LUND UNIVERSITY

Modified Forced Convergence Decoding of LDPC Codes with Optimized Decoder Parameters

Sarajlic, Muris; Liu, Liang; Edfors, Ove

Published in:
2015 IEEE 26th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)

DOI:
10.1109/PIMRC.2015.7343339

2015

Link to publication

Citation for published version (APA):
Sarajlic, M., Liu, L., & Edfors, O. (2015). Modified Forced Convergence Decoding of LDPC Codes with Optimized Decoder Parameters. In *2015 IEEE 26th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)* IEEE - Institute of Electrical and Electronics Engineers Inc.. https://doi.org/10.1109/PIMRC.2015.7343339

Total number of authors:
3

General rights
Unless other specific re-use rights are stated the following general rights apply:
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.
• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal

Read more about Creative commons licenses: https://creativecommons.org/licenses/

Take down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

LUND UNIVERSITY

PO Box 117
221 00 Lund
+46 46-222 00 00

Modified Forced Convergence Decoding of LDPC Codes with Optimized Decoder Parameters

Muris Sarajlić, Liang Liu, and Ove Edfors
Department of Electrical and Information Technology, Lund University, Sweden
Email: (muris.sarajlic, liang.liu, ove.edfors)@eit.lth.se

***Abstract*—Reducing the complexity of decoding algorithms for LDPC codes is an important prerequisite for their practical implementation. In this work we propose a reduction of computational complexity targeting the highly reliable codeword bits and show that this approach can be seamlessly merged with the forced convergence scheme. We also show how the minimum achievable complexity of the resulting scheme for given performance constraints can be found by solving a constrained optimization problem, and successfully apply a gradient-descent based stochastic approximation (SA) method for solving this problem. The proposed methods are tested on LDPC codes from the IEEE 802.11n standard. Computational complexity reduction of 55% and a 75% reduction of memory access have been observed.**

I. Introduction

Low Density Parity Check (LDPC) codes, introduced in [1] have recently become a part of a number of communication standards, like WiMAX (IEEE 802.16e), IEEE 802.11n, 802.11ad and other [2]. The primary benefit of LDPC codes lies in their excellent error correcting properties that allow the systems using them to approach the information transmission capacity of the communication channel.

High complexity of the original decoding algorithm for LDPC codes, the iterative belief-propagation/sum-product (BP/SP) algorithm [1] has driven continuous research efforts targeting the reduction of its complexity while keeping the ensuing performance degradation at a tolerable level. Basic complexity reduction schemes [3] use mathematical approximations of the functions of the original algorithm. In a number of *early termination* schemes (e.g. [4]), the iterations stop as soon as some *stopping criterion* is met. Finally, a per-bit early termination scheme, referred to as *forced convergence* [5], employs a per-bit stopping criterion for the individual codeword bits.

In [6] it has been shown how the tunability of certain parameters of reduced-complexity decoding can be exploited to find the values of the parameters that minimize the decoding complexity while maintaining satisfactory performance, for given channel conditions. This paper is the continuation of work presented in [6] with:

1) A modification of the original forced convergence (FC) algorithm that yields a larger complexity reduction with the same performance degradation as the original FC;
2) The use of an iterative gradient descent-like algorithm that tries to find the optimum values of the decoder parameters for given channel conditions.

II. Background

A. General considerations

LDPC codes [1] are linear block codes with codeword length N described by a sparse parity check matrix $\boldsymbol{H}$ with dimensions $M \times N$. Their structure can be represented in the form of a bipartite graph [7] where codeword bits are represented by bit (variable) nodes and parity checks by parity (check) nodes, with the interconnections between variable and check nodes mapped from the parity check matrix.

The decoding process for LDPC codes can be viewed as an iterative exchange of messages between adjacent variable and check nodes. In this work, a low-complexity approximation of the BP/SP algorithm, the offset min-sum (OMS) algorithm [3] is used in the analysis.

The order in which messages are exchanged between variable nodes and check nodes (message passing scheduling) has a direct impact on both the performance of the decoding algorithm and on the complexity of its implementation. In the so-called *layered* scheduling scheme [4], [8], which is used in the decoding algorithm in this work, c-nodes and their adjacent v-nodes are grouped in *layers*, and the exchange of messages between v-nodes and c-nodes is done for each layer separately, in a sequential fashion.

B. Forced convergence: theoretic background

Each v-node has an associated aposteriori LLR value, commonly denoted as Q_v. The sign of Q_v maps to bit values 0 and 1. The magnitude of Q_v corresponds to the amount of "confidence" that the v-node has in its sign. As the iterations of the decoding algorithm progress it can be observed how (for SNRs after the "turbo cliff") the magnitudes of Q_v evolve towards $+\infty$ or $-\infty$. This indicates that the nodes become increasingly confident that they are a 0 or a 1 as iterations progress.

