

在数字通信系统中,数据传输往往以帧为单位进行。接收端为了正确恢复数据信息,必须确定每帧的起止点,因此,帧同步是信号同步技术中非常关键的一环。本章主要对基于同步码插入的帧同步技术和编码辅助帧同步技术进行了研究,并给出了帧同步算法的性能评价指标。此外,本章末尾基于 DVB-S2 通信帧结构对差分检测帧同步技术进行了讨论。

5.1 帧同步技术概述

数据传输帧是通信系统进行数据发送和接收的基本单位。一个传输帧应包括帧头、帧尾、控制信息和有效的数据块。但是无论是无线通信系统还是有线通信系统一般都不会单独传送帧定位信息到接收端。常规的帧定位信息都是在发送数据帧中插入一段固定的码字,在接收端通过识别这个码字来重新定位数据帧的帧头位置,这一过程就是帧同步过程。

帧同步的实现是地面接收系统中不可或缺的关键功能之一,主要将从卫星接收到的高速数码流格式化,辨识每一帧的起始位置,然后将数据流经高速接口进入后继处理器进行处理。帧同步系统的性能会直接影响整个通信系统的性能,可以说,在同步通信系统中,“同步”是进行正确信息传输的前提。正因为如此为了保证信息的可靠传输,要求同步系统应有较高的可靠性。

帧同步按不同标准可划分为不同类型。按是否插入帧同步字段(Synchronization Word, SW),可分为 SW 辅助帧同步和盲帧同步两类。按帧长度是否可变,帧同步可分为固定帧长帧同步和可变帧长帧同步;按二进制数据在时间轴上的传输是否连续,可分为突发帧同步和连续帧同步;按是否有信道编码模块辅助定位,可分为无编码辅助帧同步和编码辅助帧同步;按是否在数据帧中插入已知帧同步字段,可分为同步码辅助帧同步和盲帧同步。不同类型的帧同步各有特点,使用的同步算法也有很大差异,需根据实际需求做出选择。

数字通信系统对帧同步通常有以下 3 点要求:

- (1) 帧同步的检测概率要大,漏警概率要小,虚警概率要低;
- (2) 同步建立时间要短,同步保持时间要长;
- (3) 在满足同步性能要求的情况下,帧同步码长尽可能短,数据传输效率高。

5.2 信号模型

考虑有帧同步字段插入的 MPSK 调制通信系统,其原始帧结构由两部分组成:长度为 L 的已知帧同步字段 SW 和长度为 N 的数据流。

假设接收机在启动帧同步模块前已实现理想载波同步和符号同步,帧同步模块的输入信号可表示为

$$r(k) = s(k) + w(k), \quad k = 0, 1, 2, \dots$$

其中, $s(k)$ 表示接收信号数据帧; $w(k)$ 是均值为 0、方差为 σ^2 的高斯白噪声。帧同步模块的主要任务就是根据接收信号 $r(k)$, 准确定位帧的起始、终止时刻, 即图 5-1 中的 k_0 与 $k_0 + L + N - 1$, 且由于 L 、 N 是确定已知的, 故帧同步的主要任务可简化为寻找每一帧数据的起始位置 k_0 。

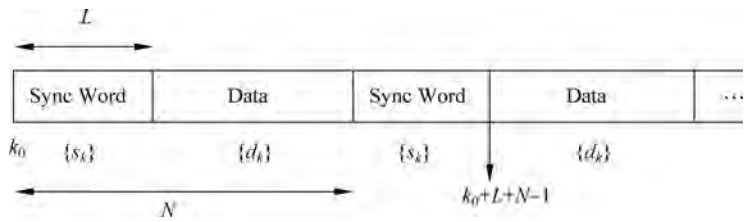


图 5-1 帧结构图

5.3 帧同步基本概念

在卫星通信中,信息数据通常采用帧方式进行传输。帧传输过程中,正确判定每一帧的起始、终止时刻,对于目标数据的提取、译码等后续处理是极为重要的。帧同步的任务就是在定时同步的基础上识别出这些数字信息帧的“开头”和“结尾”时刻,以使接收设备能正确地解释这些帧所代表的信息。

实现帧同步的方法主要有两类^[2]。一类是插入特殊码组法,它是在数字信息码序列中插入一些特殊码组作为每帧的帧头标志,在接收端根据这些特殊码组的位置来实现帧同步。插入特殊码组法又可分为连贯式插入法和间隔式插入法。另一类是自同步法,它是对信息进行适当编码,利用数据组本身之间彼此不同的特性来实现自同步,不需要专门的帧同步码组,这类似于定时同步中的直接法。

5.3.1 帧同步码组

决定第一类帧同步算法性能的关键技术是帧同步码的构造及其检测。帧同步码型选择的主要原则如下:

- (1) 要便于接收端识别,即要求帧同步码具有特定的规律性,如具有尖锐单峰特性的局部自相关函数,例如巴克码组;
- (2) 要使帧同步码的码型尽量和信息码相区别。如同步码的设计应与随机数据的分布有关,当随机数据中“+ 1”出现的概率较高时,同步码应设计为含有“- 1”较多,这样虚警概

率较低,性能较好。

(3) 对同步码组的另一个要求是识别器应该尽量简单;并且要兼顾传输效率。目前常用的帧同步码组有巴克码、Heuman-Hoffman 序列、m 序列以及 Gold 序列等。

1. 巴克码

巴克码是工程上最常用的伪随机码之一,其被广泛应用于帧同步码组的构造。因此本书将以巴克码主介绍基于同步码插入的帧同步算法。巴克码是一个有限长的非周期二进制序列,于 1953 年由 Baker 首次提出,目前已知的巴克码有 7 种,如表 5-1 所示。

表 5-1 巴克码

码长 N	巴 克 码
2	$\{+1, +1\}$
3	$\{+1, +1, -1\}$
4	$\{+1, +1, +1, -1\}, \{+1, +1, -1, +1\}$
5	$\{+1, +1, +1, -1, +1\}$
7	$\{+1, +1, +1, -1, -1, +1, +1\}$
11	$\{+1, +1, +1, -1, -1, -1, +1, -1, -1, +1, -1\}$
13	$\{+1, +1, +1, +1, +1, -1, -1, +1, +1, -1, +1, -1, +1\}$

对于一个 n 位的巴克码 $\{c_1, c_2, \dots, c_n\}$, 其中 n 为巴克码的长度, 对于任意的 c_i , 取值为 ± 1 。其局部自相关函数为

$$R(j) = \sum_{i=1}^{n-j} c_i c_{i+j} = \begin{cases} n, & j = 0 \\ 0, & 0 < j < n \\ 0, & j \geq n \end{cases}$$

以 7 位巴克码序列的自相关函数为例, 其自相关函数特性图如图 5-2 所示。

巴克码识别器是比较容易实现的, 以 7 位巴克码为例, 用 7 级移位寄存器、相加器和判决器就可以组成识别器, 如图 5-3 所示^[4]。

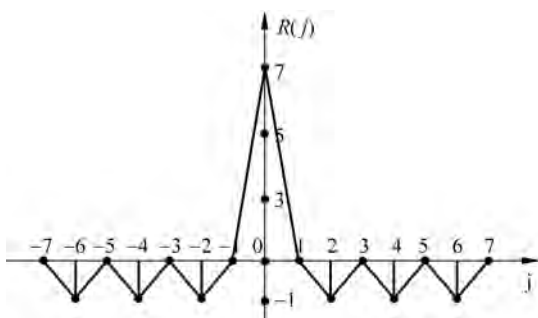


图 5-2 7 位巴克码的自相关函数特性

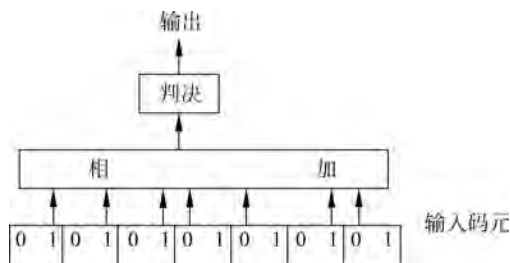


图 5-3 7 位巴克码识别器

当输入数据的 1 存入移位寄存器时, 1 端的输出电平为 $+1$, 而 0 端的输出为 -1 ; 反之, 存入数据 0 时, 0 端的输出电平为 $+1$, 1 端的电平为 -1 。各移位寄存器输出端的接法和巴克码的规律一致, 这样识别器实际上就是对输入的巴克码进行相关运算。当 7 位巴克码在图 5-4(a) 中的 t_1 时刻正好已全部进入了 7 级移位寄存器时, 7 个移位寄存器输出端都输出 $+1$,

相加后得最大输出+7；若判别器的判决门限电平定为+6,那么就在 7 位巴克码的最后一位 0 输入识别器时,识别器输出一帧同步脉冲表示一帧的开头,如图 5-4(b)所示。

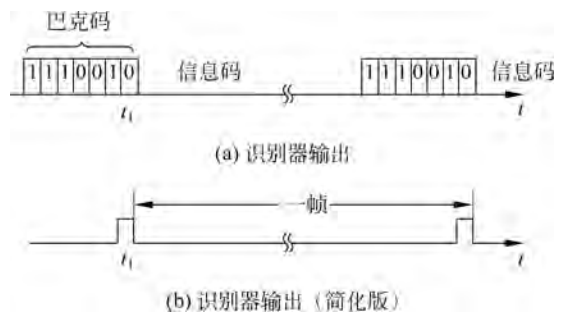


图 5-4 识别器的输出波

由于巴克码的码组只有 8 种,所以实际应用中也可采用伪随机 PN 序列。PN 序列由线性反馈移位寄存器(Linear Feedback Shift Register,LFSR)产生。 N 级的 LFSR 的生成序列的周期 $T=2N-1$ 。

图 5-5 为一个 N 级 LFSR。该模块由 N 个寄存器构成,各个寄存器从上级到下级依次传输存储信号,其运算结果又反馈到输入端。 C_i 表示反馈连接的状态,当 $C_i=0$ 时,反馈连接断开;当 $C_i=1$ 时,反馈连接连通。

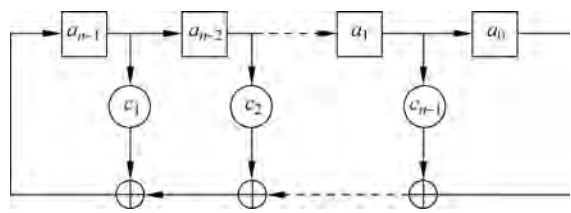


图 5-5 N 级线性反馈移位寄存器

反馈连接可以由如下的本原多项式描述：

$$f(x) = \sum_{i=1}^n C_i a_{n-i}$$

作为一种帧同步码,PN 序列的局部自相关函数同样具有尖锐的峰值,但相对于巴克码,其长度具有更多的选择性,因此广泛应用于帧同步系统。

表 5-2 列举了几组常用的最佳帧同步码。它们都符合 IRIG(Inter Range Instrumentation Group)标准并且它们在随机信息码中和有噪声干扰的情况下发生假同步的概率最小。

表 5-2 最佳帧同步码

帧同步码组位数(bit)	帧同步码组
8	B8H
16	EB90H
24	FAF320H
32	FDB18540H

5.3.2 帧同步过程

帧同步实现的方法主要有两种：逐位调整法和置位调整法^[6,5]。逐位调整法的基本原理是调整接收端本地帧同步码的相位,使之与收到的总码流中的帧同步码对准。

置位调整法也称为预置启动法。在未同步期间,接收设备处于特定的预置状态输入码流逐比特进入帧同步码组检测电路,一旦其中全部位码元与规定的帧同步码组码型相同,就立即输出一个控制信号,启动接收设备的时序发生器。然后经过一个校验周期的时间检验判断。如果未能建立正确的帧相位关系,就重复搜索过程;如果建立了正确的帧相位关系,就保持这种帧状态并结束搜索过程。

采用置位调整法的帧同步过程有搜索态、校核态和锁定态^[5,6],这3个状态之间的转换如图5-6所示^[5]。

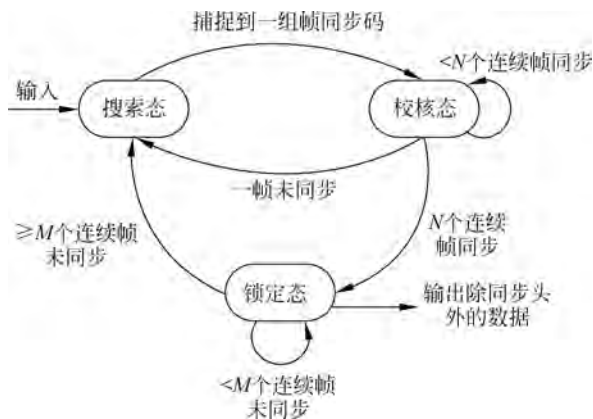


图 5-6 帧同步算法状态转换图

图5-6中为系统设定的由首次搜索态到帧同步头到进入锁定态所需要的次数,称为后方保护时间;在同步锁定状态中连续丢失同步码而退出锁定的次数,称为前方保护时间。

(1) 搜索态: 搜索态即捕捉态,当系统上电或软件清零时,帧同步进入搜索态,开始检测输入的码流中与所插入的帧同步码相同的码组,一旦帧同步码被捕获(不论真假),表明已搜索到一个同步帧头,此时系统进入校核态,否则系统一直处于搜索态直到找到帧同步码为止。

(2) 校核态: 为了防止帧同步中的“虚警”现象,需要对捕获到的帧同步码组进行真假辨别。由于每帧长度固定,所插入的帧同步码组必然会周期性重复出现,而消息码元中引起“虚警”现象的假同步码周期重复出现的概率较小。因此,在找到一组帧同步码后每隔一帧长度需要再次验证是否仍为帧同步码,若连续 N 帧的检验结果均为正确的帧同步码,则认为真是同步,系统进入锁定态;否则认为是假同步,系统重新返回搜索态。称 N 帧的个数为校核帧数, N 帧的时间为后方保护时间。

(3) 锁定态: 为了防止帧同步中的“漏警”现象,需要设定一个锁定帧数 M ,在锁定状态中只有连续 M 帧都没有检测到帧同步码才认为进入了帧失锁,此时系统返回搜索态重新循环3种状态,否则系统仍然处于锁定态。 M 帧的时间又叫前方保护时间,这样即使某帧的帧同步码组出现误码,系统也不会立即进入假失步,从而很好地避免了“漏警”现象的发生。

在数据接收的起始时刻或未同步时,帧同步进入搜索态。在数据流中寻找帧同步码,并且允许帧同步码存在误差。系统所允许的误差位数与同步码的位数有关。计算接收数据与帧同步码的不同位之和,若小于容错位数,则认为搜索到了同步码,进入校核态。为了防止信号中出现虚假同步,找到第一组同步码后跳过一帧长度必须再次确认帧同步码。若经过连续帧确认同步码后,则系统同步正确,立即转入锁定状态;否则存在假同步,返回搜索态。帧同步处于锁定状态时,只有连续帧丢失同步码才进入失步状态,并返回搜索态,否则保持在锁定态^[2,5]。

5.3.3 帧同步性能指标

帧同步系统应该建立时间短,并且在帧同步建立后应有较强的抗干扰能力。通常用漏警概率 P_m 、虚警概率 P_f 和帧同步平均建立时间 t_s 来衡量这些性能。

1. 漏警概率 P_m

由于干扰的影响会引起同步码组中的一些码元发生错误,从而使识别器漏识别已发出的同步码组。出现这种情况的概率就称为漏警概率 P_m 。例如,识别器的判决门限电平为 +6,若由于干扰,7 位巴克码有一位错误,这时相加输出为 +5,小于判决门限,识别器漏识别了帧同步码组;若在这种情况下,将判决门限电平降为 +4,识别器就不会漏识别,这时判决器容许 7 位同步码组中有一个错误码元。现在就来计算漏警概率。设 p 为码元错误概率, n 为同步码组的码元数, m 为判决器容许码组中的错误码元最大数,则同步码组码元 n 中所有不超过 m 个错误码元的码组都能被识别器识别,因而,未漏概率为

$$\sum_{r=0}^m C_n^r p^r (1-p)^{n-r}$$

故得漏警概率为

$$P_m = 1 - \sum_{r=0}^m C_n^r p^r (1-p)^{n-r} \quad (5-1)$$

2. 虚警概率 P_f

在消息码元中,也可能出现与所要识别的同步码组相同的码组,这时会被识别器误认为是同步码组而实现假同步。出现这种情况的可能性就称为虚警概率 P_f 。

因此,计算虚警概率 P_f 就是计算消息码元中能被判为同步码组的组合数与所有可能的码组数之比。设二进制消息码元出现 0 和 1 的概率相等,都为 1/2,则由该二进制码元组成 n 位码组的所有可能码组数为 2^n 个,而其中能被判为同步码组的组合数显然也与 m 有关。若 $m=0$,只有一个(C_n^0)码组能被识别;若 $m=1$,即与原同步码组差一位的码组都能被识别,共有 C_n^1 个码组。以此类推,就可求出消息码元中被判为同步码组的组合数

$\sum_{r=0}^m C_n^r$, 因而可得虚警概率为

$$P_f = 2^{-n} \sum_{r=0}^m C_n^r \quad (5-2)$$

比较式(5-1)和式(5-2)可见, m 增大,即判决门限电平降低时,漏警概率 P_m 减小,但虚警概率 P_f 增大,所以这两项指标是有矛盾的,判决门限的选取要兼顾二者。

3. 平均建立时间 t_s

假设采用置位调整法进行自相关同步,且漏同步和假同步都不发生,那么在最不利的情

另一路输出加至状态触发器 C,使系统由捕捉态转为维持态,这时 Q 端变为高电平,打开与门 2,分频器输出的脉冲就通过与门 2 形成帧同步脉冲输出,因而同步建立。

同步建立后,系统处于维持态。为了提高系统的抗干扰性能,减小漏同步概率,原理图中让触发器在维持态时 Q 端输出低电平去降低识别器的判决门限电平,这样就可以减小漏同步概率。另外,用 $2 \div N$ 电路增加系统的抗干扰性能。建立同步以后,若在分频器输出帧同步脉冲的时刻,识别器无输出,则这可能是系统真正失步,也可能是由于干扰偶尔出现的情况。只有连续出现 $2N$ 次这种情况才能认为是真正失步,这是与门 1 连续无输出,经“非”后加至与门 4 的便是高电平。分频器每输出一脉冲,与门 4 就输出一脉冲,这样连续 $2N$ 个脉冲使“ $2 \div N$ ”电路计满,随即输出一个脉冲至触发器 C,使状态由维持态转为捕捉态。当与门 1 不是连续无输出时,“ $2 \div N$ ”电路未计满就被置 0,状态就不会转换,因而系统增加了抗干扰能力。

建立同步后,消息码元中的假同步码组也可能会使识别器有输出而造成干扰。然而在维持态下,这种假识别的输出与分频器的输出是不同时出现的。因而这时与门 1 没有输出,故不会影响分频器的工作。因此,这种干扰对系统没有影响。

从以上分析可以看出,同步系统的工作划分为捕捉态和维持态后,既提高了同步系统的可靠性,又增加了系统的抗干扰能力。

5.4 基于同步码插入的帧同步技术

由于插入法广泛运用在帧同步电路中,所以本书将首先讨论基于同步码插入的帧同步技术。插入特殊码组法根据特殊码的插入位置可以分为起止式插入法、间隔式插入法和连贯式插入法^[2]。

5.4.1 起止式插入法

起止式同步法只在最早的数字电传机中得到了广泛应用。在电传机中常用的是五单位消息码。为了标记每个字的开头和结尾,在五单位消息码的前后分别加上一个单位的起始码(低电平)和 1.5 个单位的截止码(高电平),共 7.5 个码元组成一个字,如图 5-9 所示。接收端根据高电平第一次转为低电平这一特殊标志来确定一个字的起始位置,从而实现帧同步。但是这种 7.5 单位码元的非整数性给同步数字传输带来了不便。另外,在这种同步方式中,7.5 个码元中只用 5 个码元来传递消息,因此传输效率也很低。



图 5-9 起止式插入同步码组

5.4.2 间隔式插入法

间隔式插入法^[8]又称为分散插入法。它是指将帧同步码组分散地插入信息码流中,即

每隔固定数量的信息码元就插入一位帧同步码元,示意图参见图 5-10。帧同步码组的选择遵循两方面原则:一是要便于接收端识别出它,这就要求帧同步码具有特定的规律性;二是它必须尽量和信息码元区别开。间隔式插入法比较多地用在多路数字电路系统中,例如在 24 路 PCM(Pulse Code Modulation)数字电话系统中一般都采用 0、1 交替码作为帧同步码间隔插入的方法。即这一帧插入 0 码,下一帧插入 1 码,如此交替地插入。由于每帧只插入一位码元,那么它与信息码元混淆的概率将达到 50%,看似这无法检测出帧同步码,但在进行同步捕获时可以连续搜索检测数十帧,只有每帧都符合 0、1 交替的规律才认为是真同步。

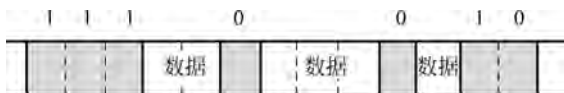


图 5-10 间隔式插入帧同步码组

间隔式分散插入法的最大优点就是帧同步码不占用过多信息时隙,所以每帧的传输效率较高,缺点就是帧同步捕获时间较长。因此这种方法较适合连续信号传输的通信系统,若是断续地发送信号,则会导致较长的帧同步码捕获时间,反而降低了传输效率。

5.4.3 连贯式插入法

连贯式插入法又称为集中插入法,是中频数字接收机中帧同步电路最常用的方法之一,它是指在每帧的开头集中插入作为帧同步码组的特殊码组。该码组应在信息码中很少出现,即使偶尔出现,也不可能按照帧的规律周期出现。在接收端处按帧的周期变化检测该特殊码组,这样便可获得帧同步信息。

连贯式插入法的关键是寻找实现帧同步的特殊码组。首先是帧同步码长的选择,针对一个通信系统,不仅要有较高的信息传输速率,同时必须有较好的抗噪声性能,但这两者通常是矛盾的,帧同步码长的选取也是如此。在传输帧同步码组与消息数据流时,由于数据信息的随机性,在消息码元中可能出现与帧同步码相同的码组,如果设定的帧同步码长较短就会使得在消息码元中发生假同步的概率增大,即把消息码元误判为帧同步码组。但是,如果增加帧同步码位数,那么虽然发生假同步的概率减小了,但在噪声干扰的影响下,越长的帧同步码出现误码的概率就越大,这样势必要延长帧同步码的捕获时间。因此帧同步码长的选择,必须兼顾两者。

其次,为了便于区别信息码元,帧同步码组应具有尖锐单峰特性的局部自相关函数和尽可能低的互相关旁瓣值,以便帧同步电路给出正确的同步指示。国际空间数据系统咨询委员会(Consultative Committee for Space Data Systems, CCSDS)推荐使用的是连贯式插入法,结构如图 5-11 所示。



图 5-11 连贯式插值帧同步码组

5.4.4 传统帧同步检测方法

目前的帧同步检测方法,一般都假设已知同步序列,并且已经实现载波同步和位同步。因此帧同步问题实际上是一个码型已知而出现时刻未知的信号检测问题。

1. 相关法

帧定位问题就是在任意 N 个连续输出的观察值传输符号中,估计帧边界的位置。作为帧同步标志的特殊码组,具有尖锐单峰特性的局部自相关函数。因此可以根据相关法,在已知帧同步码的情况下,求出其在不同延时时刻各码元对应的相关值之和。若在某一延时时刻,该值出现最大值,且该值大于某一个门限值,则认为该时刻出现了帧同步码的第一个码元。

1) 硬判决检测

基于硬判决的帧同步检测首先将输入信号 $r(k)$ 按最大后验概率准则判决如下:

$$d(k) = \begin{cases} 1, & r(k) \geq 0 \\ -1, & r(k) < 0 \end{cases}$$

滑动窗内的采样数据 $r(k) \sim r(k+L-1)$ 的判决值 $d(k) \sim d(k+L-1)$ 有 P 个以上与 SW 相同,则判定当前采样时刻 k 是帧起点位置;否则,等待下一采样时刻,重新检测。在基于硬判决的帧同步检测系统中,度量值 S 的物理意义是判决序列 $d(k) \sim d(k+L-1)$ 中与 SW 序列取值相同的数据的数量。

显然,基于硬判决 SW 检测的帧同步算法其检测性能由系统的误符号率(Symbol Error Ratio, SER)直接决定。由于通信系统的误符号率与符号信噪比 E_s/N_0 唯一相关,故当系统工作于高信噪比时,系统误符号率较低,帧同步算法的检测性能较好。然而,当系统工作于低信噪比时,系统误符号率显著升高,帧同步算法的检测性能将随之出现大幅下滑,甚至完全无法正常工作。

