A Novel Hardware Implementation Mechanism for AR4JA Codes in Deep Space Communication

Ming Li, Student Member IEEE; Mingchuan Yang*, Member IEEE; Xingqi Zhang; Qing Guo, Member IEEE

Communication Research Center
Harbin Institute of Technology
Harbin, Heilongjiang Province, 150001, China

Abstract– **Because of its parity check matrix has good systematicness, the minimum distance between codes has linear relationship with code length, AR4JA (accumulate-repeat-4-jagged-accumulate) codes is thought to be one of the most suitable error control channel codes for deep space reliable communication in the future. In order to reduce the modified min-sum algorithm decoder hardware implementation complexity of AR4JA codes, a method of constraint about clipping threshold is proposed in the paper, based on the analysis of the effects of modify factor, clipping threshold and input quantization on the performance. The simulation results show that when the modify factor A is 0.75, received signal clipping threshold c_{th} is 4, and use 8Q5 quantization, the decoding performance of the modified min-sum algorithm for AR4JA codes is close to the theoretical value. Meanwhile, in this condition it can reduce the hardware implementation complexity for just need addition, comparison and shift operation, which is beneficial to realize miniaturization of deep space communication receiver.**

***Index Terms*– deep space communication; AR4JA codes; modified min-sum algorithm; clipping threshold**

I. INTRODUCTION

In deep-space communications, the signal-to-noise ratio (SNR) of the communication link is the most severe limitation. The use of efficient error correction coding technology is an important method to solve the problem. In 2007, CCSDS (Consultative Committee for Space Data System) issued a coding standard about LDPC codes, which gives three rates, three code lengths of AR4JA codes family[1]. AR4JA Codes that belong to LDPC Codes family are considered as one of the most powerful codes known today and it can even outperform Turbo Codes for deep space communication. Because of the load limits of deep space communication, in the decoding of AR4JA Codes, huge data processing, storage, and interconnect requirements is a real challenge for decoder realization. To realize miniaturization of deep space communication receiver, for decoder hardware implementation, reduced complexity might be a key implementation issue [2]. LDPC decoding algorithms include probabilistic domain BP (Belief Propagation) decoding algorithm, log-domain BP decoding algorithm [3], the min-sum decoding algorithm [4], and the modified min-sum decoding algorithm [5]. Among these decoding algorithms, the hardware implementation complexity of modified min-sum decoding algorithm is lowest, and does not require channel estimation. Some effects of clipping and quantization on implementation of min-sum algorithm and its modifications for decoding (3, 6)-regular (8000, 4000) LDPC code is studied in [6]. Paper [7] proposes a kind of decoding design for the (2048, 1723) Reed-Solomon based LDPC (RS-LDPC) code and the (2209, 1978) array-based LDPC code, which can improve low error rate performance applying the suitable quantization and algorithm choices. In [8], it investigates the effect of the number of iteration and the decoder's input quantization to achieve a significant improvement in the hardware implementation of the decoder architecture about LDPC code, which is recommended in DVB (Digital Video Broadcasting)-S2 standards. But there is no correlative research about AR4JA codes on implementation of decoding algorithm. In this paper, the influence of key parameters on performance of modified min-sum decoding algorithm for AR4JA codes is mainly considered.

*Corresponding author: Mingchuan Yang; Email: mcyang@hit.edu.cn

II. INFLUENCE OF KEY PARAMETERS ON MODIFIED MIN-SUM DECODING ALGORITHM DECODING PERFORMANCE

A Modified min-sum decoding algorithm

AR4JA code is defined by a sparse $M\times N$ parity check matrix $\boldsymbol{H}$ where N represents the number of bits in the code block and M represents the number of parity checks. The $\boldsymbol{H}$ matrix of an AR4JA code can be illustrated graphically using a factor graph, where each bit is represented by a variable node and each check is represented by a factor (check) node. An edge exists between the variable node j and the check node i if and only if $\boldsymbol{H}(i, j) = 1$. Modified min-sum decoding algorithm operates on a factor graph, where soft messages are exchanged between variable nodes and check nodes (as in Fig.1). The algorithm is described below.

