Detection by Multiple Trellises: an Effective Approach to Phase-Uncertain Communications

Michele Franceschini, *Student Member, IEEE,* Gianluigi Ferrari, *Member, IEEE,* Riccardo Raheli, *Member, IEEE* and Aldo Curtoni

***Abstract*— In this paper, we present a novel technique for detection of an encoded information sequence transmitted over a phase-uncertain channel. In order to limit the complexity of the detector, the phase space (i.e., the interval $[0, 2\pi)$) is properly quantized. The proposed detection algorithms are based on an L-fold replication of coherent forward-backward (FB) algorithms operating on L parallel trellises, one per hypothetical phase value. In order to make the receiver robust against a possibly time-varying phase, the proposed algorithms perform a manipulation of the forward and backward metrics computed by the different FB algorithms every N trellis time steps. The performance of the proposed algorithms is investigated considering differentially encoded (DE) quaternary phase shift keying (QPSK). We also consider the serial concatenation of DE-QPSK with an outer low-density parity-check (LDPC) code. Besides having a limited complexity, the proposed SISO algorithms turn out to be robust, flexible, *blind*, i.e., no knowledge of the channel phase statistics is required, and *highly parallelizable*, as it is desirable in high-throughput future wireless communication systems.**

I. Introduction

SINCE the introduction of Turbo-Codes (TC) [1], a great effort has been devoted to develop soft-input soft-output (SISO) [2] detection algorithms suitable for iterative processing. While SISO algorithms were first derived for the additive white Gaussian noise (AWGN) channel, they have been extended to more realistic channels, such as those of interest in wireless communications. In particular, these channels are often characterized by time-varying (stochastic) parameters, whose statistics may not be available at the receiver. An example of such channels is the *phase-uncertain* channel [3], where the transmitted signal undergoes an unknown phase rotation and is affected by AWGN. Another relevant example is a fading channel [4].

Two main approaches to perform detection over channels with parametric uncertainty can be devised:

- *separate* detection and parameter estimation [3];
- *joint* detection and parameter estimation. In this case, parameter estimation is embedded in the detection process, explicitly [5] or implicitly [6].

In this paper, we present a class of low-complexity SISO algorithms, derived from the standard FB algorithm[1] [7], which allow to perform *joint* detection and decoding over a phase-uncertain channel. The proposed algorithms are insensitive to channel phase process statistics. As a practical modulation format, we consider differentially encoded (DE) quaternary phase shift keying (QPSK). In order to investigate the performance of the proposed algorithms in the low signal-to-noise (SNR) region of interest in modern communication systems, our analysis focuses, besides on a simple DE-QPSK scheme, also on the concatenation of a low-density parity-check (LDPC) code [8] with DE-QPSK. The obtained results show that the proposed class of algorithms is characterized by significant robustness against phase noise and low complexity. We point out that the proposed technique can easily be extended to any communication system where the encoder/modulator can be modeled as a finite state machine (FSM) and the channel is characterized by a single time-varying parameter, such as, for example, a frequency non-selective (flat) fading channel. These extensions are currently under investigation.

II. Detection by Multiple Trellises: Preliminaries

In order to set the problem under study and introduce the mathematical notation, we begin by reviewing a modified version of the FB algorithm suitable for channels introducing a time-invariant phase rotation [9]. Assume that the discrete-time equivalent observation, at epoch k, at the output of a channel characterized by a time-invariant phase rotation can be expressed as

$$r_k = c_k e^{j\theta} + n_k \qquad (1)$$

where c_k is a transmitted (possibly coded) symbol, n_k is an AWGN sample and θ is a random channel phase independent of the transmitted data. The *a posteriori* probability (APP) of an information symbol a_k can be expressed as follows:

$$\begin{aligned} P\{a_k|\boldsymbol{r}\} &\propto p(\boldsymbol{r}|a_k)P\{a_k\} \\ &= P\{a_k\}\int_0^{2\pi} p(\boldsymbol{r}|a_k,\vartheta)p(\vartheta)\,\mathrm{d}\vartheta \end{aligned} \qquad (2)$$