下面推导基于硬判决 SW 检测帧同步算法的检测性能。对于 BPSK 通信系统,误符号率 SER 与信噪比 E_s/N_0 有如下关系:

$$\text{SER} = \frac{1}{2} \text{erfc}\left(\frac{E_s}{N_0}\right) \quad (5-3)$$

则硬判决帧同步算法的检测概率即为滑动窗内 L 个判决数据中,至少有 $P(P \leq L)$ 个与 SW 相同的概率,故

$$P_d = \sum_{i=P}^L C_L^i (1 - \text{SER})^i \text{SER}^{L-i} \quad (5-4)$$

$$P_m = 1 - P_d = \sum_{i=0}^{P-1} C_L^i (1 - \text{SER})^i \text{SER}^{L-i} \quad (5-5)$$

其中, C_L^i 表示 L 中取 i 的组合数。

将式(5-3)代入式(5-4)、(5-5)得

$$P_d = \sum_{i=P}^L C_L^i \left(1 - \text{erfc}\left(\frac{E_s}{N_0}\right)\right)^i \left(\frac{1}{2} \text{erfc}\left(\frac{E_s}{N_0}\right)\right)^{L-i}$$

$$P_m = \sum_{i=0}^{P-1} C_L^i \left(1 - \text{erfc}\left(\frac{E_s}{N_0}\right)\right)^i \left(\frac{1}{2} \text{erfc}\left(\frac{E_s}{N_0}\right)\right)^{L-i}$$

虚警概率的分析限定于全部 L 个采样数据都落入 SW 前方空闲区的情况,则

$$P_f = 0.5^L \sum_{i=P}^L C_L^i \quad (5-6)$$

硬判决帧同步算法只需简单比对滑动窗内的判决数据 $d(k)$ 与帧同步字 SW 的一致性,

即可作出判断,算法原理简单,实现复杂度低,但其检测性能与误符号率直接相关,在低信噪比条件下检测性能较差。

2) 软判决检测

硬判决帧同步检测器直接将输入信号 $r(k)$ 按最大后验概率准则判决为 BPSK 原始调制符号(即 ± 1),虽然简化了后续处理环节,但损失了大量的信息,因而其检测性能较差。针对这一问题,使用软判决帧同步检测算法可以获得更好的性能。

所谓软判决帧同步,是指检测器保留输入信号的原始数值,直接计算滑动窗内数据与帧同步字的相关性,并以此为依据判断当前位置是否为帧起点位置。下面介绍几种常见的软判决帧同步检测器^[9]。

Massey:

$$S = \sum_{n=0}^{L-1} r(n+k)\rho(n) - \sum_{n=0}^{L-1} f(r(n+k))$$

$$f(x) = (N_0/2\sqrt{E_s})\ln(\cosh(2\sqrt{E_s}x/N_0))$$

Massey-AH:

$$S = \sum_{n=0}^{L-1} r(n+k)\rho(n) - \sum_{n=0}^{L-1} |r(n+k)|$$

Massey-AL:

$$S = \sum_{n=0}^{L-1} r(n+k)\rho(n) - \frac{\sqrt{E_s}}{N_0} \sum_{n=0}^{L-1} r^2(n+k)$$

Correlator:

$$S = \sum_{n=0}^{L-1} r(n+k)\rho(n)$$

式中, $\rho(n) = \pm 1$ 表示帧同步字 SW。Massey 检测器是学者 James L. Massey 于 20 世纪 70 年代提出的,已被证明是连续通信系统中(即有周期出现的帧同步字 SW 的系统)、无编码辅助条件下的理论最优检测器。Massey_AH 和 Massey_AL 检测器分别是 Massey 检测器在高、低信噪比条件时的近似,它们避开了复杂非线性函数 $f(x)$ 的运算,且已被证明当 $E_s/N_0 \gg 1$ 、 $E_s/N_0 \ll 1$ 时, Massey_AH、Massey_AL 检测器的检测性能接近最优 Massey 检测器。最后, Correlator 检测器即相关检测器,是在 Massey 检测器被发现前公认的“最优检测器”。然而, Massey 检测器的“理论最优性能”是在连续通信的前提下得到的,对于突发通信系统,其检测性能仍然次于传统的相关检测器。

相关检测器的输出值为

$$\begin{aligned} S(k) &= \sum_{n=0}^{L-1} r(n+k)\rho(n) \\ &= \sum_{n=0}^{L-1} s(n+k)\rho(n) + \sum_{n=0}^{L-1} w(n+k)\rho(n) \\ &= \begin{cases} L + w_\rho(k), & k = k_0 \\ w_\rho(k), & k \neq k_0 \end{cases} \end{aligned}$$

其中, $w_\rho(k) = \sum_{n=0}^{L-1} w(n+k)\rho(n)$ 是均值为 0、方差为 $L\sigma^2$ 的高斯白噪声, $k = k_0$ 表示当

前位置是帧起点位置,固有 $\sum_{n=0}^{L-1} s(n+k)\rho(n) = L$; $k \neq k_0$ 表示当前位置不是帧起点位置,固有 $s(n+k) = 0, n = 0, 1, \dots, L-1, \sum_{n=0}^{L-1} s(n+k)\rho(n) = 0$ (假设 k 处于 SW 前的空白区域)

若判决门限为 P , 即当 $s(k) \geq P$ 时, 判定帧同步成功; 否则判定未同步, 则软判决帧同步算法的漏警概率为

$$P_m = P(S(k) < \text{TH} \mid k = k_0) = \int_{-\infty}^P \frac{1}{\sigma \sqrt{2\pi L}} \exp\left(-\frac{(x-L)^2}{2L\sigma^2}\right) dx$$

虚警概率为

$$P_f = P(S(k) \geq \text{TH} \mid k \neq k_0) = \int_P^{\infty} \frac{1}{\sigma \sqrt{2\pi L}} \exp\left(-\frac{x^2}{2L\sigma^2}\right) dx$$

相关法最大的优点是实现简单, 但基于这种方法检测到的帧起始时刻不是很准确, 变化较大; 且在衰落信道下性能很差。

3) 仿真与分析

性能仿真与分析分别给出硬判决 SW 检测和软判决 SW 检测两种算法取不同判决门限时的虚警、漏警概率。仿真中, SW 序列选为长度 $L=31$ 的 m 序列, 并假设虚警概率、漏警概率同时小于 1×10^{-7} 为通信系统要求的性能指标。

(1) 硬判决 SW 检测性能分析。

由式(5-6)可知, 基于硬判决 SW 检测的虚警概率 P_f 与信噪比无关, 仅与 SW 长度 L 及判决门限 P 有关。图 5-12 给出了虚警概率 P_f 与门限 P 的对应曲线。可见, 若期望 $P_f < 1 \times 10^{-7}$, 则门限 P 的最小取值为 30。

图 5-13 给出了门限 P 取不同值时, 漏警概率 P_m 与符号信噪比 E_s/N_0 的对应关系。可见, 在符号信噪比 E_s/N_0 与 SW 长度 L 确定时, 门限 P 越大, 则漏警概率 P_m 越高。结合图 5-12 对虚警概率 P_f 的仿真结论, 此时 P 应取下限值 30 以获得尽可能低的漏警概率; 同样以 $P_m < 1 \times 10^{-7}$ 为标准, 则此时 E_s/N_0 至少需大于 4.7dB 才能满足指标要求。

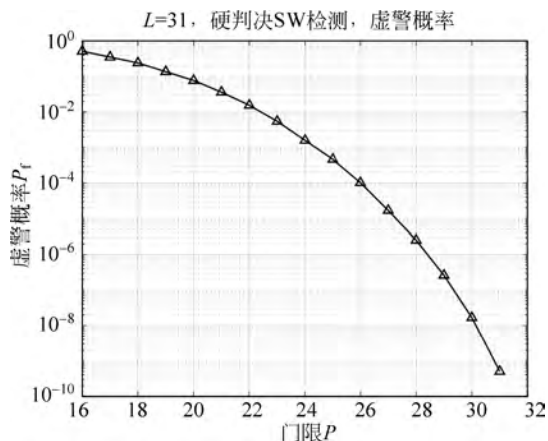


图 5-12 虚警概率 P_f 与门限 P 的对应曲线

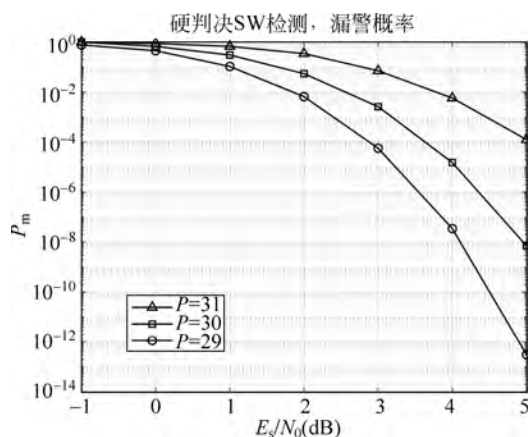


图 5-13 不同门限 P 不同信噪比下的漏警概率 P_m

综合考虑虚警、漏警两项性能指标,对于硬判决 SW 检测帧同步,当 SW 长度 $L=31$ 时,最佳判决门限 $P=30$,此时对应的系统门限工作信噪比约为 $E_s/N_0=4.7\text{dB}$ (以 $P_f, P_m < 1 \times 10^{-7}$ 为准)。

(2) 软判决 SW 检测性能分析。

不同于硬判决检测,基于软判决 SW 检测的帧同步其虚警、漏警概率均与信噪比直接相关。图 5-14 和图 5-15 分别给出了不同判决门限 P 对应的基于软判决 SW 检测的虚警概率和漏警概率曲线。

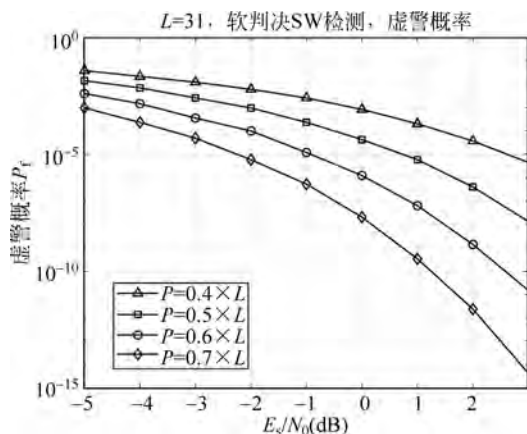


图 5-14 不同 P 对应的软判决 SW 检测虚警概率曲线

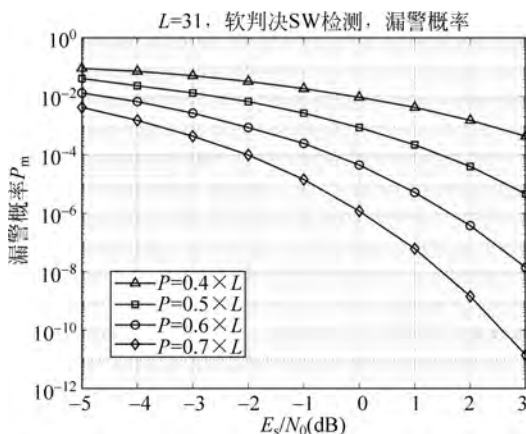


图 5-15 不同 P 对应的软判决 SW 检测漏警概率曲线

仿真结果表明,对于软判决 SW 检测,其门限对性能的影响与硬判决类似:即随着门限的提高,虚警概率 P_f 逐渐降低,而漏警概率 P_m 逐渐升高。因此,为了同时满足 $P_f, P_m < 1 \times 10^{-7}$ 的要求,需折中选择一个恰当的的门限 P ,使得系统门限信噪比最低。由图中曲线可见,当 $P=0.5L$ 时, P_f, P_m 性能相近,此时对应的系统门限工作信噪比约为 $E_s/N_0=2.5\text{dB}$ 。

(3) 无编码辅助帧同步算法性能分析小结。

对于无编码辅助 SW 检测突发帧同步,软判决算法的检测性能明显优于硬判决算法。以 $P_f, P_m < 1 \times 10^{-7}$ 为例,软判决算法的门限信噪比能够比硬判决算法降低约 2.2dB 。当然,软判决算法所需的计算量相比硬判决也更大,但总体而言其计算复杂度仍处于较低水平。因此,在实际无编码辅助帧同步系统中,软判决算法往往是较低信噪比条件下的必然选择。

2. 基于最大似然准则

针对相关性性能不佳、适用范围太窄的弊端,后人提出了一些改进方法,如 Choi Z. Y. 和 Lee Y. H. 提出了基于最大似然改进算法、双相关检测算法^[10,11],该方法适用于存在大频偏的情况,鲁棒性得到了提高。

基于最大似然准则的帧同步检测方法是通过计算似然函数最大值求解帧起始位置的帧同步检测方法,其通用表达式如下:

$$\hat{\mu} = \underset{\mu \in [0, L-1]}{\operatorname{argmax}} p(\mathbf{x} | \mu)$$

其中, $p(\mathbf{x} | \mu)$ 为条件似然函数, μ 表示帧起始位置, L 为同步码序列的长度。

设接收信号为 r_k , 同步码为 s_k , N 为观察值数,数据帧结构为如图 5-1 所示。对应似然

函数表达式为

$$L_0(\mu) = \ln p(\mathbf{x} | \mu) = \sum_{k=0}^{L-1} \sum_{l=0}^{L-1} r_{\mu+k} s_k^* r_{\mu+l} s_l \cdot \text{sinc}\{2\pi U_m(k-l)\} - \sum_{k=0}^{L-1} |r_{\mu+k}|^2$$

其中, $\text{sinc}(\cdot)$ 表示 sinc 函数, $0 \leq U_m \leq 0.5$ 。

基于最大似然法的帧同步检测使系统的性能有了明显的提高,但是 ML 估计器的运算复杂度很高。为了降低复杂度,提出了各种简化的估计器,以性能的降低来换取速度的提高。

文献[10]给出了基于最大似然准则的改进帧检测方法。该方法是通过计算条件似然函数的期望最大值估计帧起始位置。改进的似然函数形式为

$$L_1(\mu) = \sum_{i=0}^{L-1} \left\{ \left| \sum_{k=i}^{L-1} r_{\mu+k} s_k r_{\mu+k-i} s_{k-i}^* \right|^2 - \sum_{k=\mu+i}^{\mu+L-1} |r_k|^2 |r_{k-i}|^2 \right\} \quad (5-7)$$

为了平衡双相关检测项和常规相关检测项之间的误差,可进一步简化为

$$L_2(\mu) = \sum_{i=1}^{L-1} \left\{ \left| \sum_{k=i}^{L-1} r_{\mu+k} s_k r_{\mu+k-i} s_{k-i}^* \right| - \sum_{k=\mu+i}^{\mu+L-1} |r_k| |r_{k-i}| \right\}$$

取 $i=1$,有

$$L_3(\mu) = \left| \sum_{k=1}^{L-1} r_{\mu+k} s_k r_{\mu+k-1} s_{k-1}^* \right| - \sum_{k=\mu+1}^{\mu+L-1} |r_k| |r_{k-1}|$$

式(5-7)的第一项是 $r_{\mu+k} s_k^*$ 和 $r_{\mu+k-i} s_{k-i}^*$ 的相关值的幅度平方,即为相关间隔 i 为双相关检测算法^[11]:

$$L_4(\mu) = \left| \sum_{k=i}^{L-1} r_{\mu+k} s_k^* r_{\mu+k-i} s_{k-i}^* \right|$$

取 $i=1$,为 ad hoc 算法,则

$$L_5(\mu) = \left| \sum_{k=1}^{L-1} r_{\mu+k} s_k^* r_{\mu+k-1} s_{k-1}^* \right|$$

故 ad hoc 算法可看作双相关检测帧同步算法的特殊情况。

近年来,基于最大似然法则针对平坦衰落信道、频率选择性信道的帧同步算法也相继提出。例如,Arkady Kopansky 和 Maja Bystrom^[12]提出了对于平坦衰落信道、非相干解调情况下的 ML 法及其高 SNR 下的近似,随后又研究了针对瑞利衰落信道非周期插入同步模式的帧同步问题^[13],但是他们的研究尚未很好地解决错误传播的问题,即上一个帧同步位置判断不正确会影响到下一个同步位置的判断。

设系统调制方式为 QPSK,载波频率和相位理想同步,帧长为 162,同步码组长 15,对最大似然帧同步检测算法、改进最大似然帧同步检测算法、ad hoc 双相关检测算法进行虚警概率仿真。

图 5-16 给出了归一化频偏 $\Delta f T \in [0, 0.2]$, $E_b/N_0 = 6\text{dB}$ 时对于 $L_0(\mu)$ 给出了 $U_m = 0.02, 0.08, 0.15, 0.3$ 时的仿真结果。从图 5-16 中 $L_0(\mu)$ 的虚警概率变化情况可知,虚警概率和频偏鲁棒性有等价替代关系。随着 U_m 的增加,频偏鲁棒性增强,而虚警概率增加。随着频偏变化率的增加,传统相关算法和 $U_m = 0.02$ 的 $L_0(\mu)$ 的虚警概率迅速增加。而 $L_1(\mu)$ 、 $L_2(\mu)$ 、 $L_3(\mu)$ 、 $L_5(\mu)$ 受频偏的变化影响较小。而且 $L_2(\mu)$ 、 $L_3(\mu)$ 的性能优于 $L_1(\mu)$ 。这是因为式(5-7)中的两项存在误差。

图 5-17 给出了归一化频偏 $\Delta f T \in [U_m, -U_m]$, $U_m = 0.01, 0.04, 0.1$ 时不同信噪比下各帧同步算法虚警概率变化曲线。如图 5-17 所示, $L_2(\mu)$ 的性能最优。

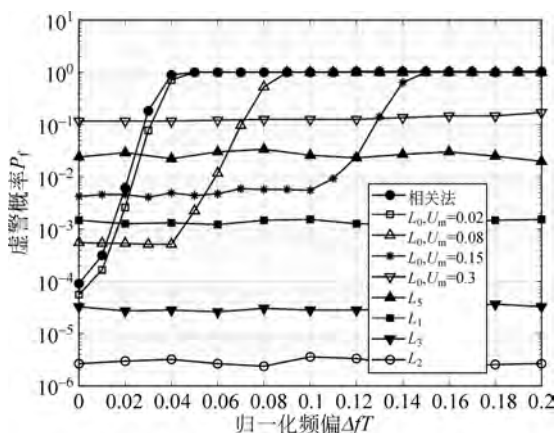


图 5-16 当 $E_b/N_0=6\text{dB}$ 时不同归一化频偏下基于最大似然准则的帧同步算法虚警概率

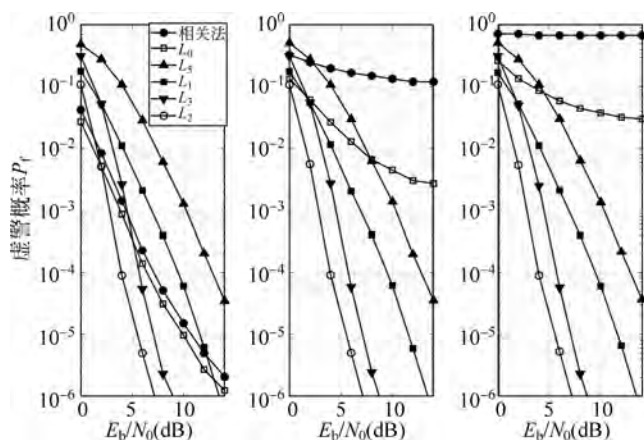


图 5-17 不同 SNR 下各帧同步算法虚警概率变化曲线

3. 似然比检验

假设检验理论已经提出了很久,近年来,Chiani M. 和 Martini M. G. 将假设检验理论中的似然比检验 (Likelihood Ratio Test, LRT) 或广义似然比检验 (Generalized Likelihood Ratio Test, GLRT) 应用于帧同步检测^[14]。

假定同步器如下工作: 从位置 k 开始, 观察连续 N 个采样组成的矢量; 基于该矢量决定该位置处是否为同步码或同步字符 (Synchronization Word, SW); 如果不是 SW, 则移到下一个位置 $k+1$, 重复上述步骤。这样就把帧同步检测问题转换为研究在比特流的每一个位置 k 处, SW 是否存在的判决问题。

令 $r=(r_1, r_2, \dots, r_N)$ 为 N 个接收信号的采样值, 同步器必须在以下 2 种可能的情况中做出选择:

$$H_0: r_i = d_i + n_i, \quad i=1, 2, \dots, N$$

$$H_1: r_i = c_i + n_i, \quad i=1, 2, \dots, N$$

其中, $(d_1, d_2, \dots, d_N)$ 为信息数据, $d_i \in \{-1, +1\}$; $(c_1, c_2, \dots, c_N)$ 为帧同步码序列; $c_i \in \{-1, +1\}$ 。

用 $f_{\mathbf{R}}(\cdot)$ 表示随机矢量 $\mathbf{R}$ 的概率密度函数, λ 表示判决门限, D_0 和 D_1 表示与 H_0 和

H_1 对应的判决信息数据,则似然比检验为

$$\Lambda(\mathbf{r}) = \frac{f_{\mathbf{r}|H_0}(\mathbf{r} | H_0)}{f_{\mathbf{r}|H_1}(\mathbf{r} | H_1)} \underset{D_1}{\overset{D_0}{>}} \lambda$$

在 AWGN 信道中,在数据等概分布的情况下,可以得到对数似然比检验:

$$\Lambda = \Lambda(\mathbf{r}) = \sum_{i=1}^N \ln(1 + e^{-2r_i c_i / \sigma^2}) \underset{D_1}{\overset{D_0}{>}} \lambda$$

GLRT 原则上分为两步:首先在假设 H_0 下估计未知数据矢量 $\hat{\mathbf{d}} = (\hat{d}_1, \hat{d}_2, \dots, \hat{d}_N)$,例如利用最大似然估计,得 $\hat{d}_i$ 的估计值为 $\hat{d}_i = \text{sgn}(r_i)$ 。其次,把估计值当作已知量,利用 LRT,得到 GLRT 为

$$\Lambda_g(\mathbf{r}) = \frac{f_{\mathbf{r}|H_1}(\mathbf{r} | H_1)}{f_{\mathbf{r}|H_0}(\mathbf{r} | H_0, \hat{\mathbf{d}})} \underset{D_0}{\overset{D_1}{>}} \lambda$$

同样,在 AWGN 下,GLRT 的对数形式为

$$\Lambda_g = \Lambda_g(\mathbf{r}) = \sum_{i=1}^N (|r_i| - r_i c_i) \underset{D_0}{\overset{D_1}{>}} \lambda$$

上述 λ 为根据 Neyman-Pearson 准则(即规定最大可容忍的虚警概率)选择的门限。

从对 LRT 和 GLRT 不同特点的研究,可以总结出如下几点:

- (1) LRT 需要已知数据分布,而 GLRT 适用于数据分布未知的情况;
- (2) LRT 依赖于 SNR,或者说,和信道信息有关;而 GLRT 与 SNR 无关,当信道条件信息未知时也适用;
- (3) 随着 SNR 的增加,LRT 趋于 GLRT;
- (4) 因为 GLRT 更易于实现,并且对于实际的 SNR,性能与 LRT 相差不大,很多情况下更偏向于使用基于 GLRT 的同步器。对于低 SNR,二者之间的差距变得很明显,LRT 性能有明显的提高。

设采用 BPSK 调制方式,同步码长 $L = 32$, $E_s/N_0 = -8\text{dB}$ 时对自相关帧同步检测和对数似然比帧同步检测进行仿真,结果如图 5-18 所示。

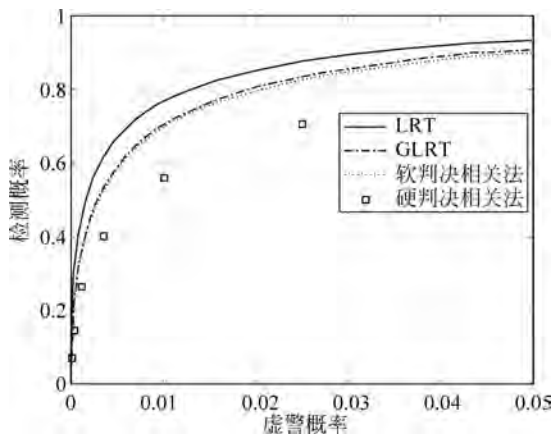


图 5-18 ROC 曲线(检测概率和虚警概率曲线)

从图 5-18 可知, LRT 和 GLRT 帧同步算法检测性能优于硬判决和软判决帧同步算法, 而且 LRT 的帧同步检测性能更优。

为了进一步考查算法性能, 图 5-19 给出了在不同虚警概率、不同门限值下 LRT、GLRT 和软判决自相关帧同步算法的检测概率随信噪比变化曲线。观察可知, LRT、GLRT 算法性能优于软判决自相关帧同步算法, 当 $P_f=10^{-4}$, $P_D=0.9$ 时基于 GLRT 的帧同步检测器优于软判决自相关帧同步检测器 2dB 以上。

