Hindawi Publishing Corporation
EURASIP Journal on Wireless Communications and Networking
Volume 2008, Article ID 385421, 9 pages
doi:10.1155/2008/385421

Research Article

Construction and Iterative Decoding of LDPC Codes Over Rings for Phase-Noisy Channels

Sridhar Karuppasami and William G. Cowley

Institute for Telecommunications Research, University of South Australia, Mawson Lakes, SA 5095, Australia

Correspondence should be addressed to Sridhar Karuppasami, sridhar.karuppasami@postgrads.unisa.edu.au

Received 1 November 2007; Revised 7 March 2008; Accepted 27 March 2008

Recommended by Branka Vucetic

This paper presents the construction and iterative decoding of low-density parity-check (LDPC) codes for channels affected by phase noise. The LDPC code is based on integer rings and designed to converge under phase-noisy channels. We assume that phase variations are small over short blocks of adjacent symbols. A part of the constructed code is inherently built with this knowledge and hence able to withstand a phase rotation of $2\pi/M$ radians, where "M" is the number of phase symmetries in the signal set, that occur at different observation intervals. Another part of the code estimates the phase ambiguity present in every observation interval. The code makes use of simple blind or turbo phase estimators to provide phase estimates over every observation interval. We propose an iterative decoding schedule to apply the sum-product algorithm (SPA) on the factor graph of the code for its convergence. To illustrate the new method, we present the performance results of an LDPC code constructed over $\mathbb{Z}_4$ with quadrature phase shift keying (QPSK) modulated signals transmitted over a static channel, but affected by phase noise, which is modeled by the Wiener (random-walk) process. The results show that the code can withstand phase noise of 2° standard deviation per symbol with small loss.

1. INTRODUCTION

In the past decade, plenty of work was done in the construction and decoding of LDPC codes [1]. In general, the code construction techniques were motivated to provide a reduced encoding complexity and better bit-error rate (BER) performance. The channels considered are generally either additive white Gaussian (AWGN) or binary erasure channels. However, many real systems are affected by phase noise (e.g., DVB-S2). The severity of the phase noise depends on the quality of the local oscillators and the symbol rate. Hence the performance of codes on the channels with phase disturbances are of practical significance.

Over the past few years, iterative decoding for channels with phase disturbance has received lots of attention [2–7]. In [2, 3], the authors have proposed algorithms to apply over a factor graph model that involves the phase noise process. They used canonical distributions to deal with the continuous phase probability density functions. In particular, their approach based on Tikhonov distribution yields a good performance. In [4], the authors developed algorithms for noncoherent decoding of turbo-like codes for the phase-noisy channels. These schemes make use of pilot symbols for either estimation or decoding. In [5], the authors showed the rotational robustness of certain codes under a constant phase offset channel with the presence of cycle slips only during the initial part of the codeword.

In [6], the authors used smaller observation intervals to tackle varying frequency offset in the context of serially concatenated convolutional codes (SCCCs). They used blind and turbo phase estimators to provide a phase estimate for every sub-block. Since the phase estimates obtained from the blind phase estimator (BPE) are phase ambiguous, each sub-block is affected by an ambiguity of $2\pi/M$ radians. By differentially encoding the sub-blocks independently, the authors tackled the phase ambiguity. However, using an inner differential encoder along with an LDPC code provides a loss in performance and the degree distributions of the LDPC code needs to be optimized [7].

The concept of smaller observation intervals in the presence of phase disturbances is attractive and offers low complexity as well. Intuitively, as the observation interval get smaller more phase variation may be tackled. On the other hand, phase estimators produce poor estimates with smaller observation intervals. However, if the phase estimation error is smaller than π/M, the decoder may be able to converge correctly.

In our earlier work [8], we used sub-blocks in a binary LDPC-coded receiver to tackle residual frequency offset. The received symbol vector was split into many sub-blocks and BPE was used to provide a phase estimate across every sub-block. We introduced the concept of "local check nodes" (LCNs) to resolve the phase ambiguity created by the BPE on the sub-blocks. Local check nodes are odd degree check nodes connected to the variable nodes present within a single sub-block. In (1), the local check nodes correspond to the top four rows of the parity-check matrix, in which the bottom (dotted) part is connected according to random construction. In this small example, the LCN degree (d_c^L) is three and if the sub-block size (N_b) is six symbols, the parity check matrix provides $N_b/d_c^L = 2$ LCNs to resolve the phase ambiguity in each sub-block

$$H = \begin{bmatrix} 111000000000 \\ 000111000000 \\ 000000111000 \\ 000000000111 \\ \ddots\ddots\ddots\ddots\ddots \\ \\ \ddots\ddots\ddots\ddots\ddots \end{bmatrix}. \tag{1}$$

The phase-ambiguity-resolved vector is decoded by an LDPC decoder. Turbo phase/frequency estimates (e.g., [9]) are obtained during iterations to facilitate the convergence. The quality of the phase ambiguity estimate is better with more LCNs. Hence with reduced sub-block sizes, the phase ambiguity estimate is less reliable and the code suffers performance degradation.

