Channel Estimation Considerations for Iterative Decoding in Wireless Communications

Pirouz Zarrinkhat
Airvana Inc.
Chelmsford, MA 01824, USA
pzarrinkhat@airvana.com

Masoud Ardakani
Dep’t of Elec. and Comp. Eng’g.,
University of Alberta
Edmonton, AB T6G 2V4, Canada
ardakani@ece.ualberta.ca

Raman Yazdani
Dep’t of Elec. and Comp. Eng’g.,
University of Alberta
Edmonton, AB T6G 2V4, Canada
yazdani@ece.ualberta.ca

Abstract—The time-varying nature of wireless channels poses a challenge for using soft-decision iterative decoders on such channels. Even if the channel gain is perfectly known to the receiver, inaccurate estimation of the additive noise power, results in incorrect log-likelihood computation at the receiver and hence significant performance degradation in the decoder. Through a detailed study of the effects of channel estimation errors on the performance of soft-decision iterative decoders in uncorrelated block-fading channels, we propose a solution which does not need a knowledge of the power of the additive noise. We show that this solution performs almost identical to the case for which a perfect knowledge of the power of the additive noise exists at the receiver. The choice of the block-fading channel reflects a situation where the equivalent variance of the additive Gaussian noise can change in a wide range and thus the decoder performance seriously relies on the knowledge of the noise power at the receiver.

I. INTRODUCTION

In the past few years, motivated by the exemplary performance of graphical codes, such as turbo-codes [1] and low-density parity-check (LDPC) codes [2], a great deal of research has been put into analysis and design of iterative decoders. By now, graphical codes are proposed for many channel models including wireless channels [3]–[5].

Due to the time-varying nature of wireless channels, soft decoding of graphical codes can become quite challenging. In most existing results, it is assumed that channel parameters, which are required for accurately computing the log-likelihood ratios (LLRs), are known to the receiver. With an incorrect estimate of these parameters, inaccurate LLRs will initialize the iterative decoder. This can significantly deteriorate the performance of soft iterative decoding algorithms.

Data-aided or blind channel estimation techniques can be employed at the receiver to obtain estimates of channel parameters. However, these solutions add to the complexity of the system, reduce the transmission rate, and are themselves subject to error and inaccuracy. Therefore, alternative solutions that can work without the channel knowledge or are robust to errors in channel estimation are of great importance.

In this work, we show that solutions that do not need any knowledge of the power of the additive white Gaussian noise (AWGN) can be developed. The focus of this work is on uncorrelated block fading (UBF) channels, i.e., when the fading coefficient of the channel remains unchanged over transmission of one codeword, but changes independently from one codeword to another. We assume that the fading gain is known to the receiver, but the noise variance is not. Hence, accurate LLRs cannot be computed at the receiver. This could happen, e.g., when the mobile moves slowly and other users’s interference can be approximated as an AWGN.

A UBF channel, during transmission of one codeword, can be modeled as a Gaussian channel. However, possible changes in the channel gain from one codeword transmission to another result in significant changes in the variance of the equivalent additive Gaussian noise. Thus the problem is significantly escalated in UBF channels. As a result, handling UBF channels without any knowledge of the noise variance is a clear indication of the robustness of the solution to changes in the noise variance. Obviously, a method which works on UBF channels can also work on Gaussian channels. As will be discussed later, some of our approaches seem to be applicable to general fading channels.

The focus of this work is on the decoder, i.e., we assume that the code is fixed and no adaptive code-selection method is used. Assuming that the noise variance is not known at the receiver, our goal is to find a choice of this parameter, for the purpose of computing the LLRs, that maximizes the chance of decoder’s success. We then compare the performance of this decoder with that of a decoder which has perfect knowledge of the noise variance.

Quite interestingly, we show that by assuming that the variance of the additive Gaussian noise is equal to $\sigma^2$, i.e., the decoding threshold of the employed code, the chance of decoding is maximized. This is a rather surprising result, because the optimum choice of this channel parameter is not a property of the channel. Moreover, we show that the performance of the system based on the above mentioned channel-estimation method is very close to the performance of a system that has perfect side information about the additive noise power. This result is an interesting one, especially because the performance of iterative decoders highly depends on the accuracy of LLR calculation (one can easily find cases where inaccurate LLR calculation causes decoding failure).

