

RESEARCH

Open Access



A state metrics compressed decoding technique for energy-efficient turbo decoder

Ming Zhan¹, Zhibo Pang^{2*} , Ming Xiao³ and Hong Wen⁴

Abstract

In the energy resource-constrained wireless applications, turbo codes are frequently employed to guarantee reliable data communication. To both reduce the power dissipation of the turbo decoder and the probability of data frame retransmission in the physical layer, memory capacity reduced near optimal turbo decoder is of special importance from the perspective of practical implementation. In this regard, a state metrics compressed decoding technique is proposed. By inserting two modules in the conventional turbo decoding architecture, a smaller quantization scheme can be applied to the compressed state metrics. Furthermore, structure of the inserted modules is described in detail. We demonstrate that one or two rounds of compression/decompression are performed in most cases during the iterative decoding process. At the cost of limited dummy decoding complexity, the state metrics cache (SMC) capacity is reduced by 53.75%. Although the proposed technique is a lossy compression strategy, the introduced errors only have tiny negative influence on the decoding performance as compared with the optimal Log-MAP algorithm.

Keywords: Energy efficiency, Turbo decoder, State metrics, SMC capacity, Compression, Decompression

1 Introduction

In recent years, turbo codes have been adopted as the channel coding scheme by some advanced communication standards [1, 2]. To improve the reliability in wireless data transmission, this family of error correction code also steps into some energy resource-constrained wireless applications [3]. In some scenarios, such as wireless sensor networks (WSNs), the sensor nodes only have limited weight of batteries, almost 80% of the overall power dissipation in the nodes is accounted for wireless data communication, and lifetime of the sensor nodes is dominated by the power dissipation of the communication module [4–6]. To reduce the transmission power and to decrease the number of data frame retransmission in the sensor nodes as much as possible, the research of turbo decoder with low power consumption and near optimal bit error rate (BER) is a key topic. However, in the engineering implementation of turbo code decoder, the maximum *a posteriori* (MAP) decoding algorithm is iteratively performed; the decoder requires large capacity memory and frequent memory accessing, which lead to high power

dissipation of the turbo decoder [7]. Therefore, conventional turbo decoder is not suitable for power resource limited WSNs scenarios. Consequently, the energy issue of turbo decoder has become a bottleneck constraint that should be seriously concerned.

To address this deficiency of the conventional turbo decoder, researchers have proposed different decoding architectures. These techniques include stopping the iteration under certain criteria [8], replacing the memory accessing with reverse calculation [9], and recently reducing the memory capacity of state metrics cache (SMC). Among these techniques, the memory-reduced decoding scheme decrease the overall power dissipation by a larger margin. Moreover, since a lower memory capacity is required, this technique is effective for the design of turbo decoder with smaller chip area. According to this strategy, the traceback decoding scheme stores different metrics and sign bits in the SMC; the SMC size is 20% reduced [10]. In [11], the Walsh-Hadamard transform is introduced to represent the forward state metrics with a smaller word width of the SMC memory. Cost of this simplification is the increased dummy decoding complexity. In order to further reduce the SMC capacity and maintain a low dummy decoding complexity, this paper proposes

*Correspondence: pang.zhibo@se.abb.com

²ABB Corporate Research, Forskargränd 7, SE-721 78 Västerås, Sweden
Full list of author information is available at the end of the article

to insert two modules in the turbo decoding architecture: in the compression module, the forward state metrics are iteratively compressed to be the metrics with smaller values. In the decompression module, the compressed metrics are used to estimate the forward state metrics. Furthermore, only simple operations such as addition, shifting, and compare are applied in the compression and decompression modules. Theoretically, there are errors in this proposed technique. But simulation results still show that the introduced errors have little impact on the decoding performance, as compared with the optimal decoding algorithm.

The rest of this paper is organized as follows. Section 2 gives a brief introduction of the MAP decoding algorithm and the derived variants. Section 3 addresses the proposed technique in detail, which include the compression and the decompression modules. In Section 4, the introduced dummy decoding complexity, the SMC capacity, and the BER performance of the proposed technique are discussed with clear analysis. At last, this paper is concluded in Section 5.

