

IRA 码预检测折线修正译码算法设计

包志祥*, 吕娜, 陈柯帆

(空军工程大学 信息与导航学院, 西安 710077)

(*通信作者电子邮箱 damon7.7@foxmail.com)

摘要: 不规则重复累积码(IRA)的译码通常采用置信传播(BP)译码算法,然而 BP 译码算法需进行双曲正切函数计算,复杂度高,不利于硬件实现。为此,提出一种基于分段函数修正和预检测机制结合的译码算法,通过对折线近似译码算法进行非均匀误差补偿,使其性能接近 BP 算法;同时引入预检测机制对校验节点信息传递进行预检测,判断出对后续迭代影响微小的对数似然信息,并将其移出迭代循环,从而减少计算量。仿真结果表明,该算法通过分段函数修正近似双曲正切函数、引入预检测机制能大大降低运算复杂度,并且具有接近 BP 算法的译码性能。

关键词: 重复累积码;置信传播译码算法;折线近似译码算法;误差补偿;预检测机制

中图分类号: TN919 **文献标志码:** A

Algorithm design of early detection and offset polyline decoding for IRA codes

BAO Zhixiang*, LYU Na, CHEN Kefan

(Information and Navigation Academy, Air Force Engineering University, Xi'an Shaanxi 710077, China)

Abstract: Irregular Repeat Accumulate (IRA) codes' decoding usually adopts Belief Propagation (BP) decoding algorithm, but BP decoding algorithm needs hyperbolic tangent calculation, so its hardware implementation is very difficult because of the high complexity. A decoding algorithm combining the early detection mechanism and offset polyline was put forward. Its performance would approach to BP algorithm via non-uniform error compensation for polyline approximation decoding algorithm. And the early detection method was introduced which observed the transmitted information of check nodes in advance, judged the lines' log-likelihood value which had negligible influence on the next iteration and moved it out of the iteration. So the computational complexity of next iterations was reduced. The simulation results show that the proposed algorithm greatly reduces the computational complexity through the offset polyline approximating the hyperbolic tangent, and the decoding performance is close to BP algorithm.

Key words: Repeat Accumulate code (RA); Belief Propagation (BP) decoding algorithm; polyline approximation decoding algorithm; error compensation; early detection mechanism

0 引言

在信道编码领域,自 1948 年香农提出信道编码理论^[1]以来,研究者们开始寻找接近香农限的纠错码,随着对 Turbo 码和低密度奇偶校验码(Low Density Parity Check Code, LDPC)的研究不断深入,1998 年,Divsalar 等^[2]提出了重复累积(Repeat Accumulate, RA)码,RA 码是一种具有线性编码时间和线性译码时间的好码,同时他们证明了 RA 码具有接近香农限的潜力。2000 年, Jin 等^[3]从不规则 LDPC 码得到启发,提出了不规则 RA (Irregular Repeat Accumulate, IRA)码。IRA 码可以取得与不规则 LDPC 码同样优越的性能。由于 IRA 码的校验矩阵也是稀疏的,所以从本质上说,IRA 码的校验矩阵也是低密度校验矩阵,因此 LDPC 码的置信传播(Belief Propagation, BP)算法和最小和(Min-Sum)算法同样适用于 IRA 码的译码^[4]。但是 BP 算法在迭代中需要进行双曲正切函数的计算,因此运算复杂度高,硬件实现困难;考虑到硬件资源分配问题^[5],RA 码最小和译码算法对校验节点更新公式进行了近似替代,降低了运算复杂度,但同时也降低了译码性能。高宏峰等^[6]提出的曲线折线化译码算法是用分段函

数代替双曲正切函数进行近似运算,也降低了 BP 算法的复杂度。

本文通过对三折线近似译码算法进行研究改进,提出一种 IRA 码预检测折线修正译码算法(Early Detection and Offset polyline decoding algorithm, EDO),通过查找表对分段折线结果进行修正,保证较好的译码性能;在折线化译码算法中引入预检测机制,使运算复杂度降低。首先,通过分段函数补偿表,对校验节点运算结果进行补偿。补偿表是对误差函数的非均匀量化,利用很小的存储空间和一次查表修正,能大大提高折线近似译码算法的性能。然后通过置信阈值,在迭代中对校验节点边界进行预检测,超过置信阈值的边在之后的迭代中将不进行运算,降低了算法的运算复杂度。最后本文进行了仿真分析,证明所提预检测折线修正译码算法具有优异性能。

