Blockwise Noncoherent M-APSK Channel: Coding Scheme for Iterative Receivers

Daniel C. Cunha and Jaime Portugheis

***Resumo*— Um esquema de codificação de faixa eficiente para receptores iterativos foi investigado. Canais não-coerentes M-APSK (do inglês, *M-ary Amplitude Phase Shift Keying*) foram considerados. O esquema propõe uma nova abordagem para um receptor iterativo baseado em grafos-fatores. A abordagem considera o uso de codificação diferencial não binária. Uma comparação de desempenho do esquema para códigos LDPC (do inglês, *Low-Density Parity-Check*) com comprimentos curtos será descrita.**

***Palavras-Chave*— Canais não coerentes, constelações M-APSK, decodificação iterativa, grafos-fatores.**

***Abstract*— A bandwidth efficient coding scheme for iterative receivers is investigated. Blockwise noncoherent M-APSK channels are considered. The scheme proposes a new approach for a iterative receiver based on factor graphs. The approach considers the use of differential encoding and non-binary modulations. A comparison of the performance scheme for LDPC codes with short lengths is described.**

***Keywords*— Noncoherent channels, M-APSK constellations, iterative decoding, factor graphs.**

I. Introduction

In many communication scenarios it is difficult to acquire the carrier phase of the received signal. In these scenarios, it is commonly assumed that the unknown carrier phase rotation is constant over a block of L symbols and independent from block to block. This scenario justifies the use of a channel model known as the blockwise noncoherent channel. The communication system modulation is usually M-ary Phase Shift Keying (M-PSK), but in the case of high spectral efficiencies, M-ary Amplitude and Phase-Shift Keying (M-APSK) with independent phase and amplitude modulations is preferable.

The capacity of a noncoherent AWGN channel in the case of input symbols drawn from an M-PSK and from an M-APSK modulation has been investigated in [11] and in [12], respectively. The results of both [11] and [12] show that for large L the capacity of a coherent channel is approached by the noncoherent one. The results of [12] indicate that the coherent capacity is approximated faster when differential encoding is used. Motivated by this latest result and by our interest in high spectral efficiency systems, a bandwidth efficient coding scheme employing differential encoding is investigated in this paper. This scheme is composed of a serial concatenation of a *Low-Density Parity-Check* (LDPC) code, an interleaver and a differential encoder. Our scheme is similar to that used in [4] for block Rayleigh fading channels. Moreover, since it is well known that joint channel estimation and decoding can be a useful technique to obtain good performance for noncoherent channels, an iterative receiver described by a factor graph is also developed [1].

In this paper we begin to analyze the proposed coding scheme with short length codes. The analysis consider phase quantization, interleaving strategy, number of iterations and schedules for the sum product algorithm associated with the receiver factor graph. The outline of the paper is organized as follows. We define the signal constellations and the channel model in Section II. In Section III, we describe the transmitter and the iterative receiver of the system, defining all the messages of the graph and the schedule for the sum product algorithm. In Section IV, some numerical results are presented. To summarize, conclusions are drawn in Section V.

II. Signal Constellations and Channel Model

We consider M-APSK constellation diagrams which consist of N different amplitude rings, each one with P phase values. The amplitude values of the rings differ by a constant factor r denominated *ring ratio*. Such constellations will be denoted by M-APSK (N, P), with $M = NP$.

The input of the channel is a vector of length L, $\mathbf{S} = [s_1, s_2, ..., s_L]$, whose components $s_l = a_l e^{j\phi_l}$ represent APSK-modulated symbols. The amplitudes a_l can assume one of N possible discrete values and ϕ_l can assume one of P discrete phases, so the signal s_l belongs to a M-APSK (N, P) constellation. The output is also a vector of length $L, \mathbf{R} = [r_1, r_2, ..., r_L]$, whose components may be expressed as

$$r_l = s_l \exp{(j\theta)} + n_l \quad , \quad l = 1, 2, ..., L \qquad (1)$$

where θ is a phase shift introduced by the channel uniformely distributed over the interval $[0, 2\pi)$. The phase remains constant over a block of L symbols and it is independent from block to block. The complex additive noise n_l are independent circularly symmetric Gaussian variables, whose real and imaginary parts are each zero mean with variance $\sigma^2 = N_0/2$.

III. Coding Scheme for Iterative Receivers

The block diagram of the communication system proposed in this work is illustrated in Fig. 1 . Following, we will describe the transmitter and the iterative receiver.

Daniel C. Cunha, Departamento de Engenharia Elétrica, Escola Politécnica (POLI), UPE, E-mail: dccunha@upe.poli.br. Jaime Portugheis, Departamento de Comunicações, FEEC, UNICAMP, E-mail: jaime@decom.fee.unicamp.br. Este trabalho foi parcialmente financiado pela FAPESP (03/05385-6).

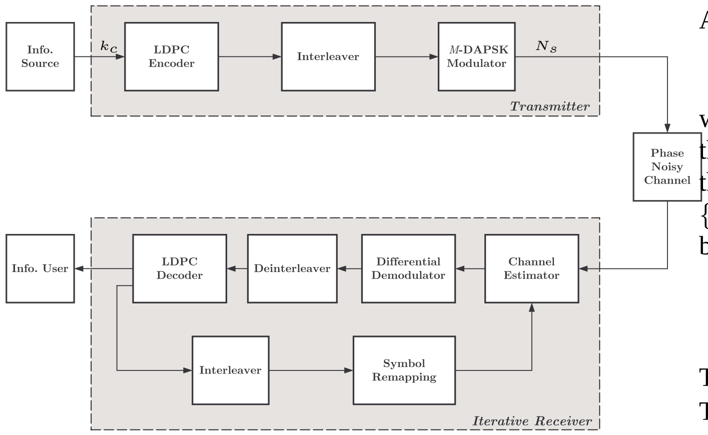

Fig. 1. System block diagram.