It can then be reasonable to stop the updating of Q_v for the very confident nodes, i.e. the nodes for which the magnitude of Q_v crosses some predefined threshold θ. Value of Q_v is therefore held at some fixed $Q_{v,frozen}$ for the remainder of the decoding process. This is referred to as *forced convergence*. Depending on the value of θ, forced convergence will result in certain performance degradation, but will also introduce a complexity reduction. By tuning the value of θ, the performance can therefore be finely traded for complexity.

III. Modified offset min-sum algorithm with layered scheduling and forced convergence

In this work, a modification of the original layered OMS algorithm with FC is introduced. To explain the nature of the modification, a concise overview of the message passing activities in the original algorithm will be given first.

One layer is assumed to be the group of v-nodes connected to one c-node. The set of v-nodes connected to c-node c is denoted by $N(c)$. Then, at iteration i, for each layer c and each v-node v belonging to this layer, the following three operations are performed:

1) Q_{temp} calculation: $Q^{(i)}_{temp,vc} = Q^{(i)}_{v} - R^{(i-1)}_{cv}$
2) R_{cv} calculation:

$$R^{(i)}_{cv} = \left[\prod_{v' \in N(c) \setminus v} \text{sign}(Q^{(i)}_{temp,v'c}) \right] \times \max \left\{ \min_{v' \in N(c) \setminus v} |Q^{(i)}_{temp,v'c}| - \omega, 0 \right\}$$

3) Q_v update: $Q^{(i)}_{v} = Q^{(i)}_{temp,vc} + R^{(i)}_{cv}$

The $Q_{temp,vc}$ can be seen as the message in which v informs c about its own sign and how confident it is that this is the actual value of its sign. On the other hand, R_{cv} is the total knowledge that other v-nodes in the layer have about the sign of v. It can be seen from the above expressions that the value of R_{cv} is influenced by Q_{temp} values in the layer that have the smallest magnitudes. V-nodes that are strongly convinced about their sign will send "strong" Q_{temp} messages of large magnitude that will not influence the value of R_{cv}. Therefore, the Q_{temp} messages for these "confident" nodes can be approximated by some constant value which is large enough.

Since the forced convergence approach also targets the "confident" v-nodes, it is natural to combine it together with the Q_{temp} approximation that was just described. The modified Q_{temp} calculation rule can then be formulated as follows:

$$Q^{(i)}_{temp,vc} = \begin{cases} Q_{v,frozen} & \text{if } v \text{ is frozen} \\ Q^{(i)}_{v} - R^{(i-1)}_{cv} & \text{otherwise} \end{cases} \tag{1}$$

The complete pseudocode formulation of the modified algorithm is given in Algorithm I. The completed notation is as follows: P_v are the apriori v-node LLRs (obtained from symbols received from the channel), ω is the offset value, a standard part of the offset min-sum algorithm. $X(\cdot)$ is a hard bit decision operator (converting Q_v to 0 or 1), $\boldsymbol{Q} = [Q_1 \quad Q_2 \ldots Q_N]$ is the codeword LLR vector and GF2$\{\cdot\}$ denotes operations in Galois field over $\{0, 1\}$.

Algorithm I: Layered OMS with FC and extrinsic message simplification

for all v-nodes v and c-nodes c **do** ▷ **Initialization**
  $R_{cv} \leftarrow 0$
  $Q_v \leftarrow P_v$
end for
$Inact = \emptyset$
for iterations i to I_{max} **do**
  for all c **do**
    for all $v \in N(c)$ **do** ▷ Q_{temp} **calculation**
      if $v \in Inact$ **then**
        $Q_{temp} \leftarrow Q_v$
      else
        $Q_{temp} \leftarrow Q_v - R_{cv}$
      end if
    end for
    $Q_{min1} \leftarrow \min_{v \in N(c)} \{|Q_{temp,vc}|\}$ ▷ R_{cv} **calculation**
    $Q_{min2} \leftarrow \min_{v \in N(c), v \neq v_{min1}} \{|Q_{temp,vc}|\}$
    $Q_{min1} \leftarrow max\{Q_{min1} - \omega, 0\}$
    $Q_{min2} \leftarrow max\{Q_{min2} - \omega, 0\}$
    $S = \prod_{v \in N(c)} sign(Q_{temp,vc})$
    for all $v \in N(c)$ AND $v \notin Inact$ **do**
      if $v = v_{min1}$ **then**
        $R_{cv} \leftarrow sign(Q_{vc}) \cdot S \cdot Q_{min2}$
      else
        $R_{cv} \leftarrow sign(Q_{vc}) \cdot S \cdot Q_{min1}$
      end if
      $Q_v \leftarrow Q_{temp,vc} + R_{cv}$ ▷ Q_v **update and thresholding**
      if $|Q_v| > \theta$ **then**
        $|Q_v| \leftarrow \theta$
        $v :\in Inact$
      end if
    end for
  end for
  if GF2$\{\boldsymbol{H} \cdot X(\boldsymbol{Q^T})\} = \mathbf{0}$ **then**
    stop iterations
  end if
end for