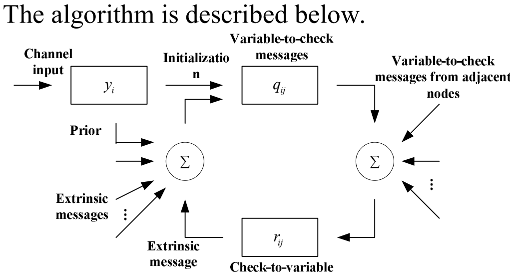

Fig.1 Modified min-sum decoding (one processing unit)

Step 1: Initialization.

$$p_i^0 = y_i \tag{1}$$

$$q_{ij}^0 = y_i \tag{2}$$

Step 2: Check node update.

$$r_{ij}^k = A\times\Big(\prod_{i'\in C(j)\setminus i} a_{i'j}^k\Big)\min_{i'\in C(j)\setminus i}\left(\beta_{i'j}^k\right) \tag{3}$$

Where $\alpha_{i'j}^{k} = sign(q_{i'j}^{k})$ and $\beta_{i'j}^{k} = |q_{i'j}^{k}|$.

Step 3: Variable node update.

$$q_{ij}^{k} = p_i^0 + \sum_{j' \in V(i) \backslash j} r_{ij'}^{k} \tag{4}$$

Step 4: Decision decoding.

$$q_i^k = p_i^0 + \sum_{j \in V(i)} r_{ij}^{k} \tag{5}$$

$$\begin{cases} q_i^k \geq 0, \hat{x}_n = 0 \\ q_i^k < 0, \hat{x}_n = 1 \end{cases} \tag{6}$$

If $H\hat{x}^T = 0$ or it reaches the maximum number of iterations max, the operation reaches the end; if not, it continues the iteration process from step 2.

Where, $V(i)$ is the set of check nodes which are connected with variable node *i*. $C(j)$ is the set of variable nodes which are connected with check node *j*. *A* is the modify factor. r_{ij}^{k} is the external check information that the variable node *j* passes to check node *i* at the *k* times of node of decoding iteration. q_{ij}^{k} is the external check information that the check node *i* passes to variable node *j* at the *k* times of node of decoding iteration. p_i^k is all the external check information that the variable node *i* receives. And max is the maximum number of iterations.

B Influence of modify factor A

The formula (3) is check node update. It is taking a step forward to improve the formula (7) in log-domain BP decoding algorithm [3].

$$r_{ij}^{k} = \left(\prod_{i' \in C(j) \backslash i} a_{i'j}^{k}\right) \Phi\left(\sum_{i' \in C(j) \backslash i} \Phi(\beta_{i'j}^{k})\right) \tag{7}$$

Where, $\Phi(x) = -\log(\tanh(x/2)) = \log \frac{e^x + 1}{e^x - 1}$.

So it can have that

$$A \times \underset{i' \in C(j) \backslash i}{\text{Min}} (\beta_{i'j}^{k}) \approx \Phi\left(\sum_{i' \in C(j) \backslash i} \Phi(\beta_{i'j}^{k})\right) \tag{8}$$

It can be seen that modify factor *A* decides the convergence rate of iterative, namely how many decoding iterations it needs to complete one decoding. The fewer number of iterations is, the faster iteration converges. Modify factor *A* has relation with row quantity of AR4JA codes check matrix ***H***. The possible value range of *A* is 0.6~0.9 [5], and the exact value should be determined by simulation.