A. Transmitter

The transmitter is composed of a LDPC encoder, an interleaver and a M-DAPSK modulator. The LDPC codes used in the transmitter are rate one-half codes obtained from [2]. The interleaver is the classical block interleaver [3]. The *M-ary Differential Amplitude Phase Shift Keying* (M-DAPSK) modulator is composed of a M-APSK mapper and a differential encoder.

The function of the mapper is to generate the M-APSK symbols from the scrambled coded bits at the output of the interleaver. Fig. 2(a) shows the signal representation of 8-APSK$(2,4)$ constellation and Fig. 2(b) shows the symbol mapping used for the 8-DAPSK modulator. Each 8-APSK symbol is mapped by a sequence of three bits $\mathbf{b}_i = [b_i^1, b_i^2, b_i^3]$. The most significative bit, b_i^1, is used to represent the amplitude level of the symbol whereas the other two bits, b_i^2 and b_i^3, indicate the phase value. At each amplitude level, the phase mapping is done so that just one bit change from a phase value to the nearest one as it is illustrated in Fig. 2(b).

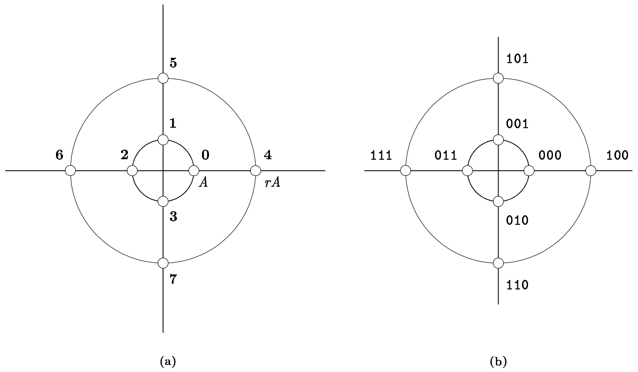

Fig. 2. (a) Signal labeling of 8-APSK$(2,4)$ constellation. (b) Symbol mapping.

The differential encoding used by the 8-DAPSK modulator is described as follows. Consider s'_i the i-th baseband differentially encoded symbol. It is obtained from s'_{i-1}, the previous differential symbol, and s_{i-1}, the symbol from the mapper. We define the label $z'_i, z'_i \in \{0,1,\ldots,7\}$, associated to the symbol s'_i according to Fig. 2(a). Likewise, the labels z_{i-1} and z'_{i-1} are related to the symbols s_{i-1} and s'_{i-1}, respectively. Accordingly, we can estabilish that

$$z'_i = z_{i-1} \oplus_M z'_{i-1} \ , \tag{2}$$

where the operator $\oplus_M$ represents modulo M sum. Although this sum is not applied to complex numbers, we will consider that the symbols s'_i, s_{i-1} and s'_{i-1} assume values in the set $\{0,1,\ldots,7\}$ and the circumstances will make the distinction between symbols and labels.

On the other hand, we can represent s'_i as

$$s'_i = a'_i \exp(j\phi'_i) \ . \tag{3}$$

This representation is also valid for symbols s'_{i-1} and s_{i-1}. The amplitude level a'_{i-1} is defined by

$$a'_{i-1} = r^{D'_{i-1}} A \ . \tag{4}$$

Similarly, $a_{i-1} = r^{D_{i-1}} A$. The variables D'_{i-1} and D_{i-1} depend on N. Considering 8-APSK$(2,4)$ constellation, we have $N = 2$. Therefore, D'_{i-1} and D_{i-1} are binary variables and $D_{i-1} = b_{i-1}^1$. We can define the variable Δ_i representing the amplitude differential encoding as

$$\Delta_i = D_{i-1} \oplus_2 D'_{i-1} \ . \tag{5}$$

Table I indicates the dependence between Δ_i and a'_i.

TABLE I

DEPENDENCE BETWEEN Δ_i AND THE AMPLITUDE LEVEL a'_i FOR 8-APSK$(2,4)$ CONSTELLATION.

Δ_i	a'_i
0	A
1	rA

As we have said, the bits b_{i-1}^2 and b_{i-1}^3 are used to map the phase value, specifically the phase ϕ_{i-1}. Table II indicates how this phase mapping occurs.

TABLE II

DIFFERENTIAL PHASE MAPPING FOR 8-APSK$(2,4)$ CONSTELLATION.

$b_{i-1}^2 b_{i-1}^3$	ϕ_{i-1}
00	0
01	$\pi/2$
11	π
10	$3\pi/2$

In face of this, the phase ϕ'_i is given by the expression

$$\phi'_i = (\phi''_i)_{mod\ 2\pi} \ , \tag{6}$$

where $\phi''_i = \phi'_{i-1} + \phi_{i-1}$. If $\phi''_i \geq 2\pi$, we need to invert the value of Δ_i before computing the amplitude level a'_i.