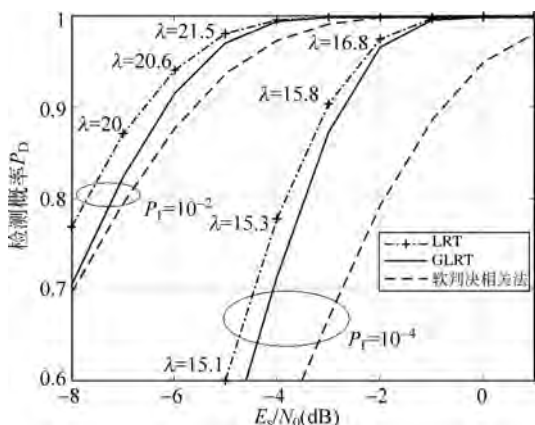


图 5-19 不同虚警概率($P_f=10^{-4}$ 和 $P_f=10^{-2}$)下检测概率随 SNR 变化曲线

5.5 编码辅助帧同步技术

编码辅助的帧同步技术是借助译码器输出后验概率信息, 基于最大似然准则实现帧同步的算法。对应的实现算法是 EM 算法, 只是似然函数中的未知变量变成了帧偏移(符号偏移) k , 算法结构框图如图 5-20 所示。

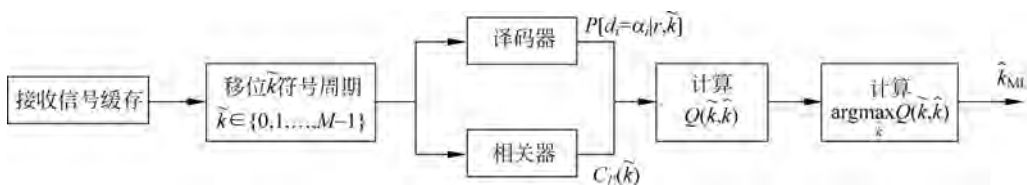


图 5-20 软判决帧同步算法实现框图

设传输帧 r 的帧长为 N , 同步码 p 的长度为 L , 则数据信息 d 长为 $N-L$ 。通过接收信号能量的检测可以判断是否有信号到达, 但是不能得到精确的信号到达时刻, 会有若干符号周期 MT (T 为符号周期) 的偏差。假设此时载波频偏和定时偏差完美同步, 重写接收信号模型如下:

$$r(t) = \sum_{n=0}^{N-1} a_n g(t - \tau - nT) + w(t) \quad (5-8)$$

其中, $r=[p \ d]$ 是传输信息序列, τ 是传输延时, $g(t)$ 是平方根升余弦滤波器脉冲响应。 $w(t)$ 为复加性高斯白噪声, 频谱密度为 N_0/E_s , E_s 为符号功率。

5.5.1 硬判决编码辅助帧同步算法

文献[16][17]提出了两种硬判决帧同步方法,均是根据不同帧偏移时的硬判决矢量满足校验方程的比例作为代价函数,来确定帧边界的。

1. 门限法

设门限为 k ,表示 M 帧数据中满足校验方程的节点数目之和。门限法计算帧偏移的表达式为

$$\hat{k} = \min_{\mu \in W} k \quad (5-9)$$

其中,

$$W = \left\{ k \left| \sum_{i=0}^{M-1} C_{k+iN} \geq \lambda, k \in [0, N-1] \right. \right\}$$

其中, C_i 表示以符号集 $\{r_i, r_{i+1}, \dots, r_{i+N-1}\}$ 为变量节点时满足校验方程的数目, r_i 表示缓存的第 i 个符号信息。

在门限法中,同步错误包括两种情况:第一,所有偏移情况下得到的满足校验矩阵的数目都低于门限值 k ;第二,估计值 $\hat{k} \neq m$ 。由此可得帧同步错误率为

$$\begin{aligned} \text{FSER} &= 1 - \frac{1}{N} \sum_{i=0}^{N-1} (1 - e_a^{(M)})^i (1 - e_b^{(M)}) \\ &= 1 - \frac{1 - e_b^{(M)}}{N e_a^{(M)}} (1 - (1 - e_a^{(M)})^N) \end{aligned} \quad (5-10)$$

其中,

$$\begin{aligned} e_a^{(M)} &= \sum_{i=\lambda}^{MN_c-1} p_0^{(M)}(i) \\ e_b^{(M)} &= \sum_{i=0}^{\lambda-1} p_1^{(M)}(i) \\ p_n^{(2)} &= p_n * p_n \\ p_n^{(M)} &= p_n^{(M-1)} * p_n \end{aligned}$$

“ $*$ ”为卷积标志, $n \in [0, 1]$, p_0 为不同步情况下离散概率密度分布函数, p_1 为同步情况下离散概率密度分布函数, $e_a^{(M)}$ 为虚警概率, $e_b^{(M)}$ 为漏警概率, i 为离散序列索引。

2. 最大值法

最大值法帧同步技术通过下式计算帧偏移:

$$\hat{k} = \operatorname{argmax}_{k \in [0, N-1]} \sum_{i=0}^{M-1} C_{k+iN}$$

因为不同步概率密度函数和同步概率密度函数存在交集,因此估计值会存在虚警概率,也是最大值法中唯一存在的帧同步错误现象。由此可得帧同步错误概率:

$$\text{FSER} = 1 - \sum_{i=0}^{MN_c} p_1^{(M)}(i) \sum_{j=1}^N a(i, j)$$

其中,

$$a(i, j) = \frac{1}{j} \binom{N-1}{j-1} (p_0^{(M)}(i))^i \sum_{l=0}^{i-1} (p_0^{(M)}(l))^{N-j}$$

以门限值法为例,硬判决帧同步译码辅助算法的硬件实现结构如图 5-21 所示。

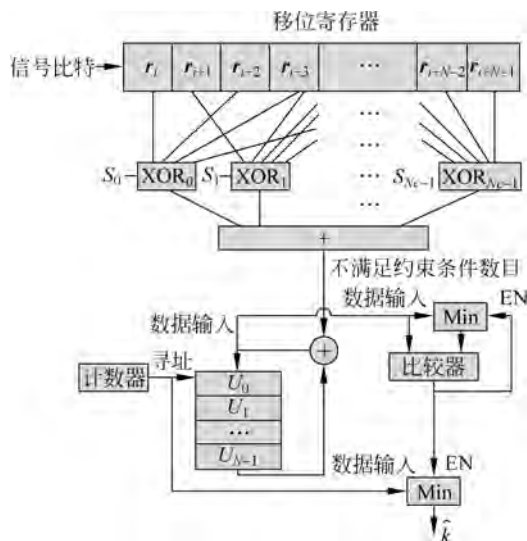


图 5-21 硬判决编码辅助帧同步结构框图

接收信号比特首先被送入移位寄存器,移位寄存器的长度为 N ,与帧长、LDPC 码字长相同,也即数据段处理以码长为单位。利用异或模块求解每个校验节点处校验方程的结果,并将检验方程的结果与扰码序列(PN 序列) S 各对应的比特值比较,并将比较结果送入多操作数加法器。加法器结果就为不满足校验方程 U 的数目。之所以计算不满足的校验方程的数目,是因为当校验方程满足时异或模块输出为 0,不利于表达。 $\{U_0, U_1, \dots, U_{N-1}\}$ 表示用来记录不同帧偏移下不满足校验方程的数目,计数器 Counter 则是用来提供 RAM 单元 $\{U_0, U_1, \dots, U_{N-1}\}$ 的地址。

当 $i < M-1$ (M 为参与计算帧数, N_c 为节点总数)时,多操作数加法器的求和输出与对应 RAM 单元内的数据相加后重新存入原地址 RAM 单元。该 RAM 单元可以采用双口 RAM 实现以保证以上操作能够在一个时钟周期内完成。当 $i = M-1$ 时,执行完双项求和运算后,不再将值存入 RAM 单元,而是与 $\min_U$ 共同送入比较器进行比较,当该值小于 $\min_U$ 时,就将 $\min_U$ 更新,并同时更新 $\min_k$ 为即时 Counter 值。

基于校验方程法则的 LDPC 码辅助帧同步算法复杂度较低,但是要在低信噪比下获得较好的帧同步性能需要多帧 LDPC 码字联合辅助帧同步,这会增加系统的时间复杂度,而且该算法不适用于突发传输系统。基于此,文献[18]提出了基于校验软信息的 LDPC 码辅助帧同步算法。校验软信息定义为 m 个校验方程成立与不成立概率的对数似然比信息之和。

5.5.2 软判决编码辅助帧同步算法

1. 基于 EM 的帧同步算法

帧同步算法的目标是确定帧的起始边界 k , EM 帧同步算法的原理是根据 ML 准则通

过最小化帧同步错误概率 $P[\hat{k} \neq k]$ 实现这一目标。

$$\hat{k}_{\text{ML}} = \underset{\tilde{k} \in U}{\operatorname{argmax}} (\ln p(\mathbf{r} | \tilde{k})) \quad (5-11)$$

其中, U 为包含所有 M 个等概率分布的帧偏移的集合。这里定义完整数据集 $\mathbf{r} = [\mathbf{p} \ \mathbf{d}]$ 。由式(5-8)可得对数似然函数方程, 则

$$\begin{aligned} \ln p(\mathbf{r} | \tilde{k}, d) &\propto -\frac{1}{2\sigma^2} \int_{-\infty}^{+\infty} |r(t) - \sum a_n g(t - \tilde{k}T - nT)|^2 dt \\ &\propto \sum_n \Re \left\{ \int_{-\infty}^{+\infty} r^*(t) a_n g(t - \tilde{k}T - nT) dt \right\} \end{aligned} \quad (5-12)$$

由于 d 与 k 相互独立, Q 函数表达式可重写为

$$\begin{aligned} Q(k, \tilde{k}) &= E_d [\ln p(\mathbf{r} | \tilde{k}, d) | \hat{k}, d] \\ &= \sum_n \Re \{ E_d [a_n | \hat{k}, \mathbf{r}] y_{n+\tilde{k}}^* \} \end{aligned}$$

其中, $y_{n+\tilde{k}} = \int_{-\infty}^{+\infty} r(t) g^*(t - \tilde{k}T - nT) dt$ 表示 $nT + \tilde{k}T$ 时刻匹配滤波器的输出信号。将 Q 函数分解为

$$Q(\tilde{k}, \hat{k}) = C_p(\tilde{k}) + C_d(\tilde{k}, \hat{k}) \quad (5-13)$$

其中,

$$\begin{aligned} C_p(\tilde{k}) &= \sum_{i=0}^{L-1} \Re \{ y_{n+\tilde{k}}^* p_i \} \\ C_d(\tilde{k}, \hat{k}) &= \sum_{i=0}^{N-1} \Re \{ y_{i+L+\tilde{k}}^* \mu_i(r, \hat{k}) \} \end{aligned}$$

其中,

$$\mu_i(r, \hat{k}) = \sum_{\{a_l\}} P[d_l = \alpha_l | r, \hat{k}] \alpha_l$$

表示符号 d_i 的后验概率平均值或称期望值。 α_l 表示星座点集合, $\mu_i(r, \hat{k})$ 是所有星座点的加权平均值, 称为软判决信息。其实传统相关法就是对式(5-13)的第一项求最大, 因此可以将式看作一个广义的相关, 即 $(y_{\tilde{k}}, \dots, y_{\tilde{k}+L-1}, y_{\tilde{k}+L}, \dots, y_{\tilde{k}+L+N-1})$ 与 $(p_0, \dots, p_{L-1}, \mu_0, \dots, \mu_{N-1})$ 相关。

迭代估计过程为: 根据式(5-11)和式(5-12)实现译码辅助估计的迭代过程, 该过程的实现首先需要确定一个初始帧头估计值 $\hat{k}^{(0)}$, 在此基础上采用帧头穷尽搜索选最大方式获得迭代输出 $\hat{k}^{(1)}$, 重复执行这一过程, 直到方程收敛。需要注意的是, EM 算法迭代初始值的确定影响着最终计算结果的正确性, 因为 EM 算法存在局部最大值收敛问题, 即在求解最大值的过程中, 如果初始迭代值距实际全局最大值较远, 则最后求解的结果可能收敛于局部最大值而错失全局最大值。这一局部收敛问题可通过式(5-14)解决:

$$\hat{k}^{(n+1)} = \hat{k}^{(n)} = \hat{k}_{\text{ML}} \quad (5-14)$$

由此求得的最大似然估计应满足:

$$\hat{k}_{\text{ML}} = \underset{\tilde{k}}{\operatorname{argmax}} Q(\tilde{k}, \hat{k}) \quad (5-15)$$

由此可得相应的帧同步器结构,穷尽搜索过程可按照式(5-15)进行计算。EM 算法实现流程图如图 5-22 所示。

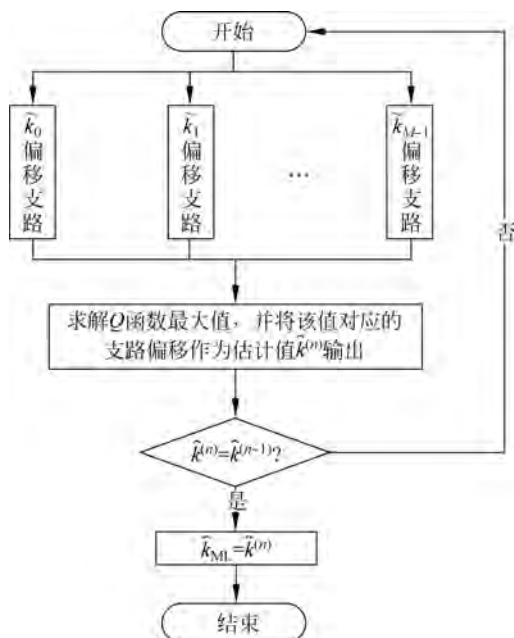


图 5-22 EM 算法流程图

2. 盲帧同步算法

盲帧同步算法是基于帧长与码长相等的假设,首先由接收机检测载波功率对接收数据帧的帧头位置进行粗估计。然后再由 LDPC 译码输出对数似然比信息的幅度均值作为代价函数,对粗估计的帧偏移进行判断。由此可得系统模型如图 5-23 所示。

$$\hat{k} = \arg \max_{k \in [-M, M]} \frac{1}{N} \sum_{i=k}^{\hat{k}+N-1} |L(c_i)|$$

其中, $L(c_i)$ 表示对应第 i 个码字的译码输出软信息。 $i=1, 2, \dots, N$, N 为帧长和码长。采样和缓存模块要缓存至少 2 帧数据用于帧偏移调整。载波粗检测是指根据接收信号能量增长与否对帧起始位置进行的粗略估计。

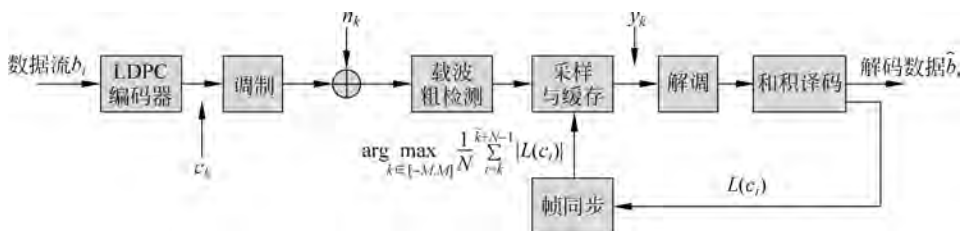


图 5-23 盲帧同步算系统框图

EM 帧同步算法与盲帧同步算法的实现原理一致,但选择的代价函数(判定标准形式)不同,从软判决帧同步实现过程可以得出两点结论:第一,软判决需要利用译码输出后验概率信息,因此每次迭代过程均需要一次完整的译码过程;第二,帧偏移的估计过程和定时偏

差的估计过程是统一的,不仅表现在估计量单位(时间偏差)上,也表现在估计算法上,因此理论上二者是可以统一的。而这一点也在文献[16]中得到了证明。此外,文献[19]提出了改进的和积译码算法,可利用最小均方差准则的相位误差估计器减少软信息相位误差,进而降低误帧率。

5.5.3 基于 LDPC 码约束条件的编码辅助帧同步算法

1. 算法模型

假设 $\mathbf{c}=(c_1, c_2, \cdots, c_N)$ 为 LDPC 编码后的码字, 经过 BPSK 调制映射为 $\mathbf{x}=(x_1, x_2, \cdots, x_N)$ 。信号通过加性高斯白噪声信道(噪声均值为零, 方差为 σ^2)接收信号序列为 $\mathbf{y}=(y_1, y_2, \cdots, y_N)$ 。图 5-24 给出了连续 $N(M+1)$ 个接收符号的帧结构示意图, 其中, N 表示每一帧的帧长, M 是联合捕获帧起始位置时所需要的总帧数。 $(kN+m)$ 是帧起始位置, 其中 $k \in (0, 1, \cdots, M)$, 将这些位置称为“帧同步位置”, 其他位置称为“非帧同步位置”。帧捕获的目标是给出帧同步位置 m 的估计值 $\hat{k}$, 其中 $\hat{k} \in [0, N-1]$ 。如果 $\hat{k}=m$, 则表明正确捕获到了帧同步位置。帧跟踪是在帧捕获后对后续帧同步位置进行判断, 确认系统是否仍处于帧同步状态。

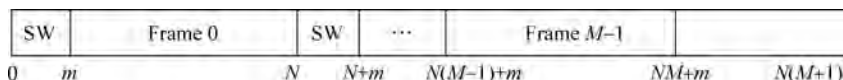


图 5-24 LDPC 编码辅助同步系统帧结构示意图

基于 LDPC 码约束条件的编码辅助帧同步算法结构如图 5-25 所示,主要分 3 个模块^[20]:模块 A 为切换控制模块,模块 B 为 LDPC 译码器,模块 C 为帧同步检测器。在初始的帧捕获阶段,模块 A 中的开关切换至 1,对接收到的信号 y 作硬判决后输入帧同步检测器,通过与门限比较判断是否检测到帧同步位置。当完成帧同步的捕获后,A 中的开关切换至 2,系统进入帧同步跟踪状态。由于 LDPC 译码器在帧同步跟踪状态已经开始工作,因

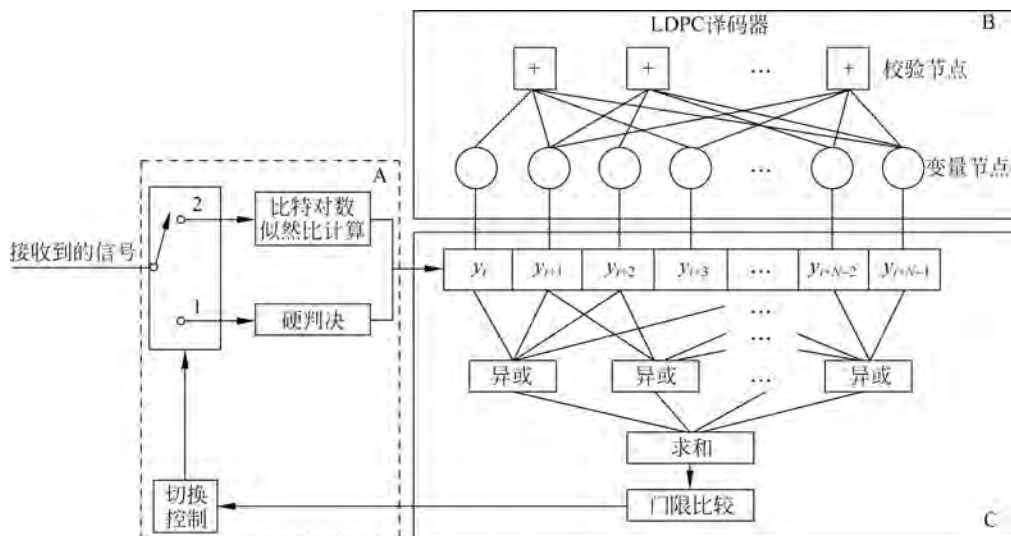


图 5-25 基于 LDPC 码约束条件的帧同步算法示意图

此可将其译码结果输入帧同步检测器。结合图 5-25,下面对基于 LDPC 码约束条件的码辅助帧同步算法进行详细介绍。

基于 LDPC 码约束条件的帧同步算法包括两个阶段:在帧捕获阶段,利用 LDPC 译码前信道符号的硬判决计算校验方程满足的比例,从而快速判断是否捕获到帧同步位置;在帧跟踪阶段,由于译码模块已经开始工作,则利用 LDPC 译码信息对帧同步位置进行跟踪,提高帧同步跟踪的可靠性。其具体步骤如下:

步骤 1,帧同步模块接收到数据后,对数据进行硬判决,得到序列 $\mathbf{y} = (y_j, y_{j+1}, \dots, y_{j+N-1})$,并存储在移位寄存器中;

步骤 2,在帧同步模块中,将硬判决序列 $\mathbf{y}$ 与 LDPC 码的校验矩阵相乘,所得矢量中 0 的个数,即为校验方程成立的个数 Θ_j ;

步骤 3,将校验方程成立的个数 Θ_j 与判决门限 λ 进行比较,如果 $\Theta_j \geq \lambda$,则表示在 $\hat{k}=j$ 处捕获到帧同步位置,接收机进入帧跟踪状态;如果 $\Theta_j < \lambda$,则表示未能捕获到帧同步位置,移位寄存器向右移动一位, $j=j+1$,重复步骤 1 至步骤 3。

显然,门限 λ 的取值对帧同步捕获性能具有重要影响。由式(5-9)如果门限 λ 取值较小,将导致虚警概率增大,漏警概率减小;反之亦然。下面将通过计算机仿真分析门限 λ 对帧同步捕获性能的影响。

在图 5-26 中,最左侧的虚线表示处在非帧同步位置时,满足编码约束条件比例的概率质量函数;右侧的实线簇是不同的 E_b/N_0 下,处在帧同步位置时,满足编码约束条件比例的概率质量函数。从图 5-26 中可以看出,在非帧同步位置,满足码约束比例的概率质量函数均值为 0.5,此时概率质量函数与 E_b/N_0 无关。与之不同,在帧同步位置,随着 E_b/N_0 增加,满足码约束条件比例的概率质量函数均值增大,其方差也逐渐增大。

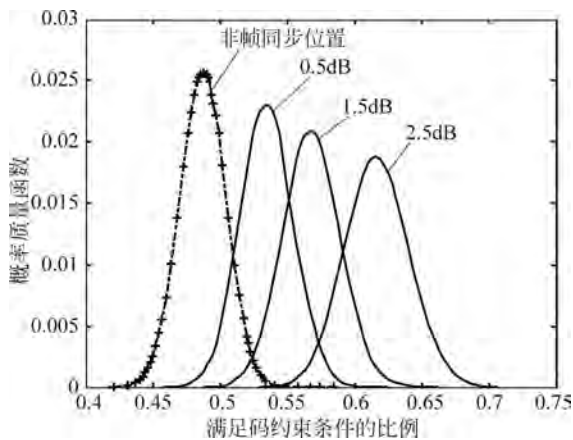


图 5-26 捕获阶段在非帧同步位置与帧同步位置,满足码约束条件比例的概率质量函数

根据图 5-26 给出的概率质量函数,可以确定采用不同门限 λ 时帧同步捕获算法的虚警概率和漏警概率,进而由式(5-10)计算误同步率。由于误同步率不是门限的单调函数,因而在每个信噪比下可能存在使误同步率最小的最优门限值 λ_{opt} 。

在 LDPC 编码通信系统中,当完成帧同步捕获后,系统进入帧同步跟踪状态,此时 LDPC 译码器开始工作。因此,可以利用 LDPC 译码器输出的可靠译码信息进行帧同步跟

踪。基于 LDPC 码辅助的帧同步跟踪算法步骤如下:

步骤 1, 计算接收信号 y 的比特对数似然比, 并将其输入 LDPC 译码器, LDPC 译码器开始迭代译码;

步骤 2, LDPC 译码器检测是否满足迭代终止条件, 即 $H_c T = 0$ 或达到最大迭代次数。若不满足迭代终止条件, 则继续进行迭代译码; 若满足条件, 则计算 $H_c T$ 所得矢量中 0 的个数, 记为 Φ ;

步骤 3, 将校验方程成立的个数 Φ 与判决门限 λ 进行比较, 如果 $\Phi \geq \lambda$, 则表示正确跟踪到帧同步位置, 系统仍处于帧同步跟踪状态; 如果 $\Phi < \lambda$, 则表示未能跟踪到帧同步位置, 系统进入跟踪校验状态;

步骤 4, 系统进入跟踪校验状态后, 如果能在下一帧的 $\hat{\mu}$ 位置检测到帧头, 则系统回到跟踪状态, 循环执行步骤 2 和步骤 3, 否则声明帧失步, LDPC 译码器停止工作, 重新回到帧捕获模式。

根据上述帧同步跟踪算法的步骤, 对帧同步位置和非帧同步位置处 LDPC 译码后满足校验方程比例的概率质量函数进行了仿真, 其中, LDPC 译码器最大迭代次数设为 10 次。仿真结果如图 5-27 所示。

