

文章编号:1007-5321(2020)04-0015-06

DOI:10.13190/j.jbupt.2019-222

# 一种动态自纠正最小和 LDPC 码的译码算法

陈 容<sup>1,2</sup>, 陈 岚<sup>1</sup>

(1. 中国科学院 微电子研究所, 北京 100029; 2. 中国科学院大学 电子电气与通信工程学院, 北京 100049)

**摘要:** 针对低密度奇偶校验(LDPC)码的译码算法复杂度和译码性能的均衡, 为了提高译码算法的可靠性和适用性, 在自纠正最小和(SCMS)算法的基础上, 提出了一种动态自纠正最小和(DSCMS)算法. 该算法在迭代译码的过程中, 根据变量节点消息设置阈值, 明确了 SCMS 算法中对消息可靠性的判断, 提高了算法的误码特性和收敛特性. 仿真结果表明, 所提出的 DSCMS 算法的误码性能和收敛性能都要优于 SCMS 算法及其改进算法. 当编码效率为 1/2 时, DSCMS 算法与 SCMS 算法相比, 最多能降低 7.15% 的迭代次数.

**关键词:** 低密度奇偶校验码; 置信传播; 最小和算法; 自纠正最小和算法

中图分类号: TN911.22

文献标志码: A

## A Dynamic Self-Corrected Minimum Sum Decoding Algorithm for LDPC Codes

CHEN Rong<sup>1,2</sup>, CHEN Lan<sup>1</sup>

(1. Institute of Microelectronics, Chinese Academy of Sciences, Beijing 100029, China;

2. School of Electronic, Electrical and Communication Engineering, University of Chinese Academy of Sciences, Beijing 100049, China)

**Abstract:** Aiming at the trade-off between the decoding complexity and the decoding performance of low-density parity check (LDPC) codes, and improving the reliability and applicability of the decoding algorithm, a dynamic self-corrected minimum sum (DSCMS) algorithm is proposed based on the self-corrected minimum sum (SCMS) algorithm. In the process of iterative decoding, the algorithm sets the threshold according to the variable node message, clarifies the judgement of the message reliability in SCMS, and improves the error characteristics and convergence characteristics of the algorithm. Simulation results show that the error performance and convergence performance of DSCMS are better than those of SCMS. When the coding rate is 1/2, the number of iterations of DSCMS can be reduced by up to 7.15% compared with SCMS.

**Key words:** low-density parity-check codes; belief propagation; minimum sum algorithm; self-corrected minimum sum algorithm

差错控制编码是提高信息传输可靠性的一种关键技术, 是大多数现代数字通信系统和数据存储系统不可或缺的重要部分<sup>[1-2]</sup>. 低密度奇偶校验(LD-

PC, low-density parity-check) 码在 1962 年被 Gallager 博士首先提出, 是一类具有很强纠错能力的编码技术<sup>[3]</sup>. 由于 LDPC 码具有描述简单、译码复杂度低、

收稿日期: 2019-10-18

基金项目: 国家科技重大专项项目(2018ZX03001006-002)

作者简介: 陈 容(1991—), 女, 博士生.

通讯作者: 陈 岚(1968—), 女, 研究员, E-mail: chenlan@ime.ac.cn.

可并行实现、使用灵活、误码平层低等优点,在实际系统中得到了广泛的应用,并被第 5 代移动通信系统的新无线电 (NR, new radio) 采纳为增强型移动宽带 (eMBB, enhanced mobile broadband) 场景中数据信道的长码编码方案<sup>[4-5]</sup>.

