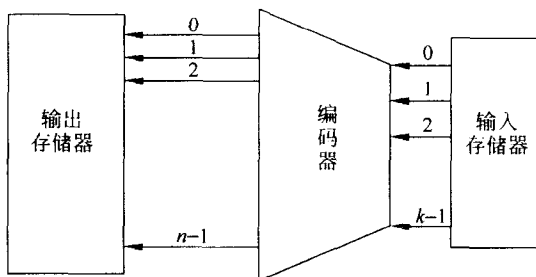


图 1.2 编码器

为了产生输出,编码器要使用当前输入帧中的符号,有时也加入大量此前帧中的符号。一般来讲,输出中包含的符号量比输入多,即加入了冗余信息。通常用码率(R)来描述一帧中码的输入符号与输出符号之比。低码率意味着高冗余,相对于高码率来说有可能提供更有效的差错控制,但要以信息吞吐量的减小为代价。

如果编码器只使用当前帧产生输出,那么这种码叫做 (n, k) 分组码,每帧的输入符号数设为 k ,对应的输出符号数是 n 。如果编码器记下了当前帧前面大量的帧,并在其编码算法中利用了这些帧,那么这种码叫做树码(tree code),通常是我们所熟知的卷积码的子集。这种情况下,输入帧中的符号数设为 k_0 ,输出帧中的符号数设为 n_0 。编码器在数据上有效扮演着滑动窗口的角色,每次只向前滑动一个小的距离,这样保证了编码器窗口中总有许多符号相对上次保持不变,如图 1.3 所示。窗口的总长度称为输入约束长度(input constraint length, 即 K),它等于当前输入帧的 k_0 个符号加上所存储的的符号数。后面的这个参数叫做存储约束长度(memory constrain length, 即 v)。

在更复杂的系统中,编码可能不止一级,并可能加入分组码和卷积码,还有可能加入一种称为交织编码的技术。这样的系统将在以后的小节中讨论。

本书所讨论的全部码都具有一个共同特性,即线性(linearity)。通常,当我们提起线性

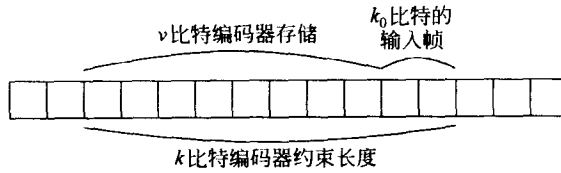


图 1.3 树编码器的滑动窗口

系统时,自然会想到输出根据输入变化而成比例变化(齐次性)。在线性系统中,输入端两个不同信号之和的输出可以看做两个信号单独作为输入时的输出之和(叠加性)。更严格地说,如果系统对输入采用函数 f 来产生输出,那么有:

$$f(cx) = c \times f(x) \text{ (齐次性)}$$

$$f(x+y) = f(x) + f(y) \text{ (叠加性)}$$

这里 c 是一个标量, x 和 y 是矢量。

线性码的定义没有上述定义那样严格,因为它不必考虑从输入到输出的映射关系,而只是考虑编码器的可能输出。然而,在实际中,线性系统将用来产生码,所以上面的定义可以应用到所有实际情况中。

线性码的标准定义如下:

- 用一个有效标量去乘以一个码序列,得到的还是一个码序列。
- 两个码序列的和仍是一个码序列。

乘法和加法运算的一般规则将在第 5 章介绍,但是对于二进制码而言,因为有效标量只有 0 和 1,所以,用 0 乘某个值,得到的结果总是 0,用 1 乘某个值,得到的结果就是值本身。二进制码的加法是用模 2 运算实现的,即对两个值执行异或操作。

关于线性码的一个简单例子将在 1.8 节中给出。尽管线性码的定义不像线性系统的定义那样严格,但在实际中,线性码总是通过线性系统来产生的。线性码中必须包括全 0 序列,因为用 0 乘以任何码序列都将得到全 0 序列。

1.5 调制

调制器可以被想成一种数模转换器,为将数字码流转换到真正的模拟世界做准备。首先,数字流将以基带(baseband)形式表示,即,信号变化的速率应该和数字符号被表示的速率相符。常用的表示方法是非归零(Non Return to Zero, NRZ)码,它用信号的 $+V$ 或 $-V$ 电平来表示比特,具体用 $+V$ 还是 $-V$ 则取决于它所要代表的比特值。如图 1.4 所示。

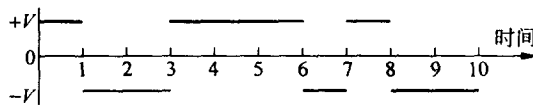


图 1.4 二进制 NRZ 码流

虽然可以传输这种信号,但通常还是将它转换到更高频的范围中。这样做的原因有:可

以利用不同的频段进行不同的传输,而且事实上,频率越高,波长越短,所需要的天线也越短。本书在大多数情况下,都假设调制是用 NRZ 基带信号乘以一个高频的正弦载波实现的,载波的频率应该是所传送的比特率的若干倍(这样,在单独 1 比特的间隔内,载波的周期数恰好是整数)。因此在 1 比特间隔内传送的信号要么是正弦载波,要么是它的反相。这种调制方式叫做二进制相移键控(Binary Phase Shift Keying, BPSK)。