2 Turbo decoding algorithm

To simplify the decoding complexity, the MAP algorithm in logarithmic domain (Log-MAP) and its derivatives are widely used [12]. For the single binary convolutional turbo code that was defined in the LTE-Advanced standard [1], by assuming the encoded sequence is transmitted through an additive white Gaussian noise (AWGN) channel, the Log-MAP decoding algorithm is shown by Eq. (1).

$$\tilde{\gamma}_k^{(z)}(s_{j_1,k-1}, s_{j_2,k}) = \frac{L_c}{2} (x_k^s y_k^s + x_k^p y_k^p) + \Lambda_{apr,k}^{(z)}(u_k) \quad (1a)$$

$$\tilde{\alpha}_k(s_{j_2,k}) = \max_{s_{j_1,k-1}}^* [\tilde{\alpha}_{k-1}(s_{j_1,k-1}) + \tilde{\gamma}_k^{(z)}(s_{j_1,k-1}, s_{j_2,k})] \quad (1b)$$

$$\tilde{\beta}_k(s_{j_1,k}) = \max_{s_{j_2,k+1}}^* [\tilde{\beta}_{k+1}(s_{j_2,k+1}) + \tilde{\gamma}_{k+1}^{(z)}(s_{j_1,k}, s_{j_2,k+1})] \quad (1c)$$

$$\begin{aligned} \Lambda_{apo,k}^{(z)}(u_k) = & \max_{(u_k=z)}^* [\tilde{\alpha}_{k-1}(s_{j_1,k-1}) + \tilde{\gamma}_k^{(z)}(s_{j_1,k-1}, s_{j_2,k}) + \tilde{\beta}_k(s_{j_2,k})] \\ & - \max_{(u_k=0)}^* [\tilde{\alpha}_{k-1}(s_{j_1,k-1}) + \tilde{\gamma}_k^{(z)}(s_{j_1,k-1}, s_{j_2,k}) + \tilde{\beta}_k(s_{j_2,k})] \end{aligned} \quad (1d)$$

$$\begin{aligned} \Lambda_{ex,k}^{(z)}(u_k) &= \Lambda_{apo,k}^{(z)}(u_k) - \Lambda_{apr,k}^{(z)}(u_k) - \Lambda_{in,k}^{(z)}(u_k) \\ \begin{cases} \Lambda_{in,k}^{(0)}(u_k) &= 0 \\ \Lambda_{in,k}^{(1)}(u_k) &= L_c y_k^s \end{cases} \end{aligned} \quad (1e)$$

In Eq. (1), z belongs to $\{0, 1\}$, $L_c = 2/\sigma^2$ (σ^2 is the noise variance of the AWGN channel), k is the decoding time slot, x_k^s and x_k^p are the transmitted codewords, y_k^s and y_k^p are the received codewords, where s and p denote the systematic and parity bits. $j \in \{0, \dots, 7\}$ is the index of the state metrics, $s_{j,k}$ is the j^{th} state at the decoding time slot k , $\tilde{\gamma}_k^{(z)}$ is the branch metric, $\tilde{\alpha}_k$ is the forward state metric, and $\tilde{\beta}_k$ is the backward state metric. For $u_k = z$, $\Lambda_{apr,k}^{(z)}(u_k)$, $\Lambda_{apo,k}^{(z)}(u_k)$ and $\Lambda_{ex,k}^{(z)}(u_k)$ are the a priori log-likelihood ratio (LLR), the a posteriori LLR and the extrinsic information, respectively.

Note that the \max^* operator in Eq. (1) is defined and simplified as follows [13]:

$$\begin{aligned} \max^*(x_1, x_2) &= \ln(\exp(x_1) + \exp(x_2)) \\ &\approx \min\{x_1, x_2\} + \max\{x_1 - x_2, 0.75(x_1 - x_2) + 0.625\} \end{aligned} \quad (2)$$

For a \max^* operator with more than two operands, Eq. (2) can be recursively applied. However, this recursion processing is not necessary in practical. By using Eq. (3), the decoding complexity can be significantly reduced, which is shown as follows [13].

$$\begin{aligned} \max^*(x_1, x_2, \dots, x_n) &\approx \max^*(y_1, y_2) \\ &\approx \min\{y_1, y_2\} + \max\{y_1 - y_2, 0.75(y_1 - y_2) + 0.625\} \end{aligned} \quad (3)$$

where y_1 and y_2 are the maximum two variables among $\{x_1, x_2, \dots, x_n\}$. In this research, Eq. (3) is adopted by Eq. (1) to calculate the forward state metrics $\tilde{\alpha}_k$, the backward state metrics $\tilde{\beta}_k$, and the a posteriori LLR $\Lambda_{apo,k}^{(z)}(u_k)$.

3 Method of proposed compression/decompression technique

3.1 Compression of the state metrics

In the hardware implementation of turbo decoder, the state metrics are stored in the last in and first out (LIFO) SMC. Existing researches have shown that the (10, 3) quantization scheme is suitable for getting satisfactory BER performance (10 is the total bits, 3 is the fractional bits) [9, 10]. To reduce the SMC capacity, we propose to compress the state metrics and to employ a (5, 3) quantization scheme in this research.