LDPC 码在最初被 Gallager 提出的时候并没有引起人们的重视,由于当时硬件技术的限制,长码的译码实现起来非常困难. 到了 20 世纪 90 年代, MacKay 和 Neal<sup>[6]</sup>证明了在与置信传播 (BP, belief propagation) 的迭代译码相结合的条件下,LDPC 码具有逼近 Shannon 限的性能,LDPC 码也因此逐渐成为学术界研究的热点. BP 算法的缺点是计算复杂度高,尤其是在消息量化比特较低时性能严重下降. Fossorier<sup>[7]</sup>在此基础上,将校验节点消息计算公式进行简化,用比较运算代替了查找表运算,提出了最小和 (MS, minimum sum) 算法. MS 算法虽然降低了计算复杂度,但译码性能有较大的损失. 后来 Chen 等<sup>[8-9]</sup>提出了偏移最小和 (OMS, offset minimum sum) 算法和归一化最小和 (NMS, normalized minimum sum) 算法,分别通过减法和除法修正校验节点的消息,获得了逼近 BP 算法的译码性能,同时计算复杂度接近最小和算法. 近年来, Savin 等<sup>[10-12]</sup>提出了自纠正最小和 (SCMS, self-corrected minimum sum) 算法及其改进算法,从变量节点消息的角度出发,提高了译码性能,并且加快了译码的收敛性.

SCMS 算法具有准最佳译码性能,但它只是将变量节点消息的符号改变作为判断消息不可靠的依据,没有明确说明如何判断消息的可靠性,并且在信噪比较高的情况下,SCMS 算法的收敛性下降明显. 为了进一步提高 LDPC 码的译码可靠性与收敛性,在 SCMS 算法的基础上,提出了一种动态自纠正最小和 (DSCMS, dynamic self-corrected minimum sum) 译码算法,该算法可以根据信道情况,在迭代译码的过程中,通过改变阈值的大小,动态调整消息可靠性的判断,提高译码性能,加快译码收敛性,同时也提高了系统的灵活性.

## 1 BP 算法

最原始的 BP 算法,也称为概率阈 BP 算法,在变量节点和校验节点之间传递的信息是节点取特定值的概率. 此译码算法采用了很多相乘计算,需要耗费较多译码时间和硬件资源,不容易在工程上实

现. 因此,将算法中的概率运算转换到对数域中,即对数似然比 (LLR, log-likelihood ratio) BP 算法<sup>[13]</sup>. 这样就能够把更多复杂的乘法运算简化成简单的加法运算,在很大程度上降低了 BP 译码算法中迭代更新运算的复杂度. 为了便于后面描述,相关变量符号的定义如下:  $\mathbf{H}$  表示 LDPC 码的校验矩阵,其对应的 Tanner 图中,包含  $N$  个变量节点和  $M$  个校验节点;  $i$  代表  $\mathbf{H}$  中的一个变量节点,  $i \in \{1, 2, \dots, N\}$ ;  $j$  代表  $\mathbf{H}$  中的一个校验节点,  $j \in \{1, 2, \dots, M\}$ ;  $C(i)$  表示与变量节点  $i$  相连的校验节点的集合;  $R(j)$  表示与校验节点  $j$  相连的变量节点的集合;  $C(i) \setminus j$  表示除  $j$  外与变量节点  $i$  相连的校验节点的集合;  $R(j) \setminus i$  表示除  $i$  外与校验节点  $j$  相连的变量节点的集合;  $L(P_i)$  信道初始概率似然比消息;  $L(r_{ji})$  表示校验节点消息 (变量节点  $i$  传向校验节点  $j$  的外部概率似然比消息);  $L(q_{ij})$  表示变量节点消息 (校验节点  $j$  传向变量节点  $i$  的外部概率似然比消息);  $L(q_i)$  表示判决量消息 (变量节点  $i$  收集到的所有消息);  $\hat{c}$  表示译码判决得到的译码序列.

LLR BP 算法的步骤可以概括如下.

### 1) 初始化

根据测试需要,设置最大迭代次数,计算信道传递给变量节点的初始概率似然比消息  $L(P_i)$ ,然后对每一个变量节点  $i$  和与其相邻的校验节点  $j \in C(i)$ ,初始化变量节点消息

$$L^{(0)}(q_{ij}) = L(P_i) \quad (1)$$

将所有校验节点消息的初始值设置为零.