The rest of this paper is organized as follows. In Section II, we briefly review some required backgrounds. In Section III, first, we introduce the problem setup and study the effects of erroneously estimating the noise variance on the performance of soft-decision iterative decoders over UBF channels. From our observations, we then draw the main conclusion of the paper about the optimum choice of this parameter for computing the LLR values. Section IV concludes the paper.
II. BACKGROUND

Consider a binary-input AWGN (BIAWGN) channel whose output $y$ is given by

$$y = x + z,$$

where $x$, the input of the channel, is uniformly chosen from $\mathcal{I} \equiv \{-1, +1\}$, and $z$, the white noise of the channel, is a zero-mean Gaussian random-variable specified by its variance $\sigma^2$. For this channel, the LLR of $x$, defined by $\ln \frac{P(x=+1|y)}{P(x=-1|y)}$ and denoted by $\text{LLR}(x)$, is given by

$$\text{LLR}(x) = \frac{2}{\sigma^2} y. \quad (2)$$

Let us assume that $C^*(\lambda(\eta), \rho(\eta))$, the ensemble of $(\lambda(\eta), \rho(\eta))$-irregular LDPC codes of length $n$, is used to transmit information over our BIAWGN channel [6], [7]. On the receiver side, $C^*(\lambda(\eta), \rho(\eta))$ can be decoded using many different message-passing algorithms. In general, a message-passing algorithm is specified by a message alphabet, a set of check update rules, and a set of variable update rules. Sum-product and min-sum are two well-known examples of soft-decision message-passing algorithms [8], [9]: the two, in fact, work with messages from the set of reals. In LLR domain implementation of sum-product, the check update rules for the degree–$i$ check-nodes and the variable update rules for the degree–$j$ variable-nodes are given by

$$\Psi_{c,i,sp}^{(0)} = \text{null};$$

$$\Psi_{c,i,sp}^{(\ell \geq 1)}(m_1, m_2, \ldots, m_{i-1}) = 2 \tanh^{-1} \left( \prod_{s=1}^{i-1} \tanh \frac{m_s}{2} \right), \quad (3)$$

and

$$\Psi_{v,j,sp}^{(0)}(m_0) = m_0;$$

$$\Psi_{v,j,sp}^{(\ell \geq 1)}(m_0, m_1, m_2, \ldots, m_{j-1}) = m_0 + \sum_{j=1}^{j-1} m_j, \quad (4)$$

respectively, where $\ell$ is the iteration number, $m_s$, $1 \leq s \leq i$, and $m_j$, $1 \leq j \leq j$, are the extrinsic incoming messages from the neighboring variable-nodes and check-nodes, correspondingly, and $m_0$ is the so-called “channel message.” Min-sum is not as accurate as sum-product, but neither is it as complex. In LLR domain, min-sum’s variable update rules are the same as sum-product’s and its check update rules for the degree–$i$ check-nodes are given by

$$\Psi_{c,i,ms}^{(0)} = \text{null};$$

$$\Psi_{c,i,ms}^{(\ell \geq 1)}(m_1, m_2, \ldots, m_{i-1}) = \text{sgn} \left( \prod_{s=1}^{i-1} m_s \right) \cdot \min_{1 \leq i \leq i-1} \{|m_i|\}. \quad (5)$$

The standard analysis method for message-passing algorithms is density evolution, which tracks the evolution of the probability density function (PDF) of the messages throughout the iterations, under the assumption of a tree-like decoding neighborhood [10]. The formulation of density evolution in closed-form is usually far too complex to be of any practical use. Therefore, numerical approximations to density evolution are often used for sum-product and min-sum [11].

