

GPS L1C 信号 LDPC 译码器设计*

王建辉, 李井源, 倪少杰, 欧 钢

(国防科技大学 电子科学与工程学院, 湖南 长沙 410073)

摘要: GPS 现代化中的 L1C 信号使用了 LDPC 码, 以提高接收机的弱信号接收能力。L1C 信号中使用的 LDPC 码没有循环或准循环结构, 这为译码器设计带来了难度。为了降低译码器实现复杂度, 在分析 L1C 信号 LDPC 码校验矩阵结构的基础上, 通过对校验节点分类和改变变量节点更新方法, 提出了一种低存储量和简化译码器控制逻辑的低复杂度译码器结构, 并通过仿真验证了译码器译码结果的正确性。

关键词: GPS L1C 信号; LDPC 码; 低复杂度译码器

中图分类号: TN914.53 **文献标志码:** A **文章编号:** 1001-2486(2013)01-0065-05

LDPC decoder for GPS L1C signal

WANG Jianhui, LI Jingyuan, NI Shaojie, OU Gang

(College of Electronic Science and Engineering, National University of Defense Technology, Changsha 410073, China)

Abstract: In GPS modernization, LDPC code is used in L1C signal to enhance the receivers' performance in weak signal environments. The LDPC codes used in L1C signal do not have cyclic or quasi-cyclic structure, which increases the complexity of decoder. To reduce the implementation complexity of decoder, the structure of parity check matrix of the LDPC code used in GPS L1C signal was analyzed. Based on the analysis, a complexity-reduced decoder was presented. The presented decoder divides the check nodes into two groups and changes the variable nodes updating way, which can reduce the storage and simplify the control logic of the decoder. The decoding results of the presented decoder are validated by simulation.

Key words: GPS L1C signal; LDPC codes; low complexity decoder

GPS 现代化在 L1 频点新增加了民用信号, 即 L1C 信号^[1-2]。L1C 信号中加入了导频通道, 这可以提高弱信号条件下的捕获与跟踪性能, 但会降低数据通道的接收功率。为了降低弱信号条件下解调导航电文的误码率, L1C 信号使用了编码增益较高的 LDPC 码^[3]。

LDPC 码^[4]最早在 1963 年由 Gallager 提出, 是继 turbo 码^[5]之后被重新发现的一种高增益信道编码^[6-7]。与 turbo 码相比, LDPC 码具有更低的错误平层, 译码器可以使用并行译码算法^[7]。L1C 信号使用了两种码长的 LDPC 码, 两种码具有类似的结构, 均为基于 PEG 算法^[8]构造的码。该类码没有循环或准循环结构, 这使得译码器的设计较复杂。本文在分析 L1C 信号 LDPC 码校验矩阵结构的基础上, 提出了一种能够降低译码器存储量和简化控制逻辑的译码器结构, 并通过仿真验证了本文提出的译码器的正确性。

1 L1C 信号中的 LDPC 码

L1C 信号帧结构如图 1 所示, 每一帧电文包含 3 个子帧。子帧 1 为 9 比特 TOI, 采用的信道编码为 BCH 码, 编码后生成 52 比特电文符号; 子帧 2 为 576 比特时钟和历书信息, 以及 24 比特 CRC 校验, 采用码率为 0.5 的 LDPC 码, 编码后生成 1200 比特电文符号; 子帧 3 为 250 比特快变电文和 24 比特 CRC 校验, 采用码率 0.5 的 LDPC 码, 编码后生成 548 比特电文符号。子帧 2 和子帧 3 编码后的 1748 比特电文符号经过维数为 38×46 的块交织器^[1]交织后(1748 比特数据逐行填满交织器, 然后逐列输出), 与经过 BCH 编码的 52 比特 TOI 数据拼接为 L1C 信号的一帧导航电文, 共 1800 比特, 持续时间为 18s。

两种码长的 LDPC 码结构类似^[1], 本文以码长 548 的码为例分析译码器结构。LDPC 码的结构可由其校验矩阵表示, L1C 信号中子帧 3 采用

* 收稿日期: 2011-12-07

基金项目: 国家部委资助项目

作者简介: 王建辉(1979—), 男, 河北定州人, 博士研究生, E-mail: wjh0369@163.com;

