Safe Early Stopping for Layered LDPC Decoding

M. Ferrari, *Member, IEEE*, S. Bellini, and A. Tomasoni, *Member, IEEE*

I. Introduction

BELIEF propagation (BP) decoding of LDPC codes and its simplified versions Min-Sum (MS), scaled-MS or offset-MS decoding require iterative activation of the Tanner graph nodes to increase the confidence about the transmitted symbols and correct the errors. Different received blocks may require a different number of iterations to be corrected. Stopping criteria able to understand when further decoding iterations are useless may help saving power consumption, processing time, or enable a larger maximum number of iterations.

Existing literature has tackled this issue, and proposed different solutions, both for LDPC and Turbo Codes (TC), based on hard or soft decisions. Stopping criteria based on hard-decisions generally test that the variable estimates get stable [1], or satisfy the Check Nodes (CNs) [2], or match both criteria [3]. The test can be applied using the CN input or posterior LLR L_p signs [4]. In case of *layered* (or *serial-C*) decoding [5], the test can be performed one CN after the other, or at the end of each iteration. Stopping criteria based on soft-decisions, e.g., triggered by the average or the minimum $|L_p|$ [6], [7], or by their Cross-Entropy [8] are often too complex to implement. Recently, a stop criterion for shuffled (or serial-V) decoding has been proposed in [9]. All these stopping criteria, being heuristic, are verified by simulation only, at medium bit error rates (BER).

In this letter we focus on *safe early stopping criteria*, that halt LDPC decoding when decisions will not change with further iterations. These criteria can be applied at very low target BERs, with the guarantee that they do not affect the error rate. In [10] the authors propose a *completely safe* early termination strategy for an M-dimensional turbo code. In this letter, we first apply that safe stopping criterion to layered LDPC decoding. Then, we propose a criterion more *efficient* than [10], in the sense that it can save more iterations. Finally, we propose two simplified safe criteria that are less efficient, but easier to test in practical LDPC implementations.

Manuscript received July 16, 2014; accepted December 19, 2014. Date of publication December 30, 2014; date of current version March 6, 2015. The associate editor coordinating the review of this paper and approving it for publication was H. Saeedi.

M. Ferrari and A. Tomasoni are with the Institute of Electronics, Computer and Telecommunication Engineering (IEIIT), National Research Council (CNR), 20133 Milan, Italy (e-mail: marco.ferrari@ieiit.cnr.it; alessandro.tomasoni@ieiit.cnr.it).

S. Bellini is with the Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, 20133 Milan, Italy (e-mail: sandro.bellini@polimi.it).

II. System Model and Properties

Layered LDPC decoding exhibits implementation advantages and fast convergence [5]. In layered decoding, the N_c CNs are activated sequentially, and each Variable Node (VN) of degree d is updated d times per iteration. Let $O_k^{(n)}$ represent the nth update of the *a posteriori* LLR of the kth VN. If we allow N_i decoding iterations, the index n evolves from 0 up to $d \cdot N_i - 1$. The nth estimate of the kth variable is

$$S_k^{(n)} \triangleq \operatorname{sign}\left(O_k^{(n)}\right) \tag{1}$$

where $\operatorname{sign}(0) = 0$. Let $\mathcal{S}_c$ be the set of VNs connected to the cth CN of degree d_c. In general each VN $k \in \mathcal{S}_c$ has its own degree d_k and update index n_k at each activation of CN c. To keep notation simple, we start considering a regular LDPC code ($d_k = d$, $\forall k$) with parity check matrix arranged in stripes as in [5]. The CN decoder is fed by

$$I_k^{(n)} \triangleq O_k^{(n-1)} - E_k^{(n-d)} \tag{2}$$

where $E_k^{(n-d)}$ is the extrinsic message generated by the CN at the previous iteration. The CN messages $E_k^{(n)}$ are eventually combined with the inputs, to achieve the updated $O_k^{(n)}$

$$O_k^{(n)} = I_k^{(n)} + E_k^{(n)}. \tag{3}$$

The update equation can also be written as

$$O_k^{(n)} = O_k^{(n-1)} + E_k^{(n)} - E_k^{(n-d)} = O_k^{(n-1)} + \Delta E_k^{(n)}. \tag{4}$$