C Influence of clipping and quantization

During the process of designing AR4JA codes, performance is usually measured via floating-point simulation. The floating-point simulation gives better results and requires less time to encode in software. However, when it comes to hardware implementation, floating-point representation is not practical and sometimes even unrealizable. Therefore, fixed-point representation is used instead of floating-point representation. Because hardware implementation deals with data in finite precision rather than infinite precision and the decoding of LDPC is an iterative process which needs lots of messages to be processed and calculated at each iteration. So the huge data processing and storage requirements are challenges for decoder hardware implementation. Trade off between hardware complexity and decoding ability and the process of deciding the accuracy of the finite precision representation is always the most efficient implementation. So setting the parameters of fixed point simulation is also a trade off process. These parameters can be selected earlier in the stage of software simulation using fixed-point approach instead of being selected in a late stage of hardware simulation[6]. All LDPC decoding algorithms are usually specified in floating-point domain. Fixed-point number representation, which implies transformation from floating-point to fixed-point, should be used to get an efficient implementation. The main goal of this quantization for hardware implementation is to find optimized parameters for clipping threshold and input quantization with an acceptable degradation of coding performance. During the process of hardware implementation, the reduction of data-path bit-widths, control complexity and memory size results in decrease of area and power consumption and increase of speed. Input data quantization and inner data quantization mainly influence the control complexity and determine bit-width of data-path and memory size directly. Besides, decoding performance is also influenced by quantization. Therefore, optimized quantization is essential for implementation complexity. Different quantization methods have different influence on decoding performance [7]. Here influence of fixed-point-number uniform quantization on decoding is mainly discussed. *Q* form is usually used to represent fixed point number. *Q* form is as follows:

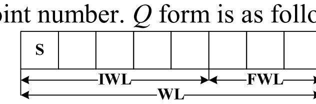

Fig.2. *Q* form of fixed point number

WL is word length of fixed point number. *IWL* is word length of integer part. *FWL* is word length of decimal part. With fixed word length, the larger *Q* value is, the higher precision fixed point number has, while dynamic range will be smaller. Precision and value range of the above fixed point number are:

$$delta = 2^{-FWL} \tag{9}$$

$$range_{signed} = [-2^{IWL-1} + delta, 2^{IWL-1} - delta] \tag{10}$$

Where, *delta* is the precision of the fixed point number, $range_{signed}$ is the value range of the fixed point number.

III. A METHOD OF CONSTRAINT ABOUT CLIPPING THRESHOLD

For received signal, its quantization performance is determined by quantization range and quantization bit number *WL*. Set sending signal as c_i (i=1, 2…n, $c_i \in \{0,1\}$) and adopt BPSK modulation ($0 \rightarrow 1$, $1 \rightarrow -1$). The modulated signal is x_i (i=1, 2…n, $x_i \in \{-1,1\}$) and supposed that it is sent with equal probability. That is $p(x_i=1)=1/2$, $p(x_i=-1)=1/2$. So the probability density of sending signal is shown below.

$$f(x_i) = \frac{1}{2}\delta(x_i - 1) + \frac{1}{2}\delta(x_i + 1) \tag{11}$$

At the same time, received signal is set as y_i (i=1, 2…n, $y_i \in R$), supposed that the channel is Gaussian channel, the channel noise n_i obeys to Gaussian distribution $N(0, \sigma^2)$, the probability density of noise n_i is

$$f(n_i) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{n_i^2}{2\sigma^2}} \quad (12)$$

The received signal $y_i=x_i+n_i$, and x_i and n_i are independent respectively, so y_i obeys to Gaussian mixture distribution. The probability density of received signal y_i is

$$f(y_i) = \frac{1}{2\sqrt{2\pi}\sigma} e^{-\frac{(y_i-1)^2}{2\sigma^2}} + \frac{1}{2\sqrt{2\pi}\sigma} e^{-\frac{(y_i+1)^2}{2\sigma^2}} \quad (13)$$

Consider the received values to be clipped symmetrically at a threshold c_{th}, and then uniformly quantized in the range $[-c_{th}, c_{th}]$. If suitable c_{th} that satisfies the following constraints is chosen, the received signal $y_i \in R$ can be changed to $y_i \in [-c_{th}, c_{th}]$.

