Hybrid Iteration Control on LDPC Decoders

Erick Amador and Raymond Knopp
EURECOM
06904 Sophia Antipolis, France
name.surname@eurecom.fr

Vincent Rezard
Infineon Technologies France
06560 Sophia Antipolis, France
vincent.rezard@infineon.com

Renaud Pacalet
TELECOM ParisTech
06904 Sophia Antipolis, France
renaud.pacalet@telecom-paristech.fr

***Abstract*—Stopping criteria for the iterative decoding of low-density parity-check codes are considered. For a successful decoding task an inherent stopping criterion is used: the fulfillment of all parity-check constraints. For an unsuccessful task the decoder usually completes a preset maximum number of iterations. Proper iteration control is required to save energy and time on unnecessary decoder operation when processing undecodable blocks. In this paper we propose an iteration control policy that is driven by the combination of two decision metrics. One is the number of satisfied parity-check constraints and the second one is provided by a specific message computation kernel: the Self-Corrected Min-Sum decoding algorithm. Our results show that this hybrid control policy offers superior performance in terms of energy efficiency compared to previously proposed techniques. In addition we show empirically how stopping criteria should be tuned as a function of false alarm and missed detection rates.**

***Keywords*—LDPC codes; iterative decoding; stopping criterion; early termination**

I. Introduction

Low-density parity-check (LDPC) codes [1] have gained a lot of interest because of their outstanding error-correction performance. These codes are currently considered for a variety of applications, ranging from next generation wireless communications to magnetic storage. The decoding process for these codes is typically an iterative one. For a successful decoding task the fulfillment of all parity-check constraints is verified, but usually for an unsuccessful task a preset maximum number of iterations is executed. Early detection of an undecodable block not only saves energy on unnecessary decoder operation but may improve the overall latency when an automatic repeat-request (ARQ) strategy is also in use.

Previously proposed iteration control policies ([2][3][4]) differ on the decision metrics used. These decision metrics are characterized by their dependence or not upon extraneous variables that must be estimated. The parameters used within the decision rule must be tuned to particular scenarios. In the previous works it has been shown how this tuning essentially trades off error-correcting performance and the average number of iterations.

In this work, we identify a decision metric provided by a specific decoding algorithm, the recently proposed Self-Corrected Min-Sum algorithm [5]. This algorithm has been shown to provide quasi-optimal error-correction performance at very low complexity. We propose to combine two decision metrics in order to control the iterative decoding task. We perform comparisons among the previous art and the proposed

$$
\mathrm{H} = \begin{bmatrix} 1\,1\,0\,1\,0\,1 \\ 1\,0\,1\,1\,1\,0 \\ 0\,1\,1\,0\,1\,1 \end{bmatrix}
$$

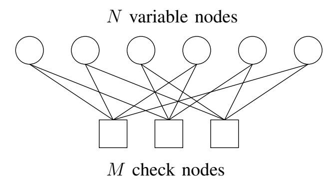

Fig. 1. LDPC code graph example

hybrid control policy in terms of error-correcting performance, average number of iterations and false alarm rate. The main advantage our work shows is the energy efficiency of the proposed policy as it exhibits empirically very low missed detection rates. Furthermore, we argue that the tuning of parameters of a stopping rule should be done based upon the false alarm and missed detection rates performance. The paper is organized as follows: Section II summarizes LDPC codes and their iterative decoding. In Section III, we show prior stopping criteria and the proposed control policy. Section IV shows simulation results and Section V concludes the paper.

II. LDPC Codes

LDPC codes are linear block codes defined by a sparse parity-check matrix $\boldsymbol{H}_{M\times N}$. This matrix defines M parity-check constraints among N code symbols. The code can be represented by a bipartite graph, where columns of $\boldsymbol{H}$ are mapped to *variable* nodes and rows are mapped to *check* nodes. The nonzero elements in $\boldsymbol{H}$ define the graph connectivity. Figure 1 shows an example matrix (non-sparse) and the code graph representation.

LDPC codes can be decoded by an iterative message-passing algorithm [6] where check nodes and variable nodes exchange extrinsic reliability values associated with each code symbol. The decoding task is stopped once all parity-check constraints are fulfilled:

$$
\boldsymbol{H} \cdot \boldsymbol{c}^T = \boldsymbol{S} = \boldsymbol{0} \tag{1}
$$

where $\boldsymbol{c}$ is a codeword and $\boldsymbol{S}$ is called the *syndrome*. If $\boldsymbol{S} \neq \boldsymbol{0}$ then the decoder completes a maximum number of iterations.

