

RESEARCH

Open Access



Continuous phase modulation with 1-bit quantization and oversampling using iterative detection and decoding

Rodrigo R. M. de Alencar^{*} , Lukas T. N. Landau and Rodrigo C. de Lamare

^{*}Correspondence:
alencar@cetuc.puc-rio.br
Centro de Estudos em
Telecomunicações, Pontifícia
Universidade Católica
do Rio de Janeiro, Rio de
Janeiros CEP 22453-900,
Brazil

Parts of this paper have been
published at Asilomar 2019

Abstract

A channel with continuous phase modulation and 1-bit ADC with oversampling is considered. Due to oversampling, higher-order modulations yield a higher achievable rate and this study presents methods to approach this with sophisticated channel coding. Iterative detection and decoding is considered, which exploits the soft information extracted from oversampled CPM sequences. Besides the strategy based on conventional channel coding techniques, a tailored bit mapping is proposed where two bits can be detected reliably at high SNR and only the third bit is affected by the uncertainties from the coarse quantization. With this, a convolutional code is only applied to the third bit. Our experiments with a turbo receiver show that the iterative detection and decoding is a promising approach for exploiting the additional information brought by oversampling. Moreover, it is shown that the proposed method based on the tailored bit mapping corresponds to a lower bit error rate in comparison with the case of conventional channel coding.

Keywords: 1-Bit quantization, Oversampling, ADC, Continuous phase modulation, BCJR, Iterative decoding

1 Introduction

The energy consumption of an analog-to-digital converter (ADC) scales exponentially with its resolution in amplitude [1]. One promising approach to overcome the limitation with respect to the ADC power consumption is 1-bit quantization, i.e., only the sign of the signal is known to the receiver. This coarse quantization is combined with oversampling at the receiver in order to compensate for the losses in terms of achievable rate, even for a noisy scenario [2], an argument that is in accordance with [3], where oversampling is used to achieve extra bits of resolution.

In [4], where a bandlimited channel is considered, a marginal benefit of 1-bit quantization and oversampling at the receiver in terms of achievable rate has been reported. In [5], a significant gain in terms of the achievable rate has been reported due to oversampling, by using a Zakai bandlimited process [6]. Both studies [4, 5] considered a noiseless channel. However, by considering the capacity per unit cost, it has been shown in [2] that 1-bit quantization and oversampling at the receiver can also

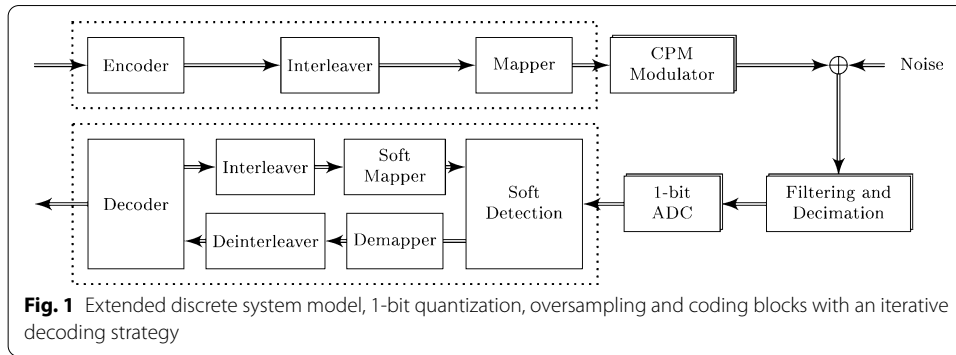
be beneficial in a noisy scenario, as it was shown later in [7] for bandlimited channels. The high signal-to-noise ratio (SNR) regime has been considered based on the concept of the generalized mutual information [8], which results in a minor benefit in terms of achievable rate. For a mid to high SNR scenario, an analytical evaluation of a lower bound on the mutual information rate is derived in [9] for a 1-bit quantized continuous-time bandlimited additive white Gaussian noise (AWGN) channel. Moreover, a practical approach for a 16-QAM system is presented in [10], where filter coefficients are optimized by maximizing the minimum distance to decision thresholds, a concept that is extended for multiple-input–single-output channels in [11]. With respect to multiple-input–multiple-output (MIMO) systems, 1-bit ADCs are attractive due to low-cost hardware and low power consumption per antenna, especially in massive MIMO systems [12], where it can also benefit from oversampling [13, 14]. In this context, channel estimation techniques with oversampling have been studied in [15]. Moreover, iterative detection and decoding for MIMO systems with 1-bit quantization has been studied in [16] to improve the overall bit error rate performance.

Oversampling and 1-bit quantization have been studied before in combination with continuous phase modulation (CPM) in [17, 18]. CPM signals are spectrally efficient, having smooth phase transitions and constant envelope [19, 20], which allows for energy-efficient power amplifiers. The information is implicitly conveyed in phase transitions, which makes the use of oversampling promising in the presence of coarse quantization at the receiver.

The achievable rate for channels with 1-bit quantization and oversampling at the receiver has been studied in [7, 21], where bandlimited and not strictly bandlimited signals are considered, respectively. In the two studies, it has been shown that the achievable rate can be lower-bounded by a truncation-based auxiliary channel law. The resulting channel has a finite state memory, where a sequence design is beneficial in terms of achievable rate. Similar ideas are considered for CPM with 1-bit quantization and oversampling at the receiver in [17], where it is shown how oversampling increases the achievable rate. Later, more practical approaches were proposed in [18], where the intermediate frequency and the waveform is considered in a geometrical analysis of the phase transitions.

In this study, we consider the design and analysis of CPM schemes with 1-bit quantization and oversampling at the receiver, employing channel coding in scenarios with higher modulation order. The present study extends our prior work on this subject presented in [22] by a detailed description of the iterative receive processing, the EXIT chart analysis and extensive number of numerical experiments.

For such cases, based on the achievable rates computed in [17], channel coding is essential for establishing reliable communications. In this context, this work extends the discrete system model for CPM signals received with 1-bit quantization and oversampling, presented in [17], for a sophisticated coding and decoding scheme. The proposed coded transmission system implies at the receiver an iterative detection and decoding strategy which consists of a BCJR algorithm for the computation of soft information and a second BCJR algorithm for the decoding process. Additionally, a sophisticated channel coding scheme, related to hierarchical modulation [23],



is proposed which exploits the special properties of the channel with 1-bit quantization. In summary, the main contributions of this paper are the following:

- Iterative detection and decoding scheme for CPM systems with 1-bit quantization and oversampling at the receiver;
- Novel phase-state-dependent bit mapping, which is designed for the 1-bit ADC problem;
- Channel coding scheme suited for the bit stream separation into sub-channels.

The rest of the paper is organized as follows. In Sect. 2 the discrete system model is presented, taking into account the ADC and channel coding with iterative detection and decoding. In Sect. 3, the discrete time description of the quantized CPM signal is presented. Section 4 discusses the soft detection process and the iterative decoding strategy. The proposed methods to enhance the performance in terms of BER are presented in Sect. 5. The numerical results are shown in Sects. 6 and 7 gives the conclusions.

Notation Bold symbols denote vectors, namely oversampling vectors, e.g., \mathbf{y}_k is a column vector with M entries, where k indicates the k th symbol in time or rather its corresponding time interval. Bold capital symbols denote matrices. Sequences are denoted with $\mathbf{x}^n = [x_1, \dots, x_n]^T$. Likewise, sequences of vectors are written as $\mathbf{y}^n = [\mathbf{y}_1^T, \dots, \mathbf{y}_n^T]^T$. A segment of a sequence is denoted as $\mathbf{x}_{k-L}^k = [x_{k-L}, \dots, x_k]^T$.

2 System model overview

The system model considered in this study is an extension of the discrete-time system model proposed in [17, 24]. In this context of CPM with 1-bit quantization and oversampling at the receiver, Fig. 1 illustrates the extension in terms of the additional coding blocks. The purpose of this extension is to design a system for reliable communication by considering sophisticated forward error correction.

For the proposed CPM system, the lower bound on the achievable rate presented in “Appendix 1” serves as the base information for choosing the code rate for channel coding. The result of the achievable rate computation in [17] shows that the scenarios with higher modulation order, e.g., $M_{\text{cpm}} = 8$, may require channel coding in order to provide communication with low probability of error, because the achievable rate at high SNR can be significantly lower than the input entropy. With this, a coding scheme must satisfy the following inequality

$$R \cdot \log_2(M_{\text{cpm}}) \leq I_{M_{\text{cpm}}} [\text{bpcu}], \quad (1)$$

where R denotes the code rate of the channel code and $I_{M_{\text{cpm}}}$ denotes the achievable rate conditioned on the corresponding CPM modulation scheme. Note that $\log_2(M_{\text{cpm}})$ is the maximum entropy rate of the input, which is an upper bound of the achievable rate.

On the transmit path, the channel encoder receives information bits and generates an encoded message adding redundant information. The encoded message is interleaved to protect the coded information against burst errors. Then, the interleaved bits are grouped according to the modulation order and mapped to CPM symbols. After that, a signal is generated by a CPM modulator and noise is generated at the receiver.

On the receive path, the signal is filtered and quantized by a 1-bit ADC. The quantized data are then processed by an iterative detection and decoding scheme. First, the binary samples are processed by a soft detection algorithm. Then, the soft information is converted to bit oriented log-likelihood ratios, which are deinterleaved subsequently. Finally, the soft information is given to the channel decoder, which returns extrinsic soft information to the detection algorithm via an interleaver and a soft mapper. In the next sections, the individual blocks are described in detail.

3 CPM with 1-bit quantization

The CPM signal in the passband with carrier frequency f_0 [19] is described by

$$s(t) = \text{Re} \left\{ \sqrt{\frac{2E_s}{T_s}} e^{j(2\pi f_0 t + \phi(t))} \right\}, \quad (2)$$

where $\text{Re}\{\cdot\}$ denotes the real part. The phase term is given by

$$\phi(t) = 2\pi h \sum_{k=0}^{\infty} \alpha_k f(t - kT_s) + \varphi_0, \quad (3)$$

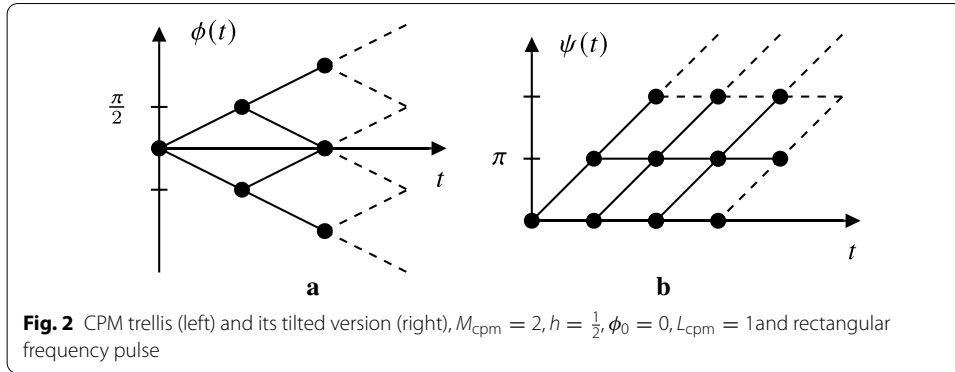
where T_s denotes the symbol duration, $h = \frac{K_{\text{cpm}}}{P_{\text{cpm}}}$ is the modulation index, $f(\cdot)$ is the phase response, φ_0 is a phase-offset, α_k represents the transmit symbols with symbol energy E_s . K_{cpm} and P_{cpm} must be relatively prime positive integers in order to obtain a finite number of phase states.

The phase response function shapes the phase transition between the phase states. It fulfills the following condition:

$$f(\tau) = \begin{cases} 0, & \text{if } \tau \leq 0, \\ \frac{1}{2}, & \text{if } \tau > L_{\text{cpm}} T_s, \end{cases}$$

where L_{cpm} is the depth of the memory in terms of transmit symbols. The phase response corresponds to the integration over the frequency pulse $g_f(\cdot)$, which is conventionally a rectangular pulse, a truncated and raised cosine pulse, or a Gaussian pulse. For an even modulation order M_{cpm} , the symbol alphabet can be described by $\alpha_k \in \{\pm 1, \pm 3, \dots, \pm(M_{\text{cpm}} - 1)\}$.