When differential encoding is applied, each block has a reference symbol in its beginning, denoted by s'_b, $b = (k-1)L + 1$, $k = 1, 2, \ldots, N_b$, where N_b is the number of transmitted blocks. Without loss of generality, we may refer to the first block and consider $s'_1 = 0$. That is, $a'_1 = A$ and $\phi'_1 = 0$. To this end, each symbol s'_i is differentially encoded considering the previous symbol s'_{i-1} as reference. Each block of symbols does not depend on the previous block, i.e., the first symbol of the block is not encoded from the last symbol of the previous block. For this reason, each block has its own fixed and independent reference symbol.

To obtain the transmission rate of the system denoted as R_s, consider the Fig. 1. The transmission rate is defined by the equation

$$R_s = \frac{k_c}{N_s} \ , \tag{7}$$

where k_c is the number of information bits at the input of the LDPC encoder and $N_s = N_b\, L$ is the number of modulated symbols at the output of the transmitter.

Since $N_b = n_c/m\,(L-1)$, we have

$$R_s = r_c\, m\, \frac{(L-1)}{L} \quad bits/symbol \ , \tag{8}$$

where $r_c = k_c/n_c$ is the LDPC code rate and m is the number of M-APSK mapping bits.

B. Iterative Receiver

It is well known that for many communication systems joint demodulation and decoding is required for optimum performance. Iterative algorithms that approximate joint channel estimation, demodulation, and decoding, are usually called *iterative receivers*. We present an approach based on factor graphs for deriving iterative message-passing receiver algorithms for channel estimation and decoding as it was done in [1]. The difference, in our case, is that we consider differential encoding and non-binary modulations.

Factor graph-based descriptions of iterative receivers appear to be extremely compact and intuitively attractive. Factor graphs are used to represent function factorizations graphically. The sum-product algorithm computes the marginals of the function that the graph represents using message passing on the graph.

Iterative receivers can be defined by a function named *objective function*. For this reason, there is a factor graph that represents this function. Our purpose is to factor the objective function and apply the sum-product algorithm on the factored graph.

In our case, the factored objective function, G_s, is given by the equation

$$\begin{aligned} G_s(\mathbf{s}', \boldsymbol{\Theta}) \;\; = \;\; & \prod_{t=1}^{m_c} I\left\{\mathbf{h}_t \cdot \mathbf{x} = 0\right\} \cdot \prod_{i=1}^{N_s} p(r_i |\, s_i', \boldsymbol{\Theta}_{\lceil i/L \rceil}) \\ \cdot \;\; & I\{s_i = V(\mathbf{b}_i)\} \cdot I\{s_i' = s_{i-1} \oplus_M s_{i-1}'\} \\ \cdot \;\; & I\{s_b' = 0\} \cdot \prod_{k=1}^{N_b} p(\boldsymbol{\Theta}_k) \ , \end{aligned} \tag{9}$$

where $\mathbf{s}'$ is the channel input vector, $\boldsymbol{\Theta}$ represents the channel state, and $I\{\cdot\}$ is denominated *indicator function*. Indicator functions are obtained from behavorial modeling of the system and are very important when factor graphs are applied to coding [5].

The term

$$\prod_{t=1}^{m_c} I\left\{\mathbf{h}_t \cdot \mathbf{x} = 0\right\} = I_C(\mathbf{x}) \tag{10}$$

represents the indicator function of the LDPC code, where $\mathbf{h}_t$ is the t-th line of the code's parity check matrix $\mathbf{H}$ and m_c is the number of the lines of $\mathbf{H}$. The function $I\{s_i = V(\mathbf{b}_i)\}$ refers to the mapping of each subset $\mathbf{b}_i$ of m bits of the codeword $\mathbf{x}$ into a modulation symbol s_i whereas the functions $I\{s_i' = s_{i-1} \oplus_M s_{i-1}'\}$ and $I\{s_b' = 0\}$ represent the differential encoding. In its turn, the term $I\{s_b' = 0\}$ refers to the reference symbol of each block of symbols. Finally, the terms

$$\prod_{i=1}^{N_s} p(r_i |\, s_i', \boldsymbol{\Theta}_{\lceil i/L \rceil}) \tag{11}$$

and

$$\prod_{k=1}^{N_b} p(\boldsymbol{\Theta}_k) \tag{12}$$

are the transition probability density and the channel state probability, respectively. In the expression (11), $\lceil \cdot \rceil$ is the ceil operator.

The objective function of the iterative receiver may be represented by the factor graph illustrated in Fig. 3. Circles represent variable nodes while black squares represent function nodes. In the lower part we have the subgraph 1 representing the LDPC decoder. It is composed of the horizontal bar C, that represents the set of check nodes, and variable nodes $(x_1, \ldots, x_{n_c})$ above the bar. The block Π_e represents the system interleaver. At last, we have the subgraph 2 grouping the symbol mapper, the differential encoder, and the channel estimator.