$$\int_{-\infty}^{-c_{th}} f(y_i)dy_i \approx 0 \quad (14)$$

$$\int_{-c_{th}}^{+c_{th}} f(y_i)dy_i \approx 1 \quad (15)$$

$$\int_{c_{th}}^{+\infty} f(y_i)dy_i \approx 0 \quad (16)$$

If received $y_i>c_{th}$ or $y_i<-c_{th}$, it is assumed that right now y_i exceeds the threshold. Let $y_i=c_{th}$ or $y_i=-c_{th}$. So if the WL and FWL of Q form are defined, then $c_{th}=2^{IWL-1}-2^{-FWL}$. Therefore, after quantization, in order to let the decoding performance close to the theoretical value, the suitable c_{th}, WL and FWL need to be chosen.

IV. SIMULATIONS AND DISCUSSIONS

To determine the optimum value of modify factor, a series of simulation (A=0.6, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9) is done on AR4JA code (assume channel is Gaussian channel). The results are shown in Fig.3. And the other simulation parameters can be seen in Table 1.

Table 1 AR4JA code simulation parameters in Gaussian channel

Information bits length	Code rate	Code length	Modulation	Decoding algorithm
1024	1/2	2048	BPSK	modified min-sum

It can be seen that in Fig.3, when modify factor A is 0.75, decoding performance is the best among 0.6~0.9. Moreover, when modify factor A is 0.75, it is very easy for hardware implemetation and it can reduce the hardware implementation complexity, because only shift operation is needed.

Received signal probability density of AR4JA code in Gaussian channel with different SNR from 0dB to 3dB (interval 0.5 dB) is shown in Fig.4. It can be seen in Fig.4 that received signal range is determined by SNR. In order to find the suitable clipping threshold c_{th} to improve the quantization precision, according to the proposed constraint formula (14), (15), and (16), quantization range can be taken as (−4, 4), that is c_{th} =4. If quantization length WL is 6, in order to satisfy c_{th} =4, then $Q_{\max}$ is 3. Similarly, WL=7, then $Q_{\max}$ =4; WL=8, then $Q_{\max}$ =5.

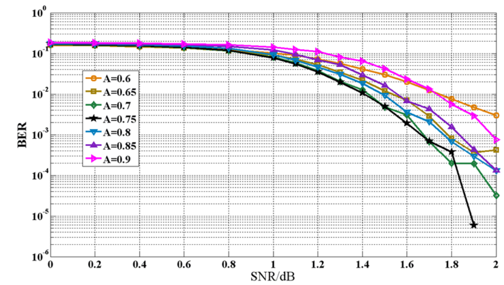

Fig.3. AR4JA code decoding performance in Gaussian channel with different modify factor A

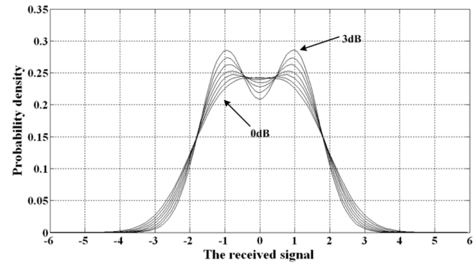

Fig.4. Received signal probability density of AR4JA code with different SNR from 0dB to 3dB (interval 0.5 dB)

During decoding, there will be some intermediate variable stored. Probability density of r_{ij}^k , q_{ij}^k and q_i^k with different SNR from 0dB to 3dB (interval 0.5 dB) are presented in Fig.5, Fig.6 and Fig.7. And the other simulation parameters can be seen in Table 1. It can be seen that in Fig.5, more than 99% intermediate variable r_{ij}^k lies among (−4, 4), thus its quantization range is the same with received signal. From Fig.6 and Fig.7, more than 99% intermediate variables q_{ij}^k and q_i^k lie among (−12, 12), but received signal quantization range is just (−4, 4). So to satisfy of non-overflow of intermediate variable during hardware calculation process, quantization bits of q_{ij}^k and q_i^k should increase. When WL=6, $WL_{\min}$ of q_{ij}^k and q_i^k should be $WL_{\min}$ =8; When WL=7, $WL_{\min}$ =9; When WL=8, $WL_{\min}$ =10. In this way, non-overflow of q_{ij}^k and q_i^k calculated results can be guaranteed. And so will decoding performance of decoder be guaranteed.