另外,可以在相对原始载波呈 90° 的位置上使用第 2 个载波,对它进行调制并将结果信号和第 1 个调制后的信号相加。换句话说,如果第 1 路 BPSK 的信号是 $\pm \cos(2\pi f_c t)$, 其中 f_c 是载波频率, t 表示时间,第 2 路信号是 $\pm \sin(2\pi f_c t)$, 那么它们的和就是:

$$s(t) = \sqrt{2} \cos(2\pi f_c t + i\pi/4) \quad i = -3, -1, +1, +3$$

这种调制叫做正交相移键控(Quadrature Phase Shift Keying, QPSK), 它优于 BPSK 的地方在于,在相同的时间内,相同的带宽下,在抗噪声性能没有损失的情况下,它传送的比特数是 BPSK 的 2 倍。调制后的信号所占据的实际带宽取决于具体的实现方法,但是一般来说,在传输速率 1 bps 的情况下,使用 BPSK 传输所占的带宽是 1 Hz, 而使用 QPSK 只占 0.5 Hz。

QPSK 的相位图如图 1.5 所示。这种从比特到相位的映射基于以下假设:每路载波用来自编码数据流的奇数或偶数位独立地得到调制。从图中可以看出,相邻的点之间只有 1 位是不同的,这是因为只有一路载波的相位发生了变化。保证相邻的点中只有一位不同的映射形式称做格雷编码(Gray Coding)。

其他调制方式包括移频键控(Frequency Shift Keying, FSK), 其中,被传送的信号的频率是由数据决定的。FSK 的优点是易于实现,但是它的抗噪声性能不如 BPSK 和 QPSK。还有许多属于连续相位调制(Continuous Phase Modulation)类型的调制方法,这些方法可以将传输波形之间相的不连续性降到最低,并以此来改善非线性功率器件产生的频谱特性。

在带宽限制的情况下,和 BPSK 或 QPSK 相比,在占用相同的带宽时,使用多级调制方式可以获得更高的比特率。在多进制相移键控(M-ary Phase Shift Keying, MPSK)中,可供传输的相位更多了。在正交调幅(Quadrature Amplitude Modulation, QAM)情况下,相位呈正交的每对载波在叠加之前各自被赋予了多种可能的振幅,以产生相位和幅度都不同的传输信号。QAM 的抗噪声能力要比相同进制的 MPSK 好,但是当系统含有非线性功率器件时,振幅的变化可能会带来一些问题。QAM 和 MPSK 均需要采用将码和调制联合起来考虑的特殊编码方法。

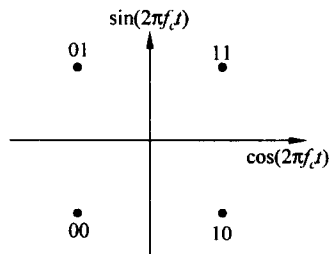


图 1.5 格雷编码 QPSK 相位图

1.6 信道

传输媒介会对信号引入一系列的影响,比如衰减、畸变、干扰和噪声等,使得信息能否被正确接收成为不确定的问题。尽管人们最容易想到错误是由信道引入的,但应当看到,实际上,是信道对解调器的影响引入了错误。

破坏传输符号的途径可以用以下一些术语来描述。

- 无记忆信道——符号与符号之间的差错概率是相互独立的。
- 对称信道——传送符号的值为 i 而接收到符号的值为 j , 这种错误发生的概率和传送符号的值为 j 而接收到符号的值为 i 的概率是相等的, i 和 j 可以取任意值。常见的例子是二进制对称信道 (Binary Symmetric Channel, BSC), 每比特发生错误的概率是 p , 如图 1.6 所示。
- 加性高斯白噪声信道 (Additive White Gaussian Noise, AWGN)——无记忆信道, 信号通过此信道会附加宽带噪声, 噪声的振幅是正态 (高斯) 分布的随机变量。
- 突发信道——这种错误的特征是: 相对较高的误符号率发生的期间与相对较低或者为 0 的误符号率发生的期间交替出现。
- 组合 (扩散) 信道——错误由突发错误和随机错误的组合构成。在实际中所有的信道都呈现出某种形式的组合行为。

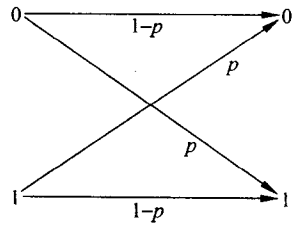


图 1.6 二进制对称信道

如果错误是随机分布的, 那么许多码都可以处于最佳工作状态, 所以发送端和接收端可能包括一些其他的元素, 如调制器前加入交织器, 解调器后加入解交织器, 它们可以使信道错误的影响随机分布。这些将在 1.12 节讨论。

