Michaelraj Kingston Roberts, Ramesh Jayabalan, 2015. An improved low-complexity sum-product decoding algorithm for low-density parity-check codes. *Frontiers of Information Technology & Electronic Engineering*, **16**(6):511-518. [doi:10.1631/FITEE.1400269]

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

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

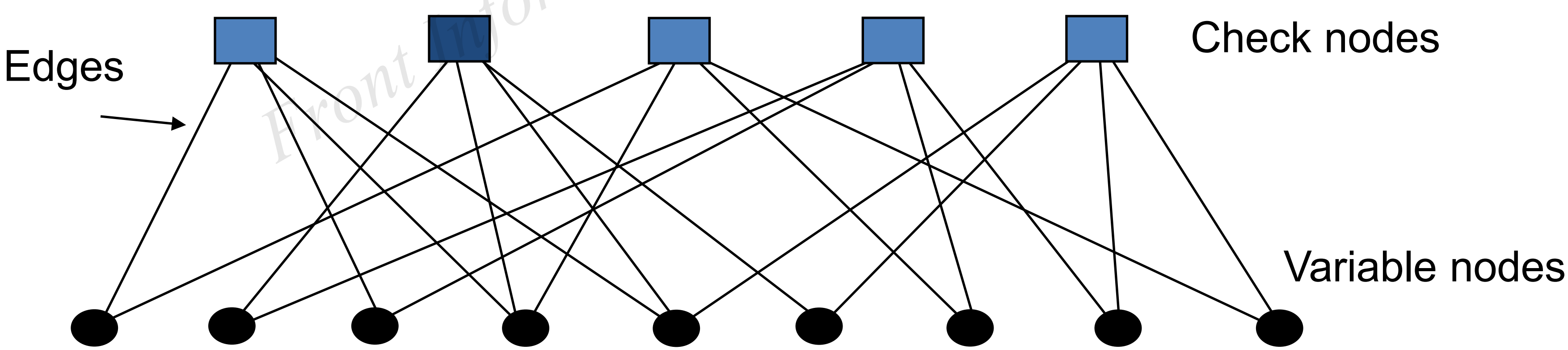
Contact: Michaelraj Kingston Roberts
E-mail: king.pane@gmail.com
ORCID: http://orcid.org/0000-0002-1484-703X

Introduction

- Error control codes (ECC) in communication systems are becoming a standard requirement for reliable transmission and storage of large amounts of data.
- 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.
- The only difference between the LDPC codes and other block codes is the amount of the sparseness of the non-zero elements in the parity-check matrix.
- The number of non-zero entries in each row or column of the parity-check matrix is collectively known as the degree distribution.

Tanner graph representation

- Originally formulated by Tanner in 1981.
- The Tanner graph is a bi-partite graph with two types of nodes called variable nodes and check nodes.
- The variable nodes and check nodes of the Tanner graph correspond to the codeword bits and parity-check bits, respectively.
- An edge joins a variable node to a check node if that bit is included in the corresponding parity-check equation.
- The edges connecting the nodes represent the non-zero elements in the parity-check matrix.



Bi-partite graph or Tanner graph representation of an LDPC code

Objective

• To propose an improved low-complexity sum-product algorithm to achieve good trade-off between decoding performance and computational complexity.

Methodology

• In order to achieve good error correcting performance without increasing the computational complexity, modifications have been carried out in both the check node and variable node processes.
• In the check node update process, the hyperbolic tangent function is replaced by the Fast Fourier transform with time shift.
• In the variable node process, an optimized integer constant is incorporated to enhance the decoding performance of the proposed algorithm.

Related works

Decoding algorithms for LDPC codes:

1) Sum-product algorithm (SPA) (Fossorier *et al*., 1999)
2) Modified SPA (Papaharalabos *et al*., 2007)
3) Simplified SPA (Lee *et al*., 2008)
4) Fast Fourier Transform based SPA (Safarnejad and Sadeghi, 2012)

Decoding steps

Initialization: set $q_{nm}^{(0)} = 2y_i / \sigma^2$ and $\sigma^2 = (2R_\mathrm{c} \cdot E_\mathrm{b}/N_\mathrm{o})^{-1}$

Step 1 (check node update):

$$r_{mn}^{(k)} = \prod_{i \in \{N(m) \backslash n\}} \mathrm{sign}(y_i) \cdot \mathrm{FFT}_\mathrm{h}^{-1}\left[\prod_{i \in \{N(m) \backslash n\}} \mathrm{FFT}_\mathrm{h}\left(\frac{2|y_i|}{\sigma^2}\right)\right]$$

Step 2 (variable node update):

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

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

Decoding steps (Con'd)

where $q'_{nm} = q^{(0)}_{nm} + \sum X_i$, $X_i = \begin{cases} M/2+1, & m_i = 0, \\ (M+1)/2, & m_i = 1, \end{cases}$ q_i=sign(q_{nm}−X_i),

$S_m = \sum_{n=0}^{N-1} h_{m,n} \cdot \hat{c}_n \pmod 2$ for m=1, 2, ..., M and $\hat{c}_n$ denotes the hard decision sequence

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., $\hat{\boldsymbol{c}} \cdot \boldsymbol{H}^{\mathrm{T}} = \boldsymbol{0}$.