Generally, the corresponding phase trellis of (3) is time variant, which means that the possible phase states are time-dependent. Because of that, the number of wrapped absolute



phase states can be larger than M_{cpm} , e.g., when $M_{\text{cpm}} = 2$ and $h = \frac{1}{2}$, there are at least four trellis states in total and even more dependent on the memory of the channel. In order to reduce the complexity at the receiver, a time invariant trellis is constructed by tilting the trellis according to the decomposition approach in [25]. This is illustrated in Fig. 2. The tilt corresponds to an extension of the phase term given by (3) as

$$\psi(t) = \phi(t) + \pi h(M_{\text{cpm}} - 1) \frac{t}{T_s}, \quad (4)$$

where the second term on the right-hand side (RHS) corresponds to the tilt of the trellis. Taking the derivative of this tilt with respect to time and dividing by 2π results in a frequency offset given by $\Delta f = h(M_{\text{cpm}} - 1)/(2T_s)$.

A modified data sequence is obtained by replacing the symbol notation with the change of variable $x_k = (\alpha_k + M_{\text{cpm}} - 1)/2$, where the corresponding symbol alphabet can be described with $x_k \in \mathbb{X} = \{0, 1, \dots, M_{\text{cpm}} - 1\}$. According to [25], substituting $t = \tau + kT_s$ and $\alpha_k = 2x_k - M_{\text{cpm}} + 1$ in Eq. (3) leads to the tilted phase expression within one symbol duration

$$\begin{aligned} \psi(\tau + kT_s) = & 2\pi h \sum_{l=0}^{k-L_{\text{cpm}}} x_l + 4\pi h \sum_{l=0}^{L_{\text{cpm}}-1} x_{k-l} f(\tau + lT_s) \\ & + \pi h(M_{\text{cpm}} - 1) \frac{\tau}{T_s} \\ & - 2\pi h(M_{\text{cpm}} - 1) \sum_{l=0}^{L_{\text{cpm}}-1} f(\tau + lT_s) \\ & + (L_{\text{cpm}} - 1)(M_{\text{cpm}} - 1)\pi h + \phi_0, \quad 0 \leq \tau < T_s. \end{aligned} \quad (5)$$

where the time-dependent terms on the RHS depend only on the variable τ , which is well-defined along one symbol duration. Applying the $\text{mod} 2\pi$ operator to the first term on the right-hand side of (5) yields

$$\begin{aligned} \left(2\pi h \sum_{l=0}^{k-L_{\text{cpm}}} x_l \right) \bmod 2\pi &= \frac{2\pi}{P_{\text{cpm}}} \left(\left(K_{\text{cpm}} \sum_{l=0}^{k-L_{\text{cpm}}} x_l \right) \bmod P_{\text{cpm}} \right) \\ &= \frac{2\pi}{P_{\text{cpm}}} \beta_{k-L_{\text{cpm}}}, \end{aligned} \quad (6)$$

which introduces the absolute phase state β_k , i.e., it is related to the 2π -wrapped accumulated phase contributions of the input symbols that occur prior to the CPM memory. With this, the phase expression for one symbol duration can be fully described by the absolute phase state $\beta_{k-L_{\text{cpm}}}$ and the previous and the current transmit symbols $x_{k-L_{\text{cpm}}+1}^k$ given by $\tilde{s}_k = [\beta_{k-L_{\text{cpm}}}, x_{k-L_{\text{cpm}}+1}^k]$. It is important to highlight that \tilde{s}_k is the appropriate state description for the modeling of the signal at the intermediate frequency. To model the signal which has passed the bandpass filter at the intermediate frequency (IF) another state description, namely s_k , will be introduced later. For better integration with 1-bit quantization, $\varphi_0 = \pi h$ is used instead of $\varphi_0 = 0$ whenever it avoids phase states placed on the axis in the constellation diagram, which is the case in all the considered examples.

In the following, a discrete-time system model description is considered which implies that the CPM phase is represented in a vector notation. The corresponding tilted CPM phase $\psi(\tau + kT_s)$ for one symbol interval, i.e., $0 \leq \tau < T_s$, is then discretized into MD samples, which composes the vector denoted by

$$\boldsymbol{\psi}_k(\tilde{s}_k) = \left[\psi\left(\frac{T_s}{MD}(kMD + 1)\right), \psi\left(\frac{T_s}{MD}(kMD + 2)\right), \dots, \psi(T_s(k + 1)) \right]^T,$$

where M is the oversampling factor and D is a higher resolution multiplier.

The receive filter $g(t)$ has an impulse response of length T_g . In the discrete model for expressing a subsequence of $(N + 1)$ oversampling output symbols it is represented in a matrix form with \mathbf{G} , as a $MD(N + 1) \times MD(L_g + N + 1)$ Toeplitz matrix, as described as follows

$$\mathbf{G} = \begin{bmatrix} [\mathbf{g}^T] & 0 \dots & 0 & 0 \\ 0 & [\mathbf{g}^T] & 0 \dots & 0 & 0 \\ & \ddots & \ddots & \ddots & \\ 0 \dots & 0 & [\mathbf{g}^T] & 0 \end{bmatrix}, \quad (7)$$

where

$$\mathbf{g}^T = \sqrt{\frac{T_s}{MD}} \left[g(L_g T_s), g\left(\frac{T_s}{MD}(L_g MD - 1)\right), \dots, g\left(\frac{T_s}{MD}\right) \right],$$

and unit energy normalization is considered with $\int_{-\infty}^{\infty} |g(t)|^2 dt = 1$ and $\|\mathbf{g}\|_2^2 = 1$. A higher sampling grid in the waveform signal, in the noise generation and in the filtering is adopted to adequately model the aliasing effect. This receives filtering yields an increase in memory in the system by L_g symbols, where $(L_g - 1)T_s < T_g \leq L_g T_s$.

The filtered samples are decimated to the vector \mathbf{z}_{k-N}^k according to the oversampling factor M , by multiplication with the D -fold decimation matrix \mathbf{D} with dimensions $M(N+1) \times MD(N+1)$, described by

$$D_{i,j} = \begin{cases} 1 & \text{for } j = (i-1)D + 1, \\ 0 & \text{otherwise.} \end{cases} \quad (8)$$

Then, the result \mathbf{z}_{k-N}^k is 1-bit quantized to the vector \mathbf{y}_{k-N}^k . These operations can be represented by the following equations:

$$\begin{aligned} \mathbf{y}_{k-N}^k &= Q(\mathbf{z}_{k-N}^k) \\ &= Q\left(\mathbf{D} \mathbf{G} \left[\sqrt{\frac{E_s}{T_s}} \mathbf{e}^{\mathbf{y}_{k-N-L_g}^k} + \mathbf{n}_{k-N-L_g}^k \right]\right), \end{aligned} \quad (9)$$

where $Q(\cdot)$ denotes the quantization operator. The quantization of \mathbf{z}_k is described by $y_{k,m} = \text{sgn}(\text{Re}\{z_{k,m}\}) + j\text{sgn}(\text{Im}\{z_{k,m}\})$, where m denotes the oversampling index which runs from 1 to M and $y_{k,m} \in \{1+j, 1-j, -1+j, -1-j\}$. The vector $\mathbf{n}_{k-N-L_g}^k$ contains complex zero-mean white Gaussian noise samples with variance $\sigma_n^2 = N_0$.

4 Soft detection and decoding

The soft detection relies on the maximum a posteriori (MAP) decision metric for each bit, which corresponds to the a posteriori probability (APP) given the received sequence \mathbf{y}^n . For the considered system, the bit APPs can be approximately computed via a Bahl–Cocke–Jelinek–Raviv (BCJR) algorithm [26] based on an auxiliary channel law. Prior to computing those probabilities, the channel model for the system is required, i.e., the output of soft information by the detection algorithm depends on the set of channel output probabilities. This section covers the mentioned topics and describes the iterative decoding subsequently.

4.1 Auxiliary channel law and channel output probability

Depending on the receive filter, noise samples are correlated which then implies dependency on previous channel outputs, such that the channel law has the form $P(\mathbf{y}_k | \mathbf{y}^{k-1}, \mathbf{x}^n)$. In this case, the consideration of an auxiliary channel law $W(\cdot)$ is required, which reads as

$$\begin{aligned} W(\mathbf{y}_k | \mathbf{y}^{k-1}, \mathbf{x}^n) &= P(\mathbf{y}_k | \mathbf{y}_{k-N}^{k-1}, \mathbf{x}^n) \\ &= P(\mathbf{y}_k | \mathbf{y}_{k-N}^{k-1}, \beta_{k-L_{\text{cpm}}-L_g-N}, \mathbf{x}_{k-L_{\text{cpm}}-L_g-N+1}^k) \\ &= P(\mathbf{y}_k | \mathbf{y}_{k-N}^{k-1}, \beta_{k-L}, \mathbf{x}_{k-L+1}^k) \\ &= \frac{P(\mathbf{y}_{k-N}^k | \beta_{k-L}, \mathbf{x}_{k-L+1}^k)}{P(\mathbf{y}_{k-N}^{k-1} | \beta_{k-L}, \mathbf{x}_{k-L+1}^{k-1})}, \end{aligned} \quad (10)$$

where the dependency on N previous channel outputs is taken into account with $L = L_{\text{cpm}} + L_g + N$ being the total memory of the system. With this, the BCJR algorithm relies on an extended state representation denoted by

$$s_k = \begin{cases} [\beta_{k-L+1}, x_{k-L+2}^k], & \text{if } L > 1, \\ [\beta_k], & \text{if } L = 1. \end{cases} \quad (11)$$

Consequently, the notation of the auxiliary channel law can be written in terms of the state notation s_k

$$\begin{aligned} W(\mathbf{y}_k | \mathbf{y}^{k-1}, x^n) &= P(\mathbf{y}_k | \mathbf{y}_{k-N}^{k-1}, s_k, s_{k-1}) \\ &= \frac{P(\mathbf{y}_{k-N}^k | s_k, s_{k-1})}{P(\mathbf{y}_{k-N}^{k-1} | s_{k-1})}. \end{aligned} \quad (12)$$

These probabilities involve a multivariate Gaussian integration in terms of

$$P(\mathbf{y}_{k-N}^k | s_k, s_{k-1}) = \int_{\mathbf{z}_{k-N}^k \in \mathbb{Y}_{k-N}^k} p(\mathbf{z}_{k-N}^k | s_k, s_{k-1}) d\mathbf{z}_{k-N}^k, \quad (13)$$

where \mathbb{Y}_{k-N}^k represents the quantization interval which belongs to the channel output symbol \mathbf{y}_{k-N}^k , described in (9). The vector \mathbf{z}_{k-N}^k is a complex Gaussian random vector that describes the input of the ADC, with mean vector $\mathbf{m}_z = \mathbf{D}\mathbf{G} \left[\sqrt{E_s/T_s} \mathbf{e}^{\mathbf{y}_{k-N-Lg}^k} \right]$ and covariance matrix $\mathbf{K}_z = \sigma_n^2 \mathbf{D}\mathbf{G}\mathbf{G}^H \mathbf{D}^T$, with \mathbf{D} and \mathbf{G} as introduced before. In order to numerically evaluate (13) using an existing quasi-Monte Carlo integration algorithm, based on methods developed in [27] and implemented in the MATLAB-function `mvncdf()`, a real valued formulation is required. To achieve this, the conditional probability density function $p(\mathbf{z}_{k-N}^k | s_k, s_{k-1})$ is written as follows

$$\begin{aligned} p(\mathbf{z}_{k-N}^k | s_k, s_{k-1}) &= \frac{1}{\sqrt{(2\pi)^{2M(N+1)} |\mathbf{K}'_z|}} \\ &\times \exp \left(-\frac{1}{2} \left(\mathbf{z}_{k-N}^{k'} - \mathbf{m}'_z \right)^T \mathbf{K}'_z^{-1} \left(\mathbf{z}_{k-N}^{k'} - \mathbf{m}'_z \right) \right), \end{aligned} \quad (14)$$

where $|\cdot|$ denotes the determinant, $\mathbf{z}_{k-N}^{k'} = \left[\text{Re}\{\mathbf{z}_{k-N}^k\}^T, \text{Im}\{\mathbf{z}_{k-N}^k\}^T \right]^T$ and the mean vector \mathbf{m}'_z contains the real and imaginary components in a stacked fashion as given by

$$\mathbf{m}'_z = \begin{bmatrix} \text{Re}\{\mathbf{m}_z\} \\ \text{Im}\{\mathbf{m}_z\} \end{bmatrix} = \begin{bmatrix} \text{Re} \left\{ \mathbf{D}\mathbf{G} \left[\sqrt{\frac{E_s}{T_s}} \mathbf{e}^{\mathbf{y}_{k-Lg-N}^k} \right] \right\} \\ \text{Im} \left\{ \mathbf{D}\mathbf{G} \left[\sqrt{\frac{E_s}{T_s}} \mathbf{e}^{\mathbf{y}_{k-Lg-N}^k} \right] \right\} \end{bmatrix}. \quad (15)$$