The computational complexity of the decoding task resides in the operation performed at the check nodes of the code graph, indeed it is in here where the tradeoff between error-correcting performance and complexity takes place. Optimal message computation is performed by the Sum-Product algorithm [6] at the expense of high complexity. The Min-Sum

(MS) algorithm [7] performs a sub-optimal message computation at reduced complexity. Several correction methods have been proposed to recover the performance loss of the MS algorithm by downscaling the messages computed using a normalization or an offset value, [7]. It has been argued recently in [5] that the sub-optimality of MS decoding is not due to the overestimation of the check node messages, but instead to the loss of the symmetric Gaussian distribution of these messages. This symmetry can be recovered by eliminating unreliable variable node messages or *cleaning* the inputs of the check node operation. By doing so, [5] introduces the Self-Corrected MS (SCMS) decoding, which exhibits quasi-optimal error-correcting performance. An input to the check node operation is identified as *unreliable* if it has changed its sign with respect to the previous iteration. Unreliable messages are *erased* and are no longer propagated along the code graph.

Because of its outstanding error-correcting performance and low complexity we looked closely at the behavior of the SCMS decoding in order to assist the early detection of undecodable blocks.

III. STOPPING CRITERIA

Iterative decoding algorithms are inherently dynamic since the number of iterations depends upon several factors. Proper iteration control policies should identify decodable and undecodable blocks in order to improve on energy expenditure and overall task latency. Convergence of a codeword is detected by verifying equation (1) (syndrome check) while non-convergence is usually detected by completing a preset maximum number of iterations.

A. Prior Art

Several works have proposed stopping criteria for the iterative decoding of LDPC codes. The authors in [8] proposed a termination criterion that detects so-called *soft-word cycles*, where the decoder is trapped in a continuous repetition without concluding in a codeword. This is achieved by storing and comparing the soft-words generated after each decoding iteration. This is carried out by means of content-addressable memories. This criterion saves on average iterations but clearly introduces storage elements.

In [2] a stopping criterion was proposed based upon the monitoring of the variable node reliability (VNR), defined as the sum of the magnitudes of the variable node messages. This decision rule stops the decoding process if the VNR does not change or decreases within two successive iterations. This comes from the observation that a monotonic increasing behavior is expected from the VNR of a block achieving convergence. The criterion is switched off once the VNR passes a threshold value that is channel dependent.

The criterion proposed in [9] is similar to the one in [2], it monitors the convergence of the mean magnitude of the variable node reliabilities. The decision rule uses two parameters tuned by simulations that are claimed to be channel independent.

The authors in [3] proposed a criterion that uses the number of satisfied parity-check constraints as the decision metric. Given the syndrome $\boldsymbol{S} = [s_1, s_2, \ldots, s_M]^T$, the number of satisfied constraints at iteration l is:

$$N_{spc}^{l} = M - \sum_{m=1}^{M} s_m \tag{2}$$

The decision rule monitors the behavior of this metric, tracking the increments and their magnitudes as well as the persistence of such behavior. In this rule three threshold values are used, all claimed to be channel independent.

A similar scheme was presented in [4] as it monitors the summation of the checksums of all parity-checks. This is indeed the complement of the decision metric used in [3]. The decision rule monitors this metric and uses two threshold values that are dependent upon signal-to-noise ratio (SNR) to make a decision.

The above control policies have been derived based upon the observation of the characteristic behavior shown by a particular decision metric within the decoding task. The decision metrics used by these control policies are characterized by their dependence or not upon extraneous variables. Estimating these variables (e.g., SNR) raises the implementation effort. In Section IV we show empirically how the tuning of the parameters used for a decision rule essentially trades off the false alarm rate and missed detection rate of undecodable blocks.

B. Proposed Control Policy

SCMS decoding introduces the concept of *erased messages*, messages which are deemed useless and are discarded after each decoding iteration. A formal treatment behind the concept of *erased messages* can be found in [5], but intuitively the number of messages erased per iteration provides some measure on the reliability (convergence) of the decoding task. For example, the less messages erased the more reliable the decoding task. Through simulations we observed the total number of erased messages per iteration to indentify the possibility to detect earlier an unsuccesful decoding task and also convergence. In the case of an undecodable block the number of erased messages fluctuates around a mean value (dependent upon the SNR), whereas for a decodable block this metric approaches zero relatively fast. In Figure 2 we show how the percentage of erased messages evolves with each decoding iteration for an instance of a decodable and an undecodable block. This corresponds to the decoding of the code defined in [10] with codeword length 1944 and coding rate 1/2 over the AWGN channel with QPSK modulation, with a maximum of 60 decoding iterations at $E_b/N_0 = 1dB$.