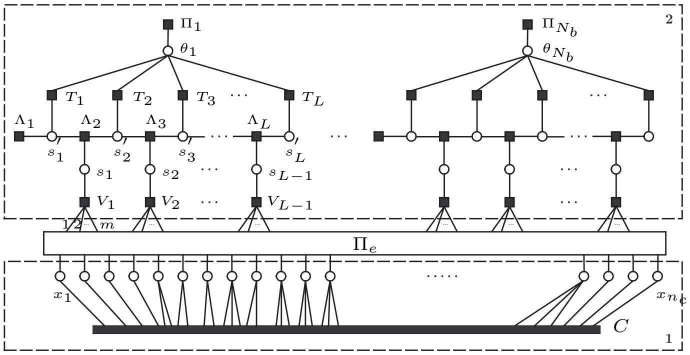

Fig. 3. Factor graph that represents the iterative receiver for blockwise noncoherent AWGN channel considering differential encoding. $C \triangleq I_C(\mathbf{x})\,; T_i \triangleq p(r_i|s_i', \theta_{\lceil i/L \rceil})\,; \Pi_k \triangleq p(\theta_k)\,; \Lambda_i \triangleq I\{s_i' = s_{i-1} \oplus_M s_{i-1}'\}; \Lambda_1 \triangleq I\{s_1' = 0\}; V_i \triangleq I\{s_i = V(\mathbf{b}_i)\}$.

1) Message Passing: The messages exchanged by variable and check nodes follow the rules defined in [6]. The message passing in the subgraph 1, i.e., in the graph of the LDPC decoder, is described in [7]. We will focus on the message passing in the subgraph 2.

The variable node θ_k sends the message $\mu_{\theta_k \to T_i}(\theta_k)$, defined by

$$\mu_{\theta_k \to T_i}(\theta_k) = \mu_{\Pi_k \to \theta_k}(\theta_k) \cdot \prod_{\substack{j=(k-1)L+1 \\ j \neq i}}^{kL} \mu_{T_j \to \theta_k}(\theta_k) \ , \tag{13}$$

where $\mu_{T_j\rightarrow\theta_k}(\theta_k)$ is the message sent from T_j to θ_k , and $\mu_{\Pi_k\rightarrow\theta_k}(\theta_k)$ is the message sent from Π_k to θ_k . The message $\mu_{\Pi_k\rightarrow\theta_k}(\theta_k)$ describes the phase distribution introduced by the channel.

The function node T_i is characterized by the transition probability density, and is represented by

$$\begin{aligned} T_i(r_i, s_i^{'}, \theta_k) &\triangleq p(r_i|\, s_i^{'}, \theta_k) \\ &= \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{\left\|r_i - s_i^{'}e^{j\theta_k}\right\|^2}{2\sigma^2}\right) \ . \quad (14) \end{aligned}$$

According to the update rules of the sum-product algorithm [6], the message $\mu_{T_i\rightarrow\theta_k}(\theta_k)$ may be written as

$$\begin{aligned} \mu_{T_i\rightarrow\theta_k}(\theta_k) &= \sum_{\sim\{\theta_k\}} T_i(r_i, s_i^{'}, \theta_k)\cdot\mu_{s_i^{'}\rightarrow T_i}(s_i^{'}) \\ &= \sum_{s_i^{'}} p(r_i|\, s_i^{'}, \theta_k)\cdot\mu_{s_i^{'}\rightarrow T_i}(s_i^{'}) \ . \quad (15) \end{aligned}$$

Meanwhile, the message $\mu_{T_i\rightarrow s_i^{'}}(s_i^{'})$ is defined by

$$\mu_{T_i\rightarrow s_i^{'}}(s_i^{'}) = \int_0^{2\pi} p(r_i|\, s_i^{'}, \theta_k)\mu_{\theta_k\rightarrow T_i}(\theta_k)d\theta_k \ . \quad (16)$$

The integral substitutes the summary because θ_k is a continuous variable. Depending on the complexity of $\mu_{\theta_k\rightarrow T_i}(\theta_k)$, computing (16) can become difficult. Therefore, we will use a canonical distribution to the message $\mu_{\theta_k\rightarrow T_i}(\theta_k)$ [1]. We will consider a quantized canonical distribution where θ_k has Z discrete values. Consequently, the parameterized message $\mu_{\theta_k\rightarrow T_i}^{'}(\theta_k)$ is given by

$$\mu_{\theta_k\rightarrow T_i}^{'}(\theta_k) = \sum_{z=1}^{Z} a_z\delta(\theta_k - \widehat{\theta}_{kz}) \ . \quad (17)$$

Another good approximations, using Fourier and Tikhonov parameterizations, were applied in iterative algorithms proposals to another models of phase-noisy channels [8], [9].

To find the coefficients a_z, we compute the nominal message given by (13), assuming discrete phases $\widehat{\theta}_{kz}$. In other words, the coefficients a_z are given by

$$\begin{aligned} a_z &= \mu_{\theta_k\rightarrow T_i}(\theta_k = \widehat{\theta}_{kz}) = \mu_{\Pi_k\rightarrow\theta_k}(\theta_k = \widehat{\theta}_{kz}) \\ &\cdot \prod_{\substack{j=(k-1)L+1 \\ j\neq i}}^{kL} \mu_{T_j\rightarrow\theta_k}(\theta_k = \widehat{\theta}_{kz}) \ . \quad (18) \end{aligned}$$

Definitely, the coefficients a_z are the parameters to be passed during this stage of the algorithm and the message $\mu_{T_i\rightarrow s_i^{'}}(s_i^{'})$ may be written as

$$\begin{aligned} \mu_{T_i\rightarrow s_i^{'}}(s_i^{'}) &= \sum_{\sim\{s_i^{'}\}} T_i(r_i, s_i^{'}, \theta_k)\mu_{\theta_k\rightarrow T_i}(\theta_k) \\ &= \sum_{z=1}^{Z} a_z \exp\left(-\frac{\left\|r_i - s_i^{'}e^{j\widehat{\theta}_{kz}}\right\|^2}{2\sigma^2}\right). \quad (19) \end{aligned}$$