where $\boldsymbol{r}$ is the vector of the received observables and the notation "$\propto$" indicates that the first member is proportional to the second through a constant independent on the transmitted information symbol a_k. Since, conditionally on the phase ϑ, the channel is AWGN, the quantity $p(\boldsymbol{r}|a_k,\vartheta)$ can be computed via a standard FB algorithm.[2] Assuming that θ is uniformly distributed, i.e., $p(\vartheta) = 1/2\pi$, one can approximate the integral in (2) through the following finite sum:

$$P\{a_k|\boldsymbol{r}\} \mathrel{\tilde{\propto}} P\{a_k\}\sum_{\vartheta\in\mathcal{T}} p(\boldsymbol{r}|a_k,\vartheta) \qquad (3)$$

where $\mathcal{T} = \{\vartheta_1,\ldots,\vartheta_L\}$ is a set of *properly chosen* L phase values. This corresponds to running several standard FB algorithms (each one associated with a value ϑ_i, $i = 1,\ldots,L$)

The Authors are with the Dipartimento di Ingegneria dell'Informazione, University of Parma, I-43100 Parma, Italy. E-mail: mfrance@tlc.unipr.it

[1]Also widely known as BCJR algorithm from the initials of the original proposers [7].

[2]The reader should observe that while θ is the random variable which denotes the true channel phase, ϑ is a dummy integration or summation variable in (2) and (3), respectively.

in parallel and averaging their output to obtain a quantity approximately proportional to the APP [9].

In the following, we will denote the *forward state metrics* computed during the forward recursion of an FB algorithm as $\{\alpha^{(i)}(S_k)\}$, where the superscript i refers to the FB algorithm associated to the quantized phase value ϑ_i and S_k denotes the FSM state in this trellis. In particular, we assume that $S_k \in \{0, \ldots, \Sigma - 1\}$, where Σ is the number of states of the encoder/modulator. Similarly, we denote the *backward state metrics* as $\{\beta^{(i)}(S_k)\}$.

III. Novel SISO Algorithms

We now extend the previous approach in order to account for a possibly time-varying channel phase. If we assume a slowly varying channel phase, the discrete-time observable can be modeled as in (1) by incorporating a time-varying phase process $\{\theta_k\}$

$$r_k = c_k e^{j\theta_k} + n_k \ . \tag{4}$$

By suitably modeling the stochastic process $\{\theta_k\}$, one could try to develop an exact APP algorithm. Since, in general, the statistics of the process $\{\theta_k\}$ are not known, we prefer to perform a *heuristic* manipulation of the SISO algorithm described in Section II.

Consider the SISO algorithm summarized by (3) and relative to a channel introducing a time-invariant phase. If an hypothetical ϑ_i is different from the *true* channel phase θ, then $p(\boldsymbol{r}|a_k, \vartheta_i)$ is expected to be very small. In fact, it can be verified that the forward and backward state metrics $\{\alpha_k^{(i)}(S_k)\}$ and $\{\beta_k^{(i)}(S_k)\}$ exhibit an exponential decay as functions of the epoch k. In particular, the decay is slowest in the FB algorithm associated to the phase value ϑ_j which is closest to the *true* channel phase θ, leading to state metrics $\{\alpha_k^{(j)}(S_k)\}$ and $\{\beta_k^{(j)}(S_k)\}$ relatively much larger than those computed by the other FB algorithms, i.e., with $i \neq j$.

If the phase is time-varying, we expect that $\{\alpha_k^{(i)}(S_k)\}$ and $\{\beta_k^{(i)}(S_k)\}$ will try to *adapt* to the phase changes. This adaptiveness is limited by the fact that state metrics exhibit a "low-pass filter" behavior, i.e., state metrics have *memory* and can change only slowly. Based on the assumptions made on the phase process, our goal is then to derive practical SISO algorithms by limiting this memory.

A useful observation is that, while in standard applications an FB algorithm is insensitive to a possible multiplication of all forward or backward state metrics by a constant in the algorithm (3), the relative weight of different trellises is important. Accordingly, the algorithm turns out to be insensitive to a *normalization* of the metrics only if this normalization is carried out, at a given epoch, over all forward or backward state metrics of all FB algorithms.

We first assume a block phase model in which θ is uniformly distributed over $[0, 2\pi)$ and is constant over N consecutive time intervals, whereas it is independent from block to block—this assumption, however, will be relaxed in the following. At each length-N interval, i.e., at epochs $k = lN$, $l \in \mathbb{N}$, one could, "average" $\{\alpha_k^{(i)}(S_k)\}$ (and, similarly, $\{\beta_k^{(i)}(S_k)\}$)

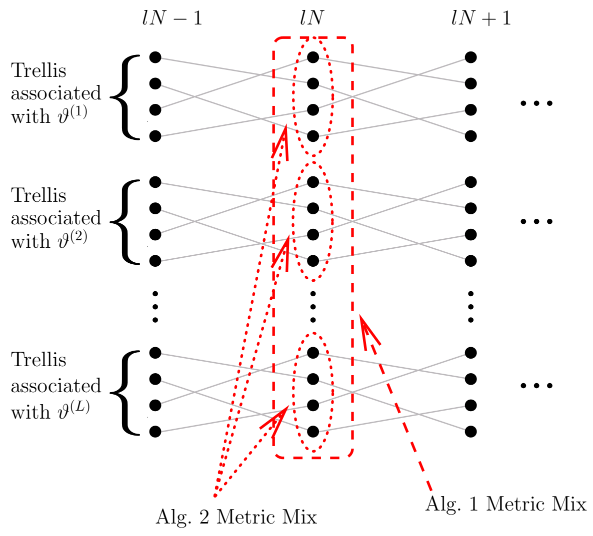

Fig. 1. Pictorial exemplification of the metric mix in the two proposed algorithms.

according to the following rule:

$$\alpha_k^{(i)}(S_k) \longleftarrow \sum_{j=1}^{L} \alpha_k^{(j)}(S_k) \quad i = 1, \ldots, L \quad \forall S_k \tag{5}$$

where the notation "$\longleftarrow$" represents the assignment of a new value. We refer to N as *inter-mix interval*. This corresponds to averaging, for every given state S_k, the state metrics relative to all phases in $\mathcal{T}$: in other words, the state metrics associated with the same state in the various trellises are averaged. We will refer to this algorithm as Algorithm 1. Following the guidelines in [10], this algorithm can be shown to be the optimal APP algorithm *if* the phase is uniformly distributed over the discrete set $\mathcal{T}$, as opposed to uniformly distributed over $[0, 2\pi)$.

If we model the channel phase as slowly time-varying, i.e., we assume it can change in adjacent epochs, and we allow a manipulation of $\{\alpha_k^{(i)}(S_k)\}$ and $\{\beta_k^{(i)}(S_k)\}$ only at epoch $k = lN$, with $l \in \mathbb{N}$, we should take into account possible harmful propagation of the metric distortion due to phase changes in the forward and backward recursions. Heuristically, we have discovered that a simple solution to this distortion propagation consists of performing a normalization of $\{\alpha_k^{(i)}(S_k)\}$ (and, similarly, of $\{\beta_k^{(i)}(S_k)\}$) as follows:

$$\alpha_k^{(i)}(S_k) \longleftarrow \frac{\alpha_k^{(i)}(S_k)}{\sum_{s_k} \alpha_k^{(i)}(s_k)} \quad i = 1, \ldots, L \quad \forall S_k \, . \tag{6}$$

This corresponds to a normalization of the state metrics within each FB algorithm, i.e., trellis by trellis, as opposed to a normalization amongst all trellises (as considered in Algorithm 1). We will refer to this algorithm as Algorithm 2.

These manipulations can be interpreted as a combining or *mixing* of the metrics $\{\alpha_k^{(i)}(S_k)\}$ and $\{\beta_k^{(i)}(S_k)\}$. Fig. 1 gives a pictorial description of the proposed algorithmic family, highlighting the *metric mix* for both Algorithms 1 and 2. Each depicted trellis is associated to a coherent FB algorithm which assumes a given channel phase ϑ_i, $i = 1, \ldots, L$. The

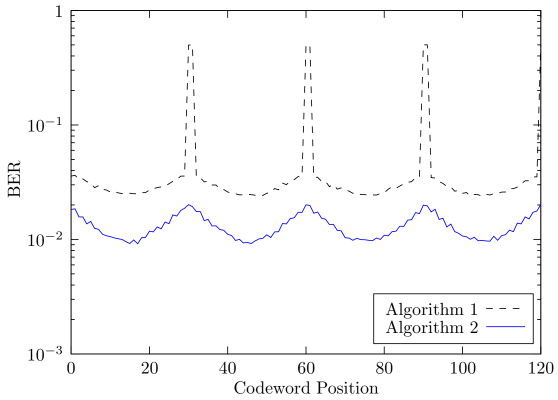

Fig. 2. BER performance of each codeword bit for Algorithm 1 and 2 DE-QPSK. $N = 15$ and $E_b/N_0 = 6.0$ dB.

metric mix for Algorithm 1 is shown to "manipulate" the metrics of all trellises, whereas the metric mix for Algorithm 2 "manipulates" each trellis independently of the other trellises.

IV. Numerical Results

In this section, we assume that transmission over the AWGN channel is affected by a Wiener phase process $\{\theta_k\}$ described by the following recursive relation:

$$\theta_k = \theta_{k-1} + w_k \mod 2\pi \tag{7}$$

where $\{w_k\}$ is a discrete-time white Gaussian process. The standard deviation of w_k, denoted as σ_θ, is representative of the phase noise intensity. As indicated in Section I, the chosen modulation format is DE-QPSK.

Both Algorithm 1 and Algorithm 2 introduced in Section III are considered. For Algorithm 1, the number of quantized phase values is $L = 32$, i.e., 8 values per phase interval between adjacent QPSK symbols: in other words $\vartheta_i = 2\pi \times i/32$. For Algorithm 2, $L = 8$ and $\vartheta_i = 2\pi \times i/8$, accounting for a phase interval of only $\pi/2$. In fact, it is well known that for a time-invariant channel phase, the symmetry of DE-QPSK enables to perform detection accounting only for a phase interval $(0, 2\pi/4)$ [9], [10]. The particular symmetry of Algorithm 2 allows to perform this complexity reduction introducing limited penalty, also in the presence of a time-varying channel phase.

In Fig. 2, Algorithm 1 and 2 are investigated as detectors for DE-QPSK without outer code. The information symbols are transmitted in blocks. The bit error rate (BER) performance relative to the first 120 bits in the transmitted blocks is plotted, versus the bit position in the block. The signal-to-noise ratio (SNR) E_b/N_0, where E_b is the energy per information bit and N_0 is the single-sided noise power spectral density is equal to 6.0 dB. The inter-mix interval is $N = 15$ and the phase noise parameter $\sigma_{theta} = 5.0°$. One can note that the curves exhibit periodicity $2N$, since each QPSK symbol encodes 2 bits. In particular for both algorithms the BER is minimum at the middle point between metric mixes. Algorithm 1, moreover, exhibit a floor, since bits corresponding to the metric mix, i.e., bits at positions $l2N$ and $l2N+1$, are decided at random. This floor has little impact in a concatenated scheme, as that described in the following paragraphs, since bits at positions $l2N$ and $l2N + 1$ are characterized by APP equal to 0.5, i.e., they behave as punctured bits.

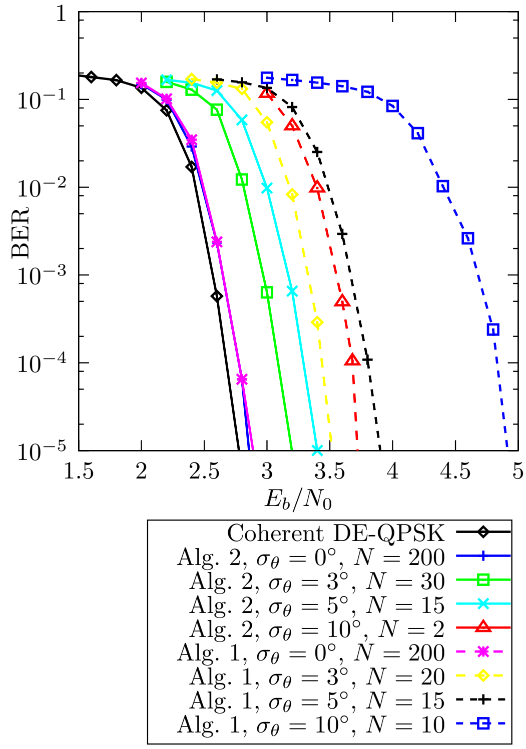

Fig. 3. BER performance of LDPC-coded DE-QPSK schemes where the proposed algorithms (both Algorithm 1 and Algorithm 2) are used.

We now assume that the information sequence is encoded by an outer regular (3,6) LDPC code [8]. The codeword length is set to 6000. The decoder uses a standard LDPC decoder as an external SISO module which exchanges extrinsic information with the DE-QPSK inner detector, where the proposed SISO algorithms are used instead of a coherent SISO algorithm for DE-QPSK. This can obviously be interpreted as a serially concatenated coding scheme.[3] The maximum number of iterations is set to 100.

In Fig. 3, the performance of the described schemes is shown in terms of BER versus SNR. The solid curve represents the performance for transmission over an AWGN channel (without phase noise), where a standard FB algorithm is used in the inner detector—this is the reference performance, relative to an ideal coherent receiver. The remaining curves show the performance obtained with the proposed algorithms. In particular, the curves marked as "Alg1" correspond to the performance of the scheme with Algorithm 1 and those marked as "Alg2" correspond to the performance of the scheme with Algorithm 2. For each algorithm, several values of the phase noise standard deviation σ_θ and inter-mix interval N are considered. The results in Fig. 3 show that, even in the presence of a significant phase noise (for instance, $\sigma_\theta = 10°$), it is possible to "blindly" process the metrics of the trellises while still achieving an SNR loss as limited as 1 dB.

In Fig. 4, the SNR needed to achieve a BER equal to 10^{-3} is shown as a function of σ_θ for both considered algorithms.

[3] A similar scheme is considered also in [11], where the focus is, however, on LDPC code optimization.

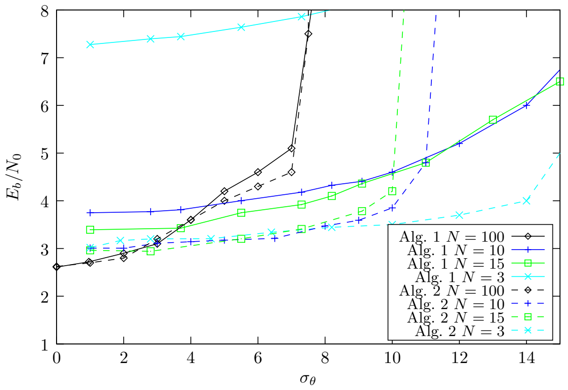

Fig. 4. SNR needed to achieve BER$= 10^{-3}$, considering Algorithms 1 and 2, as a function of σ_θ.

The system and the simulation parameters are those of Fig. 3. One can conclude that the proposed algorithms are blind with respect to the phase noise intensity σ_θ, *as long as* this intensity is lower than a particular value which is a function of N. Beyond this critical value, the SNR needed to achieve the given BER value, i.e., 10^{-3}, diverges rapidly.

From the results in Fig. 3 and Fig. 4, one can conclude that Algorithm 2 performs better than Algorithm 1. This can be attributed to the strong approximations made by Algorithm 1 in "erasing" the phase memory at regular intervals, in an environment in which the phase varies slowly, yet continuously. This leads to wrong metrics in the proximity of time epochs which are multiple of N (where the erase operation is carried out). In other words, Algorithm 1 enforces the strongest metrics among the trellises whereas Algorithm 2 keeps the distribution of the metrics inside each trellis but erases the different weighings of the trellises. If the channel phase has undergone a significant change during the inter-mix interval, and the state metrics have followed this change too slowly, at the end of this interval the "most significant" trellis, i.e., the one characterized by largest state metrics, is not guaranteed to be the one associated with the phase ϑ_i closest to the true channel phase θ. Therefore, the metric mix in Algorithm 1 would enforce the "(decoding) direction" of the wrong trellis, whereas the metric mix in Algorithm 2 would account for possible phase variations, giving all trellises the same "credit."

V. Complexity Analysis and Discussion

In this section, we investigate the complexity of the proposed algorithms with a simple-minded, yet meaningful, approach. In order to highlight the advantages of the proposed algorithms, we compare their complexity with the complexity of the finite-memory FB algorithm proposed in [6]. We will evaluate the computational complexity in terms of additions and multiplications *per trellis section* during a *single recursion*.

We preliminarily denote as $\text{Comp}_{\text{coher}}$ the complexity of an FB algorithm used by a coherent detector. It is possible to show that this complexity is

$$\text{Comp}_{\text{coher}} = \text{O}(\Sigma M) \tag{8}$$

where M is the cardinality of the information symbol set and the notation $\text{O}(\cdot)$ stands for "on the order of." For simplicity, we assume that the complexity of the coherent receiver is the same in terms of multiplications and additions. Moreover, for ease of comparison between different algorithms, we assume that $\text{Comp}_{\text{coher}}$ is *exactly* ΣM.

TABLE I

Complexity per trellis section during a single recursion.

Algorithm	Complexity (Add)	Complexity (Mul)
SISO Algorithm 1	$\Sigma ML + \frac{\Sigma L}{N}$	ΣML
SISO Algorithm 2	$\Sigma ML + \frac{\Sigma L}{N}$	$\Sigma ML + \frac{\Sigma L}{N}$
Finite-memory	$\Sigma M^{N_{\text{fm}}+1}$	$\Sigma M^{N_{\text{fm}}+1}$

We first evaluate the complexity of the proposed algorithms. For both algorithms, L trellis diagrams (each one equal to that of the coherent FB algorithm) are used. Therefore, this increases the complexity of the proposed SISO algorithms to $L\text{Comp}_{\text{coher}}$. At this point, one has to consider the additional complexity of the mixing operations.

- For Algorithm 1, from the updating rule (5) one can conclude that 1 addition (over L phase values) for each state has to be carried out. Therefore, ΣL supplementary additions have to be considered. Since a mixing operation takes place every N transitions, the complexity increase, in terms of additions per trellis section, is $\Sigma L/N$.
- For Algorithm 2, from the updating rule (6) one can conclude that 1 addition (over Σ states) for each component trellis diagram has to be performed. At this point, one division has to be carried out per state and trellis component. Therefore, a mixing operation requires, overall, $L\Sigma$ additions and $L\Sigma$ divisions. The complexity (per trellis section) increase is, therefore, $L\Sigma/N$ in terms of additions, and $L\Sigma/N$ in terms of multiplications/divisions.

The finite-memory FB algorithm proposed in [6] is characterized by a "trellis expansion," in order to partially take into account the channel memory. This memory expansion is described by a finite-memory parameter N_{fm}, which characterizes the number of supplementary information symbols considered in the definition of a state in the trellis diagram at the receiver. The number of states for the computation of the state metrics in a detector/decoder, where a finite-memory FB algorithm is used, is $\Sigma M^{N_{\text{fm}}}$. Therefore, one can conclude that the complexity increases proportionally to $M^{N_{\text{fm}}}$. Denoting by Comp_{fm} the complexity per trellis section (either in terms of additions or multiplications/divisions) in each recursion of a finite-memory FB algorithm, one can write:

$$\text{Comp}_{\text{fm}} = \text{Comp}_{\text{coher}} M^{N_{\text{fm}}}.$$

The complexity of the proposed SISO algorithms and the finite-memory FB algorithm in [6] are summarized in Table I. We remark that the complexity computation is, in practice, implementation-dependent. A noteworthy case is the implementation of the algorithm on a generic purpose processing

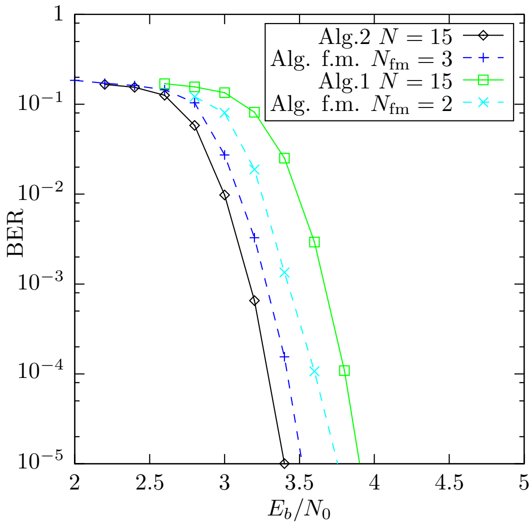

Fig. 5. BER performance of LDPC-coded DE-QPSK schemes where Algorithm 2 and finite memory detection are used. The phase noise parameter is $\sigma_\theta = 5°$.

unit, which usually leads to the serial computation of the quantities involved in the FB algorithms. In particular, it is well known that, due to the negative exponential behavior of the forward and backward state metrics, periodic normalization of these metrics, carried over all states and all trellises, is needed. This normalization takes place at arbitrary time epochs, but usually every $10 \div 100$ time steps. This normalization could be easily modified in order to implement the metric mix described in this paper. Moreover, the proposed SISO algorithms are *intrinsically* highly parallelizable. Exploiting properly this characteristic in the implementation could lead to significant latency reduction and increase of the decoding throughput.

At this point, a careful reader can observe that since the finite-memory approach in [6] is very different from the SISO algorithms proposed in this paper, a meaningful complexity comparison between these algorithms should be carried out for a given performance level. To this purpose, we consider a complexity comparison for the same BER performance at the same SNR, using the same LDPC-coded DE-QPSK scheme used in Section IV. As shown in Fig. 5, the performance obtained by Algorithm 2 with $N = 15$ and $L = 8$, in a scenario with $\sigma_\theta = 5°$, is obtained, when considering the finite-memory FB algorithm, by setting $N_{\text{fm}} = 3$. The remaining simulation parameters are set as in Section IV. The complexity of Algorithm 2, is (in terms of additions) $\Sigma ML + L\Sigma/N \simeq 32\Sigma$, whereas the complexity of the finite-memory FB algorithm is $\Sigma M^{3+1} = 256\Sigma$. This large difference is due to the fact that the complexity of the finite-memory FB algorithm grows *exponentially* with the memory length (quantified by N_{fm}). This exponential behavior of complexity can be seen as a paradox, since, slower phase processes require larger values of N_{fm}. As a consequence, good channels, exhibiting slow phase variations, need larger complexity than bad channels, with high phase dynamics. The proposed SISO Algorithms overcome this paradox, since high values of N require lower complexity.

In [9], [12], [13], other low complexity approaches to the phase noise impairment are proposed, which account for a block constant phase. The additional strength of our algorithms, in particular of Algorithm 2, consists of its improved behavior in case of stationary phase-varying channels in conjunction with the complete blindness with respect to the channel phase statistics. Further investigation of the relationship between the proposed SISO algorithms and those in [9], [12], [13] is currently being pursued.

VI. Conclusions

In this paper, we have derived two simple and effective SISO algorithms by "manipulating" the classical FB algorithm in order to account for the presence of a time-varying unknown phase. The resulting algorithms have been investigated considering uncoded and LDPC-coded DE-QPSK transmission over an AWGN channel with Wiener phase noise. In particular, the algorithm denoted as Algorithm 2, is shown to guarantee large robustness against phase noise, and to operate blindly with respect to the phase noise process statistics. We have shown that the use of the proposed SISO algorithms in the inner DE-QPSK detector leads to a performance very close to that of an ideal coherent scheme. Given their low complexity and high parallelizability, the proposed algorithms are attractive for future high-throughput wireless communication systems.

References

[1] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: turbo-codes," *IEEE Trans. Commun.*, vol. 44, no. 10, pp. 1261–1271, October 1996.

[2] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "A soft-input soft-output APP module for iterative decoding of concatenated codes," *IEEE Commun. Lett.*, vol. 1, pp. 22–24, January 1997.

[3] U. Mengali and A. N. D'Andrea, *Synchronization Techniques for Digital Receivers (Applications of Communications Theory)*. Plenum Press, 1997.

[4] J. G. Proakis, *Digital Communications*, 4th ed. New York: McGraw-Hill, 2001.

[5] A. Anastasopoulos and K. M. Chugg, "Adaptive Soft-Input Soft-Output algorithms for iterative detection with parametric uncertainty," *IEEE Trans. Commun.*, vol. 48, no. 10, pp. 1638–1649, October 2000.

[6] G. Ferrari, G. Colavolpe, and R. Raheli, "Finite memory detection: optimality and reality," in *Proc. Intern. Symp. on Turbo Codes & Relat. Topics*, Brest, France, September 2003.

[7] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inform. Theory*, vol. 20, pp. 284–287, March 1974.

[8] R. G. Gallager, *Low-Density Parity-Check Codes*. Cambridge, MA: MIT Press, 1963.

[9] C. Rong-Rong, R. Koetter, U. Madhow, and D. Agrawal, "Joint non-coherent demodulation and decoding for the block fading channel: a practical framework for approaching shannon capacity," *IEEE Trans. Commun.*, vol. 51, no. 10, pp. 1676 – 1689, October 2003.

[10] M. Peleg, S. Shamai (Shitz), and S. Galán, "Iterative decoding for coded noncoherent MPSK communications over phase-noisy AWGN channel," *IEE Proceedings-Commun.*, vol. 147, pp. 87–95, April 2000.

[11] M. Franceschini, G. Ferrari, , R. Raheli, and A. Curtoni, "Serial concatenation of LDPC codes and differentially encoded modulations," in *Proc. Int. Symp. Inform. Theory and Applic.* (ISITA'04), Parma, Italy, October 2004, also presented at the *Communication Theory Workshop*, Capri Island, Italy, May 2004.

[12] J. K. M. Mackenthun, "A fast algorithm for multiple-symbol differential detection of MPSK," *IEEE Trans. Commun.*, vol. 33, pp. 1471 – 1474, February/March/April 1994.

[13] I. Motedayen-Aval and A. Anastasopoulos, "Polynomial-complexity noncoherent symbol-by-symbol detection with application to adaptive iterative decoding of turbo-like codes," *IEEE Trans. Commun.*, vol. 51, pp. 197–207, February 2003.