IV. Optimizing the LDPC decoder parameters

A. Problem formulation

It has been shown in [6] that, for a general vector $\boldsymbol{\rho}$ of environment settings (such as SNR or fading properties) and minimum performance requirement FER_c (expressed in terms of the frame error rate FER), the value of the threshold θ that minimizes the computational complexity $C(\theta, \boldsymbol{\rho})$ of the decoding algorithm is found by solving the general optimization problem

$$\begin{aligned} \underset{\theta}{\text{minimize}} \quad & C(\theta, \boldsymbol{\rho}) \\ \text{subject to} \quad & \text{FER}(\theta, \boldsymbol{\rho}) \leq \text{FER}_c \end{aligned} \tag{2}$$

In order to achieve optimum complexity reduction, the optimum value of θ should be found and applied for any value of the current environment settings. Put in simple terms, it should be adapted to the channel.

This work uses an analytical model of the computational complexity that is drawn from the algorithm structure. In the derivation of the model, tilde (˜) will be used to denote random terms. Layers will be indexed by l, decoding algorithm iterations by i and individual decoded blocks (different runs of the decoding algorithm) by b. The number of active v-nodes (nodes that have not yet been frozen) in layer l (and at iteration i and block b) is denoted by $\tilde{n}_a^{(b,i,l)}$. The total number of v-nodes adjacent to the c-node in layer l and the total number of c-nodes in the code (both deterministic and following from the code construction) are denoted by $n^{(l)}$ and $|c|$, respectively.

Number of iterations performed in the decoding of block b is denoted by $\tilde{I}^{(b)}$.

It should be pointed out that the number of active nodes $\tilde{n}_a^{(b,i,l)}$ and the number of iterations $\tilde{I}^{(b)}$ are *discrete random variables*; the randomness of $\tilde{n}_a^{(b,i,l)}$ is the consequence of applying FC, and $\tilde{I}^{(b)}$ is random due to the early termination (parity check at the end of each iteration). Probability mass functions $f_N(\tilde{n}_a^{(b,i,l)};\theta,\boldsymbol{\rho})$ and $f_I(\tilde{I}^{(b)};\theta,\boldsymbol{\rho})$ are parameterized by θ and $\boldsymbol{\rho}$. Owing to the inherent complexity of the LDPC code structure and the nonlinearity of the decoding algorithm, these pmfs are in general case extremely hard (if not impossible) to obtain in closed form.

As in [6], the complexity is given in the number of additions (assumed equivalent in complexity as comparisons) performed per decoded block. Complexity analysis of the decoding algorithm is based on the complexity analysis for a single layer l:

- Complexity of the Q_{temp} calculation section in one layer is
$$\tilde{C}_{Q_{temp}}^{(b,i,l)} = \tilde{n}_a^{(b,i,l)} \tag{3}$$
- Complexity of the R_{cv} calculation section depends on the number of different $|Q_{temp,vc}|$ values among which the two minimum elements are chosen. This number is denoted by $\tilde{n}_x^{(b,i,l)}$. The set of values of $|Q_{temp,vc}|$ from which the two minimum elements are picked is formed by $|Q_{temp,vc}|$ from active nodes, and a single θ value representing all the frozen nodes. Therefore
$$\tilde{n}_x^{(b,i,l)} = \min\{\tilde{n}_a^{(b,i,l)}+1, n^{(l)}\} \tag{4}$$
and the complexity of this section is
$$\tilde{C}_{R_{cv}}^{(b,i,l)} = \tilde{n}_x^{(b,i,l)} + \lceil \log_2 \tilde{n}_x^{(b,i,l)} \rceil + 2, \tag{5}$$
derived from the optimum complexity of finding two minimum elements in an unsorted array [9] and the four additions in lines 17 and 18.
- Finally, the complexity of the Q_v update and thresholding section is
$$\tilde{C}_{Q_v}^{(b,i,l)} = 2\tilde{n}_a^{(b,i,l)} \tag{6}$$