After computing $\mu_{T_i\rightarrow s_i^{'}}(s_i^{'})$, the messages from the part of the graph related to channel estimator are ended and the differential demodulation subgraph begins to compute its messages. This stage is characterized by the execution of the forward/backward algorithm [6]. The forward stage computes the messages $\mu_{s_i^{'}\rightarrow\Lambda_{i+1}}(s_i^{'})$ and $\mu_{\Lambda_i\rightarrow s_i^{'}}(s_i^{'})$.

The message $\mu_{s_i^{'}\rightarrow\Lambda_{i+1}}(s_i^{'})$ is given by the equation

$$\mu_{s_i^{'}\rightarrow\Lambda_{i+1}}(s_i^{'}) = \mu_{\Lambda_i\rightarrow s_i^{'}}(s_i^{'})\cdot\mu_{T_i\rightarrow s_i^{'}}(s_i^{'}) \ . \quad (20)$$

Specifically at the beginning of each k-th block of symbols, we have that the message $\mu_{\Lambda_i\rightarrow s_i^{'}}(s_i^{'})$ is given by

$$\mu_{\Lambda_i\rightarrow s_i^{'}}(s_i^{'}) = \begin{cases} 1, & s_i^{'} = 0 \\ 0, & s_i^{'} = 1, 2, \ldots M-1 \end{cases} \ , \quad (21)$$

where $i = (k-1)L+1, k = 1, 2, \ldots, N_b$. The initial values to the message $\mu_{\Lambda_i\rightarrow s_i^{'}}(s_i^{'})$ in (21) are due to the knowledge of the first symbol of each block, the reference symbol, by the receiver. For the other block symbols, the message $\mu_{\Lambda_i\rightarrow s_i{'}}(s_i^{'})$ is computed by the equation

$$\begin{aligned} \mu_{\Lambda_i\rightarrow s_i^{'}}(s_i^{'}) &= \sum_{\sim\{s_i^{'}\}} \Lambda_i(s_{i-1}^{'}, s_{i-1}, s_i^{'})\cdot\mu_{s_{i-1}^{'}\rightarrow\Lambda_i}(s_{i-1}^{'}) \\ &\cdot \quad \mu_{s_{i-1}\rightarrow\Lambda_i}(s_{i-1}) \ , \quad (22) \end{aligned}$$

where $\Lambda_i(s_{i-1}^{'}, s_{i-1}, s_i^{'})$ is the indicator function $I\{s_i^{'} = s_{i-1} \oplus_M s_{i-1}^{'}\}$.

After the forward stage, the backward stage starts with each variable node $s_i^{'}$ and each function node Λ_i responsible for processing two messages. First, the variables nodes $s_i^{'}$ update the messages $\mu_{s_i^{'}\rightarrow T_i}(s_i^{'})$ and compute $\mu_{s_i^{'}\rightarrow\Lambda_i}(s_i^{'})$. The updating of $\mu_{s_i^{'}\rightarrow T_i}(s_i^{'})$ is given by

$$\mu_{s_i^{'}\rightarrow T_i}(s_i^{'}) = \mu_{\Lambda_{i+1}\rightarrow s_i^{'}}(s_i^{'})\cdot\mu_{\Lambda_i\rightarrow s_i^{'}}(s_i^{'}) \ , \quad (23)$$

except for the last node of each block where $\mu_{s_i^{'}\rightarrow T_i}(s_i^{'}) = \mu_{\Lambda_i^{'}\rightarrow s_i^{'}}(s_i^{'})$, because it is a variable node with only two neighbors. The other message computed by the variable nodes $s_i^{'}$, $\mu_{s_i^{'}\rightarrow\Lambda_i}(s_i^{'})$, is given by

$$\mu_{s_i^{'}\rightarrow\Lambda_i}(s_i^{'}) = \mu_{T_i\rightarrow s_i^{'}}(s_i^{'})\cdot\mu_{\Lambda_{i+1}\rightarrow s_i^{'}}(s_i^{'}) \ . \quad (24)$$

Following, the function nodes Λ_i compute the messages $\mu_{\Lambda_i\rightarrow s_{i-1}^{'}}(s_{i-1}^{'})$ and $\mu_{\Lambda_i\rightarrow s_{i-1}}(s_{i-1})$, where

$$\begin{aligned} \mu_{\Lambda_i\rightarrow s_{i-1}^{'}}(s_{i-1}^{'}) = \sum_{\sim\{s_{i-1}^{'}\}} &\Lambda_i(s_{i-1}^{'}, s_{i-1}, s_i^{'})\cdot\mu_{s_i^{'}\rightarrow\Lambda_i}(s_i^{'}) \\ &\cdot \quad \mu_{s_{i-1}\rightarrow\Lambda_i}(s_{i-1}) \quad (25) \end{aligned}$$

and

$$\begin{aligned} \mu_{\Lambda_i\rightarrow s_{i-1}}(s_{i-1}) = \sum_{\sim\{s_{i-1}\}} &\Lambda_i(s_{i-1}^{'}, s_{i-1}, s_i^{'})\cdot\mu_{s_{i-1}^{'}\rightarrow\Lambda_i}(s_{i-1}^{'}) \\ &\cdot \quad \mu_{s_i^{'}\rightarrow\Lambda_i}(s_i^{'}) \ . \quad (26) \end{aligned}$$