Seen from Eq. (1b), for each decoding time slot k , there are eight forward state metrics $\tilde{\alpha}_k(s_{j_2,k})$, $j_2 \in \{0, \dots, 7\}$. To facilitate the compression of these metrics, Eq. (4) is used for normalization. Since the decoding algorithm

is performed in the logarithmic domain, when the same value is subtracted from the eight forward state metrics at time slot k , value of the *a posteriori* LLR $\Lambda_{apo,k+1}^{(z)}(u_{k+1})$ is not affected by replacing $\tilde{\alpha}_k(s_{j_2,k})$ with $\alpha'_k(s_{j_2,k})$.

$$\alpha'_k(s_{j_2,k}) = \tilde{\alpha}_k(s_{j_2,k}) - \tilde{\alpha}_k(s_{0,k}) \quad (4)$$

Subsequently, $\alpha'_k(s_{j_2,k}), j_2 \in \{1, \dots, 7\}$ are recursively compressed by using Eq. (5) and noted that the value of $\alpha'_k(s_{0,k})$ is zero as implied by Eq. (4).

$$\alpha'_k(s_{j_2,k}) = \frac{\alpha'_k(s_{j_2,k}) - \alpha'_k(s_{j_2-1,k})}{4} \quad (5)$$

In Eq. (5), $1/4$ is the compression coefficient, and this division operation can be realized by using one 2-bits right shifting in hardware implementation. Considering that the values of $\alpha'_k(s_{j_2,k}), j_2 \in \{1, \dots, 7\}$ may be positive or negative, when the (5, 3) quantization scheme is adopted, the most significant bit represents the sign bit, while the rest bits represent the absolute value of the compressed forward state metrics. However, when $|\alpha'_k(s_{j_2,k})|$ is larger than 1.875, the (5, 3) scheme is not sufficient to quantize the compressed metrics. Therefore, a compare unit is employed to decide whether the next round of iterative compression should be performed: (i) if $\max(|\alpha'_k(s_{j_2,k})|) > 1.875$, $\alpha'_k(s_{j_2,k})$ are feedback to the compression module, where Eq. (4) is applied for the next round of compression; (ii) if $\max(|\alpha'_k(s_{j_2,k})|) \leq 1.875$, $\alpha'_k(s_{j_2,k})$ are output and then are stored in the LIFO SMC. Since $\alpha'_k(s_{j_2,k})$ are 10 bits quantized, and 7 bits are assigned for the integer part, at most 4 times of iterative compression is enough to guarantee $\max(|\alpha'_k(s_{j_2,k})|)$ is no more than 1.875. So, the number of iterative compression times I_k is an important parameter for the decompression, where 00, 01, 10, and 11 in binary denote the number of iterative compression times of 1, 2, 3, and 4 in decimal, respectively. As a result, when the compression procedure is finished, the number of iterative compression times and the compressed state metrics will be stored in the SMC. Furthermore, since $\alpha'_k(s_{0,k})$ equals to zero for each decoding time slot k , it is not necessary to store this metric in the SMC. For clear illustration, the word structure of the SMC is shown by Fig. 1 as below.

3.2 Decompression of the state metrics

In the backward direction, I_k and $\alpha'_k(s_{j_2,k})$ are read out from the LIFO SMC to estimate their original values, which is performed in the decompression module. Noted that $\alpha'_k(s_{0,k}) = 0$, we propose to decompress $\alpha'_k(s_{j_2,k}), j_2 \in \{7, \dots, 1\}$ by using Eq. (6).

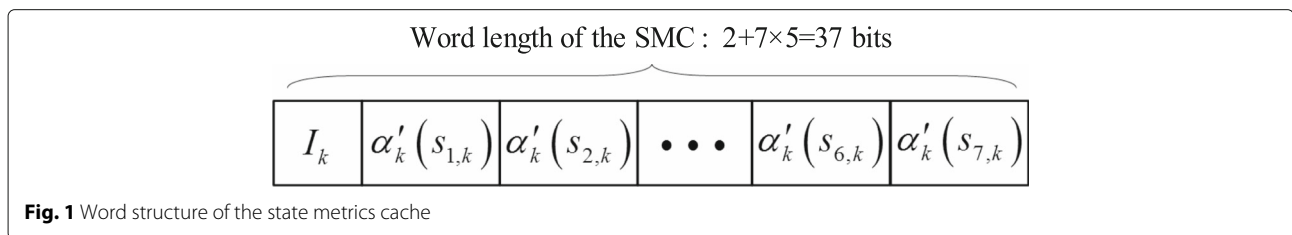
$$\alpha'_k(s_{j_2,k}) = 4\alpha'_k(s_{j_2,k}) + \alpha'_k(s_{j_2-1,k}), j_2 \in \{7, \dots, 1\} \quad (6)$$

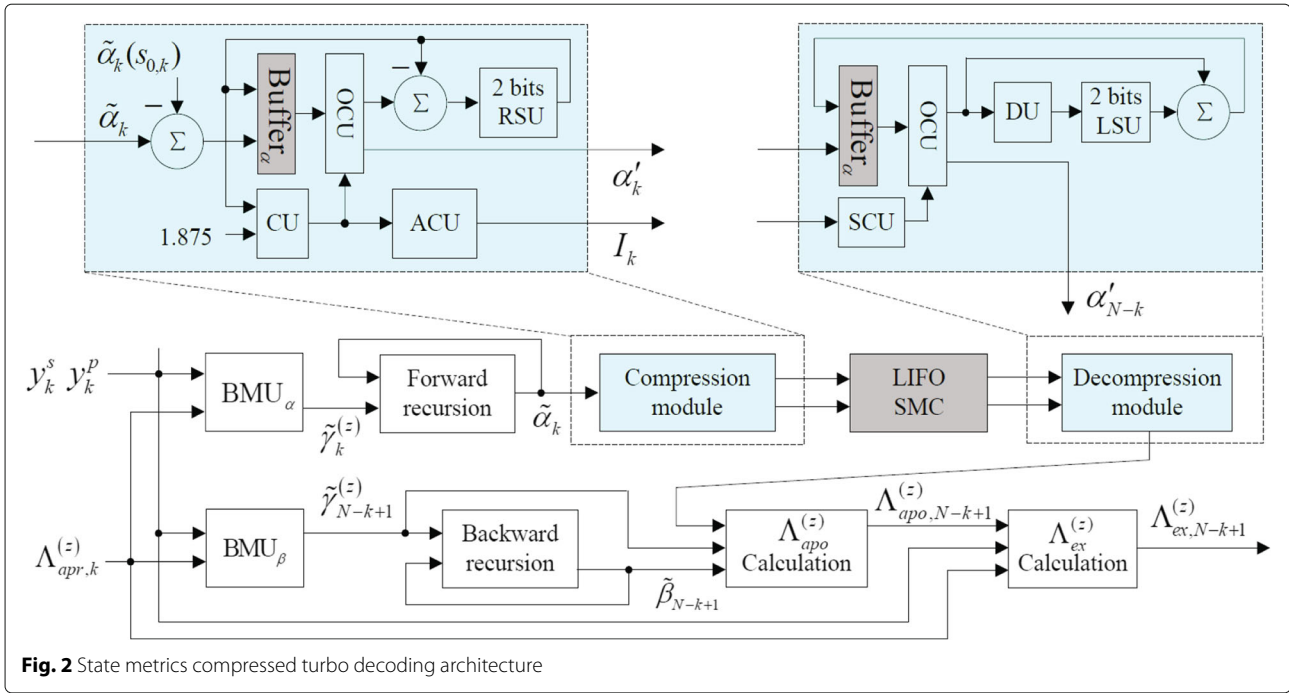
Equation (6) is the inverse calculation of Eq. (5), and I_k is used to decide how many times Eq. (6) should be recursively performed. For example, if $I_k = 10$ in binary, Eq. (6) is 3 times recursively performed, and then the decompressed state metrics are output to the *a posteriori* LLR calculation module. It should be noted that, $\alpha'_k(s_{j_2,k})$ in Eq. (4) are 10 bits quantized, when they are compressed by using Eq. (5), they will be 5 bits stored in the SMC. Considering the finite word length effect, when these metrics are used for decompression, errors will be introduced during the decompression procedure, which have negative effect on the decoding performance. As it can be seen from the simulation results in Fig. 6, the BER is slightly lost as compared with the Log-MAP algorithm.

3.3 State metrics compression based decoding architecture

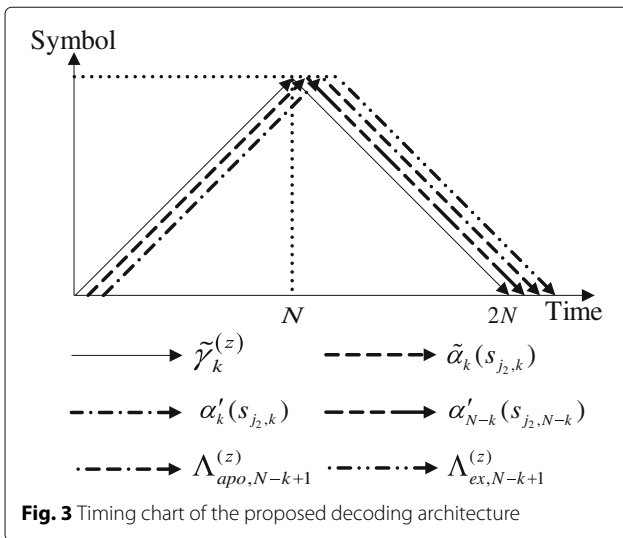
Based on the above described compression and decompression processes, two modules are inserted into the conventional decoding architecture. Assuming N denotes the decoding window length, the proposed decoding architecture is presented in Fig. 2, while Fig. 3 is the corresponded timing chart.

As shown in Fig. 2, in the forward direction, $\tilde{\gamma}_k^{(z)}$ are calculated in the branch metrics unit (BMU), and then are input to the forward recursion module, where $\tilde{\alpha}_k(s_{j_2,k})$ are recursively calculated. Instead of been stored in the LIFO SMC, $\tilde{\alpha}_k(s_{j_2,k})$ are input to the compression module. In this module, the output control unit (OCU) is applied to enable the compression, and the compare unit (CU) provides the trigger signal. At first, the buffer is initialized as $\alpha'_k(s_{j_2,k})$, one adder and one 2-bits right shifting unit (RSU) are used to perform the compression. The compressed metrics are feedback to the adder, the buffer and the compare unit (CU). If $\max(|\alpha'_k(s_{j_2,k})|) > 1.875$





is true, the OCU is triggered to enable the next round of compression, while the addition counter unit (ACU) increases I_k by 1 in decimal. Otherwise, the compression procedure is completed. Subsequently, the metrics in the buffer and the counting result in the ACU will be stored in the LIFO SMC. For the decompression module in the backward direction, I_k is input to the subtraction counter unit (SCU), by which the OCU is triggered to enable the decompression. Note that a delay unit (DU) is applied to adjust the time slot, and one 2-bits left shifting unit (LSU) is used to realize the multiply operation.



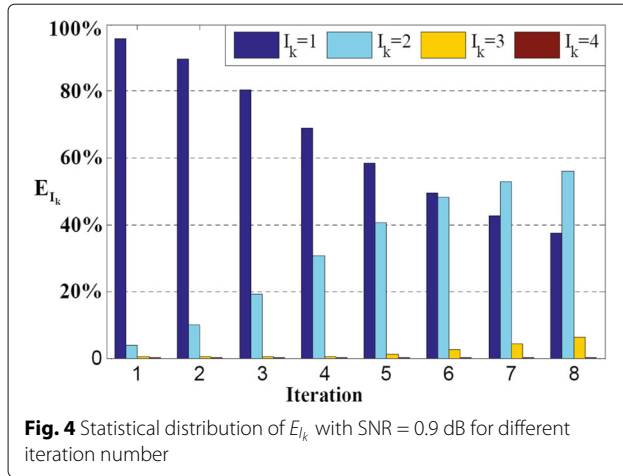
4 Results and discussion

As presented in Fig. 2, two modules are embedded in the conventional turbo decoder architecture. To demonstrate the effectiveness of this technique, a set of MATLAB scripts is developed for verification, as the improved decoding algorithm detailed in Section 2 is used to construct the state metrics compressed decoding architecture. Build on this software platform, the introduced dummy decoding complexity, the SMC capacity, and the BER performance are discussed in this section.

4.1 Dummy decoding complexity

For the compression module, Eq. (4) shows seven addition operations are performed before the compression. When the OCU is enabled, one adder and one 2-bits RSU are employed to calculate $\alpha'_k(s_{j_2,k})$, $j_2 \in \{1, \dots, 7\}$ recursively. Subsequently, seven compare operations are performed in the CU to decide whether the next round of compression should be enabled, and the ACU should increase I_k by 1 or not. Similarly, in the decompression module, the SCU decides how many times the decompression procedure should be performed. By Eq. (6) and Fig. 2, the I_k in OCU is used to enable the decompression module, in which one DU, one 2-bits LSU and one addition operation are performed to estimate the state metrics for each round of decompression. Therefore, I_k is the key parameter to analyze the dummy decoding complexity of the proposed technique.