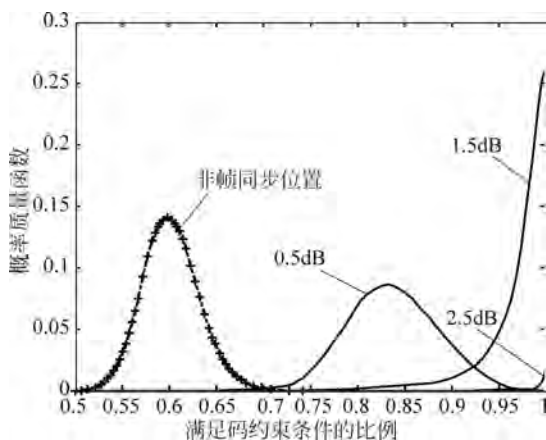


图 5-27 跟踪阶段在非帧同步位置与帧同步位置, 满足码约束条件比例的概率质量函数

由图 5-27 可以看出, 在非帧同步位置, 满足校验方程比例的概率质量函数均值为 0.6, 此时概率密度函数与 E_b/N_0 无关。在帧同步位置, 随着 E_b/N_0 的增加, 满足校验方程比例的概率质量函数均值增大, 其方差也变大。与图 5-26 给出的采用接收信号硬判决作为帧同步检测器输入信号的结果相比, 图 5-27 中采用 LDPC 译码信息作为帧同步检测器输入信号所获得的满足码约束条件比例具有更大的均值, 这是因为 LDPC 译码器能够提供更可靠的比特估计值。因此, 在帧同步跟踪阶段的同步检测性能优于捕获阶段。

值得说明的是, 一方面, 基于 LDPC 码约束的帧捕获算法和基于 LDPC 码辅助的帧跟踪算法共用了模块 C(见图 5-25) 中的寄存器、异或、求和计算单元; 另一方面, 上述逻辑单元可用于实现 LDPC 译码迭代终止条件 $H_c T = 0$ 的判断。因此, 基于 LDPC 码约束条件的编码辅助帧同步算法所增加的接收机复杂度非常小。

2. 仿真分析

本部分对基于 LDPC 码约束条件的帧同步算法进行了性能仿真, 并与传统的基于导频

辅助帧同步算法进行了比较。考虑 IEEE 802.11n 标准所采用的 LDPC 码(1944,972),码率为 1/2,采用 BPSK 调制方式,蒙特卡罗仿真次数为 1×10^6 次。

图 5-28 给出了两种帧同步捕获算法的误帧同步率比较,其中基于导频辅助帧同步算法利用的帧同步码分别由两组和三组 13 比特的巴克码组成^[9]。从图 5-18 中可以看出,首先,增加导频序列长度可以降低误同步率。然而,这种方法将导致传输效率的进一步下降;其次,基于 LDPC 码约束的帧同步捕获算法的性能在高信噪比下优于传统的基于导频辅助帧同步捕获算法。但是在低信噪比下,由于对接收符号进行硬判决的误比特率较高,此时利用 LDPC 编码约束条件并不能为检测器带来增益。此外,图 5-28 中的结果进一步验证了误同步率不是门限的单调函数。在不同的信噪比下,可以找到使误同步率最小的最佳门限值。

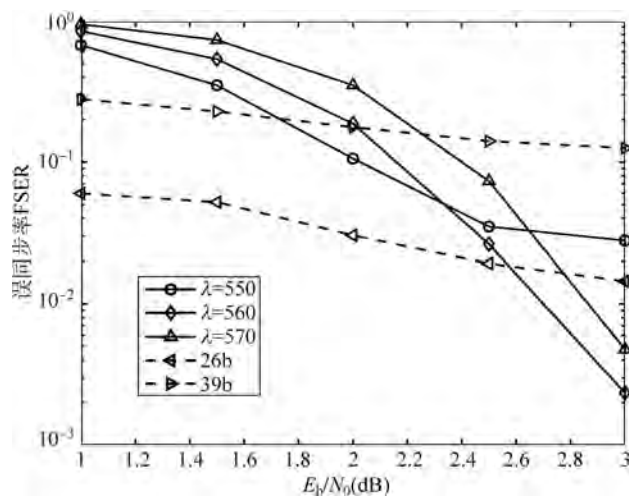


图 5-28 不同信噪比下的误同步率(实线:基于 LDPC 码约束的帧同步捕获算法,虚线:基于导频辅助的帧同步捕获算法)

图 5-29 给出了通过仿真得到的最佳门限值随信噪比的变化曲线。在实际工程应用中,可以通过建立查找表的方式选择最佳同步捕获门限。同理,也可以采用这种方法对基于 LDPC 码辅助的帧同步跟踪算法的最佳门限值进行研究,此处不再赘述。

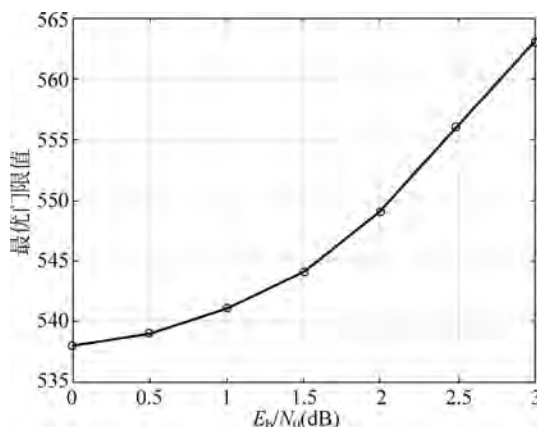


图 5-29 不同信噪比下,基于 LDPC 码约束的帧同步捕获最优门限

图 5-30 给出了基于导频辅助的帧同步检测器、基于 LDPC 码约束的帧捕获检测器和基于 LDPC 码辅助的帧跟踪检测器的工作特性曲线。其中,基于导频辅助的帧同步检测器利用的导频序列是由两组 13 比特巴克码组成。从图 5-30 可以看出,在低信噪比下,基于 LDPC 码辅助的帧跟踪检测器的检测性能明显优于其他两种检测器,这是因为基于 LDPC 码辅助的帧跟踪检测器利用 LDPC 译码器输出的可靠译码信息对帧同步位置进行跟踪,从而提高帧同步跟踪性能。当然,由于其需要进行 LDPC 译码,所以在实际中适用于帧同步位置的可靠跟踪,而其他两种方法适用于帧同步位置的快速捕获。

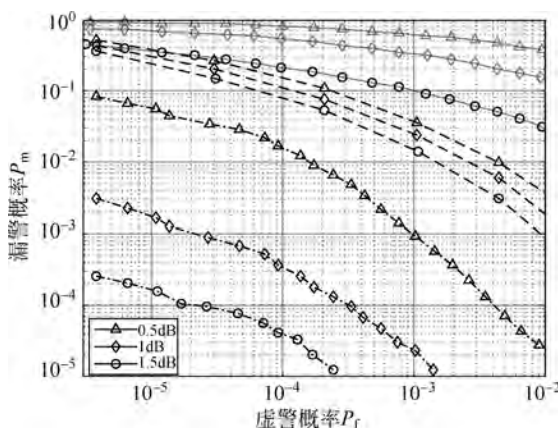


图 5-30 不同检测器的接收机工作特性曲线(实线:基于 LDPC 码约束的帧捕获检测器,虚线:基于导频辅助的帧捕获检测器,点画线:基于 LDPC 码辅助的帧跟踪检测器)

3. 基于 LDPC 编码辅助帧同步算法的时间特性分析

帧同步性能参数除漏警概率 P_m 和虚警概率 P_f 之外还包括平均帧同步捕获时间 T_a 、平均帧同步捕获校验时间 $T_{a,c}$ 、平均帧同步保持时间 T_k 、平均帧同步失帧时间 T_h 、平均失帧间隔时间 T_f 以及平均确认帧失步时间 T_l 等。

1) 平均帧同步捕获时间 T_a

平均帧同步捕获时间 T_a 是指从发现失步立即开始搜索,到第一次检测到帧同步码所经历的平均时间,也是为采取帧同步校核措施的情况下的帧同步捕获时间 $T_{a,c}$ ^[20],其与帧同步建立时间 t_s 是同一个概念。在不采取帧同步校核措施的情况下,第一次检测到帧同步码就确定为进入同步状态,不需要进行帧同步确认。

假设帧长为 N ,帧头长为 m , T 和 T_f 分别表示符号周期和帧周期且 $T_f = NT$,那么一帧数据中平均有 $N-m$ 个非同步码位可能出现虚警。则经过 LDPC 编码的帧同步位置是唯一的,而非帧同步位置个数为 $N-1$ 个。如果在非帧同步位置上没有发生虚警现象,那么在非帧同步位置上的停留时间为一个符号周期 T ;如果在非帧同步位置上发生虚警现象,那么系统认为检测到帧起始位置,将其误判为帧同步位置,从而帧同步模块会跳过一帧的时间间隔 T_f ,进入帧同步跟踪检测状态。在发生虚警现象的情况下,由于系统误判,导致在非帧同步位置处停留时间变长进而会增加平均帧同步捕获时间。需要说明的是,一般情况下,LDPC 编码系统要求的虚警概率 $e_a^{(M)}$ 很低,所以连续几帧在同一个非帧同步位置上发生虚警现象的概率非常小。

平均帧同步捕获时间 T_a 可以分以下 3 部分进行计算,首先计算在非同步码位上的停

留时间 $\Delta t'_a$, 其次计算从非同步位到同步位所经历的时间 t'_s , 最后计算在一定漏警概率条件下的结果。

图 5-31 是考虑发生虚警现象时的帧同步捕获流程图, 其中 H_0 表示系统处于非帧同步位置。在非帧同步位置处的平均停留时间 $T_{a,1}$ 可以表示为

$$\begin{aligned}
 T_{a,1} &= (1 - P_f)T + P_f(1 - P_f)(T + T_f) + P_f^2(1 - P_f)(T_s + 2T_f) + \cdots \\
 &= (1 - P_f)T(1 + P_f + P_f^2 + P_f^3 + \cdots) + (1 - P_f)T_f(P_f + 2P_f^2 + 3P_f^3 + \cdots) \\
 &= (1 - P_f)T \sum_{k=0}^{\infty} P_f^k + P_f(1 - P_f)T_f \sum_{k=1}^{\infty} kP_f^{k-1} \\
 &= T + P_f T_f \times \frac{1}{1 - P_f}
 \end{aligned}$$

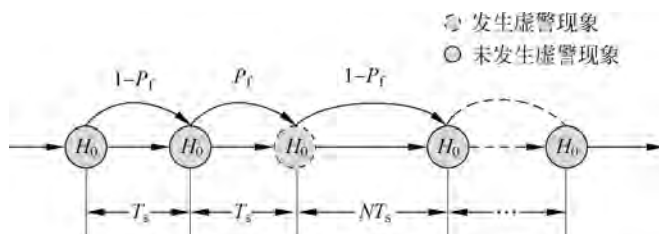


图 5-31 帧同步捕获流程图

式中, P_f 表示在非帧同步位置时, 基于 LDPC 码约束的帧同步捕获检测器的虚警概率。考虑到 $0 < P_f < 1$, 上式可简化。

采用 LDPC 码约束的帧同步捕获算法时, 在帧同步位置处不会发生虚警现象, 所以不考虑在帧同步位置处的停留时间。

其次, 计算帧同步捕获检测器从非帧同步位置转移到帧同步位置的时间 $T_{a,2}$ 。在一帧数据符号中, 有 $N-1$ 个非帧同步位置, 并且在这些位置处都可能发生虚警现象。那么, 从非帧同步位置移动到帧同步位置持续的最长时间 $T_{a,2}^{\max}$ 和最短时间 $T_{a,2}^{\min}$ 分别为

$$\begin{aligned}
 T_{a,2}^{\max} &= (N-1)T_{a,1} \\
 &= \frac{(N-1)[(N-1)P_f + 1]T}{1 - P_f} \\
 &= \frac{[(N-1)P_f + 1]T_f}{1 - P_f} - T \\
 T_{a,2}^{\min} &= T
 \end{aligned} \tag{5-16}$$

假设在系统开机时, 帧同步捕获检测器所处的初始检测位置是均匀分布的。那么, 从非帧同步位置转移到帧同步位置的时间 $T_{a,2}$ 可以表示为

$$T_{a,2} = \frac{1}{2}(T_{a,2}^{\max} + T_{a,2}^{\min}) = \frac{1}{2} \cdot \frac{[(N-1)P_f + 1]T_f}{1 - P_f} \approx \frac{1}{2}T_{a,2}^{\max} \tag{5-17}$$

最后, 考虑发生漏警现象时, 计算最终的帧同步捕获时间 T_a 。在帧同步位置处可能发生漏警现象, 此时系统会误认为该位置是非帧同步位置, 需要再经过 $T_{a,2}$, 才有可能检测到正确的帧同步位置。因此, 非帧同步位置移动到帧同步位置时, 即在帧同步位置处校验方程成立的个数 Θ_j 超过判决门限 λ 后, 可以推出平均帧同步捕获时间 T_a 为

$$\begin{aligned}
T_a &= (1 - P_m)T_{a,2} + P_m(1 - P_f)(T_{a,2} + T_{a,2}^{\max}) + P_f^2(1 - P_f)(T_{a,2} + 2T_{a,2}^{\max}) + \cdots \\
&= \sum_{k=0}^{\infty} (1 - P_m)P_m^k(T_{a,2} + kT_{a,2}^{\max}) \\
&= T_{a,2} + \frac{P_m T_{a,2}^{\max}}{1 - P_m}
\end{aligned}$$

式中, P_m 表示在帧同步位置处基于 LDPC 码约束的帧同步捕获检测器的漏警概率。考虑到 $0 < P_m < 1$, 上式可以被化简为

$$T_a = T_{a,2} + \frac{P_m T_{a,2}^{\max}}{1 - P_m}$$

将式(5-17)和式(5-16)代入上式, 可得

$$T_a = T_{a,2} + \frac{P_m}{1 - P_m} \times 2T_{a,2} \approx \frac{1 + P_m}{1 - P_m} T_{a,2} = \frac{1 + P_m}{1 - P_m} \cdot \frac{T_f}{2} \frac{[(N - 1)P_f + 1]}{1 - P_f}$$

以帧周期对上式进行归一化, 可得归一化平均帧同步捕获时间为

$$T_a/T_f = \frac{1 + P_m}{2(1 - P_m)} \cdot \frac{[(N - 1)P_f + 1]}{1 - P_f}$$

由上式可以看出, 归一化平均帧同步捕获时间与 LDPC 编码码长 N 、基于 LDPC 码约束帧同步捕获检测器的虚警概率 P_f 和漏警概率 P_m 这 3 个参数有关。

2) 平均帧同步捕获校验时间 $T_{a,c}$

一般情况下, 为了保证系统帧同步捕获的可靠性, 会采取捕获校验。接收机开始工作时, 帧同步模块进入捕获状态, 在此状态下, 对数据帧的每个位置进行检测, 当首次检测到帧头时, 接收机由捕获状态转为捕获校验状态。在捕获校验状态下, 如果连续 α 次在相应的位置检测到帧头, 则说明接收机检测到正确的帧同步位置, 接收机声明帧同步, 开始正常工作。但是如果在捕获校验过程中, 出现任何一次漏警情况, 那么校验计数器会清零重新开始进行帧同步捕获。假设采取 α 次帧捕获校验, 那么从开始帧同步捕获过程到最终系统声明帧同步成功所经过的平均时间即为平均帧同步捕获校验时间 $T_{a,c}$ 。当系统没有采取捕获校验时, 可以认为平均帧同步捕获校验时间等同于平均帧同步捕获时间。

考虑帧同步捕获校验的示意图如图 5-32 所示, 其中 i 表示第 i 次检测到帧同步位置, H_0 表示系统处于非帧同步位置, H_1 表示系统处于帧同步位置。

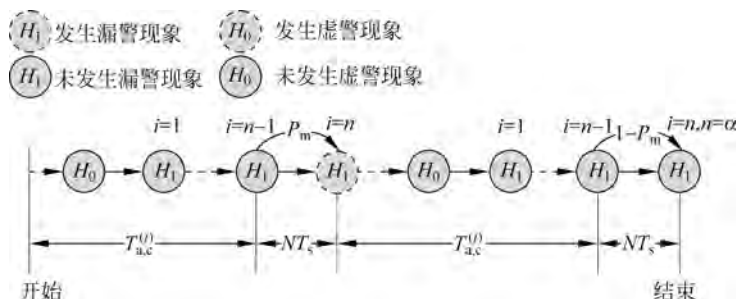


图 5-32 帧同步捕获校验流程图

对图 5-32 可以这样理解：帧同步捕获检测器连续 $\alpha - 1$ 次 ($\alpha > 1$) 在同一个位置检测到帧头, 如果在下一帧的同一个位置处仍然检测到帧头, 那么此时校验计数器 $n = \alpha$, 表示连续

α 次检测到帧同步位置,说明成功实现帧同步捕获;如果在下一帧的同一个位置处发生漏警情况,则校验计数器清零,需要重新开始进行帧同步捕获,直到连续 α 次检测到帧同步位置。因此,可以得到平均帧同步捕获校验时间 $T_{a,c}^{(\alpha)}$ 为

$$\begin{aligned} T_{a,c}^{(\alpha)} &= (1 - P_m)(T_{a,c}^{(\alpha-1)} + T_f) + 2P_m(1 - P_m)(T_{a,c}^{(\alpha-1)} + T_f) + \\ &\quad 3P_m^2(1 - P_m)(T_{a,c}^{(\alpha-1)} + T_f) + \cdots \\ &= \sum_{k=0}^{\infty} (k+1)P_m^k(1 - P_m)(T_{a,c}^{(\alpha-1)} + T_f) \end{aligned}$$

由于 $0 < P_m < 1$, 上式可简化为

$$T_{a,c}^{(\alpha)} = \frac{T_{a,c}^{(\alpha-1)} + T_f}{1 - P_m}, \quad \alpha > 1 \quad (5-18)$$

当 $\alpha=1$ 时,平均帧同步捕获校验时间 $T_{a,c}^{(\alpha)}$ 等于前面推导出的平均帧同步捕获时间 T_a , 即

$$T_{a,c}^{(1)} = T_a = \frac{[(N-1)^2 P_f + 2(1 - P_f)(1 - P_m) + 2(N-1)]T}{2(1 - P_f)(1 - P_m)}, \quad \alpha > 1$$

式(5-18)可以进一步简化为

$$\begin{aligned} T_{a,c}^{(\alpha)} &= \frac{T_{a,c}^{(\alpha-1)} + T_f}{1 - P_m} \\ &= \frac{T_f}{1 - P_m} + \frac{T_f}{(1 - P_m)^2} + \cdots + \frac{T_f}{(1 - P_m)^{\alpha-1}} + \frac{T_{a,c}^{(1)}}{(1 - P_m)^{\alpha-1}} \\ &= \sum_{k=1}^{\alpha-1} \frac{T_f}{(1 - P_m)^k} + \frac{T_{a,c}^{(1)}}{(1 - P_m)^{\alpha-1}} \\ &= \frac{T_f}{P_m} \left[\frac{1}{(1 - P_m)^{\alpha-1}} - 1 \right] + \frac{[(N-1)^2 P_f + 2(1 - P_f)(1 - P_m) + 2(N-1)]T}{2(1 - P_f)(1 - P_m)^\alpha} \end{aligned}$$

进而,可以得到归一化帧同步捕获校验时间:

$$\frac{T_{a,c}^{(\alpha)}}{T_f} = \frac{1}{P_m} \left[\frac{1}{(1 - P_m)^{\alpha-1}} - 1 \right] + \frac{(N-1)^2 P_f + 2(1 - P_f)(1 - P_m) + 2(N-1)}{2N(1 - P_f)(1 - P_m)^\alpha} \quad (5-19)$$

从(5-19)可以看出,前面得到的平均帧同步捕获时间 T_a 是平均帧同步捕获校验时间 $T_{a,c}^{(\alpha)}$ 的特殊情况。因此,本章后面将二者统称为“平均帧同步捕获时间”,即

$$\frac{T_{a,c}^{(\alpha)}}{T_f} = \begin{cases} \frac{[(N-1)^2 P_f + 2(1 - P_f)(1 - P_m) + 2(N-1)]T}{2(1 - P_f)(1 - P_m)}, & \alpha = 1 \\ \frac{1}{P_m} \left[\frac{1}{(1 - P_m)^{\alpha-1}} - 1 \right] + \frac{(N-1)^2 P_f + 2(1 - P_f)(1 - P_m) + 2(N-1)}{2N(1 - P_f)(1 - P_m)^\alpha}, & \alpha > 1 \end{cases} \quad (5-20)$$

由上式可以看出,归一化平均帧同步捕获时间 $T_{a,c}^{(\alpha)}$ 与码长、帧同步捕获检测器的虚警概率 P_f 、漏警概率 P_m 及捕获校验次数 α 有关系。平均帧同步捕获时间 $T_{a,c}$ 是所有帧同步参数中最重要的参数,由于一般系统中帧同步处于系统解调的首要环节,帧同步时间的快慢直接影响到整个系统的稳定速度。特别是对于卫星数字通信系统,一般都要求系统快速进入同步稳定状态。在基于导频辅助的同步系统中还需要在帧同步前提下提取导频信息进行载波同步。有效的同步检测方案和门限参数的选择是进行快速帧同步的关键。

3) 平均确认帧失步时间 T_l

当系统确认帧同步之后,解调和译码模块开始正常工作。为了保证帧同步的有效性,系统会进入帧同步跟踪状态,在帧同步的跟踪状态中,如果在相应的位置处没有检测到帧头,且系统没有采取帧跟踪校验措施,则接收机重新进入帧同步捕获状态。可见,如果不采取适当的帧跟踪校验措施,有可能因为漏警概率较大导致系统频繁地进入帧失步状态。因此,从系统确认帧同步到第一次发现帧失步的平均时间 $T_h^{(1)}$ 可表示为

$$T_v^{(1)} = T_f P'_m + 2T_f (1 - P'_m) P'_m + 3T_f (1 - P'_m) 2P'_m + \dots$$

$$= \sum_{k=1}^{\infty} k T_f P'_m (1 - P'_m)^{k-1}$$

由于 $0 < 1 - P'_m < 1$, 上式可简化为

$$T_v^{(1)} = \frac{T_f}{P'_m} \quad (5-21)$$

其中, P'_m 表示系统进入帧同步跟踪状态后,帧同步跟踪检测器利用 LDPC 译码器输出的可靠译码信息得到的漏警概率。通过图 5-30 可以看出,当 E_b/N_0 为 1.5dB 时,虚警概率为 1×10^{-4} 时,基于导频辅助的帧跟踪检测器的漏警概率为 0.3,而基于 LDPC 码辅助的帧跟踪检测器的漏警概率 5×10^{-5} ,分别代入式(5-21)发现,后者会保证系统长时间处于帧同步状态。如果系统采用帧跟踪校验措施,后者会使系统的帧同步保持时间更长。

在系统采取帧跟踪校验措施后,如果相应的位置处没有检测到帧头,即发生漏警情况,则接收机进入跟踪校验状态,在该状态下,LDPC 译码器正常工作,若能在相应的位置处检测到帧头,则接收机回到跟踪状态;如果系统发生帧失步,并且不是检测器发生漏警情况,即相应的检测位置是非帧同步位置,那么可能发生虚警现象,导致系统从帧失步进入“假同步”状态。因此,下面研究从发生帧失步到系统确认帧失步所经过的时间,简称为“平均确认帧失步时间” $T_l^{(\beta)}$ 。

采取帧跟踪校验措施的示意图如图 5-33 所示。当系统发生帧失步,即在相应帧头处校验方程成立的个数 Φ 小于判决门限 λ ,此时系统没有检测到帧同步位,帧跟踪校验计数器开始计数,直到连续计数 β 次系统确认帧失步。如果在计数器计数过程中,帧跟踪校验发生虚警现象,则计数器会清零,系统恢复到帧同步跟踪状态,但此时系统工作于“假同步”状态,其实系统已经失步。