Total complexity of decoding one layer is
$$\tilde{C}^{(b,i,l)} = 3\tilde{n}_a^{(b,i,l)} + \tilde{n}_x^{(b,i,l)} + \lceil \log_2 \tilde{n}_x^{(b,i,l)} \rceil + 2, \tag{7}$$
and the complexity of decoding one block is then
$$\tilde{C}^{(b)} = \sum_{i=1}^{\tilde{I}^{(b)}} \sum_{l=1}^{|c|} \tilde{C}^{(b,i,l)} \tag{8}$$

Finally, a sample mean of $\tilde{C}^{(b)}$ over a window of W blocks is taken as an estimate of the complexity $C(\theta,\boldsymbol{\rho})$:
$$\hat{C}(\theta,\boldsymbol{\rho}) = \tilde{C} = \frac{1}{W}\sum_{b=1}^{W}\tilde{C}^{(b)} = \frac{1}{W}\sum_{b=1}^{W}\sum_{i=1}^{\tilde{I}^{(b)}}\sum_{l=1}^{|c|}\tilde{C}^{(b,i,l)} \tag{9}$$

If environment conditions $\boldsymbol{\rho}$ are assumed constant over W blocks and if additionally there is no dependence between noise or decoded data between different blocks, $\tilde{C}^{(b)}$ can be assumed to be an i.i.d. random variable coming from an unknown discrete pmf. Then, from the central limit theorem it follows that the distribution of $\hat{C}(\theta,\boldsymbol{\rho})$ is approximately
$$\mathcal{N}\left(C(\theta,\boldsymbol{\rho}), \sigma_C^2(\theta,\boldsymbol{\rho})\right), \tag{10}$$
with $C(\theta,\boldsymbol{\rho}) = \mathbb{E}[\hat{C}(\theta,\boldsymbol{\rho})]$. Note that $C(\theta,\boldsymbol{\rho})$ is not available in closed form; it is only possible to obtain its (noisy) estimate $\hat{C}(\theta,\boldsymbol{\rho})$.

In order to solve the optimization problem (2), $\text{FER}(\theta,\boldsymbol{\rho})$ needs to be obtained as well. Similar to $C(\theta,\boldsymbol{\rho})$, $\text{FER}(\theta,\boldsymbol{\rho})$ is not known in closed form and has to be estimated. This can be done in the usual way of counting block errors over a window of W blocks and then dividing by W. Formally,
$$\widehat{\text{FER}}(\theta,\boldsymbol{\rho}) = \frac{1}{W}\sum_{b=1}^{W}\mathbb{1}_{err}^{(b)}, \tag{11}$$
where $\mathbb{1}_{err}^{(b)}$ is an indicator function equal to 1 when block b is in error, and 0 otherwise. Values of the indicator function are Bernoulli distributed, and it is well known [10] that for a large enough W, $\widehat{\text{FER}}(\theta,\boldsymbol{\rho})$ is approximately distributed as
$$\mathcal{N}\left(\text{FER}(\theta,\boldsymbol{\rho}), \frac{\text{FER}(\theta,\boldsymbol{\rho})(1-\text{FER}(\theta,\boldsymbol{\rho}))}{W}\right) \tag{12}$$

We can therefore conclude that, instead of the cost and constraint functions from (2), in practice we can only obtain their noisy estimates:
$$\hat{C}(\theta,\boldsymbol{\rho}) = C(\theta,\boldsymbol{\rho}) + \eta, \tag{13}$$
$$\widehat{\text{FER}}(\theta,\boldsymbol{\rho}) = \text{FER}(\theta,\boldsymbol{\rho}) + \epsilon, \tag{14}$$
where, following from (10) and (12), η and ϵ are approximately zero-mean Gaussian with pdf parameterized by $\boldsymbol{\rho}$ and θ.

Optimization of θ is then performed using the noisy function estimates and is formulated as
$$\begin{aligned} \underset{\theta}{\text{minimize}} \quad & \hat{C}(\theta,\boldsymbol{\rho}) \\ \text{subject to} \quad & \widehat{\text{FER}}(\theta,\boldsymbol{\rho}) \le \phi_c, \end{aligned} \tag{15}$$
with ϕ_c being the new value of the constraint that accounts for the random nature of $\widehat{\text{FER}}$ and introduces a "safety margin".

B. Problem solution

A family of optimization methods, known collectively as *stochastic approximation methods* is known to be applicable to optimization problems in which the cost function is not known and can only be observed through its noisy estimates (measurements), like in (15). The first stochastic approximation (SA) method was proposed by Kiefer and Wolfowitz in [11] and has been followed by a host of similar methods (e.g. simultaneous perturbation SA by Spall, [12]).