We also define the signs

$$s_k^{(n)} \triangleq \operatorname{sign}\left(I_k^{(n)}\right), \quad \sigma_k^{(n)} \triangleq \operatorname{sign}\left(E_k^{(n)}\right). \tag{5}$$

MS decoding usually replaces BP in practical implementations. The minimum input reliability can be corrected by a scale factor $\alpha \leq 1$ or by an offset $\beta \geq 0$ [11] to give the CN output. In one formula the extrinsic reliability reads

$$\left|E_k^{(n)}\right| = \max\left(0, \alpha \cdot \min_{l \neq k, l \in \mathcal{S}_c} \left|I_l^{(n)}\right| - \beta\right). \tag{6}$$

The sign of the extrinsic message reads

$$\sigma_k^{(n)} = \begin{cases} \prod_{l \neq k, l \in \mathcal{S}_c} s_l^{(n)} & \text{if } \left|E_k^{(n)}\right| > 0 \\ 0 & \text{if } \left|E_k^{(n)}\right| = 0. \end{cases} \tag{7}$$

Finite precision implementations require rounding or truncation, and saturation to a maximum value. We collect all these effects, including scaling or offset of (6), in a function f

$$\left|E_k^{(n)}\right| = f\left(\min_{l\neq k, l\in\mathcal{S}_c}\left|I_l^{(n)}\right|\right). \tag{8}$$

We assume that the following property holds.

Property 1: The scalar function $f(x)$ in (8) is monotonically non-decreasing, and $0 \leq f(x) \leq x$. Therefore

$$\left|E_k^{(n)}\right| \leq \min_{l\neq k, l\in\mathcal{S}_c}\left|I_l^{(n)}\right|. \tag{9}$$

The inequality (9) is obvious for (6). If $E_k^{(n)}$ and $I_l^{(n)}$ are quantized in the same alphabet, rounding, ceiling, flooring and saturation do not change the inequality. Besides, (8) is the composition of monotonic functions (minimum, scaling, offset, rounding, ceiling, flooring and saturation), and is monotonic. Almost all practical CN decoders proposed so far fit this model. An exception is the conditional offset MS [11], for which monotonicity is not guaranteed.

A final remark concerns the behavior of a CN decoder obeying (4)–(5)–(7)–(8) and Property 1. If the input LLRs satisfy the CN, the output signs match the inputs and the extrinsic messages can not disagree, i.e.,

$$\prod_{l\in\mathcal{S}_c} s_l^{(n)} > 0 \quad \Rightarrow s_k = S_k, \quad \sigma_k S_k \geq 0. \tag{10}$$

A less obvious property is the following.

Property 2: If a CN is satisfied by the outputs $O_k^{(n)}$ but not by the inputs $I_k^{(n)}$, the only sign mismatch $s_j^{(n)} \neq S_j^{(n)}$ affects the minimum $|I_j^{(n)}|$. Elsewhere $s_k^{(n)} = S_k^{(n)}$.

Proof: Since, by hypothesis,

$$\prod_{l\in\mathcal{S}_c} S_l^{(n)} > 0, \quad \prod_{l\in\mathcal{S}_c} s_l^{(n)} \leq 0 \tag{11}$$

in the update (3) an odd number of variables change their sign. For these variables $|E_k^{(n)}| > |I_k^{(n)}|$. According to (9), this can happen only for one variable, say the jth, with the minimum $|I_j^{(n)}|$. For $k \neq j$, $s_k^{(n)} = S_k^{(n)}$.

III. Safe Early Termination Strategies

A heuristic early stopping criterion requires that the outputs satisfy the CNs and remain unchanged for at least one iteration. We call this criterion *CN Satisfaction and Sign Stability*, namely **SS**

$$\mathbf{SS}: \begin{cases} S_k^{(m)} = S_k^{(m-1)} = \ldots = S_k^{(m-d+1)}, & \forall k \\ \prod_{k\in\mathcal{S}_c} S_k^{(m)} = 1, & \forall c. \end{cases} \tag{12}$$

Condition **SS** is not *safe*: we have verified that the posterior LLRs can identify a codeword, be stable for a whole iteration, and still change signs during subsequent iterations.