Accordingly, the covariance matrix is denoted as

$$\begin{aligned} \mathbf{K}'_z &= \mathbf{E} \left\{ \begin{bmatrix} \mathbf{D} \text{Re}\{\mathbf{G}\} & -\mathbf{D} \text{Im}\{\mathbf{G}\} \\ \mathbf{D} \text{Im}\{\mathbf{G}\} & \mathbf{D} \text{Re}\{\mathbf{G}\} \end{bmatrix} \begin{bmatrix} \text{Re}\{\mathbf{n}_{k-Lg-N}^k\} \\ \text{Im}\{\mathbf{n}_{k-Lg-N}^k\} \end{bmatrix} \right. \\ &\quad \times \left. \begin{bmatrix} \text{Re}\{\mathbf{n}_{k-Lg-N}^k\} \\ \text{Im}\{\mathbf{n}_{k-Lg-N}^k\} \end{bmatrix}^T \begin{bmatrix} \mathbf{D} \text{Re}\{\mathbf{G}\} & -\mathbf{D} \text{Im}\{\mathbf{G}\} \\ \mathbf{D} \text{Im}\{\mathbf{G}\} & \mathbf{D} \text{Re}\{\mathbf{G}\} \end{bmatrix}^T \right\}. \end{aligned} \quad (16)$$

As detailed in [18], the number of evaluations n_{ev} of the multivariate integral in (13) required for the model respects the proportion described by

$$n_{ev} \propto 4^M M_{\text{cpm}}^L, \quad (17)$$

where 4^M is the number of all possible observed complex vectors \mathbf{y}_k and M_{cpm}^L is proportional to the number of all the possible state transitions. With this, the evaluation of all channel output probabilities becomes computationally expensive when the oversampling factor M , the modulation order M_{cpm} and the overall channel memory L are high valued.

4.2 BCJR algorithm based on the auxiliary channel law

With the purpose of evaluate the APPs for the bit sequence, the value $P(s_k, s_{k-1} | \mathbf{y}^n)$ must be determined. This can be achieved by normalizing the joint probability $P(s_k, s_{k-1}, \mathbf{y}^n)$, which can be decomposed into

$$\begin{aligned} P(s_k, s_{k-1}, \mathbf{y}^n) &= P(s_k, s_{k-1}, \mathbf{y}^{k-1}, \mathbf{y}_k, \mathbf{y}_{k+1}^n), \\ &= P(s_{k-1}, \mathbf{y}^{k-1}) \cdot P(s_k, \mathbf{y}_k | \mathbf{y}^{k-1}, s_{k-1}) \cdot P(\mathbf{y}_{k+1}^n | s_k, \mathbf{y}^k) \\ &\approx \underbrace{W(s_{k-1}, \mathbf{y}^{k-1})}_{f_{k-1}(s_{k-1})} \cdot \underbrace{W(s_k, \mathbf{y}_k | \mathbf{y}^{k-1}, s_{k-1})}_{\gamma_k(s_{k-1}, s_k)} \cdot \underbrace{W(\mathbf{y}_{k+1}^n | s_k, \mathbf{y}^k)}_{b_k(s_k)}. \end{aligned} \quad (18)$$

where the considered auxiliary channel law $W(\cdot)$ as introduced in (10) is applied. The factor $\gamma_k(s_{k-1}, s_k)$ can be rewritten as

$$\begin{aligned} \gamma_k(s_{k-1}, s_k) &= W(s_k, \mathbf{y}_k | \mathbf{y}^{k-1}, s_{k-1}) \\ &= P(\mathbf{y}_k | \mathbf{y}_{k-N}^{k-1}, s_k, s_{k-1}) \cdot P(s_k | s_{k-1}), \end{aligned} \quad (19)$$

which relates to the channel output probability and to the state transition probabilities. The factor $f_{k-1}(s_{k-1})$ is the forward probability which contains all the paths in the trellis that leads to the state s_{k-1} . The forward recursion used to compute $f_{k-1}(s_{k-1})$ is described by

$$\begin{aligned} f_k(s_k) &= W(s_k, \mathbf{y}^k) = \sum_{s_{k-1}} W(s_k, s_{k-1}, \mathbf{y}^k) \\ &= \sum_{s_{k-1}} W(s_k, \mathbf{y}_k | \mathbf{y}^{k-1}, s_{k-1}) \cdot W(s_{k-1}, \mathbf{y}^{k-1}) \\ &= \sum_{s_{k-1}} \gamma_k(s_{k-1}, s_k) \cdot f_{k-1}(s_{k-1}). \end{aligned} \quad (20)$$

Finally, $b_k(s_k)$ is the backward probability that contains all possible paths from state s_k to s_n . Similarly, $b_k(s_k)$ is computed using the backward recursion

$$\begin{aligned}
b_k(s_k) &= W(\mathbf{y}_{k+1}^n | s_k, \mathbf{y}^k) = \sum_{s_{k+1}} W(s_{k+1}, \mathbf{y}_{k+1}^n | s_k, \mathbf{y}^k) \\
&= \sum_{s_{k+1}} W(s_{k+1}, \mathbf{y}_{k+1} | \mathbf{y}^k, s_k) \cdot W(\mathbf{y}_{k+2}^n | s_k, s_{k+1}, \mathbf{y}^{k+1}) \\
&= \sum_{s_{k+1}} \gamma_k(s_k, s_{k+1}) \cdot b_{k+1}(s_{k+1}).
\end{aligned} \tag{21}$$

With this, an approximate value $P_{\text{aux}}(s_k, s_{k-1} | \mathbf{y}^n)$ of the probability $P(s_k, s_{k-1} | \mathbf{y}^n)$ can be computed by normalizing for $P(\mathbf{y}^n)$, equation

$$P_{\text{aux}}(s_k, s_{k-1} | \mathbf{y}^n) \cdot P(\mathbf{y}^n) = f_{k-1}(s_{k-1}) \cdot \gamma_k(s_{k-1}, s_k) \cdot b_k(s_k) \tag{22}$$

Note that for the recursions described in (20) and (21), initial values for $f_{k=0}(s_k)$ and $b_{k=n+1}(s_k)$ are required. While the BCJR algorithm can be carried out using the steps above, it is practical to employ a matrix form of the algorithm described in [28].

4.3 A posteriori probabilities and soft information

As illustrated in the previous section, with the channel output probability as input, the BCJR algorithm provides the probabilities $P_{\text{aux}}(s_{k-1}, s_k | \mathbf{y}^n)$ used for computing the bit APPs. Letting $\mathbf{d}^m = [d_1, \dots, d_m]^T$, where $m = n \cdot \log_2(M_{\text{cpm}})$, as the interleaved bit sequence that is mapped into CPM symbols, its bit APPs are described by the summation

$$P(d_q = d | \mathbf{y}^n) \Big|_{q=(k-1) \cdot \log_2(M_{\text{cpm}}) + i} = \sum_{\substack{\forall s_{k-1}, s_k \geq x_k \\ \text{such that} \\ \text{map}_k^{-1}(x_k, i) = d}} P_{\text{aux}}(s_k, s_{k-1} | \mathbf{y}^n), \tag{23}$$

where $d_q = \text{map}_k^{-1}(x_k, i)$ with bit index $q = (k-1) \cdot \log_2(M_{\text{cpm}}) + i$ denotes extraction of the bit d_q , which corresponds to the i^{th} bit (most significant bit first) of the k^{th} demapped symbol with $i \in \{1, 2, \dots, \log_2(M_{\text{cpm}})\}$. The posterior probabilities $P(d_q | \mathbf{y}^n)$ are the natural choice for the soft information $s(d_q)$ about the demapped bits. With binary random variables described by the probabilities $s(d_q = 0) = P(d_q = 0 | \mathbf{y}^n)$ and $s(d_q = 1) = P(d_q = 1 | \mathbf{y}^n)$, the use of log-likelihood ratios (LLR) given the received sequence \mathbf{y}^n is appropriate [29] and obtained with

$$L(d_q | \mathbf{y}^n) = \ln \frac{P(d_q = 0 | \mathbf{y}^n)}{P(d_q = 1 | \mathbf{y}^n)} = \ln \frac{s(d_q = 0)}{s(d_q = 1)}, \tag{24}$$

which can be decomposed, according to [29, 30], into an extrinsic and an a priori LLR described by

$$\begin{aligned}
L(d_q | \mathbf{y}^n) &= L_{\text{ext}}(d_q | \mathbf{y}^n) + \ln \frac{P(d_q = 0)}{P(d_q = 1)} \\
&= L_{\text{ext}}(d_q | \mathbf{y}^n) + L_a(d_q).
\end{aligned} \tag{25}$$

The extrinsic LLR $L_{\text{ext}}(d_q | \mathbf{y}^n)$ represents the information about d_q contained in \mathbf{y}^n and $P(d_j)$ for all $j \neq q$. The a priori LLR $L_a(d_q)$ describes the available a priori information about d_q . The extrinsic LLR sequence is deinterleaved, according to the permutation

adopted. The resultant sequence represents the detected soft information used as input for the channel decoder as represented in Fig. 1.

4.4 Iterative detection and decoding

This work uses an iterative detection and decoding procedure that consists of the exchange of soft information between the detector and the channel decoder. Initially, the state transition probabilities are considered to be uniformly distributed, i.e., $P(s_k | s_{k-1}) = 1/M_{\text{cpm}}$. Such assumption is suboptimal, but it is possible to take into account that state transitions can have different probabilities by feeding back updated extrinsic soft information about the code bits.