SA methods are based on the classic gradient descent algorithm, in which a starting point is chosen and the optimum is approached iteratively by following the direction of the negative gradient. The difference between the deterministic

Algorithm II: Stochastic approximation with feasible set projection

1: Initialize θ_0
2: **for** k from 0 to $I_{max} - 1$ **do**
3: $\quad a_k = \frac{a}{(k+1)^\alpha}, \quad c_k = \frac{c}{(k+1)^\gamma}$
4: $\quad$ **if** $(\theta_k - c_k) < 0$ OR $\widehat{\text{FER}}(\theta_k - c_k, \boldsymbol{\rho}) > \phi_c$ **then**
5: $\quad\quad$ stop iterations
6: $\quad$ **end if**
7: $\quad \frac{\hat{\partial}}{\partial\theta} C(\theta_k, \boldsymbol{\rho}) = \frac{\hat{C}(\theta_k + c_k, \boldsymbol{\rho}) - \hat{C}(\theta_k - c_k, \boldsymbol{\rho})}{2c_k}$
8: $\quad \theta_{k+1} = \theta_k - a_k \frac{\hat{\partial}}{\partial\theta} C(\theta_k, \boldsymbol{\rho})$
9: **end for**
10: **if** $\widehat{\text{FER}}(\theta_k, \boldsymbol{\rho}) \leq \phi_c$ AND $\theta_k \geq 0$ **then**
11: $\quad \hat{\theta}^* = \theta_k$
12: **else**
13: $\quad \hat{\theta}^* = \theta_{k-1}$
14: **end if**

gradient descent and SA is that SA uses a noisy estimate of the gradient instead of its actual value.

For the constrained problem (15), the iterates have to be confined to the set of feasible points; this is modeled by a projection operator Π_Θ that projects the iterates back onto the feasible set Θ. The recursive expression for SA with feasible set projection, applied to the optimization problem (15) is given by

$$\theta_{k+1} = \Pi_\Theta \left\{ \theta_k - a_k \frac{\hat{\partial}}{\partial\theta} C(\theta_k, \boldsymbol{\rho}) \right\}, \tag{16}$$

with the "gradient estimate" $\frac{\hat{\partial}}{\partial\theta} C(\theta_k, \boldsymbol{\rho})$ calculated as

$$\frac{\hat{\partial}}{\partial\theta} C(\theta_k, \boldsymbol{\rho}) = \frac{\hat{C}(\theta_k + c_k, \boldsymbol{\rho}) - \hat{C}(\theta_k - c_k, \boldsymbol{\rho})}{2c_k} \tag{17}$$

The SA-based iterative algorithm with feasible set projection for estimating θ^* that solves (15) at a given environment setting $\boldsymbol{\rho}$ is given by Algorithm II.

Some practical information regarding the optimization algorithm:

- Feasible set Θ is defined as

$$\Theta = \left\{ \theta \geq 0 \mid \widehat{\text{FER}}(\theta) \leq \phi_c \right\} \tag{18}$$

  Negative values of θ produce undefined behaviour of the OMS-FC algorithm, hence the nonnegative constraint imposed on θ.
- Projection Π_θ is implemented in lines 4-6 and 10-14 of the algorithm.
- Initial point θ_0 and the finite difference step $\theta_0 - c_0$ are considered to be in Θ.
- Following the practical advice given in [13], the parameters α, γ, a and c of the sequences a_k and c_k are chosen as follows:
  - $\alpha = 0.602$, $\gamma = 0.101$
  - At the first iteration, a is set to $\frac{\Delta\theta_0}{\left|\frac{\hat{\partial}}{\partial\theta} C(\theta_0, \boldsymbol{\rho})\right|}$ where $\Delta\theta_0$ is the desired step in the first iteration.
  - Value of c is set to the estimated value of the standard deviation of $\hat{C}(\theta, \boldsymbol{\rho})$.

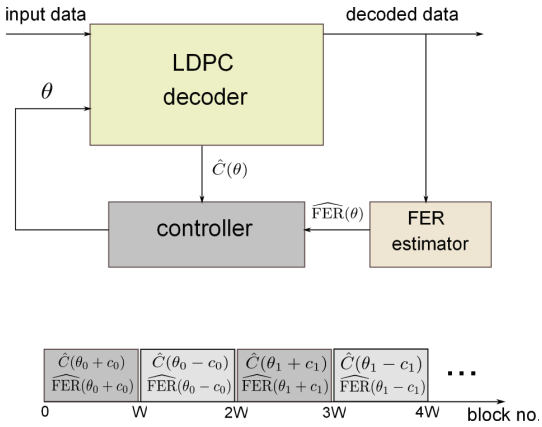

Fig. 1: System diagram and output timeline

- The new value of the constraint ϕ_c is determined from confidence intervals for $\widehat{\text{FER}}(\theta, \boldsymbol{\rho})$ from property (12). It is the value that, when chosen as the constraint, guarantees with a certain probability that the *actual* $\text{FER}(\theta, \boldsymbol{\rho})$ will be smaller or equal than the original constraint FER_c.