欧钢(通信作者), 男, 教授, 博士, 博士生导师, E-mail: ougangcs@163mail.com

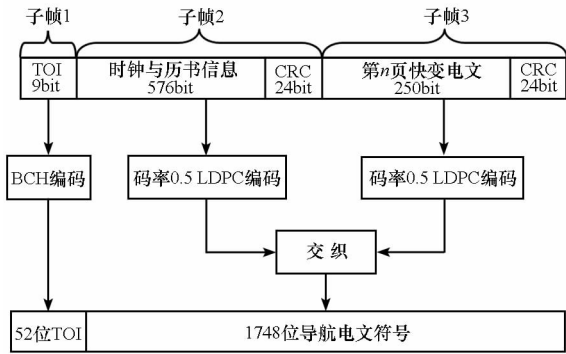


图 1 LIC 信号帧结构

Fig. 1 Frame structure of LIC signal

的 LDPC 码的校验矩阵结构如图 2 所示。其中, A 为随机构造的稀疏矩阵(“1”的个数远少于“0”的个数), D 为 1, C 和 E 为稀疏的行矢量, B 为稀疏的列矢量, T 为可逆的下三角矩阵。 T 及其逆矩阵的结构如图 3、图 4 所示(两个矩阵维数均为 273×273), 其中空白处代表“0”, 点代表“1”。

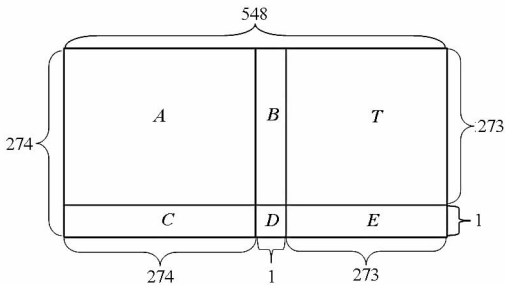


图 2 LIC 信号中 LDPC 码校验矩阵结构

Fig. 2 Parity check matrix structure of LDPC code in LIC signal

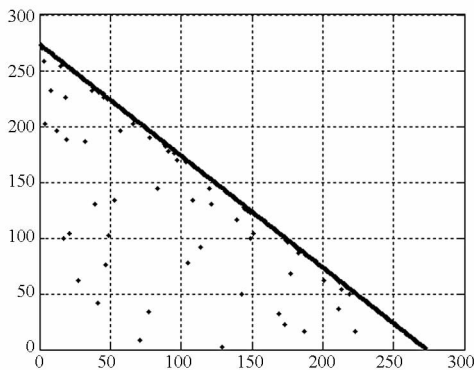


图 3 T 矩阵结构

Fig. 3 Structure of matrix T

文献 [1] 给出了编码器结构, 设码字矢量(行矢量)为 $c = (s, p_1, p_2)$, 其中 s 为信息位矢量, p_1 和 p_2 为校验位矢量。由 $Hc^T = 0$ 可得