Following [6, 8], but with a different perspective, we addressed the problem of phase noise for BPSK signals in the presence of a binary LDPC-coded system [10]. In particular, we incorporated the observation that, even under large phase disturbances the variation in phase over adjacent symbols are normally small. We created a set of check nodes called "global check nodes" (GCNs) that converge irrespective of phase rotations (0 or π radians) in any sub-blocks. We used BPE or TPE to provide a phase estimate in each sub-block. After the convergence of GCN, we used only one LCN per sub-block to resolve the phase ambiguity present in the sub-block. We found that even under relatively large phase noise and observation intervals, the method provided a good performance for BPSK signals. We did not make use of pilot symbols and the complexity is low. However, we found that the extension of the above approach to higher-order modulations was very difficult with a binary LDPC code. In particular, with a binary LDPC code, constructing global check nodes that converge irrespective of a phase rotation (a multiple of $2\pi/M$ radians) in the sub-blocks was difficult.

This paper addresses the problem of extending the above code construction technique to higher-order signal constellations based on integer rings. Specifically, we construct LDPC codes over rings with certain constraints on the placement of edges and edge gains such that they, along with sub-block phase estimation techniques, provide good performance under phase-noisy channels with low complexity. Under a noiseless channel, we present edge constraints based on integer rings generalized for any phase-symmetric modulation scheme, under which the convergence of the global check nodes is guaranteed in the presence of phase ambiguities in any sub-block. Similarly, we present generalized edge constraints for the local check node such that they are able to resolve the phase ambiguity in the sub-block. To illustrate the concepts discussed in this paper under a phase-noisy channel, we show the performance of an LDPC code constructed over $\mathbb{Z}_4$ with codewords mapped onto QPSK modulation, where the transmitted symbol $s_k \in \{s_k^m = e^{j((\pi/2)m+\pi/4)}\}$, $m = \{0, 1, 2, 3\}$.

The remainder of the paper is organized as follows. In Section 2, we discuss the channel model considered for our simulations. In Section 3, we address the effects of phase ambiguity on the check nodes and discuss the construction of global and local check nodes. In Section 4, we explain code construction and present a matrix inversion technique to obtain the generator matrix. In Section 5, we explain the receiver architecture and detail the iterative decoding for the convergence of these codes. We also show the additional computational complexity required due to the phase estimation process. In Section 6, we discuss the BER performance of the proposed receiver under phase noise conditions using the code constructed over $\mathbb{Z}_4$ for QPSK signal set. In Section 7, we discuss the benefits of the blind phase estimator in reducing the computational complexity involved with the turbo phase estimation and also show the BER performance of the low-complexity iterative receiver with the $\mathbb{Z}_4$ code under phase noise conditions. We conclude in Section 8 by summarizing the results of this paper.

2. CHANNEL MODEL

An information sequence is encoded by an (N, K) nonbinary LDPC code constructed over integer rings ($\mathbb{Z}_M$), where N and K represent the length and dimension of the code and $\mathbb{Z}_M$ denote the integers $\{0, 1, 2, \dots, M-1\}$ under addition modulo M, respectively. The alphabets over $\mathbb{Z}_M$ are mapped onto complex symbols $\mathbf{s}$ using phase shift keying (PSK) modulation with M phase symmetries. The complex symbols are transmitted over a channel affected by carrier phase disturbance and complex additive white Gaussian noise.

Ideal timing and frame synchronization are assumed and henceforth, all the simulations assume one sample per symbol. At the receiver, after matched filtering and ideal sampling, we have

$$r_k = s_k e^{j\theta_k} + n_k, \quad k = 0, 1, \dots, N_s - 1, \tag{2}$$

where s_k, r_k, θ_k, and n_k are the kth component of the vectors $\mathbf{r}, \mathbf{s}, \theta$, and $\mathbf{n}$, of length N_s, respectively. The noise samples n_k contain uncorrelated real and imaginary parts with zero mean and two-sided power spectral density (PSD) of $N_0/2$.

The phase noise process θ_k is generated using the Wiener (random-walk) model described by

$$\theta_k = \theta_{k-1} + \Delta_k, \quad k = 1, 2, \ldots, N_s - 1, \tag{3}$$

where Δ_k is a white real Gaussian process with a standard deviation of σ_Δ. θ_0 is generated uniformly from the distribution $(-\pi, \pi)$.

Let us divide the received symbol vector $\mathbf{r}$ of length N_s into B sub-blocks of length N_b. Assuming small phase variations over adjacent symbols, we may approximate the phase variations on the symbol in the lth sub-block by a mean phase offset $\widetilde{\theta}_l \in (-\pi, \pi)$. Similar to (2), the received sequence can be expressed as

$$r_{k'} \simeq s_{k'} e^{j\widetilde{\theta}_l} + n_{k'}, \quad l = 0, 1, \ldots, B - 1, \tag{4}$$

where $k' = N_b l + k$, $k = 0, 1, \ldots, N_b - 1$. While the channel model in (2) is used in our simulations, we use the approximate model in (4) for the code construction and receiver-side processing. The approximate phase offset over lth sub-block, $\widetilde{\theta}_l \in (-\pi, \pi)$ can be represented as the summation of an ambiguous phase offset $\phi_l \in (-\pi/M, \pi/M)$ and the phase ambiguity $\alpha_l \in \{0, 2\pi/M, 4\pi/M, \ldots, 2(M - 1)\pi/M\}$.

The proposed receiver tackles modest to high levels of phase noise. For instance, the phase noise considered in this paper (Wiener model with σ_Δ of 1° and 2°) is several times larger than the phase noise mentioned in the European Space Agency model (Wiener model with σ_Δ= 0.3° per symbol [2]). However, due to the assumptions made in (4), the proposed receiver will not be able to tackle large amounts of phase noise, such as the Wiener model with σ_Δ= 6° per symbol in [2, 3].