Quantization precision is a key influence factor on decoding hardware implementation. Performances of AR4JA codes are discussed below (A=0.75, clipping threshold (−4, 4), quantization bits 6Q3, 7Q4 and 8Q5 respectively). The simulated results of AR4JA code decoding performance in Gaussian channel with different WL and FWL are shown in Fig.8. From Fig.8, it can be seen that when threshold of y_i is c_{th}=4, decoding performance of 8Q5 is obviously superior to performances of 6Q3 and 7Q4. Because when IWL is fixed, the greater quantization bit is, the bigger Q is, and the higher quantization precision is. Quantization precision of 8Q5 reaches 10^{-5}, while for 7Q4 quantization precision is only 10^{-4} and for 6Q3 quantization precision is only 10^{-3}.

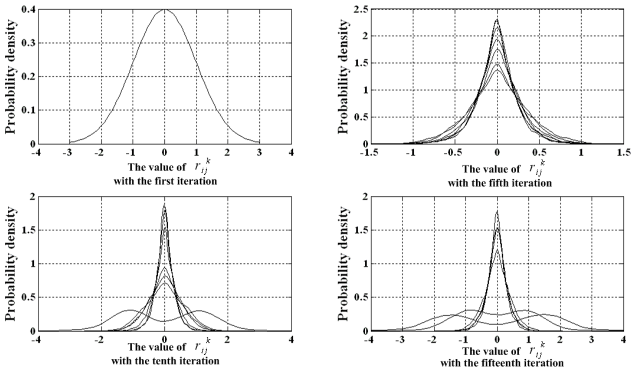

Fig.5. Probability density of intermediate variable r_{ij}^k of AR4JA code with different SNR from 0dB to 3dB (interval 0.5 dB)

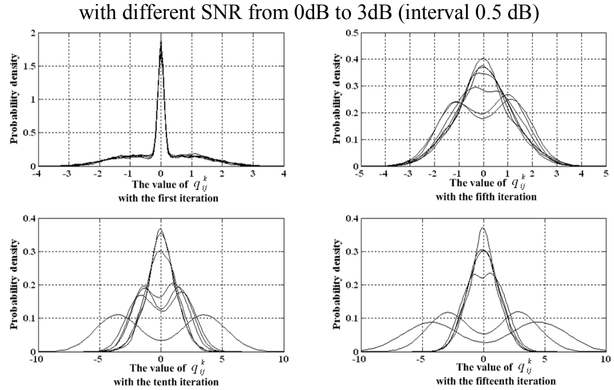

Fig.6. Probability density of intermediate variable q_{ij}^k of AR4JA code with different SNR from 0dB to 3dB (interval 0.5 dB)

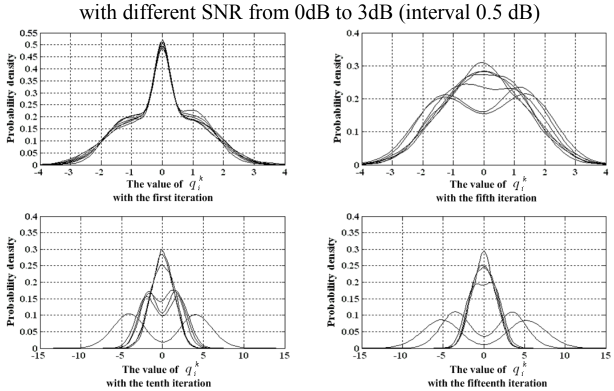

Fig.7. Probability density of intermediate variable q_i^k of AR4JA code with different SNR from 0dB to 3dB (interval 0.5 dB)