### 2) 迭代处理

#### 步骤 1 校验节点消息处理

对所有的校验节点  $j$  和与其相邻的变量节点  $i \in R(j)$ ,第  $l$  次迭代时,计算校验节点消息

$$L^{(l)}(r_{ji}) = 2 \tanh^{-1} \left\{ \prod_{i' \in R(j) \setminus i} \tanh \left[ \frac{1}{2} L^{(l-1)}(q_{ij'}) \right] \right\} \quad (2)$$

#### 步骤 2 变量节点消息处理

对所有的变量节点  $i$  和与其相邻的校验节点  $j \in C(i)$ ,第  $l$  次迭代时,计算变量节点消息:

$$L^{(l)}(q_{ij}) = L(P_i) + \sum_{j' \in C(i) \setminus j} L^{(l)}(r_{ji'}) \quad (3)$$

#### 步骤 3 译码判决

对所有变量节点计算判决量消息

$$L^{(l)}(q_i) = L(P_i) + \sum_{j \in C(i)} L^{(l)}(r_{ji}) \quad (4)$$

判决规则:若  $L^{(l)}(q_i) > 0$ ,则  $\hat{c}_i = 0$ ,否则  $\hat{c}_i = 1$ .

### 3) 停止

若  $H\hat{c}^T = \mathbf{0}$ , 或者达到最大迭代次数, 则结束运算; 否则从步骤1继续迭代。

## 2 MS 算法及其改进算法

虽然 LLR BP 算法已经在概率域 BP 算法的基础上简化了很多运算, 但是其校验节点的更新计算中含有双曲正切函数及其反函数, 在实现上依然很复杂, 因此 Fossorier<sup>[7]</sup> 对其进行简化, 提出了 MS 算法。

### 2.1 MS 算法

MS 算法的简化原理: 因为  $\tanh x, \tanh^{-1} x$  都是奇函数,  $\tanh |x|$  在  $0 \sim 1$  之间取值, 并且是  $|x|$  的单调递增函数, 故 MS 算法校验节点的更新计算可以表示为

$$L(r_{ji}) = \prod_{i' \in R_j \setminus i} \operatorname{sgn} [L(q_{i'j})] \min_{i' \in R_j \setminus i} [ |L(q_{i'j})| ] \quad (5)$$

MS 算法是 LLR BP 算法的一种近似, 在校验节点的更新中, 使用最小值和次小值代替了要传递的置信度消息。这种近似替代牺牲了一部分性能, 但大大降低了硬件复杂度, 实现过程中, 只有“求和”运算和“比较”运算。

比较式(2)与式(5)可以发现, 近似后的表达式中, 符号与原来一致, 而幅度大于原来的值。基于此, Chen 等<sup>[8]</sup> 提出了 NMS 算法和 OMS 算法, 来修改增加的幅度, 补偿 MS 算法近似处理所损失的性能。

1) NMS 算法。通过除以一个数值来降低消息置信度的幅度, 校验消息表示为

$$L(r_{ji}) \leftarrow \frac{1}{\alpha} L(r_{ji}) \quad (6)$$

其中  $\alpha > 1$  称为校正因子, 此时改进算法称为 Normalized MS 算法。

2) OMS 算法。通过减去一个数值来降低消息置信度的幅度, 校验消息表示为

$$L(r_{ji}) \leftarrow \operatorname{sgn} [L(r_{ji})] \max [ |L(r_{ji})| - \beta, 0 ] \quad (7)$$

其中  $\beta$  称为偏移因子, 此时改进算法称为 Offset MS 算法。校正因子和偏移因子的具体数值可以由密度进化渐近分析工具得到。

### 2.2 SCMS 算法

NMS 和 OMS 算法是在校验节点消息处理上对 MS 高估的消息置信度做出一定的修正。而 Savin 等提出的 SCMS 算法是从变量节点消息的更新出发,

修正消息传递过程中的置信度。SCMS 算法的原理是: 在变量节点消息处理过程中, 根据变量节点消息的符号波动, 有选择的擦除“不可信”的消息, 这样可以加快收敛, 在一定程度上恢复了 MS 算法损失的性能。下面为 SCMS 算法的节点消息在第  $l$  次迭代中的更新过程。

#### 1) 变量节点消息更新

$$L^{(l)}(q_{ij}) = L^{(l-1)}(q_i) - L^{(l-1)}(r_{ji}) \quad (8)$$

$$e_{ij}^{(l)} = [ \sim e_{ij}^{(l-1)} ] [ s_{ij}^{(l)} \oplus s_{ij}^{(l-1)} ] \quad (9)$$

$$L^{(l)}(q_{ij}^{\text{SC}}) = [ e_{ij}^{(l)} = 1 ] ? 0 : L^{(l)}(q_{ij}) \quad (10)$$