$$As^T + Bp_1^T + Tp_2^T = 0 \quad (1)$$

$$Cs^T + Dp_1^T + Ep_2^T = 0 \quad (2)$$

由以上两式解得

$$p_1^T = (D + ET^{-1}B)^{-1}(ET^{-1}A + C)s^T \quad (3)$$

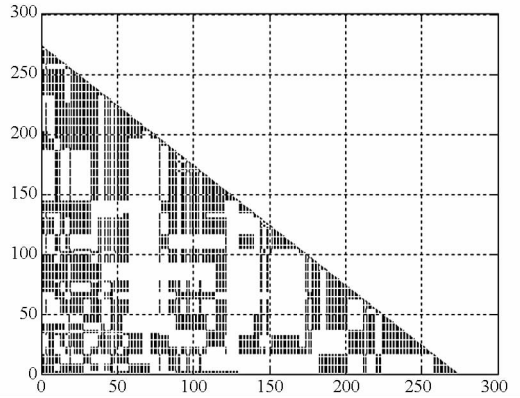


图 4 T 的逆矩阵结构

Fig. 4 Structure of inverse matrix of T

$$p_2^T = T^{-1}(As^T + Bp_1^T) \quad (4)$$

LIC 采用的 LDPC 码为系统码, 编码时只需要根据式 (3)、(4) 求出校验位即可。 p_1 的维数为 1×1 , 计算 p_1 时只需要行矢量与列矢量的乘积运算, 行矢量 $(D + ET^{-1}B)^{-1}(ET^{-1}A + C)$ 可以预先计算并存储, 其中 $(D + ET^{-1}B)^{-1}$ 的值为“1”; 计算 p_2 时需要矩阵与矢量的乘积和矢量加法运算, 矩阵 $T^{-1}A$ 与列矢量 $T^{-1}B$ 可预先计算并存储。

2 译码器实现复杂度

接收机对 LDPC 码的译码一般采用 BP 算法简化算法(如最小和算法、归一化最小和算法等)^[9]。归一化最小和算法计算复杂度低且性能与 BP 算法差距较小^[9], 本文的译码器设计采用归一化最小和算法。

首先介绍与 LDPC 码的校验矩阵有关的几个概念。将校验矩阵的每一列用一个圆圈表示, 每个圆圈称为一个变量节点, 变量节点数目与校验矩阵列数相同, 每个变量节点的序号与相应的列序号相同。将校验矩阵的每一行用一个正方形表示, 每一个正方形称为一个校验节点, 校验节点数目与校验矩阵行数相同, 每个校验节点的序号与相应的行序号相同。若矩阵的第 j 行第 i 列元素为“1”, 则将第 j 个校验节点和第 i 个变量节点用一条直线连接, 该直线称为边, 边的数量与校验矩阵中“1”元素数量相同。校验节点、变量节点和边构成了校验矩阵的 Tanner 图^[10]。图 5 为一个三行四列矩阵及其 Tanner 图。记与第 i 个变量节点相连的校验节点序号集合为 M_i (图 5 中 $M_1 = \{1, 3\}$), 该集合元素个数与校验矩阵第 i 列“1”的个数相同; 记与第 j 个校验节点相连的变量节点序号集合为 N_j (图 5 中 $N_1 = \{1, 2, 4\}$), 该集合元素个数与校验矩阵第 j 行“1”的个数相同。

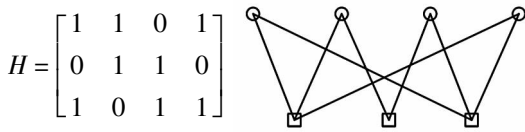


图 5 校验矩阵与 Tanner 图

Fig.5 Parity check matrix and Tanner graph

LDPC 码的 BP (belief propagation) 译码算法是一种基于消息传递的迭代译码算法,在 Tanner 图中无环^[11]时其译码性能与最大似然译码器相同。BP 算法第 l 次迭代中,第 j 个校验方程满足的条件下,第 i 个码元的对数似然比^[11] (码元为“0”的概率与为“1”的概率的比值的对数)记为 r_{ji}^l (文献[11]的 5.4 节给出了表达式及推导过程)。第 l 次迭代中第 j 个校验节点传递给第 i 个变量节点的软信息为 r_{ji}^l ,每次迭代生成的 r_{ji}^l 的数量与校验矩阵第 j 行“1”的个数相同。第 l 次迭代中,在接收到与第 i 个变量节点相连的所有校验节点 (第 j 个除外) 的软信息后,第 i 个变量节点在下次迭代中传递给第 j 个校验节点的对数似然比记为 q_{ij}^l (文献[11]的 5.4 节给出了表达式和推导过程),每次迭代生成的 q_{ij}^l 的数量与校验矩阵第 i 列“1”的个数相同。 r_{ji}^l 与 q_{ij}^l 的取值范围与接收机中码跟踪环输出的相关积分值量化位数有关。

设发送的码字第 i 位码元为 $c_i (1 \leq i \leq 548)$, 首先将二进制码序列映射为适合信道传输的符号序列 $x_i = 1 - 2c_i$, 经过信道传输和接收机量化后接收到的受噪声干扰的序列为 $y_i = x_i + n_i$, 其中 n_i 为量化后的高斯白噪声 (y_i 的取值范围与码环输出时采用的量化位数有关)。