III. UNCERTAINTY ABOUT COMMUNICATION CHANNELS

It can be shown that amongst all message-passing algorithms, sum-product proffers the best error performance when the code’s factor graph is cycle-free [8], [9], [12]. For BIAWGN channels, however, sum-product does so only when the noise variance $\sigma^2$ is perfectly known: in fact, a considerable degradation in sum-product’s error performance is observed if not an accurate estimate of $\sigma^2$ is in hand. The reason is that an error in the estimation of $\sigma^2$ results in underestimation or overestimation of the LLR values (see (2)), and consequently, the algorithm will be initiated incorrectly, which may eventually lead to its failure.

Min-sum, on the other hand, is universal and, at the expense of its inferior error performance compared to sum-product, does not require any estimate of $\sigma^2$ [8], [9]. This is because in min-sum, $2/\sigma^2$ can be factored out from all the update rules, and as such, being unaware of the actual value of $\sigma^2$ does not affect the final decisions. This issue assumes great importance if one recalls that, in the real-world applications, an accurate estimate of the noise variance is simply not available at all times.

Assuming a fixed estimated noise variance, we now evaluate the performance of sum-product over UBF channels. The output $y$ of a UBF channel is given by

$$y = R \cdot x + z,$$  \quad (6)

where $x$ and $z$ are defined as in (1), and the random variable $R$, which remains fixed over a block and changes independently from block to block, is the known fading coefficient. Equivalently, we can rewrite (6) as

$$y' = R \frac{y}{R} = x + \frac{z}{R},$$  \quad (7)

which means that over a block interval, we are dealing with a BIAWGN channel whose noise variance is $\sigma^2_{\text{est}} = \sigma^2 / R^2$. Notice that the equivalent variance of the noise $\sigma^2_{\text{est}}$ can significantly vary from one block to another.

In this work, we are mostly interested in investigating the performance of sum-product over a wide range of actual and estimated values of the noise variance. In order to quantitatively measure the performance of sum-product, we do as follows. For a given code ensemble, a given actual noise variance $\sigma^2_{\text{act}}$, and a given estimated noise variance $\sigma^2_{\text{est}}$, we find the required number of iterations to achieve a desired target message error-rate $p_t$ and we use this number as a comparison measure.

Using density evolution, letting $D_{v \rightarrow c}(\cdot)$ represent the PDF of the variable-to-check messages in iteration $\ell$, and denoting the required number of iterations by $\ell^*(p_t)$, we have

$$\ell^*(p_t) \equiv \min \left\{ \ell \in \mathbb{Z}^+ : \int_{-\infty}^{0} D_{v \rightarrow c}^{(\ell)}(\zeta) \cdot d\zeta \leq p_t \right\},$$  \quad (8)
where $Z^*$ represents the set of non-negative integers.

Before introducing our results, we first provide some examples in which we compare $\ell^*(10^{-6})$ for different (regular and irregular) LDPC code ensembles with different rates, different ranges of $\sigma_{act}^2$ (or equivalently, $E_b/N_0,act$), and different values of $\sigma_{est}^2$ (or equivalently, $E_b/N_0,est$).

**Example 1:** Let us assume that the regular ensemble $C^\infty(\eta^2, \eta^5)$ is used to transmit information at rate $\frac{1}{2}$ over a UBF channel. The noise threshold, $E_b/N_0^*$ of this code is 1.1015 dB.

We are unaware of the instantaneous values of $E_b/N_0,act = -10 \cdot \log_{10} \sigma_{act}^2$ and use, instead, a fixed $E_b/N_0,est = -10 \cdot \log_{10} \sigma_{est}^2$ to calculate the LLRs that are required for initiating sum-product. Fig. 1 depicts $\ell^*(10^{-6})$ for several values of $E_b/N_0,est$ when $E_b/N_0,act$ (in dB) is in the interval $[1, 3]$. Note that widest convergence region is achieved when $E_b/N_0,est = 1.1015$ dB. Also, the values of $\ell^*(10^{-6})$ for $E_b/N_0,est = 1.1015$ dB are either minimum or very close to it for $E_b/N_0,act$ (in dB) $\in [1, 3]$. In other words, the best overall performance of sum-product is obtained when $E_b/N_0,est = E_b/N_0^*$.

**Example 2:** Consider the same setup as in Example 1 with the exception of the ensemble being $C^\infty(\eta^2, \eta^5)$, which is still regular but of rate $\frac{1}{4}$. The noise threshold of this code is at $E_b/N_0^* = 0.9568$ dB.