By detecting the characteristic monotonic decreasing behavior of the total number of erased messages when the decoder enters a convergence state, it is possible to save energy on potential undecodable blocks. The *erased messages* metric follows the cumulative quality of the arguments for the parity-check constraints, allowing in fact to observe the dynamics and evolution of the decoding process with fine granularity.

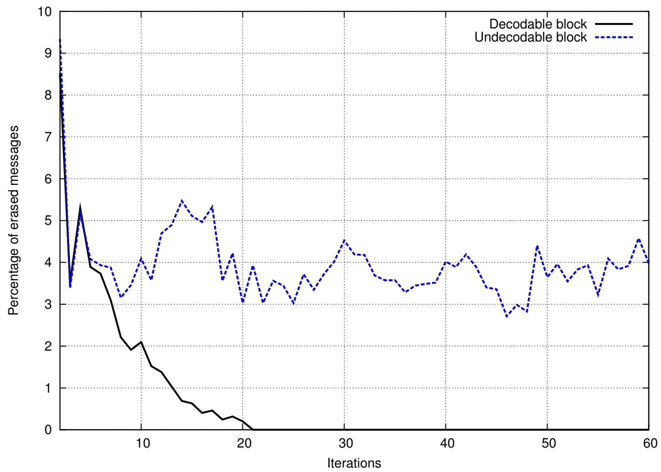

Fig. 2. Percentage of erased messages

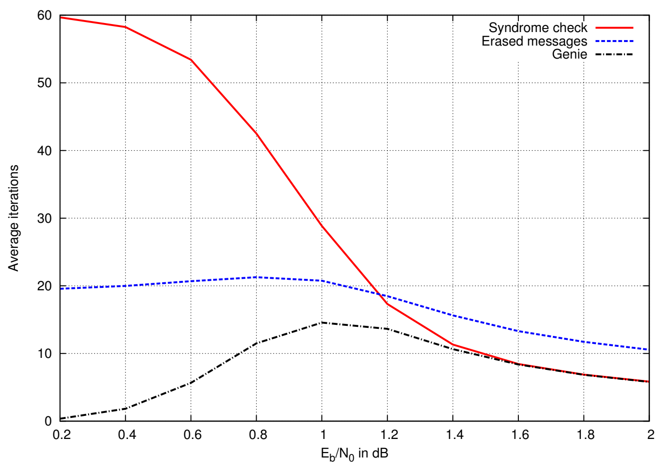

Fig. 3. Average iterations for stopping rules

In Figure 3 we show the average number of decoding iterations as a function of SNR for the same simulation scenario of Figure 2 for several stopping rules:

1) Syndrome check verification.
2) Erased messages metric. Decoding is halted when either the number of erased messages equals zero or a non-convergence condition is satisfied. For nonconvergence detection we allow only a fixed number of increments of this metric.
3) Genie. An ideal stopping rule with foreknowledge of the transmitted block, in this case decoding would not even start on an undecodable block.

The syndrome check and the genie criteria correspond to the empirical bounds of any valid stopping rule. From Figure 3 it is clear that the number of erased messages may be used as a decision metric to detect earlier undecodable blocks, but indeed is not suitable to detect early convergence as the absence of erased messages within an iteration is not a necessary condition for convergence.

From these observations we use the erased messages metric to detect an undecodable block and the syndrome check for decodable blocks. We propose a stopping rule that follows the evolution of the total number of erased messages by counting the increments of this metric and halting the decoding task once the number of increments exceeds a given threshold T. This threshold is a static parameter that essentially trades error-correcting performance and the average number of iterations. Algorithm 1 outlines the proposed decision rule.