3. EFFECT OF PHASE AMBIGUITIES ON THE CHECK NODES

In this section, we address the effect of phase rotations that are multiples of $2\pi/M$ radians on the global and local check nodes of an LDPC code constructed over $\mathbb{Z}_M$. Let $H_{i,j}$ be the elements of the parity check matrix participating in the ith check node such that,

$$\sum_{j=1}^{d_c} H_{i,j} x_j = 0 \pmod{M}, \tag{5}$$

where d_c is the degree of the check node, x_j is the jth symbol participating in the ith check node and the value of $H_{i,j}$ is chosen from the nonzero elements of $\mathbb{Z}_M$. In the remaining subsections, we denote the degree of the GCN and LCN as d_c^G and d_c^L, respectively.

3.1. Global check nodes

Unlike local check nodes, the edges of the GCN are spread across many sub-blocks. Let p be the number of global check node edges connected to symbols present within one sub-block. Say, all symbols in that sub-block are rotated by $2\pi t/M$ radians, where $t \in \{0, 1, \ldots, M - 1\}$. As a result, the check equation in (5) becomes

$$\sum_{j=1}^{p} H_{i,j}(x_j + t) + \sum_{j=p+1}^{d_c^G} H_{i,j} x_j = \sum_{j=1}^{p} H_{i,j} t + \sum_{j=1}^{d_c^G} H_{i,j} x_j = t \sum_{j=1}^{p} H_{i,j.} \tag{6}$$

Thus for arbitrary integer t, (6) becomes zero only if

$$\sum_{j=1}^{p} H_{i,j} = 0 \pmod{M}. \tag{7}$$

In the case of binary LDPC code, p should be even in order to satisfy (7). For LDPC codes over higher-order rings, p can either be odd or even depending on the values of $H_{i,j}$. In this work, we select the values of $H_{i,j}$ from the set of nonzero divisors of $\mathbb{Z}_M$ ($\{1, 3\}$ from $\mathbb{Z}_4$) to avoid problems during matrix inversion. As a result, p becomes even in the case of LDPC code over integer rings which further makes d_c^G as well, even.

Example 1. Assume an LDPC code constructed over $\mathbb{Z}_4$ with $B = 4$ sub-blocks. Consider a degree-8 GCN whose edges are connected to two symbols per sub-block ($p = 2$) and the corresponding edge gains be $g = [1, 3, 1, 3, 3, 1, 1, 3]$. One set of symbols that satisfies this check is $x = [3, 2, 3, 1, 1, 3, 0, 1]$. Let us assume that sub-block one and four are rotated by $\pi/2$ and π radians, respectively. Therefore, the sub-block rotated version of x, say $x^r = [0, 3, 3, 1, 1, 3, 2, 3]$. It can be seen that x^r still satisfies the parity check equation with the same g. Note that each sub-block has one edge with value "1" and another with "3," whose sum is 0 (mod 4) as required by (7).

3.2. Local check nodes

Local check nodes resolve the phase ambiguity present in a sub-block. Let the elements $H_{i,j}$ participating in check i be selected from a single sub-block such that,

$$\sum_{j=1}^{d_c^L} H_{i,j} \neq 0 \pmod{2}. \tag{8}$$

Alternatively, (8) represents that the element $\sum_{j=1}^{d_c^L} H_{i,j}$ is chosen from the set of nonzero divisors from $\mathbb{Z}_M$, which is achieved by performing the summation over modulo 2 rather than M. If modulo M is used, the check node will not resolve certain phase ambiguities as explained below.

If all the symbols x_j participating in ith local check node are rotated by $2\pi t/M$ radians, then using (5) and (8), we can show that for every t there exists a distinct residue (mod M) which provides a solution for the phase ambiguity present on the participating symbols x_j. Considering all the operations below are modulo M,

$$\sum_{j=1}^{d_c^L} H_{i,j}(x_j + t) = \sum_{j=1}^{d_c^L} H_{i,j} x_j + \sum_{j=1}^{d_c^L} H_{i,j} t = t \sum_{j=1}^{d_c^L} H_{i,j}. \tag{9}$$

Hence t can be written as

$$t = \left(\sum_{j=1}^{d_c^L} H_{i,j}(x_j + t) \right) \times \left(\sum_{j=1}^{d_c^L} H_{i,j} \right)^{-1} \pmod{M}. \tag{10}$$

In case where the $\sum_{j=1}^{d_c^L} H_{i,j}$ do not have a multiplicative inverse in $\mathbb{Z}_M$ (say $\sum_{j=1}^{d_c^L} H_{i,j}$ equals a zero divisor), then (9) is satisfied for any $t \in$ {zero divisors in $\mathbb{Z}_M$} and hence the phase ambiguity estimate is not unique. Thus choosing $\sum_{j=1}^{d_c^L} H_{i,j}$ with a multiplicative inverse in $\mathbb{Z}_M$ ensures phase ambiguity resolution. Further, by selecting the edge gains of the LCN from the nonzero divisors of $\mathbb{Z}_M$, which are odd integers less than M, we require an odd number of edges to satisfy (8). Hence the degree of the local check node d_c^L is always considered to be odd in this work.