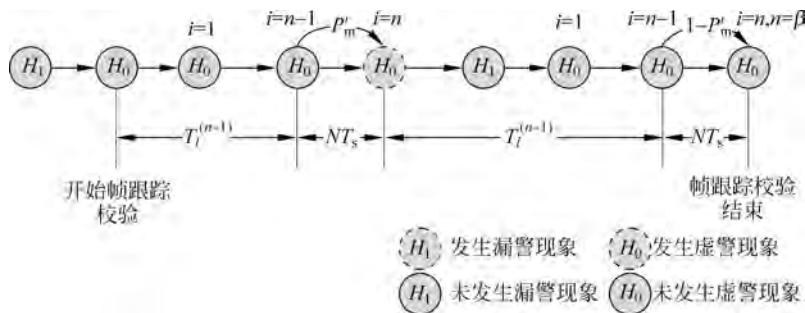


图 5-33 帧跟踪校验流程图

假设系统从第一次检测到帧失步,连续 $\beta-1$ 次检测到帧失步所经过的时间为 $T_l^{(\beta-1)}$ 。如果下一帧没有出现虚警情况,则帧跟踪校验计数器继续累加计数,直到系统设定的 β 次,此时系统确认帧失步,重新进入帧同步捕获状态。如果计数器在没有达到系统设定的 β 次之前出现虚警情况,则计数器清零,重新开始计数。因此,从第一次检测到帧失步,至帧跟踪校验计数器累计为 β 所经过的时间 $T_l^{(\beta)}$ 可以表示为

$$\begin{aligned} T_l^{(\beta)} &= (1 - P'_f)(T_l^{(\beta-1)} + T_f) + 2P'_f(1 - P'_f)(T_l^{(\beta-1)} + T_f) + \\ &\quad 3P'^2_f(1 - P'_f)(T_l^{(\beta-1)} + T_f) + \cdots \\ &= \sum_{k=1}^{\infty} k(1 - P'_f)(T_l^{(\beta-1)} + T_f)P'^{k-1}_f \\ &= \frac{T_l^{(\beta-1)} + T_f}{1 - P'_f} \end{aligned} \quad (5-22)$$

当 $\beta=1$ 时,由于发生帧失步的时刻是均匀分布的,所以从第一次检测到帧失步,再次检测到帧失步所经过的时间 $T_l^{(1)}$ 表示为

$$\begin{aligned} T_l^{(1)} &= (1 - P'_f) \frac{T_f}{2} + P'_f(1 - P'_f) \left(\frac{T_f}{2} + T_f \right) + \\ &\quad P'^2_f(1 - P'_f) \left(\frac{T_f}{2} + 2T_f \right) + \cdots \\ &= \sum_{k=0}^{\infty} \frac{T_f(1 - P'_f)P'^k_f}{2} + \sum_{k=1}^{\infty} kT_f(1 - P'_f)P'^k_f \end{aligned}$$

由于 $0 < P'_f < 1$, 因此上式可简化为

$$\begin{aligned} T_l^{(1)} &= \frac{T_f}{2} + \frac{T_f P'_f}{1 - P'_f} \\ &= \frac{T_f(1 + P'_f)}{2(1 - P'_f)} \end{aligned} \quad (5-23)$$

结合式(5-22)与式(5-23),推导出采取 β 次帧同步跟踪校验的平均确认帧失步时间 $T_l^{(\beta)}$ 为

$$\begin{aligned} T_l^{(\beta)} &= \frac{T_l^{(1)}}{(1 - P'_f)^{\beta-1}} + \left(\frac{T_f}{1 - P'_f} + \frac{T_f}{(1 - P'_f)^2} + \cdots + \frac{T_f}{(1 - P'_f)^{\beta-1}} \right) \\ &= \frac{T_l^{(1)}}{(1 - P'_f)^{\beta-1}} + \sum_{k=1}^{\beta-1} \frac{T_f}{(1 - P'_f)^k} \\ &= \frac{T_l^{(1)}}{(1 - P'_f)^{\beta-1}} + \frac{[(1 - P'_f)^{1-\beta} - 1]T_f}{P'_f} \\ &= \frac{T_f(1 + P'_f)}{2(1 - P'_f)^{\beta}} + \frac{[(1 - P'_f)^{1-\beta} - 1]T_f}{P'_f} \end{aligned}$$

进而可以得到归一化平均确认帧失步时间为

$$\frac{T_l^{(\beta)}}{T_f} = \begin{cases} \frac{(1 + P'_f)}{2(1 - P'_f)}, & \beta = 1 \\ \frac{(1 + P'_f)}{2(1 - P'_f)^{\beta}} + \frac{(1 - P'_f)^{1-\beta} - 1}{P'_f}, & \beta > 1 \end{cases} \quad (5-24)$$

从上式可以看出,采取 β 次帧同步跟踪校验的归一化平均确认帧失步时间 $T_l^{(\beta)}$ 主要由

基于 LDPC 码辅助帧同步跟踪检测器的虚警概率 P_f' 和跟踪校验次数 β 确定。需要说明的是,与传统的帧同步算法不同,式(5-24)和式(5-20)中的虚警概率不相同,分别是基于 LDPC 码约束的帧同步捕获检测器和基于 LDPC 码辅助的帧同步跟踪检测器的虚警概率,这是由于在帧同步捕获和跟踪时采用不同的检测器。

4. 仿真分析

本节将对上述 LDPC 码辅助帧同步算法的时间参数进行仿真,包括平均帧同步捕获时间和平均确认帧失步时间。仿真中考虑 IEEE802.11n 标准中的 LDPC 码(1944,972),码率为 1/2,采用 BPSK 调制方式,经过 10^6 次蒙特卡罗仿真。在仿真中,基于导频辅助的帧捕获检测器采用两组 13 比特巴克码作为导频序列,而基于 LDPC 码辅助的帧跟踪检测器中,LDPC 译码器迭代次数为 10 次。

1) 平均帧同步捕获时间

图 5-34 给出了未采取捕获校验措施时,基于 LDPC 码约束的帧捕获检测器的平均帧同步捕获时间与虚警概率的关系曲线。可以看出,平均帧同步捕获时间和虚警概率之间不是单调的关系,存在一个特定的虚警概率值使得平均帧同步捕获时间最短。并且,可以观察到在相同的虚警概率下,信噪比较高的时候,其平均捕获时间较短,对比图 5-30,这是由于在较高信噪比下,基于 LDPC 码约束的帧捕获检测器性能较好,可以快速捕获到帧同步位置。

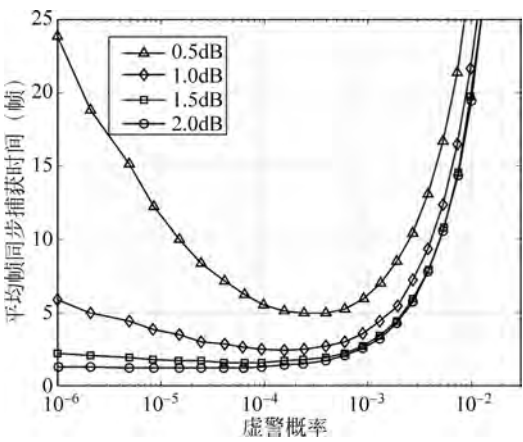


图 5-34 未采取捕获校验措施时,基于 LDPC 码约束的帧捕获检测器的平均帧同步捕获时间

表 5-3 给出了未采取捕获校验时,基于 LDPC 码约束的帧捕获检测器的最优虚警概率、对应的检测概率和平均帧同步捕获时间,这对于工程实现具有一定的参考价值。

表 5-3 未采取捕获校验措施下的平均帧同步捕获时间

E_b/N_0 (dB)	最优的虚警概率	检测概率	平均帧同步捕获时间(帧)
0.5	3.498e-4	0.2778	4.822
1.0	2.188e-4	0.5191	2.336
1.5	1.032e-4	0.7717	1.426
2.0	3.040e-5	0.9256	1.112

图 5-35 和图 5-36 分别是 E_b/N_0 为 2dB 和 3dB 下,采取捕获校验措施时的平均帧同步捕获时间与虚警概率的关系曲线。图 5-35 和图 5-36 中,检测器 A 表示基于 LDPC 码约束

的帧捕获检测器,检测器 B 表示基于导频辅助的帧捕获检测器。从图 5-35 和图 5-36 中可以看出,当校验次数设置为 1~5 时,基于 LDPC 码约束的帧捕获检测器的平均帧同步捕获时间与虚警概率不是单调的关系,存在一个特定的虚警概率值使得平均帧同步捕获时间最短,并且校验次数越多,平均捕获时间越长,验证了式(5-20)的有效性。

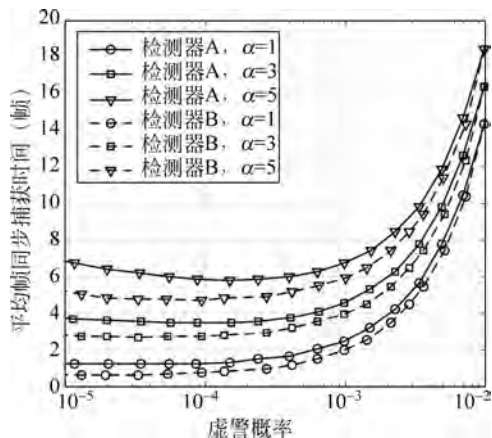


图 5-35 采取捕获校验措施时的平均帧同步捕获时间, SNR=2dB

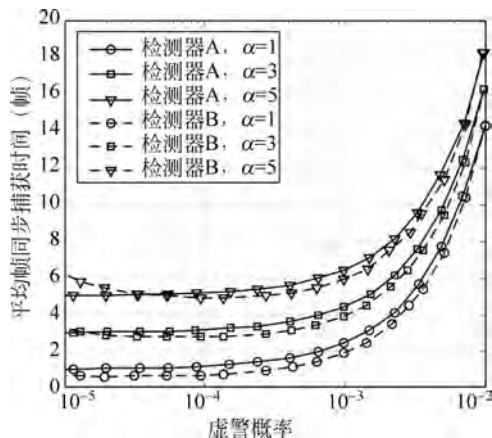


图 5-36 采取捕获校验措施时的平均帧同步捕获时间, SNR=3dB

2) 平均确认帧失步时间

图 5-37 是基于 LDPC 码辅助的帧跟踪检测器在不同校验次数($1 \leq \beta \leq 5$)下的平均确认帧失步时间。由图 5-37 可以看出,当虚警概率小于 10^{-2} 时,不同校验次数下的平均确认帧失步时间趋近于恒定值,并且随着校验次数的增加,平均确认帧失步时间也随之增加。通过仿真发现,基于导频辅助的帧同步检测器与基于 LDPC 码辅助的帧跟踪检测器的平均确认帧失步时间几乎是相同的,因此图 5-37 仅给出了前者的平均确认帧失步时间。

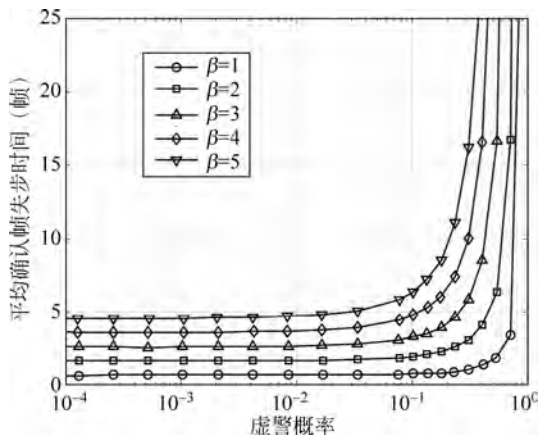


图 5-37 不同校验次数下的平均确认帧失步时间

5.6 DVB-S2 通信系统的帧同步技术

与以前的卫星数字视频广播标准相比,第二代卫星数字视频广播具有更接近香农极限的系统性能^[21]。可实现在低信噪比和大频偏情况下的卫星通信,最低信噪比要求为-2.35dB,而最大载波频率偏移为 5MHz,相对于 25MBaud 的符号率,归一化载波频率偏移为 0.2^[22]。为了实现在这一条件下的帧同步过程,DVB-S2(Digital Video Broadcasting-Satellite2)具有特定的帧结构。

5.6.1 DVB-S2 帧结构

DVB-S2 的物理层帧的结构如图 5-38 所示,可以看到,物理层帧包括 PL Header 和数据部分,PL Header 部分由 SOF 和 PLSC 字段组成共 90 个符号,PLSC 字段为 RM(64,7)码(Reed-Muller)与扰码序列异或的结果。

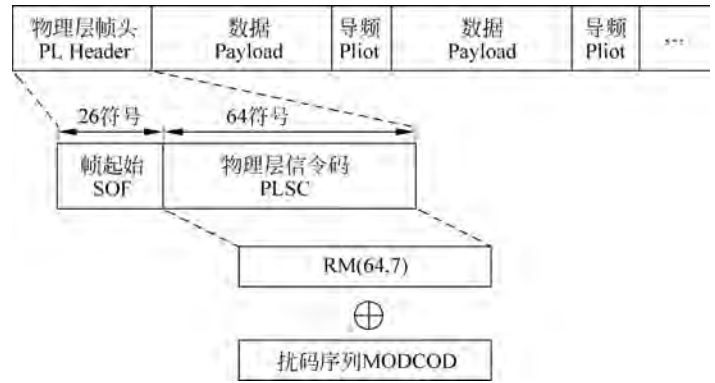


图 5-38 DVB-S2 帧结构

PL Header 中 SOF 字段为 26 个符号,表示物理层帧的起始位置,固定为 18D2E82_{HEX}。PLSC 由 MODCOD 字段经过 RM(64,7)编码得到,MODCOD 字段为 7bit,通知接收机关于调制方案,码率,导频配置,以及 LDPC 编码数据的长度信息等。RM(64,7)码的编码结构如图 5-39 所示^[23]。

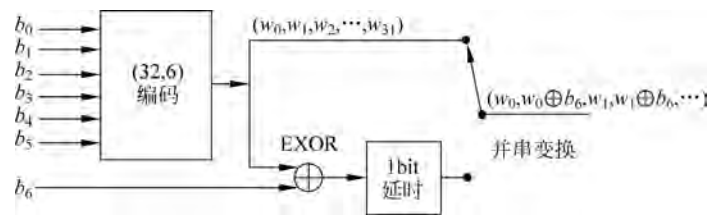


图 5-39 RM(64,7)码编码结构

首先,前 6bit $b_0, b_1, \dots, b_5$ 经过 RM(32,6)码的编码器完成编码,其生成矩阵为

$$G = \begin{bmatrix} 010101010101010101010101010101 \\ 00110011001100110011001100110011 \\ 00001111000011110000111100001111 \\ 00000000111111110000000011111111 \\ 00000000000000001111111111111111 \\ 11111111111111111111111111111111 \end{bmatrix}$$

然后,将得到的 32 位 $w_0, w_1, \dots, w_{31}$ 与第 7bit b_6 进行异或,得到 64 位码字为

$$\begin{cases} c_{2k} = w_k \\ c_{2k+1} = w_k \oplus b_6 \end{cases}$$

其中, $k=0, 1, 2, \dots, 31$ 。

最后,将生成的 64 位与下列加扰序列进行异或完成物理层信令加扰。

0111000110011101100000111100100101010011010000100010110111111010

此外,帧头部分采用 $\pi/2$ -BPSK 进行映射为

$$\begin{cases} z_{2k} = e^{j\frac{\pi}{4}} e^{jy_{2k}\pi} \\ z_{2k+1} = e^{j\frac{3\pi}{4}} e^{jy_{2k+1}\pi} \end{cases} \quad (5-25)$$

其中, $y_{2k} = c_{2k} \oplus s_{2k}$, $y_{2k+1} = c_{2k+1} \oplus s_{2k+1}$, s_{2k} 和 s_{2k+1} 为加扰序列中第 k 个和第 $2k+1$ 个比特, z_{2k} 和 z_{2k+1} 分别为 y_{2k} 和 y_{2k+1} 经 $\pi/2$ BPSK 映射后的符号 $k=0, 1, 2, \dots, 44$ 。

5.6.2 帧同步校核保护方法

为了克服信道噪声和外部干扰对帧同步检测带来的不利影响,在 DVB-S2 系统帧同步检测中也引入了帧同步校核和帧同步保护措施,目前应用最为广泛的是国际电信联盟标准化组织推荐的帧同步校核保护方案。在该方案中,帧同步的实现过程如图 5-40 所示,为了降低虚警概率,加入了同步校核阶段,只有连续在 α 帧数据流的同一位置处检测到了帧同步

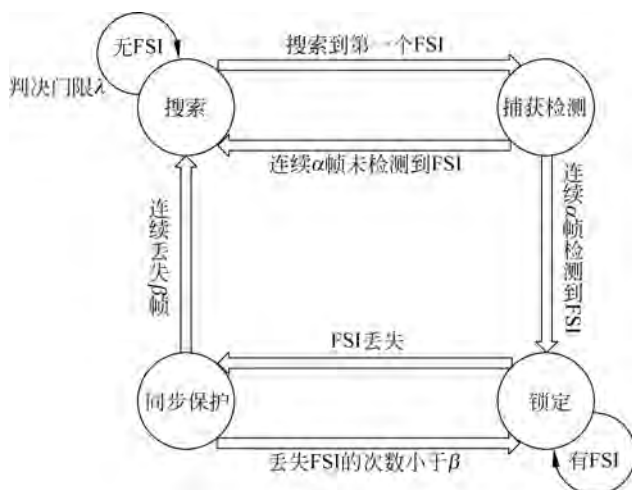


图 5-40 国际电信联盟标准化组织建议的帧同步过程状态图

注: FSI(Frame Synchronization Input)表示帧同步信号

码,即检测峰值输出都大于设定门限,才确认系统进入了帧同步状态;在帧同步保持阶段,为了降低漏警对帧同步的影响,只有连续 β 次检测帧同步码失败,才确认系统帧同步丢失,进入失步状态并重新开始帧同步搜索。

为了更精细地刻画帧同步的技术性能,除了常用的平均帧同步捕获时间,平均帧同步捕获校验时间和平均确认帧失步时间外,下面再补充一些用于衡量帧同步系统性能的时间指标参数定义。

平均帧同步捕获时间 T_a : 系统从失步状态开始,到第一次检测到帧同步码所经历的平均时间。

平均帧同步捕获校验时间 $T_{a,c}$: 在采取 α 次同步校核措施的情况下,从开始帧同步搜索到确认同步所经历的时间,它包括了平均帧同步搜索时间。

平均确认帧失步时间 T_h : 在帧同步保持状态下,从第一次检测到帧同步码失败到最终确认帧失步所经历的平均时间,它包括了平均帧同步保持时间。

平均帧同步保持时间 T_k : 系统在同步保持的情况下,从同步状态到第一次检测到同步丢失的平均时间。

平均帧失步间隔时间 T_l : 在同步状态下,由于噪声等的影响,使得帧同步信号丢失的平均间隔时间。

帧同步各时间指标参数的示意图如图 5-41 所示。其中平均帧同步捕获时间 T_a (或平均帧同步捕获时间 $T_{a,c}$,有校核保护措施时)是帧同步的最重要指标参数,它直接反映了帧同步检测(校核)方法的有效性和系统帧同步的快速性,一般要求同步搜索和捕获时间越短越好。而平均确认帧失步时间 T_h 反映了帧同步保护方案的可靠性,选择合适的同步保护次数不仅能够保证同步的正确性,还能保证有效的同步保持时间,一般要求平均帧同步保持时间 T_k 越长越好。

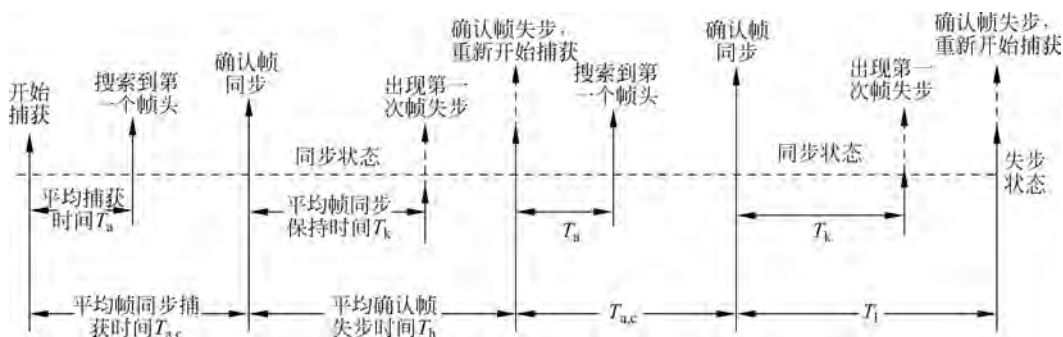


图 5-41 帧同步时间指标参数示意图

5.6.3 基于后验检测积分(PDI)的帧同步检测技术

自相关检测器容易受到频偏和相偏的影响,因此很多学者研究了基于后验检测积分的检测器^[24-26]。基于后验检测积分的检测器首先对接收的信号采取相关操作,然后将其均匀分为 n 组进行差分运算后求和或直接分组求和。此种帧同步检测器主要分为两种类型:差分后验检测积分(Differential Post-Detection Integration, DPDI)和非相关后验检测积分

(Non-Coherent Post-Detection Integration, NCPDI)。3 种 PDI 检测结构 (DPDI-Abs^[24]、DPDI-Real^[25] 和 NCPDI^[26]) 的示意图可用图 5-42 表示。

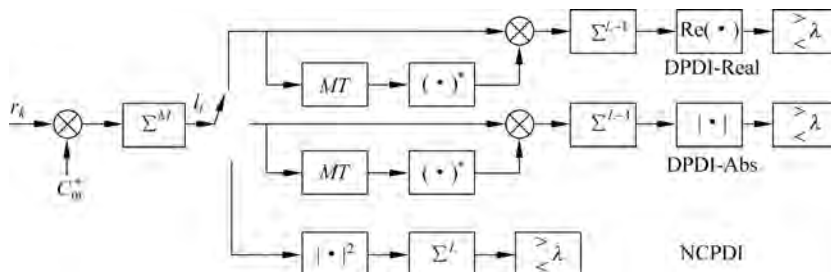


图 5-42 DPDI-Abs、DPDI-Real、NCPDI 帧同步码检测结构图

下面给出 3 种检测器的检测统计量。

DPDI-Abs:

$$\Lambda_{\text{DPDI-Abs}} = \left| \sum_{i=0}^{n-1} \ell_i \ell_{i-1}^* \right|$$

DPDI-Real:

$$\Lambda_{\text{DPDI-Real}} = \text{Re} \left(\sum_{i=0}^{n-1} \ell_i \ell_{i-1}^* \right)$$

NCPDI:

$$\Lambda_{\text{NCPDI}} = \sum_{i=0}^{n-1} |\ell_i|^2$$

其中, $\ell_i = \sum_{k=iL'_p}^{(i+1)L'_p} c_k^* r_k$, L'_p 是后验检测积分长度, 并且满足 $nL'_p = L_{\text{SOF}}$ 。 L_{SOF} 为同步码或者帧头长度。最佳 L'_p 取值时根据系统的频率误差 ΔfT , 利用相干长度积分准则 (Coherent Integration Length Dimensing, CHILD) 得到的^[25]:

$$L'_p \approx \frac{3}{8\Delta fT}$$

对于本例的 DVB-S2 系统, 它的 SOF 同步码段为 26 位, 当归一化频率误差 $\Delta fT = 0.2$ 时, $L'_p \approx 2$, 则 $n \approx 13$ 。

5.6.4 差分检测帧同步方案

针对 DVB-S2 系统, Sun^[116] 提出了一种根据 PL Header 部分固定的 SOF 字段数据和 PLSC 字段的编码规则来实现帧同步, 其结构如图 5-43 所示。可知该方案根据寄存器可分为两部分, 分别对应 SOF 字段和 PLSC 字段。接收数据共轭相乘后送入寄存器中, 前 25 个寄存器的数据乘上对应的 SOF 系数并求和, 随后的 64 个寄存器的数据每隔一个寄存器乘上对应的 PLSC 系数后求和, 求和后的两个结果分别进行相加和相减, 取模后选择其中最大者作为相关值送入峰值检测模块。

由于帧头 SOF 字段数据为固定的 26bit, 那么将经 $\pi/2$ -BPSK 映射后的数据进行共轭相乘即可求得 25 个 SOF 系数, 即

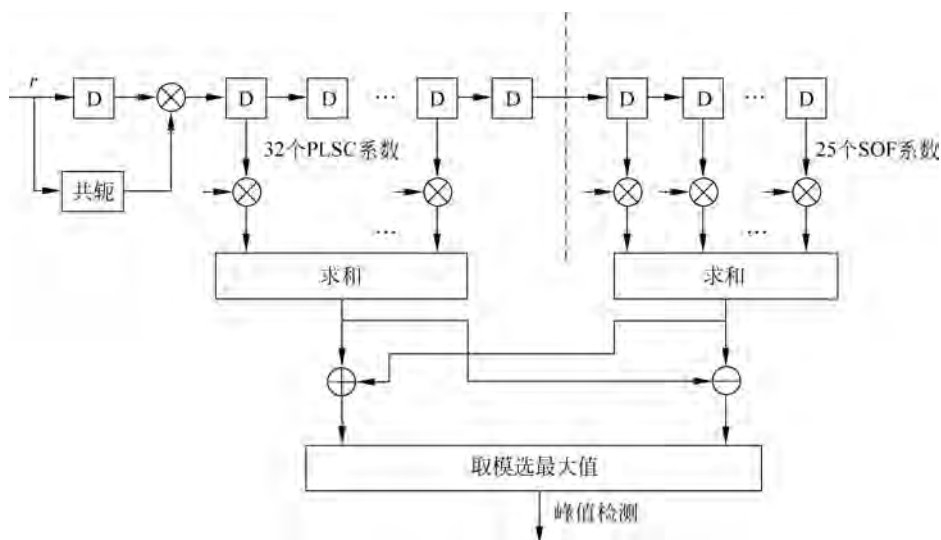


图 5-43 差分检测帧同步方案

$$d_i^{\text{SOF}} = z_i^* z_{i+1} = \begin{cases} e^{j\frac{\pi}{2}} e^{j(y_{i+1}-y_i)\pi} & i = 0, 2, \dots, 24 \\ e^{-j\frac{\pi}{2}} e^{j(y_{i+1}-y_i)\pi} & i = 1, 3, \dots, 23 \end{cases}$$

