Finite Memory: Optimality and Reality

Gianluigi Ferrandi, Giulio Colavolpe and Riccardo Raheli

Università degli Studi di Parma
Dipartimento di Ingegneria dell’Informazione
Parco Area delle Scienze 181A, I-43100 Parma, Italy
E-mail: first-name.last-name@unipr.it

Abstract: In this paper, we present a general approach to combined detection and decoding for channels with memory. A probabilistic derivation of the well-known Viterbi, forward-backward and sum-product algorithms (VA, FB and SP, respectively) shows that a basic metric emerges naturally under very general causality and finite memory conditions. This result implies that detection solutions based on one algorithm can be systematically extended to other algorithms. For stochastic channels described by a suitable parametric model, a finite dependence property is shown to imply this finite memory condition. Unfortunately, this property is seldom met in practice and optimality cannot be claimed. Asymptotical optimality for increasing complexity can sometimes be achieved, so the remaining issue is the algorithm computational efficiency for finite complexity.

Keywords: MAP sequence/symbol detection, iterative detection, graph-based detection, adaptive detection.

1. Introduction

The problem of decoding and detection over noisy channels has long been considered in the literature. In particular, the Viterbi algorithm (VA) [1] provides an efficient way to implement the maximum a posteriori (MAP) sequence detection criterion, based on a suitable trellis representation of the communication system. The key characteristic of the VA is the recursive computation of path metrics associated with a number of surviving paths which is kept constant and equal to the number of trellis states.

In [2], an efficient algorithm to compute the a posteriori probability (APP) of a particular symbol, with complexity on the same order of the VA, is proposed. This algorithm consequently allows to implement exactly the MAP symbol detection criterion. The trellis-based algorithm in [2], usually termed, after the authors, BCJR algorithm, is based on a forward recursion and a backward recursion, and it is thus also referred to as forward-backward (FB) algorithm. The discovery of the so-called turbo codes [3] and of the concept of iterative decoding\(^1\) calls for soft-output algorithms, i.e., algorithms computing “reliability values,” rather than decisions, for the transmitted symbols. Hence, the FB algorithm, which allows the exact computation of the APP, has received a new and constant attention.

In the last years, low density parity check (LDPC) codes, originally invented in [4], were rediscovered [5]. In fact, a simple iterative message passing algorithm, operating on a graph descriptive of the linear block code [4, 6], can be devised for computing reliability values for the transmitted symbols. In particular, this algorithm is termed sum-product (SP) algorithm [7]. In [7], it is shown that if the factor graph corresponding to a linear block code does not contain cycles, then the SP algorithm allows the exact computation of the APP of the transmitted symbols.

In this paper, we show that in order to perform detection for channels with memory, possibly stochastic, the same basic metric, is the key ingredient to implement any of the considered hard-output or soft-output detection algorithms (VA, FB and SP). The algorithms are obtained by a probabilistic derivation based on minimal causality and finite memory conditions. For instance, a metric derived for a VA can be systematically extended to FB and SP algorithms. In particular, we point out that (i) performing combined detection and decoding of trellis codes transmitted over channels with memory leads naturally to the introduction of an augmented trellis (with an increased number of states), whereas (ii) in the case of a Tanner graph-based detection for linear block codes [6], taking into account the channel memory leads to the introduction of another level of nodes in the factor graph (separate detection and decoding). In both cases it is possible to conclude that there is an expansion of the original trellis or graph structures.

In stochastic channels, a finite dependence property implies the finite memory condition. Unfortunately, the finite dependence property rarely holds. However, reasonable detection algorithms can be derived. A few detection strategies (noncoherent and linear predictive) are considered as examples.

2. Causality and Finite Memory

We consider a generic time-continuous transmission system, depicted in Fig. 1. A sequence of inde-
dependent and identically distributed $M$-ary information symbols $\{a_k\}$ are transmitted successively from epoch 0 to epoch $K - 1$. A sequence of information symbols is denoted in vector notation as

$$a_{k_2}^{k_1} = (a_{k_1}, a_{k_1 + 1}, \ldots, a_{k_2}) \ , \ k_2 \geq k_1. \quad (1)$$

For brevity, the entire sequence is denoted by $a$. This sequence is input to the encoder and modulator. The coded and modulated signal is denoted as $s(t, a)$ to emphasize its dependence on the information sequence. The channel is viewed as a noiseless filter (possibly stochastic) with output signal $x(t, a)$, rendered noisy by the addition of thermal noise $n(t)$. The received signal $r(t)$ is observed by the demodulation and decoding block, which outputs a decision sequence $\hat{a}$.