Fig. 1 shows the block diagram of the complete system for the optimization of LDPC decoder parameters, together with the timeline of the decoder outputs (naturally, averaged over the entire duration of one slot). From this diagram it is evident that this is a "black box" optimization method in which a controller unit, implementing Algorithm II, chooses the inputs to the system, estimates the gradient of the cost function from the observed system outputs and decides on the new input values based on the gradient estimate.

V. Simulation and Results

The described reduced complexity decoding algorithm and the optimization algorithm were tested in three different setups, based on three different LDPC codes from the IEEE 802.11n standard [2], with code rates and blocklengths given in Table I.

TABLE I: LDPC codes used in the simulations

Code 1	R = 1/2, N = 1944
Code 2	R = 1/2, N = 648
Code 3	R = 3/4, N = 648

The selected channel is AWGN and the modulation for all three setups is QPSK. The rates of the codes determine their operational SNR ranges. Codes 1 and 2 are suitable for use at low SNRs, whereas code 3 is better suited for use in the mid-SNR range. Different values of SNR are chosen as different states of the environment setting $\boldsymbol{\rho}$. Three decoders are compared in terms of complexity:

1) The plain OMS decoder, without FC
2) A "lazy" (that is, non-adaptive) OMS FC decoder with extrinsic message simplification that uses one value of θ over the entire tested range of SNR values. This value of θ is selected as minimum θ that satisfies the performance

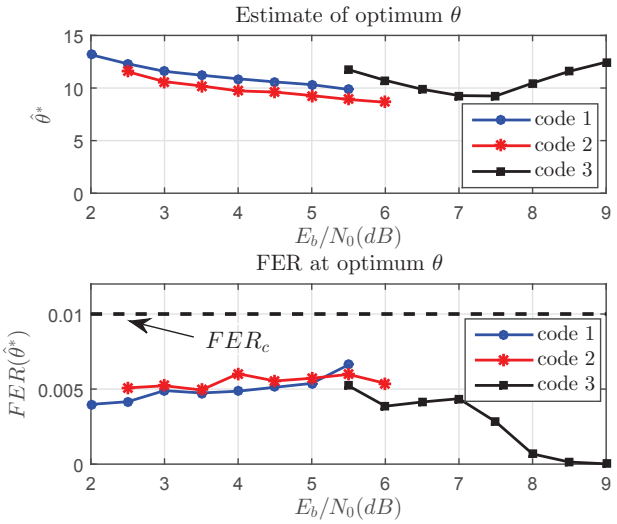

Fig. 2: Estimate of optimum θ and FER at $\hat{\theta}^*$

requirement for all SNRs, while yielding a complexity reduction compared to plain OMS.

3) An "optimized" OMS FC decoder with extrinsic message simplification using the estimated optimum value of the threshold, $\hat{\theta}^*$ (provided by the SA algorithm) at each SNR point.

Averaging window length W in the optimization is set to 1000. At each SNR point, optimization is run for 10 different random seeds, and the final value is the sample mean of the results obtained from these different runs. Performance constraint FER_c is set to 10^{-2}, and for $W = 1000$ and $Pr(\widehat{\text{FER}} \leq \text{FER}_c) = 0.95$ this translates to $\phi_c = 5.4 \cdot 10^{-3}$.

In Fig. 2 the values of $\hat{\theta}^*$ at each SNR are given together with $\text{FER}(\hat{\theta}^*)$ (averaged over 50 000 blocks and therefore considered the "true" value). The obtained values of FER confirm that the performance constraint $\text{FER} \leq 10^{-2}$ is satisfied at every $\hat{\theta}^*$. It was observed that, at high SNRs, $\hat{\theta}^*$ for code 3 do not follow the same trend of decrease with SNR as in the two other codes due to a limited number of optimization iterations (set to 100). With a larger number of iterations it is possible to attain smaller values of $\hat{\theta}^*$ at these SNRs.

The complexity, normalized by the maximum number of iterations and the number of information bits in the block, and the savings of lazy and adaptive schemes compared with the plain OMS scheme are shown in Fig. 3. Complexity $C(\theta)$ for all three decoders is averaged over 50 000 blocks and is therefore considered the "true" value. The results lend themselves to a comparison with the results in [6], since the same code (IEEE 802.11n, R=1/2, N=648) is analyzed in both works. In [6], optimum θ (found by a grid search) yielded maximum complexity savings of 35% compared to plain OMS for this particular code and the original OMS FC algorithm. In this work, the modified OMS FC algorithm achieves a 53% complexity reduction at the same SNR point, thereby confirming that the simple extrinsic message modification in (1)