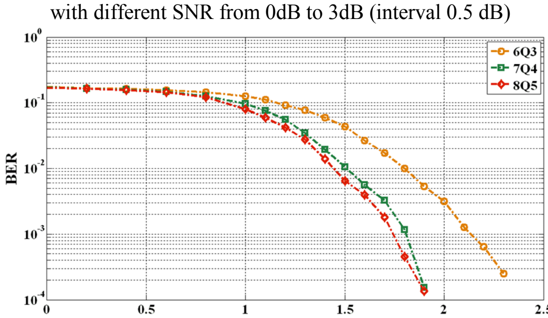

Fig.8. AR4JA code decoding performance in Gaussian channel with different *WL* and *FWL*

V. CONCLUSION

The parity check matrix of AR4JA codes has good systematicness, the minimum distance between codes has linear relationship with code length. So AR4JA codes is thought to be one of the most suitable error control channel codes and even outperform Turbo Codes for deep space reliable communication in the future. This paper studies the modified min-sum decoding algorithm. Its performance is closer to log-domain BP decoding algorithm and its hardware implementation complexity is also low. In order to reduce the decoder hardware implementation complexity of AR4JA codes, based on the analysis of the effects of modify factor, clipping threshold and input quantization on the performance of the modified min-sum decoding algorithm for AR4JA codes, a method of constraint about clipping threshold is proposed. The simulation results show that when the modify factor A is 0.75, received signal clipping threshold c_{th} is 4, and use 8Q5 quantization, then the decoding performance of the modified min-sum algorithm for AR4JA codes is close to the theoretical value. Meanwhile, in this condition it can reduce the hardware implementation complexity for just need addition, comparison and shift operation, which is beneficial to realize miniaturization of deep space communication receiver. But only information bits length=1024, code rate=1/2 of AR4JA code is considered in this paper. Whether the conclusion is applicable for length=4096 or 16384, code rate=2/3 or 4/5 of AR4JA codes or not, attention will be paid to it in the next research direction.

ACKNOWLEDGMENT

The paper is sponsored by “National Natural Science Foundation of China” (No. 61032003), and “the Fundamental Research Funds for the Central Universities”(Grant No.HIT.NSRIF.2012021).

REFERENCES

[1] CCSDS 131.1-O-2.2007. Low Density Parity Cheek Codes for Use in Near-Earth and Deep Space Applications. Washington, DC, USA, CCSDS, 2007.

[2] ZHANG Nai-tong, LI Hui, ZHANG Qin-yu. Thought and Developing Trend in Deep Space Exploration and Communication [J]. Journal of Astronautics, 2007, 28(4):786-793.

[3] R. G. Gallager. Low-density parity-check codes. IEEE Trans. Inform. Theory, Jan.1962, 8:21-28.

[4] M. P. C. Fossorier, M. Mihaljevic, and H. Imai. Reduced complexity iterative decoding of low-density parity check codes based on belief propagation [J].IEEE Trans. Commun. 1999, 47:673-680.

[5] Jinghu Chen, Fossorier, M.P.C. Near optimum universal belief propagation based decoding of LDPC codes, IEEE Trans. on Comm, 2002, 50(3): 406-414.

[6] Jianguang Zhao, Zarkeshvari F, Banihashemi, A.H. On implementation of min-sum algorithm and its modifications for decoding low-density Parity-check (LDPC) codes. IEEE Trans on Comm, 2005, 53(4): 549-554.

[7] Zhengya Zhang, Dolecek, L, Nikolic, B, Anantharam, V, Wainwright, M. Design of LDPC Decoders for Improved Low Error Rate Performance: Quantization and Algorithm Choices. IEEE Trans on Comm, 2009, 57(11): 3258-3268.

[8] Shaker S.W. DVB-S2 LDPC Finite-Precision Decoder. 2011 13th International Conference on Advanced Communication Technology (ICACT). 2011:1383 – 1386.