记取符号函数为

$$\text{sign}(x) = \begin{cases} 1, & x > 0 \\ -1, & x \leq 0 \end{cases} \quad (5)$$

最小和算法的计算步骤:

(1) 初始化

$$q_{ij}^0 = y_i \quad (6)$$

(2) 校验节点更新

$$r_{ji}^l = \lambda \left(\prod_{k \in N_j, k \neq i} \alpha_{kj}^{l-1} \right) \min_{k \in N_j, k \neq i} \beta_{kj}^l \quad (7)$$

其中, $\alpha_{kj}^l = \text{sign}(q_{kj}^{l-1})$, $\beta_{kj}^l = |q_{kj}^{l-1}|$, λ 为归一化因子。

(3) 变量节点更新

$$q_{ij}^l = y_i + \sum_{k \in M_i, k \neq j} r_{ki}^l \quad (8)$$

(4) 译码判决

$$q_i^l = y_i + \sum_{k \in M_i} r_{ki}^l \quad (9)$$

$$\hat{c}_i = \begin{cases} 0, & q_i^l > 0 \\ 1, & q_i^l < 0 \end{cases} \quad (10)$$

为了控制译码时延,避免译码器陷入死循环,需要为译码器预先设定一个最大迭代次数。每次迭代完成后,若 $H\hat{c}^T = 0$ 或达到最大迭代次数,结束运算,否则回到步骤(2)。

完成上述译码器运算需要存储每个校验节点和变量节点的连接关系。校验矩阵每行最多有 8 个“1”,最少有 7 个“1”,整个矩阵共有 2071 个“1”,所以存储整个矩阵需要存储 2071 个“1”的位置。译码器需要分别存储每个校验节点和变量节点的连接关系,这就需要以行和列的形式存储两次校验矩阵。此外,译码器还需要存储每个码字 548 个码元对应的信道初始信息,译码器迭代译码过程中需要存储变量节点软信息和校验节点软信息。

常规的译码器结构如图 6 所示。其中 CNU 单元为校验节点更新单元,VNU 单元为变量节点更新单元。校验节点或变量节点更新时,首先通过校验矩阵读入该节点的连接关系。所有校验节点完成更新后才能进行变量节点更新。

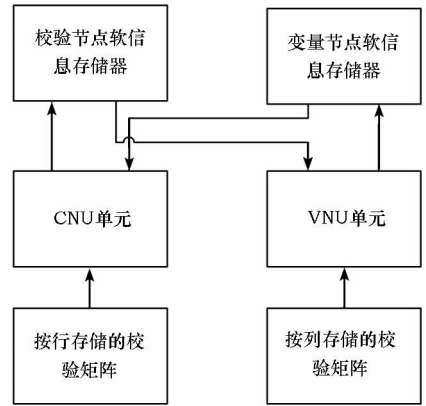


图 6 常规 LDPC 译码器结构图

Fig.6 Structure of traditional LDPC decoder

译码器进行校验节点更新时需要首先读入与该节点相连的所有变量节点的软信息,不同校验节点的重量不同,这增加了译码器控制逻辑设计的难度。变量节点的重量分布更复杂,这进一步增加了译码器的复杂度。按行存储的校验矩阵结构如图 7 所示,方格中数字为该行包含的“1”的列序号(从 1 开始计数)。若直接存储整个矩阵需要一个 274×8 的存储器。但由于矩阵每行包含“1”的个数不同,还需标识出每行包含的“1”的个数,这会导致译码器控制逻辑复杂。按列存储的校验矩阵存在同样的问题。

1	21	153	209	234	249	276	
54	120	210	233	254	276	277	
⋮							
69	131	183	225	260	267	547	548
23	86	177	227	275	546	548	

图 7 按行存储的校验矩阵存储结构

Fig. 7 Structure of parity check matrix stored row by row

3 低复杂度译码器设计

为了降低不规则度分布对译码器设计的影响,本文提出了如下译码器结构设计原则。

(1)按行重调整校验矩阵行的顺序,使行重相同的行相邻;