In Section 3, the described compression procedure shows I_k is a variable that depend on if $\max(|\alpha'_k(s_{j_2,k})|) > 1.875$ is



true, which means the difference between $\alpha'_k(s_{j_2,k})$, $j_2 \in \{1, \dots, 7\}$ is the most important factor. As a result, I_k may take different values with the same signal to noise ratio (SNR) but with different iteration number, or with the same iteration number but with different SNR. To this purpose, we define D_{I_k} as the total amount for I_k equals to 1, 2, 3, and 4 in decimal, respectively. Consequently, the percentage E_{I_k} for D_{I_k} are calculated as:

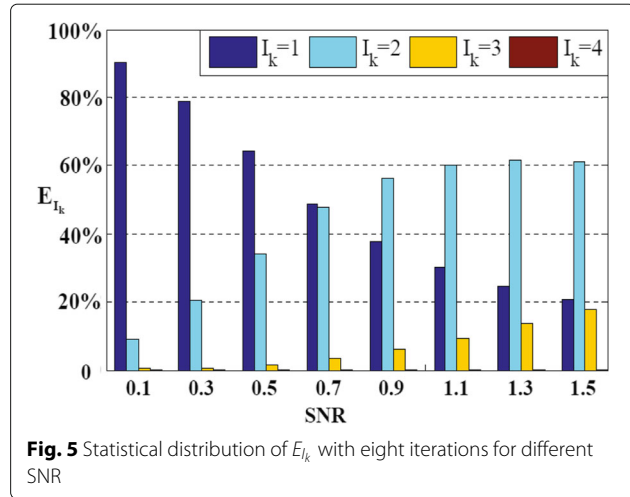
$$E_{I_k} = \frac{D_{I_k}}{\sum_{I_k=1}^4 D_{I_k}} \times 100 \quad (7)$$

Assuming the frame length is 1440, the statistical distribution of E_{I_k} with the same SNR but with different iteration number is presented in Fig. 4. Although SNR = 0.9 dB is a special case, the results illustrated in Fig. 4 still represent a generic tendency: as the iteration goes on, more rounds of compression/decompression will be performed.

Similarly, the statistical distribution of E_{I_k} with 8 iterations but with different SNR is presented in Fig. 5. As SNR increases, the difference among the forward state metrics $\tilde{\alpha}_k(s_{j_2,k})$, $j_2 \in \{0, \dots, 7\}$ becomes larger, i.e., the probability for I_k to represent a larger value will increase accordingly.

4.2 SMC capacity

At the cost of dummy decoding complexity that performed in the compression and decompression modules, the compressed state metrics can be 5 bits quantized. As presented in Fig. 1, one I_k (2-bits represented) and



seven compressed metrics should be stored in the SMC for each time slot ($1 \times 2 + 7 \times 5 = 37$ bits). Compared with the conventional decoding architecture, where the quantization scheme is (10, 3) and eight state metrics should be stored in the SMC ($8 \times 10 = 80$ bits), the proposed technique have reduced the SMC capacity by 53.75%. To summarize, the SMC organization and capacity comparison are illustrated in Table 1.

4.3 BER simulation

To investigate the decoding performance of the proposed technique, the simulation environment is set as follows: for the LTE-Advanced standard defined turbo code [1], two kinds of frame length are used for demonstration (800 and 1440 bits, respectively), the code rate is 1/3, and the encoded sequences are transmitted through an AWGN channel. Three decoding algorithms, i.e., the optimal Log-MAP algorithm, the state metrics compressed decoding technique (the near optimal decoding algorithm detailed in Section 2 is implemented), and the maximum Log-MAP (Max-Log-MAP) algorithm, are used for comparison. For each frame, the employed decoding algorithm is 8 times performed iteratively. Moreover, the quantization schemes in Table 2 are adopted [9–11], and δ is the scaling factor [14].

Seen from Fig. 6, the Log-MAP algorithm gets the best BER performance, about 0.2 dB superior to the Max-Log-MAP algorithm, but only slightly outperforms that of the