Example 2. Let us consider the code and rotations as in Example 1. Let the code include a degree-3 LCN whose edges with gains $[1, 3, 1]$ are connected to the first sub-block. A set of symbols that satisfies this check is $x = [3, 0, 1]$. Due to the rotation of $\pi/2$ radians in the first sub-block, $x^r = [0, 1, 2]$. Using (10), we can evaluate that $t = 1$ which corresponds to $\pi/2$ radians.

4. NONBINARY CODE CONSTRUCTION

We apply the above set of principles in constructing codes that are beneficial in dealing with phase noise channels. Similar to [11], we construct a binary code and choose the nonzero divisors from $\mathbb{Z}_M$ as edge gains such that check conditions as described in Section 3 are satisfied.

4.1. Code construction

Following Section 2, let us say we have "B" sub-blocks of length N_b. A binary parity check matrix $H_{N-K \times N}$ is constructed such that it involves two parts:

$$H = \begin{bmatrix} H_{\text{resolving}} \\ \cdots\cdots\cdots\cdots \\ H_{\text{converging}} \end{bmatrix}. \tag{11}$$

The upper $(B \times N)$ part of the matrix, called $H_{\text{resolving}}$, involves B local check nodes in contrast to N_b/d_c LCNs as in our previous method [8], which are used to resolve the phase ambiguity in B sub-blocks. The lower $(N - K - B \times N)$ part of the matrix, called $H_{\text{converging}}$, contains $N - K - B$ check nodes whose neighbours are selected such that their convergence is independent of the phase ambiguities in the sub-block. We assume the degree of all the local (global) check nodes to be equal to d_c^L (d_c^G). The codes are designed to be check biregular, (i.e., with two different degrees, d_c^L and d_c^G). However, there is no constraint on the variable node degree.

We construct the code as per the following procedure.

(1) *Construction of local check nodes:* the edges of the local check node are connected to the first d_c^L symbols of the sub-block for which it resolves the phase ambiguity. For example, assuming $d_c^L = 3$, let $H_{ij} = 1$ where j corresponds to the first 3 columns of each sub-block. However, we can arbitrarily choose the set of d_c^L symbols from any part of the sub-block.

(2) *Construction of global check nodes:* for every symbol, the parity checks in which the symbol participates are randomly chosen based on its degree and (7). As in Example 1, every global check node participates in only two symbols from a sub-block. Care was taken to avoid short cycles after constructing every column.

To illustrate the local and global check nodes, a small parity check matrix (H) is shown in (12). The first *four* rows corresponding to the local check nodes ($H_{\text{resolving}}$) are shown at the top. The two rows below the local check nodes are connected globally and also have $p = 2$ edges connected to symbols from a sub-block. The restriction of two edges per sub-block provides a better connectivity in the code. The same technique is continued to construct the remaining global check nodes in the dotted part of the matrix. The local and the global check nodes shown in the first and fifth rows of the H-matrix are used in the previous examples. A portion of the Tanner graph of the H matrix, in (12), is shown in Figure 1. Local check nodes (shaded checks) and their edges (solid lines) are distinguished from the global check nodes and their edges (dash-dotted lines)

$$H = \begin{bmatrix} 131000000000000000000000 \\ 000000111000000000000000 \\ 000000000000333000000000 \\ 000000000000000000313000 \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ 100300010300030100001030 \\ 010300031000003100130000 \\ \vdots\ddots\ddots\ddots\ddots\ddots\ddots\ddots\ddots\ddots\ddots\vdots \\ \vdots\ddots\ddots\ddots\ddots\ddots\ddots\ddots\ddots\ddots\ddots\vdots \end{bmatrix}. \tag{12}$$

4.2. Some comments on encoding

We used the Gaussian elimination (GE) approach to obtain a systematic generator matrix. Even though the edge gains of the parity check matrix are nonzero divisors, we encountered zero divisors ({2} in the case of $\mathbb{Z}_4$) during GE in the diagonal part of the matrix. To avoid this problem, we interchanged columns across the parity check matrix such that we obtain a generator matrix (G) corresponding to the column-interchanged parity check matrix (H'). Since we wanted to use the original H matrix instead of H', we created a permutation table (P) to record the columns

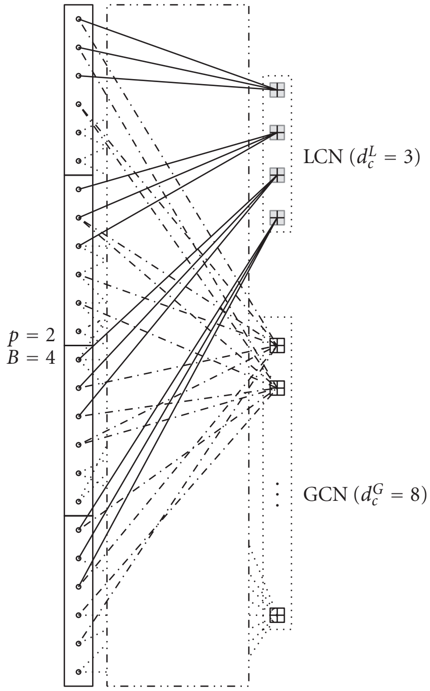

Figure 1: Tanner graph of the H matrix in (12), illustrating local and global check nodes.

that are interchanged during inversion. Alternate inversion techniques may avoid the use of permutation table P.

A summary of the communication system used in the simulations is given in Figure 2. The message is encoded by the generator matrix G to produce the codeword (c). The codeword c undergoes inverse permutation to produce c'. The codeword is transmitted through the composite channel. Since the permuted-encoded symbols are the codewords of the original code H, the decoder decodes the codeword. The decoded codeword $\hat{c}'$ is again permuted to give the original codeword $\hat{c}$.