As in [10], we obtain a safe criterion if we add the condition that the growth of the extrinsic messages in the last $d-1$ updates *supports in sign* the *a-posteriori* decisions:

$$\mathbf{D}: \Delta E_k^{(m-i)} S_k^{(m-i)} \geq 0, \quad \forall k, i = 0, 1, \ldots, d-2. \tag{13}$$

The following Theorem can be stated.

Theorem 1: During layered decoding of an LDPC code, if at sub-iteration $n-1$ conditions **SS** and **D** hold, further iterations will only reinforce the decoder decisions, i.e.,

$$\mathbf{SSD} \Rightarrow \begin{cases} S_k^{(m)} = S_k^{(n-1)} \\ \left|O_k^{(m)}\right| \geq \left|O_k^{(n-1)}\right|, \end{cases} \quad \forall m > n-1. \tag{14}$$

Proof: We will show that if **SSD** holds at sub-iteration $n-1$, it holds also at sub-iteration n, i.e.,

$$S_k^{(n)} = S_k^{(n-1)}, \quad \Delta E_k^{(n)} S_k^{(n)} \geq 0, \quad \forall k. \tag{15}$$

Besides, if **SSD** holds at sub-iteration n, then

$$\left|O_k^{(n)}\right| = S_k^{(n)} O_k^{(n)} = S_k^{(n)}\left(O_k^{(n-1)} + \Delta E_k^{(n)}\right) \geq \left|O_k^{(n-1)}\right|. \tag{16}$$

Thus the Theorem will be eventually proven by induction.

From the update rules (2)–(4) we have

$$I_k^{(n)} = I_k^{(n-d)} + \sum_{m=n-d+1}^{n-1} \Delta E_k^{(m)}. \tag{17}$$

Thus by **D** and **SS** we have

$$\left(I_k^{(n)} - I_k^{(n-d)}\right) S_k^{(n-d)} = \sum_{m=n-d+1}^{n-1} \Delta E_k^{(m)} S_k^{(n-d)} \geq 0. \tag{18}$$

This means that for all inputs $I_k^{(n-d)}$ that agreed in sign with $S_k^{(n-d)}$, the sign is confirmed and the reliability is increased (i.e., $|I_k^{(n)}| \geq |I_k^{(n-d)}|$). Conversely if $s_j^{(n-d)} \neq S_j^{(n-d)}$ either the sign $s_j^{(n)}$ at iteration n is correct or the mismatch survives.

We need to discuss only two cases: messages $E_k^{(n-d)}$ generated by $d_c - 1$ inputs without sign mismatches (i.e., $s_l^{(n-d)} = S_l^{(n-d)}$, $\forall l \neq k$), and messages $E_k^{(n-d)}$ with some mismatched input.

In the former case, since $|I_l^{(n)}| \geq |I_l^{(n-d)}|, \forall l \neq k$, we have $|E_k^{(n)}| \geq |E_k^{(n-d)}|$ by Property 1, and $\sigma_k^{(n-d)} S_k^{(n-d)} \geq 0$, $\sigma_k^{(n)} S_k^{(n-d)} \geq 0$ by **SS** and (10). Therefore

$$\Delta E_k^{(n)} S_k^{(n-d)} = \left|E_k^{(n)}\right| - \left|E_k^{(n-d)}\right| \geq 0. \tag{19}$$

In the latter case, by Property 2 and **SS** there is just one mismatch, say $s_j^{(n-d)} \neq S_j^{(n-d)}$. We have

$$\begin{aligned} \Delta E_k^{(n)} S_k^{(n-d)} &= S_k^{(n-d)} E_k^{(n)} - S_k^{(n-d)} E_k^{(n-d)} \\ &= S_k^{(n-d)} E_k^{(n)} + \left|E_k^{(n-d)}\right|. \end{aligned} \tag{20}$$

because $\sigma_k^{(n-d)} \neq S_k^{(n-d)}$. The first addend can be negative, but by Property 2 the input with $s_j^{(n)} \neq S_j^{(n-d)}$ is the one with the minimum reliability and $|I_j^{(n)}| \leq |I_j^{(n-d)}|$ by (18). This, in turn, means that $|E_k^{(n)}| \leq |E_k^{(n-d)}|$, by Property 1. Therefore $\Delta E_k^{(n)} S_k^{(n-d)} \geq 0$ also in this case.