The encoder/modulator block in Fig. 1 is a generic system which evolves, upon receiving at its input the information sequence $a$, through a sequence of states $\{\mu_0, \mu_1, \ldots\}$. In many realistic cases, the encoder/modulator can be described as a time-invariant finite state machine (FSM), e.g., trellis coded or continuous phase modulation. In this case, the state $\mu_k$ belongs to a set of finite cardinality and a time-invariant “next-state” function $ns(\cdot, \cdot)$ describes the evolution of the system as

$$\mu_k = ns(\mu_{k-1}, a_{k-1}). \quad (2)$$

Therefore, the evolution of the encoder/modulator can be described through a trellis diagram, in which there are $M$ exiting branches (in correspondence with $M$ different information symbols) from each state. In the rest of the paper, the initial state $\mu_0$ is assumed to be known.

The received signal can be expressed as

$$r(t) = x(t, a) + n(t). \quad (3)$$

By means of a discretization process, the received signal $r(t)$ can be converted into a time-discrete sequence $r$ [8]. Let us consider a discrete observable $r$ with dimensionality equal to that of the information sequence $a$. As a consequence, there is one observable $r_k$ per information symbol $a_k$, or, formally, $r = r_0^{K-1}$, with a notation similar to that used for the information sequence.

The considered model is based on a sampling rate of one sample per symbol, which is practically sufficient in many cases. In a more general setting, there may be two or more elements of $r$ per information symbol $a_k$, e.g., when a convolutional code or a time-varying channel is considered. These cases can be encompassed by the proposed formulation interpreting

$$\begin{array}{c} \{r_k\} \\
\text{ENC} \\
\text{MOD} \\
\text{s(t, a)} \\
\text{CHANNEL} \\
\text{FILTER} \\
\text{x(t, a)} \\
\text{DET} \\
\text{DEC} \\
\{\hat{a}_k\} \end{array}$$

Figure 1: Communication system.

each observation $r_k$ as a vector of suitable dimensionality.

A causality condition for the considered communication system can be formulated in terms of statistical dependence of the observation sequence $r_0^k$, up to epoch $k$, on the information sequence. Accordingly, a system is causal if

$$p(r_k^n | a_k^0) = p(r_k^n | a_k^0, \mu_k = c) \quad (4)$$

Similarly, a finite memory condition can be formulated as follows:

$$p(r_k | r_0^{k-1}, a_k^0) = p(r_k | r_0^{k-1}, a_k^0, \mu_k = c) \quad (5)$$

where $C$ is a suitable finite memory parameter and $\mu_k = c$ represents the state, at epoch $k - C$, of the encoder/modulator. The finite memory condition (5) is an extension of the folding condition introduced in [9], obtained by accounting for the encoder/modulator state $\mu_k$. It can easily be proved (see Appendix I and Appendix II) that causality and finite memory conditions imply the following equalities:

$$p(r_k | r_0^{k-1}, a_k^0) = p(r_k | r_0^{k-1}, a_k^0, \mu_k = c), \forall D \geq C \quad (6)$$

$$p(r_k | r_0^k, a_k^0) = p(r_k | r_0^{k+1}, a_k^0, \mu_k = c+1) \quad (7)$$

The encoder/modulator block in Fig. 1 can be often decomposed into the cascade of an encoder and a memoryless mapper. In this case, the causality and finite memory conditions imply analogous relations between the observation sequence $r$ and code sequence $c = c_0^{K-1}$, where $c_0$ is a generic code symbol. Accordingly, these conditions can be formulated as follows:

$$p(r_k^n | c) = p(r_k^n | c_0^k) \quad (8)$$

$$p(r_k | r_0^{k-1}, c_0^k) = p(r_k | r_0^{k-1}, c_0^k) \quad (9)$$

We remark, however, that these conditions involve the transmission channel only and do not imply (4) and (5) in general. A case of interest may be that of a linear block code followed by a memoryless modulator. In particular, a linear block code is not guaranteed to be causal\(^3\) so that the channel causality (8) does not imply the system causality (4).

\(^3\)Block-wise causality must be indeed satisfied.
In the case of a linear block code, a trellis representation is possible [2], but the trellis is time-
variant, both in terms of states and branches. In this case, the evolution of the encoder/modulator could be described by a time-variant next-state function ns, . It is immediate to conclude that a Tanner graph representation for a linear block code—where the parity checks determine the structure of the graph—can be more appealing, especially if the parity check equations involve a few symbols (as in the case of LDPC codes) [10].