因此,对于 SOF 字段进行差分相关的相关值为

$$\begin{aligned} \Lambda_{\text{SOF}} &= \sum_{i=1}^{25} r_{i-1} r_i^* d_{i-1}^{\text{SOF}} \\ &= z_i e^{j(2\pi\Delta f i + \theta)} z_{i+1}^* e^{-j(2\pi\Delta f (i+1) + \theta)} z_i^* z_{i+1} \\ &= 25 e^{-j2\pi\Delta f} \end{aligned}$$

由于 PLSC 字段 64 位码字中相邻奇数位与偶数位异或为 b_6 , 即根据 b_6 的取值, 64 位码字奇数位和偶数位相同或相异, 经 $\pi/2$ BPSK 映射后奇数位与偶数位共轭相乘 $z_{2k} z_{2k+1}^*$ ($k=0, 1, \dots, 31$) 为固定值 $+j$ 或 $-j$, 奇数位与偶数位共轭相乘得

$$\begin{aligned} z_{2k} z_{2k+1}^* &= e^{j\frac{\pi}{4}} e^{j(c_k \oplus s_{2k})\pi} e^{-j\frac{3\pi}{4}} e^{-j(c_k \oplus b_6 \oplus s_{2k+1})\pi} \\ &= e^{-j\frac{\pi}{2}} e^{j(c_k \oplus s_{2k} - c_k \oplus b_6 \oplus s_{2k+1})\pi} \end{aligned} \quad (5-26)$$

而对于指数运算 $e^{j(a \pm b)\pi}$, 若 a 和 b 均为 0 或者 1 时, $e^{j(a \pm b)\pi}$ 的值与 $e^{j(a \oplus b)\pi}$ 的值相等, 加法运算为

$$\begin{cases} e^{j(0+0)\pi} = e^{j0\pi} = e^{j(0 \oplus 0)\pi} \\ e^{j(0+1)\pi} = e^{j\pi} = e^{j(0 \oplus 1)\pi} \\ e^{j(1+0)\pi} = e^{j\pi} = e^{j(1 \oplus 0)\pi} \\ e^{j(1+1)\pi} = e^{j2\pi} = e^{j(1 \oplus 1)\pi} \end{cases}$$

减法运算为

$$\begin{cases} e^{j(0-0)\pi} = e^{j0\pi} = e^{j(0 \oplus 0)\pi} \\ e^{j(0-1)\pi} = e^{-j\pi} = e^{j(0 \oplus 1)\pi} \\ e^{j(1-0)\pi} = e^{j\pi} = e^{j(1 \oplus 0)\pi} \\ e^{j(1-1)\pi} = e^{j0\pi} = e^{j(1 \oplus 1)\pi} \end{cases}$$

因此,将式(5-26)改写为

$$\begin{aligned} z_{2k} z_{2k+1}^* &= e^{-j\frac{\pi}{2}} e^{j(c_k \oplus s_{2k} \oplus c_k \oplus b_6 \oplus s_{2k+1})\pi} \\ &= e^{-j\frac{\pi}{2}} e^{j(b_6 \oplus s_{2k} \oplus s_{2k+1})\pi} \end{aligned}$$

若 PLSC 段对应的 32 个系数为 $d_k^{\text{PLSC}} = e^{-j\frac{\pi}{2}} e^{-j(s_{2k} \oplus s_{2k+1})\pi}$, 易见此 32 个系数只与扰码 s 有关, 将 $z_{2k} z_{2k+1}^*$ 与系数 d_k^{PLSC} 相乘得

$$\begin{aligned} z_{2k} z_{2k+1}^* d_k^{\text{PLSC}} &= e^{-j\frac{\pi}{2}} e^{j(b_6 \oplus s_{2k} \oplus s_{2k+1})\pi} e^{j\frac{\pi}{2}} e^{-j(s_{2k} \oplus s_{2k+1})\pi} \\ &= e^{jb_6\pi} \end{aligned} \quad (5-27)$$

由式(5-27)可知,接收数据共轭相乘并与对应的系数相乘后得到的新数据只与 b_6 有关,因此 PLSC 字段可用于实现帧同步。

为了使帧同步对频偏不敏感,接收数据共轭差分后必须满足相位一致条件。那么相邻的接收符号进行共轭相乘得

$$\begin{aligned} r_i r_{i+1}^* &= z_i z_{i+1}^* e^{j(2\pi\Delta f i + \theta)} e^{-j(2\pi\Delta f (i+1) + \theta)} + n_i z_{i+1}^* e^{-j(2\pi\Delta f (i+1) + \theta)} + \\ &\quad n_{i+1}^* z_i e^{j(2\pi\Delta f i + \theta)} + n_i n_{i+1}^* \\ &= z_i z_{i+1}^* e^{-j2\pi\Delta f} + N_i \end{aligned} \quad (5-28)$$

由式(5-28)可知,若不考虑噪声对符号相关的影响 N_i , 相邻接收符号共轭差分后相位一致,这样相同相位的数据累加使得帧同步对频偏不敏感。

那么,根据 PLSC 段数据,相邻的奇数位与偶数位进行差分相关的相关值为:

$$\begin{aligned} \Lambda_{b_6} &= \sum_{k=0}^{31} r_{2k} r_{2k+1}^* d_k^{\text{PLSC}} \\ &= \sum_{k=0}^{31} z_{2k} z_{2k+1}^* e^{-j2\pi\Delta f} e^{j\frac{\pi}{2}} e^{-j(s_{2k} \oplus s_{2k+1})\pi} \\ &= 32 e^{jb_6\pi} e^{-j2\pi\Delta f} \end{aligned}$$

将根据 SOF 字段求得的 Λ_{SOF} 和根据 PLSC 字段求得的相关值 Λ_{b_6} 分别相加和相减,取绝对值后选择其中最大者送入峰值检测模块,根据 SOF 字段和 PLSC 字段进行差分相关的相关值为

$$\Lambda_D = \max\{|\Lambda_{\text{SOF}} + \Lambda_{b_6}|, |\Lambda_{\text{SOF}} - \Lambda_{b_6}|\}$$

本节对 PDI 检测结构的算法性能进行 3 项仿真,并与差分检测结构的性能进行比较。仿真条件为 $\text{SNR} = -2\text{dB}$,归一化频差为 $\Delta f T = 0.2$,系统采用含有导频的 QPSK 调制方式,帧长 $N = 33\ 282$ 个符号。

1. 不同检测结构对频率误差的敏感度仿真

为了分析不同检测结构的频率误差敏感度,图 5-44 给出了频率误差分别为 0 和 0.2 时的 5 种检测结构的接收特性曲线(ROC 曲线)。其中的 DD-SOF 和 DD-PL Header 分别表示利用帧头 26 个符号的 SOF 和利用 90 个符号的 PL Header 进行帧同步检测的仿真结果。由图 5-44 可见,3 种 PDI 检测结构都对频率误差很敏感,在频率误差小的情况下,DPDI-Real 算法性能最优,DPDI-Abs 和 DD-SOF 其次,NCPDI 算法性能最差;但是在频率误差比较大时,3 种 PDI 算法受频率误差的影响都比较大,同一虚警概率下的漏警概率急剧增大,检测

性能下降,而差分检测结构基本不受频率检测误差的影响,DD-SOF 检测算法性能比其他几种检测结构在频率误差方面的鲁棒性都要好。对于 DVB-S2 特殊的帧结构,当同时利用帧头的 64 位的 PLSCODE 联合进行差分运算时,如图 5-44 中的 DD-PL Header 曲线所示,性能比 DD-SOF 算法更优越。因此,对于像 DVB-S2 这种存在较大频率误差的系统,差分检测结构具有更强的频率适应性。

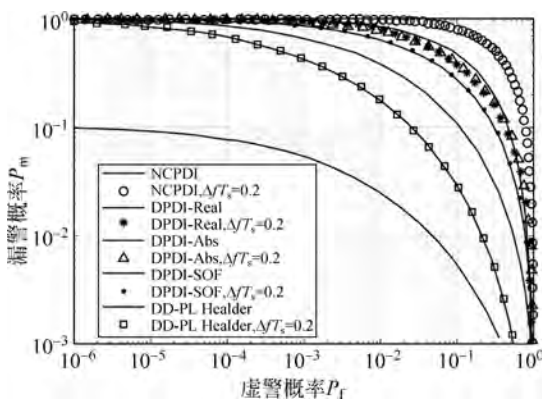


图 5-44 不同检测器的接收特性(ROC)曲线

2. 平均帧同步捕获时间仿真与比较

图 5-45(a)是信噪比为-2dB,频率误差为 0 时,各种检测结构的平均帧同步捕获时间随虚警概率变化的曲线。可见,虚警概率与平均捕获时间呈非线性关系,存在一个最佳的虚警概率,使得同步捕获时间最短,这是由于判决门限越大,虚警概率越大,检测概率越小,延长了同步捕获的时间;当判决门限过小时,虚警概率增大,由于虚警导致确认帧同步的时间增加。表 5-4 给出了各检测结构在只采用单次捕获确定帧同步的条件时,最佳虚警概率下的帧同步平均捕获时间列表,表明在没有频率误差影响时,DPDI 和 DD-PL Header 都具有平均捕获时间短的优点。

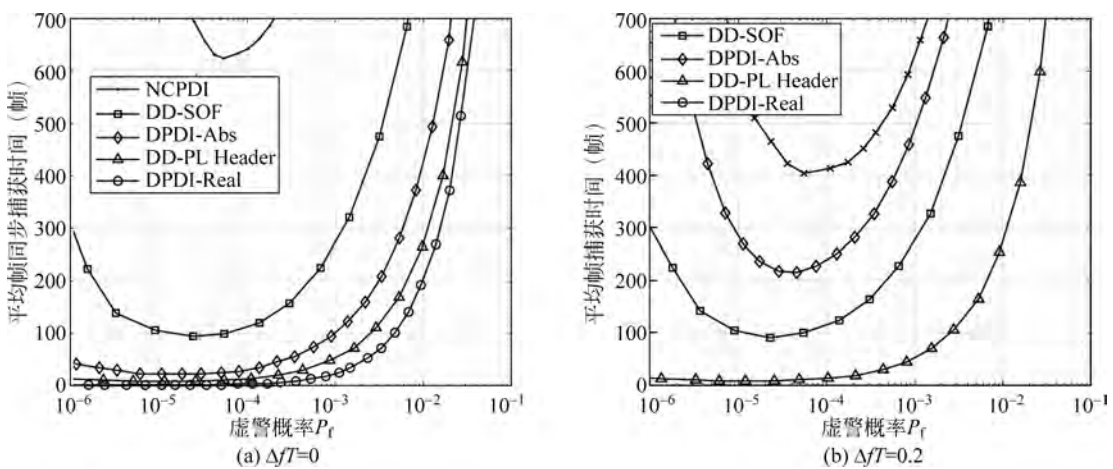


图 5-45 帧同步平均捕获时间与虚警概率关系曲线

表 5-4 单次校核的平均帧同步捕获时间(SNR = -2dB)

频率偏差 ΔfT	检测结构	检测门限	虚警概率 $P_f /$ ($\times 10^{-5}$)	漏警概率 P_m	平均捕获时间 (帧)
0	DPDI-Real	68.2	2.0	0.8989	2
	DD-PL Header	70.5	1.5	0.1741	8
0	DPDI-Abs	29.4	1.0	0.055	24
	DD-SOF	45	3.5	0.0354	98
	NCPDI	154.3	7.5	0.0056	620
0.2	DD-PL Header	69.7	2.0	0.186	9
	DD-SOF	48	2.7	0.021	91
	DPDI-Real	59.4	8.0	0.91	400
	DPDI-Abs	28	5.4	0.86	215

图 5-45(b)表示信噪比为-2dB,频率误差为0.2时,各检测算法的平均帧同步捕获时间变化情况。可见,DD算法的帧同步平均捕获时间基本不变,而DPDI算法的帧同步捕获时间增加,性能远低于DD算法。

由前面的分析可知,PDI检测结构受频差影响大,而差分检测结构几乎不受频率误差的影响。当频差很小时,DPDI-Real的同步性能优于差分检测结构及其他检测结构,但当频率误差较大时,差分检测结构具有同步时间短的优势。因此当频率误差很小时,帧同步适宜采用DPDI检测结构,当频差很大时,差分检测结构比较适合,但其复杂度要高一些。

5.6.5 改进的差分检测帧同步

首先,文献[20]对Sun提出的差分帧检测同步器进行了简化,简化后的硬件结构如图5-46所示,图中 $f_{\text{plsc},i}$ 和 $f_{\text{sof},j}$ 分别表示PLSC和SOF段的系数。由于物理层帧头PLSC段信令信息的第7比特置零,因此可以对Sun提出的差分检测帧同步器进行简化,不需要对 Δ_{plsc} 和 Δ_{sof} 分别求和、求差和比较操作,直接进行求和即可。

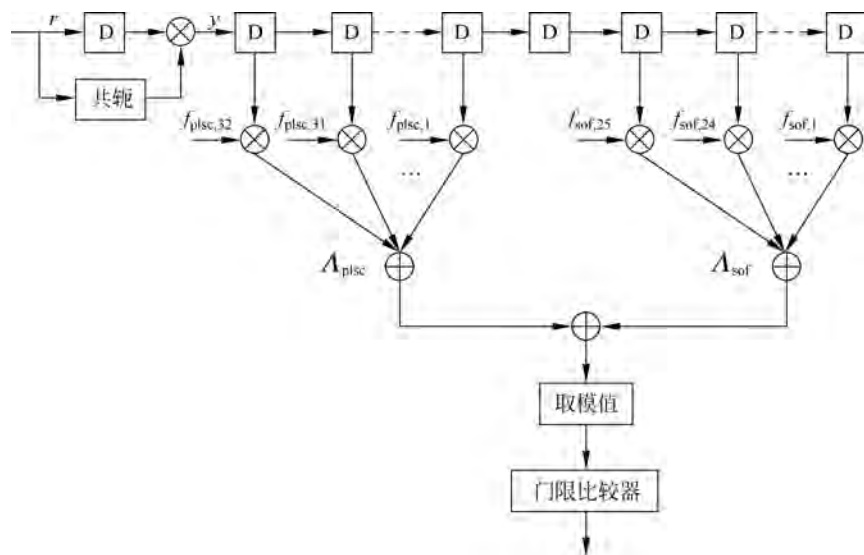


图 5-46 简化的差分帧同步检测器

其次,卫星自适应传输系统为了在短时间内成功捕获到帧头,需要提高帧同步器的检测概率。由于自适应传输系统的每一帧除了帧头之外,还有导频符号可以辅助进行帧同步检测。因此,文献[20]提出了一种联合 SOF 段、PLSC 段和 Pilot 段的帧同步检测器,其结构如图 5-47 所示。下面分别对图 5-47 中 SOF 段、PLSC 段和 Pilot 段的相乘系数进行推导。

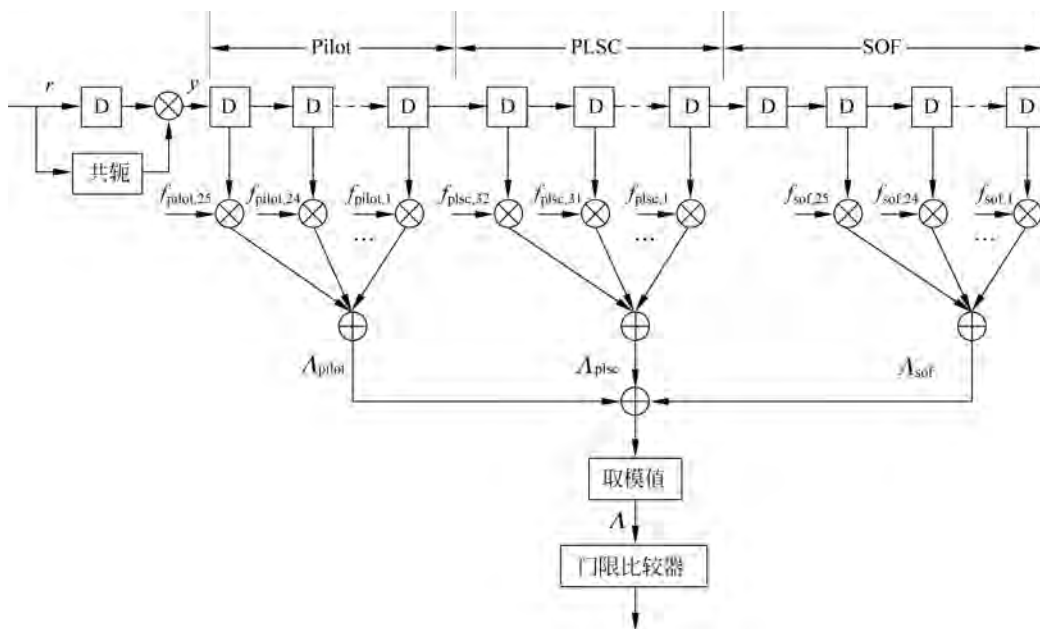


图 5-47 改进的联合 SOF、PLSC 和 Pilot 段的差分检测帧同步器

假设经过定时恢复后数据符号表示为

$$r_k = c_k e^{j(2\pi k \Delta f + \varphi_0)} + n_k$$

其中, c_k 是发送符号,且满足 $E\{|c_k|^2\} = 1$, Δf 和 φ_0 分别是频偏和相偏, n_k 是加性高斯白噪声。信号进入帧同步检测器,进行差分操作后可被表示为

$$y_k = r_k r_k^* = c_k c_k^* e^{j(-2\pi \Delta f)} + n_k'$$

其中, n_k' 为噪声项。从图 5-47 中可以看出,移位寄存器包括 3 部分: SOF 段、PLSC 段和 Pilot 段。差分操作之后的信号 y_k 被存储在移位寄存器中,与相应的系数分别相乘,然后整体求和,将其与设定的门限进行比较,从而判断是否检测到帧起始位置。

帧头的 SOF 段数据为 $s_{\text{SOF},k} = 182\text{D}2\text{E}82_{\text{HEX}}, k = 1, 2, \dots, 26$, 经过 $\pi/2$ -BPSK 调制之后的信号为

$$c_k = e^{j\theta_k}, \quad \theta_k = \begin{cases} \pi/4, & \text{mod}(k, 2) = 1 \text{ 且 } s_{\text{SOF},k} = 0 \\ 5\pi/4, & \text{mod}(k, 2) = 1 \text{ 且 } s_{\text{SOF},k} = 1 \\ 3\pi/4, & \text{mod}(k, 2) = 0 \text{ 且 } s_{\text{SOF},k} = 0 \\ 7\pi/4, & \text{mod}(k, 2) = 0 \text{ 且 } s_{\text{SOF},k} = 1 \end{cases} \quad (5-29)$$

其中, mod 表示取余操作。因此, SOF 段所对应的 25 个系数可以通过下式进行计算,可得

$$f_{\text{SOF},k} = c_k^* c_{k+1} = e^{j(\theta_{k+1} - \theta_k)} \quad (5-30)$$

将式(5-29)代入式(5-30)可以得到 SOF 段对应的 25 个系数分别为

$$f_{\text{SOF},k} = \{-j, -j, -j, -j, j, j, j, j, -j, j, j, j, -j, j, j, -j, -j, j, -j, -j, j, -j, -j, j, -j, -j\}$$

SOF 段数据经过差分之后,分别与对应的 25 个系数相乘求和可以得到

$$\Lambda_{\text{SOF}} = \sum_{k=1}^{25} y_k f_{\text{SOF},k} = \sum_{k=1}^{25} r_k r_{k+1}^* f_{\text{SOF},k} = 25e^{-j2\pi\Delta f} + n''$$

其中, n'' 为噪声项。

PLSC 段的前 6 比特信息经过 RM 编码之后为 $c_i (i=1,2,\dots,32)$, 用伪随机序列 $m = (m_1, m_2, \dots, m_{64})^T$ 对其进行加扰,最后经过 $\pi/2$ -BPSK 调制之后的信号为

$$z_k = e^{j(\phi_k + \varphi_k)}$$

其中,

$$\phi_k = \begin{cases} 0, & m_k = 0 \\ \pi, & m_k = 1 \end{cases}$$

$$\varphi_k = \begin{cases} \pi/4, & \text{若 } k=2i-1 \text{ 且 } c_i=0 \\ 5\pi/4, & \text{若 } k=2i-1 \text{ 且 } c_i=1 \\ 3\pi/4, & \text{若 } k=2i \text{ 且 } c_i=0 \\ 7\pi/4, & \text{若 } k=2i \text{ 且 } c_i=1 \end{cases}$$

PLSC 段对应的 32 个系数为

$$f_{\text{plsc},k} = z_{2i-1}^* z_{2i} = e^{j(\pi/2 + \phi_{2i} - \phi_{2i-1})}$$

经过计算得到 32 个系数为

$$f_{\text{plsc},k} = \{-j, j, j, -j, -j, -j, j, -j, -j, j, j, j, j, j, -j, -j, -j, -j, j, j, -j, j, j, -j, j, j, -j, -j, -j, -j, -j, -j\}$$

PLSC 段差分后的数据 y_k 分别与上述 32 个系数相乘,最后求和可得

$$\Lambda_{\text{plsc}} = \sum_{i=1}^{32} y_k f_{\text{plsc},i} = \sum_{i=1}^{32} r_{2i-1} r_{2i}^* f_{\text{plsc},i} = 32e^{-j2\pi\Delta f} + n'''$$

其中, n''' 为噪声项。

Pilot 段数据是在第 I 象限的 QPSK 调制星座点 $s_k = e^{j\pi/4}$, 由于数据段和 Pilot 段需要加扰,经过加扰之后的 Pilot 段数据为

$$x_k = e^{j(\pi/4 + m_k \pi/2)}, \quad k=1,2,\dots,26$$

$$m_k = \{3, 1, 3, 0, 3, 2, 2, 2, 0, 0, 0, 2, 0, 2, 2, 0, 0, 1, 3, 3, 1, 0, 3, 1, 0, 1\}$$

其中, m_k 是一种 Gold 序列产生的伪随机序列。因此, Pilot 段所对应的 25 个系数可以通过下式进行计算:

$$f_{\text{pilot},k} = x_k^* x_{k+1} = e^{j(m_{k+1} - m_k) \pi/2}$$

计算得到 Pilot 段对应的 25 个系数分别为

$$f_{\text{pilot},k} = \{-1, -1, j, -j, -j, 1, 1, -1, 1, 1, -1, -1, -1, 1, -1, 1, j, -1, 1, -1, -j, -j, -1, -j, j\}$$

经过差分之后的 Pilot 段数据 y_k 分别与对应的系数 $f_{\text{pilot},k}$ 相乘求和可以得

$$\Lambda_{\text{pilot}} = \sum_{k=1}^{25} y_k f_{\text{pilot},k} = \sum_{k=1}^{25} r_k r_{k+1}^* f_{\text{pilot},k} = 25e^{-j2\pi\Delta f} + n''''$$

其中, n''' 为噪声项。

在如图 5-47 所示的结构下, SOF 段、PLSC 段和 Pilot 段分别与各自的系数相乘之后, 最终输出 Δ 可以表示为

$$\begin{aligned}\Delta &= |\Lambda_{\text{SOF}} + \Lambda_{\text{plsc}} + \Lambda_{\text{pilot}}| \\ &= |25e^{-j2\pi\Delta f} + 32e^{-j2\pi\Delta f} + 25e^{-j2\pi\Delta f} + \omega| \\ &= |82e^{-j2\pi\Delta f} + \omega|\end{aligned}$$

其中, ω 为噪声项。将此输出峰值 Δ 与设定的门限 λ 进行比较, 从而判断是否检测到帧起始位置, 可根据系统要求的检测概率或者虚警概率设定门限 λ 。

下面对改进差分帧同步检测器与 Sun 提出的差分检测帧同步器进行性能仿真与比较。设置归一化频偏为 0.2, 蒙特卡罗仿真 10^7 次。

在较高信噪比下, 改进算法与 Sun 提出的算法性能较接近, 因此图 5-48 给出了信噪比为 -1dB 时, 不同判决门限下两种算法的虚警概率和检测概率的概率密度函数。从图 5-48 中看出, 与 Sun 提出的差分检测帧同步器相比, 改进差分检测帧同步器的检测和虚警概率密度函数均值的间距拉大, 也意味着在相同的虚警概率下, 改进差分检测帧同步器的检测概率比 Sun 提出的差分检测帧同步器的检测概率更大。