The iterative detection and decoding process relies on the feedback of soft information from the decoder to adjust the transition probabilities of the soft detector, which becomes aware of the underlying code. Let $c^m = [c_1, \dots, c_m]^T$ the code bit sequence that represents the encoded message. The soft information $s(c_q)$, with $q \in \{1, \dots, m\}$, is the input of the channel decoder, which is able to compute an update version $s'(c_q)$ of this soft information

$$s'(c_q) = P(c_q | s(c_1), s(c_2), \dots, s(c_m)). \quad (26)$$

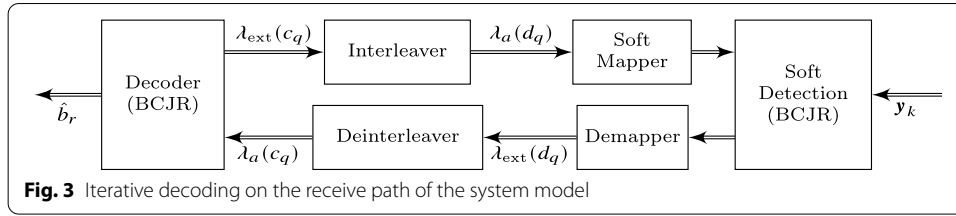
The redundancy introduced during the encoding process certifies that the reliability of $s'(c_q)$ is usually improved by the channel decoder when compared to $s(c_q)$. This soft information can be interleaved into the sequence corresponded to $s'(d_q)$, which is incorporated to the soft detector that uses this knowledge acquired from the channel decoder to recompute $s(d_q)$. This message passing algorithm is done iteratively with soft information exchange between the detection and decoding steps. With soft information exchange, the BER performance can be improved, but for approaching the optimal results, the transfer of the extrinsic information

$$s_{\text{ext}}(c_q) = P(c_q | s(c_1), \dots, s(c_{q-1}), s(c_{q+1}), \dots, s(c_m)), \quad (27)$$

from one iteration to the next is necessary, which is considered as crucial [29]. This extrinsic soft information characterizes the information about c_q contained in $s(c_1), \dots, s(c_m)$ except $s(c_q)$. The idea of passing extrinsic soft information between the receive algorithms was first proposed in [30] for decoding turbo codes and has been applied to coded data transmission over channels with intersymbol interference (ISI) [28, 31], where it is called turbo equalization. In practical terms, it is often more convenient to replace the two probabilities $s(d_q = 0)$ and $s(d_q = 1)$ by the LLRs in (24) and in (25) [29]

$$\begin{aligned} \lambda(d_q) &= L(d_q | y^n) = \ln \frac{s(d_q = 0)}{s(d_q = 1)}, \\ &= \lambda_{\text{ext}}(d_q) + \lambda_a(d_q), \end{aligned} \quad (28)$$

where $\lambda_{\text{ext}}(d_q)$ and $\lambda_a(d_q)$ are the extrinsic and the a priori LLRs, respectively. Accordingly, instead of the extrinsic soft information in the feedback path $s_{\text{ext}}(c_q)$, the extrinsic LLR is considered



$$\lambda_{\text{ext}}(c_q) = \ln \frac{\mathfrak{s}_{\text{ext}}(c_q = 0)}{\mathfrak{s}_{\text{ext}}(c_q = 1)} = \ln \frac{P(c_q = 0 | \mathfrak{s}(c_1), \dots, \mathfrak{s}(c_{q-1}), \mathfrak{s}(c_{q+1}), \dots, \mathfrak{s}(c_m))}{P(c_q = 1 | \mathfrak{s}(c_1), \dots, \mathfrak{s}(c_{q-1}), \mathfrak{s}(c_{q+1}), \dots, \mathfrak{s}(c_m))}.$$

A decomposition similar to (25) can be applied with $\lambda(c_q)$

$$\begin{aligned} \lambda_{\text{ext}}(c_q) &= \ln \frac{P(c_q = 0 | \mathfrak{s}(c_1), \dots, \mathfrak{s}(c_m))}{P(c_q = 1 | \mathfrak{s}(c_1), \dots, \mathfrak{s}(c_m))} - \ln \frac{\mathfrak{s}(c_q = 0)}{\mathfrak{s}(c_q = 1)}, \\ &= \lambda(c_q) - \ln \frac{\mathfrak{s}(c_q = 0)}{\mathfrak{s}(c_q = 1)}, \\ &= \lambda(c_q) - \lambda_a(c_q). \end{aligned} \quad (29)$$

This extrinsic information, can be computed during the channel decoding process by means of employing a second BCJR algorithm [28]. It can then be interleaved into new values for the a priori soft information $\lambda_a(d_q)$ and soft mapped to state transition probabilities $P(s_k | s_{k-1})$. This soft mapping is achieved by bringing the log-likelihood representation back to a probability description [28], which is given by

$$\begin{aligned} P(s_k | s_{k-1}) &= P(x_k) \\ &= \prod_{i=1}^{\log_2(M_{\text{cpm}})} \frac{\exp(-b'_i \cdot \lambda_a(d_q))}{1 + \exp(-\lambda_a(d_q))} \Big|_{q=(k-1) \cdot \log_2(M_{\text{cpm}}) + i}, \\ x_k &= \text{map}([b'_1, \dots, b'_{\log_2(M_{\text{cpm}})}]), \end{aligned} \quad (30)$$

where x_k is the input symbol that produce the state transition from the state s_{k-1} to s_k , $[b'_1, \dots, b'_{\log_2(M_{\text{cpm}})}]$ is the bit sequence which such symbol is mapped to, and $\lambda_a(d_q)$ represents the interleaved extrinsic soft information fed back by the channel decoder of the bit d_q . The probabilities $P(s_k | s_{k-1})$ can be used to perform the soft detection step again by updating the transition probabilities in the BCJR algorithm, detailed in Sect. 4.2.

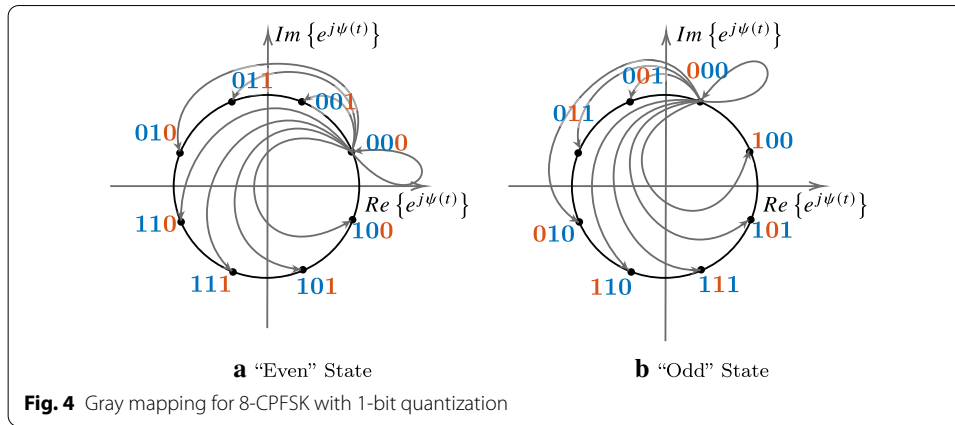
In the final step of the channel decoder, i.e., after the execution of the required number of iterations, the soft information on the information bits is computed

$$\lambda(b_r) = \ln \frac{P(b_r = 0 | \mathfrak{s}(c_1), \dots, \mathfrak{s}(c_m))}{P(b_r = 1 | \mathfrak{s}(c_1), \dots, \mathfrak{s}(c_m))}, \quad \forall r \in \{1, \dots, m \cdot R\}, \quad (31)$$

where R is the used code rate. At last, the estimation of the original information bits is evaluated based on the sign of the corresponding soft information $\lambda(b_r)$

$$\hat{b}_r = \begin{cases} 1, & \text{if } \lambda(b_r) < 0, \\ 0, & \text{if } \lambda(b_r) \geq 0, \end{cases} \quad (32)$$

The iterative decoding steps are illustrated in Fig. 3 and described in Table 1, where the soft information exchange between detector and decoder is represented with the



considered LLRs. The soft decoding step performed by the channel decoder is dependent of the used code. For completeness and illustration purposes, “Appendix 2” describes the BCJR-based decoding algorithm for a simple convolutional code.

5 Modified sub-channel coding

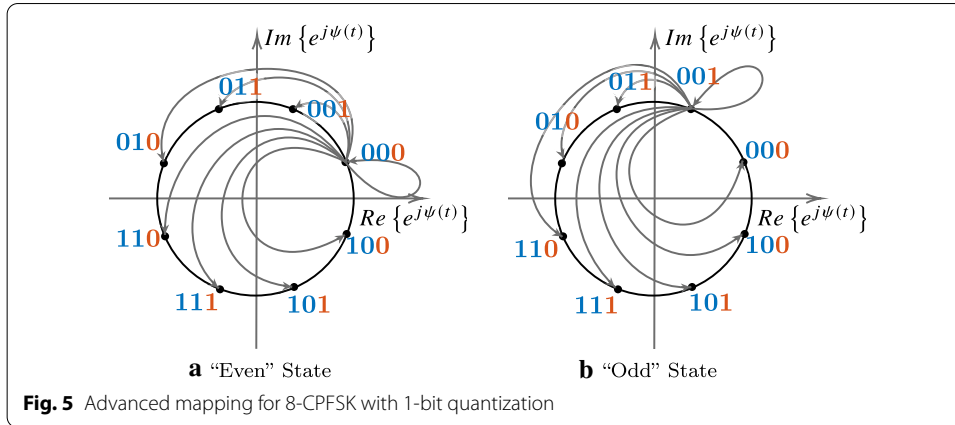
The extra amount of information provided by the oversampling is often not enough for the system to provide reliable communication, which motivated the proposition of the CPM system in Fig. 1 with additional coding blocks. With the purpose of further improving the performance of the proposed system, this section covers a different coding strategy and an alternative bit mapping scheme illustrated for the example of $M_{\text{cpm}} = 8$.

5.1 Bit mapping

For the considered CPM waveform with $L_{\text{cpm}} = 1$, the 1-bit quantization of the in-phase and quadrature components leads to a four-level phase decision, which grants two bits of information when the sample at time $\tau = T_s$ is observed in a noise-free scenario. This is the reason why, when noise is considered, the computed achievable rate for $M_{\text{cpm}} = 4$ reaches 2 bpcu in a high SNR regime. For the case with $M_{\text{cpm}} = 8$, more than two bits per channel use can be achieved with oversampling, which allows for the extraction of more information along the phase transition between the phase states. This idea is the key to understanding the motivation to study bit mapping alternatives, and how the available information at the receiver is distributed for the bits. In summary, this subsection presents two mapping strategies and how they deal with bit allocation.

Figure 4 shows the CPM tilted phase constellation on how bit sets are associated to the phase transition when using the Gray mapping, given an “even” or an “odd” initial state, whose parity is defined by the parity of the absolute phase state described in (6). The symbols are distinguished by on how much the phase is increased for each possible input.

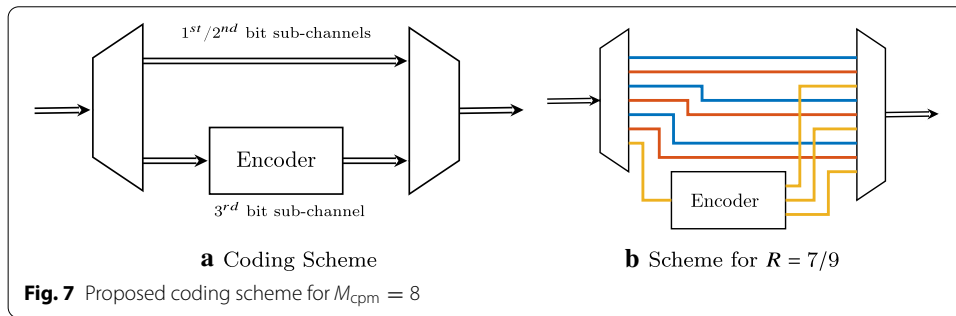
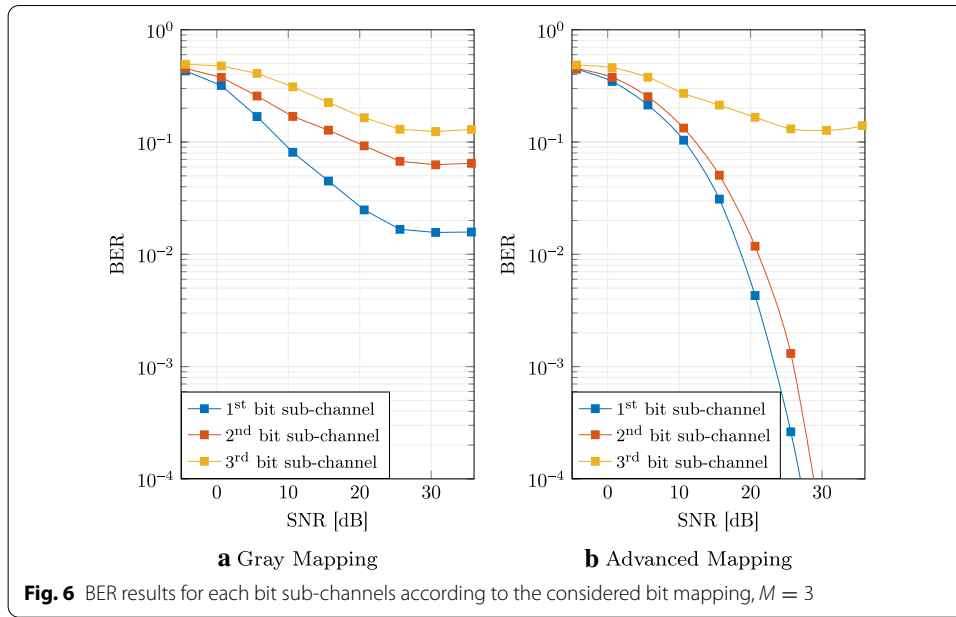
The established Gray mapping scheme implies well-known benefits for conventional communication systems. However, in the following, it is proposed to modify the Gray coding scheme in order to enable the exploitation of the properties of the CPM system with 1-bit quantization and oversampling at the receiver. Similarly to the Gray mapping, Fig. 5 illustrates the CPM tilted phase transitions with the corresponding bits