5. RECEIVER ARCHITECTURE AND ITERATIVE DECODING SCHEDULE

The receiver architecture to tackle large phase disturbances is shown below in Figure 3. We used the SPA algorithm for LDPC codes over rings, similar to [12]. In the case of an AWGN channel, the SPA may be applied over the entire code for convergence. However, in the presence of phase disturbances, phase estimators provide an ambiguous phase estimate and hence the SPA is applied only over the rotationally invariant part of the factor graph, that is, the graph involving global check nodes only.

This section discusses the application of SPA on the factor graph of the code with phase offset on every sub-block such that the benefits of local and global check nodes are achieved. Thus we split up the decoding into three phases as described below.

(1) Converging phase.

(a) The likelihood vector, of length M, for the kth variable node is initialized with the channel likelihoods, $p(r_k \mid s_k = s_k^m) = (1/2\pi\sigma^2)\exp\{-(|r_k - s_k^m|^2)/2\sigma^2\}$, where $m = \{0, 1, \ldots, M-1\}$, $k = \{0, 1, \ldots, N_s - 1\}$ and σ^2 is the noise variance.

(b) The SPA is applied over the $H_{\text{converging}}$ part of the code alone. Local check nodes are not used. The messages coming from these nodes are assigned to be equiprobable.

(c) After every d iterations, the turbo phase estimator (TPE) [9] estimates the phase offset $\hat{\phi}_l$, which is given by

$$\hat{\phi}_l = \arg\Bigg(\sum_{k'} r_{k'} a_{k'}^*\Bigg), \tag{13}$$

where k', as defined in (4), is the kth component in the lth sub-block and $a_{k'}^*$ is the complex conjugate of the soft symbol estimate. The soft symbol estimate $a_{k'}$ of the symbol $s_{k'}$ is given by

$$a_{k'} = \sum_{m=0}^{M-1} s_{k'}^m p\big(s_{k'} = s_{k'}^m \mid \tilde{r}_{k'}\big), \tag{14}$$

where $p(s_{k'} = s_{k'}^m \mid \tilde{r}_{k'})$ is the a posteriori probability that symbol $s_{k'} = s_{k'}^m$. The received symbol vector corresponding to lth sub-block is corrected using the turbo phase estimate $\hat{\phi}_l$.

(d) The likelihoods are recalculated from $\tilde{\mathbf{r}}$ after phase correction and are used to update the messages that are passed on to the global check node.

(e) Steps (a)–(c) are repeated until all the global check nodes are satisfied.

(2) Resolving phase.

(a) As the symbol a posteriori probabilities at the variable nodes are good enough at the end of converging phase, a hard decision is taken on the symbols, which corresponds to $(x_j + t)$ in (10). These hard decisions are used to evaluate the sub-block phase ambiguity estimates $\hat{\alpha}_l = 2\pi t/M$ using local check nodes as in (10), which are further used to correct the received symbol values, giving $\tilde{\mathbf{r}}'$. In general, the decoder converges at the end of this stage.

(3) Final phase.

(a) If required, SPA is continued over the entire code involving both $H_{\text{resolving}}$ and $H_{\text{converging}}$ until either the syndrome $(H\hat{c}'^T = 0)$ is satisfied or a specified number of iterations are reached. Turbo phase estimation or phase ambiguity resolution is is not required at this phase.

5.1. Comments on turbo phase estimation

In general, turbo phase estimation can provide a phase estimate in the range $(-\pi, \pi)$. However, during the converging

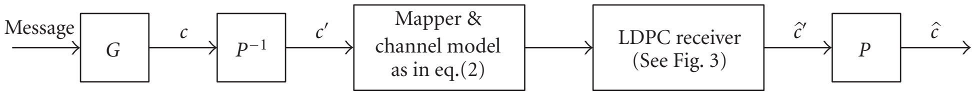
Message
G
c
P^{-1}
c'
Mapper & channel model as in eq.(2)
LDPC receiver (See Fig. 3)
$\hat{c}'$
P
$\hat{c}$

Figure 2: Communication system.

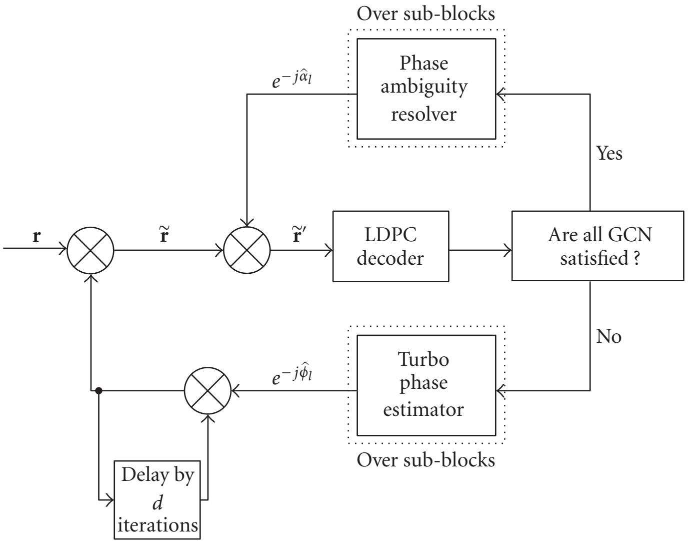

Figure 3: Proposed LDPC receiver architecture.