Algorithm 1 Stopping Criterion - SCMS

ϵ_m: number of erased messages in row m
M: set of check nodes
f_s: boolean function for syndrome check, equation (1)
$count \leftarrow 0$; $S_\epsilon^l \leftarrow 0$
for all $iterations\ 1 < l \leq iterations_{max}$ **do**
  for all rows $m \in M$ **do**
    Decode row m
    $S_\epsilon^l \leftarrow S_\epsilon^l + \epsilon_m$
  end for
  if (f_s) **then**
    Halt decoding (convergence)
  end if
  if $(S_\epsilon^l > S_\epsilon^{l-1})$ **then**
    $count \leftarrow count + 1$
  end if
  if $(count > T)$ **then**
    Halt decoding (non-convergence)
  end if
end for

Due to space limitations we do not show the comparison between the rule in Algorithm 1 and the previous art, nevertheless we outline the rationale behind the idea of combining two decision metrics for a control policy. The objective of a stopping criterion can be formulated as the detection of an undecodable block. Thus the possible outcomes of such criterion may be a hit, a false alarm and a missed detection. A false alarm corresponds to the halting of the decoding task that would have been successful in the absence of such stopping rule. This indeed generates unnecessary retransmissions in ARQ protocols. On the other hand a missed detection represents useless energy expenditure and an unnecessary delay to request a retransmission. Even though any stopping criteria can be tuned to make arbitrarily small the average number of iterations this has an impact on the false alarm rate. In [3] the authors showed empirically how the average number of iterations and the false alarm rate are complementary. We investigated further by looking at the missed detection rate, as this indeed can provide hints into a criterion's efficiency. Our results showed that the stopping rule in Algorithm 1 exhibits a very low missed detection rate in relation to [3] and [2]. For this reason we propose to enhance the performance of the previous rules by adding the number of erased messages per iteration as another decision metric on a SCMS-based LDPC decoder. Figure 4 shows the proposed hybrid iteration control system.

We selected the number of parity-check constraints metric ([3]) as it offers less computational complexity than the VNR metric ([2]), Table I compares the cited stopping rules along

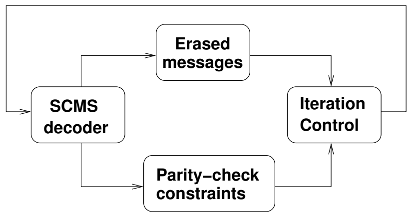

Fig. 4. Hybrid iteration control system

TABLE I
COMPLEXITY OF DECISION RULES

Criterion	Operations		Tuning Parameters	Data Type
	Compare	Add		
Shin [3]	3	M+3	3	Integer
Kienle [2]	1	N	1	Real
Algorithm 1	2	M+2	1	Integer

with Algorithm 1. The number of operations is given as a function of the dimensions of the parity-check matrix. N is usually much larger than M (e.g., twice for a rate 1/2 code), this means that on the number of calculations alone the criterion by Kienle is the most complex one. Furthermore, the type of data used by this criterion requires full resolution real quantities, this indeed imposes a more complex datapath (within a VLSI implementation) when compared to the proposed criterion and the one by Shin.

Therefore by observing the performance (error-correction, average iterations, false alarm and missed detection rates) of the mentioned stopping criteria we propose the hybrid iteration control policy for SCMS-based LDPC decoders such that two decision metrics are monitored in order to detect decodable and undecodable blocks. Even though it is possible to monitor all previously proposed decision metrics we found out that the erased messages metric provides the most effective detection for undecodable blocks (in the sense of exhibiting the lowest missed detection rate). In the next section, we provide results when utilizing the hybrid technique by using both Algorithm 1 and the criterion in [3] embodied as in Figure 4.

IV. RESULTS AND CRITERIA COMPARISON

All stopping criteria can reduce the average number of iterations depending upon the tuning of the decision parameters used within their control policy. This has consequences of different aspects that are worth investigating. In the following we tune the stopping criteria by [3] and [2] along with the proposed hybrid control to be used in the SCMS decoding within the simulation scenario described in the previous section.

Figure 5 shows the simulated error-correcting performance (bit error rate, BER) for the tested criteria. The stopping criteria can be tuned to be close in performance, for the case of the criterion in [3] (Shin) the parameters used were $\theta_d = 6, \theta_{max} = 4$ and $\theta_{spc} = 825$; for the criterion in [2] (Kienle) $MB = 16$ was used. The hybrid criterion uses $T = 22$ and the same setup just mentioned for [3].

Figure 6 shows the average number of iterations for the stopping criteria. The syndrome check and the genie are once again provided to observe the achievable empirical bounds.