图 5-49 给出了不同信噪比下, 改进差分检测帧同步器与 Sun 提出的差分检测帧同步器的工作特性曲线 (Receiver Operating Characteristic Curve, ROC)。从图 5-49 中可以看出, 在相同的虚警概率下, 改进的差分检测帧同步器的检测性能优于 Sun 提出的差分检测帧同步器, 同时这也验证了图 5-48 给出的结论。

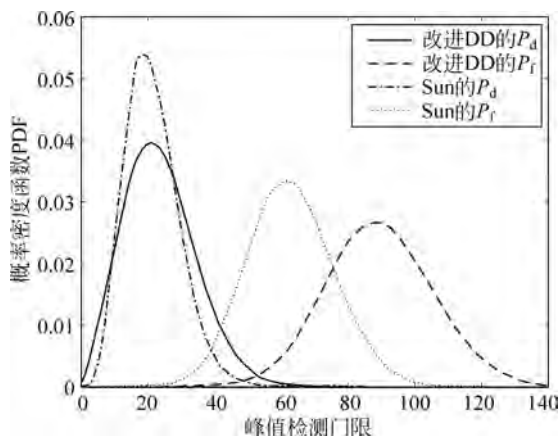


图 5-48 检测概率和虚警概率的概率密度函数,
SNR = -1dB

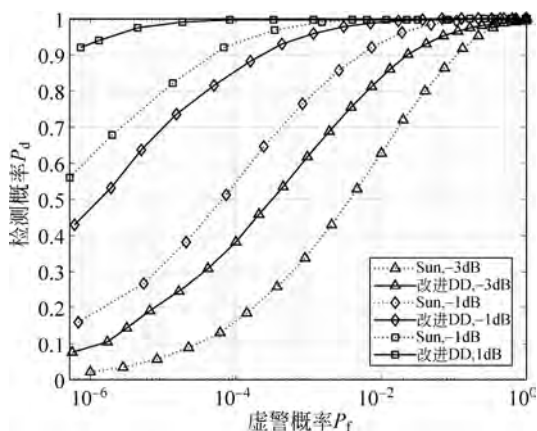


图 5-49 不同信噪比下的工作特性曲线

图 5-50 给出了在信噪比为 -1dB 时, 改进差分帧同步检测器与 Sun 提出的差分检测帧同步器的平均帧同步捕获时间。改进差分检测帧同步器缩短了平均帧同步捕获时间, 并且随着捕获校验次数 α 的增加, 比 Sun 提出的差分检测帧同步器所需的平均帧同步捕获时间更少。

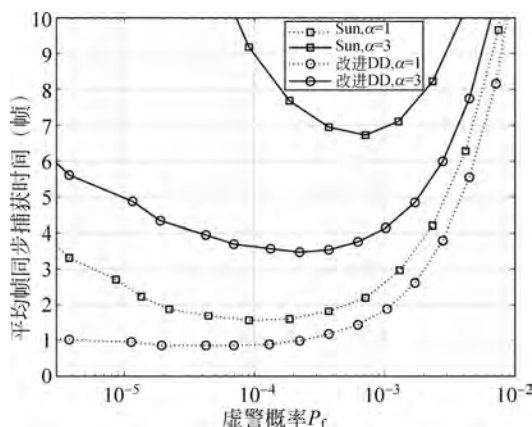


图 5-50 改进差分检测帧同步器与 Sun 提出的差分检测帧同步器的平均帧同步捕获时间, SNR = -1 dB

5.6.6 多重相关峰值检测帧同步方案

根据差分检测帧同步方案的分析可知, PLSC 字段能用于帧同步的实质是利用 PLSC 字段相邻的奇数位和偶数位进行差分相关后, 相关值在相位上与 b_6 保持着相对确定的关系, 并且满足相位一致条件。

对于 PLSC 字段采用的 RM(32, 6) 码, 除信息比特 b_5 外, b_0, b_1, b_2, b_3, b_4 均具有与 b_6 相同的性质, 如式 (5-31) 所示。

$$\begin{cases} w_0 \oplus w_1 = w_2 \oplus w_3 = \cdots = w_{30} \oplus w_{31} = b_0 \\ w_0 \oplus w_2 = w_1 \oplus w_3 = \cdots = w_{29} \oplus w_{31} = b_1 \\ w_0 \oplus w_4 = w_1 \oplus w_5 = \cdots = w_{27} \oplus w_{31} = b_2 \\ w_0 \oplus w_8 = w_1 \oplus w_9 = \cdots = w_{23} \oplus w_{31} = b_3 \\ w_0 \oplus w_{16} = w_1 \oplus w_{17} = \cdots = w_{15} \oplus w_{31} = b_4 \end{cases} \quad (5-31)$$

令使得编码比特 $c_{p_i^k} \oplus c_{q_i^k} = b_k$ 的编码比特序号对 (p_i^k, q_i^k) 构成的编码比特序列号对序列为 $T_{b_k} = \{(p_0^k, q_0^k), (p_1^k, q_1^k), \cdots, (p_{31}^k, q_{31}^k)\}$, 其中 $k = 0, 1, 2, 3, 4, 6$ 。并且 b_0, b_1, b_2, b_3, b_4 对应的的相关值分别为:

$$\begin{cases} \Lambda_{b_0} = 32e^{j b_0 \pi} e^{-j 2 \pi \Delta f \times 2} \\ \Lambda_{b_1} = 32e^{j b_1 \pi} e^{-j 2 \pi \Delta f \times 4} \\ \Lambda_{b_2} = 32e^{j b_2 \pi} e^{-j 2 \pi \Delta f \times 8} \\ \Lambda_{b_3} = 32e^{j b_3 \pi} e^{-j 2 \pi \Delta f \times 16} \\ \Lambda_{b_4} = 32e^{j b_4 \pi} e^{-j 2 \pi \Delta f \times 32} \end{cases}$$

多重相关峰值检测方法将 $b_0, b_1, b_2, b_3, b_4, b_6$ 以及 SOF 字段对应的的相关值, 先取绝对值再求和后, 送入峰值检测模块, 如式 (5-32) 所示。

$$\Lambda_M = |\Lambda_{\text{SOF}}| + |\Lambda_{b_6}| + \sum_{k=0}^4 |\Lambda_{b_k}| \quad (5-32)$$

5.6.7 基于 RM 码部分译码辅助差分检测帧同步

基于部分译码辅助差分检测帧同步方案的主要步骤概括如下：

步骤 1, 根据 SOF 字段数据进行差分相关得到的相关值。

$$\Delta_{\text{SOF}}(\mu) = \sum_{i=1}^{25} r_{u+i-1} r_{u+i}^* d_{i-1}^{\text{SOF}}$$

式中, μ 为需要判断的帧起始位置。

步骤 2, 利用 PLSC 字段的数据, 分别根据信息位 b_k 对应的编码比特序号对序列, 抽取数据进行相关运算得到的相关值。

$$\Delta_{b_k}(\mu) = \sum_{i=0}^{31} \tilde{r}_{\mu+p_i^k} \tilde{r}_{\mu+q_i^k}^*$$

步骤 3, 无频偏情况和频偏存在时分别根据式(5-33)和式(5-34)计算 b_k 对应的软信息 $R_i^k(\mu)$ 。

$$R_i^k(\mu) = 2 \tanh^{-1} \left(\tanh \left(\frac{\text{LLR}_{\mu+p_i^k}}{2} \right) \cdot \tanh \left(\frac{\text{LLR}_{\mu+q_i^k}}{2} \right) \right) \quad (5-33)$$

$$R_i^k(\mu) = 4 (\tilde{r}_{\mu+p_i^k}^1 \cdot \tilde{r}_{\mu+q_i^k}^1 + \tilde{r}_{\mu+p_i^k}^Q \cdot \tilde{r}_{\mu+q_i^k}^Q) \quad (5-34)$$

步骤 4, 根据 T_{b_k} 中所有编码比特序号对计算得到的中间信息 $R_i^k(\mu)$ 求和, 输出 b_k 的似然信息 $L_k(\mu)$ 为:

$$L_k(\mu) = \sum_{i=0}^{31} R_i^k(\mu)$$

步骤 5, 将 SOF 字段和 PLSC 字段的数据, 进行部分译码辅助的差分相关运算得到的相关值如式(5-35)所示。

$$\Delta(\mu) = \Delta_{\text{SOF}}(\mu) + \text{sgn}(L_6(\mu)) \cdot \Delta_{b_6}(\mu) + \sum_{k=0}^4 \text{sgn}(L_k(\mu)) \cdot \Delta_{b_k}(\mu) \quad (5-35)$$

步骤 6, 根据 $\Delta(\mu)$ 判断帧起始位置 $\hat{\mu}$:

$$\hat{\mu} = \underset{\mu}{\text{argmax}} \{ \Delta(\mu) \}$$

该方案同时利用 PL Header 的 SOF 字段和 PLSC 字段的数据实现帧同步, 其结构框图如图 5-51 所示。对于 SOF 字段, 将相邻的数据进行共轭相乘后, 与对应的 25 个 SOF 系数相乘并求和。而 PLSC 字段, 首先, 根据 $b_0, b_1, b_2, b_3, b_4, b_6$ 对应的编码位序号对序列, 抽取数据分别进行共轭相乘和中间信息计算; 其次, 将中间信息和共轭相乘的结果分别求和得到似然信息和部分相关值; 再次, 根据似然信息的符号调整差分相关得到相关值的相位; 最后, 将根据 SOF 字段和 PLSC 字段求得的所有相关值求和, 送入峰值检测模块。

本节将根据 Reed-Muller 码的编码比特两两异或为同一信息位, 并且两位间的间隔相同的编码特性, 对 $b_0, b_1, b_2, b_3, b_4, b_6$ 进行部分译码, 分别给出无频偏情况和有频偏情况下的基于部分译码的差分检测帧同步过程。此外, 考虑到在部分译码时, 先得对接收数据解扰才能计算信息位的似然信息, 而传统的差分相关运算将接收数据共轭相乘后再与本地系数相乘, 本地系数为扰码位对应的符号共轭相乘的结果, 为了减少运算可先对接收数据解扰后再共轭相乘。设接收符号 r_t 解扰后表示为 $\tilde{r}_t = r_t \times e^{j\pi s_t}$, s_t 为第 t 个扰码位。

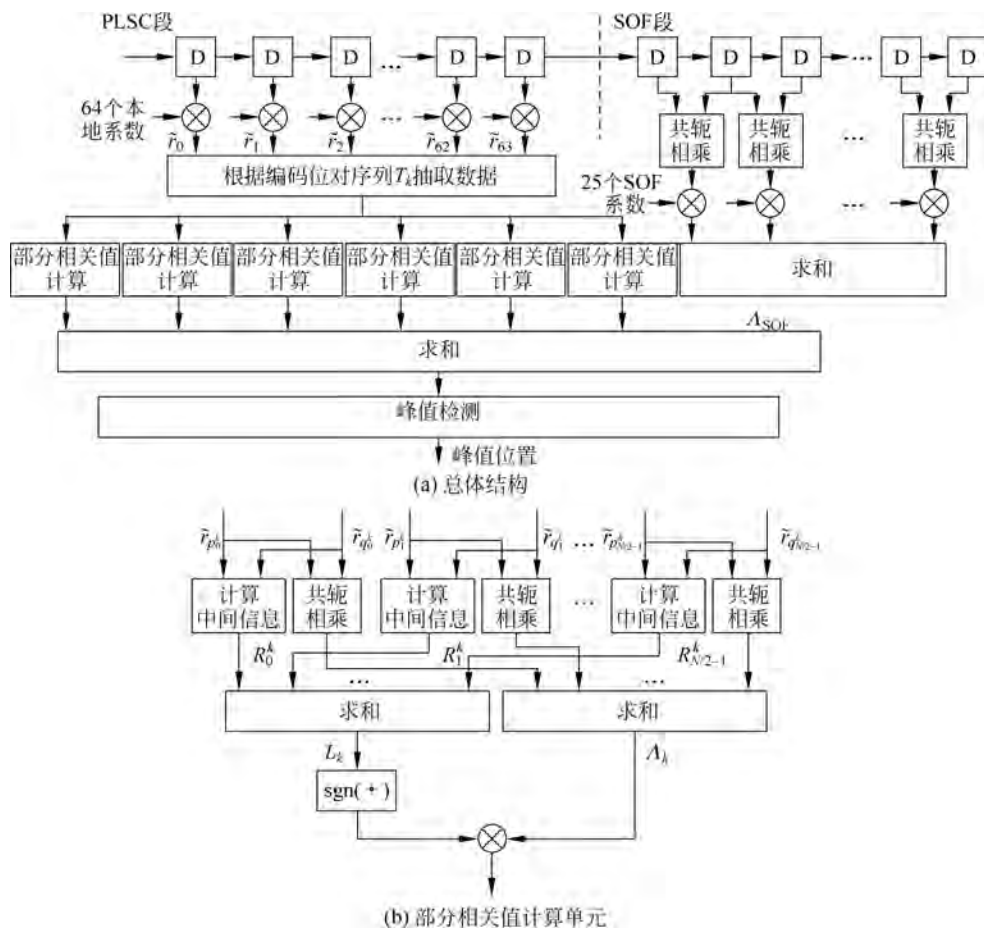


图 5-51 基于部分译码辅助差分检测帧同步结构框图

1. 无频偏情况下的帧同步过程

在无频偏情况下,直接通过计算接收数据的对数似然比对 RM(64,7)码进行部分译码,然后根据译码的结果消除相关值中信息位 b_k 的影响。基于部分译码辅助差分检测帧同步主要分为 4 个步骤。

步骤 1,计算接收数据的对数似然比,采用 $\pi/2$ -BPSK 映射时,接收数据对应的对数似然比为

$$\begin{cases} \text{LLR}_{2j} = \frac{\sqrt{2}(\tilde{r}_{2j}^Q + \tilde{r}_{2j}^I)}{\sigma^2} \\ \text{LLR}_{2j+1} = \frac{\sqrt{2}(\tilde{r}_{2j+1}^Q - \tilde{r}_{2j+1}^I)}{\sigma^2} \end{cases} \quad (5-36)$$

式中, LLR_{2j} 和 LLR_{2j+1} 分别表示为奇数位和偶数位数据对应的对数似然比, $\tilde{r}_{2j}^Q$ 和 $\tilde{r}_{2j}^I$ 分别为符号 $\tilde{r}_{2j}$ 的虚部和实部, $\tilde{r}_{2j+1}^Q$ 和 $\tilde{r}_{2j+1}^I$ 分别为符号 $\tilde{r}_{2j+1}$ 的虚部和实部, σ^2 为 AWGN 的噪声方差。

步骤 2,根据 b_k 对应的编码位序号对序列 T_{b_k} 的编码位序号对 (p_i^k, q_i^k) ,选取符号 $\tilde{r}_{p_i^k}$

和 $\tilde{r}_{q_i^k}$ 的对数似然比 $\text{LLR}_{p_i^k}$ 和 $\text{LLR}_{q_i^k}$, 并根据式(5-37)计算 b_k 的中间信息 R_i^k 。

$$R_i^k = 2 \tanh^{-1} \left(\tanh \left(\frac{\text{LLR}_{p_i^k}}{2} \right) \cdot \tanh \left(\frac{\text{LLR}_{q_i^k}}{2} \right) \right) \quad (5-37)$$

式中, $\tanh(x) = \frac{e^{2x} - 1}{e^{2x} + 1}$, $\tanh^{-1}(x) = 0.5 \ln \left(\frac{1+x}{1-x} \right)$ 。

步骤 3, 根据中间信息 R_i^k 求和计算 b_k 对应的似然信息 L_k , 具体为:

$$L_k = \sum_{i=0}^{31} R_i^k \quad (5-38)$$

步骤 4, 计算得到信息位 $b_0, b_1, b_2, b_3, b_4, b_6$ 对应的似然信息后, 根据式(5-39)将 $b_0, b_1, b_2, b_3, b_4, b_6$ 和 SOF 字段对应的相关值求和。

$$\Lambda = \Lambda_{\text{SOF}} + \text{sgn}(L_6) \cdot \Lambda_{b_6} + \sum_{k=0}^4 \text{sgn}(L_k) \cdot \Lambda_{b_k} \quad (5-39)$$

步骤 2 计算中间信息 R_i^k 除了通过式(5-37)计算外, 还可通过最小和算法、偏移最小和算法以及查表法实现, 具体实现分别如式(5-40)、式(5-41)和式(5-42)所示。

$$R_i^k = \text{sgn}(\text{LLR}_{p_i^k}) \cdot \text{sgn}(\text{LLR}_{q_i^k}) \cdot \min(|\text{LLR}_{p_i^k}|, |\text{LLR}_{q_i^k}|) \quad (5-40)$$

$$R_i^k = -\text{sgn}(\text{LLR}_{p_i^k}) \cdot \text{sgn}(\text{LLR}_{q_i^k}) \cdot \max(\min(|\text{LLR}_{p_i^k}|, |\text{LLR}_{q_i^k}|) - \beta, 0) \quad (5-41)$$

$$R_i^k = \text{sgn}(\text{LLR}_{p_i^k}) \cdot \text{sgn}(\text{LLR}_{q_i^k}) \cdot \min(|\text{LLR}_{p_i^k}|, |\text{LLR}_{q_i^k}|) + \text{LUT}(\text{LLR}_{p_i^k}, \text{LLR}_{q_i^k}) \quad (5-42)$$

式(5-41)中, β 为偏移因子, 可通过密度演化和计算机仿真方法得到, 式(5-42)中 $\text{LUT}(a, b) = \log(1 + e^{-|a+b|}) - \log(1 + e^{-|a-b|})$, 可通过查找表实现。

2. 有频偏情况下的帧同步过程

在有频偏情况下, 由于载波频偏影响, 接收信号与参考的星座点之间存在一个相位偏移, 因而无法准确计算接收信号的对数似然比。然而, 当载波频率偏移固定或随机抖动较小时, 根据 T_{b_k} 的编码位序号对 (p_i^k, q_i^k) , 抽取的符号 $\tilde{r}_{p_i^k}$ 和 $\tilde{r}_{q_i^k}$ 之间的相位差, 与信息位 b_k 的取值和 $p_i^k - q_i^k$ 有关。此外, 两符号间的距离可直接反映符号间相位差的大小, 因此根据符号 $\tilde{r}_{p_i^k}$ 和 $\tilde{r}_{q_i^k}$ 之间的距离, 在有频偏的情况下可对 RM(64, 7) 码进行部分译码。

当 $k=0, 1, 2, 3, 4$ 时, 根据 T_{b_k} 的编码位序号对 (p_i^k, q_i^k) , 抽取的符号 $\tilde{r}_{p_i^k}$ 和 $\tilde{r}_{q_i^k}$ 进行共轭相乘后的相位差为 $b_k \pi - 2\pi \Delta f (p_i^k - q_i^k)$, 并且 $p_i^k - q_i^k$ 为常数 η_k , 如式(5-43)所示。

$$\begin{aligned} \tilde{r}_{p_i^k} \cdot \tilde{r}_{q_i^k}^* &= e^{j b_k \pi} e^{-j(2\pi \Delta f \times p_i^k + \theta)} e^{j(2\pi \Delta f \times q_i^k + \theta)} \\ &= e^{j b_k \pi} e^{-j 2\pi \Delta f (p_i^k - q_i^k)} \\ &= e^{j(b_k \pi - 2\pi \Delta f \eta_k)} \end{aligned} \quad (5-43)$$

因此, 当 $-2\pi \Delta f \eta_k < \frac{\pi}{2}$ 时, 若符号 $\tilde{r}_{p_i^k}$ 和 $\tilde{r}_{q_i^k}$ 的距离 D_i^k 与符号 $\tilde{r}_{p_i^k}$ 和 $-\tilde{r}_{q_i^k}$ 的距离 $\hat{D}_i^k$ 满足 $\hat{D}_i^k - D_i^k > 0$, 则 b_k 判决为 0, 否则判决为 1, 如图 5-52 中(a)和(b)所示。当 $-2\pi \Delta f \eta_k > \frac{\pi}{2}$, 若 $\hat{D}_i^k - D_i^k < 0$, 则 b_k 判决为 1, 否则判决为 0, 如图 5-53 中(a)和(b)所示。其中距离 D_i^k

和 $\hat{D}_i^k$ 分别如下所示。

$$D_i^k = (\tilde{r}_{p_i^k}^I - \tilde{r}_{q_i^k}^I)^2 + (\tilde{r}_{p_i^k}^Q - \tilde{r}_{q_i^k}^Q)^2$$

$$\hat{D}_i^k = (\tilde{r}_{p_i^k}^I + \tilde{r}_{q_i^k}^I)^2 + (\tilde{r}_{p_i^k}^Q + \tilde{r}_{q_i^k}^Q)^2$$

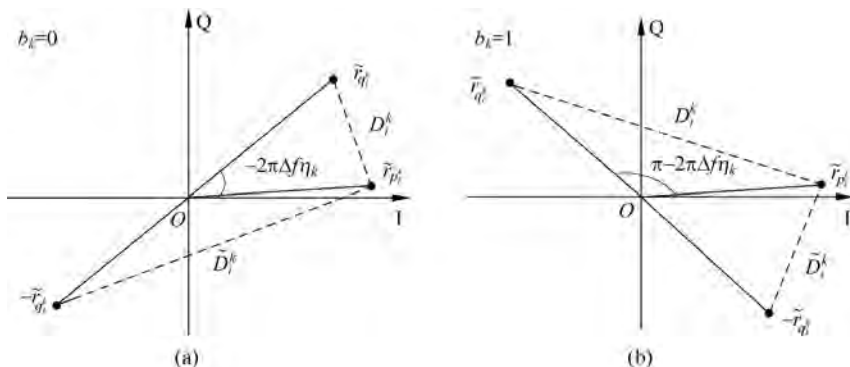


图 5-52 $-2\pi\Delta f\eta_k < \frac{\pi}{2}$ 的情况, 根据 $\tilde{r}_{p_i^k}$ 和 $\tilde{r}_{q_i^k}$ 的距离 D_i^k 与 $\tilde{r}_{p_i^k}$ 和 $-\tilde{r}_{q_i^k}$ 的距离 $\hat{D}_i^k$ 对 b_k 译码

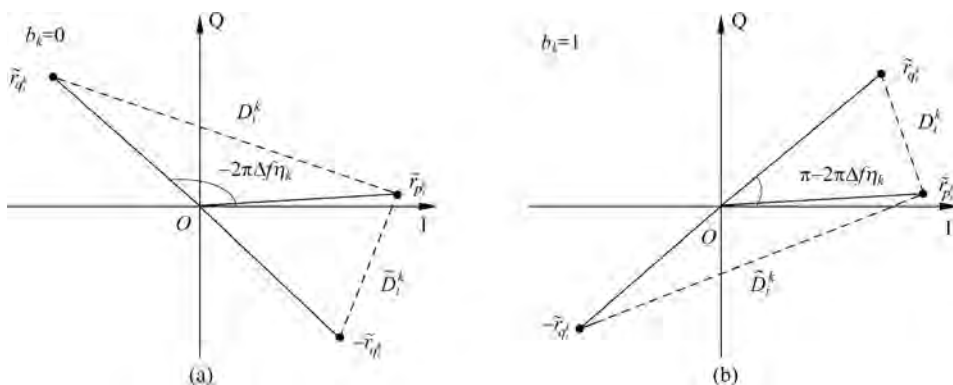


图 5-53 $-2\pi\Delta f\eta_k > \frac{\pi}{2}$ 的情况, 根据 $\tilde{r}_{p_i^k}$ 和 $\tilde{r}_{q_i^k}$ 的距离 D_i^k 与 $\tilde{r}_{p_i^k}$ 和 $-\tilde{r}_{q_i^k}$ 的距离 $\hat{D}_i^k$ 对 b_k 译码

当 $k=6$ 时, 由于采用 $\pi/2$ -BPSK 映射, 奇数位和偶数位的接收符号本身存在 $\pi/2$ 的相位差, 根据 T_{b_6} 的编码位序号对 (p_i^6, q_i^6) , 抽取的符号 $\tilde{r}_{p_i^6}$ 和 $\tilde{r}_{q_i^6}$ 进行共轭相乘后的相位差为 $-\frac{\pi}{2} + b_6\pi - 2\pi\Delta f\eta_6$ 。将 $\tilde{r}_{q_i^6}$ 顺时针旋转 $\frac{\pi}{2}$ 相角后, 使得符号 $\tilde{r}_{p_i^6}$ 和 $\tilde{r}_{q_i^6}e^{j\frac{\pi}{2}}$ 进行共轭相乘后的相位差为 $b_6\pi - 2\pi\Delta f\eta_6$, 同样可根据符号 $\tilde{r}_{p_i^6}$ 和 $\tilde{r}_{q_i^6}e^{-j\frac{\pi}{2}}$ 之间的距离 D_i^6 与符号 $\tilde{r}_{p_i^6}$ 和 $-\tilde{r}_{q_i^6}e^{-j\frac{\pi}{2}}$ 的距离 $\hat{D}_i^6$ 对 b_6 进行译码。