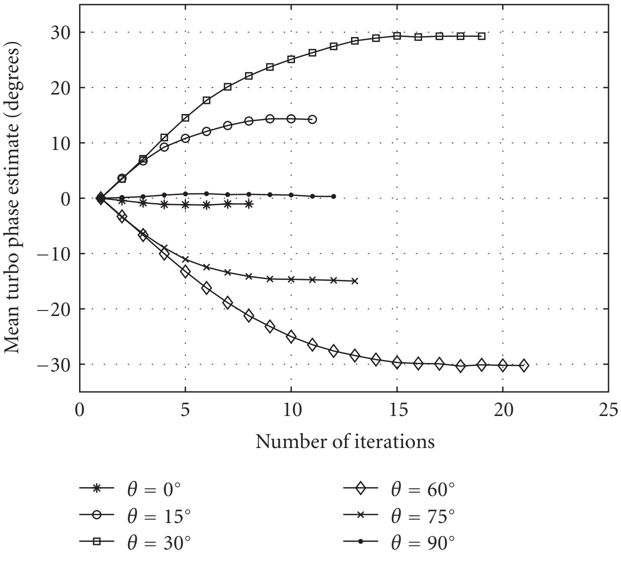

Figure 4: Evolution of turbo phase estimates over sub-blocks during convergence.

phase of this code, the decoder converges to a codeword which is rotationally equivalent to the transmitted codeword. Hence the turbo phase estimator provides a phase estimate whose range lies between $(-\pi/M, \pi/M)$. This is illustrated in Figure 4, which shows the mean trajectories of the turbo phase estimates over a sub-block of 100 symbols at an $E_b/N_0 = 2$ dB under a constant phase offset (θ).

5.2. Computational complexity

The computational complexity of the proposed LDPC receiver can be evaluated as the summation of the complexities of the LDPC decoder and the phase estimator/ambiguity resolver. The computational complexity of the nonbinary LDPC decoder is dominated by the check node decoder with $\mathcal{O}(M^2)$ operations. Reducing the computational complexity of the nonbinary LDPC decoder is an active area of research [13, 14]. In this paper, we concentrate only on the additional complexity involved in the receiver due to the turbo phase estimation in (13) and ambiguity resolution.

Since the decoding algorithm works in the probability domain, the a posteriori probability of the symbols $p(s_k = s_k^m \mid \tilde{r}_k)$ are directly available from the decoder. Given the a posteriori probability vector of length M, for the kth symbol, the soft symbol estimate of the symbol s_k can be calculated according to (14). To estimate N_s soft symbol estimates, we require $2(M-1)N_s$ real additions and $2MN_s$ real multiplications. Given the soft symbol estimates, the evaluation of turbo phase estimate for B sub-blocks requires an additional $4N_s$ real multiplications, $2(N_s - B)$ real additions and B lookup table (LUT) access for evaluating the arg function. Correcting every symbol by the turbo phase estimate requires 4 real multiplications and 2 real additions. Thus the total complexity involved for estimating and correcting a symbol for its phase offset using a turbo phase estimator per iteration (O_{TPE}) is given as

$$O_{\text{TPE}} = [2M+8]^{\times} + \left[2M+4-\frac{2B}{N_s}\right]^{+} + \left[\frac{B}{N_s}\right]^{\text{LUT}}, \qquad (15)$$

where $[\cdot]^{\times}$, $[\cdot]^{+}$, and $[\cdot]^{\text{LUT}}$ correspond to the number of real multiplications, real additions, and look-up table access, respectively. The complexity involved in resolving phase ambiguity per symbol is very small. Also phase ambiguity resolution is required only once per decoding.

Thus the additional complexity of the receiver, mainly due to turbo phase estimation, is relatively small. In the case of the LDPC code described in Section 6, the additional complexity per symbol per iteration is approximately equivalent to $([16]^{\times} + [12]^{+})$ operations, assuming $d = 1$.

6. BER PERFORMANCE OF THE PROPOSED RECEIVER

We constructed a binary LDPC code of $N = 3000$, $K = 1500$, $R = 0.5$ for a sub-block size of $N_b = 100$ symbols. Through simulations, we found that the code with sub-block size of 100 symbols gives the best BER performance for the amounts of phase noise considered in this paper. The degree distributions of this binary code were obtained through EXIT charts [15] such that they converged at an E_b/N_o of 1.3 dB. The variable node and check node distributions, in terms of node perspective, were $\lambda(x) = 0.8047x^3 + 0.0067x^4 + 0.1887x^8$ and $\rho(x) = 0.02x^3 + 0.98x^8$, respectively. The code corresponds to $B = 30$ sub-blocks over the codeword. We replaced the edge gains of this code from the nonzero divisors of $\mathbb{Z}_4$ such that they follow the constraints discussed in Section 3. Turbo phase estimation was done after every iteration ($d = 1$), only during the converging phase. Iterations are performed until the codeword converges, or to a maximum of 200 iterations. However, we found that on an average in the waterfall region, less than 40 iterations are required for convergence. Simulations are performed either until 100 codeword errors are found or up to 500,000 transmissions.

Simulation results in Figure 5 show the performance of our receiver in Figure 3 under phase noise conditions. For a constant phase offset, there is a small degradation of around 0.3 dB from the coherent performance at a BER of 10^{-5}. This loss is due to the proposed application schedule of SPA on the code, which did not include local check nodes during the convergence phase and the degraded performance of the turbo phase estimator with reduced sub-block size. However, thereafter with a small loss, the code is able to tolerate a phase noise with $\sigma_\Delta = 2°$ per symbol.