1 IRA 码基本原理

1.1 IRA 码编码结构

IRA 码与不规则 LDPC 码相对应,指的是信息节点或者校验节点度数不相等的 RA 码。一般情况下,信息节点的度

收稿日期:2015-01-07;修回日期:2015-04-07。 基金项目:国家自然科学基金资助项目(61472443, 61302153)。

作者简介:包志祥(1991-),男,安徽庐江人,硕士研究生,主要研究方向:航空数据链;吕娜(1970-),女,陕西西安人,教授,博士,主要研究方向:军事航空通信、航空数据;陈柯帆(1990-),男,四川南充人,硕士研究生,主要研究方向:航空数据链。

数不相等,校验节点的度数是相等的。图1为IRA码的Tanner图表示。

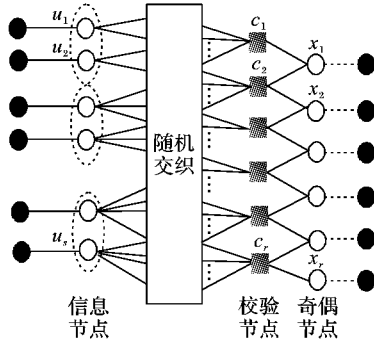


图1 IRA码的Tanner图表示

IRA码的参数为 $(f_1, f_2, \dots, f_i, \dots, f_j; a)$, 其中满足 $\sum_{i=1}^j f_i = 1$, 且 $f_i \geq 0$; 通常 $f_1 = 0$, f_i 表示度数为 i 的信息节点的比例系数, a 是正整数, a 是校验节点连接信息节点边的个数。变量节点分为信息节点、奇偶节点, 左边 s 个表示信息节点, 右边 r 个表示奇偶节点, 中间为 r 个校验节点。信息节点与校验节点之间通过 $r \cdot a$ 个边随机交织连接, 校验节点与奇偶节点之间为 Z 字型连接。

非系统 IRA 码在 $a > 1$ 时译码算法不能收敛, 因此 IRA 码主要限于系统 IRA 码。IRA 码的编码框图如图2所示, 信息节点产生 ra 个信息, 经过随机交织后得到的值为 $(C_1, C_2, \dots, C_{ra})$, 设 $x_0 = 0$, 则奇偶节点的输出为 $x_j = x_{j-1} + \sum_{i=1}^a C_{(j-1)a+i} (j = 1, 2, \dots, r)$ 。

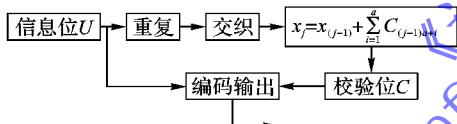


图2 IRA码编码器

1.2 IRA码的BP译码算法

IRA 的 BP 译码算法^[4]中, 信息是以概率密度的对数似然比 (Loglikelihood Ratio, LLR) 在所有边上传递的, 即 $M = \ln p(0)/p(1)$ 。定义从信息节点 u 到校验节点 c 传递的信息为 $M[u, c]$, 从奇偶节点 x 传递给校验节点 c 的信息为 $M[x, c]$, 从校验节点 c 传递给变量节点的信息为 $M[c, u]$ 和 $M[c, x]$; 信息传递初始时, 变量节点传出的信息为信道接收的 LLR 值, 设为 $M[u]$ 。

从信息节点 u 和奇偶节点 x 传递给校验节点的信息为:

$$M[u, c_i] = M[u] + \sum_{c' \in N(u) \setminus c_i} m[c', u] \quad (1)$$

$$M[x, c_i] = M[u] + M[c', x]; c' \in N(x) \setminus c_i \quad (2)$$

校验节点 c 传递给变量节点的信息为:

$$M[c, u_i] = 2 \operatorname{arctanh} \left(\prod_{u' \in N(c) \setminus u_i} \tanh(M[u', c]/2) \cdot \prod_{j=1}^2 \tanh(M[x_j, c]/2) \right); i = 1, 2, \dots, a \quad (3)$$

$$M[c, x_j] = 2 \operatorname{arctanh} \left(\prod_{i=1}^a \tanh(M[u_i, c]/2) \cdot \tanh(M[x', c]/2) \right); j = 1, 2 \quad (4)$$