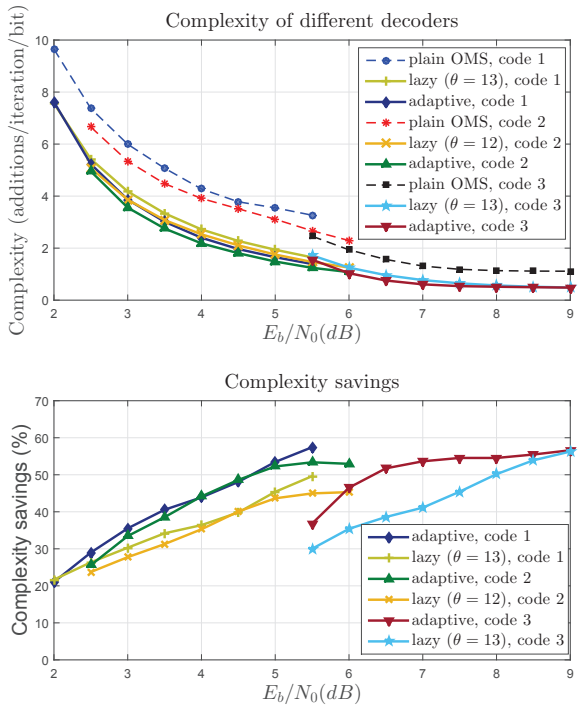

Fig. 3: Comparison of complexities for different decoders and codes

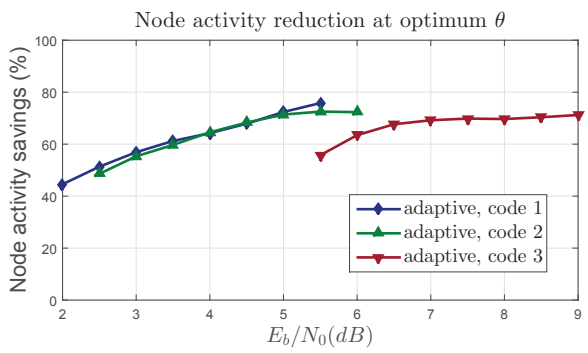

Fig. 4: Comparison of node activity savings for different codes with adaptive decoding

can result in significant savings of computational complexity. It can also be observed that the channel-adaptive decoder can bring in additional 5 - 12% of complexity reduction compared to the "lazy" decoder, emphasizing the general notion that adapting the system to its environment is beneficial for system performance. It should be noted that the controller in Fig. 1 is of negligible complexity compared to the decoder.

In actual hardware implementations of decoders, a large part of total energy consumption is due to memory access activities [14]. It is therefore beneficial to estimate the reduction in memory access when analyzing decoding schemes with reduced complexity. Although this heavily depends on the actual

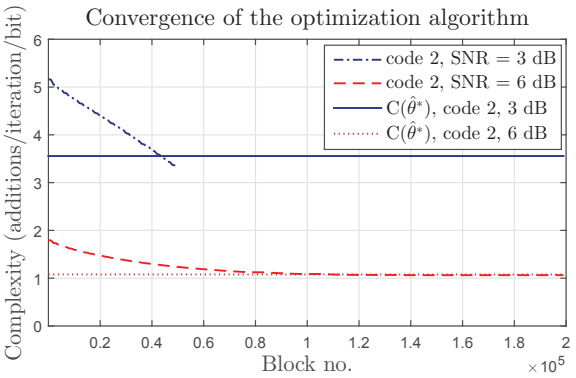

Fig. 5: Convergence behaviour of the SA optimization algorithm at different SNRs

implementation and memory design, some conclusions can be drawn from the structure of the algorithm. It can be identified that most memory access activity (reading/writing) will occur in "Q_{temp} calculation" and "Q_v update" sections of the decoding algorithm. Since both of these sections are performed when the v-node is active, it can then be safely assumed that the reduction in memory access will be proportional to the reduction of the number of active nodes. The node activity of the channel-adaptive OMS FC algorithm (at $\hat{\theta}^*$) is therefore compared to the plain OMS and the corresponding reduction is presented in Fig. 4. A very high (up to 75%) decrease of v-node activity suggests that the presented complexity reduction scheme can be expected to yield a highly energy efficient hardware implementation, both in terms of computations as well as memory access.