#### 2) 校验节点消息更新

$$L^{(l)}(r_{ji}) = \prod_{i' \in R_j \setminus i} s_{i'j}^{(l)} \min_{i' \in R_j \setminus i} [ |L^{(l)}(q_{i'j}^{\text{SC}})| ] \quad (11)$$

#### 3) 判决量更新

$$L^{(l)}(q_i) = L(P_i) + \sum_{j \in C_i} L^{(l)}(r_{ji}) \quad (12)$$

其中: 符号“ $\sim$ ”为取反; 符号“ $\oplus$ ”为异或; 符号“ $?:$ ”为条件运算符;  $e_{ij}^{(l)}$  用来指示本次迭代中  $L^{(l)}(q_{ij})$  被擦除的位置, 避免相邻 2 次迭代重复擦除同一位置的消息;  $s_{ij}^{(l)}$  为本次迭代中  $L^{(l)}(q_{ij})$  的符号;  $L^{(l)}(q_{ij}^{\text{SC}})$  为根据擦除规则得到的新的变量节点消息。

SCMS 算法沿用了 MS 算法在校验节点消息处的更新, 而在变量节点消息的更新上, 增加了一个可靠度的判断, 如果变量节点消息的符号在连续 2 次迭代中发生改变, 则视为不可靠消息, 将其置为 0, 等待下一次更新。

在传统 SCMS 算法的基础上, 针对如何更准确地判断出哪些变量节点的信息应该被擦除的问题, 约束的 SCMS (CSCMS, constrained SCMS) 算法<sup>[12]</sup> 中增加了一个判断条件:

$$L^{(l)}(q_{ij}^{\text{CSC}}) = L^{(l)}(q_{ij}) I[ L^{(l)}(q_{ij}), L^{(l-1)}(q_{ij}) ] \quad (13)$$

若  $s_{ij}^{(l)} = s_{ij}^{(l-1)}$ , 或者  $|L^{(l-1)}(q_{ij})| < |L^{(l)}(q_{ij})|$ ,  $I[ L^{(l)}(q_{ij}), L^{(l-1)}(q_{ij}) ] = 1$ ; 否则为 0。CSCMS 算法与 SCMS 算法不同之处在于: 当变量节点消息符号发生反转时, 需要考虑  $|L^{(l)}(q_{ij})|$  的大小。

## 3 动态自纠正最小和算法

SCMS 算法阻止了不可靠信息的传递扩散, 提高了算法的性能, 但并没有明确说明如何判断消息的可靠性, 并且在信噪比较高的情况下, SCMS 算法的收敛性有所下降。基于 SCMS 算法, 提出的 DSC-

MS 算法在判断变量节点消息的可靠性时,增加了阈值,动态调整擦除消息的范围区间,因此能够更好地适应不同的环境. 另外, DSCMS 算法在校验节点更新上采用 NMS 算法,从校验节点处理上保证译码算法的性能. DSCMS 算法在变量节点消息更新过程中遵循的规则:如果变量节点消息向相反符号方向变动大于设定的阈值时,则判定为不可靠消息,并在更新中置为 0,如图 1 所示.

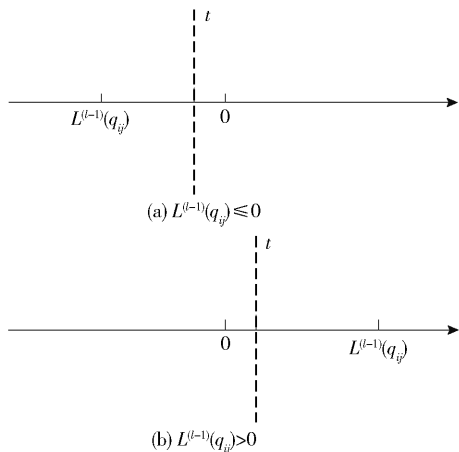


图 1 DSCMS 算法可靠消息判定示意图

图 1 中,  $L^{(l-1)}(q_{ij})$  为上一次迭代中的变量节点消息,  $t$  为所设的阈值. 若  $L^{(l-1)}(q_{ij})$  为负值,那么当  $L^{(l)}(q_{ij})$  落在  $t$  的右侧,则会被擦除;若  $L^{(l-1)}(q_{ij})$  为正值,那么当  $L^{(l)}(q_{ij})$  落在  $t$  的左侧,则会被擦除.