**Table 1** Iterative detection and decoding algorithm steps**Iterative detection and decoding**

- (1) *Initialization*: Set $P(s_k|s_{k-1}) = 1/M_{\text{cpm}}$ for every possible state transition, to evaluate the transition probabilities $\gamma_k(s_{k-1}, s_k)$ in (19)
- (2) *Soft detection and demapping*: Perform the BCJR algorithm described in Sect. 4.2, compute the a posteriori probabilities in (23) based on the adopted bit mapping and evaluate the extrinsic LLR sequence corresponding to $\lambda_{\text{ext}}(d_q)$ for all $q \in \{1, \dots, m\}$
- (3) *Deinterleaving*: Permute the LLR sequence $[\lambda_{\text{ext}}(d_1), \dots, \lambda_{\text{ext}}(d_m)]$ into $[\lambda_d(c_1), \dots, \lambda_d(c_m)]$
- (4) *Soft decoding*: Use the SISO decoder to compute the soft information $\lambda(b_r)$ and feedback the extrinsic soft information $\lambda_{\text{ext}}(c_q)$ for all $q \in \{1, \dots, m\}$
- (5) *Termination*: verify the stop condition, i.e., whether the number of iterations has been achieved. If so, the bit estimations \hat{b}_r are computed for all $r \in \{1, \dots, m \cdot R\}$, vide (32)
- (6) *Interleaving*: Permute the extrinsic soft information sequence $[\lambda_{\text{ext}}(c_1), \dots, \lambda_{\text{ext}}(c_m)]$ into $[\lambda_d(d_1), \dots, \lambda_d(d_m)]$
- (7) *Soft mapping*: compute new values of $P(s_k|s_{k-1})$ based on the soft mapping in (30) and update values for $\gamma_k(s_{k-1}, s_k)$
- (8) Go to (2)

for the novel mapping scheme, termed advanced mapping. The proposed modification allows that the information conveyed in the orthant of the received signal can be readily extracted. The additional information that is brought by the temporal oversampling is then conveyed in the third bit of the CPM symbol in case of $M_{\text{cpm}} = 8$. For the interpretation of the novel approach, one can consider that each bit of the CPM symbol corresponds to a separate binary sub-channel. Accordingly, the illustrated case with $M_{\text{cpm}} = 8$ can be understood as a system that consists of three different binary sub-channels. Considering the advanced mapping strategy, two sub-channels can each yield up to 1 bit per channel use and the third sub-channel yields a lower achievable rate which depends on the oversampling factor.

The Gray mapping shuffles the uncertainties brought by the coarse quantization for every bit sub-channel, whereas the advanced mapping aims to concentrate the uncertainty on the third bit, by doing a circular shift on the Gray mapping when phase transition happens from a phase state of "odd" order. Figure 6b illustrates what has been discussed by showing increased BER performance on the first two bits of the mapping in comparison with Fig. 6a.



5.2 Sub-channel coding scheme

The coding scheme which we refer to as the conventional approach consists of the system model illustrated in Fig. 1 that contemplates the use of channel coding for forward error correction (FEC), interleavers to protect the system against burst errors and an iterative decoding procedure as part of a sophisticated channel decoding approach. However, no regard toward bit sub-channel performance, presented in Fig. 6, is taken into account. With the proposed advanced mapping, it is known a priori on the transmitter side which bits in terms of binary sub-channels require protection by forward error correction. This circumstance can be exploited by sophisticated sub-channel coding. In this context, it is proposed a modified coding scheme that has the same structure of the conventional one, but applies different code rates for the bit sub-channels separately, i.e., in transmission, bit sub-channels streams that are more sensitive to the coarse quantization, noise or channel impairments would be protected by a stronger channel code.

For the case study, no coding is applied to the first two bit sub-channels whereas a strong convolutional code is applied to the third sub-channel. This is illustrated in Fig. 7a, where in (b) a code rate of $1/3$ is applied to the third sub-channel, which corresponds to an overall code rate of $R = 7/9$.

6 Results and discussion

All the computations rely on $M_{\text{cpm}} = 8$ and modulation index $h = \frac{1}{M_{\text{cpm}}}$. The used frequency pulse is the 1REC [19]. A suboptimal bandpass filter is considered with

$$g(t) = \sqrt{\frac{1}{T_g}} \text{rect}\left(\frac{t - T_s/2}{T_g}\right) \cdot e^{j2\pi \Delta f(t - T_s/2)}, \quad (33)$$

where $T_g = \frac{1}{2} T_s$. Such filter is similar to the integrate and dump receiver considered in [32], but with its frequency response centered in low-IF. Note that the common receiver based on a matched filter bank is hardware demanding and not compatible with the considered 1-bit approach. The extraction of the soft information is done based on an auxiliary channel law $W(\cdot)$ with the parameter $N = 0$, cf. (10). The SNR is given by the ratio between the transmit power and the product of the noise power spectral density N_0 and the two-sided 90% power containment bandwidth $B_{90\%}$

$$\text{SNR} = \frac{\lim_{T \rightarrow \infty} \frac{1}{T} \int_T |x(t)|^2 dt}{N_0 B_{90\%}} = \frac{E_s}{N_0} (T_s B_{90\%})^{-1}, \quad (34)$$

where $B_{90\%}$ denotes the 90% power containment bandwidth. The considered SNR definition can be interpreted as the average energy per *Nyquist interval* divided by the noise power density.

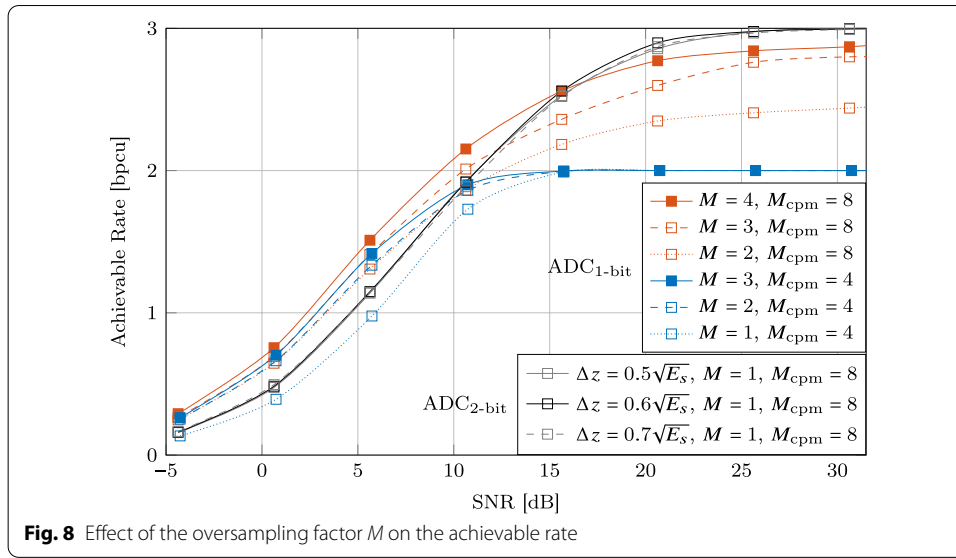
This work uses convolutional codes as the simplest non-trivial example for channel coding. Convolutional codes are characterized by their constraint length and its generator polynomials [33]. Channel codes with larger constraint length and longer blocks have better performance in general, but it requires more computation resources on the decoder.

High rate convolutional codes, e.g., 3/4, can be implemented using the puncturing technique to create any desired code rate from a basic low-rate code [34]. This is used to adapt the code rate while keeping a low-complexity decoder. Puncturing can be applied to modify the system throughput and robustness level by changing the puncturing pattern, which describes the bits that are propagated and discarded. The decoder must take this pattern into account to compute metrics and survivor paths. Puncturing is performed using the puncturing patterns described in Table 2

A block of information bits is randomly generated and forwarded to the encoder according to the discrete system model described in Fig. 1. S-Random interleavers are used such that they are generated for each analyzed block.

6.1 Achievable rate

The lower bound on the achievable rate is computed the way it is presented in “Appendix 1.” Figure 8 shows how oversampling increases the achievable rate due to the extra information provided during the symbol transitions. With $M_{\text{cpm}} = 4$ the achievable rate reaches $\log_2(M_{\text{cpm}}) = 2$ [bpcu] for higher values of SNR. For lower SNR values, when the achievable rate is below 2 [bpcu], consideration of channel coding is required for establishing reliable communication. For cases with higher modulation order, such as $M_{\text{cpm}} = 8$, the achievable rate of a CPM system with 1-bit quantization in general does



not reach the source entropy rate of $\log_2(M_{\text{cpm}}) = 3$ [bpcu] independent of the SNR. With this, the development of coding techniques is required, in order to design a system capable to provide reliable communication. In addition, the proposed system is compared with the same CPM system which is equipped with a uniform 2-bit quantizer with thresholds $\{-\Delta z, 0, \Delta z\}$ at the receiver for real and imaginary part that operates at symbol rate sampling rate.¹ Considering that the 2-bit ADC is implemented as a Flash ADC, it consists of three comparators which implies three comparator operations per symbol duration, which is then equivalent to the 1-bit oversampling with $M = 3$. While the performance in terms of achievable rate is comparable, the system that uses 1-bit ADCs does not require automatic gain control.

6.2 EXIT chart analysis

In this subsection, the iterative detection and decoding process is illustrated based on extrinsic information transfer (EXIT) charts [35]. The mutual information between the bit \mathcal{D} and its respective a priori LLR Λ as random processes reads as

$$\begin{aligned} I(\mathcal{D}; \Lambda) &= H(\mathcal{D}) - H(\mathcal{D}|\Lambda) \\ &= 1 - \int_{-\infty}^{\infty} P(\Lambda) \sum_{\mathcal{D} \in \{0,1\}} P(\mathcal{D}|\Lambda) \log_2 \frac{1}{P(\mathcal{D}|\Lambda)} d\Lambda, \end{aligned} \quad (35)$$

where the conditional probabilities are according to the LLR definition given by

$$\begin{aligned} P(\mathcal{D} = 0|\Lambda = \lambda) &= \frac{e^{\lambda/2}}{e^{-\lambda/2} + e^{\lambda/2}}, \\ P(\mathcal{D} = 1|\Lambda = \lambda) &= \frac{e^{-\lambda/2}}{e^{-\lambda/2} + e^{\lambda/2}}. \end{aligned} \quad (36)$$

¹ Note that symbol rate sampling corresponds to subsampling because the considered CPM signals have a larger bandwidth than $\frac{1}{T_s}$.