Again, we are unaware of the instantaneous values of $E_b/N_0,act = -10 \cdot \log_{10}(0.5 \cdot \sigma_{act}^2)$ and use a fixed $E_b/N_0,est = -10 \cdot \log_{10}(0.5 \cdot \sigma_{est}^2)$ to calculate the initial LLRs for sum-product. Fig. 2 depicts $\ell^*(10^{-6})$ for several values of $E_b/N_0,est$ when $E_b/N_0,act$ (in dB) is in the interval $[0.8, 3]$. Note that the best overall performance of sum-product is obtained when $E_b/N_0,est = E_b/N_0^*$.

**Example 3:** Consider the same setup as in the previous examples, but for the irregular ensemble $C^\infty(\lambda_\eta(\eta), \rho_\eta(\eta))$, where $\lambda_\eta(\eta) = 0.2882 \cdot \eta + 0.2466 \cdot \eta^2 + 0.0223 \cdot \eta^3 + 0.0955 \cdot \eta^4 + 0.3474 \cdot \eta^5$ and $\rho_\eta(\eta) = \eta^6$. The rate of this ensemble equals $\frac{1}{2}$ and its noise threshold, $E_b/N_0^*$, equals 0.9367 dB and 1.2521 dB for sum-product and min-sum, respectively.

Again, we use a fixed $E_b/N_0,est$ to calculate the initial LLRs for sum-product. Fig. 3 depicts $\ell^*(10^{-6})$ for several values of $E_b/N_0,est$ when $E_b/N_0,act$ (in dB) is in the interval $[0.2, 2]$. Note that the best overall performance of sum-product is obtained when $E_b/N_0,est = E_b/N_0^*$.
There are a number of interesting observations in the previous examples:

1) Using code’s threshold as the estimate of the noise variance (for computing the LLRs) results in the widest convergence range.

2) It also results in near-optimal performance over a very wide range of noise variance values.

3) This decoding solution, similar to min-sum, does not need any estimate of the noise variance. This is while it always outperforms min-sum.

4) Using a noise variance estimate less than the code’s threshold (for computing the LLRs) results in a very poor performance. (See the curves corresponding to $E_b/N_0,est = -1$ dB in Fig’s 1–3.)

These are by no means mere coincidences. In fact, we can show that when no knowledge of the noise variance is available, the best choice of $E_b/N_0,est$ for computing the LLR values is, in general, given by $E_b/N_0^\ast$. To do so, we first prove a theorem which states that for a fixed $E_b/N_0,est$ (or equivalently, for a fixed $\sigma_{act}^2$), $\ell^*(p_t)$ decreases with $E_b/N_0,act$ (or equivalently, $\ell^*(p_t)$ increases with $\sigma_{act}^2$). This is not as obvious as it sounds when $E_b/N_0,est \neq E_b/N_0,act$. For a fixed $E_b/N_0,est$, as $E_b/N_0,act$ increases, on the one hand the channel condition improves, but on the other, the gap between the actual and the estimated values of $E_b/N_0$ widens. It is not at all clear whether the decoder can tolerate significant errors in estimation of the noise variance, which result in incorrect computation of the LLR values. As an example, Fig. 1 shows that when $E_b/N_0,act = 2$ dB, an estimation $E_b/N_0,est$ of $-1$ dB results in the failure of the decoding. Similarly, there is no guarantee that an estimation $E_b/N_0,est$ of $1.015$ dB is safe when $E_b/N_0,act \gg 1.015$ dB.

Theorem: Under sum-product and Gaussian message assumption [13], $\ell^*(\sigma_{act}^2, p_t)$ increases, not necessarily strictly, with $\sigma_{act}^2$, for a given code ensemble and a given estimated noise variance $\sigma_{est}^2$.

In the following, we prove this theorem for the case where $d_v$ is large ($d_v \rightarrow \infty$). The proof for the general case has still eluded us, although extensive simulations support the validity of the theorem for any $d_v$.