Finally, by **SS** at time $n-1$ and (4),

$$S_k^{(n-d)} O_k^{(n)} = \left|O_k^{(n-1)}\right| + S_k^{(n-d)} \Delta E_k^{(n)} \geq \left|O_k^{(n-1)}\right| > 0 \tag{21}$$

where the strict inequality comes from CN satisfaction. From the above equation we conclude that $S_k^{(n-d)} = S_k^{(n)}$. Thus $\Delta E_k^{(n)} S_k^{(n)} \geq 0, \forall k$, i.e., **SSD** holds at sub-iteration n.

Note that with MS and scaled-MS decoding (without quantization and saturation), Theorem 1 is in agreement with [10]. In fact, layered decoding of LDPC codes may also be seen as Turbo Decoding of a Serially Concatenated TC with N_c Single Parity Check component codes. Besides, MS and scaled-MS behave like Max-Log and scaled Max-Log in TC, if **SS** holds.

Carefully inspecting the proof of Theorem 1, it is apparent that condition **D** is not necessary. It is enough that

$$\sum_{m=n-d+1}^{n-1} \Delta E_k^{(m)} S_k^{(m)} \geq 0. \tag{22}$$

In fact, by repeatedly applying (4) we obtain $O_k^{(n-1)} - O_k^{(m)} = \sum_{t=m+1}^{n-1} \Delta E_k^{(t)}$. Then, condition **D** can be replaced by

$$\mathbf{D2}: \left(O_k^{(m)} - O_k^{(m-i)}\right) S_k^{(m)} \geq 0, \qquad \forall k, i = 1 \ldots d-1 \tag{23}$$

and the following theorem holds.

Theorem 2: During layered decoding of an LDPC code, if at sub-iteration $n-1$ conditions **SS** and **D2** hold, then further iterations will only reinforce the decisions, i.e.,

$$\mathbf{SSD2} \Rightarrow \begin{cases} S_k^{(m)} = S_k^{(n-1)} \\ \left|O_k^{(m)}\right| \geq \left|O_k^{(n-1)}\right|, \end{cases} \qquad \forall m > n-1. \tag{24}$$

With irregular LDPC codes, or regular codes not arranged in stripes as in [5], it is no longer guaranteed that all VNs have the same degree d and the same update index n in (8). The kth VN has degree d_k, while CN irregularity is implicitly managed by the set $\mathcal{S}_c$. In (8), at each activation, the CN decoder exploits the most recently updated input messages $I_l^{(n_l(k,n))}$, where $n_l(k,n)$ counts the number of updates of variable l, till the nth update of variable k. Of course these indexes run with the iterations i at proper rate d_k, i.e., $\lfloor n_k/d_k \rfloor = \lfloor n_l/d_l \rfloor = i$. Equations (6) and (9) need the same generalization. In (7), (10), (11), and (12) that link LLRs from different VNs, we need to replace the common n or m with specific update indexes n_k, n_l, m_k. These are only formal changes. Theorems 1 and 2 hold even with a different update index for each VN.

IV. Reduced Complexity Safe Stopping Criteria

In practical implementations, condition **D2** is not easy to check. The last $d-1$ updates should be stored for each VN to evaluate $O_k^{(n)} - O_k^{(n-d+1)}$. Condition **D** is easier as $O_k^{(n)} - O_k^{(n-1)}$ at each update matches the difference between the new extrinsic $E_k^{(n)}$ and the old one $E_k^{(n-d)}$ that is already needed to evaluate (2) during the CN update. In the following we will also work on the simplification of condition **D**, but let us first focus on testing **SS**.