1.7 解调

1.7.1 相干解调

解调器决定被传送符号的值, 并将那些判决结果传给下一级。通常这是用传送符号的副本作某种相关运算来实现的。以 BPSK 为例, 与单独一个固定相位的载波作相关运算后, 从检测器产生的输出结果要么为正, 要么为负。如果没有噪声, 那么检测到的信号电平应该是 $\pm\sqrt{E_r}$, 这里 E_r 是接收到的每一比特信号中的能量。AWGN 信道的影响会附加给一个电平为 n 的噪声, n 服从均值为 0、标准差为 σ 的高斯分布。概率密度由下式给出:

$$p(n)dn = \frac{1}{\sigma\sqrt{2\pi}}e^{-n^2/2\sigma^2}$$

高斯噪声具有平坦的能谱, 它的噪声电平通常用单边噪声能谱密度 (Single-sided Noise Power Spectral Density, 记做 N_0) 来描述。单位比特间隔内的高斯噪声的方差 σ^2 是 $N_0/2$ 。事实上, 我们可以认为有一个总方差为 N_0 的噪声, 一半加到副本的同相或反相上, 另一半用于相位积分, 所以不会影响到检测器。因此, 格雷码编码的 QPSK 的性能和 BPSK 完全一样, 因为两路载波可以各自作为独立的 BPSK 信号得到解调, 分别受到具有相同标准差的独立的高斯噪声的影响。

解调器通过检测到的信号的极性来做出判决。如果收到的电平是正极性的, 解调器便

假定对应于副本的值被传送。如果收到的电平为负,解码器则认为另一个值被传送。因此,如果无噪声时的输出电平为 $-\sqrt{E_b}$,再附加一个大于 $+\sqrt{E_b}$ 的噪声值,或是无噪声时的输出电平为 $+\sqrt{E_b}$,再附加一个小于 $-\sqrt{E_b}$ 的噪声值,解码器将会发生错误。只考虑前者的情形,我们可以发现,发生错误的概率恰好就是高斯噪声具有大于 $+\sqrt{E_b}$ 的值的概率:

$$p = \frac{1}{\sqrt{\pi N_0}} \int_{\sqrt{E_b}}^{\infty} e^{-n^2/N_0} dn$$

变量替换 $t = n/\sqrt{N_0}$ 则有:

$$p = \frac{1}{\sqrt{\pi}} \int_{\sqrt{\frac{E_b}{N_0}}}^{\infty} e^{-t^2/N_0} dt$$

$$p = \frac{1}{2} \operatorname{erfc} \sqrt{\frac{E_b}{N_0}} \quad \text{其中,} \quad \operatorname{erfc} = \frac{2}{\sqrt{\pi}} \int_t^{\infty} e^{-t^2} dt \quad (1.1)$$

函数 $\operatorname{erfc}(x)$ 叫做互补误差函数,它的值一般用查表的方式可以得到。注意这里 $\operatorname{erfc}(x)$ 的最大值为 1.0,这样最大的误比特概率为 0.5,这是很有意义的,因为即使完全不去接收信号,我们也有 50% 的概率猜中比特值。

1.7.2 差分解调

我们用 BPSK 和 QPSK 进行传输的时候,通常会遇到更为复杂的问题。如果没有其他的信息,接收端不可能判断出绝对相位,所以也就不能判断出两个相位中哪个代表 0、哪个代表 1。这是由于传输路径中的延时(等价于相位的移动)是未知的。因此,比特值通常基于相位之间的差值;根据精确解调方法,这也叫做差分编码相移键控(Differentially Encoded Phase Shift Keying, DEPSK)或差分相移键控(Differential Phase Shift Keying, DPSK)。从调制的角度来看,这两种说法是完全一致的,即比特值 0 通常导致相位变化,而比特值 1 则使相位保持不变。接收端虽然可能不知道相位的绝对值,但是它应该能够知道相位究竟是变化了还是保持不变。两种解调器实现方法的差别总结如下。

- DEPSK——和普通的 PSK 一样,解调器保留两个载波相位中某一个的副本,并用该副本和接收到的信号作相关运算。然后将相关运算结果的符号和前一相关值进行比较;符号变化表明是数据比特 0,符号不变则表示是数据比特 1。和普通的 PSK 相比,如果此时相位被错误接收而前一刻相位被正确接收,或此时相位被正确接收而前一刻被错误接收,就会出现误比特。因此,在 BPSK 解调器中造成 1 比特差错的噪声在 DEPSK 解调器中将会导致两个连续的误比特,所以 DEPSK 的误比特概率近似是 BPSK 的两倍。
- DPSK——解调器用前一刻接收到的相位作为下一比特的副本。正相关表示数据值 1,负相关表示数据值 0。误比特同样倾向于成对相关,不过整体性能会变差。事实上,DPSK 的误比特概率遵循不同的曲线形状: