Pragmatic phase noise compensation for high-order coded modulations

Marco Martalò1,2✉, Gianluigi Ferrari1, Muhammad Asim1, Jonathan Gambini3, Christian Mazzucco3, Giacomo Cannilire3, Sergio Bianchi3, Riccardo Raheli1

1Department of Information Engineering, University of Parma, Italy
2E-Campus University, Novedrate (CO), Italy
3European Research Center, Huawei Technologies, Milan, Italy
✉ E-mail: marco.martalo@unipr.it

Abstract: This study discusses synchronisation in phase noise-impaired spectrally efficient communication systems employing high-order modulations. In particular, an iterative receiver, where demodulation and decoding are separate from maximum a posteriori probability (MAP) synchronisation, is presented. The authors’ separate approach is tailored to the design of pragmatic iterative receiver schemes employing ‘off-the-shelf’ demodulation and decoding blocks. This allows full compatibility with already existing systems, which is attractive from the implementation viewpoint. The proposed MAP synchronisation algorithm also requires very limited knowledge of the phase noise process and achieves near coherent performance with moderate computational complexity. Although the approach is very general, the authors discuss its performance for low-density parity-check-coded pilot symbol-aided quadrature amplitude modulation schemes, demonstrating that a significantly lower computational complexity can be achieved with respect to benchmark joint receivers.

1 Introduction

Wireless and optical communication systems employ oscillators to generate signals used for frequency conversions and synchronisation purposes. Oscillators should generate ideal sinusoidal waveforms, but in practice they are subject to noise, i.e. temporal instabilities and spectral dispersions, which are detrimental for the performance of communication systems. In fact, this noise may cause signal distortion, information loss, and high bit error rate (BER). The noise in the waveform generated by a realistic oscillator is denoted as phase noise [1] and it is a well-known topic in the literature. From an information-theoretic perspective, the ultimate performance in the presence of phase noise has been studied in terms of the achievable information rate under several channel models [2–5].

The impact of phase noise on the performance of communication systems has been also studied from the receiver design viewpoint. In [6], the authors consider the use of a turbo code and design an iterative receiver based on phase discretisation, showing also that the phase can be effectively estimated only by the first component decoder. Another approach, reviewed in [7], is based on an iterative maximum likelihood (ML) algorithm used in an expectation-maximisation (EM) receiver. This approach can be denoted as separate, since detection/decoding is performed separately from phase synchronisation. Extensions to iterative phase locked loop (PLL)-based schemes are proposed in [8–10], but are tailored for specific coded modulation schemes with small size constellations. Variational Bayesian detection with a proper phase noise model is proposed in [11], in which, as an alternative approach, synchronisation and decoding are performed jointly. In [12], the authors exploit the phase error statistics to devise an ML detector and related suboptimal versions based on approximations of the probability density function (PDF) of the phase error. In [13], a Gaussian sum approach is proposed to derive a phase estimation and tracking algorithm in a blind context, i.e. a scenario where no initial phase knowledge is available to eliminate ambiguities. A maximum a posteriori probability (MAP)-based soft-input soft-output (SISO) decoder for a specific case study with 8-phase shift keying (PSK) modulation and turbo coding related to the solution here presented has been brought to our attention after the conclusion of this work [14]. To the best of our knowledge, a benchmark joint solution is proposed in [15], where a suitable factor graph, modelling both the channel code and the phase noise process, is considered for the maximisation of the a posteriori probabilities (APPs) of the information symbols. Therefore, [15] will be considered as the reference algorithm for phase noise compensation in the remainder of this paper.

We derive a MAP phase estimation algorithm, which is then employed in an efficient iterative receiver, where demodulation and decoding operations are separated from the synchronisation. This allows the design of an effective pragmatic receiver for large constellation sizes, which are a key tool to achieve high spectral efficiencies in modern communication systems. The devised algorithm has the key advantage that it does not depend on the considered modulation/coding scheme and requires very limited statistical knowledge of the phase noise process. In fact, in order to keep the computational complexity as low as possible, we first assume that the phase noise is approximately constant over a (sufficiently short) interval. The resulting phase estimation strategy is referred to as block window (BW). The assumption of constant phase noise is then relaxed by employing a sliding window to smooth the piecewise estimation of the BW method; this approach will be referred to as phasor linear prediction (PLP), due to similarities with the phase estimation approach proposed in [16].

Despite the proposed approach is very general, as a representative case study we consider low-density parity-check (LDPC)-coded quadrature amplitude modulation (QAM) schemes. The results of our analysis show very good performance for medium–high constellation sizes and relatively strong phase noise. The proposed receiver exhibits an appreciably lower computational complexity with respect to the joint approach in [15], at the cost of a minor performance degradation. This paper expands upon a preliminary conference version [17].
For small constellation sizes, i.e. $b < b_1$, where $b_1$ is a small positive integer, all bits are channel encoded. Therefore, the spectral efficiency is $\eta = Rb$ bits per symbol.

For larger constellation sizes, i.e. for $b > b_1$, $b_1$ bits are channel encoded and $b - b_1$ are left uncoded (free bits). In particular, the channel encoded bits identify one of $2^{b_1}$ possible subsets in a constellation with $M = 2^b$ points. The free bits, instead, label the points of the selected subset using Gray mapping [19, 20]. According to Ungerboeck set partitioning [21], subsets are observed at the output of a receiver matched filter can be expressed as

$$r_k = x_k e^{j\theta_k} + n_k \quad k = 0, 1, 2, \ldots$$

where $\{x_k\}$ are the transmitted symbols and $\{n_k\}$ are independent and identically distributed (i.i.d.) Gaussian noise complex samples with per-component variance $\sigma^2 = N_0/2$. The per-bit signal-to-noise ratio (SNR) $y_\nu$ can be defined as follows

$$y_\nu \equiv \frac{E_b}{N_0} = \frac{1}{N_0} \frac{E_s}{\eta} = \frac{1 + q}{\eta}.$$

The discrete-time model of phase noise considered in this paper is the well-known Wiener model used in most of the literature [1–6, 23].

$$\theta_k = \theta_{k-1} + \sigma_\nu w_k \quad k = 0, 1, 2, \ldots$$

where $w_k \sim \mathcal{N}(0, 1)$ are i.i.d. zero mean and unit-variance Gaussian increments, $\sigma_\nu$ is a system parameter characterising the phase noise intensity and $\theta_{-1}$ is an initial phase uniformly distributed in $[0, 2\pi)$. This model is representative (in discrete time) of the experimental (continuous) time-varying phase noise process affecting local oscillators, which exhibits a quadratic power spectrum decay [24].

The principle of the iterative receiver, shown in Fig. 1, is based on two separate blocks, which perform SISO demodulation/decoding and phase estimation, and iteratively exchange their own information. In particular, phase estimation, which will be described in Section 3, exploits the symbol APPs output by a standard off-the-shelf SISO decoder. On the other hand, the phase estimates are used by the demodulation/decoding block to compensate and ‘clean’ the received observable. The complete iterative receiver structure is detailed in Section 4.

### 3 APP-based phase synchronisation

Accurate symbol-by-symbol phase tracking typically requires large computational complexity, especially in schemes employing high-order coded modulations. In order to reduce the computational complexity of the phase estimation algorithm, we first approximate the channel phase noise as constant over a (sufficiently short) observation window composed by $\ell$ consecutive samples. This means that we perform block-wise phase noise estimation. In fact, the phase estimate on a length-$\ell$ block is used to compensate for the phase rotations over all symbols of the same block. Moreover, phase estimates of consecutive blocks will differ from each other. This window length $\ell$ is a compromise system parameter which needs to be properly chosen. In fact, a small value of $\ell$ is desirable to make the assumption of constant phase noise realistic; on the other hand,
a large value of $\ell$ may be useful to average the detrimental effects of AWGN.

In Section 3.1, we first focus on disjoint consecutive windows leading to the BW approach. Then, in Section 3.2 a sliding window strategy, referred to as PLP, is also derived as an extension of the BW approach.

### 3.1 Block window

Let us define the following vector of observables associated with a length-$\ell$ window starting at the $k$th epoch

$$\mathbf{r}_k = [r_k, r_{k+1}, \ldots, r_{k+\ell-1}]$$

$\ell = 0, 1, \ldots$

Therefore, the entire sequence of received samples in the same codeword is composed by some number $n_\rho$ of disjoint blocks, i.e.

$$\mathbf{r} = [\mathbf{r}_0, \mathbf{r}_1, \ldots, \mathbf{r}_{n_\rho-1}]$$

where the $i$th $\ell$-symbol block of observables is given by

$$\mathbf{r}_i = [r_{i\ell}, \ldots, r_{i\ell+\ell-1}]; i = 0, 1, 2, \ldots, n_\rho - 1.$$

Similarly, we denote the $i$th disjoint block of corresponding transmitted symbols and estimated phases, respectively, as

$$\mathbf{u}_i = [u_{i\ell}, \ldots, u_{i\ell+\ell-1}]$$

$$\boldsymbol{\hat{\theta}}_i = [\hat{\theta}_{i\ell}, \ldots, \hat{\theta}_{i\ell+\ell-1}].$$

As previously mentioned, the estimated phase processes over the $\ell$ symbols of the $i$th block are kept constant, i.e. $\hat{\theta}_i = \hat{\theta}_i \mathbf{1}_\ell$, where $\hat{\theta}_i$ is a scalar and $\mathbf{1}_\ell$ denotes the length-$\ell$ vector with all elements equal to 1. In other words, the BW phase estimate for the $i$th sample, $k = 0, 1, 2, \ldots$, is equal to

$$\hat{\theta}_i = \hat{\theta}_i; k = i\ell, i\ell + 1, \ldots, (i+1)\ell - 1.$$

Using the assumption of constant phase noise over a block, the MAP phase estimate on the $i$th block is

$$\hat{\theta}_i = \arg \max_{\theta \in [0, 2\pi]} p(\theta | \mathbf{r}_i, \mathbf{x}_i) = \arg \max_{\theta \in [0, 2\pi]} p(\mathbf{r}_i | \theta)$$

$$= \arg \max_{\theta \in [0, 2\pi]} \sum_{\mathbf{x}_i \in \mathcal{X}^\ell} p(\mathbf{r}_i | \theta, \mathbf{x}_i) p(\mathbf{x}_i)$$

in which $p(\cdot | \cdot)$ denotes a conditional PDF and the last line is obtained marginalising with respect to the block of transmitted symbols and discarding the irrelevant ratio of a priori PDFs $p(\theta) / p(\mathbf{r}_i)$, since $\theta$ has a uniform marginal distribution over the interval $[0, 2\pi)$. Assuming the APP of the transmitted symbols is known, one can define the following modified APP-aided MAP phase estimate

$$\hat{\theta}_i = \arg \max_{\theta \in [0, 2\pi]} \sum_{\mathbf{x}_i \in \mathcal{X}^\ell} p(\mathbf{r}_i | \theta, \mathbf{x}_i) p(\mathbf{x}_i | \mathbf{r})$$

where $\mathcal{X}^\ell$ denotes the set of length-$\ell$ sequences with elements drawn from the set $\mathcal{X}$ and $P(\mathbf{x}_i | \mathbf{r})$ represents the APP of the transmitted symbols given the received block. According to the channel model (1), the conditional PDF $p(\mathbf{r}_i | \theta, \mathbf{x}_i)$ is known and Gaussian. The term $P(\mathbf{x}_i | \mathbf{r})$ can be easily computed from the bit log-likelihood ratios (LLRs) generated by the channel decoder for the considered modulation as described above.

Let us now define the following length-$\ell$ vector

$$\mathbf{a}_\ell = [\alpha_i, \alpha_{i+1}, \ldots, \alpha_{i+\ell-1}]$$

where the generic element $\alpha_k, k = 0, 1, \ldots$, is the centre of gravity of the transmitted constellation, obtained by weighting the possible symbol values by the corresponding APPs, at epoch $k$. Specifically, the vector $\mathbf{a}_\ell$ can be computed as

$$\mathbf{a}_\ell = \mathbf{P}_\ell \mathbf{\hat{x}}$$

where $\mathbf{\hat{x}} = [x^{(0)}_\ell, x^{(1)}_\ell, \ldots, x^{(M-1)}_\ell]$ is the vector of constellation symbols and $\mathbf{P}_\ell$ is an $\ell \times M$ matrix with elements

$$P_{\ell m} = P(\tilde{x}_\ell = x^{(m)}_\ell | \mathbf{r})$$

in which $\tilde{x}_\ell$ is the hypothetical symbol at epoch $i\ell + l$, $l = 0, 1, \ldots, \ell - 1$, and $x^{(m)}_\ell \in \mathcal{X}$ is a symbol from the used constellation, $m = 0, 1, \ldots, M - 1$. In other words, $p_{\rho_\ell m}$ is the APP that the symbol transmitted at time instant $i\ell + l$ is equal to the $m$th constellation symbol. Note that, if a constellation symbol $x^{(m)}_\ell \in \mathcal{X}$ has probability close to 1 at epoch $i\ell + l$, then $\alpha_{i\ell+l} \approx x^{(m)}_\ell$. In particular, at time instances corresponding to pilot symbols, $\alpha_{i\ell+l} = x_{i\ell+l}$. On the other hand, if no APP is available, $\alpha_{i\ell+l}$ corresponds to the unweighted centre of the constellation.

The computation of the APPs $p_{\rho_\ell m}$ in (4) can be performed in the following way. An $M$-ary symbol $x_i$ is labelled by $b = \log_2 M$ bits $(c^{(0)}_k, \ldots, c^{(b-1)}_k)$, $c^{(b-1)}_k \in \{0, 1\}$. Therefore, assuming that the bits in a given symbol are a posteriori independent, one can write

$$p_{\rho_\ell m} = \prod_{j=1}^{M} p(c^{(j)}_l | \mathbf{r}) = \prod_{j=1}^{M} \frac{e^{-c^{(j)}_l}}{1 + e^{-c^{(j)}_l}}$$

where

$$\mathcal{L}^{(j)}_{i\ell+l} = \log \frac{p(c^{(j)}_l = 0 | \mathbf{r})}{p(c^{(j)}_l = 1 | \mathbf{r})}$$

are the LLRs output either by the SISO channel decoder (coded bits) or by the SO demapper (free bits). The assumption of a posteriori independence among these bits is expedient to the derivation of the iterative scheme. It may be reasonably verified for channel encoded bits, provided a sufficiently long pseudo-random LDPC code is used. On the other hand, this assumption may be critical for free (encoded) bits unless proper interleaving is used. However, our numerical results show good performance, so this assumption will not be further discussed.

After straightforward manipulations, see, e.g. [27], the APP-based phase estimate for the $i$th symbol ($k = 0, 1, \ldots$) can be expressed as

$$\hat{\theta}_{i\ell} = \arg \left[ \mathbf{r}_i^\dagger \mathbf{a}_\ell \right]$$

where the subscript BW has been introduced to specify the block-wise strategy and $(\cdot)^\dagger$ denotes the transpose complex conjugate, so that an inner product between vectors is considered.

### 3.2 Sliding window

As described in Section 3.1, the BW strategy is based on the use of disjoint consecutive $\ell$-symbol windows. In this section, we extend the phase estimation to the use of an $\ell$-symbol sliding window. Using the vector definitions given in Section 3.1, the BW-based strategy for a length-$\ell$ sliding window starting at epoch $j$ can be written as

$$\hat{\theta}_{j\ell} = \arg \left[ \mathbf{r}_j^\dagger \mathbf{a}_\ell \right]$$

$k = j, j + 1, \ldots, j + \ell - 1$.
One can observe that a specific sample \( r_j \) at epoch \( k \) appears in \( \ell \) overlapping windows, the first starting at epoch \( k - \ell + 1 \) and the last starting at epoch \( k \). Therefore, the following phase estimate at the 4th epoch can be heuristically considered

\[
\hat{\theta}_{k,\text{PLP}} = \arg \left[ \sum_{j=0}^{\ell-1} r_{j+k}(a_{k-j})^H \right]
\]

(6)

where the subscript PLP will be shortly justified. The rationale behind (6) is to smooth the phase estimate (5) by averaging over all possible blocks which the \( k \)th sample belongs to. In the square brackets in (6), the term \( r_{j+k}a_{k-j}^* \) appears \( \ell \) times, the terms \( r_{k-1}a_{k-1}^* \) and \( r_{k+1}a_{k+1}^* \) appear \( \ell - 1 \) times, the terms \( r_{k-2}a_{k-2}^* \) and \( r_{k+2}a_{k+2}^* \) appear \( \ell - 2 \) times and so on. Therefore, (6) can be also written, in scalar form, as

\[
\hat{\theta}_{k,\text{PLP}} = \arg \left[ \sum_{j=-\ell+1}^{\ell-1} \pi_j r_{j+k}a_{k-j}^* \right]
\]

(7)

where the following weights can be recognised

\[
\pi_j = \ell - |j| \quad j = -\ell + 1, \ldots, 0, 1, \ldots, \ell - 1.
\]

The triangular shape of the coefficients \( \{\pi_j\} \) in (7) is intuitively pleasing, since at the 4th epoch the highest weight is assigned to the 4th observable (associated with the 4th transmitted symbol).

The phase estimation strategy (7) is a generalised (APP-based) instance of the PLP approach to phase estimation proposed in [16]. Therefore, we refer to this phase estimation strategy as APP-based PLP. An efficient implementation of (7) should rely on a shift register for the computation of (6). In fact, at the 4th time instant, the shift register contains the terms \( r_{k-4}(a_{k-4})^H \), \( l = 0, \ldots, \ell - 1 \); at epoch \( k + 1 \), the new term \( r_{k+1}(a_{k+1})^H \) is input and the ‘oldest’ term \( r_{k-\ell+4}(a_{k-\ell+4})^H \) is dropped. The computational complexity of this implementation is discussed in Section 6.

\[ r \rightarrow \text{PHASE PRE-COMP} \rightarrow \hat{r} \rightarrow y[n] \rightarrow \text{(BIT) SO DEM} \rightarrow \text{SO CHANNEL DEC [with MLC]} \rightarrow \text{INFO BITS} \rightarrow \hat{a} \]

\[ r \rightarrow e^{-j\hat{\theta}[n-1]} \rightarrow \text{block delay (on iteration)} \]

\[ e^{-j\hat{\theta}[n]} \rightarrow \text{ACCUMULATION} \rightarrow \text{PHASE ESTIMATION} \rightarrow \text{APP-BASED PHASE EST} \rightarrow \text{SYMBOL APP CALC} \rightarrow L_{\text{out},[\ell]}[n] \rightarrow \text{INFO BITS} \rightarrow \hat{a} \]

\[ \text{Fig. 2} \quad \text{Block diagram of the proposed iterative receiver. The dashed line is used only with MLC} \]

4 Iterative receiver

The iterative receiver, shown as the right-most block in Fig. 1, is detailed in Fig. 2, in which dashed lines correspond to connections needed only in the presence of MLC, whereas concentric circles describe element-wise products between vectors.

The first operation at the receiver is a coarse phase pre-compensation of the received observables \( \{r_k\} \) by means of an interpolator, which is based on the use of pilot symbols in the transmitted frame. We denote this phase estimate as \( \hat{\phi}_k \).

Although various interpolation schemes with different performance can be designed, in this paper, we limit ourselves to a linear interpolation to minimise the system delay [28]. In particular, we use \( N_p = 1 \) pilot symbol every \( N \) data symbols; therefore, if the \( k \)th epoch corresponds to a pilot symbol, then the next pilot symbols appear at time instants \( k + N + 1 \), \( k + 2(N + 1) \), \( k + 3(N + 1) \) and so on. At the positions of the pilot symbols, the phase noise is estimated as the difference between the phase of the received sample and the phase of the transmitted pilot symbol

\[
\hat{\phi}_k = \angle r_k x_k^*.
\]

(8)

Note that this estimation is asymptotically exact at high SNR; in fact, in the high-SNR regime \( r_k \approx x_k e^{j\phi_k} \) and, therefore, \( \hat{\phi}_k \approx \theta_k \).

The phase noise affecting the intermediate data symbols is pre-estimated by linearly interpolating two consecutive phase estimates \( \hat{\phi}_k \) and \( \hat{\phi}_{k+1} \) to obtain \( \hat{\phi}_{k} \). In particular, the \( k \)th data symbol is located between two pilot symbols at positions \( [k/N] \) and \( [k/(N + 1)] \), respectively, where \( [x] \) denotes the largest integer smaller than or equal to \( x \). Therefore, the phase estimate at epoch \( k \) can be written as

\[
\hat{\phi}_k = \hat{\phi}_{[k/N]} + \left( \hat{\phi}_{[k/(N + 1)]} - \hat{\phi}_{[k/N]} \right) \left( k - [k/N] \right) \frac{1}{N + 1}.
\]

The use of pre-compensation is expedient to perform the first demodulation/decoding act, which is required to perform APP-based phase estimation. The observable at the output of the pre-compensation block, denoted as \( \tilde{r}_k \), is obtained as a deotation by \( \tilde{\phi}_k \) of the observable \( r_k \) and can be expressed as

\[
\tilde{r}_k = r_k e^{-j\hat{\phi}_k} = x_k e^{(\theta_k - \hat{\phi}_k)} + \tilde{n}_k.
\]

(9)
where \( \{ \tilde{h}_i \} \) are still i.i.d. Gaussian samples with the same variance of \( \{ n_i \} \). The phase noise \( \{ \tilde{\eta}_k - \tilde{\phi}_k \} \) affecting \( \{ \tilde{r}_k \} \) may have a different statistical characterisation with respect to \( \{ \tilde{\eta}_i \} \). However, since the proposed APP-based synchronisation algorithms (namely, BW and PLP) require very limited a priori statistical characterisation of the phase noise, they can still be applied directly and a more accurate statistical characterisation of the phase noise after pre-compensation is not required.

The observables \( \{ \tilde{r}_k \} \) at the output of the pre-compensation block feed the input of the SISO QAM demodulator. We consider a simplified demodulator version for which the approximate LLRs associated with the \( k \)th observable output by the demodulator can be written as [29]

\[
L(c_{j}^{(i)} | r_k) = \frac{1}{2\sigma_u} \left[ \min_{x_j \in \mathcal{X}_j} |\tilde{r}_k - x_j|^2 - \min_{x_j \in \mathcal{X}_j} |\tilde{r}_k - x_{j'}|^2 \right] 
\]

where \( \mathcal{X}_j \) denotes the set of QAM symbols with bit \( c_{j}^{(i)} \) in position \( j \), \( j = 1, 2, \ldots, \log_2 M \), and the following max-log approximation for sufficiently large SNR has been used [30]

\[
\ln \sum_c e^{u(c)} \simeq \max_u f(u)
\]

in which \( u \) belongs to a discrete set and \( f(u) \) is a given function of \( u \). These soft values are used in a SISO decoder which outputs the LLRs on the multi-level coded bits. We now detail the structure of the (multi-level) SISO decoder.

Off-the-shelf channel decoding, e.g. message passing for LDPC codes with internal decoding iterations, is performed for small constellations where all bits are channel encoded. For large constellations, where part of the bits is channel encoded and part is left unencoded, multi-stage decoding [18] is performed. In particular, the soft outputs on the \( b_i \) coded bits output by the LDPC decoder are used to compute the APP of each \( b_i \)-bit subset, denoted as \( S_{k}^{(i)} \), as

\[
P(S_{k}^{(i)}) = \prod_{m=1}^{b_i} P(c_{m}^{(i)} = c_{m}^{(i)})
\]

where \( c_{m}^{(i)} \) is the value of the \( m \)th bit in the label associated with the \( k \)th subset and, to simplify the notation, conditioning on the received signal and dependence on the iteration index are here understood. By using the total probability theorem, one obtains the APPs of the free bits \( f_{i}^{(i)} \) associated with the \( k \)th symbol in terms of the subset APPs as

\[
P(f_{i}^{(i)}) = \sum_{x_{k}} P(S_{k}^{(i)} | f_{i}^{(i)}, S_{k}^{(i)}) P(S_{k}^{(i)})
\]

where the APPs \( P(S_{k}^{(i)} | f_{i}^{(i)}, S_{k}^{(i)}) \) over the QAM subset \( S_{k}^{(i)} \) can be computed in terms of subset LLRs similar to those described in (10). The computational complexity of (11) can be reduced by observing that there are typically a few subsets, namely \( B \), with higher probability and neglecting the contribution from less likely subsets. Considering only the retained subsets, the APPs on the free bits can be therefore computed as

\[
P(f_{i}^{(i)}) = \frac{1}{\sum_{s=1}^{B} P(S_{k}^{(i)})} \sum_{s=1}^{B} P(f_{i}^{(i)} | S_{k}^{(i)}) P(S_{k}^{(i)})
\]

where the factor \( 1/\sum_{s=1}^{B} P(S_{k}^{(i)}) \) is needed for correct normalisation. Results, not reported here for conciseness, show that taking into account the sole most likely subset, i.e. using \( B = 1 \), leads to minor performance degradation and a significant complexity saving. According to this reduced-complexity approximation with \( B = 1 \), hard decisions on the code bits are considered to select the most likely subset of size \( 2^{b_i-b_1} \), so that the APPs of the free bits are then computed restricting (10) to this selected subset.

The LLRs of the (multi-level) coded bits are combined to generate APPs on the constellation symbols, which in turn are passed to the phase synchronisation block. The cascade of phase estimation and demodulation/decoding steps is referred to as an external iteration and \( n_i \) denotes its maximum number. The 0th iteration, corresponding to \( n = 0 \), refers to a first detection/decoding act on the observables \( \tilde{r}_k \) at the output of the pre-compensation block. The LLRs for all symbols in a codeword are collected in the vector \( \mathbf{L}_{\text{out}}[n] \), where \( n \) denotes the external iteration index.

The iterative algorithm can now be summarised as follows (see Fig. 2). Let us define the vectors \( \mathbf{y}[n], \hat{r} \) and \( \hat{\theta}[n - 1] \), by analogy with vector \( r \), as the collection of the corresponding samples within a codeword at the indicated iteration. At the \( n \)th iteration, \( n = 1, \ldots, n_i \), APP-based synchronisation is performed based on the observable vector \( \mathbf{y}[n] \), which is given by

\[
y[n] = \hat{r} \circ e^{-j\hat{\theta}[n-1]}\]

where \( \circ \) denotes an element-wise vector product, and the exponential is applied element-wise to the vector \( \hat{\theta}[n-1] \). The observable \( \mathbf{y}[n] \) is used in the demapping/decoding blocks to compute the symbol APPs needed by (5) or (6) for BW or PLP phase estimation, respectively. Since \( \mathbf{y}[n] \) is a derotated version, at each iteration, of the observable \( \hat{r} \), the phase estimate output by the APP-based synchroniser corresponds to a residual phase error \( \hat{\theta}[n] \), which must be accumulated to obtain the overall phase estimate in (12)

\[
\hat{\theta}[n] = \hat{\theta}[n-1] + \hat{\theta}[n].
\]

The final hard decision on the transmitted information bits in a codeword, denoted as \( \hat{a} \), is made at the end of the \( n_i \)th external iteration.

### 5 Performance analysis

We investigate the BER performance of the proposed receiver schemes with iterative synchronisation and demodulation/decoding for an LDPC-coded QAM scheme according to the simulation set-up summarised in Table 1.

We first investigate the impact of the window length \( \ell \) on the system performance. In Fig. 3, the BER is shown, as a function of \( \ell \), considering a scheme with 64-QAM, \( n_g = 5 \) external iterations, pilot spacing \( N = 50 \) and various values of \( \gamma_b \) (12.1 and 12.7 dB) and \( \sigma_n^2 \) (0.1° and 1°). Both BW and PLP estimation algorithms are considered. From the results in Fig. 3, a compromise value of \( \ell \) can be selected in order to minimise the BER, as discussed at the beginning of Section 3. In particular, \( \ell = 24 \) is a good window

### Table 1 Simulation set-up

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>pilot constellation</td>
<td>4-QAM constellation with energy ( E_s = 2.5E_b )</td>
</tr>
<tr>
<td>LDPC scheme</td>
<td>MLC scheme</td>
</tr>
<tr>
<td>type of LDPC code</td>
<td>Compliant with the exponent parity-check matrix structure described in the WiMAX standard [31, Paragraph 8.4.9.2.5, Annex H]: ( n_g = 12 ) rows, ( n_b = 96 ) columns, spreading factor ( Z = 84 ) constant row weight ( w_r = 32 ) and average column weight ( w_c = 4 )</td>
</tr>
<tr>
<td>maximum number of LDPC internal iterations</td>
<td>50 (our results show that, if BER &lt; ( 10^{-4} ), 5 internal iterations are sufficient)</td>
</tr>
</tbody>
</table>
length with the BW phase estimation strategy, for all considered values of $\gamma_0$ and $\sigma_\Delta$. With the PLP phase estimation strategy the performance improves, since the lowest achievable BER is lower than that relative to the BW case. In general, $\ell = 32$ guarantees approximately a good performance in all the considered PLP cases. Further results (not reported here for brevity) show that good performance can be achieved with $\ell = 32$ and various systems settings, i.e. small or large constellation sizes, mild or strong phase noise and BW or PLP phase estimation. This means that $\ell = 32$ can be used in all considered scenarios demonstrating the proposed algorithms require very limited a priori knowledge on the phase noise statistics.

In Fig. 4, the BER is shown, as a function of $\gamma_0$, considering $n_t = 5$ external iterations, $\ell = 32$, pilot spacing $N = 50$ and various values of $\sigma_\Delta$. The performance of systems embedding BW and PLP strategies is compared for two modulation formats: (Fig. 4a) 16-QAM and (Fig. 4b) 1024-QAM. As a reference, in both figures we also show the performance without iterations, i.e. in the case where, after interpolation, demodulation and decoding are carried out just once. For $\sigma_\Delta = 0^\circ$, due to the residual noise introduced by interpolator and phase estimator one can observe a small performance loss – on the order of $0.15 \div 0.2$ dB – with respect to an ideal coherent system which is fed with exact phase information and does not use interpolation and synchronisation. As expected, the higher the phase noise intensity, the worse the performance; the larger the constellation size, the lower the robustness against phase noise. For strong phase noise intensity, the performance improvement brought by the PLP phase estimation strategy, with respect to the BW strategy, is more evident. The performance of the proposed iterative receivers may improve when the number of external iterations increases. However, other simulation results (not shown here for brevity) indicate that increasing the number of iterations from 3 to 5 leads to a remarkable performance improvement, whereas increasing them from 5 to 10 leads to negligible further advantages.

In Fig. 5, the BER is shown, as a function of $\gamma_0$, for 1024-QAM, $\sigma_\Delta = 0.21^\circ$, $n_t = 5$ external iterations, pilot spacing $N = 48$ and window length $\ell = 32$. The performance of the proposed iterative schemes (with BW and PLP phase estimation strategies) is compared with the performance of the joint iterative receiver proposed in [15], referred to as ‘joint MAP’, operating on $r$. As expected, the best performance is obtained by the reference joint MAP scheme, but the PLP approach exhibits only a 0.1 dB loss. Further results, not shown here for shortness, show that the PLP-based approach is more robust for large constellation sizes, due to the fact that the relative loss, with respect to the joint MAP scheme, decreases with the constellation size. Moreover, the joint MAP approach of [15] requires the design of a proper factor graph for the iterative receiver and knowledge of the phase noise intensity. Finally, note that the considered value of $\sigma_\Delta = 0.21^\circ$ is remarkable, if compared with $1.85^\circ$, which is approximately half the minimum phase difference between 1024-QAM constellation points having the same energy.
6 Computational complexity

In this section, we compare the computational complexity of the proposed iterative synchronisation and decoding algorithms with that of the joint MAP scheme. All algorithms are based on successive (external) iterations between two main functional operations: (i) phase synchronisation and (ii) SISO decoding, in which the symbol APPs are computed by exploiting the selected error correction code. In this paper, we only focus on the first task, since the computational burden related to the decoding block of the LDPC code is common to all the algorithms. The computational complexity is defined as the number of real multiplications and additions per modulated symbol and external iteration. Note that one complex multiplication requires four real multiplications and additions per modulated symbol and external iteration. The relative computational loads of the proposed algorithms are detailed in Table 2.

On the basis of the evaluation outlined above, one can define the normalised computational load of the proposed algorithms as the following ratios between the number of multiplications (additions) required by the proposed solutions and the corresponding ones required by joint MAP

\[ \xi_m = \frac{\text{Number of multiplications in BW or PLP}}{\text{Number of multiplications in [15]}} \]

\[ \xi_A = \frac{\text{Number of additions in BW or PLP}}{\text{Number of additions in [15]}} \]

The relative computational loads \( \xi_m \) and \( \xi_A \) can be computed on the basis of Table 2. The resulting values are shown, as functions of \( M \), in Fig. 6. It can be observed that \( \xi_m \) and \( \xi_A \) are always less than one, i.e. BW and PLP have a computational complexity lower than that of the joint MAP algorithm. Moreover, the overall computational loads of the proposed solutions, with respect to [15], are decreasing functions of the constellation size. In particular, \( \xi_m \) saturates for large constellation sizes, say \( M \geq 64 \), whereas \( \xi_A \) keeps on increasing.

Therefore, the joint MAP algorithm entails numbers of multiplications and additions equal to \( 8 + 18M \) and \( 8 + M(11 + \log_2 M) \), respectively. We remark that the computational complexity of the joint MAP approach, here derived independently, is compliant with that given in [32, Table I]. The number of elementary operations per modulated symbol and external iteration required by the synchronisation function for BW, PLP and joint MAP algorithms are detailed in Table 2.

Table 2 Number of elementary operations per modulated symbol and external iteration required for synchronisation by BW, PLP and joint MAP algorithms

<table>
<thead>
<tr>
<th>Operation</th>
<th>BW</th>
<th>PLP</th>
<th>Joint MAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real multiplications</td>
<td>( \frac{2 \log_2 M}{n} + \frac{2 \log_2 M}{N/n} ) ( n ) + ( \frac{1}{n_0 + 1} ) ( n_0 + 1 ) + ( \frac{1}{n_0 + 1} ) ( n_0 + 1 ) + 3M )</td>
<td>( \frac{2 \log_2 M}{n} + \frac{2 \log_2 M}{N/n} ) ( n ) + ( \frac{1}{n_0 + 1} ) ( n_0 + 1 ) + ( \frac{1}{n_0 + 1} ) ( n_0 + 1 ) + 3M )</td>
<td>( 8 + M(11 + \log_2 M) )</td>
</tr>
<tr>
<td>Real divisions</td>
<td>( \frac{1}{N} ) ( n ) ( n_0 + 1 )</td>
<td>( \frac{1}{N} ) ( n ) ( n_0 + 1 )</td>
<td>( 6 + M )</td>
</tr>
<tr>
<td>Exponentials</td>
<td>( 2M )</td>
<td>( 2M )</td>
<td>( 2M )</td>
</tr>
<tr>
<td>Absolute values</td>
<td>( M )</td>
<td>( \log_2 M )</td>
<td>( \log_2 M )</td>
</tr>
<tr>
<td>Logarithms</td>
<td>( \log_2 M )</td>
<td>( \log_2 M )</td>
<td>( \log_2 M )</td>
</tr>
<tr>
<td>Normalisation</td>
<td>( (1 - \frac{\log_2 M}{n}) ) ( n_0 + 1 ) ( n_0 + 1 )</td>
<td>( (1 - \frac{\log_2 M}{n}) ) ( n_0 + 1 ) ( n_0 + 1 )</td>
<td>( 8 + 18M )</td>
</tr>
<tr>
<td>Minimum</td>
<td>( 2\log_2 M )</td>
<td>( 2\log_2 M )</td>
<td>( 2\log_2 M )</td>
</tr>
</tbody>
</table>

### Fig. 6 Normalised computational loads of the proposed synchronisation algorithms, as functions of \( M \), for the considered simulation settings
decreasing. Therefore, both BW and PLP approaches have a significantly lower computational complexity, with respect to the joint MAP approach, for medium-high constellation sizes, yet incurring a minor performance loss as discussed in the previous Section 5.

7 Concluding remarks

This paper presents low-complexity MAP phase estimation algorithms which may operate effectively with very limited statistical knowledge of the channel phase noise process. These algorithms can be employed in an iterative receiver with separate synchronisation and decoding. In particular, two estimation strategies are considered. First, under the assumption of constant phase noise over a sufficiently short interval, a BW strategy is presented. Then, an improved PLP phase estimation strategy, which encompasses the presence of a sliding window to smooth the piecewise phase estimate of the BW approach is proposed. The results with pilot-aided LDPC-coded QAMs show that the PLP solution guarantees a good performance for medium-high constellation sizes and challenging phase noise scenarios, also with respect to the benchmark joint approach. The proposed solutions are particularly attractive from the implementation viewpoint for spectrally efficient communications employing high-order modulations, due to their low computational complexity and pragmatic nature which allows the use of off-the-shelf detection and decoding subblocks.

8 References