To test condition **SS** the d_c signs should be carried on during CN processing, and compared to the new ones when CN processing is completed. A condition easier to check during layered decoding is whether the signs of the CN inputs match the old outputs and satisfy the CNs. The satisfaction of the CN by the input signs is tested during the extrinsic sign evaluation, and the condition $s_k^{(n)} = S_k^{(n-1)}$ can be checked during *extrinsication* (2) of the LLRs. The first condition should be tested for the last $d-1$ updates, whereas the latter needs be true for the last d updates. We name it *condition* **SSi**.

$$\mathbf{SSi}: \begin{cases} s_k^{(m-i)} = S_k^{(m-i-1)}, & \forall k,\ i = 0, 1 \ldots d-2 \\ \prod_{k \in \mathcal{S}_c} s_k^{(m-i)} = 1, & \forall c \end{cases} \tag{25}$$

TABLE I

Safe Criteria Complexity

	SSiB	SSiA	SSD	SSD2
N_{XOR}/N_v	$2d + R - 2$		$3d + R - 3$	
N_{ADD}/N_v	$(1-R)(1-d^{-1})$	$3(1-R)(1-d^{-1})$	$d-1$	
N_{MEM}/N_v	0		1	$1 + (d-1)N_q$

Note that if **SSi** holds at time m, then $S_k^{(m)} = S_k^{(m-1)} = \ldots S_k^{(m-d+1)}$, i.e., also **SS** holds at time m, and the following corollary holds.

Corollary 1: During layered decoding of an LDPC code, if at sub-iteration $n-1$ conditions **SSi** and **D** hold, then further iterations will only reinforce the decisions, i.e.,

$$\mathbf{SSiD} \Rightarrow \begin{cases} S_k^{(m)} = S_k^{(n-1)} \\ |O_k^{(m)}| \geq |O_k^{(n-1)}|, \end{cases} \qquad \forall m > n-1. \tag{26}$$

With the aim of reducing the complexity of the test for condition **D**, note that the signs of $E_k^{(m)}$ support $S_k^{(m)}$ under condition **SSi**. Thus, a condition simpler to test than **D** is

$$\mathbf{A}: |E_k^{(m-i)}| \geq |E_k^{(m-d-i)}|, \qquad \forall k, i = 0, 1, \ldots, d-2. \tag{27}$$

Also note that for all practical CN decoders that obey (8), $|E_k^{(n)}| \in \{\mu^{(n)}, \mu'^{(n)}\}, \forall k \in \mathcal{S}_c$ with $\mu^{(n)} \leq \mu'^{(n)}$. Then, condition **A** can be tested with few comparisons between unsigned values. An even simpler condition is

$$\mathbf{B}: \min_k \left|E_k^{(m-i)}\right| \geq \max_k \left|E_k^{(m-d-i)}\right|, \quad \forall k, i = 0, 1 \ldots d-2. \tag{28}$$

which requires just one comparison between unsigned values, namely $\mu^{(m)} \geq \mu'^{(m-d)}$. Since $\mathbf{B} \Rightarrow \mathbf{A}$ we have the following Corollary.

Corollary 2: During layered decoding of an LDPC code, if at sub-iteration $n-1$ condition **SSi** holds, and also **B** or at least **A** holds, then further iterations will only reinforce the decisions, i.e.,

$$\mathbf{SSiB} \Rightarrow \mathbf{SSiA} \Rightarrow \begin{cases} S_k^{(m)} = S_k^{(n-1)} \\ \left|O_k^{(m)}\right| \geq \left|O_k^{(n-1)}\right|, \end{cases} \qquad \forall m > n-1.$$

In Table I we compare the number of sign comparisons (N_{XOR}), differences (N_{ADD}) and memory bits (N_{MEM}) required to test the four safe criteria presented so far.[1]

V. Simulation

The efficiency of the discussed stopping criteria has been tested by simulation for a large ($N_v \approx 30$ kb), high rate (4/5) regular (3,15) code, and two small ($N_v = 1$ kb), rate 1/2, irregular codes. To compare the *safe* criteria proposed, we have tested also the most popular unsafe stopping criteria. Criterion **S** verifies that all CNs are satisfied by the messages O_k at the end of an iteration. Criterion $\mathbf{S_d}$ verifies that all CNs are satisfied by the messages $O_k^{(n)}$ at their processing time. Criterion **SS** requires CN satisfaction and sign stability as defined in (12).

Fig. 1(a) refers to the large LDPC code with rate $R = 4/5$. We plot the probability that the number of iterations exceeds

[1] N_q is the number of quantization bits used to store each O_k, d is the VN degree and R is the code rate. Complexity is normalized by N_v, to make it independent of the block size. N_{XOR} and N_{ADD} refer to just one iteration.