Simulation results

Decoding performance:

- The LDPC codes were designed by the procedure as described by Jiang *et al.* (2012).
- To illustrate decoding performance of the proposed algorithm, (N, K) regular LDPC codes belonging to WLAN standard (IEEE, 2015) and Wi-MAX standard (IEEE, 2009) have been considered.
- For all simulation process of this work, the maximum number of decoding iterations was chosen to be 20 (Lee *et al*., 2008).

Decoding performance

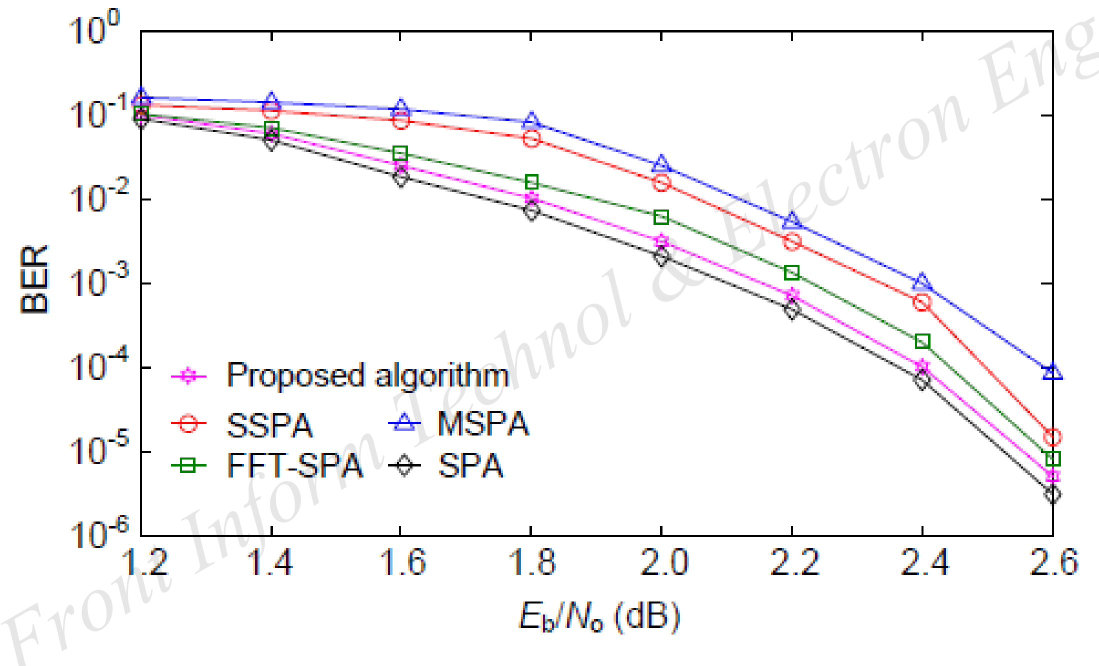


Fig. 1 Bit error rate performance comparisons for a (648, 324) regular LDPC code under various SPA algorithms

Decoding performance (Con'd)

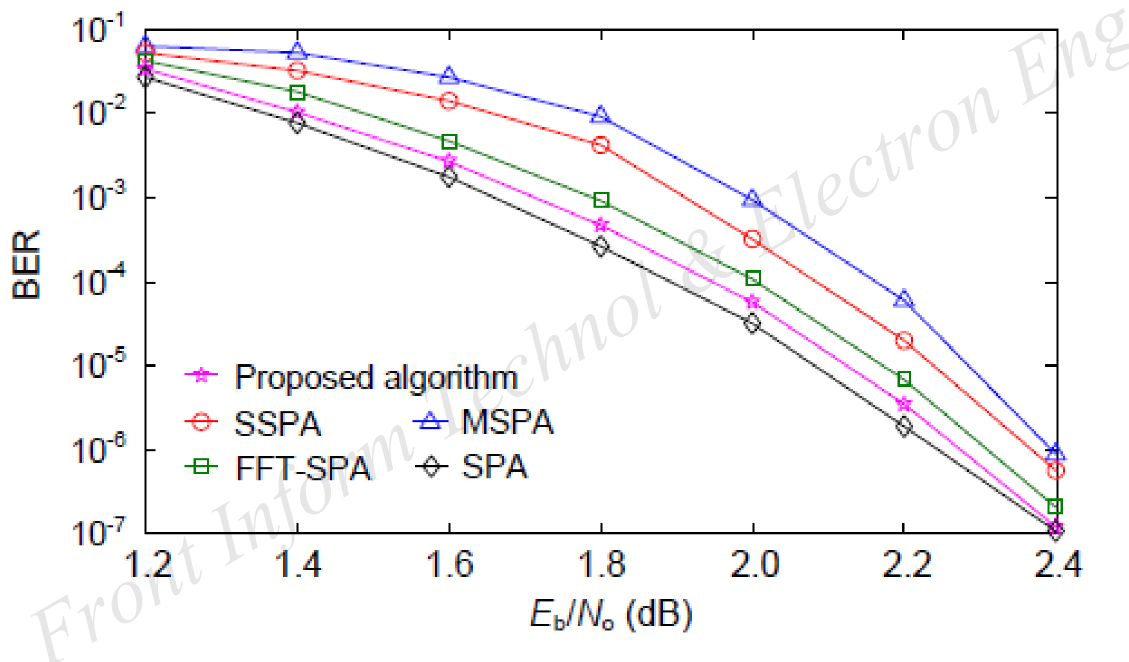


Fig. 2 Bit error rate performance comparisons for a (2304, 1152) regular LDPC code under various SPA algorithms

Decoding performance (Con’d)

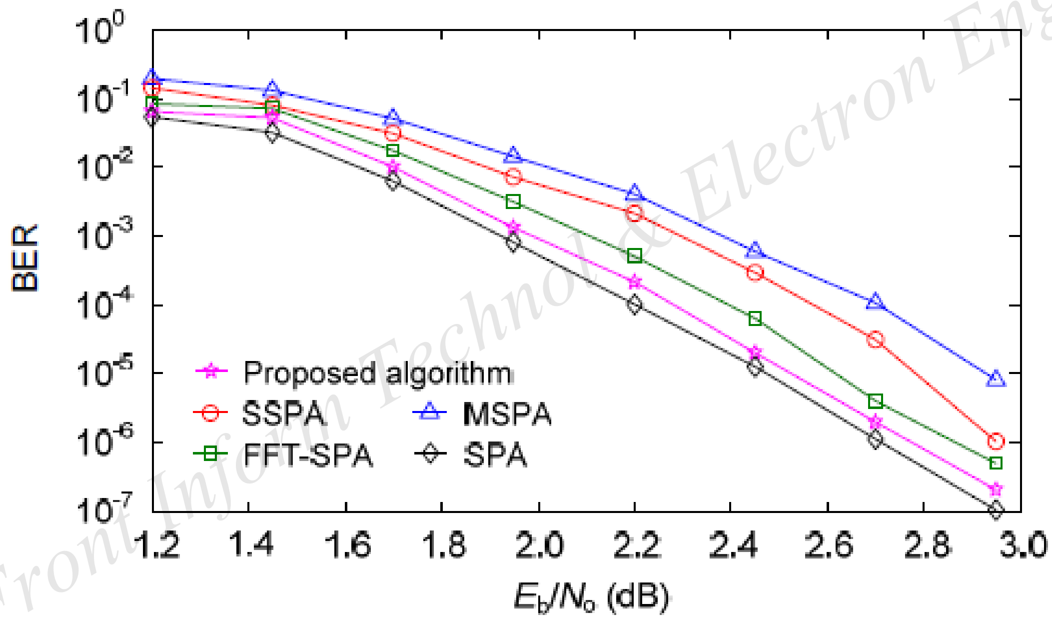


Fig. 3 Bit error rate performance comparisons for a (1296, 864) regular LDPC code under various SPA algorithms

Decoding performance comparison

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

Overall coding gain improvement comparison

Decoding algorithm	Coding gain improvement					
	WLAN standard				Wi-MAX standard	
	(648, 324)		(1296, 864)		(2304, 1152)	
	SNR (dB)	BER	SNR (dB)	BER	SNR (dB)	BER
FFT-SPA	0.04	10^{-4}	0.10	10^{-5}	0.06	10^{-5}
SSPA	0.09	10^{-4}	0.28	10^{-5}	0.25	10^{-5}
MSPA	0.19	10^{-4}	0.46	10^{-5}	0.27	10^{-5}

Computational complexity analysis

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

Conclusions

- An improved low-complexity SPA is proposed and analyzed to reduce the computational complexity of the conventional SPA and its variants.
- Simulation results show that the proposed algorithm achieves an overall coding gain improvement of 0.04–0.46 dB.
- Furthermore, the proposed 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 can achieve good error correcting performance close to SPA without increasing the computational complexity.