(2)由于列重分布较为复杂,更新变量节点时按行的顺序更新,同步完成所有校验节点和变量节点的更新。

LIC 信号中码长 548 的 LDPC 码的校验矩阵所有行中,重量为 7 的行共有 121 行,重量为 8 的行共有 153 行。将校验矩阵的所有行重新排序,矩阵的前 121 行重量均为 7,后 153 行重量均为 8。将校验节点按重量分为两组,每组内节点重量相同。按行存储校验矩阵需要两个维数分别为 121×7 和 153×8 的存储器,结构如图 8 所示,两个存储器中每行重量均相同,不需要额外标明每行的重量。

1	21	153	209	234	249	276
---	----	-----	-----	-----	-----	-----

⋮

23	86	177	227	275	546	548
----	----	-----	-----	-----	-----	-----

(a) 重量为 7 的行存储器

74	87	212	222	255	273	279	280
----	----	-----	-----	-----	-----	-----	-----

⋮

69	131	183	225	260	267	547	548
----	-----	-----	-----	-----	-----	-----	-----

(b) 重量为 8 的行存储器

图 8 行顺序调整后的校验矩阵存储结构

Fig. 8 Storage structure of parity check matrix with rows rearranged

本文提出的译码器结构如图 9 所示,只需按行存储一次校验矩阵,存储量比常规译码器少 1 倍。为了在更新完校验节点的同时完成变量节点更新,需要一个维数 1×548 的变量节点软信息累加器,每个校验节点更新完成后将校验节点软信息累加到变量节点软信息累加器的相应位置。

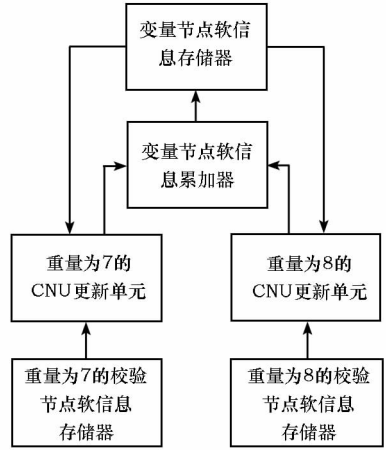


图 9 本文译码器结构图

Fig. 9 Structure of the presented decoder

译码器每次迭代的运算流程:

(1)利用信道初始信息 y_i 初始化变量节点软信息累加器。

(2)更新重量为 7 的校验节点。按校验矩阵的连接关系读取相应的变量节点软信息和校验节点软信息,按式(7)完成校验节点更新。将本次迭代得到的 7 个该校验节点的 r_{ji}^l 分别累加到变量节点软信息累加器的对应位置。

(3)更新重量为 8 的校验节点。按校验矩阵的连接关系读取相应的变量节点软信息和校验节点软信息,按式(7)完成校验节点更新。将本次迭代得到的 8 个该校验节点的 r_{ji}^l 分别累加到变量节点软信息累加器的对应位置。

(4)利用变量节点软信息累加器的值更新变量节点软信息存储器。

经过上述流程就完成了译码器的一次迭代运算,完成所有校验节点更新后也同步完成了所有变量节点的更新。第 l 次迭代完成后变量节点软信息存储器中的值为 q_i^l ,在下次迭代中 q_{ij}^l 的计算式为

$$q_{ij}^l = q_i^l - r_{ji}^l \quad (11)$$

本文提出的译码器控制逻辑框图如图 10 所示(译码器为全串行结构)。与常规译码器控制流程相比,本文提出的译码器在每次迭代过程中仅需读取校验节点的连接关系(按行读取校验矩阵),不需要频繁读取变量节点的连接关系和校验矩阵的列重分布,不需要设置专门的 VNU 单元。每次迭代中,前 121 个校验节点更新流程相同,后 153 个校验节点更新流程相同。所有校验节点更新完成后,变量节点也同步更新完成。

