Frontiers of Information Technology & Electronic Engineering
www.zju.edu.cn/jzus; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

An improved low-complexity sum-product decoding algorithm for low-density parity-check codes

Michaelraj Kingston ROBERTS†, Ramesh JAYABALAN

(*Department of Electronics and Communication Engineering, PSG College of Technology, Coimbatore 641004, India*)

†E-mail: king.pane@gmail.com

Received July 24, 2014; Revision accepted Sept. 25, 2014; Crosschecked May 7, 2015

Abstract: In this paper, an improved low-complexity sum-product decoding algorithm is presented for low-density parity-check (LDPC) codes. In the proposed algorithm, reduction in computational complexity is achieved by utilizing fast Fourier transform (FFT) with time shift in the check node process. The improvement in the decoding performance is achieved by utilizing an optimized integer constant in the variable node process. Simulation results show that the proposed algorithm achieves an overall coding gain improvement ranging from 0.04 to 0.46 dB. Moreover, when compared with the sum-product algorithm (SPA), the proposed decoding algorithm can achieve a reduction of 42%–67% of the total number of arithmetic operations required for the decoding process.

Key words: Computational complexity, Coding gain, Fast Fourier transform (FFT), Low-density parity-check (LDPC) codes, Sum-product algorithm (SPA)

doi:10.1631/FITEE.1400269 **Document code:** A **CLC number:** TN911

1 Introduction

Error correcting codes (ECC) are used in nearly all forms of electronic communication and storage systems to facilitate error free transmission of data within a specified bandwidth. Low-density parity-check (LDPC) codes (Gallager, 1962) are one of the very few next generation error correcting codes that allow transmission of data at a rate close to Shannon's limit (Chung *et al.*, 2001). Since their rediscovery (MacKay and Neal, 1997), LDPC codes with their simple decoding procedure can correct the channel errors at a relatively low signal-to-noise ratio (SNR) with feasible complexity. Due to the superior error correcting capability and inherent parallelism for hardware implementation, LDPC codes have been preferred for advanced wireless communication standards such as DVD-S2 (Morello and Mignone, 2006), Wi-MAX (IEEE, 2009), WLAN (IEEE, 2015), and International Telecommunication Unit Transmission Systems (ITU-T).

Among various decoding algorithms, the conventional sum-product algorithm (SPA) (Fossorier *et al.*, 1999) based on the concept of iterative message passing (Richardson and Urbanke, 2001) achieves the best decoding performance compared to other decoding algorithms. Although SPA achieves the best bit error rate (BER) performance, the check node process requires a lot of logarithmic and multiplicative computations to exchange log likelihood ratio (LLR) information between the nodes. This in turn leads to very high computational complexity, and thus makes it unsuitable for hardware implementations. The simplified sum-product algorithm (SSPA) (Lee *et al.*, 2008) and modified sum-product algorithm (MSPA) (Papaharalabos *et al.*, 2007) have been proposed with an objective to eliminate the high computational complexity of the conventional SPA. These algorithms focus on modifying and simplifying the nonlinear functions by dividing them into

ORCID: Michaelraj Kingston ROBERTS, http://orcid.org/0000-0002-1484-703X

approximated quantization regions. These methodologies, however, slightly reduce the computational complexity at the expense of performance degradations. A simple hard decision based low-complexity decoding algorithm for LDPC codes (Chandrasetty and Aziz, 2011) was proposed with a view to reduce the implementation complexity of the conventional SPA. Recently, a fast Fourier transform based sum-product algorithm (FFT-SPA) for decoding LDPC lattices (Safarnejad and Sadeghi, 2012) and Abelian groups (Goupil *et al.*, 2007) was proposed to reduce the computational complexity of SPA. However, the main drawback of utilizing the conventional FFT (Sorensen *et al.*, 1986) is that it increases the computation time required for processing the large data sets of information. Furthermore, the conventional FFT is unreliable for the processing of large data sets over the channel with a large bandwidth.

In this paper, an improved low-complexity sum-product decoding algorithm is proposed for LDPC decoding. The objective of this proposed algorithm is to reduce the total number of computations required in the check node process. The following modifications are carried out in the check node and variable node processes: (1) A fast Fourier transform with time shift (FFT_h) (Johnson and Frigo, 2007; Yuan *et al.*, 2011) is incorporated in the check node process to reduce the multiple nonlinear operations; (2) An optimized integer weight constant is incorporated in the variable node process to improve the decoding performance (Chandrasetty and Aziz, 2011). The performance of the proposed algorithm is validated with Wi-MAX (IEEE, 2009) and WLAN (IEEE, 2015) standard LDPC codes. The decoding performance of the proposed algorithm is compared with those of SPA, SSPA, and MSPA.

2 Related work

2.1 Sum-product algorithm

The conventional SPA (Fossorier *et al.*, 1999) is an iterative message passing algorithm which utilizes high-precision LLR messages as the input for the decoding process (Richardson and Urbanke, 2001). The decoding operation of SPA is carried out in two phases of message passing between the edges of check nodes and variable nodes in a Tanner graph. Due to the utilization of a soft decision based message passing process, the iterative decoding process of SPA relies solely on the updated information between the two nodes.

Let q_{nm} be the message passed from the nth variable node (bit node) to the mth check node (VTC), and r_{mn} the message passed from the mth check node to the nth variable node (CTV). For the conventional SPA, the VTC message can be split into its sign and magnitude:

$$q_{nm}=\mu_{nm}\cdot\lambda_{nm}, \tag{1}$$

where μ_{nm}=sign(q_{nm}) and λ_{nm}=|q_{nm}|.

In the beginning of the decoding process (at iteration 0), the nth VTC message $q_{nm}^{(0)}=2y_i/\sigma^2$ is initially assigned to the LLR information, where y_i denotes the received input and σ^2 the variance of the binary-input additive white Gaussian noise (AWGN) channel. The intrinsic information obtained from the channel is passed over to check nodes from variable nodes to start the decoding process.

The CTV information updated at the check node unit (CNU) can be computed as

$$\begin{aligned} r_{mn}=2&\prod_{i\in\{N(m)\backslash n\}}\operatorname{sign}(y_i)\\ &\cdot\operatorname{arctanh}\left(\prod_{i\in\{N(m)\backslash n\}}\tanh\frac{2|y_i|}{\sigma^2}\right),\end{aligned} \tag{2}$$

where $N(m)$ denotes a set of bits participating in the mth check. This updated message r_{mn} is sent to the corresponding variable nodes for further processing.

The VTC information updated can be computed using the following equation:

$$q_{nm}=q_{nm}^{(0)}+\sum_{\substack{m\in\{M(n)\backslash m'\},\\ m'\neq m}} r_{mn}, \tag{3}$$

where $M(n)$ denotes a set of checks participating in the nth bit. The hard decision of LLR for a variable node n can be computed as

$$q_n=q_{nm}^{(0)}+\sum_{m\in M(n)} r_{mn}, \tag{4}$$

which gives approximated codewords that are required to satisfy appropriate parity-check equations.

The decoding process is continued until parity-check conditions are satisfied by the estimated codewords or until the maximum number of iterations is reached. When compared with the other soft decision based LDPC decoding algorithms, SPA achieves the best BER performance. However, the very high computational complexity issue of SPA makes it unsuitable for efficient hardware implementation.

2.2 Modified sum-product algorithm

To overcome the computational complexity of the conventional SPA, MSPA was proposed (Papaharalabos *et al*., 2007). It focuses on modifying the hyperbolic tangent (tanh) and inverse hyperbolic tangent (arctanh) functions to reduce the computational complexity. This process is carried out by splitting the hyperbolic tangent functions into seven regions and representing it by eight quantization values (He *et al*., 2002), respectively.

MSPA utilizes multiplication and division operations in the check node update process to update the soft LLR information. Compared to SPA, the computational complexity of MSPA is significantly reduced by utilizing the quantization values (Papaharalabos *et al*., 2007) with a modified hyperbolic tangent function and its inverse. The expressions for the modified hyperbolic tangent function and inverse hyperbolic tangent function are given as

$$\tanh_{\text{modified}}(x)=\begin{cases}\tanh(x), & |x|<x_{\text{o}},\\ \text{sign}(x)\cdot\tanh(x_{\text{o}}), & |x|\geq x_{\text{o}},\end{cases} \tag{5}$$

$$\text{arctanh}_{\text{modified}}(x)=\begin{cases}\text{arctanh}(x), & |x|<x_{\text{o}},\\ \text{sign}(x)\cdot\text{arctanh}(x_{\text{o}}), & |x|\geq x_{\text{o}},\end{cases} \tag{6}$$

where x is the bipolar value and x_{o} the clipping value.

However, MSPA suffers from severe BER performance degradation (0.25 dB) compared to SPA due to the approximation errors caused by quantization schemes. In addition to performance degradation, the approximation error increases the total number of decoding iterations required to correct the channel errors at a relatively low SNR.

2.3 Simplified sum-product algorithm

The simplified sum-product algorithm (Lee *et al*., 2008) rectifies the drawbacks of MSPA (Papaharalabos *et al*., 2007). It reduces the computational complexity of the check node process by removing additional shifting operations required for updating check node information. Moreover, in SSPA the total number of arithmetic computations is reduced by replacing multiplication and division operations with addition and subtraction operations along with additional logarithmic and exponential functions. These arithmetic functions are then concurrently computed along with the hyperbolic tangent function and its inverse in the quantization table. Due to these modifications, the negative part of the quantization table can be eliminated compared to MSPA. The check node process of the SSP algorithm is given as

$$r_{mn}=\prod_{i\in\{N(m)\backslash n\}}\text{sign}(y_i)\cdot\exp\left\{\text{arctanh}\left[\sum_{i\in\{N(m)\backslash n\}}\ln\left(\tanh\frac{2|y_i|}{\sigma^2}\right)\right]\right\}. \tag{7}$$

The variable node process of SSPA is similar to that of the conventional SPA. By replacing multiplication and division by addition and subtraction operations respectively, SSPA outperforms MSPA in terms of computational complexity. Moreover, by utilizing only the positive part of the quantization table SSPA achieves BER performance closest to SPA and outperforms MSPA by 0.8 dB (Lee *et al.*, 2008).

3 An improved low-complexity decoding algorithm for regular LDPC codes

Although the computational complexity of SSPA is better than that of MSPA and the conventional SPA, the decoding performance is still average when compared to a conventional SPA. The quantization levels used in LLR values significantly impact the decoding performance. Recently, FFT-SPA for decoding LDPC lattice (Safarnejad and Sadeghi, 2012) has been proved to achieve good decoding performance with reduced computational complexity for the elements in the lattice subgroup. However, for the transmitting and processing of a large set of input sequences with a small number of non-zero elements over a channel with a large bandwidth, the conventional FFT method is unsuitable. This is mainly due to the sparseness property of LDPC codes which require longer execution time and additional memory resources to store the temporarily processed data. Therefore, to reduce the computational complexity without degrading the decoding performance, modifications have been done in both the check node and variable node processes of the proposed algorithm. In the check node process, FFT_h is utilized to reduce the overall computational complexity. In the variable node process, an optimized integer constant is used to improve the decoding performance of the proposed algorithm.

For an (N, K) regular LDPC code, both the variable node degree (d_v) and check node degree (d_c) are small compared to the block length (N) and information length (K). Let $C=(C_0, C_1, ..., C_{N-1})\in\{\text{GF}(2)\}^N$ denote the codeword C which is mapped into the bipolar sequence $x=(x_0, x_1, ..., x_{N-1})$ before transmission, where $x_n=2C_n-1$ with $0\le n\le N-1$. Let $y=(y_0, y_1, ..., y_{N-1})$ be the soft decision received at the receiver. For $0\le n\le N-1$, $y_n=x_n+w_n$ where w_n is the Gaussian random variable with zero mean and variance σ^2 and is independent of x_n. Assuming that AWGN has a power spectral density $N_\text{o}/2$ (N_o is the noise spectral density) for a code rate of R_c, we have $\sigma^2=(2R_\text{c}\cdot E_\text{b}/N_\text{o})^{-1}$, where E_b is the energy per bit. Let the corresponding binary hard decision of the received sequence be $Z_n=1$ (if $q_n^{(k)}<0$). Then $S=(S_0, S_1, \ldots, S_{M-1})$ gives the syndrome of hard decision sequence $\boldsymbol{Z}^{(0)}$.

The check node and variable node operations of the proposed algorithm are described as follows:

Initialization: The initial iteration number is set as $k=0$ and the maximum number of decoding iterations is set to k_max. For $0\le n\le N-1$, let $q_{nm}^{(0)}$ be the LLR of a prior probability of bit n. Then set

$$q_{nm}^{(0)}=2y_i/\sigma^2. \tag{8}$$

Step 1 (check node update): For $0\le m\le M-1$ and $0\le n\le N-1$, the message passed from the check node to variable node $r_{mn}^{(k)}=[r_{mn}^{(k)}(0), r_{mn}^{(k)}(1), ..., r_{mn}^{(k)}(u-1)]^{-1}$ can be computed as

$$r_{mn}^{(k)}=\prod_{i\in\{N(m)\backslash n\}}\text{sign}(y_i)\cdot F^{-1}\left[\prod_{i\in\{N(m)\backslash n\}}F\left(\frac{2|y_i|}{\sigma^2}\right)\right], \tag{9}$$

where F and F^{-1} denote the discrete Fourier transform and its inverse, respectively, and u denotes the alphabet size. Similarly,

$$\begin{aligned} r_{mn}^{(k)}=&\prod_{i\in\{N(m)\backslash n\}}\text{sign}(y_i)\\ &\cdot\text{FFT}_\text{h}^{-1}\left[\prod_{i\in\{N(m)\backslash n\}}\text{FFT}_\text{h}\left(\frac{2|y_i|}{\sigma^2}\right)\right],\end{aligned} \tag{10}$$

where FFT_h and FFT_h^{-1} denote the fast Fourier transform with time shift and its inverse, respectively.

Step 2 (variable node update): For $0\le m\le M-1$ and $0\le n\le N-1$, q_{nm} can be computed as

$$\begin{aligned} q_{nm}^{(k)}=&q'_{nm}+\sum_{m\in M(n)}\left(\prod_{i\in\{N(m)\backslash n\}}\text{sign}(y_i)\right)\\ &\cdot\text{FFT}_\text{h}^{-1}\left[\prod_{i\in\{N(m)\backslash n\}}\text{FFT}_\text{h}\left(\frac{2|y_i|}{\sigma^2}\right)\right],\end{aligned} \tag{11}$$

where

$$q'_{nm}=q_{nm}^{(0)}+\sum X_i, \tag{12}$$

and

$$X_i=\begin{cases}M/2+1, & m_i=0,\\ (M+1)/2, & m_i=1.\end{cases} \tag{13}$$

X_i is the optimized integer constant and m_i denotes the bit suggestion from the ith check node. Then

$$q_i=\text{sign}(q_{nm}-X_i). \tag{14}$$

Therefore, the simplified variable node update equation is given as

$$\begin{aligned} q_{nm}^{(k)}=&q'_{nm}+\sum_{m\in M(n)}\left[1-2(S_m\oplus\hat{C}_n)\right]\\ &\cdot\text{FFT}_\text{h}^{-1}\left[\sum_{i\in\{N(m)\backslash n\}}\text{FFT}_\text{h}\left(\frac{2|y_i|}{\sigma^2}\right)\right],\end{aligned} \tag{15}$$

where $\hat{C}_n$ is the hard output of the decoder and $S_m=\sum_{n=0}^{N-1}h_{m,n}\cdot\hat{c}_n(\text{mod }2)$ for $m=1, 2, ..., M$ ($h_{m,n}$ is the (m, n)th entry of the parity-check matrix $\boldsymbol{H}$ and $\hat{c}_n$

denotes the syndrome of the hard decision sequence). Then we have

$$q_n^{(k)} = q'_{nm} + \sum_{m\in M(n)} \left[1-2(S_m \oplus Z_n^{(k)})\right] \cdot \mathrm{FFT}_\mathrm{h}^{-1}\left[\sum_{i\in N(m)} \mathrm{FFT}_\mathrm{h}\left(\frac{2|y_i|}{\sigma^2}\right)\right]. \tag{16}$$

Step 3 (decision): The hard decision $\boldsymbol{Z}^{(0)}=(Z_0^{(0)}, Z_1^{(0)}, ..., Z_{N-1}^{(0)})$ is determined by $Z_n^{(k)}=0$ if $q_n^{(k)}\geq 0$, and $Z_n^{(k)}=1$ if $q_n^{(k)}<0$. If $\boldsymbol{Z}^{(k)}\cdot\boldsymbol{H}^\mathrm{T}=\boldsymbol{0}$, output $\boldsymbol{Z}^{(k)}$ as the decoded codeword and the decoding process is stopped; otherwise, go back to step 1.

Step 4: This process is repeated until the maximum number of decoding iterations is reached or parity-check conditions are satisfied by the estimated codewords, i.e., $\hat{\boldsymbol{c}}\cdot\boldsymbol{H}^\mathrm{T}=\boldsymbol{0}$.

A unique difference between the proposed algorithm and SPA is that the proposed algorithm does not require the complex hyperbolic tangent function during the check node update process. Instead of utilizing the hyperbolic tangent function, it is replaced by $\mathrm{FFT_h}$ (Johnson and Frigo, 2007; Yuan *et al.*, 2011) in the check node update process. The main difference between $\mathrm{FFT_h}$ and FFT (Sorensen *et al.*, 1986) is that in $\mathrm{FFT_h}$ the input data are shifted circularly before transmission. A unique advantage of utilizing the $\mathrm{FFT_h}$ function when compared to the FFT function and hyperbolic tangent function is that it requires less computation time to process large data sets of information. Moreover, the $\mathrm{FFT_h}$ function does not require additional memory resources to store the temporarily processed data obtained during each iteration. However, the utilization of the $\mathrm{FFT_h}$ function brings performance degradation of more than 0.1 dB, which is compensated for by introducing an optimized integer constant during the variable node update process. These optimized integer constant values found through heuristic simulations are added to the LLR values to improve the decoding performance of the proposed algorithm. Furthermore, in the decoding process of the proposed algorithm, the original LLR values are required for initialization of the decoding process. The original LLR values are then updated during each step of the decoding process and are subsequently utilized for the remainder of the decoding process.

4 Simulation results

To illustrate the decoding performance of the proposed decoding algorithm, (N, K) regular LDPC codes belonging to the WLAN standard (IEEE, 2015) and Wi-MAX standard (IEEE, 2009) have been considered. (648, 324) and (1296, 864) regular LDPC codes belonging to the WLAN standard and a (2304, 1152) regular LDPC code belonging to the Wi-MAX standard have been utilized for simulation. The MATLAB simulations were carried out assuming that the encoded information bits were binary phase shift keying (BPSK) modulated and transmitted over an AWGN channel. The LDPC codes were designed by the procedure as described by Jiang *et al.* (2012). For all simulation results of this work, the maximum number of decoding iterations was chosen to be 20 (Lee *et al.*, 2008).

4.1 Decoding performance

Fig. 1 shows the decoding performance of a (648, 324) regular LDPC code with the code rate of 1/2 belonging to the WLAN standard. The simulation results show that SPA and the proposed decoding algorithm exhibit better decoding performance with 20 decoding iterations compared with FFT-SPA, SSPA, and MSPA. At a BER of 10^{-4}, the proposed algorithm achieves a coding gain improvement of 0.04, 0.09, and 0.19 dB when compared with FFT-SPA, SSPA, and MSPA, respectively. Similarly, at a BER of 10^{-4}, the proposed algorithm exhibits a performance degradation of 0.02 dB when compared with SPA. However, the performance degradation suffered by the proposed algorithm is much less when

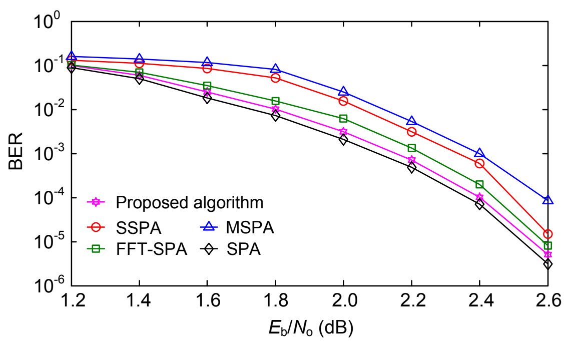

Fig. 1 Bit error rate performance comparisons for a (648, 324) regular LDPC code under various SPA algorithms

compared with the performance degradation of 0.06, 0.11, and 0.21 dB suffered by FFT-SPA, SSPA, and MSPA, respectively.

Fig. 2 shows the decoding performance of a (2304, 1152) regular LDPC code with the code rate of 1/2 belonging to the Wi-MAX standard. The simulation results show that SPA and the proposed algorithm exhibit better decoding performance with 20 decoding iterations compared with FFT-SPA, SSPA, and MSPA. At a BER of 10^{-5}, the proposed algorithm achieves a coding gain improvement of 0.06, 0.25, and 0.27 dB when compared with FFT-SPA, SSPA, and MSPA, respectively. Similarly, at a BER of 10^{-5}, the proposed algorithm exhibits a performance degradation of 0.04 dB when compared with SPA. However, the performance degradation suffered by the proposed algorithm is much less when compared with the performance degradation of 0.10, 0.29, and 0.31 dB suffered by FFT-SPA, SSPA, and MSPA, respectively.

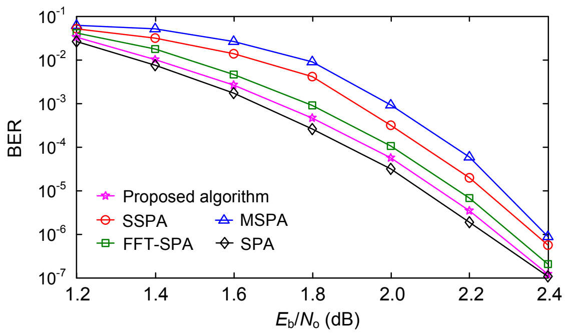

Fig. 2 Bit error rate performance comparisons for a (2304, 1152) regular LDPC code under various SPA algorithms

Fig. 3 shows the decoding performance of a (1296, 864) regular LDPC code with the code rate of 2/3 belonging to the WLAN standard. The simulation results show that SPA and the proposed decoding algorithm exhibit better decoding performance with 20 decoding iterations compared with FFT-SPA, SSPA, and MSPA. At a BER of 10^{-5}, the proposed algorithm achieves a coding gain improvement of 0.10, 0.28, and 0.46 dB when compared with FFT-SPA, SSPA, and MSPA, respectively. Similarly, at a BER of 10^{-5}, the proposed algorithm exhibits a performance degradation of 0.04 dB when compared with SPA. However, the performance degradation suffered by the proposed algorithm is much less when compared with the performance degradation of 0.11, 0.32, and 0.50 dB suffered by FFT-SPA, SSPA, and MSPA, respectively.

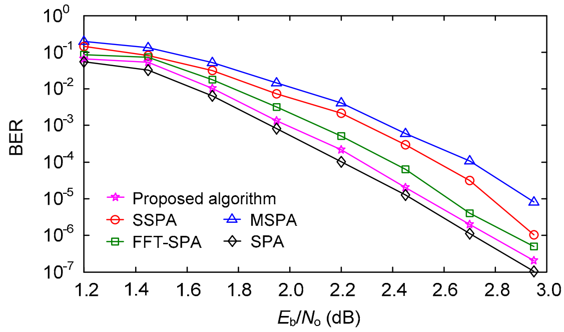

Fig. 3 Bit error rate performance comparisons for a (1296, 864) regular LDPC code under various SPA algorithms

The overall decoding performance comparisons of the proposed decoding algorithm with SPA, FFT-SPA, MSPA, and SSPA are shown in Table 1. The proposed decoding algorithm achieves decoding performance close to SPA by outperforming FFT-SPA, MSPA, and SSPA. Moreover, the proposed decoding algorithm achieves BER coding gain ranging from 0.04 to 0.46 dB when compared with FFT-SPA, MSPA, and SSPA.

Table 1 Overall decoding performance comparisons of various SPA algorithms

Decoding algorithm	SNR (dB)			BER		
	C_1*	C_2**	C_3*	C_1*	C_2**	C_3*
SPA	2.38	2.08	2.47	10^{-4}	10^{-5}	10^{-5}
Proposed	2.40	2.12	2.51	10^{-4}	10^{-5}	10^{-5}
FFT-SPA	2.44	2.18	2.58	10^{-4}	10^{-5}	10^{-5}
SSPA	2.49	2.37	2.79	10^{-4}	10^{-5}	10^{-5}
MSPA	2.59	2.39	2.97	10^{-4}	10^{-5}	10^{-5}

SNR: signal-to-noise ratio; BER: bit error rate. C_1: (N, K)=(648, 324); C_2: (N, K)=(2304, 1152); C_3: (N, K)=(1296, 864). * WLAN standard; ** Wi-MAX standard

4.2 Analysis of computational complexity

The total number of arithmetic operations required at the check node process determines the overall complexity of the decoding algorithm. The conventional SPA suffers from high computational complexity due to the utilization of the hyperbolic tangent function. Furthermore, SPA requires large amounts of look-up-tables to store the large set of processed soft information which is obtained during the iterative message passing process.

In the proposed algorithm, the hyperbolic tangent function is replaced by FFT_h without the requirement of reducing the nonlinear function to an approximated quantization table. The overall computation complexity comparison for the proposed algorithm, FFT-SPA, and the conventional SPA is shown in Table 2. The formulae for calculating the computational complexity of the conventional SPA, FFT-SPA, and the proposed algorithm are given as $d_c[4u^2+(d_c-2)u]$, $d_c[2(4u\log_2 u-6u+8)+(d_c-2)u]$, and $d_c[4u\log_2 u-6u+8+(d_c-1)u]$, respectively.

Table 2 Computational complexity comparison for SPA, FFT-SPA, and the proposed decoding algorithm

Alphabet size	Total number of arithmetic computations required		
	SPA	FFT-SPA	Proposed
2	144	96	84
4	480	288	216
8	1728	864	576

It can be observed from Table 2 that the proposed algorithm requires fewer arithmetic operations in the decoding process compared with FFT-SPA and SPA. Moreover, it is evident that the proposed algorithm achieves up to 67% and 42% reductions in the total number of arithmetic operations required for the decoding process when compared to SPA and FFT-SPA, respectively. It can be concluded from Table 2 that as the alphabet size increases, the computational complexity also increases gradually. However, the computational complexity of the proposed algorithm is still lower than those of the conventional SPA and FFT-SPA. In addition to the computational complexity, the proposed decoding algorithm achieves better decoding performance compared with SSPA, FFT-SPA, and MSPA for (648, 324), (2304, 1152), and (1296, 864) regular LDPC codes.

5 Conclusions

In this paper, an improved low-complexity sum-product decoding algorithm is presented for LDPC codes. In the proposed decoding algorithm, the high computational complexity issue in the check node process is sorted out by utilizing the FFT_h function. This modification in the check node update process significantly reduces the overall arithmetic computations required for updating CTV messages compared to SPA. However, the modification in the check node process degrades the BER performance by more than 0.1 dB compared to that of SPA. Therefore, to improve the decoding performance of the proposed algorithm, an optimized integer constant is employed in the variable node update process. In the variable node update process, the proposed algorithm completely utilizes the high-precision LLR information using an optimized weighted integer constant for improving the BER performance. The simulation results show that the proposed algorithm achieves an overall coding gain improvement of 0.04–0.46 dB. Moreover, when compared with SPA, the proposed decoding algorithm can reduce the total number of arithmetic operations required for the decoding process by 42%–67% for an alphabet size 2, 4, or 8. Therefore, the proposed algorithm with a simple decoding process and good error correcting performance is most suited for long-haul optical WLAN and Wi-MAX communication systems.

References

Chandrasetty, V.A., Aziz, S.M., 2011. FPGA implementation of a LDPC decoder using a reduced complexity message passing algorithm. *J. Netw.*, **6**(1):36-45. [doi:10.4304/jnw.6.1.36-45]

Chung, S.Y., Forney, G.D., Richardson, T.J., *et al.*, 2001. On the design of low-density parity-check codes within 0.0045 dB of the Shannon limit. *IEEE Commun. Lett.*, **5**(2):58-60. [doi:10.1109/4234.905935]

Fossorier, M.P.C., Mihaljevic, M., Imai, H., 1999. Reduced complexity iterative decoding of low-density parity check codes based on belief propagation. *IEEE Trans. Commun.*, **47**(5):673-680. [doi:10.1109/26.768759]

Gallager, R.G., 1962. Low-density parity-check codes. *IRE Trans. Inform. Theory*, **8**(1):21-28. [doi:10.1109/TIT.1962.1057683]

Goupil, A., Colas, M., Gelle, G., *et al.*, 2007. FFT-based BP decoding of general LDPC codes over Abelian groups. *IEEE Trans. Commun.*, **55**(4):644-649. [doi:10.1109/TCOMM.2007.894089]

He, Y.C., Sun, S.H., Wang, X.M., 2002. Fast decoding of LDPC codes using quantisation. *Electron. Lett.*, **38**(4):189-190. [doi:10.1049/el:20020131]

IEEE, 2009. IEEE Standard for Local and Metropolitan Area

Networks Part 16: Air Interface for Broadband Wireless Access Systems. IEEE Std 802.16-2009. [doi:10.1109/IEEESTD.2009.5062485]

IEEE, 2015. IEEE Draft Standard for Information Technology—Telecommunications and Information Exchange Between Systems Local and Metropolitan Area Networks—Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. IEEE P802.11-REVmc/D4.0.

Jiang, X., Lee, M.H., Qi, J., 2012. Improved progressive edge-growth algorithm for fast encodable LDPC codes. *EURASIP J. Wirel. Commun. Netw.*, **2012**:178.1-178.10. [doi:10.1186/1687-1499-2012-178]

Johnson, S.G., Frigo, M., 2007. A modified split-radix FFT with fewer arithmetic operations. *IEEE Trans. Signal Process.*, **55**(1):111-119. [doi:10.1109/TSP.2006.882087]

Lee, M.H., Han, J.H., Sunwoo, M.H., 2008. New simplified sum-product algorithm for low complexity LDPC decoding. IEEE Workshop on Signal Processing Systems, p.61-66. [doi:10.1109/SIPS.2008.4671738]

MacKay, D.J.C., Neal, R.M., 1997. Near Shannon limit performance of low density parity check codes. *Electron. Lett.*, **33**(6):457-458. [doi:10.1049/el:19970362]

Morello, A., Mignone, V., 2006. DVB-S2: the second generation standard for satellite broad-band services. *Proc. IEEE*, **94**(1):210-227. [doi:10.1109/JPROC.2005.861013]

Papaharalabos, S., Sweeney, P., Evans, B.G., *et al*., 2007. Modified sum-product algorithm for decoding low-density parity-check codes. *IET Commun.*, **1**(3):294-300. [doi:10.1049/iet-com:20060173]

Richardson, T.J., Urbanke, R.L., 2001. The capacity of low-density parity-check codes under message-passing decoding. *IEEE Trans. Inform. Theory*, **47**(2):599-618. [doi:10.1109/18.910577]

Safarnejad, L., Sadeghi, M.R., 2012. FFT based sum-product algorithm for decoding LDPC lattices. *IEEE Commun. Lett.*, **16**(9):1504-1507. [doi:10.1109/LCOMM.2012.073112.120996]

Sorensen, H.V., Heideman, M., Burrus, C.S., 1986. On computing the split-radix FFT. *IEEE Trans. Acoust. Speech Signal Process.*, **34**(1):152-156. [doi:10.1109/TASSP.1986.1164804]

Yuan, L., Tian, X., Chen, Y., 2011. Pruning split-radix FFT with time shift. Proc. Int. Conf. on Electronics, Communications and Control, p.1581-1586. [doi:10.1109/ICECC.2011.6066654]