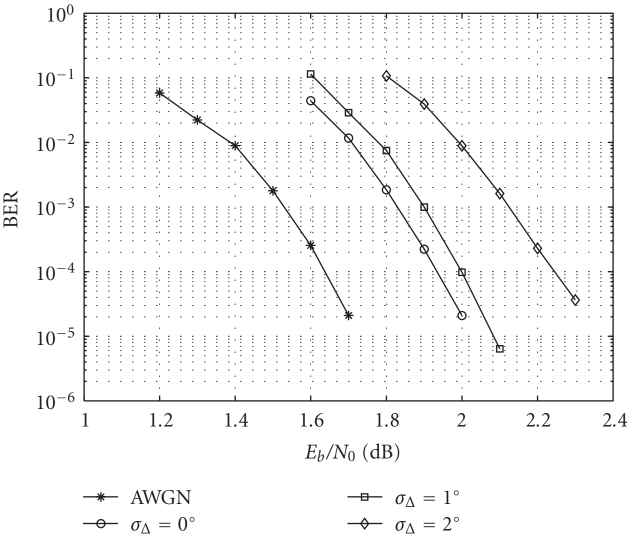

FIGURE 5: Performance of the proposed receiver in Figure 3 with QPSK and the Wiener phase model.

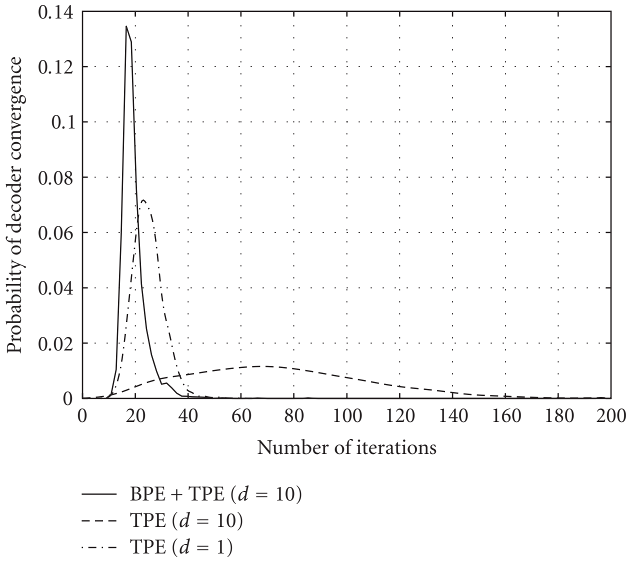

FIGURE 6: Convergence improvement due to an initial blind phase estimator.

7. LOWER COMPLEXITY ITERATIVE RECEIVER

In this section, we show that the computational complexity involved with the turbo phase estimation can be reduced by using a blind phase estimator just once, before the iterative receiver proposed in Figure 3.

7.1. *Comments on initial phase estimation*

The performance of the LCN-based phase ambiguity resolution (PAR) algorithm degrades with the amount of phase offset present on the symbols participating in the LCN. Hence in our earlier work [8], we used a BPE to provide phase estimate for every sub-block of symbols before resolving PAR using the local check nodes. However, in the current work, we are able to delay the PAR on the sub-blocks since the code can converge with the phase ambiguous estimates obtained from the TPE alone. Hence the proposed architecture does not require the use of a blind phase estimator. However, by employing an initial BPE for coarse phase estimation and correction of the sub-blocks, the number of iterations required for convergence can be reduced. Figure 6 illustrates the benefit of blind phase estimation at an $E_b/N_0 = 2.1$ dB with a Wiener phase noise of 1° standard deviation per symbol.

It also shows that the computational complexity due to TPE can be reduced, approximately a factor of 10, by

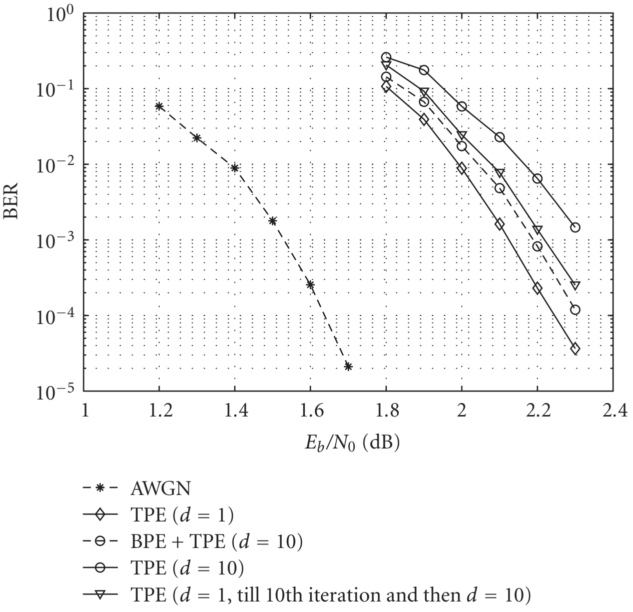

Figure 7: Performance of the low complexity receiver discussed in Section 7 under phase noise with $\sigma_\Delta = 2^\circ$ per symbol.

using the BPE once before the iterative receiver and then periodically using the turbo phase estimator.