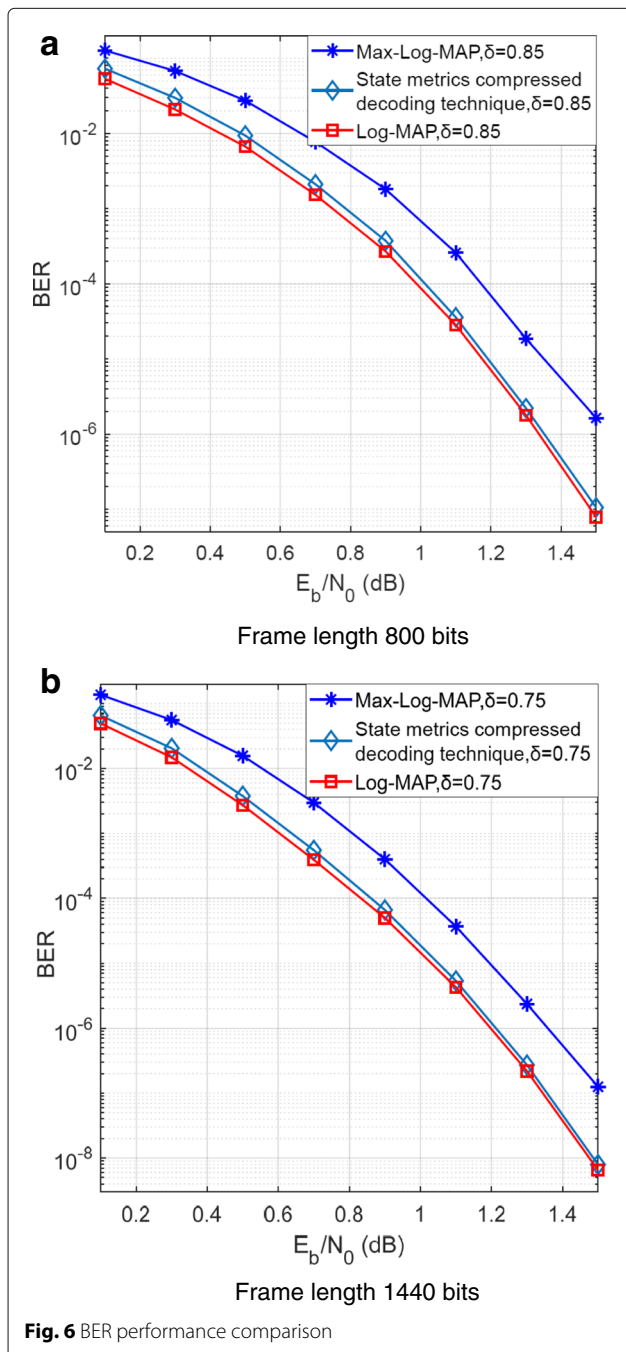
Table 1 Comparison of SMC organization and capacity

Decoding scheme	SMC organization			Total
	Quantization scheme	Number of stored metrics	Iteration number	
Conventional decoding scheme	(10, 3)	8	0	$8 \times 10 \times N$ (100%)
State metrics compressed decoding scheme	(5, 3)	7	I_k (2 bits)	$(7 \times 5 + 2) \times N$ (46.25%)

Table 2 Quantization schemes adopted in the simulation

Metric	y_k^s, y_k^p	$\tilde{\alpha}_k, \tilde{\beta}_k$	$\tilde{\gamma}_k^{(z)}, \Lambda_{apr,k}^{(z)}, \Lambda_{ex,k}^{(z)}$	$\Lambda_{apo,k}^{(z)}$	$\alpha_k'(s_{j2,k})$	l_k
(q, f)	(5, 3)	(10, 3)	(9, 3)	(11, 3)	(5, 3)	(2, 0)

state metrics compressed decoding technique. Thanks for the simplified \max^* operator in Section 2, although the inserted compression and decompression modules have introduced some errors, the resulted BER loss is limited, about 0.05 dB at BER of 10^{-2} and then becomes smaller as SNR increase.

**Fig. 6** BER performance comparison

5 Conclusions

In the energy resource-limited applications where turbo code is adopted for data transmission, memory-reduced turbo decoder is an effective solution to reduce the overall power dissipation. In this paper, a state metrics compressed turbo decoding technique is proposed. It has been shown that, by inserting one compression and one decompression modules in the conventional decoding architecture, a quantization scheme with smaller SMC word length can be applied, and the SMC capacity is reduced by 53.75%. Dummy decoding complexity in the compression and decompression modules is analyzed in detail. In most cases of different iteration number but with the same SNR, and of different SNR with the same iteration number, one or two rounds of compression/decompression are implemented. For the LTE-Advanced standard defined turbo code, the proposed decoding technique clearly outperforms the Max-Log-MAP algorithm in BER performance, and only slightly degraded as compared with the optimal Log-MAP algorithm.

Abbreviations

ACU: Addition counter unit; AWGN: Additive white Gaussian noise; BER: Bit error rate; BMU: Branch metrics unit; CU: Compare unit; DU: Delay unit; Log-MAP: The MAP algorithm in logarithmic domain; LLR: Log-likelihood ratio; LIFO: Last in and first out; LSU: Left shifting unit; MAP: The maximum a posteriori; Max-Log-MAP: Maximum Log-MAP; OCU: Output control unit; RSU: Right shifting unit; SMC: State metrics cache; SCU: Subtraction counter unit; SNR: Signal to noise ratio; WSNs: Wireless sensor networks

Acknowledgements

The authors thank the researchers from the School of Electronics and Information Engineering, Southwest University, for the helpful insights on the improvement of this manuscript.

Funding

This work was supported in part by the National Natural Science Foundation of China under Grant 61671390, in part by the China Postdoctoral Science Foundation under Grant 2015M570776, and in part by the Fundamental Research Fund for Central University (Southwest University) under Grant SWU113044.