3. Detection Strategies (Optimality)

Based on the causality and finite memory conditions introduced in Section 2, a probabilistic derivation of important detection/decoding algorithms is now presented. In particular, conditions (4) and (5) will be applied for trellis-based algorithms (VA and FB), whereas conditions (8) and (9) will be applied for factor graph-based algorithms (SP).

3.1. Viterbi Algorithm

The VA is an efficient method to perform MAP sequence detection. The causality and finite memory conditions (4) and (5), the independence of the information symbols, and the chain factorization rule allow one to derive the following:

\[ P\{a|r\} \sim p(r|a)P\{a\} \]
\[ = \prod_{k=0}^{K-1} p(r_k|r_0^{k-1}, a)P\{a_k\} \]
\[ = \prod_{k=0}^{K-1} p(r_k|r_0^{k-1}, a_0^k)P\{a_k\} \]
\[ = \prod_{k=0}^{K-1} p(r_k|r_0^{k-1}, a_{k-C}, \mu_k-C)P\{a_k\} \]

(10)

where the symbol \(\sim\) indicates that two quantities are monotonically related with respect to the variable of interest (in this case, \(a\)). Note that the last step in (10), where the finite memory condition is applied, holds if \(k \geq C\), i.e., in the algorithm regime. In the initial transient for \(k < C\), (10) holds assuming that negative indexes are replaced by 0.

Defining augmented trellis state and branch (transition) as follows:

\[ S_k \triangleq (a_{k-C}, \mu_k-C) \]
\[ T_k \triangleq (S_k, a_k) = (a_{k-C}, \mu_k-C) \]

the k-th factor in (10) can be expressed as

\[ \gamma_k(T_k) \triangleq p(r_k|r_0^{k-1}, a_{k-C}, \mu_k-C)P\{a_k\}. \]

(13)

As well known, the algorithm can now be formulated in the logarithmic domain, by defining the branch metrics \(\lambda_k(T_k) \triangleq \log_2 \gamma_k(T_k)\).

The state \(S_k\) is augmented with respect to the state \(\mu_k\) of the encoder/modulator. This corresponds to considering combined detection and decoding. The next-state function \(NS,S\) for this augmented trellis can be straightforwardly expressed as:

\[
S_{k+1} = NS(S_k, a_k) = (a_{k-C+1}, ns(\mu_{k-C}, a_{k-C})).
\]

(14)

3.2. Forward-Backward Algorithm

The FB algorithm allows to implement the MAP symbol criterion, since it explicitly computes the APP \(P\{a_k|r\}\). Based on the causality and finite memory conditions (4) and (5), the following probabilistic derivation of a FB algorithm is obtained:

\[
P\{a_k|r\} = \sum_{a_k^{k+1}} \sum_{\mu_k-C} P\{a_k^{k+1}, \mu_k-C|r\} \]
\[ = \sum_{a_k^{k+1}} \sum_{\mu_k-C} \cdot P\{a_k^{k+1}, \mu_k-C\} \]
\[ = \sum_{a_k^{k+1}} \sum_{\mu_k-C} \cdot P\{a_k^{k+1}, \mu_k-C\} \]

\[
\times p(a_k^{k+1}, a_0^k)P\{a_k\} \]
\[ = \sum_{a_k^{k+1}} \sum_{\mu_k-C} \cdot P\{a_k^{k+1}, \mu_k-C\} \]
\[ \cdot P(r_k|a_0^{k+1}, a_k)P\{a_0^{k+1}\} \]
\[ \cdot P(r_0^k|a_0^{k+1}, \mu_k-C)P\{a_0^{k+1}\} \]

(15)

where Bayes, marginalization, and chain rules have been used. Based on (7) and causality, it follows that

\[
p(r_{k-1}^1|a_{k-C}, \mu_k-C) = \sum_{a_{k-C}} p(r_{k-1}^1|a_{k-C}, \mu_k-C). \]

(16)

Based on causality, one can also write:

\[
p(r_{k-1}^0|a_{k-C}, \mu_k-C) = p(r_{k-1}^0|a_{k-C}, \mu_k-C). \]

(17)

Recalling the independence of the information symbols, (15) can be rewritten as follows:

\[
P\{a_k|r\} \sim \sum_{a_{k-C}} \sum_{\mu_k-C} \cdot P(r_k|a_0^{k+1}, a_{k-C+1}, \mu_k-C+1) \]
\[ \cdot p(r_k|a_0^{k+1}, a_{k-C}, \mu_k-C) \]
\[ \cdot P(a_0^{k+1}|a_{k-C}, \mu_k-C) \]
\[ \cdot P\{a_0^{k+1}\} \]

(18)

Considering augmented state \(S_k\) and transition \(T_k\) as in (11) and (12), and defining

\[
\beta_{k+1}(S_{k+1}) \triangleq p(r_{k+1}^k|a_{k-C+1}, \mu_k-C+1) \]
\[ \alpha_k(S_k) \triangleq p(r_{k+1}^k|a_{k-C}, \mu_k-C) \]

(19)

(20)
the APP in (15) can be finally formulated as follows:

\[
P\{a_k | r \} \sim \sum_{S_k} \beta_{k+1}(S_{k+1}) \gamma_k(T_k) \alpha_k(S_k)
\]  

(21)

where \( \gamma_k(T_k) \) is defined in (13).

Based on the causality and finite memory conditions, the quantities \( \alpha_k(S_k) \) and \( \beta_{k+1}(S_{k+1}) \) can be computed by means of a forward and a backward recursion, respectively. More precisely, in Appendix III it is shown that

\[
\alpha_k(S_k) = \sum_{S_{k-1}} \alpha_{k-1}(S_{k-1}) \gamma_{k-1}(T_{k-1})
\]

(22)

\[
\beta_k(S_k) = \sum_{S_{k+1}} \beta_{k+1}(S_{k+1}) \gamma_k(T_k)
\]

(23)

where the notation \( T_k : S_k \) indicates all transitions \( T_k \) compatible with state \( S_k \). As usual, proper boundary conditions \{\( \alpha_0(S_0) \)\} and \{\( \beta_{K+1}(S_{K+1}) \)\} must be specified. The algorithm can be also formulated in the logarithmic domain based on the branch metric \( \lambda_k(T_k) = \log \gamma_k(T_k) \).

We remark that this FB formulation is based on very general causality and finite memory conditions, and it does not make any assumption on the specific nature of the channel. We also remark that any detection strategy designed for implementation with a VA can be systematically extended to a FB algorithm.

3.3. Sum-Product Algorithm

The application of the SP algorithm [7] to a factor graph representing the APP of the transmitted code sequence \( c \), given the observation sequence \( r \), allows the exact or approximate computation of the symbol marginal APPs [7]. Therefore, this algorithm may be used to implement a MAP symbol detection algorithm.

The code sequence APP may be expressed as

\[
P\{c | r \} \sim P\{c\} p(r|c) = P\{c\} \prod_{k=0}^{K-1} p(r_k | r_{0:k-1}, c_0^k)
\]

(24)

where the causality condition (8) has been used.

Assuming that the a priori distribution of the transmitted codewords is uniform and denoting by \( \chi(c) \) the code characteristic function\(^4\), under the finite memory condition (9) we have

\[
P\{c | r \} \sim \chi(c) \prod_{k=0}^{K-1} p(r_k | r_{0:k-1}, c_0^k, c_{k+1}^c).
\]

(25)

\(^4\text{Following [7], } P\{c\} = \chi(c) / M^K, \text{ where } M^K \text{ is the number of codewords.}\)

Figure 2: Overall factor graph for \( C = 2 \).

The corresponding factor graph is shown in Fig. 2 for \( C = 2 \), representing both the code constraints (described by \( \chi(c) \)) and the channel behavior. With respect to SP-based decoding schemes for linear block codes (e.g., LDPC codes) over a memoryless channel, additional factor nodes must be added at the bottom of the graph, as shown in Fig. 2. These additional factor nodes perform a marginalization, based on the channel model, without taking into account the code constraints—the application of the SP algorithm to this factor graph leads to a scheme for separate detection and decoding. This approach is different from that proposed in [11] in which new variable nodes, representing the unknown channel parameters, are introduced.

The quality of the convergence of the SP algorithm to the exact marginal probabilities is in general determined by the girth of the graph.\(^5\) As an example, in designing LDPC codes, cycles of length 4 must be avoided to ensure decoding convergence. The graph derived from the proposed factorization has, in general, girth 4, involving the factor nodes which model the channel behavior. However, we verified by computer simulations that these length-4 cycles often do not affect the convergence of the algorithm (see also [12] for details). If this is not the case, as for transmissions over ISI channels, factor graph transformations can be adopted [13].

3.4. Exact Applications

Significant examples where the causality and finite memory conditions strictly hold can be found in transmission over channels with finite inter-symbol interference (ISI), possibly encompassing a nonlinearity with finite memory. The quantity \( \gamma_k(T_k) \) in (13) simplifies, by dropping the conditioning observables, to

\[
\gamma_k(T_k) = p(r_k | a_k^{k-c} , \mu_k-c) P\{a_k\}
\]

(26)

where, in this case, \( C \) equals the channel dispersion. We remark that \( \gamma_k(T_k) \) in (26) can be directly used

\(^5\text{A cycle is a closed path in the graph and its length is defined as the corresponding number of path edges. The length of the shortest cycle is the girth of the graph.}\)
both in a VA and a FB algorithm. A similar property holds for (9), of interest, for example, in the case of transmission of linear block codes over ISI channels. Hence, the SP algorithm can also be applied [13].

4. Stochastic Channels (Reality)

In the case of a channel characterized by parameters affected by stochastic uncertainty, the observations \( r_k \) are dependent, so that the channel memory may not be finite. A very general parametric model for the observation \( r_k \) is the following:

\[
r_k = g(a_{k-L}, \mu_{k-L}, \theta_k^k) + n_k
\]  

(27)

where \( L \) is an integer, \( \theta_k^k \) is a sequence of stochastic parameters independent from \( a \), and \( n_k \) is an additive noise sample. Under this channel model, the following finite dependence property

\[
p(r_k | r_{k-1}^{k+\infty}, a_k) = p(r_k | r_{k-1}^{k-\infty}, a_k^k, \mu_k, \theta_k^k)
\]  

(28)

where \( N \) is defined as dependence length parameter, is sufficient to guarantee a finite memory condition. In fact, as shown in Appendix IV, (28) implies the following:

\[
p(r_k | r_{k-1}^{k-1}, a_k) = p(r_k | r_{k-1}^{k-1}, a_k^c, \mu_k, \theta_k^c)
\]  

(29)

where the finite memory parameter is \( C = N + L \). It is immediate to recognize that (29) represents a special case of (5). As a consequence, all the derivations in the previous section hold.

A statistical description of the stochastic parameter allows us to compute \( \gamma_k(T_k) \) according to:

\[
\gamma_k(T_k) = \frac{p(r_k | r_{k-1}^{k-N}, T_k)}{p(r_k | r_{k-1}^{k-N}, S_k)} P\{a_k\} = \frac{E_{\theta_k^k} [p(r_k | r_{k-1}^{k-N}, \theta_k^k)]}{E_{\theta_k^c} [p(r_k | r_{k-1}^{k-N}, \theta_k^c)]} P\{a_k\}. \tag{30}
\]

Unfortunately, the above exact result is limited by the fact that in realistic scenarios the finite dependence property (28) is seldom met exactly [9, 14]. However, this result suggests a reasonable approach to devise approximate detection algorithms whenever the conditional observations are asymptotically independent for increasing index difference.

5. Examples of Applications

As a first example, we assume that the channel introduces an unknown phase rotation, modeled as a time-invariant random variable \( \theta \) with uniform distribution in \([0, 2\pi)\). We consider coded linear modulations at the transmitter side. In this case, the samples at the output of a matched filter have the following expression:

\[
r_k = c_k e^{j\theta} + n_k
\]  

(31)

where \( c_k \) represents the modulated, and possibly coded, symbol and \( n_k \) is an additive white Gaussian noise (AWGN) sample of variance \( \sigma_n^2 \). The observation (31) is a special case of (27) with \( L = 0 \) (\( C = N \)) and \( \theta_k = \theta \). It is immediate to conclude that, being \( \theta \) time-invariant, the channel memory is infinite. Hence, the finite dependence property can be claimed in an approximate sense only. On the basis of the considered phase model, the metric \( \gamma_k \) can be expressed as follows:

\[
\gamma_k(T_k) = \frac{E_{\theta} [p(r_k | r_{k-c}^c, \theta)] \cdot P\{a_k\}}{E_{\theta} [p(r_k | r_{k-c}^c, S_k)] \cdot P\{a_k\}} \approx e^{-\frac{1}{2\sigma_n^2} \left( \sum_{c=0}^{C-1} r_{k-c} \bar{c}_{k-c} \right)^2} \cdot P\{a_k\}. \tag{32}
\]

As a second example, we consider transmission over a flat fading channel \( (L = 0 \text{ and } C = N) \). Assuming, for the sake of simplicity, that a sampling rate of one symbol per information symbol is adequate, the observation can be expressed as:

\[
r_k = f_k c_k + n_k \tag{33}
\]

where \( \{f_k\} \) is a sequence of Gaussian random variables with autocorrelation sequence modeled according to isotropic scattering, i.e., given by \( E\{f_k f_{k-n}\} = J_0(2\pi BTn) \), where \( J_0() \) is the zero-th order Bessel function of the first kind and \( BT \) is the normalized Doppler rate. In this case, the finite dependence property is an approximation as well, and the quantity \( \gamma_k \) can be computed considering linear prediction [15], i.e.,

\[
\gamma_k(T_k) = \frac{1}{2\pi\sigma_k^2} e^{-\frac{1}{4\sigma_k^2} \left( \sum_{c=0}^{C-1} r_{k-c} \bar{c}_{k-c} \right)^2} \cdot P\{a_k\} \tag{34}
\]

where the finite dependence parameter \( C \) can be interpreted in this case as the prediction order, \( \{w_i\}_{i=1}^C \), are the prediction coefficients, and \( \sigma_n^2 \) represents the prediction error.

The performance of the considered decoding algorithms is assessed by means of computer simulations in terms of bit-error-rate (BER) versus the bit-signal-to-noise ratio \( E_b/N_0 \), \( E_b \) being the received energy per information bit and \( N_0 \) the one-sided noise power spectral density.

We first consider iterative detection of a parallel concatenated convolutional code (PCCC). In particular, we consider a PCCC of rate 1/2 with 16-state binary recursive systematic convolutional (RSC) component codes with generators \( G_1 = (37)_8 \) and \( G_2 = \)
Figure 3: Noncoherent iterative detection of a PCCC with BPSK.

Figure 4: Linear prediction-based iterative detection of a SCCC with QPSK.

(21). The inner pseudo-random bit interleaver is 32 x 32. The output modulation is binary phase-shift keying (BPSK). The overall code is noncoherently non-catastrophic [16]. In Fig. 3, the performance of a receiver based on noncoherent FB algorithms at each component decoder is shown for various values of the assumed memory parameter C. For comparison, the coherent performance, i.e., assuming perfect phase synchronization at the receiver side, is also shown. In all cases, 5 decoding iterations are considered. The complexity of the noncoherent FB detection algorithm becomes significant for C > 6, and reduced-state detection techniques for FB algorithms, as proposed in [17], become fundamental.

We then consider transmission of a serially concatenated convolutional code (SCCC) on a Rayleigh flat fading channel with normalized Doppler rate \( BT = 0.01 \). The code consists of an outer 4-state, rate-1/2 convolutional code connected through a length-

1024 pseudo-random interleaver to an inner 4-state, rate-1/2 convolutional code. The respective generator polynomial matrices are given by \( G_o(D) = [1 + D + D^2 + D^3] \) and \( G_c(D) = [1 + D + D^2] \) \(/(1 + D + D^3)\). The output symbols are mapped to a quaternary PSK (QPSK) constellation with Gray mapping. In Fig. 4, the performance of a receiver based on linear prediction-based detection at the inner decoder is shown, for various values of the prediction order C. For comparison, the coherent performance, i.e., assuming that the fading coefficients are perfectly known, is also shown. In all cases, 5 decoding iterations are considered. As in the previous example, increasing further the prediction order calls for the use of complexity reduction techniques.

In Fig. 5, the performance of the SP algorithm on the described factor graphs for different values of \( C \) is shown in the case of the previous random-phase channel. The code is a (3,6) regular LDPC code with codewords of length 4000. BPSK modulation is used and a maximum of 200 iterations of the SP algorithm on the overall graph is allowed, using the flooding schedule. A pilot symbol every 19 coded bits is added for ambiguity problems, and accounted for in the computation of the signal-to-noise ratio. Even in this case, for increasing values of \( C \), the performance approaches that of the corresponding coherent receiver. In this case as well, the complexity can be reduced by applying techniques similar to reduced-state sequence detection [13]. Moreover, by suitably modifying the factor graphs, in the case of equal energy signaling it is possible to obtain a modified SP algorithm with complexity linear in \( C \) [12].

6. Concluding Remarks

In this paper, a general detection framework for channels with memory has been presented. We showed that the same basic metric can be used in all
considered hard-output and soft-output algorithms. Accounting for the channel memory, one can consider combined trellis-based detection and decoding for trellis codes or separate graph-based detection and decoding for linear block codes.

For stochastic channels, the proposed approach based on a finite dependence property, can sometimes guarantee asymptotical optimality for increasing finite memory parameter \( C \), at the expense of a slow convergence rate. Hence, new methods to approach optimality characterized by a better computational efficiency and faster convergence to the optimal performance could be investigated. The relationship between combined detection and decoding for graph-based algorithms could also be further investigated.

**Appendix I: Proof of (6)**

We now show that (5) implies (6):

\[
p(r_k | r_{0}^{k-1}, a_{k-D}^{k}, h_{k-D}) = \sum_{a_{0}^{k-D-1}} p(r_k | r_{0}^{k-1}, a_{0}^{k}, h_{k-D}) \cdot P\{a_{0}^{k-D-1} | r_{0}^{k-1}, a_{k-D}^{k}, h_{k-D}\}. \]

(35)

If a sequence \( a_{0}^{k-D-1} \), given \( h_{k-D} \), is incompatible with \( h_{k-D} \), then \( P\{a_{0}^{k-D-1} | r_{0}^{k-1}, a_{k-D}^{k}, h_{k-D}\} = 0 \). Hence, for any sequence \( a_{0}^{k-D-1} \) compatible with \( h_{k-D} \)

\[
p(r_k | r_{0}^{k-1}, a_{0}^{k}, h_{k-D}) = p(r_k | r_{0}^{k-1}, a_{0}^{k}) = p(r_k | r_{0}^{k-1}, a_{0}^{k}, a_{k-C}^{k}, h_{k-C}) \]

(36)

where the last equality is based on the finite memory condition (5). Hence, (35) becomes:

\[
p(r_k | r_{0}^{k-1}, a_{0}^{k}, h_{k-D}) = \sum_{a_{0}^{k-D-1}} p(r_k | r_{0}^{k-1}, a_{0}^{k}, a_{k-C}^{k}, h_{k-C}) \cdot P\{a_{0}^{k-D-1} | r_{0}^{k-1}, a_{k-C}^{k}, h_{k-C}\}.
\]

(37)

where, based on the causality and finite memory conditions each term in (39) can be expressed as

\[
p(r_{i} | r_{0}^{i-1}, a_{0}^{K-1}) = p(r_{i} | r_{0}^{i-1}, a_{k-C+i-1}, h_{k-C+i-1}). \]

(40)

Hence, from (39) it is immediate to write:

\[
p(r_{k+i} | r_{0}^{i-1}, a_{0}^{K-1}) = \prod_{i=k+1}^{K-1} p(r_{i} | r_{0}^{i-1}, a_{k-C+i-1}, h_{k-C+i-1}). \]

(41)

Finally, (38) becomes:

\[
p(r_{k+i} | r_{0}^{i-1}, a_{0}^{K-1}) = \sum_{a_{k-C+i}^{K-1}} p(r_{k+i} | r_{0}^{i-1}, a_{k-C+i}^{K-1})P\{a_{k-C+i}^{K-1} \}
\]

\[
= p(r_{k+i} | r_{0}^{i-1}, a_{k-C+i}^{K-1}, h_{k-C+i-1}). \]

(42)

**Appendix III: Proof of (22) and (23)**

Applying the Bayes and marginalization rules, it is possible to write:

\[
a_{k}(S_{k}) = p(r_{k}^{k-1} | a_{k-C}^{k}, h_{k-C})P\{a_{k-C}^{k}, h_{k-C}\}
\]

\[
= \sum_{h_{c-C}} p(r_{k}^{k-1} | a_{k-C}^{k}, h_{k-C})P\{a_{k-C}^{k}, h_{k-C}\}
\]

\[
= \sum_{h_{c-C}} P\{a_{k-C}^{k-1} \} | a_{k-C}^{k}, h_{k-C} \} = p(r_{k}^{k-1} | a_{k-C}^{k}, h_{k-C}^{k-1}). \]

(43)

Indicating concisely by \( T_{k-C} : S_{k} \) the summation set in (43) and applying Bayes and chain factorization rules, (43) can be expressed as follows:

\[
a_{k}(S_{k}) = \sum_{T_{k-C} : S_{k}} p(r_{k}^{k-1} | a_{k-C}^{k}, h_{k-C})P\{a_{k-C}^{k}, h_{k-C}\}
\]

\[
= \sum_{T_{k-C} : S_{k}} \sum_{T_{k-C}} P\{a_{k-C}^{k-1} \} | a_{k-C}^{k}, h_{k-C} \} = p(r_{k}^{k-1} | a_{k-C}^{k}, h_{k-C}^{k-1}). \]