Assuming ergodicity for the LLR process Λ , the expectation in Eq. (35) can be replaced by a time average. Thus, the mutual information can be computed from a large number \mathcal{N} of generated LLRs as follows

$$\begin{aligned} I(\mathcal{D}; \Lambda) &= 1 - E \left\{ \sum_{\mathcal{D} \in \{0,1\}} P(\mathcal{D}|\Lambda) \log_2 \frac{1}{P(\mathcal{D}|\Lambda)} \right\}, \\ &\approx 1 - \frac{1}{\mathcal{N}} \sum_{n=1}^{\mathcal{N}} \sum_{d \in \{0,1\}} P(d|\lambda_n) \log_2 \frac{1}{P(d|\lambda_n)}, \end{aligned} \quad (37)$$

which is numerically evaluated by the consideration of large number of channel realizations. The LLR sequences corresponding to $\lambda_a(d_q)$, $\lambda_{\text{ext}}(d_q)$, $\lambda_a(c_q)$ and $\lambda_{\text{ext}}(c_q)$ for all $q \in \{1, \dots, m\}$ are related to the mutual information I_a^d , I_{ext}^d , I_a^c and I_{ext}^c , respectively, which are computed with Eq. (37) at every iteration step. Those LLRs are considered a priori information updated by the BCJR detector and decoder, as displayed in Fig. 3. Once interleaving and deinterleaving are neutral operations for the statistical description, the assertions $I_a^d = I_{\text{ext}}^c$ and $I_a^c = I_{\text{ext}}^d$ are true. According to [36], and [37], those a priori LLRs can be approximated as independent and identically distributed Gaussian random variables. This is done to simplify the computation of the transfer information functions T_d and T_c of the detector and decoder, respectively, which are described by

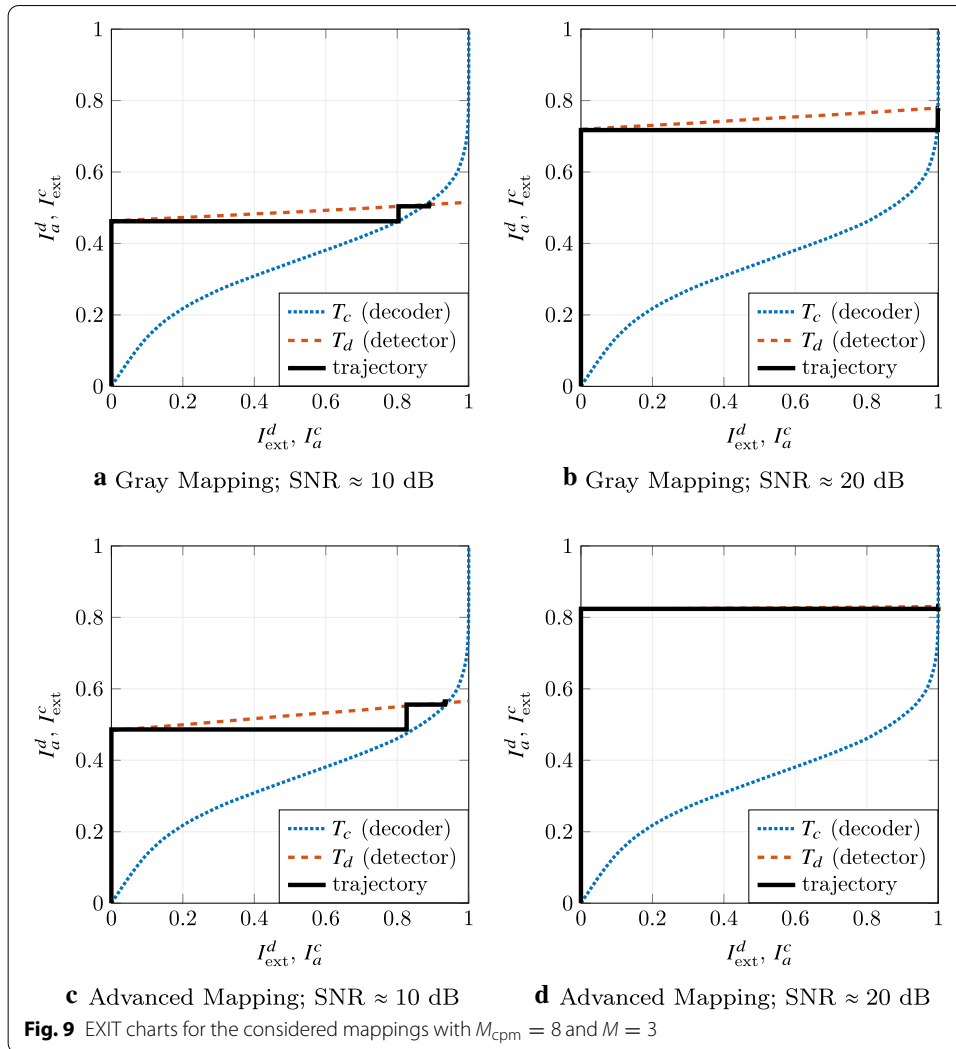
$$\begin{aligned} I_{\text{ext}}^d &= T_d(I_a^d), \\ I_{\text{ext}}^c &= T_c(I_a^c), \end{aligned}$$

where for each input mutual information, random Gaussian distributed a priori LLRs can be generated with a simulation tool to evaluate the detector and the decoder separately by measuring the mutual information between the output extrinsic LLRs and the encoded bit sequence. This procedure is done to record the mutual information evolution with the transfer functions.

Figure 9 shows examples of EXIT chart analysis for the proposed iterative detection and decoding. Various cases have been evaluated where it was observed that iterative processing yields a benefit especially for cases with low SNR. Figure 9a, c illustrates examples for this case. For cases at high SNR, it has been observed that in many cases it is not required to consider more than 2 iterations, as illustrated in Fig. 9b. Finally, it has been observed that considering advanced mapping instead of the Gray mapping significantly changes the exit chart of the detector, as shown in Fig. 9d. In such a case, the detection and decoding process has quasi converged after the first iteration.

6.3 Bit error rate

Figure 10 shows the BER for a system with Gray mapping and modulation order of $M_{\text{cpm}} = 8$ with different oversampling factors. The results show that oversampling can significantly improve the BER performance. Moreover, the results confirm that there is only marginally gain from considering more than two iterations for the iterative receive processing, which is in line with the EXIT chart analysis. Moreover, the BER of the corresponding system with symbol rate sampling and 2-bit ADC is shown for comparison. The BER performance indicate that a higher resolution in time is beneficial at low and

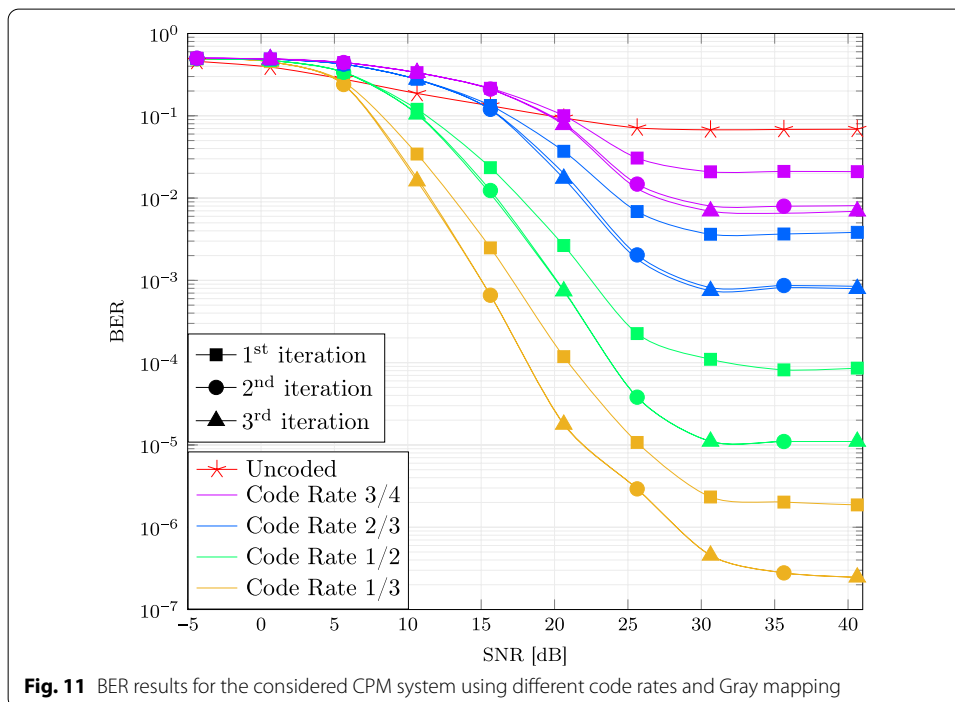
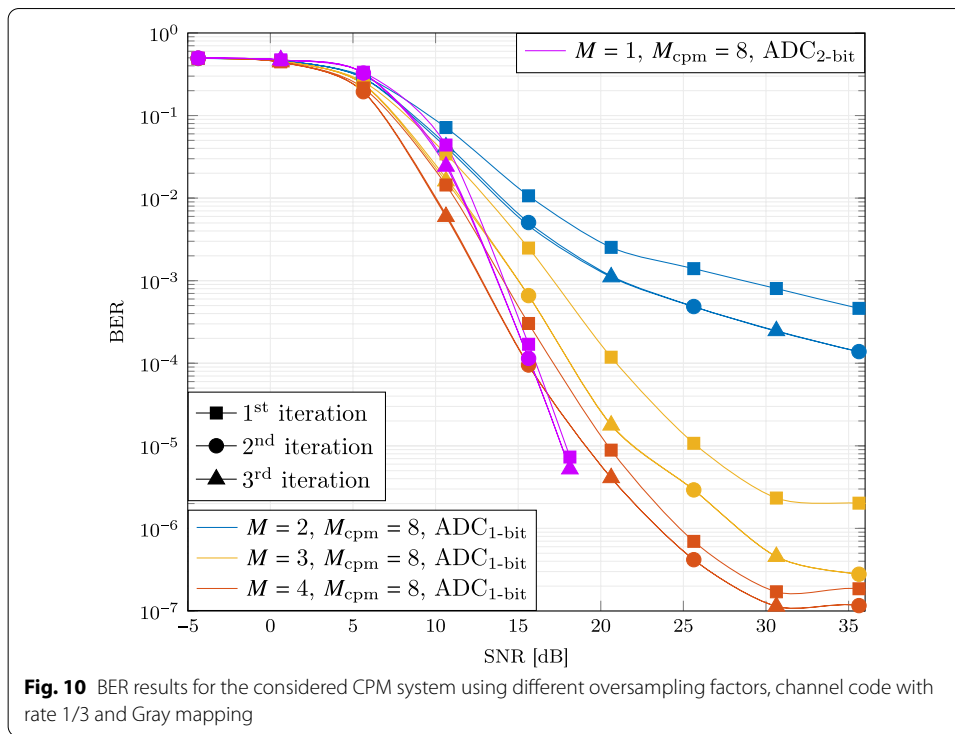


medium SNR and that a higher resolution in amplitude is beneficial in the high SNR regime.