Outline of the Proof: We show that for a regular ensemble $C^\infty(\eta^{d_k-1}, \eta^{d_k-1})$ and a fixed estimated noise variance $\sigma_{est}^2$, if $\sigma_{act1}^2 < \sigma_{act2}^2$, then $\ell^*(\sigma_{act1}^2, p_t) \leq \ell^*(\sigma_{act2}^2, p_t)$, provided that $d_v$ is large. The extension to irregular code ensembles is straightforward and is left to the reader.

We assume that the all-zero codeword (all-one channel-word) has been transmitted [10]. From (6), we can then conclude that the channel outputs are independent Gaussian random variables with mean 1 and variance $\sigma_{act}^2$. The decoding is started by computing the channel messages using (2). As our estimate of the noise variance is $\sigma_{est}^2$, it is not difficult to see that the channel messages will be independent Gaussian random variables with mean

$$\mu_o = \frac{2}{\sigma_{est}^2},$$

and variance

$$\sigma_o^2 = \frac{4 \cdot \sigma_{act}^2}{\sigma_{est}^2}.$$ 

The decoding then continues by iteratively exchanging messages between the check-nodes and the variable-nodes, using the update rules given by (3) and (4). Notice that the density of channel messages is not a symmetric Gaussian density as the variance $\sigma_o^2$ is not twice the mean $\mu_o$. Only when the power of the additive Gaussian noise is perfectly known, i.e., $\sigma_{est} = \sigma_{act}^2$, this density reduces to a symmetric Gaussian density.

Let $\mu_{c}$ and $\mu_{v}$ denote the means, and $\sigma_{c}^2$ and $\sigma_{v}^2$ the variances, of the outgoing messages from the check-nodes and the variable-nodes, respectively. Let further $p_{c_{1}}$ and $p_{c_{2}}$ denote the probability of check-nodes’s and variable-nodes’s outgoing messages being negative, and therefore erroneous, respectively. Under the Gaussian assumption, it can easily be seen that

$$p_{c_{1}} = \frac{1}{2} \cdot \text{erfc}\left(\frac{1}{\sqrt{2}} \cdot \frac{\mu_{c}}{\sigma_{c}}\right),$$

and

$$p_{c_{2}} = \frac{1}{2} \cdot \text{erfc}\left(\frac{1}{\sqrt{2}} \cdot \frac{\mu_{v}}{\sigma_{v}}\right),$$

where $\text{erfc}(\cdot)$ represents the complementary error function. We see that $p_e \equiv p_{c_{1}}$ depends on $\mu_{c}/\sigma_{c}$ and is a strictly decreasing function of $\mu_{v}/\sigma_{v}$. Now, consider the two cases $\sigma_{act1}^2 = \sigma_{act2}^2$ (case 1) and $\sigma_{act1}^2 < \sigma_{act2}^2$ (case 2).

At the end of iteration 0, we will have, from (4),

$$\mu_{c_{1}}(0) = \frac{2}{\sigma_{est}} = \mu_{v_{2}}(0);$$

$$\sigma_{c_{1}}^2(0) = \frac{4 \cdot \sigma_{act1}^2}{\sigma_{est}^2} < \frac{4 \cdot \sigma_{act2}^2}{\sigma_{est}^2} = \sigma_{v_{2}}^2(0),$$

Therefore, we get

$$\frac{\mu_{c_{1}}(0)}{\sigma_{c_{1}}(0)} > \frac{\mu_{v_{2}}(0)}{\sigma_{v_{2}}(0)},$$

and from (10),

$$p_{c_{1}}(0) < p_{c_{2}}(0).$$

This means that at the end of iteration 0, the situation is better for $\sigma_{act1}^2 = \sigma_{act1}^2$ (case 1).