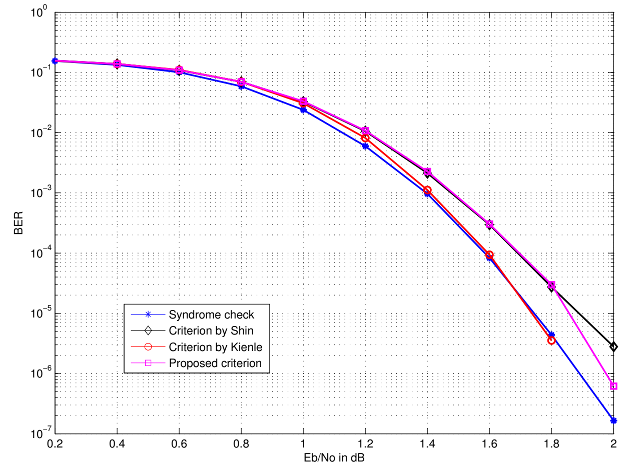

Fig. 5. Error-correcting performance for stopping criteria

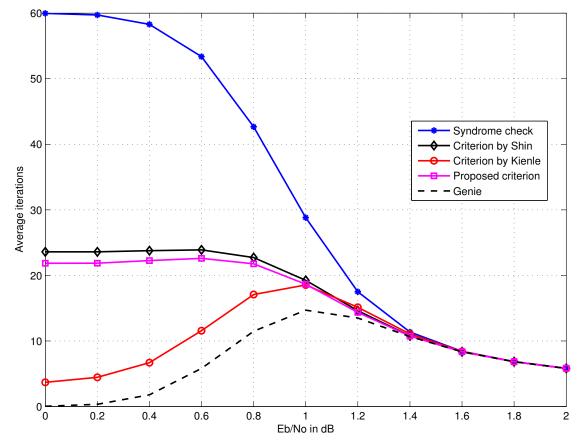

Fig. 6. Average iterations for stopping criteria

Here the tradeoff between average interations and performance loss is evident. From these figures the criterion by Kienle shows an advantage for less number of iterations for the low SNR region with the smallest performance loss, but this criterion shows the highest false alarm rate (FAR) on the same SNR region.

In Figure 7 we show the FAR of the simulated stopping criteria. This is a relevant figure of merit since the stopping mechanism on its own can be responsible for unnecessary retransmissions. We can observe how the criterion by Kienle shows less false alarms on the high SNR region, this is due to the inherent threshold that is used within this criterion to disable the stopping rule, but on the other hand this criterion shows the highest false alarm rate for the low SNR region. The comparison between the proposed criterion and the one by Shin is much closer and indeed can be tuned to have a similar performance.

So far we can observe that the criterion by Kienle in the low SNR region exhibits the lowest average number of iterations but leads to the highest number of retransmissions. In general the FAR of these criteria is relatively close, so we proceed to investigate their missed detection performance. Indeed the missed detection rate (MDR) can provide further insights into which criterion is actually saving energy without incurring into

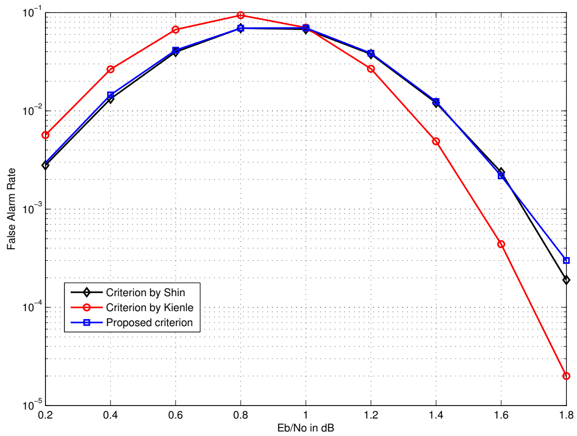

Fig. 7. False alarm rate of stopping criteria

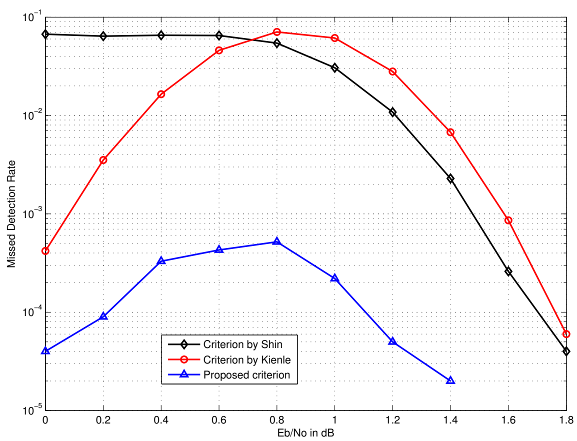

Fig. 8. Miss rate of stopping criteria

any penalty. Figure 8 shows the MDR for the investigated criteria. The criterion by Kienle performs better than Shin for the low SNR region, but this no longer holds as the SNR increases. The most relevant result is that the proposed hybrid criterion achieved a MDR at least one order of magnitude less than the best of the other ones.