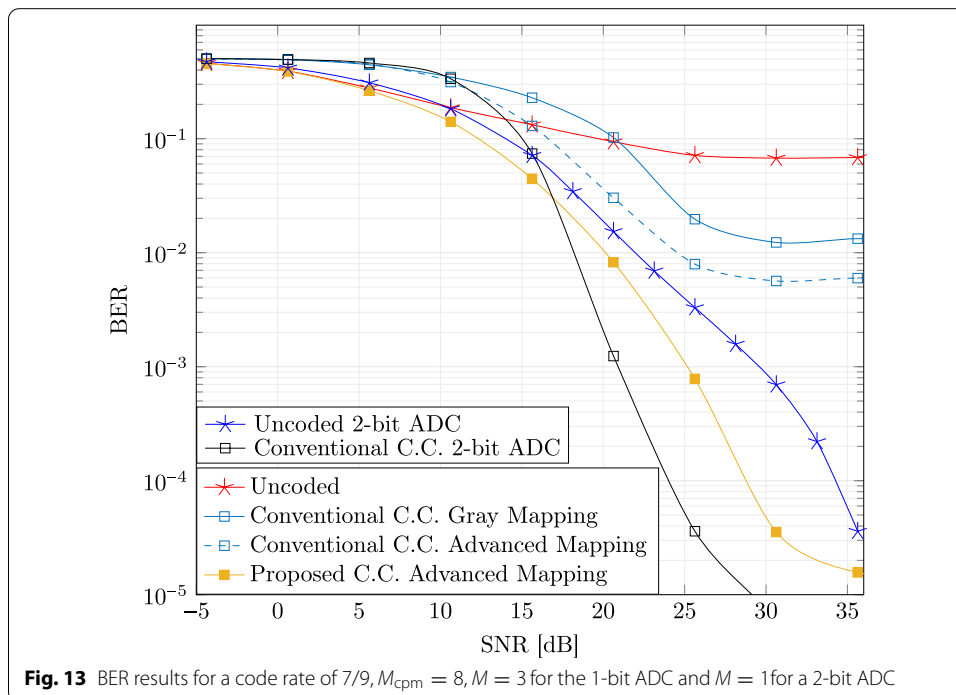
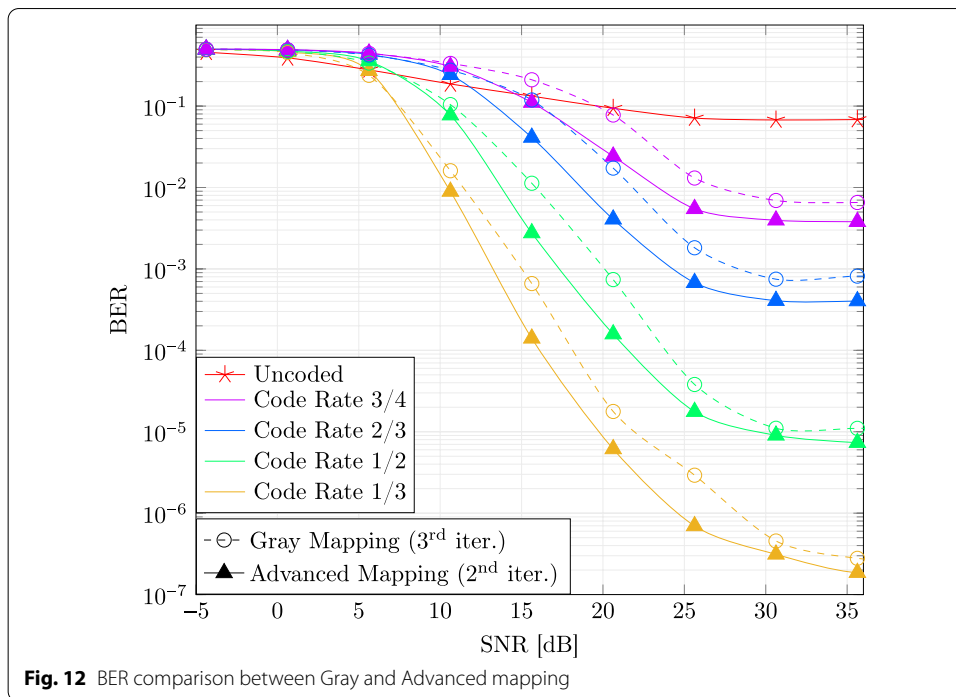
For the case with modulation order of $M_{\text{cpm}} = 8$ and oversampling factor of $M = 3$, channel coding is applied according to the extended discrete model shown in Fig. 1. Different code rates have been considered which is summarized in Table 2. The results presented in Fig. 11 confirm that iterative decoding scheme can improve the BER performance.

Additionally, the Gray mapping is compared against the novel advanced bit mapping. Figure 12 illustrates the BER simulation results, which confirm that system designs using advanced mapping outperform system designs using Gray mapping in terms of BER. For systems with Gray mapping, three iterations are considered and for systems with advanced mapping only two, because no further performance gain was observed by increasing the iterations for the latter.

Finally, in Fig. 13, the BER results for the proposed coding scheme are presented. The conventional coding scheme uses a convolutional code with rate 7/9, generator



polynomial (57) and puncturing pattern 11|01|01|10|10|01|11. In contrast, the proposed coding scheme uses a convolutional code with rate 1/3 for the third bit sub-channel as shown in Fig. 7b. In this last scenario, the use of the proposed coding



scheme brings a consistent performance gain from low to high SNR levels. The performance gain can be explained by the fact that the proposed method exploits apriori knowledge about the reliability of the binary sub-channels.

Table 2 Puncturing patterns for codes with constraint length $K_{cc} = 3$, generator polynomial in octal form

Code rate	Generator	Puncturing pattern
1/3	(5 7 7)	1 1 1
1/2	(5 7)	1 1
2/3	(5 7)	1 1 0 1
3/4	(5 7)	1 1 0 1 1 0

The proposed schemes are compared with the corresponding system with resolution in amplitude with 2-bit quantization. Such an ADC, when realized as flash ADC, corresponds to the same number of comparator operations per time interval. While the system with 1-bit quantization requires a higher SNR for achieving a low BER, it has practical benefits such as the relaxed requirements on the automatic gain control and the low noise amplifier.

7 Conclusions

Iterative detection and decoding applied to a CPM system with 1-bit quantization and over-sampling at the receiver has been studied. Different code rates have been considered, and it turns out that channel coding is beneficial in all the cases. Additional performance gain can be achieved by using the proposed tailored bit mapping strategy in combination with a coding scheme that considers different binary sub-channels separately.

Abbreviations

ADC: Analog-to-digital converter; APP: A posteriori probability; AWGN: Additive white Gaussian noise; BER: Bit error rate; BCJR: Bahl–Cocke–Jelinek–Raviv; bpcu: Bits per channel use; CPM: Continuous phase modulation; EXIT: Extrinsic information transfer; IF: Intermediate frequency; ISI: Intersymbol interference; LLR: Log-likelihood ratio; MAP: Maximum a posteriori; RHS: Right-hand side; SNR: Signal-to-noise ratio.

Acknowledgements

Not applicable.

Authors' contributions

All authors contributed to the conception and design of the study. RA drafted the manuscript and did the simulation work. All authors contributed to the interpretation of the results. LL and RdL reviewed and edited the manuscript. All authors read and approved the final manuscript.

Funding

This work has been funded by ELIOT ANR18-CE40-0030, FAPESP 2018/12579-7 Project and FAPESP 2015/24499-0. It was also supported by CNPq, CAPES and FAPERJ.

Availability of data and materials

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Appendix 1: A lower-bound on the achievable rate

According to [38, 39], the achievable rate is expressed and lower-bounded by using an auxiliary channel law [7, 21]

$$\begin{aligned}
 \bar{I}(X_k; Y_k) &= \lim_{n \rightarrow \infty} \left(-\frac{1}{n} \log_2 P(\mathbf{y}^n) + \frac{1}{n} \log_2 P(\mathbf{y}^n | \mathbf{x}^n) \right), \\
 &\geq \lim_{n \rightarrow \infty} \frac{1}{n} (-\log_2 W(\mathbf{y}^n) + \log_2 W(\mathbf{y}^n | \mathbf{x}^n)).
 \end{aligned} \tag{38}$$

Table 3 State transition description of the code ^(5 7)

State	Next state	Input	Output
00	00	0	0 0
	10	1	1 1
01	00	0	1 1
	10	1	0 0
10	01	0	0 1
	11	1	1 0
11	01	0	1 0
	11	1	0 1

where the auxiliary channel law in (10) is used to reduce the complexity of the computation, such that

$$P(\mathbf{y}_k | \mathbf{y}^{k-1}, x^k) \approx W(\mathbf{y}_k | \mathbf{y}^{k-1}, x^k) = P(\mathbf{y}_k | \mathbf{y}_{k-N}^{k-1}, x_{k-N-L}^k),$$

As illustrated in [21], the probabilities $W(\mathbf{y}^n)$ and $W(\mathbf{y}^n | x^n)$ in (38) can be computed recursively with the forward recursion (20) of the BCJR algorithm covered in Sect. 4.2. By using the state notation s_k , $W(\mathbf{y}^n)$ is determined with $W(\mathbf{y}^k) = \sum_{s_k} W(s_k, \mathbf{y}^k) = \sum_{s_k} f_k(s_k)$, whereas the conditional probability $W(\mathbf{y}^n | x^n)$ is computed with the following recursion rule

$$\begin{aligned} W(\mathbf{y}^k | x^n) &= \tilde{f}_k = W(\mathbf{y}_k | \mathbf{y}^{k-1}, x^n) \cdot W(\mathbf{y}^{k-1} | x^n) \\ &= P(\mathbf{y}_k | \mathbf{y}_{k-N}^{k-1}, s_k, s_{k-1}) \cdot \tilde{f}_{k-1}. \end{aligned} \quad (39)$$

The initialization of the algorithm is performed by defining the metrics $f_{k=0}(s_k)$ and $\tilde{f}_{k=0} = 1$. In practice, it was observed in [38] that during the recursion steps, those branch metrics tend to go to zero with few iterations; therefore, the recursion (20) is slightly changed to

$$f_k(s_k) = \mu_k \sum_{s_{k-1}} P(\mathbf{y}_k | \mathbf{y}_{k-N}^{k-1}, s_k, s_{k-1}) \cdot P(s_k | s_{k-1}) \cdot f_{k-1}(s_{k-1}), \quad (40)$$

where the scaling factors μ_k are chosen such that $\sum_{s_k} f_k(s_k) = 1$. Regarding the recursion in (39), auxiliary variables $\tilde{\mu}_k$ are defined as $\tilde{\mu}_k = P(\mathbf{y}_k | \mathbf{y}_{k-N}^{k-1}, s_k, s_{k-1})^{-1}$, with the aim to express the desired information rate quantity as

$$-\frac{1}{n} \log_2 W(\mathbf{y}^n) + \frac{1}{n} \log_2 W(\mathbf{y}^n | x^n) = \frac{1}{n} \sum_{k=1}^n \log_2 \mu_k - \frac{1}{n} \sum_{k=1}^n \log_2 \tilde{\mu}_k, \quad (41)$$

when a large value for n is considered.

Table 4 Channel decoder with BCJR algorithm

Input	
$\Gamma_r = \begin{bmatrix} s(c_{2r-1} = 0)s(c_{2r} = 0) & s(c_{2r-1} = 1)s(c_{2r} = 1) & 0 & 0 \\ 0 & 0 & s(c_{2r-1} = 0)s(c_{2r} = 1) & s(c_{2r-1} = 1)s(c_{2r} = 0) \\ s(c_{2r-1} = 1)s(c_{2r} = 1) & s(c_{2r-1} = 0)s(c_{2r} = 0) & 0 & 0 \\ 0 & 0 & s(c_{2r-1} = 1)s(c_{2r} = 0) & s(c_{2r-1} = 0)s(c_{2r} = 1) \end{bmatrix}$	
for $r = 1, 2, 3, \dots, m/2$	
$\mathbf{U}(0) = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \mathbf{U}(1) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$	
Initialization	
$\mathbf{f}_0 = [1 \ 0 \ 0 \ 0]^T$	
$\mathbf{b}_{m/2+1} = [1 \ 0 \ 0 \ 0]^T$	
BCJR Algorithm	
for $r = 1$ to $m/2$ do	
$\mathbf{f}_r = \Gamma_r \cdot \mathbf{f}_{r-1}$	▷ forward recursion
$\mathbf{f}_r = \mathbf{f}_r / (\mathbf{f}_r^T \cdot \mathbf{1}_4)$	▷ normalization
end for	
for $r = m/2$ to 1 do	
$\mathbf{b}_r = \Gamma_r^T \cdot \mathbf{b}_{r+1}$	▷ backward recursion
$\mathbf{b}_r = \mathbf{b}_r / (\mathbf{b}_r^T \cdot \mathbf{1}_4)$	▷ normalization
end for	
Output	
$P(b_r = b s(c_1), s(c_2), \dots, s(c_m)) = \mathbf{b}_r^T \cdot (\mathbf{U}(b) \odot \Gamma_r) \cdot \mathbf{f}_{r-1} / (\mathbf{b}_r^T \cdot \Gamma_r \cdot \mathbf{f}_{r-1})$	

Appendix 2: Channel decoder example with BCJR algorithm

To illustrate an example on how to compute (29) and (31), i.e., the outputs of the channel decoder, a memory-2 convolutional code characterized by the generator polynomial (57) is considered. It can be described by the states transitions triggered by the bit inputs, which are illustrated in Table 3.

To describe a BCJR-based channel decoder algorithm, it is practical to employ a matrix notation, which requires the following definitions:

- \mathbf{f}_r is the vector form of the forward probabilities for all states at time index r ;
- \mathbf{b}_r is the vector form of the backward probabilities for all states at time index r ;
- Γ_r is the matrix form of the transition probabilities at time index r .