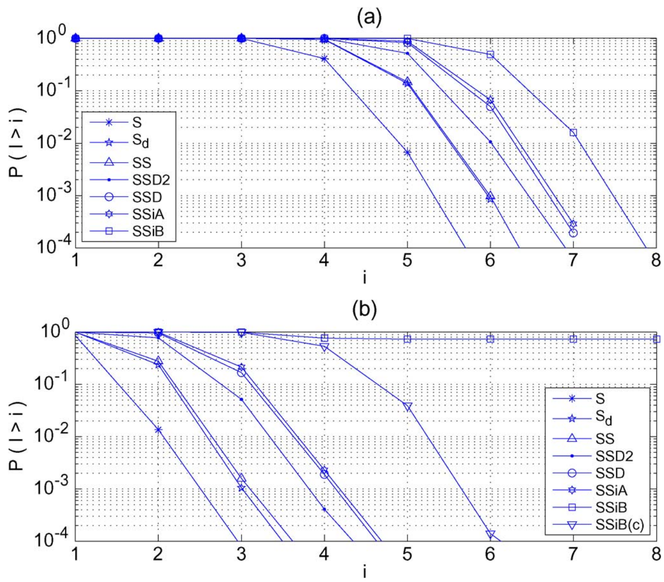

Fig. 1. Probability distribution of the number of iterations required by the scaled-MS decoder ($\alpha = 3/4$) to meet the various early stopping criteria: (a) $R = 0.8$, $SNR = 2.5$ dB; (b) $R = 0.5$, $SNR = 2$ dB.

the abscissa i before the stopping criterion is met. Obviously, **S** is the most efficient criterion. $\mathbf{S_d}$ is less efficient than **S** and only slightly more efficient than **SS**. In practice, $\mathbf{S_d}$ becomes true only after sign stability is achieved.

SSD2, the most efficient *safe* criterion, requires half an iteration more than **SS**. Surprisingly, **SSD** and **SSiA** have the same efficiency, and require less than half iteration more than **SSD2**. This makes **SSiA** an interesting safe solution, requiring only one iteration more than $\mathbf{S_d}$ or **SS**. The simpler **SSiB** criterion needs one additional iteration.

In Fig. 1(b) we plot the cumulative probability distribution of the number of iterations for a small ($N_v = 1000$), $R = 1/2$, irregular LDPC code of polynomials $\lambda(x) = 0.636x^2 + 0.364x^{14}, \rho(x) = 0.502x^7 + 0.498x^8$ and threshold $E_s/N_0 = 0.9$ dB. Many of the previous comments are still valid. The number of iterations is lower than for the first code, due to the small block. The most interesting difference is that criterion **SSiB** almost always fails to halt decoding. This is due to irregularity of the VN degrees. For each CN, $\min_k |E_k^{(n)}|$ is very likely to belong to the set of VNs of degree 3, whereas $\max_k |E_k^{(n)}|$ has degree 15. Then it is very unlikely that the minimum grows larger than the previous maximum in just one iteration, as requested by (28). We have designed also an irregular LDPC with less VN degree dispersion, namely $\lambda(x) = 0.928x^2 + 0.072x^4, \rho(x) = 0.332x^4 + 0.668x^6$. In this case **SSiB** works, despite the (mild) irregularity, as shown by the curve labeled **SSiB(c)** in Fig. 1(b). In conclusion, **SSiB** works only with regular, or almost regular, LDPC codes.

Unsafe stopping criteria may halt the decoder with errors still correctable by successive iterations, thereby increasing the BER. The amount of this increase depends on the LDPC code. When **S** or **SS** is tested, decoding is stopped only by valid codewords, thus errors due to termination practically occur only in the error-floor region. We have chosen a small ($N_v = 240$), high rate ($R = 0.825$), LDPC code with small minimum Hamming distance ($d_H = 4$) and high error-floor. Fig. 2 shows that a BER difference occurs when unsafe criteria (**S**, **SS**) are