考虑 $-2\pi\Delta f\eta_k < \frac{\pi}{2}$ 情况, 根据 T_{b_k} 的编码位序号对 (p_i^k, q_i^k) 计算 D_i^k 和 $\hat{D}_i^k$, 并根据式(5-44)计算 b_k 的中间信息 R_i^k :

$$\begin{aligned}
 R_i^k &= \hat{D}_i^k - D_i^k \\
 &= (\tilde{r}_{p_i^k}^I + \tilde{r}_{q_i^k}^I)^2 + (\tilde{r}_{p_i^k}^Q + \tilde{r}_{q_i^k}^Q)^2 - (\tilde{r}_{p_i^k}^I - \tilde{r}_{q_i^k}^I)^2 + (\tilde{r}_{p_i^k}^Q - \tilde{r}_{q_i^k}^Q)^2 \\
 &= 4(\tilde{r}_{p_i^k}^I \cdot \tilde{r}_{q_i^k}^I + \tilde{r}_{p_i^k}^Q \cdot \tilde{r}_{q_i^k}^Q)
 \end{aligned} \quad (5-44)$$

根据 T_{b_k} 中所有编码位序号对计算 b_k 的似然信息 L_k 。然后,根据式(5-39)对 $b_0, b_1, b_2, b_3, b_4, b_6$ 以及 SOF 字段对应的相关值进行求和。

3. 仿真结果

基于 RM 码辅助的检测结构实际上是针对 DVB-S2 系统的特殊帧头产生方式提出的,此方法充分利用了帧头 PLSC 码的相关性,在原来的差分检测结构基础上进行了结构加强。下面通过仿真对其性能进行分析。

1) 频率敏感度的仿真

图 5-54 是基于 RM 码辅助的检测结构在归一化频率误差为 $\Delta f T = 0.2$ 和相位误差等于 $\theta = 10^\circ$ 时的 ROC 曲线。由图 5-54 可见,这种检测结构基本不受频偏和相差的影响。

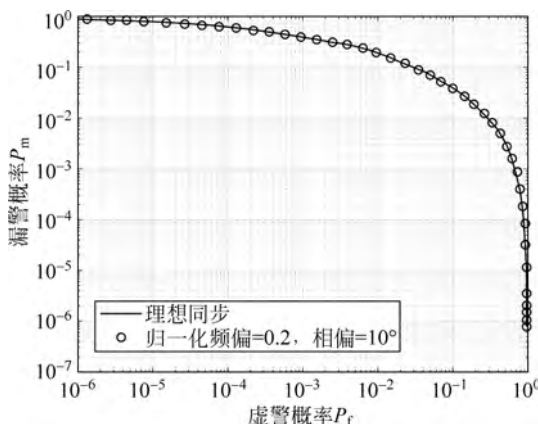


图 5-54 基于 RM 码辅助的检测结构的 ROC 曲线, SNR = -2dB

2) 平均同步捕获时间仿真

为了比较加入不同码组时 RM 码辅助检测结构的性能差异,通过蒙特卡罗仿真对性能进行分析比较。由图 5-55 可见,随着加入码组结构数量的增加,同步性能比原结构的性能

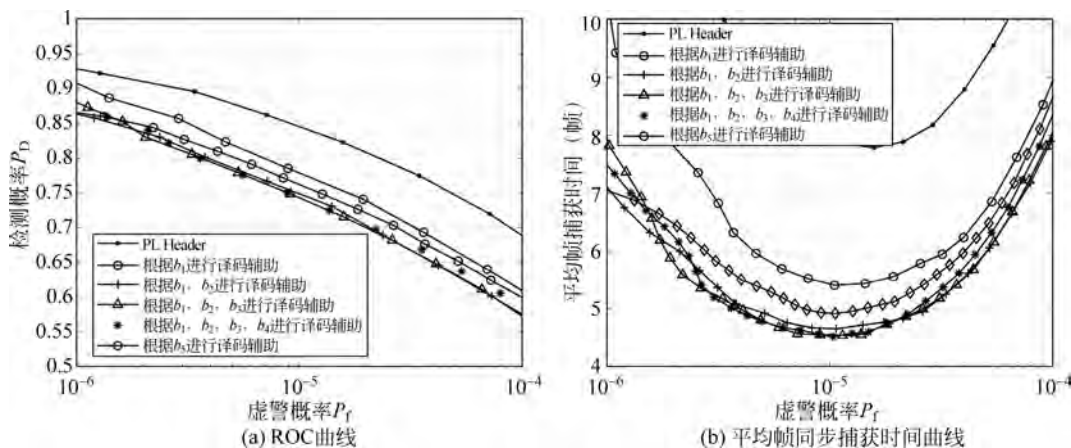


图 5-55 基于 RM 码辅助的检测结构的 ROC 曲线和平均帧同步捕获时间曲线, SNR = -2dB

有所提升,同一虚警概率下检测概率最大可提高 10%,最小的平均帧同步捕获时间(当 $P_f = 10^{-5}$ 时),由原来的 8 帧降为 5 帧,比原来的平均捕获时间缩短了 37.5%。

基于 RM 码辅助的帧检测结构比差分检测结构性能有所提升,但由于其对 PLSC 进行了多次系数相关及复乘运算,使得复杂度大大增加,在实际中使用较少。

5.6.8 3 种方案仿真结果

本节根据帧同步错误概率(Frame Synchronization Error Rate,FSER),比较基于 RM 码部分译码辅助差分检测帧同步方案、差分检测帧同步方案以及多重相关峰值检测帧同步方案的帧同步性能。

1. 无频偏情况

在 AWGN 信道下,仿真比较了差分检测帧同步方案,仿真结果如图 5-56 所示。由图可知,基于 RM 码部分译码辅助差分检测帧同步方案与多重相关峰值检测和差分检测帧同步方案相比较,帧同步性能分别有 0.6dB 和 3.3dB 的性能增益。

2. 有频偏情况

首先,在 AWGN 信道下,归一化频偏为 0.01 时,仿真了基于 RM 部分译码辅助差分检测帧同步方案的性能,仿真结果如图 5-57 所示。可知,当使用 b_0, b_1, b_2, b_6 对应的编码比特序号对序列进行译码辅助的差分检测帧同步时,帧同步性能最优。将 b_3, b_4 对应的编码比特序号对序列用于帧同步时,性能恶化,主要由于根据 b_3, b_4 对应的编码比特序号对序列抽取数据的间隔过大,导致根据符号间的距离对 b_3, b_4 进行译码的可靠性降低。

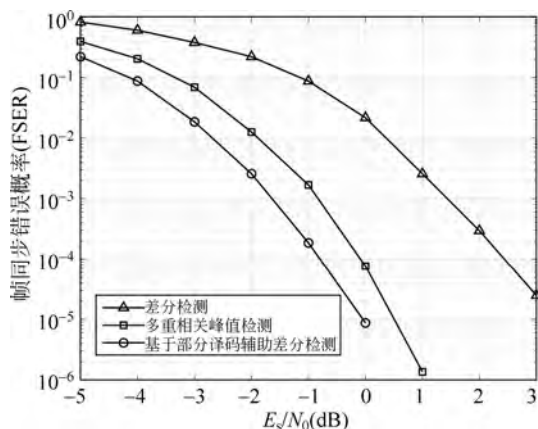


图 5-56 基于部分译码的差分检测帧同步方案在无频偏情况下的性能

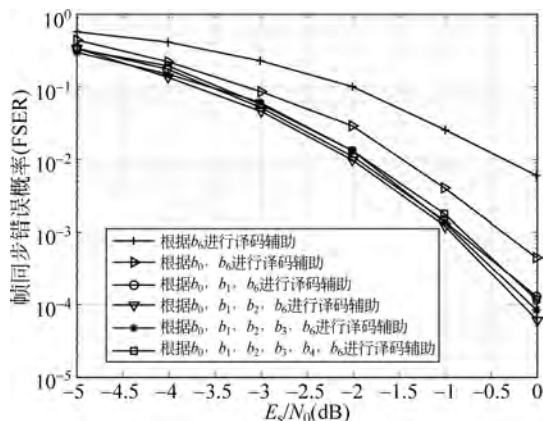


图 5-57 归一化频偏为 0.01 时,基于部分译码的差分检测帧同步方案性能

然后,在归一化频偏为 0.01 时,仿真比较了差分检测帧同步方案、多重相关峰值检测与根据符号间的距离对 b_0, b_1, b_2, b_6 进行译码辅助的差分检测帧同步方案的性能,仿真结果如图 5-58 所示。可知,基于 b_0, b_1, b_2, b_6 译码辅助差分检测帧同步方案,与差分检测帧同步方案和基于 b_0, b_1, b_2, b_6 多重相关峰值检测相比,性能得到了改善。此外,基于 b_0, b_1, b_2, b_6 译码的差分检测帧同步方案的性能与基于 $b_0, b_1, b_2, b_3, b_4, b_6$ 多重相关峰值检测性能相比,在低信噪比下帧同步性能依然得到改善,但是在信噪比大于 0.5dB 时,帧同步性能反而变差。

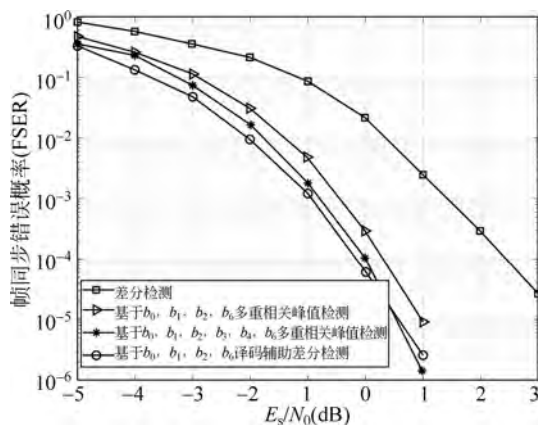


图 5-58 译码辅助差分检测帧同步方案与已有方案的帧同步性能比较

最后,为了验证基于部分译码辅助差分检测帧同步方案工作的频偏范围,在信噪比为 0dB 时,仿真了不同归一化频偏下的基于部分译码辅助差分检测帧同步和多重相关峰值检测的帧同步性能,两种方案分别基于 $b_6, b_0, b_6, b_0, b_1, b_6$ 和 b_0, b_1, b_2, b_6 的编码作用以及 SOF 字段进行帧同步,仿真结果如图 5-59 所示。可知,根据基于 $b_6, b_0, b_6, b_0, b_1, b_6$ 和 b_0, b_1, b_2, b_6 的译码辅助差分检测帧同步方案的帧同步性能优于多重相关峰值检测时的归一化频偏范围分别为 $[-0.08, 0.08]$ 、 $[-0.05, 0.05]$ 、 $[-0.03, 0.03]$ 和 $[-0.02, 0.02]$ 。

3. 复杂度分析

本节针对差分检测帧同步方案、多重相关峰值检测方法和基于 RM 码部分译码辅助的差分检测帧同步方案的复杂度进行分析。3 种帧同步方案都利用 SOF 字段和 PLSC 字段进行帧同步,对于 SOF 字段,均采用差分相关方法计算相关值。然而对于 PLSC 字段,虽然 3 种方案都利用 PLSC 字段采用的 RM 码的特殊编码结构进行帧同步,但处理方法不同,因此通过计算 RM 码用于帧同步时,进行一次相关值计算的运算量进行复杂度分析和对比,令 RM 码的码字长度为 N 。

差分检测帧同步方案仅抽取 PLSC 字段的奇数位与偶数位共轭相乘后与相关系数相乘完成相关运算,则其中共轭相乘的乘法次数为 $2N$ (1 次共轭相乘等价于 4 次实数相乘),与相关系数相乘的乘法运算次数为 $0.5N$,总的乘法运算次数则为 $2.5N$,相关运算单元还需进行 $0.5N - 1$ 次加法完成一次相关值的计算。

多重相关峰值检测方法在差分检测帧同步方案的基础上增加了其他的编码比特序号对序列抽取数据进行帧同步,假设 RM 码对应的编码比特序号对序列的数目为 L ,其中包括差分检测帧同步方案采用的编码比特序号对序列,即抽取奇数位和偶数位的数据进行共轭相乘。因此,多重相关峰值检测方法进行相关运算时的乘法运算次数为 $2.5N \cdot L$,加法运算次数为 $(0.5N - 1) \cdot L$ 。此外,多重相关峰值检测方法的峰值合并将不同编码比特序号对序列对应的相关值取绝对值后再求和,需 L 次取绝对值运算和 $L - 1$ 次加法运算。因此,取绝对值运算次数为 L 。

基于 RM 码部分译码辅助的差分检测帧同步方法在多重相关峰值检测方法的基础上,利用对 RM 码进行部分译码的结果消去相关值中物理层信令的不确定性,并且在进行相关运算之前,将本地系数与接收数据相乘实现解扰,此时需要 N 次乘法运算。分别对无频偏

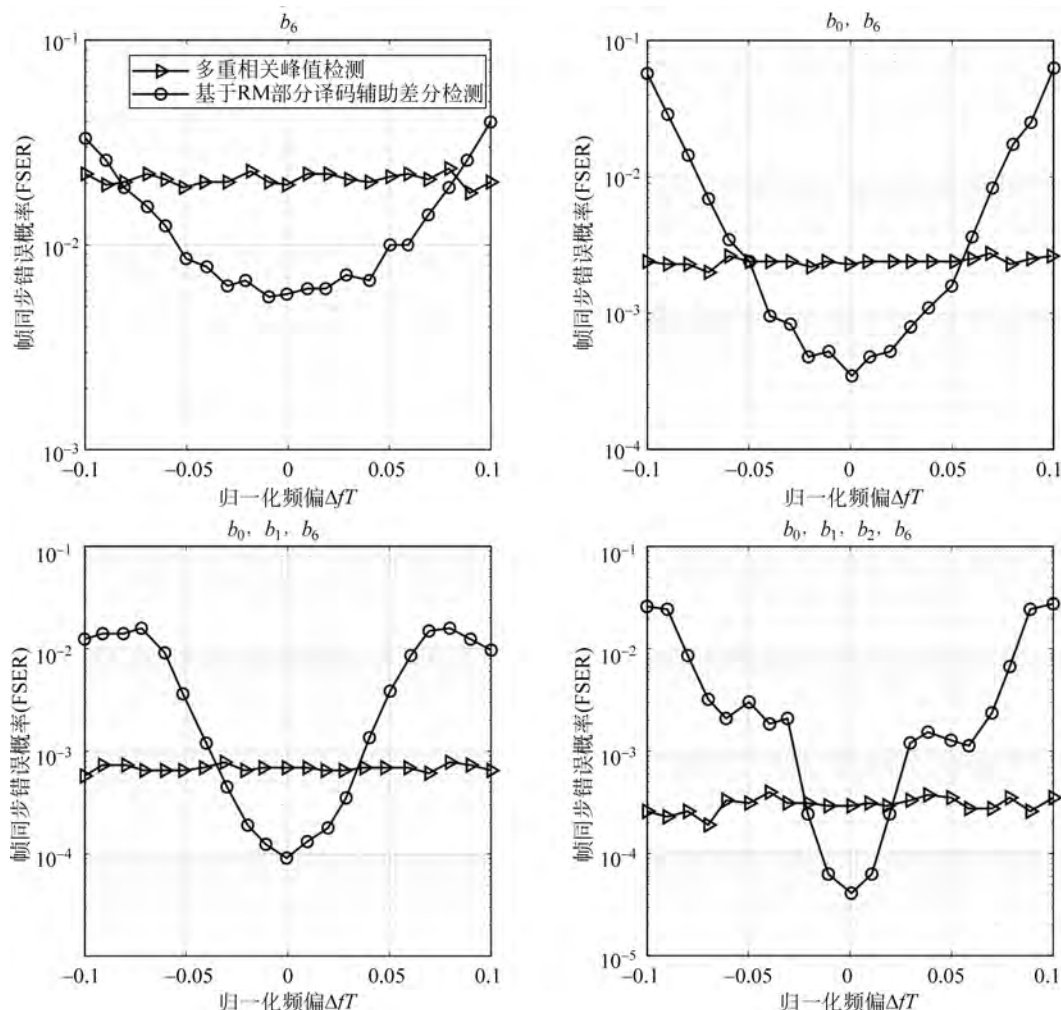


图 5-59 SNR=0dB 时,不同归一化频偏下的帧同步性能

情况下和有频偏情况下采用基于 RM 码部分译码辅助的差分检测帧同步方法实现帧同步的复杂度进行分析。

在无频偏情况下,步骤 1 对数似然比计算,如式(5-36)所示,乘法运算次数为 $2N \cdot L$,加法运算次数为 $N \cdot L$;步骤 2 中间信息计算,如式(5-37)所示,乘法运算次数为 $2N \cdot L$,加法运算次数为 $3N \cdot L$,指数运算次数为 $2N \cdot L$,对数运算次数为 $0.5N \cdot L$;步骤 3 中似然信息计算,如式(5-38)所示,进行了 $(0.5N-1) \cdot L$ 次加法;步骤 4 中若只合并 PLSC 部分的相关值,如式(5-39)所示,则需 L 次取符号运算和乘法运算,以及 $L-1$ 次加法运算。此外,根据编码比特序列计算相关值时的运算复杂度与多重相关峰值检测方法相同,但不必与相关系数相乘,因而乘法运算次数为 $2N \cdot L$,加法运算次数为 $0.5N \cdot L-1$ 。因此,基于 RM 码部分译码辅助的差分检测帧同步方法的乘法运算次数为 $6N \cdot L+L+N$,加法运算次数为 $5N \cdot L-2$,指数运算次数为 $2N \cdot L$,对数运算次数为 $1.5N \cdot L$,取符号运算次数为 L 。

在有频偏情况下,直接根据接收数据计算中间信息,如式所示,乘法运算为 $1.5N \cdot L$,

加法运算为 $0.5N \cdot L$, 后续部分处理与无频偏情况相同, 则基于 RM 码部分译码辅助的差分检测帧同步方法的乘法运算次数为 $3.5N \cdot L + L + N$, 加法运算次数为 $1.5N \cdot L - 2$, 取符号运算为 L 。

综上所述, 表 5-5 给出了 3 种帧同步方法的复杂度对比。为了使 3 种帧同步方法的复杂度对比更加直观, 表 5-6 给出了 DVB-S2 中 PLSC 字段采用 RM(64,7) 码时的复杂度对比, 此时 $N=64, L=6$ 。从表 5-6 可知, 基于 RM 码部分译码辅助的差分检测帧同步方法的复杂度, 相对于差分检测帧同步方法和多重相关峰值检测方法有所提高。

表 5-5 3 种帧同步方法的复杂度对比

运算类型	差分检测	多重相关峰值检测	基于 RM 码部分译码辅助(无频偏)	基于 RM 码部分译码辅助(有频偏)
加法运算(次)	$0.5N - 1$	$0.5N \cdot L - 1$	$5N \cdot L - 2$	$1.5N \cdot L - 2$
乘法运算(次)	$2.5N$	$2.5N \cdot L$	$6N \cdot L + L + N$	$3.5N \cdot L + L + N$
取绝对值运算(次)	0	L	0	0
取符号运算(次)	0	0	L	L
指数运算(次)	0	0	$2N \cdot L$	0
对数运算(次)	0	0	$0.5N \cdot L$	0

表 5-6 基于 RM(64,7) 码进行帧同步时的 3 种帧同步方法的复杂度对比

运算类型	差分检测	多重相关峰值检测	基于 RM 码部分译码辅助(无频偏)	基于 RM 码部分译码辅助(有频偏)
加法运算(次)	31	191	1918	574
乘法运算(次)	160	960	2374	1414
取绝对值运算(次)	0	6	0	0
取符号运算(次)	0	0	6	6
指数运算(次)	0	0	768	0
对数运算(次)	0	0	192	0

本章首先分析了差分检测帧同步方案、多重相关峰值检测帧同步方案的特点, 并对根据编码比特序号对序列抽取 PLSC 字段数据进行相关运算的相关特性进行了分析; 然后, 在多重相关峰值检测帧同步方案的基础上, 提出了基于 RM 码部分译码辅助差分检测帧同步方案, 通过对 RM 码进行部分译码, 利用译码的结果消除相关峰值中信息比特的影响, 改善帧同步性能; 最后, 通过仿真分析了 3 种帧同步方案的性能。仿真结果表明, 基于 RM 码部分译码辅助差分检测帧同步方案相较于差分检测帧同步方案和多重相关峰值检测帧同步方案在性能上均有很大的改善。

参考文献

[1] Shark L K, Terrell T J, Simpson R J. Adaptive frame synchronizer for digital satellite communication systems[J]. Radar & Signal Processing Iee Proceedings F, 1988, 135(1): 51-59.

[2] 苏向东. 高速帧同步格式化器[D]. 成都: 电子科技大学, 2003.

[3] 李玮. 基于 CCSDS 标准的帧同步算法研究及其 FPGA 实现[D]. 成都: 西南交通大学, 2009.

- [4] 夏俊. 帧同步电路的设计和分析[J]. 电讯技术, 2006, 2: 155-158.
- [5] 樊昌信, 张甫翊, 徐炳祥, 等. 通信原理[M]. 5 版, 北京: 国防工业出版社, 2001.
- [6] Massey J L. Optimum frame synchronization[J]. IEEE Transactions on Communications, 1972, 20 (2): 115-119.
- [7] Choi Z Y, Lee Y H. Frame synchronization in the presence of frequency offset[J]. IEEE Transactions on Communications, 2002, 50(7): 1062-1065.
- [8] Choi Z Y, Lee Y H. On the use of double correlation for frame synchronization in the presence of frequency offset[C]. Vancouver: 1999 IEEE International Conference on Communications (Cat. No. 99CH36311), 1999, 2: 958-962.
- [9] Kopansky A, Bystrom M. Frame synchronization for noncoherent demodulation on flat fading channels [C]. New Orleans: 2000 IEEE International Conference on Communications, 2000: 312-316.
- [10] Kopansky A, Bystrom M. Detection of a periodically embedded synchronization patterns on rayleigh fading channels[J]. IEEE Transactions on Communications, 2006, 54(11): 1928-1932.
- [11] Chiani M, Martini M G. On sequential frame synchronization in AWGN channels[J]. IEEE Transactions on Communications, 2006, 54(2): 339-348.
- [12] Wymeersch H, Moeneclaey. ML frame synchronization for turbo and LDPC codes[C], Australia: Proceeding International Symposium. DSP and Communications Systems, 2003: 15-22.
- [13] Lee D U, Kim H, Jones C R, et al. Pilotless frame synchronization for LDPC-coded transmission systems[J]. IEEE Transactions on Signal Processing (S1053-587X), 2008, 56(7): 2865-2874.
- [14] Lee D U, Kim H, Jones C R, et al. Pilotless frame synchronization via LDPC code constraint feedback [J]. IEEE Communications Letters, 2007, 11(8): 683-685.
- [15] 陈智雄, 苑津莎. 基于准循环 LDPC 码译码软信息的码辅助帧同步算法[J]. 系统仿真学报, 2011, 23(9): 1956-1960.
- [16] Matsumoto W, Imai H. Blind synchronization with enhanced sum-product algorithm for low-density parity-check codes[C]. Honolulu: The 5th International Symposium on Wireless Personal Multimedia Communications, 2002, 3: 966-970.
- [17] 李智信. 卫星自适应传输中的关键技术研究[D]. 北京: 北京理工大学, 2014.
- [18] Mustafa E, Sun F, Lin-Nan L. DVB-S2 low density parity check codes with near Shannon limit performance[J]. International Journal of Satellite Communications & Networking, 2004, 22(3): 269-279.
- [19] Park J W, Sunwoo M H, Pan S K, et al. An efficient data-aided initial frequency synchronizer for DVB-S2 [C]. Shanghai: Proceedings of IEEE Workshop on Signal Processing Systems, 2007: 645-650.
- [20] 罗加兴. 新一代卫星数字广播中的接收机同步算法研究[D]. 天津: 天津大学, 2017.
- [21] Corazza G E, Pedone R, Villanti M. Frame acquisition for continuous and discontinuous transmission in the forward link of satellite systems[J]. International Journal of Satellite Communications and Networking, 2006, 24(2): 185-201.
- [22] Villanti M, Salmi P, Corazza G E. Differential post detection integration techniques for robust code acquisition[J]. IEEE Transactions on Communications, 2007, 55(11): 2172-2184.
- [23] Viterbi A J. CDMA: principles of spread spectrum communication[M]. Boston: Addison Wesley Longman Publishing Co., Inc., 1995.