After a backward stage, the messages $\mu_{\Lambda_i\rightarrow s_{i-1}}(s_{i-1})$ go to the lower part toward the LDPC decoder. Since the variable nodes s_i have two neighbors, they just forward the messages $\mu_{\Lambda_i\rightarrow s_{i-1}}(s_{i-1})$, renamed as $\mu_{s_i\rightarrow V_i}(s_i)$, to the function nodes V_i . In addition to that, the nodes s_i process the message $\mu_{s_i \to \Lambda_{i+1}}(s_i)$ that is equal to $\mu_{V_i \to s_i}(s_i)$.

The message $\mu_{V_i \to s_i}(s_i)$ is given by

$$\mu_{V_i \to s_i}(s_i) = \sum_{\sim\{s_i\}} I\{s_i = V(\mathbf{b}_i)\} \prod_{n=1}^{m} \mu_{b_i^n \to V_i}(b_i^n) \ . \qquad (27)$$

The other message processed by nodes V_i , $\mu_{V_i \to b_i^\alpha}(b_i^\alpha)$, is obtained by

$$\mu_{V_i \to b_i^\alpha}(b_i^\alpha) = \sum_{\sim\{b_i^\alpha\}} I\{s_i = V(\mathbf{b}_i)\} \mu_{s_i \to V_i}(s_i) \prod_{\substack{n=1 \\ n \neq \alpha}}^{m} \mu_{b_i^n \to V_i}(b_i^n) \qquad (28)$$

For 8-DAPSK modulation, $m = 3$ in the equations (27) and (28).

To end the description, we have the messages $\mu_{b_i^\alpha \to V_i}(b_i^\alpha)$ that are messages sent from the variable nodes representing codewords to the modulation mapping nodes. This message corresponds to *a priori* probabilities of the coded bits and is given by

$$\mu_{b_i^\alpha \to V_i}(b_i^\alpha) = P(b_i^\alpha) \ , \qquad (29)$$

where

$$\mu_{b_i^\alpha \to V_i}(b_i^\alpha = 0) = \frac{\exp(LLR(b_i^\alpha))}{1 + \exp(LLR(b_i^\alpha))} \qquad (30)$$

and

$$\mu_{b_i^\alpha \to V_i}(b_i^\alpha = 1) = \frac{1}{1 + \exp(LLR(b_i^\alpha))} \ . \qquad (31)$$

The *Log-Likelihood Ratio* (LLR) is expressed by

$$LLR(b_i^\alpha) \triangleq \ln\left(\frac{P(b_i^\alpha = 0)}{P(b_i^\alpha = 1)}\right) \ . \qquad (32)$$

2) Schedule: Different from a cycle-free factor graph, the sum-product algorithm begins at the nodes x_i and $s_i^{'}$. The messages $\mu_{s_i^{'} \to T_i}(s_i^{'})$ are computed by

$$\mu_{s_i^{'} \to T_i}(s_i^{'}) = \begin{cases} 1, & s_i^{'} = 0 \\ 0, & s_i^{'} = 1, 2, \ldots M-1 \end{cases} \ , \qquad (33)$$

where $i = (k-1)L + 1, k = 1, 2, \ldots, N_b$. The other symbols $s_i^{'}$ of each block send initial messages $\mu_{s_i^{'} \to T_i}(s_i^{'}) = 1/M$. Meanwhile, the messages $\mu_{b_i^\alpha \to V_i}(b_i^\alpha)$ are computed by the equations (30) and (31), considering a initial value of $LLR(b_i^\alpha)$ equal to zero.

After that, the nodes T_i , Π_k , and V_i process the messages $\mu_{T_i \to \theta_k}(\theta_k)$, $\mu_{\Pi_k \to \theta_k}(\theta_k)$ and $\mu_{V_i \to s_i}(s_i)$.

The next stage of the schedule is responsible for computing messages $\mu_{\theta_k \to T_i}(\theta_k)$ and $\mu_{s_i \to \Lambda_{i+1}}(s_i)$. Later, the nodes T_i process messages $\mu_{T_i \to s_i^{'}}(s_i^{'})$. With these messages, the forward/backward stage starts and the messages $\mu_{\Lambda_i \to s_i^{'}}(s_i^{'})$ and $\mu_{s_i^{'} \to \Lambda_{i+1}}(s_i^{'})$ are computed in the forward stage.

The backward stage, that is responsible for messages $\mu_{s_i^{'} \to \Lambda_i}(s_i^{'})$, $\mu_{s_i^{'} \to T_i}(s_i^{'})$, $\mu_{\Lambda_i \to s_{i-1}}(s_{i-1})$ and $\mu_{\Lambda_i \to s_{i-1}^{'}}(s_{i-1}^{'})$, comes next. From this point, the flow of messages goes down to compute $\mu_{s_i \to V_i}(s_i)$. After that, the nodes V_i process the messages $\mu_{V_i \to b_i^\alpha}(b_i^\alpha)$ through the interleaver.