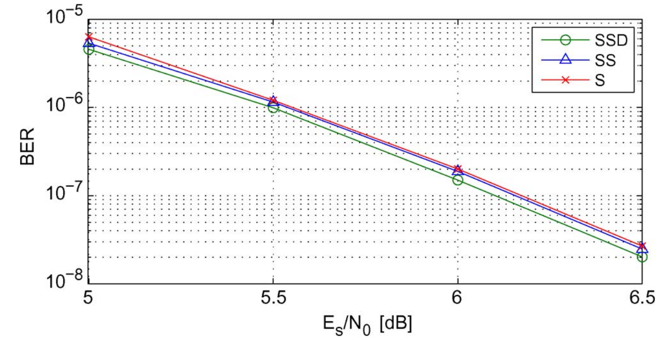

Fig. 2. Example of performance of decoders with *safe* versus *unsafe* stopping criteria (Reed Solomon-LDPC code, $R = 0.825$, $N_v = 240$, $d_H = 4$).

applied in place of safe ones (**SSD**). In this case the differences are small. In general we do not know without simulation. The real advantage of safe stopping criteria is that no validation is needed, since they are guaranteed not to affect the decoder performance. This is useful in particular with very low error floors, when it is hard to test the impairment coming from unsafe stopping criteria.

VI. Conclusion

We have stated and proven a safe early stopping criterion for layered LDPC decoding (**SSD**). A more efficient criterion (**SSD2**) and two less efficient but less complex, still safe stopping criteria (**SSiA**, **SSiB**) are obtained from **SSD**.

For three different LDPC codes, we have simulated these criteria and compared their efficiency, also against CN-based criteria, that are simpler but unsafe.

References

[1] J. Li, G. He, H. Hou, Z. Zhang, and J. Ma, "Memory efficient layered decoder design with early termination for LDPC codes," in *Proc. IEEE ISCAS*, Rio de Janeiro, Brazil, May 2011, pp. 2697–2700.

[2] Y. Ueng, Y. Wang, C. Lin, J. Hsu, and P. Ting, "Modified layered message passing decoding with dynamic scheduling and early termination for QC-LDPC codes," in *Proc. IEEE ISCAS*, Taipei, Taiwan, May 2009, pp. 121–124.

[3] M. Jiang and J. Dou, "An efficient stopping criterion of LDPC decoding for optical transmission," in *Proc. 9th ICOCN*, Nanjing, China, Oct. 2010, pp. 275–278.

[4] A. Darabiha, A. Carusone, and F. R. Kschischang, "Power reduction techniques for LDPC decoders," *IEEE J. Solid-State Circuits*, vol. 43, no. 8, pp. 1835–1845, Aug. 2008.

[5] M. M. Mansour and N. R. Shanbhag, "High-throughput LDPC decoders," *IEEE Trans. VLSI Syst.*, vol. 11, no. 6, pp. 976–996, Dec. 2003.

[6] F. Kienle and N. Wehn, "Low complexity stopping criterion for LDPC code decoders," in *Proc. IEEE 61st VTC*, Stockholm, Sweden, May 2005, pp. 606–609.

[7] J. Li, X. You, and W. Li, "Early stopping for LDPC decoding: Convergence of mean magnitude (CMM)," *IEEE Commun. Lett.*, vol. 10, no. 9, pp. 667–669, Sep. 2006.

[8] L. Kocarev *et al.*, "Nonlinear dynamics of iterative decoding systems: Analysis and applications," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1366–1384, Apr. 2006.

[9] X. Liu, J. Cai, and L. Wu, "Improved decoding algorithm of serial belief propagation with a stop updating criterion for LDPC codes and applications in patterned media storage," *IEEE Trans. Magn.*, vol. 49, no. 2, pp. 829–836, Feb. 2013.

[10] A. Hunt, S. Crozier, K. Gracie, and P. Guinand, "A completely safe early-stopping criterion for max-log Turbo code decoding," in *Proc. 4th ISTC Related Topics*, Munich, Germany, Apr. 2006, pp. 1–6.

[11] J. Chen, A. Dholakia, E. Eleftheriou, M. P. C. Fossorier, and X.-Y. Hu, "Reduced-complexity decoding of LDPC codes," *IEEE Trans. Commun.*, vol. 53, no. 8, pp. 1288–1299, Aug. 2005.