对每个信息节点的所有输入求和:

$$s[u] = \sum_{c' \in N(u)} M[c', u] + M_n(u) \quad (5)$$

如果 $s(u) \geq 0$, 则译码为 0; 否则译码为 1。

1.3 IRA码的三折线近似译码算法

式(3)、(4)中存在双曲正切函数及反双曲正切函数的运算, 运算量较大, 工程实现困难。高宏峰等^[6]提出了曲线折线化算法。根据 IRA 码校验节点更新公式, 校验节点的输出信息 L 为:

$$L = \prod_{i=1}^{a+1} \operatorname{sgn}(m_i) f \left(\sum_{i=1}^{a+1} f(|m_i|) \right) \quad (6)$$

其中: m_i 为从变量节点到校验节点的输入信息; L 为输出信息。

$$f(m) = \ln((e^m - 1)/(e^m + 1)) \quad (7)$$

经过式(8)的变换, 可以看出校验节点输出信息公式可变为加法和 $f(m)$ 的计算。由于 $f(m)$ 函数运算比较复杂, 如果利用分段直线函数代替其进行计算, 那么就降低了译码运算的复杂度。将 $f(m)$ 用三段函数 $f'(x)$ 代替, 就称之为三折线近似算法。高宏峰等^[6]经过计算, 给出三段函数 $f'(x)$ 的表达式:

$$f'(x) = \begin{cases} -4.67x + 5, & x \leq 0.8841 \\ -0.241x + 1.07, & 0.8841 < x \leq 5 \\ 0, & x > 5 \end{cases} \quad (8)$$

2 IRA码预检测折线修正译码算法

三折线译码算法使用三个分段函数(如式(8)所示)对校验节点更新公式(式(6))中 $f(x)$ 进行近似代替。与 BP 算法相比, 其不需进行正反双曲正切函数的计算, 运算复杂度大大降低, 且便于硬件实现, 但其译码性能也有所降低。本文提出预检测折线修正译码算法, 首先对三折线近似译码算法中的分段函数计算进行补偿; 然后引入预检测机制, 在迭代过程中对校验节点边传递信息进行预检测, 选出不需要继续迭代的边信息, 在降低算法运算复杂度的同时保证较高的译码性能。

2.1 分段函数补偿

IRA 码校验节点的输出信息 L 和输入信息 $m_i (i = 1, 2, \dots, a, a+1)$ 之间存在关系:

$$\tanh(L/2) = \prod_{i=1}^{a+1} \tanh(m_i/2) \quad (9)$$

由于 $0 \leq \tanh(x) < 1$, 且 $\tanh(x)$ 为单调递增函数, 所以:

$$\operatorname{sgn}(L) = \prod_{i=1}^{a+1} \operatorname{sgn}(m_i)$$

$$\ln(\tanh(|L|/2)) = \sum_{i=1}^{a+1} \ln(\tanh(|m_i|/2))$$

因为

$$\tanh\left(\frac{|m_i|}{2}\right) = \frac{e^{|m_i|} - 1}{e^{|m_i|} + 1}$$

所以

$$f(|m_i|) = -\ln(\tanh(|m_i|/2)) = \ln\left(\frac{e^{|m_i|} + 1}{e^{|m_i|} - 1}\right)$$

由此可得 $f(x) = \ln\left(\frac{e^x + 1}{e^x - 1}\right)$ 及校验节点输出信息更新公式(式(6))。

图3(a)为 $f(x)$ 和三折线函数 $f'(x)$ (式(8))的曲线图, 图3(b)为 $f'(x)$ 和 $f(x)$ 的误差曲线图, 误差 $\delta = f'(x) - f(x)$ 。由图3可以看出, 随着 x 值的增大, 两个函数的值无限接近, 当 x 值较小时, 其误差值较大; x 值较大时, 误差较小, 且误差大小分布不均匀。所以在进行误差补偿修正时, 采用非均匀