Before entering in the LDPC subgraph, the messages arriving at the nodes x_i are converted to log-domain by the equation

$$L_c(x_i) = \ln\left(\frac{\mu_{V_i \to x_i}\,(x_i = 0)}{\mu_{V_i \to x_i}\,(x_i = 1)}\right) \ , \qquad (34)$$

providing the initial LLRs of the log sum-product algorithm. At this point, the messages in LDPC subgraph follow the rules defined in [10].

Finally, the algorithm checks the stop condition, i.e., if $\widehat{\mathbf{x}}\mathbf{H}^T = \mathbf{0}$. If it is not verified, the algorithm computes the messages $\mu_{x_i \to V_i}(x_i)$, or equivalently, $\mu_{b_i^\alpha \to V_i}(b_i^\alpha)$, using (30) and (31), and considering $LLR(b_i^\alpha) = L(Q_i)$ [10]. Then, a new iteration begins.

IV. Simulation Results

In this section, we present the system performance considering the proposed joint demodulation and decoding algorithm. The performance is assessed by computer simulations in terms of *Bit-Error Rate* (BER) versus E_b/N_0 , E_b being the received signal energy per information bit. For simplicity, we assume the transmission of all-zero codeword.

To begin with, we consider the rate one-half LDPC code $(96, 48)$. The M-APSK modulation is the 8-APSK$(2, 4)$ with uniform input distribution and ring ratio r=2.42 . The phase introduced by the channel is constant over a block of $L = 9$ symbols. Fig. 4 shows the system performance for three levels of phase quantization. The performance related to the number of iterations of the algorithm is also shown. We consider a block interleaver with $N_r = 4$ rows and $N_c = 24$ columns. The estimative of the channel capacity illustrated in Fig. 4 is obtained from an average of the upper and lower bounds defined in [12].

By the curves of Fig. 4, we note that, for a fixed value of the phase quantization level (Z), the increase of the number of iterations (N_{it}) improves the system performance. The coding gain obtained, with a increase of N_{it} by a factor of 10 and considering $BER = 10^{-3}$, is approximately 2.55 dB, 1.5 dB and 2.3 dB , respectively for $Z = 8, 16$ and 32. In addition, the variation of phase quantization level, given a fixed number of iterations, also improves the system performance. For $N_{it} = 500$, the coding gain obtained with the increase of Z from 8 to 16 is 0.7 dB, and from 16 to 32, is 0.4 dB . In spite of being a decreasing gain with the increase of Z, it shows a better performance in face of a refined estimation of the channel phase. On the other hand, the change of Z from 16 to 32, for $N_{it} = 50$, causes a little loss in performance for E_b/N_0 below 14 dB.

Another important point to analyze is the influence of the block interleaver in the system performance. The interleaver Π_e exchange the bits of a codeword that are sent to the M-DAPSK modulator. The change in parameters N_r and N_c results in different interleavers. Actually, a change in the interleaver corresponds to a permutation of columns of the matrix $\mathbf{H}$. An attempt to justify the influence of interleaving in LDPC *Bit-Interleaved Coded Modulation* (BICM) schemes is presented in [13].

Fig. 5 shows the system performance and the effect of two distinct configurations of interleaving matrix for LDPC code $(1008, 504)$.The modulation scheme is the same of the

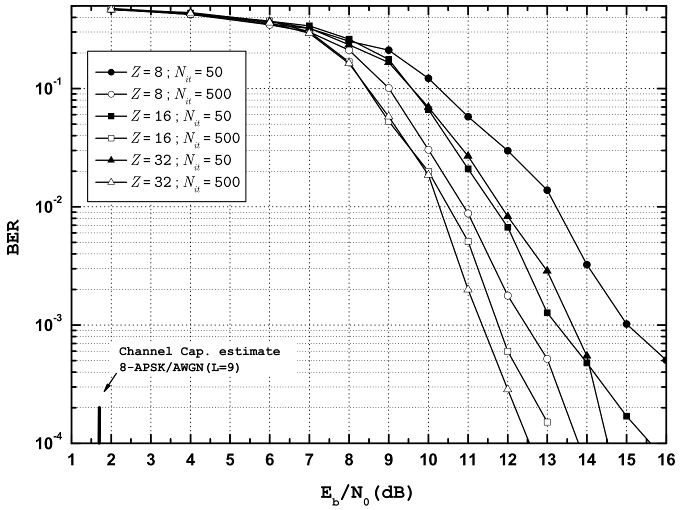

Fig. 4. System performance considering LDPC code $(96, 48)$. Phase quantization level: $Z = 8, 16, 32$. Number of iterations: $N_{it} = 50, 500$. $R_s = 1.33\ bits/symb$.