阈值设定的考虑因素:如果  $t$  偏离  $L^{(l-1)}(q_{ij})$  较远,则只有较少数量的变量节点消息被修改,性能提升有限;另一方面,如果  $t$  偏离  $L^{(l-1)}(q_{ij})$  较近,则会有大量的变量节点消息被修改,扩大了不可靠消息的范围,引起译码性能下降. 在这里,将阈值  $t$  定义为

$$t = \theta L^{(l-1)}(q_{ij}) \quad (14)$$

其中:  $\theta$  为调节因子,其变化范围可取  $-0.5 \leq \theta \leq 0.5$ ,这样得到的  $t$  值不会偏离  $L^{(l-1)}(q_{ij})$  太近或太远. 同时,  $t$  是一个关于  $L^{(l-1)}(q_{ij})$  的相对值,而  $L^{(l-1)}(q_{ij})$  在每次迭代中是变化的,  $t$  也会随着迭代更新而改变,因此判断可靠性的标准是自适应更新的.

DSCMS 算法的消息更新过程描述如下:

#### 1) 变量节点消息更新

$$L^{(l)}(q_{ij}) = L^{(l-1)}(q_i) - L^{(l-1)}(r_{ji}) \quad (15)$$

$$e_{ij}^{(l)} = [\sim e_{ij}^{(l-1)}] \{s_{ij}^{(l-1)} [L^{(l)}(q_{ij}) - t] < 0\} \quad (16)$$

$$L^{(l)}(q_{ij}^{\text{DSC}}) = [e_{ij}^{(l)} = 1] ? 0 : L^{(l)}(q_{ij}) \quad (17)$$

#### 2) 校验节点消息更新

$$L^{(l)}(r_{ji}) = \prod_{i' \in R_j \setminus i} s_{i'j}^{(l)} \min_{i' \in R_j \setminus i} [|L^{(l)}(q_{i'j}^{\text{DSC}})|] \frac{1}{\alpha} \quad (18)$$

#### 3) 判决量更新

$$L^{(l)}(q_i) = L(P_i) + \sum_{j \in C_i} L^{(l)}(r_{ji}) \quad (19)$$

其中:  $\alpha$  为 NMS 算法中的校正因子,  $L^{(l)}(q_{ij}^{\text{DSC}})$  为根据新的擦除规则得到的变量节点更新消息. 可以看出,当阈值  $t = 0$  时, DSCMS 算法即退变为传统的 SCMS 算法.

## 4 仿真结果与比较

### 4.1 DSCMS 算法的性能仿真

仿真中选择 5G NR 标准中的 LDPC 码,针对码率为 1/2、码长为 4 224 bit 的情况<sup>[14]</sup>,在 Matlab 平台中模拟加性高斯白噪声 (AWGN, additive white Gaussian noise) 信道,编码比特采用二进制相移键控 (BPSK, binary phase shift keying) 调制方式进行传输,基本仿真参数如表 1 所示.

表 1 仿真参数

参数	取值
码长/bit	4 224
编码块数	1 000
最大迭代次数	30
校正因子 $\alpha$	1.2
调节因子 $\theta$	-0.2

在文献[10,12]中有其他算法的对比,在此只对 SCMS、CSCMS、DSCMS 算法进行仿真测试与比较,误比特率 (BER, bit error ratio)、误块率 (BLER, block error ratio) 及平均迭代次数结果如图 2 所示.

从图 2 可以看出, DSCMS 算法在误码特性和收敛特性上都优于 SCMS 算法. 而 CSCMS 算法在低信噪比 (SNR, signal noise ratio) 区域表现不佳,在 SNR 较高时,性能要优于 SCMS 算法. 取图 2(c) 中 SCMS、CSCMS 和 DSCMS 算法在不同信噪比下的平均迭代次数统计如表 2 所示.