The performance for each stopping criterion depends upon the tuning of the decision-making parameters. In Figure 9 we show the FAR and MDR for different choices of tuning parameters that result in different average number of iterations. These results are from the same simulated scenario for $E_b/N_0 = 1dB$. From this we can observe the tradeoff involving FAR and the average number of iterations for all criteria. In general the criteria can reduce the average number of iterations but this would result in a higher FAR, this tradeoff must be selected based upon the particular target application (required throughput and allowable retransmissions). Furthermore we can observe the relationship between MDR and average number of iterations. In this respect the proposed criterion exhibits the best performance. From this figure we can see how a proper tuning of the parameters for a decision rule must consider the relationship between FAR and MDR. FAR refers to the penalty risk introduced by the stopping rule, whereas MDR refers to how effective the stopping rule is for

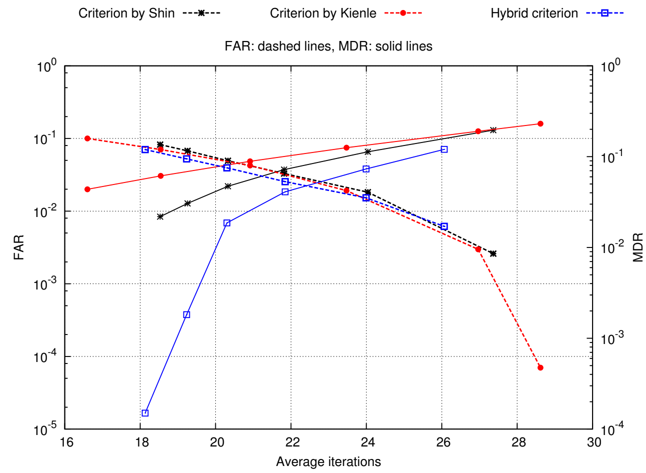

Fig. 9. FAR and MDR for stopping rules with different tuning of parameters

detecting undecodable blocks.

V. Conclusion

We have presented a stopping criterion for the iterative decoding of LDPC codes based upon the combination of two decision metrics that follow the convergence behavior of the decoding task. Motivated by the quasi-optimal error-correcting performance of the SCMS decoding kernel, we enhanced the performance of the previous art by adding the number of erased messages per iteration as a second decision metric for proper iteration control. We achieved a notorious decrease in the average number of missed detections for the iteration control policy, making it the best choice in terms of energy efficiency. Furthermore, we showed empirically how the proper tuning of stopping criteria should consider both FAR and MDR in order to accurately assess their performance.

References

[1] R. Gallager, "Low-Density Parity-Check Codes," *IRE Trans. Inf. Theory*, vol. 7, pp. 21–28, January 1962.

[2] F. Kienle and N. Wehn, " Low Complexity Stopping Criterion for LDPC Code Decoders ," in *Proc. of IEEE VTC 2005-Spring* , 2005, pp. 606–609.

[3] D. Shin, K. Heo, S. Oh, and J. Ha, " A Stopping Criterion for Low-Density Parity-Check Codes ," in *Proc. of IEEE VTC 2007-Spring* , 2007, pp. 1529–1533.

[4] Z. Cui, L. Chen, and Z. Wang, " An Efficient Early Stopping Scheme for LDPC Decoding ," in *13th NASA Symposium on VLSI Design* , 2007.

[5] V. Savin, " Self-Corrected Min-Sum Decoding of LDPC Codes ," in *Proc. of IEEE International Symposium on Information Theory* , 2008, pp. 146–150.

[6] F.R. Kschischang, B. Frey, and H.A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inf. Theory*, vol. 47, pp. 498–519, Febraury 2001.

[7] J. Chen and M. Fossorier, " Near Optimum Universal Belief Propagation based Decoding of LDPC codes ," in *IEEE Trans. on Comm.* , 2002, pp. 406–414.

[8] G. Glikiotis and V. Paliouras, "A Low-Power Termination Criterion for Iterative LDPC Code Decoders," *IEEE Workshop on Signal Processing Systems Design and Implementation*, pp. 122–127, November 2005.

[9] Jin Li, Xiao hu You, and Jing Li, " Early Stopping for LDPC Decoding: Convergence of Mean Magnitude (CMM) ," in *IEEE Communications Letters* , 2006, number 9.

[10] IEEE-802.11n, "Wireless LAN Medium Access Control and Physical Layer Specifications: Enhancements for Higher Throughput," *P802.11n-2009*, October 2009.