7.2. BER performance

The code described in Section 6 was used to simulate the BER performance of the iterative receiver with low computational complexity. The blind phase estimator was used to estimate and correct the phase disturbance present in each sub-block of the received symbol vector, following which the phase-corrected symbol vector was fed into the iterative receiver in Figure 3. During the convergence phase, turbo phase estimates were obtained once in 10 iterations ($d = 10$). At $\sigma_\Delta = 2^\circ$ per symbol, Figure 7 shows the advantage of a blind phase estimator in terms of BER performance. The result compares three distinct cases with the normal receiver, where turbo phase estimation was performed in every iteration. The presence of blind phase estimator allows us to include turbo phase estimator only once in every 10 iterations with a small loss of 0.05 dB. However, without blind phase estimator, performing turbo phase estimation only once in every 10 iterations shows significant degradation. As shown, the performance can be improved by including turbo phase estimation for more iterations, particularly the early stages of the decoder, during which the LDPC decoder provides a lot of new information regarding the symbols.

8. CONCLUSION

In this paper, we addressed the problem of LDPC code-based iterative decoding under phase noise channels from a code perspective. We proposed construction of ring-based codes for higher-order modulations that work well with sub-block phase estimation techniques of low complexity. The code was constructed using the new constraints outlined in Section 3 such that it not only converges under sub-block phase rotations, but also estimates them. We also showed the property of ring-based check nodes under the presence of phase ambiguity based on their edge gains in a generalized manner. As part of our future work, we are looking at ways to construct code without explicitly constructing local check nodes for PAR. The sub-block size used in the simulation results shown earlier, has not been optimized and we believe that the method can be extended to adjust the observation interval and phase model depending on the amount of phase noise.

ACKNOWLEDGMENTS

The authors wish to acknowledge helpful discussions with Dr. Steven S. Pietrobon on this topic and also thank reviewers for their useful comments.

REFERENCES

[1] R. Gallager, “Low density parity-check codes,” *IEEE Transactions on Information Theory*, vol. 8, no. 1, pp. 21–28, 1962.

[2] G. Colavolpe, A. Barbieri, and G. Caire, “Algorithms for iterative decoding in the presence of strong phase noise,” *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 9, pp. 1748–1757, 2005.

[3] A. Barbieri, G. Colavolpe, and G. Caire, “Joint iterative detection and decoding in the presence of phase noise and frequency offset,” *IEEE Transactions on Communications*, vol. 55, no. 1, pp. 171–179, 2007.

[4] I. Motedayen-Aval and A. Anastasopoulos, “Polynomial-complexity noncoherent symbol-by-symbol detection with application to adaptive iterative decoding of turbo-like codes,” *IEEE Transactions on Communications*, vol. 51, no. 2, pp. 197–207, 2003.

[5] R. Nuriyev and A. Anastasopoulos, “Rotationally invariant and rotationally robust codes for the AWGN and the noncoherent channel,” *IEEE Transactions on Communications*, vol. 51, no. 12, pp. 2001–2010, 2003.

[6] W. G. Cowley and M. S. C. Ho, “Transmission design for Doppler-varying channels,” in *Proceedings of the 7th Australian Communications Theory Workshop (AusCTW ’06)*, pp. 110–113, Perth, Australia, February 2006.

[7] M. Franceschini, G. Ferrari, R. Raheli, and A. Curtoni, “Serial concatenation of LDPC codes and differential modulations,” *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 9, pp. 1758–1768, 2005.

[8] S. Karuppasami and W. G. Cowley, “LDPC code-aided phase-ambiguity resolution for QPSK signals affected by a frequency offset,” in *Proceedings of the 8th Australian Communications Theory Workshop (AusCTW ’07)*, pp. 47–50, Adelaide, Australia, February 2007.

[9] N. Noels, V. Lottici, A. Dejonghe, et al., “A theoretical framework for soft-information-based synchronization in iterative (turbo) receivers,” *EURASIP Journal on Wireless Communications and Networking*, vol. 2005, no. 2, pp. 117–129, 2005.

[10] S. Karuppasami, W. G. Cowley, and S. S. Pietrobon, “LDPC code construction and iterative receiver techniques for channels with phase noise,” in *Proceedings of the 67th IEEE Vehicular Technology Conference (VTC ’08)*, Singapore, May 2008.

[11] D. Sridhara and T. E. Fuja, “LDPC codes over rings for PSK modulation,” *IEEE Transactions on Information Theory*, vol. 51, no. 9, pp. 3209–3220, 2005.

[12] M. C. Davey and D. MacKay, “Low-density parity check codes over GF(q),” *IEEE Communications Letters*, vol. 2, no. 6, pp. 165–167, 1998.

[13] D. Declercq and M. Fossorier, “Decoding algorithms for nonbinary LDPC codes over GF(q),” *IEEE Transactions on Communications*, vol. 55, no. 4, pp. 633–643, 2007.

[14] A. Voicila, D. Declercq, F. Verdier, M. Fossorier, and P. Urard, “Low-complexity, low-memory EMS algorithm for nonbinary LDPC codes,” in *Proceedings of IEEE International Conference on Communications (ICC ’07)*, pp. 671–676, Glasgow, Scotland, UK, June 2007.

[15] S. ten Brink, G. Kramer, and A. Ashikhmin, “Design of low-density parity-check codes for modulation and detection,” *IEEE Transactions on Communications*, vol. 52, no. 4, pp. 670–678, 2004.