In iteration 1, the incoming variable-to-check messages are independent Gaussian random variables with error probabilities $p_{e_{1}}$, and $p_{e_{1}} > p_{e_{2}}$, for case 1 and case 2, respectively. For the check-nodes, we can use Lemma 4.1 in [2] to conclude that

$$p_{e_{1}}^{(\ell \geq 1)} = \frac{1 - (1 - 2 \cdot p_{e_{1}}^{(\ell - 1)})^{d_v - 1}}{2},$$

which clearly shows that $p_{e_{1}}$ is strictly increasing in $p_{e_{1}}$. Therefore, we get $p_{e_{1}}(1) < p_{e_{2}}(1)$, and therefrom, assuming
Gaussian messages at the output of the check-nodes, we get
\[
\frac{\mu_{c_1}(1)}{\sigma_{c_1}(1)} > \frac{\mu_{c_2}(1)}{\sigma_{c_2}(1)}.
\] (15)

At the end of iteration 1, we will have, from (4),
\[
\begin{align*}
\mu_{v_1}(1) &= \mu_{v_1}(0) + (d_v - 1) \cdot \mu_{c_1}(1), \\
\mu_{v_2}(1) &= \mu_{v_2}(0) + (d_v - 1) \cdot \mu_{c_2}(1), \\
\sigma_{v_1}^2(1) &= \sigma_{v_1}^2(0) + (d_v - 1) \cdot \sigma_{c_1}^2(1), \\
\sigma_{v_2}^2(1) &= \sigma_{v_2}^2(0) + (d_v - 1) \cdot \sigma_{c_2}^2(1).
\end{align*}
\] (16)

Thus, we get, from (11) and (16),
\[
\frac{\mu_{v_1}(1)}{\sigma_{v_1}(1)} = \frac{2/\sigma_{v_1}^2 + (d_v - 1) \cdot \mu_{c_1}(1)}{\sqrt{4 \cdot \sigma_{v_1}^2 / \sigma_{c_1}^2 + (d_v - 1) \cdot \sigma_{c_1}^2(1)}}
\] (17)

and
\[
\frac{\mu_{v_2}(1)}{\sigma_{v_2}(1)} = \frac{2/\sigma_{v_2}^2 + (d_v - 1) \cdot \mu_{c_2}(1)}{\sqrt{4 \cdot \sigma_{v_2}^2 / \sigma_{c_2}^2 + (d_v - 1) \cdot \sigma_{c_2}^2(1)}}
\] (18)

Now, as \(d_v \to \infty\), we can get
\[
\frac{\mu_{v_1}(1)}{\sigma_{v_1}(1)} \approx \sqrt{d_v} \cdot \frac{\mu_{c_1}(1)}{\sigma_{c_1}(1)} \quad \text{and} \quad \frac{\mu_{v_2}(1)}{\sigma_{v_2}(1)} \approx \sqrt{d_v} \cdot \frac{\mu_{c_2}(1)}{\sigma_{c_2}(1)}
\] (19)

and from (10),
\[
p_{c_1}(1) < p_{c_2}(1).
\] (20)

This means that at the end of iteration 1, the situation is still better for the case \(\sigma_{\text{act}}^2 = \sigma_{\text{act}}^2\), (case 1).

With similar reasoning, it is not difficult to see that at the end of any iteration \(\ell > 1\), the situation is better for the case \(\sigma_{\text{act}}^2 = \sigma_{\text{act}}^2\), (case 1). As such, a desired target error-rate \(p_t\) is achieved faster for case 1. This is equivalent to saying that \(\ell^* = \sigma_{\text{act}}^2, p_t\) increases with \(\sigma_{\text{act}}^2\).

We now know that regardless of \(E_b/N_0, \ell^*(p_t)\) indeed decreases with \(E_b/N_0\). As such, it can be argued that by selecting \(E_b/N_0, \ell^*(p_t)\) successful decoding is guaranteed over the widest possible range of \(E_b/N_0\) values. To see this, notice that \(E_b/N_0^*\) is the minimum value of \(E_b/N_0\) for which the decoder’s success is possible, provided that perfect knowledge of the noise variance is available. Therefore, by selecting \(E_b/N_0^*\) as our estimate of \(E_b/N_0\), we guarantee successful decoding for \(E_b/N_0 \in [E_b/N_0^*, \infty]\). Also, using any other estimate, the convergence range will be smaller since the decoder will not succeed for \(E_b/N_0 \neq E_b/N_0^*\). This is because that convergence at the decoding threshold of a code is possible, only if exact LLRs are calculated and used for initializing the decoder. That is, \(E_b/N_0, E_b/N_0, E_b/N_0^*\).