量化能使量化结果最优^[7]。

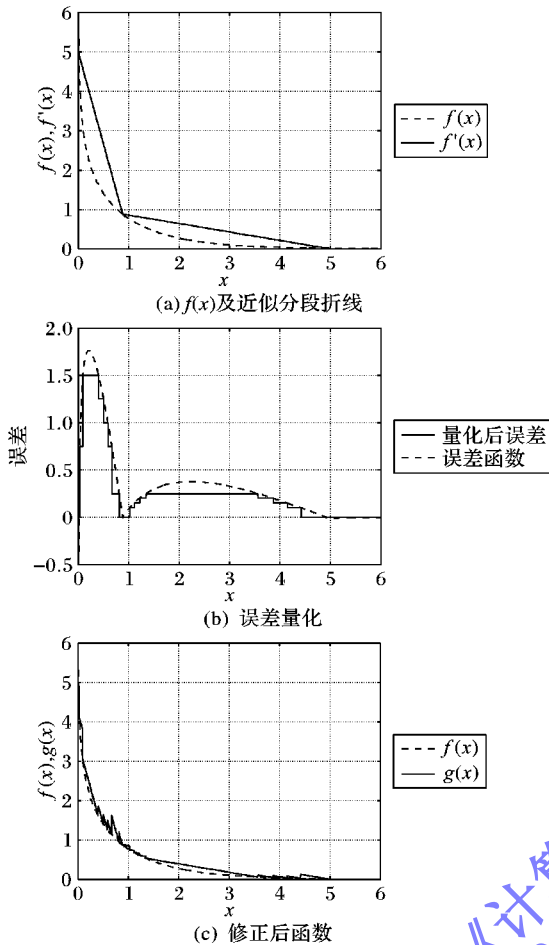


图3 误差及函数修正

图3中: $g(x)$ 为进行补偿修正后的函数;原函数 $f(x)$ 为BP算法中校验节点更新所用的变换公式;分段函数 $f'(x)$ 是高洪峰等^[6]所提折线化译码算法中所用三折线分段函数; δ 为分段函数与原函数之间的误差函数。故有:

$$f(x) \approx g(x) = f'(x) - \delta \quad (10)$$

则校验节点输出信息 L 的更新公式由式(6)变为:

$$L = \prod_{i=1}^{a+1} \text{sgn}(m_i) g\left(\sum_{i=1}^{a+1} g(|m_i|)\right) \quad (11)$$

经研究误差函数值与输入值关系后,对误差函数进行非均匀量化,量化结果如图3(b)所示。现在大多现场可编程门阵列(Field-Programmable Gate Array, FPGA)都是基于查表的,所以对分段函数进行补偿的量化后误差函数也以查表的形式进行。本文所用的误差函数表如表1所示。

表1 误差函数修正补偿表

x	δ
(1.027, 1.115], (4.153, 4.421]	0.10
(1.115, 1.22], (3.871, 4.153]	0.15
(1.22, 1.35], (3.569, 3.871]	0.20
(1.35, 3.569], (0.668, 0.814]	0.25
(0.035, 0.095], (0.589, 0.668]	0.75
(0.504, 0.589]	1.00
(0.405, 0.504]	1.25
(0.095, 0.405]	1.50
其他	0

从图3(c)可以看出,进行补偿修正后的函数 $g(x)$ 与原函数 $f(x)$ 非常接近,误差小,因此,进行补偿修正后算法的性能会接近BP算法,同时避免进行正反双曲正切函数的计算。

2.2 校验节点预检测机制

引入预检测机制是为了降低计算复杂度。校验节点的预检测机制,是通过观察校验节点传递的对数似然比数值,在多大的情况下不影响下一次迭代^[8]。这就意味着满足条件的校验节点边不需要进行后续迭代,因此降低了算法复杂度。

译码算法中是通过校验节点计算公式对校验节点的似然信息进行更新,而预检测机制是在校验更新中加入判断似然值大小,以决定是否进行更新,故称为预检测。若满足条件则退出节点更新过程,若不满足则用计算公式进行更新似然值,据此设计预检测机制流程。

在译码的校验节点信息更新环节,首先对其传递给变量节点的对数似然信息进行预检测,检测结果与所设置阈值 T_c 相比较,若不小于 T_c ,则将该边的传递剔除迭代;若小于 T_c ,则利用校验节点更新公式对其进行更新迭代。