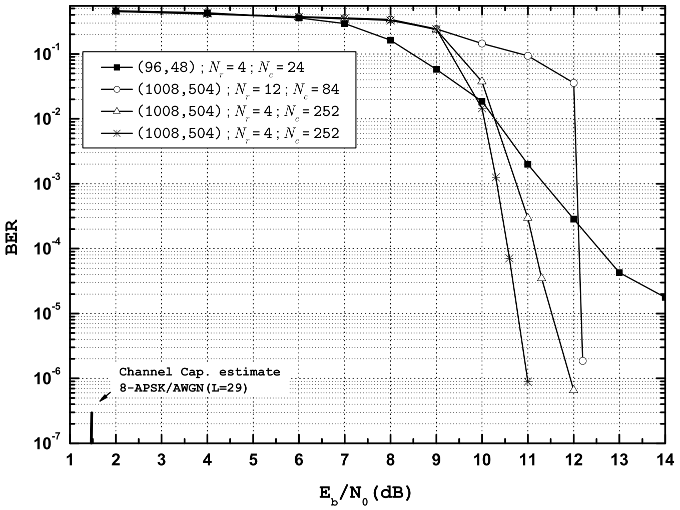

Fig. 5. System performance considering LDPC code $(1008, 504)$ for some configurations of interleaving matrix. $\circ, \triangle : Z = 16$, $N_{it} = 200$. $\blacksquare, * : Z = 32$, $N_{it} = 500$. $R_s = 1.45\ bits/symb$.

previous case, but the phase introduced by the channel is constant over a block of $L = 29$ symbols. We can see that the interleaver whose parameters value $N_r = 4$ and $N_c = 252$ provides a little coding gain when compared to first one, who has $N_r = 12$ and $N_c = 84$. For $BER = 10^{-3}$, the gain is approximately $1.3\ dB$. Besides that, increasing Z and N_{it} for this matrix configuration, we have an additional gain of $0.5\ dB$.

Although the system performance presented here is far away from the channel capacity estimative, we hope this scenario changes with the increasing of the block size of the LDPC codes. It is important to mention that the codes considered in this work have small block size when compared to the LDPC codes traditionally used in noncoherent channels. Franceschini et al. used LDPC codes with block size 12000 serially concatenated with differential *M-ary Phase Shift Keying* (M-PSK) modulations in [14] whereas Colavolpe et al. considered LDPC codes with block size 64800 in [8].

V. CONCLUSIONS

A coding scheme for iterative receivers using blockwise noncoherent M-APSK channel was investigated. To propose high spectral efficient systems for phase-noisy channels, we chose constellations of phase-amplitude modulated signals for the transmitter. In terms of receiver, the proposal of decoding algorithm was innovative since there is not a description of graph-based iterative receivers using differential encoding and non-binary modulations in the literature. The results showed that the system performance was satisfactory and it can become better with the use of longer codes. The phase quantization level, the number of iterations and the interleaving strategy are parameters to be investigated in the search for better results. At the same time, an analysis of the schedule of the sum-product algorithm can simplify the simulations for longer codes.

REFERENCES

[1] A.P.Worthen and W.E.Stark, "Unified Design of Iterative Receivers using Factor Graphs," *IEEE Trans. on Information Theory*, vol. 47, n. 2, pp. 843-849, Feb 2001.

[2] D.J.C.MacKay, Encyclopedia of Sparse Graph Codes, Home-page of the Inference Group, Cavendish Lab., Cambridge Univ. Available at: http://www.inference.cam.ac.uk/mackay/codes/data.html. Access: 01 Dec 2005.

[3] C.Heegard and S.B.Wicker, **Turbo Coding**. Kluwer Academic Publishers, 1999.

[4] R.Chen, R.Koetter, U.Madhow and D.Agrawal, "Joint Noncoherent Demodulation and Decoding for the Block Fading Channel: A Practical Framework for Approaching Shannon Capacity," *IEEE Trans. on Communications*, vol. 51, n. 10, pp. 1676-1689, Oct 2003.

[5] D.C.Cunha, Blockwise Noncoherent M-APSK Channel: Capacity and Coding Schemes for Iterative Receivers, PhD Thesis, State University of Campinas, Brazil, 2006.

[6] F.R.Kschischang, B.J.Frey and H.-A.Loeliger, "Factor Graphs and the Sum-Product Algorithm," *IEEE Trans. on Information Theory* , vol. 47, n. 2, pp. 498-519, Feb 2001.

[7] W.E.Ryan, **CRC Handbook for Coding and Signal Processing for Recording Systems**. CRC Press, 2004, cap. An Introduction to LDPC Codes.

[8] G.Colavolpe, A.Barbieri and G.Caire, "Algorithms for Iterative Decoding in the Presence of Strong Noise," *IEEE Journal on Selected Areas in Communications*, vol.23, n.9, pp. 1748-1757, Sep 2005.

[9] G.Ferrari, G.Colavolpe and R.Raheli, **Detection Algorithms for Wireless Communications with Applications to Wired and Storage Systems**. New York: J.Wiley, Hardcover, 2004.

[10] X.-Y.Hu, E.Eleftheriou, D.-M.Arnold and A.Dholakia, "Efficient Implementations of the Sum-Product Algorithm for Decoding LDPC Codes," In Proceedings of *IEEE GLOBECOM 2001*, vol. 2, pp. 1036-1036E, San Antonio, Texas, USA, Nov 2001.

[11] M.Peleg, S.Shamai(Shitz), "On the Capacity of the Blockwise Incoherent MPSK Channel," *IEEE Trans. on Communications*, vol.46, n.5, pp. 603-609, May 1998.

[12] D.C.Cunha and J.Portugheis, "Capacity of Blockwise Noncoherent M-APSK/AWGN Channels," *Electronics Letters*, vol.42, n.14, pp. 816-817, Jul 2006.

[13] R.D.Maddock and A.H.Banihashemi, "Reliability-Based Coded Modulation With Low-Density Parity-Check Codes," *IEEE Trans. on Communications*, vol.54, n.3, pp. 403-406, Mar 2006.

[14] 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, n.9, pp. 1758-1768, Set 2005.