(44)

Owing to causality and independence of the information symbols, respectively, the following identities hold:

\[
p(r_{0}^{k-2} | a_{k-C}^{k-2}, h_{k-C}^{k-1}) = p(r_{0}^{k-2} | a_{k-C}^{k-2}, h_{k-C}^{k-1}) \]

\[
P\{a_{k-C}^{k-1} \} | a_{k-C}^{k-2}, h_{k-C}^{k-1} \} = P\{a_{k-C}^{k-1} \} | a_{k-C}^{k-2}, h_{k-C}^{k-1} \} \]

(45)

Finally, (44) can be expressed as follows:

\[
a_{k}(S_{k}) = \sum_{T_{k-C} : S_{k}} \sum_{T_{k-C}} p(r_{k}^{k-2} | a_{k-C}^{k-2}, h_{k-C}^{k-1})P\{a_{k-C}^{k-1} \} | a_{k-C}^{k-2}, h_{k-C}^{k-1} \}
\]

\[
= \sum_{T_{k-C} : S_{k}} \sum_{T_{k-C}} p(r_{k}^{k-2} | a_{k-C}^{k-2}, h_{k-C}^{k-1})P\{a_{k-C}^{k-1} \} | a_{k-C}^{k-2}, h_{k-C}^{k-1} \}
\]

(46)

\[
a_{k}(S_{k}) = \sum_{T_{k-C} : S_{k}} \sum_{T_{k-C}} p(r_{k}^{k-2} | a_{k-C}^{k-2}, h_{k-C}^{k-1})P\{a_{k-C}^{k-1} \} | a_{k-C}^{k-2}, h_{k-C}^{k-1} \}
\]

\[
= \sum_{T_{k-C} : S_{k}} \sum_{T_{k-C}} p(r_{k}^{k-2} | a_{k-C}^{k-2}, h_{k-C}^{k-1})P\{a_{k-C}^{k-1} \} | a_{k-C}^{k-2}, h_{k-C}^{k-1} \}
\]

(47)
\[
    = \sum_{T_{k-1}: S_k} \gamma_{k-1}(T_{k-1}) a_{k-1}(S_{k-1})
\]
which corresponds to (22).

The backward recursion (23) can be similarly obtained.

**Appendix IV: Proof of (29)**

Based on (28), one can write

\[
    p(r_k|r_0^{k-1}, a_k^0) = p(r_k|r_{k-N}^k, a_k^0) = \frac{p(r_{k-N}^k|a_0^k)}{p(r_{k-N}^k|a_0^k)}
\]

The conditional (probability density function) pdf at the numerator of (48) can be expressed, by applying the total probability theorem, as follows:

\[
    p(r_{k-N}^k|a_0^k) = \int \cdots \int p(r_{k-N}^k|\theta_0^k) p(\theta_0^k|a_0^k) d\theta_0^k.
\]

Owing to the considered observation model (27), it is immediate to conclude that

\[
    p(r_{k-N}^k|a_0^k, \theta_0^k) = p(r_{k-N}^k|a_{k-L}^k, \mu_{k-L}, \theta_0^k)
\]

where \( C = N + L \). Being the stochastic parameters independent from the information symbols, the second pdf inside the integral in (49) can be equivalently expressed as \( p(\theta_0^k|a_{k-L-L}^k, \mu_{k-L}) \). Finally, the integral (49) becomes:

\[
    p(r_{k-N}^k|a_0^k) = \int \cdots \int p(r_{k-N}^k|a_{k-L}^k, \mu_{k-L}, \theta_0^k)
\]

\[
    \cdot p(\theta_0^k|a_{k-L}^k, \mu_{k-L}) d\theta_0^k
\]

\[
    = p(r_{k-N}^k|a_{k-L}^k, \mu_{k-L}).
\]

Applying the same reasoning to the denominator of (48) (taking also into account the causality of the system), one can conclude that

\[
    p(r_k|r_0^{k-1}, a_k^0) = \frac{p(r_{k-N}^k|a_{k-L}^k, \mu_{k-L})}{p(r_{k-N}^k|a_{k-L}^k, \mu_{k-L})}
\]

\[
    = p(r_k|r_{k-L-L}^k, a_{k-L}^k, \mu_{k-L})
\]

which corresponds to (29).

**REFERENCES**