表 2 不同信噪比下各算法的平均迭代次数

算法	SNR/dB				
	1.1	1.2	1.3	1.4	1.5
SCMS	25.677	23.257	21.323	19.341	17.409
CSCMS	26.026	23.808	21.151	18.997	16.927
DSCMS	24.381	22.133	20.097	17.959	16.195



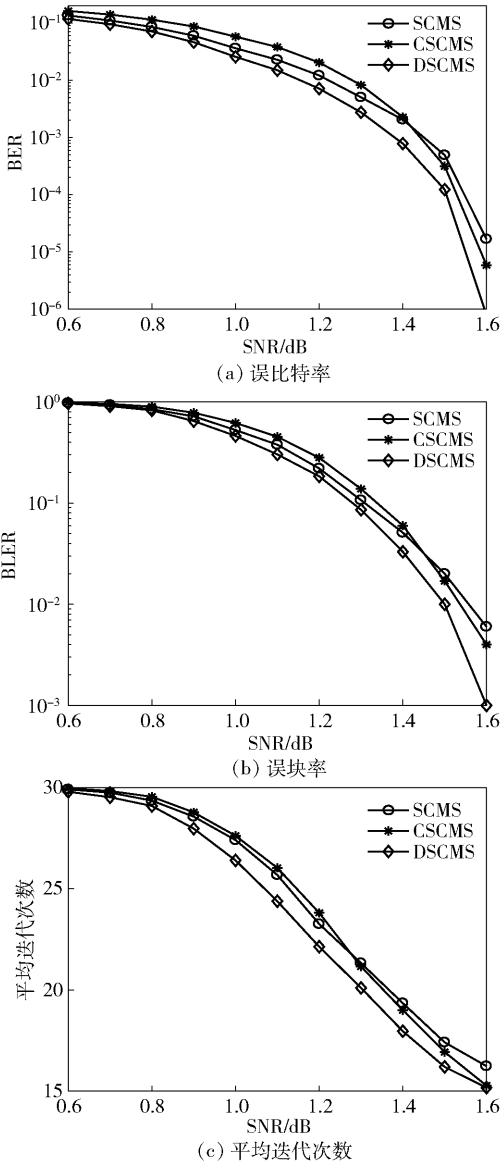


图2  $R = 1/2$  时, SCMS、CSCMS、DSCMS 算法的仿真曲线

从表 2 可以看出,随着信道 SNR 的增大,LDPC 译码的迭代次数在逐渐降低,DSCMS 算法的收敛特性要优于 SCMS 算法,在高 SNR 区域与 CSCMS 算法相近. 这是因为 DSCMS 与 CSCMS 算法都能够抑制不可靠消息的传递,加快了译码的收敛速度. 而 DSCMS 算法对擦除消息的动态判断能够适应更广泛的信道环境. 通过计算可得,在收敛特性上,DSCMS 算法较 SCMS 算法最多能降低 7.15% 的迭代次数,平均降低 5.95%.

4.2 译码算法的复杂度分析

表 3 示出了 BP、MS、NMS、SCMS、CSCMS、DSCMS 算法单次迭代的计算复杂度,其中  $N$  表示编码

长度, $K$  表示有效信息长度, $W$  为校验矩阵的列重.

表 3 译码算法计算量比较

算法	乘法	除法	加法
BP	$11NW - 6(N + K)$	$N(W + 1)$	$N(3W + 1)$
MS	0	0	$N(4W - 1) + K(\text{lb}W - 2)$
NMS	0	$NW$	$N(4W - 1) + K(\text{lb}W - 2)$
SCMS	0	0	$N(4W - 1) + K(\text{lb}W - 2)$
CSCMS	0	0	$N(5W - 1) + K(\text{lb}W - 2)$
DSCMS	$NW$	$NW$	$N(5W - 1) + K(\text{lb}W - 2)$

以 MS 算法为基础的改进算法,在 BP 算法的基础上,减少了大量的乘法和除法运算,在实现中会大大降低计算的复杂度. 而 DSCMS 算法在变量节点消息处理中增加了少量的乘法和除法运算,在实现时,可将阈值调节因子定为  $1/2$ 、 $1/4$ 、 $1/8$  等,转化为简单的移位和加法运算得到相应的结果<sup>[15]</sup>. 总体上来说,DSCMS 算法在增加少量计算复杂度的情况下,提高了译码算法的性能,降低了迭代次数.