With this, the channel decoder algorithm follows the description covered in [28], which is presented in Table 4. By considering the input bits, $\mathbf{U}(b)$ is introduced to the vector notation as a matrix of 1's and 0's that translates the sum of applicable probabilities, similar to (23), into an element-wise multiplication, represented by the operator \odot .

The given algorithm is capable of producing the output soft information in (26) with the expressions

$$\begin{aligned}
s'(c_{2r-1}) &= P(c_{2r-1} = c | s(c_1), s(c_2), \dots, s(c_m)) \\
&= \mathbf{b}_r^T \cdot (V_{-1}(c) \odot \mathbf{\Gamma}_r) \cdot \mathbf{f}_{r-1} / (\mathbf{b}_r^T \cdot \mathbf{\Gamma}_r \cdot \mathbf{f}_{r-1}), \\
s'(c_{2r}) &= P(c_{2r} = c | s(c_1), s(c_2), \dots, s(c_m)) \\
&= \mathbf{b}_r^T \cdot (V_0(c) \odot \mathbf{\Gamma}_r) \cdot \mathbf{f}_{r-1} / (\mathbf{b}_r^T \cdot \mathbf{\Gamma}_r \cdot \mathbf{f}_{r-1}),
\end{aligned}$$

where

$$\begin{aligned}
V_{-1}(0) &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, & V_{-1}(1) &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \\
V_0(0) &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, & V_0(1) &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix},
\end{aligned}$$

are matrices that relate to the state transitions with respect to the output bits of the channel code presented in Table 3. With this, the extrinsic LLR described in (29) can be computed as follows

$$\lambda'_{\text{ext}}(c_{2r+i}) = \ln \frac{\mathbf{b}_r^T \cdot (V_i(0) \odot \mathbf{\Gamma}_r) \cdot \mathbf{f}_{r-1}}{\mathbf{b}_r^T \cdot (V_i(1) \odot \mathbf{\Gamma}_r) \cdot \mathbf{f}_{r-1}} - \lambda(c_{2r+i}),$$

where $i \in \{-1, 0\}$. As asserted in [28], this equation faces problems when a numerical evaluation is performed due to the logarithm nature of the difference on the right-hand side (RHS) of the equality. To solve this for practical implementations, $\lambda'_{\text{ext}}(c_q)$ is computed as follows:

$$\lambda'_{\text{ext}}(c_{2r+i}) = \ln \frac{\mathbf{b}_r^T \cdot (V_i(0) \odot \mathbf{\Gamma}_{\text{ext},i,r}) \cdot \mathbf{f}_{r-1}}{\mathbf{b}_r^T \cdot (V_i(1) \odot \mathbf{\Gamma}_{\text{ext},i,r}) \cdot \mathbf{f}_{r-1}}, \quad (42)$$

where $\mathbf{\Gamma}_{\text{ext},i,r}$ are extrinsic transition matrices that has the dependency on the input LLR $\lambda(c_{2r+i})$ removed, while computing $\lambda'_{\text{ext}}(c_{2r+i})$. For example, the extrinsic transition matrix $\mathbf{\Gamma}_{\text{ext},-1,r}$ corresponding to $\lambda'_{\text{ext}}(c_{2r-1})$, for the convolutional code that has been considered so far, is given by

$$\mathbf{\Gamma}_{\text{ext},-1,r} = \begin{bmatrix} s(c_{2r} = 0) & s(c_{2r} = 1) & 0 & 0 \\ 0 & 0 & s(c_{2r} = 1) & s(c_{2r} = 0) \\ s(c_{2r} = 1) & s(c_{2r} = 0) & 0 & 0 \\ 0 & 0 & s(c_{2r} = 0) & s(c_{2r} = 1) \end{bmatrix}.$$

Received: 12 June 2020 Accepted: 6 November 2020

Published online: 20 November 2020

References

1. R.H. Walden, Analog-to-digital converter survey and analysis. *IEEE J. Sel. Areas Commun.* **17**(4), 539–550 (1999)
2. T. Koch, A. Lapidot, Increased capacity per unit-cost by oversampling, in *Proceedings of the IEEE Convention of Electrical and Electronics Engineers in Israel* (Eilat, Israel, 2010)
3. H. Grewal, *Oversampling the adc 12 for higher resolution* (Texas Instruments: Application Report, 2006)
4. E.N. Gilbert, Increased information rate by oversampling. *IEEE Trans. Inf. Theory* **39**(6), 1973–1976 (1993)
5. S. Shamai, Information rates by oversampling the sign of a bandlimited process. *IEEE Trans. Inf. Theory* **40**(4), 1230–1236 (1994)

6. M. Zakai, Band-limited functions and the sampling theorem. *Inf. Control* **8**(2), 143–158 (1965)
7. L. Landau, M. Dörpinghaus, G.P. Fettweis, 1-bit quantization and oversampling at the receiver: Communication over bandlimited channels with noise. *IEEE Commun. Lett.* **21**(5), 1007–1010 (2017)
8. W. Zhang, A general framework for transmission with transceiver distortion and some applications. *IEEE Trans. Commun.* **60**(2), 384–399 (2012)
9. S. Bender, M. Dörpinghaus, G.P. Fettweis, On the achievable rate of bandlimited continuous-time 1-bit quantized AWGN channels, in *Proc. IEEE Int. Symp. Inform. Theory (ISIT)* (Aachen, Germany, 2017), pp. 2083–2087
10. L. Landau, S. Krone, G.P. Fettweis, Intersymbol-interference design for maximum information rates with 1-bit quantization and oversampling at the receiver, in *Proceedings of the International ITG Conference on Systems, Communications and Coding* (Munich, Germany, 2013)
11. A. Gokceoglu, E. Björnson, E.G. Larsson, M. Valkama, Waveform design for massive MISO downlink with energy-efficient receivers adopting 1-bit ADCs, in *Proc. IEEE Int. Conf. Commun. (ICC)* (Kuala Lumpur, Malaysia, 2016), pp. 1–7
12. S. Jacobsson, G. Durisi, M. Coldrey, U. Gustavsson, C. Studer, Throughput analysis of massive MIMO uplink with low-resolution ADCs. *IEEE Trans. Wirel. Commun.* **16**(6), 4038–4051 (2017)
13. A.B. Üçüncü, A.Ö. Yılmaz, Oversampling in one-bit quantized massive MIMO systems and performance analysis. *IEEE Trans. Wirel. Commun.* **17**(12), 7952–7964 (2018)
14. A. Gokceoglu, E. Björnson, E.G. Larsson, M. Valkama, Spatio-temporal waveform design for multi-user massive MIMO downlink with 1-bit receivers. *IEEE J. Sel. Top. Signal Process.* **11**(2), 347–362 (2017)
15. Z. Shao, L.T.N. Landau, R.C. de Lamare, Channel estimation using 1-bit quantization and oversampling for large-scale multiple-antenna systems, in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.* (Brighton, United Kingdom, 2019), pp. 4669–4673
16. Z. Shao, R.C. de Lamare, L.T.N. Landau, Iterative detection and decoding for large-scale multiple-antenna systems with 1-bit ADCs. *IEEE Wirel. Commun. Lett.* **7**(3), 476–479 (2018)
17. L.T.N. Landau, M. Dörpinghaus, R.C. de Lamare, G.P. Fettweis, Achievable rate with 1-bit quantization and oversampling using continuous phase modulation-based sequences. *IEEE Trans. Wirel. Commun.* **17**(10), 7080–7095 (2018)
18. S. Bender, M. Dörpinghaus, G.P. Fettweis, The potential of continuous phase modulation for oversampled 1-bit quantized channels, in *Proc. of the IEEE Int. Workshop on Signal Processing Advances in Wireless Communications* (Cannes, France, 2019)
19. J. Anderson, T. Aulin, C.E. Sundberg, *Digital Phase Modulation* (Plenum Press, New York, 1986)
20. C.E. Sundberg, Continuous phase modulation. *IEEE Commun. Mag.* **24**(4), 25–38 (1986)
21. L.T.N. Landau, M. Dörpinghaus, G.P. Fettweis, 1-Bit quantization and oversampling at the receiver: sequence-based communication. *EURASIP J. Wirel. Commun. Netw.* **17**, 83 (2018)
22. R.R.M. de Alencar, L.T.N. Landau, R.C. de Lamare, Continuous phase modulation with 1-bit quantization and oversampling using iterative detection and decoding, in *2019 53rd Asilomar Conference on Signals, Systems, and Computers* (Pacific Grove, CA, USA, 2019), pp. 1729–1733
23. J. Hong, P.A. Wilford, A hierarchical modulation for upgrading digital broadcast systems. *IEEE Trans. Broadcast.* **51**(2), 223–229 (2005)
24. L. Landau, M. Dörpinghaus, R.C. de Lamare, G.P. Fettweis, Achievable rate with 1-bit quantization and oversampling at the receiver using continuous phase modulation, in *Proc. of the IEEE Int. Conf. on Ubiquitous Wireless Broadband* (Salamanca, Spain, 2017)
25. B.E. Rimoldi, A decomposition approach to CPM. *IEEE Trans. Inf. Theory* **34**(2), 260–270 (1988)
26. L. Bahl, J. Cocke, F. Jelinek, J. Raviv, Optimal decoding of linear codes for minimizing symbol error rate (corresp.). *IEEE Trans. Inf. Theory* **20**(2), 284–287 (1974)
27. A. Genz, Numerical computation of multivariate normal probabilities. *J. Comput. Graph. Stat.* **1**, 141–149 (1992)
28. M. Tüchler, A.C. Singer, Turbo equalization: an overview. *IEEE Trans. Inf. Theory* **57**(2), 920–952 (2011)
29. J. Hagenauer, E. Offer, L. Papke, Iterative decoding of binary block and convolutional codes. *IEEE Trans. Inf. Theory* **42**(2), 429–445 (1996)
30. C. Berrou, A. Glavieux, P. Thitimajshima, Near Shannon limit error-correcting coding and decoding: turbo-codes, in *Proc. IEEE Int. Conf. Commun. (ICC)*, vol. 2 (Geneva, Switzerland, 1993), pp. 1064–1070
31. C. Douillard, M. Jezequel, C. Berrou, Iterative correction of intersymbol interference: turbo equalization. *Eur. Trans. Telecommun.* **6**(5), 507–511 (1995)
32. M.H.M. Costa, A practical demodulator for continuous phase modulation, in *Proc. IEEE Int. Symp. Inform. Theory (ISIT)* (Trondheim, Norway, 1994), p. 88
33. G.D. Forney, Convolutional codes I: algebraic structure. *IEEE Trans. Inf. Theory* **16**(6), 729–738 (1970)
34. J.B. Cain, G.C. Clark, J.M. Geist, Punctured convolutional codes of rate $(n-1)/n$ and simplified maximum likelihood decoding. *IEEE Trans. Inf. Theory* **25**(1), 97–100 (1979)
35. S. ten Brink, Convergence behavior of iteratively decoded parallel concatenated codes. *IEEE Trans. Commun.* **49**(10), 1727–1737 (2001)
36. S.-J. Lee, A.C. Singer, N.R. Shanbhag, Analysis of linear turbo equalizer via exit chart, in *GLOBECOM'03. IEEE Global Telecommunications Conference (IEEE Cat. No. 03CH37489)*, vol. 4 (San Francisco, CA, USA, 2003), pp. 2237–22424
37. M. Tüchler, R. Koetter, A.C. Singer, Turbo equalization: principles and new results. *IEEE Trans. Commun.* **50**(5), 754–767 (2002)
38. D.M. Arnold, H.-A. Loeliger, P.O. Vontobel, A. Kavcic, W. Zeng, Simulation-based computation of information rates for channels with memory. *IEEE Trans. Inf. Theory* **52**(8), 3498–3508 (2006)
39. H.D. Pfister, J.B. Soriaga, P.H. Siegel, On the achievable information rates of finite state ISI channels, in *Proc. IEEE Glob. Comm. Conf. (GLOBECOM)* (San Antonio, TX, USA, 2001)

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.