Authors' contributions

MZ proposed the state metrics compression/decompression technique for the turbo decoder architecture design, and carried out the analysis to demonstrate the effectiveness of this idea, which include the comparison of dummy computational complexity, SMC capacity, and BER performance. From the perspective of practical engineering, ZP gave the direction on the wireless sensor networks of which the power-efficient turbo decoder is to be applied. MX proposed the instruction on how to perform the compression/decompression procedures, together with computational complexity analysis. HW recommended the near optimal MAP decoding algorithm and the quantization scheme for simulation, which are the basis of this research work. All authors read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Author details

¹The Key Laboratory of Networks and Cloud Computing Security of University of Chongqing, College of Electronic and Information Engineering, Southwest University, Chongqing 400715, China. ²ABB Corporate Research, Forskargränd

7, SE-721 78 Västerås, Sweden. ³School of Electrical Engineering, KTH Royal Institute of Technology, Stockholm SE-100 44, Sweden. ⁴National Key Laboratory of Science and Technology on Communications at University of Electronic Science and Technology of China, Chengdu 611731, China.

Received: 1 November 2017 Accepted: 25 May 2018

Published online: 15 June 2018

References

1. 3rd Generation partnership project, Multiplexing and channel coding, 3rd Generation Partnership Project, TS 36.212 version v11.1.0 Release 11, 2013. [Online]. Available: http://www.etsi.org/deliver/etsi_ts/136200_136299/136212/11.01.00_60/ts_136212v110100p.pdf
2. Digital Video Broadcasting (DVB), Second Generation DVB Interactive Satellite System (DVB-RCS2); Part 2: Lower Layers for Satellite Standard, Digital Video Broadcasting (DVB), ETSI EN 301 545-2 V1.2.1, 2014, [Online]. Available: http://www.etsi.org/deliver/etsi_en/301500_301599/30154502/01.02.01_60/en_30154502v010201p.pdf
3. FB Matthew, L Liang, GM Robert, et al, 20 years of turbo coding and energy-aware design guidelines for energy-constrained wireless applications. *IEEE Communication Surveys & Tutorials*. **18**(1), 8–28 (2016)
4. J Haghighat, H Behroozi, DV Plant, in *IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*. Joint decoding and data fusion in wireless sensor networks using turbo codes (IEEE, Cannes, 2008), pp. 1–5. <https://ieeexplore.ieee.org/document/4699729/>
5. H Cam, in *IEEE International Conference on Communications (ICC)*. Multiple-input turbo code for joint data aggregation, source and channel coding in wireless sensor networks (IEEE, Istanbul, 2006), pp. 3530–3535. <https://ieeexplore.ieee.org/document/4025020/>
6. YH Yitbarek, K Yu, J Åkerberg, M Gidlund, M Björkman, in *2014 IEEE International Conference on Industrial Technology (ICIT)*. Implementation and evaluation of error control schemes in industrial wireless sensor networks (IEEE, Busan, 2014), pp. 730–735. <https://ieeexplore.ieee.org/document/6895022/>
7. L Li, GM Robert, BM Al-Hashimi, L Hanzo, A low-complexity turbo decoder architecture for energy-efficient wireless sensor networks. *IEEE Trans. VLSI Syst.* **21**(1), 14–22 (2013)
8. CH Lin, CC Wei, Efficient window-based stopping technique for double-binary turbo decoding. *IEEE Commun. Lett.* **17**(1), 169–172 (2013)
9. DS Lee, IC Park, Low-power Log-MAP decoding based on reduced metric memory access. *IEEE Trans. Circ. Syst. I: Regular Papers*. **53**(6), 1244–C1253 (2006)
10. CH Lin, CY Chen, AY Wu, Low-power memory-reduced traceback MAP decoding for double-binary convolutional turbo decoder. *IEEE Trans. Circ. Syst. I: Regular Papers*. **56**(5), 1005–1016 (2009)
11. M Martina, G Masera, State metric compression techniques for turbo decoder architectures. *IEEE Trans. Circ. Syst.* **58**(5), 1119–1128 (2011)
12. M Martina, S Papaharalabos, PT Mathiopoulos, G Masera, Simplified Log-MAP algorithm for very low-complexity turbo decoder hardware architectures. *IEEE Trans. Instrum. Meas.* **63**(3), 531–537 (2014)
13. M Zhan, J Wu, ZZ Zhang, et al, Low-complexity error correction for ISO/IEC/IEEE 21451-5 sensor and actuator networks. *IEEE Sensors J.* **15**(5), 2622–2630 (2015)
14. J Vogt, A Finger. Improving the Max-Log-MAP turbo decoder. *Electron. Lett.* **36**(23), 1937–1938 (2000)

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com