设 $\text{EarlyDetect}[c \rightarrow y]$ 表示从校验节点向奇偶节点传递信息的标志位, $\text{EarlyDetect}[c \rightarrow u]$ 表示从校验节点向奇偶节点传递信息的标志位, m_{cy} 、 m_{cu} 表示校验节点传递的似然信息大小, T_c 是设定的阈值。预检测机制具体流程如下:

预检测机制流程如下:

- 1) Initialization
 $\text{EarlyDetect}[c \rightarrow y] = 0;$
 $\text{EarlyDetect}[c \rightarrow u] = 0;$
- 2) check node to parity node
 If $m_{cy}(i) > T_c$, then $\text{EarlyDetect}[c \rightarrow y(i)] = 1$
 Else $m_{cy} = \prod_{i=1}^{a+1} \text{sgn}(m_i) f\left(\sum_{i=1}^{a+1} f(|m_i|)\right)$
- 3) check node to info node
 If $m_{cu}(i) > T_c$, then $\text{EarlyDetect}[c \rightarrow u(i)] = 1$
 Else $m_{cu} = \prod_{i=1}^{a+1} \text{sgn}(m_i) f\left(\sum_{i=1}^{a+1} f(|m_i|)\right)$

如果 $\text{EarlyDetect}[\cdot] = 1$, 就不需要对其进行更新,只需将其先前的值进行传递。因此,选择合适的预检测阈值 T_c 是非常重要的^[9]。

对于阈值 T_c 如何选取,主要通过仿真进行,本文选取IRA码运用BP译码算法进行译码性能仿真实验。通过观察译码仿真时校验节点传递似然信息的大小发现,在信噪比低于1 dB时,传递的似然值较小,阈值 T_c 选取超过4即对译码性能基本无影响;当信噪比超过1.5 dB之后,校验节点传递似然值大小增大趋势明显;在信噪比达到2 dB时,约有10%校验节点连接边数的似然值传递大小超过9,剩余边数似然值也与9接近,数值较大。此时译码水平已达到 10^{-6} 级,综合考虑,若阈值 T_c 选取9或以上则对译码性能影响较小。

由预检测机制原理可知, T_c 值选取越大对译码性能影响越小,但剔除节点的信息传递边数就越少,减少的计算量也就越小。所以初步选择 $T_c = 7, 8, 9$ 进行仿真实验。将预检测机制直接引入BP算法,以排除分段补偿对译码性能的影响,仿真结果如图4所示。

从图4中可以看出,当 $T_c = 7, 8$ 时,随着信噪比增大,译码性能失真越来越严重,甚至有出现误码平层的趋势。当 $T_c = 9$ 时,译码性能与BP算法相差不大。由此可以得出,当阈值 T_c 值选取较小时,剔除似然比较大的节点多,会严重影响译码性能,且 T_c 选取越小,译码性能失真越严重。综合以上

所述,之后仿真建模时选取 $T_c = 9$ 。

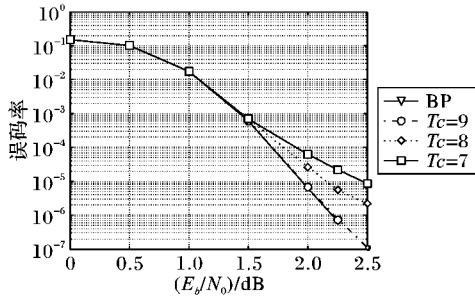


图4 不同 T_c 下译码性能比较

3 仿真与结果分析

3.1 仿真建模

仿真实验采用了 IRA 码,参数为: $f_3 = 0.252744$, $f_{11} = 0.811476$, $f_{12} = 0.327162$, $f_{36} = 0.184589$, $f_{48} = 0.154029$, $a = 8$,码率 $R = 1/2$,码长为 1992。将信息位 $s = 991$ 进行 IRA 编码后所得码字称为 1 帧。仿真采用 BPSK 调制,IRA 码经加性高斯白噪声 (Additive White Gaussian Noise, AGN) 信道传输^[10],仿真实验采用归一化信噪比 E_b/N_0 ,其值可表示为:

$$E_b/N_0 = A^2/(2R\delta^2) \quad (12)$$

其中: E_b 是单位比特信号的平均能量; N_0 是高斯白噪声的功率谱密度; A 是所传输信号的幅值; R 是 IRA 码的码率; δ^2 为高斯白噪声的方差。仿真中采用蒙特卡洛 (Monte Carlo, MC) 仿真方法, M 为数据长度,用下式进行误码率统计:

$$P_e = \frac{1}{M} \sum_{k=1}^M g[Y(k)] \quad (13)$$

$$g[Y(k)] = \begin{cases} 1, & Y(k) \neq U(k) \\ 0, & Y(k) = U(k) \end{cases} \quad (14)$$

共进行两组仿真实验:

1) 比较三折线译码算法及其进行分段函数补偿后的译码性能,同时与 BP 算法作比较。帧长 10000,迭代次数 50。

2) 对文中所提 IRA 码预检测折线修正译码算法 (EDO) 进行仿真,并与三折线近似译码算法、BP 算法等进行比较,论证算法性能。帧长 100000,迭代次数 50, $T_c = 9$ 。

3.2 复杂度分析

表 2 是 IRA 码在不同译码算法下校验节点进行一次信息更新时复杂度比较, a 表示校验节点的度数,即 Tanner 图中每个校验节点所连边的个数。

表2 算法复杂度比较表

算法	tanh	atanh	乘法	加法	查表	仿真时间/h
BP	$a+2$	$a+2$	$3(a+2)$	—	—	43
LUT-BP	—	—	$3(a+2)$	—	$2(a+2)$	—
line3	—	—	$2(a+2)$	$5(a+2)$	—	28
EDO	—	—	$2(a+2)$	$7(a+2)$	$a+2$	34

表 2 中:BP,为置信传播算法;LUT-BP 为用查表实现双曲正切函数的 BP 算法;line3 为三折线译码算法;EDO 为本文所提算法。由表 2 可以看出,传统的 BP 算法在进行一次校验节点更新时需计算 $2(a+2)$ 次双曲正切函数和 $3(a+2)$ 次乘法运算。若硬件实现采用查表法计算双曲正切函数,则每次需查表 $2(a+2)$ 次,由于本文不需与 LUT-BP 算法比较译码性能,所以未对其进行性能仿真,只给出计算复杂度参考。

文献[10]指出,用查找表法实现 BP 算法,查找表大小为 2048 时性能接近传统 BP 算法,硬件实现较为复杂。本文所提 EDO,若不考虑经预检测剔除的校验节点边数,与 BP 算法相比,不需要进行双曲正切函数的计算,同时在校验节点一次更新情况下减少 $a+2$ 次乘法计算;与三折线译码算法相比,EDO 算法更新一次校验节点只增加 $2(a+2)$ 次加法运算与 $a+2$ 次查表,但是其译码性能却与传统 BP 算法接近。表 3 为仿真时经预检测算法后等价校验节点个数的统计平均, a^* 表示经预检测剔除影响因子较低的校验节点链接边后的校验节点平均度数。由表 3 可以看出,随着信噪比增加,预检测所剔除的校验节点信息传递边数逐渐增加。

表3 预检测算法数量表

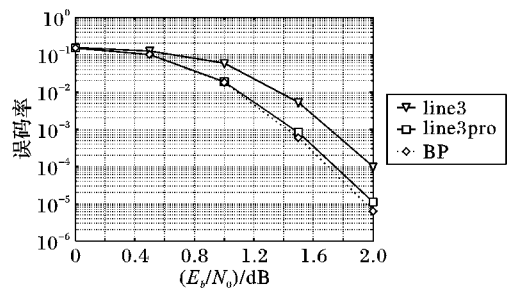
SNR/dB	a^*	SNR/dB	a^*
0	8.00	1.5	7.55
0.5	7.97	2.0	7.16
1.0	7.73	2.5	6.52

为了对仿真用时进行统计比较,考虑到 Matlab 仿真调用函数所用时间等的影响,对仿真时间作了近似统计,完成各算法的仿真时间见表 2 最后一列,说明本文所提算法运算复杂度与 BP 算法相比大大减少。