为了验证本文提出的译码器输出结果的正确性,将一组随机生成的数据输入本文译码器和常

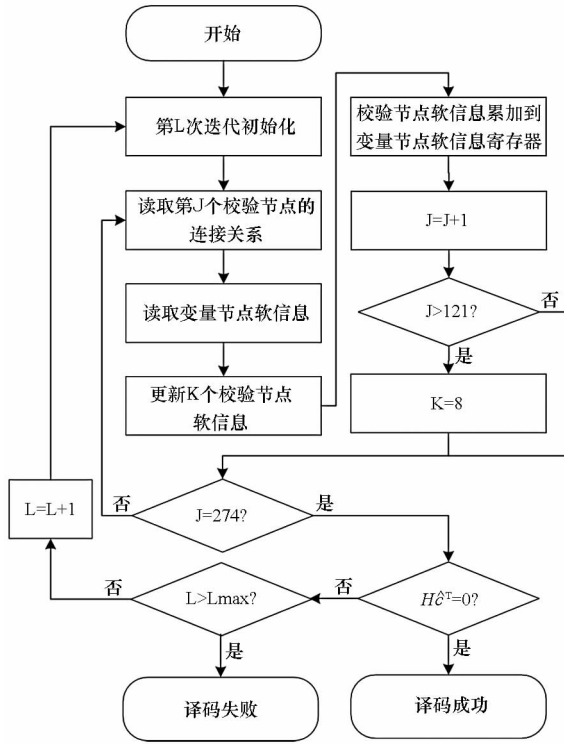


图 10 译码器控制逻辑框图

Fig. 10 Control logic frame of the presented decoder

规结构的译码器,固定迭代 100 次,并记录每次迭代输出的 $q_i^l (1 \leq l \leq 100, 1 \leq i \leq 548)$ 。每次迭代两种译码器输出的结果相同,因此本文译码器结构能够输出正确的译码结果。

4 结束语

在分析 GPS L1C 信号中 LDPC 码校验矩阵结构的基础上,提出了一种能够简化控制逻辑、降低存储量的译码器结构。本文提出的译码器在每次迭代中,能够在完成所有校验节点更新后同步完成所有变量节点更新,不需要读取校验矩阵的列,因此只需要按行存储校验矩阵,降低了存储量。本文译码器按校验节点重量分布将所有校验

节点分成两组,每组内节点具有相同的重量,可分别按规则码方式更新校验节点软信息。

与常规译码器相比,本文译码器需要一个额外的维数为 1×548 的变量节点软信息累加器,以同步完成校验节点和变量节点软信息更新。

参考文献 (References)

- [1] Navstar Global Positioning System. Navstar GPS space Segment/User segment L1C interfaces(Revision A) [EB/OL], 2010. <http://www.losangeles.af.mil/shared/media/document/AFD-100813-047.pdf>. (2010-6-30)
- [2] Betz J W, Blanco M A, Cahn C R, et al. Description of the L1C signal [C]//Proceedings of ION GNSS, 2006: 2080-2091.
- [3] Dafesh P A, Vallés E L, Sklar D J, et al. Data message performance for the future L1C GPS signal [C]//ION GNSS 20th International Technical Meeting of the Satellite Division, 2007: 2519-2528.
- [4] Gallager R G. Low-density parity-check codes[M]. Cambridge MA: MIT Press, 1963.
- [5] Berrou C, Glavieux A, Thitimajshima P. Near Shannon limit error-correcting and decoding: turbo codes [C]//IEEE Int Conf on Communications, 1993: 1064-1070.
- [6] Mackay D J C, Neal R M. Near Shannon-limit performance of low-density parity-check codes[J]. Electronics Letters, 1997, 33(6): 457-458.
- [7] Richardson T J, Urbanke R L. The capacity of low-density parity-check codes under message-passing decoding[J]. IEEE Trans on Inform Theory, 2001, 47(2): 599-618.
- [8] Hu X Y, Eleftheriou E, Arnold D M. Progressive edge-growth Tanner graphs [C]//IEEE Global Telecommun Conf, 2001: 995-1001.
- [9] Fossorier M P C, Mihaljevic M, Imai H. Reduced complexity iterative decoding of low density parity check codes based on belief propagation[J]. IEEE Trans on Communications, 1999, 47(5): 673-680.
- [10] Tanner R M. A recursive approach to low complexity codes [J]. IEEE Trans on Inform Theory, 1981, 27(6): 533-547.
- [11] Ryan W E, Lin S. Channel codes-classical and modern[M]. Cambridge University Press, 2009.