Finally, we shortly turn to practical implications of using the SA algorithm to find the optimum θ. Fig. 5 shows the rate of convergence of the optimization algorithm for two single runs (i.e. without averaging over different seeds) at different values of SNR. At the lower SNR value, the iterations stop after approximately 50 000 decoded blocks because the performance constraint is violated; at the higher SNR, they continue until the maximum number of iterations is exhausted, but the process can be seen to converge after around 100 000 decoded blocks. To put this into time perspective, we assume information bitrate of 100 Mbps. Given that one block of code 2 has 324 information bits, the convergence times for the two described cases are then $\approx 0.16s$ and $\approx 0.32s$, respectively. That indicates that SA can be used to tune the decoder to the optimum value of θ in real-time, provided that the channel is static or with very low mobility. The benefit of this approach lies in the fact that the optimization algorithm is of negligible complexity compared to the decoding algorithm and also in the fact that it does not need any channel information (conversely, $\hat{\theta}^*$ produced by the algorithm implicitly contains a channel estimate).

VI. Conclusion

This work proposes a modified rule for calculating the extrinsic messages in the LDPC decoding algorithm, in which the extrinsic messages corresponding to highly reliable bits can be simply approximated with the aposteriori LLRs, thereby reducing the computational complexity. It is additionally proposed that this modification is merged with the forced convergence scheme. It has been shown how the computational complexity of the resulting decoding algorithm can be modeled analytically, and how a gradient-descent based optimization scheme can be successfully applied to this model to find the maximum complexity reduction that the algorithm can achieve under some predefined performance constraints. The overall results show significant reduction of computational as well as memory access complexity, indicating high energy efficiency of a possible hardware implementation of the algorithm. Finally, it is shown that maximum complexity reduction is achieved if the parameters of the decoder are adapted to the environment.

Acknowledgment

The work presented is a part of the Digitally Assisted Radio Evolution (DARE) project, and the authors would like to thank the Swedish Foundation for Strategic Research (SSF) for providing the funds for the project.

References

[1] R. G. Gallager, *Low-Density Parity-Check Codes.* Cambridge, MA: MIT Press, 1963.

[2] IEEE 802.11n-2012, IEEE Standard for Information Technology-Telecommunications and information exchange between systems, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications [Online]. Available: http://standards.ieee.org/getieee802/download/802.11-2012.pdf

[3] J. Chen, A. Dholakia, E. Eleftheriou, M.P.C. Fossorier, and X.Y. Hu, "Reduced-Complexity Decoding of LDPC Codes," *IEEE Tran. Comm.*, vol.53, no.8, pp.1288-1299, Aug. 2005.

[4] D.E. Hocevar, "A reduced complexity decoder architecture via layered decoding of LDPC codes," *IEEE Workshop on Signal Process. Syst., 2004.*, pp.107-112, Oct. 2004.

[5] E. Zimmermann, P. Pattisapu, P. K. Bora, and G. Fettweis, "Reduced Complexity LDPC Decoding Using Forced Convergence," *Proc. 7th Int. Symp. on Wireless Personal Multimedia Commun.*, Padova, Italy, Sep. 2004, vol. 3, pp. 243246, WA2-2.

[6] M. Sarajlić, L. Liu, O. Edfors, "Reducing the Complexity of LDPC Decoding Algorithms: An Optimization-Oriented Approach", *Personal Indoor and Mobile Radio Communications (PIMRC), 2014 IEEE 25th International Symposium on*, Sep. 2014.

[7] R.M. Tanner, "A recursive approach to low complexity codes," *IEEE Trans. Inf. Theory*, vol.27, no.5, pp.533-547, Sep. 1981.

[8] E. Sharon, S. Litsyn, J. Goldberger, "An efficient message-passing schedule for LDPC decoding," *Proc. 23rd IEEE Conv. of Elec. and Electron. Engineers in Israel, 2004.*, pp.223-226, Sept. 2004.

[9] P. V. Ramanan, L. Hyafil, "New algorithms for selection", *J. of Algorithms*, vol. 5, no. 4, pp. 557-578, Dec. 1984.

[10] M. C. Jeruchim, P. Balaban, K. S. Shanmugan, *Simulation of Communication Systems*, 2nd ed. New York: Kluwer Academic Publishers, 2002.

[11] J. Kiefer and J. Wolfowitz, "Stochastic estimation of the maximum of a regression function", *Ann. Math. Stat.*, vol. 23, pp. 462-466, 1952.

[12] J. C. Spall, "Multivariate stochastic approximation using a simultaneous perturbation gradient approximation", *IEEE Trans. Autom. Control*, vol. 37, no. 3, pp. 332-341, 1992.

[13] J. C. Spall, "Implementation of the simultaneous perturbation algorithm for stochastic optimization," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 34, no. 3, pp. 817-823, Jul. 1998.

[14] C. Studer, N. Preyss, C. Roth, and A. Burg, "Configurable high-throughput decoder architecture for quasi-cyclic LDPC codes," in *Proc. 42nd Asilomar Conf. on Signals, Systems and Computers*, pp. 1137-1142, Oct. 2008.