5 结束语

在 SCMS 算法的基础上提出了一种动态自纠正最小和算法,根据变量节点消息设置阈值,来判断消息是否可靠,并对不可靠消息进行擦除,等待下一次迭代更新. 阈值设定为变量节点消息的相对值,会随着迭代的更新而改变,因此判断可靠性的标准是自适应更新的. 仿真结果表明,在增加少量计算复杂度的情况下,DSCMS 算法在一定 SNR 范围内都表现出了较好的性能,能够有效提升系统的误码特性与收敛特性,并且提高了系统的灵活性.

参考文献:

[1] Jung Y, Jung Y, Lee S, et al. Low-complexity multi-way and reconfigurable cyclic shift network of QC-LDPC decoder for Wi-Fi/WIMAX applications[J]. IEEE Transactions on Consumer Electronics, 2013, 59(3): 467-475.

[2] Zhang Yishan, Zhang Chun, Yan Zhiyuan, et al. A high-through put multi-rate LDPC decoder for error correction of solid-state drives[C] // 2015 IEEE Workshop on Signal Processing Systems( SiPS). Hangzhou: IEEE, 2015: 1-6.

[3] Gallager R G. Low-density parity-check code[J]. IRE Transactions on Information Theory, 1962, 8(1): 21-28.

[4] Richardson T, Kudekar S. Design of low-density parity

- check codes for 5G new radio[J]. IEEE Communications Magazine, 2018, 56(3): 28-34.
- [5] 张睿, 朱敏, 张冀, 等. 面向 5G 的递增冗余 HARQ 传输方案研究[J]. 北京邮电大学学报, 2018, 41(5): 92-97.
- Zhang Rui, Zhu Min, Zhang Ji, et al. Study on 5G incremental redundancy HARQ transmission strategy[J]. Journal of Beijing University of Posts and Telecommunications, 2018, 41(5): 92-97.
- [6] MacKay D J C, Neal R M. Near Shannon limit performance of low density parity check codes[J]. Electronics Letters, 1996, 32(18): 1645-1646.
- [7] 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.
- [8] Chen J, Fossorier M P C. Density evolution for two improved BP-based decoding algorithms of LDPC codes[J]. IEEE Communications Letters, 2002, 6(5): 208-210.
- [9] Chen J, Fossorier M P C. Near optimum universal belief propagation based decoding of low-density parity check codes[J]. IEEE Transactions on Communications, 2002, 50(3): 406-414.
- [10] Savin V. Self-corrected min-sum decoding of LDPC codes[C]//2008 IEEE International Symposium on Information Theory. Toronto: IEEE, 2008: 146-150.
- [11] Boncalo O, Amaricai A, Mihancea P F, et al. Memory trade-offs in layered self-corrected min-sum LDPC decoders[J]. Analog Integrated Circuits and Signal Processing, 2016, 87(2): 169-180.
- [12] 韩少聪, 高飞飞, 李云洲, 等. 一种改进的自纠正最小和 LDPC 码的译码算法[J]. 电信科学, 2013(6): 95-99.
- Han Shaocong, Gao Feifei, Li Yuzhou, et al. An improved self-correct min-sum decoding algorithm for low-density parity-check code[J]. Telecommunications Science, 2013(6): 95-99.
- [13] 袁东风, 张海刚, 马丕明, 等. LDPC 码理论与应用[M]. 北京: 人民邮电出版社, 2008: 72-78.
- [14] 3GPP TSG RAN WG1. Multiplexing and channel coding (Release 15): 3GPP TS 38.212 V15.1.1-2018 [S]. Valbonne: 3GPP, 2018: 28.
- [15] Xiang Bo, Shen Rui, Pan An, et al. An area-efficient and low-power multirate decoder for quasi-cyclic low-density parity-check codes[J]. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2009, 18(10): 1447-1460.