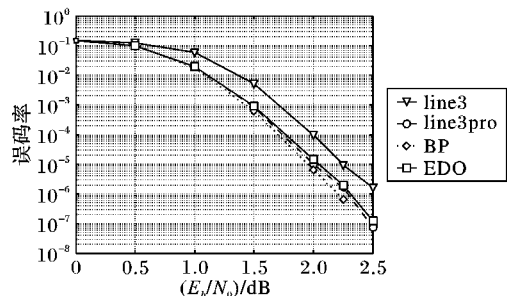
采用非均匀量化误差,只需 8 个值的查找表,若每一个值使用 16 位的浮点数表示^[11],补偿值的查找表只需 128 b,相比传统查找表 BP 算法,所需内存大大减少,同时乘法运算数量减少 1/3,这就意味着更易于硬件实现。

3.3 结果分析

图 5(a)、(b) 分别是 IRA 码采用二进制相移键控 (Binary Phase Shift Keying, BPSK) 调制,经过 AWGN 信道第一组、第二组仿真的译码性能曲线。



(a) 误差补偿后译码性能曲线



(b) 各算法译码性能比较

图5 译码性能仿真曲线

图 5 中 line3 表示三折线近似译码算法,BP 表示传统置信传播算法,line3pro 为对三折线进行非均匀分段补偿后的误差修正后算法,EDO 为本文所提算法。迭代次数 50 次。由图 4 可以看出,本文提出的 IRA 码预检测折线修正译码算法

在阈值 T_c 选取合适的情况下,性能接近 BP 算法。

从 5(a) 中可以看出,对折线近似译码算法中的分段函数进行修正后,译码性能大大提高,与 BP 算法性能接近。图 5(b) 为本文所提算法 EDO 性能与其他译码算法译码性能的仿真比较图。比较 EDO 曲线与 line3pro 曲线可知经预检测后计算量虽然有所下降,但是性能也出现些许下降,但是预检测机制在选取阈值 $T_c = 9$ 时降低算法复杂度的同时,译码性能失真微小。从图 5 中可以看到,对分段函数进行修正补偿,并引入预检测机制之后,算法的性能与 BP 算法接近。

4 结语

本文提出 IRA 码预检测折线修正译码算法,对三折线近似译码算法进行误差补偿修正,增强译码性能;引入预检测机制,降低算法的复杂度。经分析和仿真,证明了所提算法接近 BP 算法的优越性能,同时降低了 BP 算法的运算复杂度,便于硬件实现。

另外,本文所提算法是针对系统 IRA 码的,深入研究广泛适用非系统 IRA 码和规则 RA 码的译码算法是进一步研究的目标。

参考文献:

- [1] SHANNON C E. A mathematical theory of communication [J]. The Bell System Technical Journal, 1948, 27(6): 379 - 423.
- [2] DIVSALAR D, JIN H, MCELLECE R J. Coding theorems for Turbo-like codes [C]// Proceedings of the 36th Annual Allerton Conference on Communication Control and Computing. Piscataway: IEEE, 1998, 9: 201 - 210.
- [3] JIN H, KHANDEKAR A, KH A, *et al.* Irregular repeat-accumulate codes [C]// Proceedings of the 2nd International Symposium on Turbo Codes and Related Topics. Piscataway: IEEE, 2000: 1 - 8.
- [4] FOSSORIER M P C, MIHALJEVIC M, IMAI H. Reduced complexity iterative decoding of low-density parity check codes based on belief propagation [J]. IEEE Transactions on Communications, 1999, 47(5): 673 - 680.
- [5] ABRARDO A, BELLESCHI M, DETTI P, *et al.* A min-sum approach for resource allocation in communication systems [C]// Proceedings of the 2011 IEEE International Conference on Communications. Piscataway: IEEE, 2011: 1 - 6.
- [6] GAO H, XU Z, WU Y. Reduced complexity decoding algorithms for irregular repeat accumulate codes [J]. Journal of University of Electronic Science and Technology of China, 2005, 34(1): 40 - 43. (高宏峰, 许宗泽, 吴援明. IRA 码简化译码算法的研究 [J]. 电子科技大学学报, 2005, 34(1): 40 - 43.)
- [7] HU Z, LIU H. A low-complexity LDPC decoding algorithm for hierarchical broadcasting: design and implementation [J]. IEEE Transactions on Vehicular Technology, 2013, 62(4): 1843 - 1849.
- [8] LEE H-C, UENG Y-L. LDPC decoding scheduling for faster convergence and lower error floor [J]. IEEE Transactions on Communications, 2014, 62(9): 3104 - 3113.
- [9] KIM M H, PARK T D, KIM C S *et al.* An FPGA design of low power LDPC decoder for high-speed wireless LAN [C]// Proceedings of the 2010 12th IEEE International Conference on Communication Technology. Piscataway: IEEE, 2010: 1460 - 1463.
- [10] CHEN X, SHEN Y. Research of RA coding algorithm based on AWGN channel [J]. Journal of Networks, 2012, 7(4): 605 - 612.
- [11] ZHANG J-Y, JIN M-L. An improved decoding algorithm of Repeat Accumulate (RA) codes [J]. Journal of Electronics & Information Technology, 2011, 33(11): 2616 - 2621. (张敬毅, 金明录. 一种改进的 RA 译码算法 [J]. 电子与信息学报, 2011, 33(11): 2616 - 2621.)
- [12] ZHANG L, JIANG Y. A low-hardware consumption FPGA based configurable LDPC decoder [C]// Proceedings of the 2013 International Symposium on Intelligent Signal Processing and Communications Systems. Piscataway: IEEE, 2013: 221 - 224.
- [13] CHANDRASETTY V A, JOHNSON S J, LECHNER G. Memory-efficient quasi-cyclic spatially coupled low-density parity-check and repeat-accumulate codes [J]. IET Communications, 2014, 17(8): 3179 - 3188.
- [10] LIU Z, LIU Z, TANG X. Node self-localization algorithm based on modified particle swarm optimization [J]. Journal of Central South University: Science and Technology, 2012, 43(4): 1371 - 1376. (刘志坤, 刘忠, 唐小明. 基于改进型粒子群优化的节点自定位算法 [J]. 中南大学学报: 自然科学版, 2012, 43(4): 1371 - 1376.)
- [11] LI L, DU Y. Application of modified particle swarm optimization in node locating of wireless sensors networks [J]. Computer Applications and Software, 2014, 31(4): 69 - 72. (李凌燕, 杜永贵. 改进型粒子群优化在 WSNs 节点定位中的应用 [J]. 计算机应用与软件, 2014, 31(4): 69 - 72.)
- [12] WANG Y, YANG J. Localization in wireless sensor network based on improved particle swarm optimization algorithm [J]. Computer Engineering and Applications, 2014, 50(18): 99 - 102. (王亚子, 杨建辉. 改进粒子群算法的无线传感器网络节点定位 [J]. 计算机工程与应用, 2014, 50(18): 99 - 102.)
- [13] WANG S-S, SHIH K-P, CHANG C-Y. Distributed direction-based localization in wireless sensor networks [J]. Computer Communications, 2007, 30(6): 1424 - 1439.
- [14] LIU Y, LYU X, WANG X. Application of hybrid genetic algorithm in WSNs localization [J]. Transducer and Microsystem Technologies, 2014, 33(2): 150 - 153. (刘彦隆, 吕显朋, 王相国. 混合遗传算法在 WSNs 定位中的应用 [J]. 传感器与微系统, 2014, 33(2): 150 - 153)
- [15] SHI Y, EBERHART R. A modified particle swarm optimizer [C]// Proceedings of the 1998 IEEE International Conference on Evolutionary Computation. Piscataway: IEEE, 1998: 69 - 73.
- [16] PAVLYUKEVICH I. Lévy flights, non-local search and simulated annealing [J]. Journal of Computational Physics, 2007, 226(2): 1830 - 1844.
- [17] ZHANG A, YE X, HU H, *et al.* Improved DV-Hop localization algorithm based on RSSI of every jump and jump distance correction [J]. Journal of Instrument, 2012, 33(11): 2552 - 2559. (张爱清, 叶新荣, 胡海峰, 等. 基于 RSSI 每跳分级和跳距修正的 DV-Hop 改进算法 [J]. 仪器仪表学报, 2012, 33(11): 2552 - 2559.)

(上接第 1522 页)