It can also be seen that using \(E_b/N_0, E_b/N_0^*\), the decoder performs very close to the optimum performance obtained when the channel’s noise variance is perfectly known. This is because at \(E_b/N_0\) values close to \(E_b/N_0^*\), this choice performs very close to the optimum, and at very large values of \(E_b/N_0\), this choice, as well as other choices, has a performance very close to that of min-sum. The latter is because by choosing a large \(E_b/N_0\) for computing the LLRs, the true LLR values are all multiplied with a large number (recall that \(LLR = \frac{2}{\pi} \cdot y\)). As a result, most of all messages will be either +1 or −1, and therefore, the check update rules of sum-product simplifies to those of min-sum. Hence, all the curves in Fig’s 1–3 are converging to the curve for min-sum.

The implication of these results is that if the decoder has no knowledge of the channel’s noise variance, it could safely use \(E_b/N_0^*\), which is a property of the code and not of the channel conditions, as the channel estimate for computing the LLRs. Therefore, it is safe to remove any channel estimation process that is used for the purpose of computing LLRs. This results in significant complexity savings.

Now consider a situation in which the decoder knows that \(E_b/N_0\) varies in the rage \([a, b]\), but the actual value of \(E_b/N_0\) is unknown. If \(a < E_b/N_0^*\), according to above discussion, we have to select \(E_b/N_0^*\) as our estimate of \(E_b/N_0\), because it gives rise to the widest convergence region. If \(a > E_b/N_0^*\), we can easily argue that selecting \(a\) as our estimate of \(E_b/N_0\) will do better than selecting the threshold. This can be understood, noticing that at any point in the interval \([a, b]\), \(a\) is a better estimate of \(E_b/N_0\) than \(E_b/N_0^*\), because it is closer to the actual value. Similarly, it can be argued that this choice gives near-optimum results.

IV. CONCLUDING REMARKS

Assuming that the power of the additive white Gaussian noise in the channel is not known to the iterative LDPC decoder, we proposed a decoding strategy whose performance is almost identical to the case where this parameter is perfectly known. To show the efficiency of the method, we considered block fading channels, where, due to the significant changes in the fading gain, the equivalent noise power can change in a wide range.

The solution is based on selecting \(\sigma_{\text{act}}^2 = \sigma^2\) for computing the channel LLR values. Here, \(\sigma^2\) is the decoding threshold of the code and independent form the channel. This simple strategy is shown to perform almost as good as when the \(\sigma_{\text{act}}^2\) is perfectly known to the receiver. Thus, using this solution, any channel estimation module for the purpose of computing accurate LLRs can be removed from the receiver. This results in significant complexity saving in practical applications. This solution, similar to min-sum decoding, does not require a knowledge of \(\sigma_{\text{act}}^2\). However, it has a significantly wider convergence region compared to min-sum.

While our result is proved only for uncorrelated block fading channels and LDPC codes, it seems to be applicable to a much wider class of channels and codes. This is because, the main idea behind our method is that on an improved channel, even if \(\sigma_{\text{act}}^2\) is kept fixed, decoding is expected to improve. Thus, by making sure that around \(\sigma^2\), where decoding is most difficult, we have an accurate channel estimation and thus successful decoding, the decoding is guaranteed over the interval \(\sigma_{\text{act}}^2 \in [0, \sigma^2]\).
REFERENCES

[1] C. Berrou, A. Glavieux, and P. Thitimajshima, “Near Shannon limit error-
correcting coding and decoding: Turbo-codes (1),” in Conf. Rec. IEEE


code optimization of low density parity-check codes on Rayleigh fading

OFDM systems over correlated fading channels: Performance analysis
2002.

2005.

[6] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, and D. A. Spielman,
“Improved low-density parity-check codes using irregular graphs,” IEEE

capacity-approaching irregular low-density parity-check codes,” IEEE

Linköping University, Sweden, 1996.

Int’l Symp. Turbo Codes & Related Topics, (Brest, France), pp. 12–


product decoding of low-density parity-check codes using a Gaussian