

RESEARCH

Open Access

Low-complexity *a posteriori* probability approximation in EM-based channel estimation for trellis-coded systems

Nico Aerts*, Iancu Avram and Marc Moeneclaey

Abstract

When estimating channel parameters in linearly modulated communication systems, the iterative expectation-maximization (EM) algorithm can be used to exploit the signal energy associated with the unknown data symbols. It turns out that the channel estimation requires at each EM iteration the *a posteriori* probabilities (APPs) of these data symbols, resulting in a high computational complexity when channel coding is present. In this paper, we present a new approximation of the APPs of trellis-coded symbols, which is less complex and requires less memory than alternatives from literature. By means of computer simulations, we show that the Viterbi decoder that uses the EM channel estimate resulting from this APP approximation experiences a negligible degradation in frame error rate (FER) performance, as compared to using the exact APPs in the channel estimation process.

Keywords: *A posteriori* symbol probability; ML estimation; EM algorithm; Trellis-coded modulation; Viterbi decoder

1 Introduction

When the channel between a source and destination node is not known, it is primordial for the destination to estimate this channel in order to decode the transmitted information. Typically, the source assists the destination with this task by transmitting known pilot symbols along with the unknown data symbols. Making use of only these pilot symbols, the destination is able to estimate the channel. The drawback of this pilot-aided method is that the channel information contained in the data part of the signal is not harvested during the estimation. Hence, in order to obtain an accurate channel estimate, a large number of pilot symbols should be present, yielding a substantial reduction of both power and bandwidth efficiency.

To accommodate these problems, the iterative expectation-maximization (EM) algorithm [1,2] can be used to also exploit the signal energy associated with the unknown data symbols during the channel estimation; this way, much less pilot symbols are needed to achieve a given estimation accuracy. Application of the EM algorithm requires that at each iteration the *a posteriori* probabilities (APPs) of these data symbols be calculated. When

using a trellis code to map the information bits on the data symbols, the Viterbi algorithm [3] minimizes the frame error rate (FER) by performing maximum likelihood (ML) sequence detection. The exact APPs of the trellis-coded data symbols are obtained by means of the Bahl-Cocke-Jelinek-Raviv (BCJR) algorithm [4], which however is roughly three times as complex as the Viterbi algorithm [5].

Several low-complexity approximations of the BCJR algorithm have been proposed in the literature, mainly in the context of iterative soft decoding of concatenated codes, referred to as turbo decoding. Among them are the max-log maximum *a posteriori* probability (MAP) algorithm [5] and the soft-output Viterbi algorithm (SOVA) [6], which are roughly twice as complex as the Viterbi algorithm [7], and the soft-output M-algorithm (SOMA) [8], which reduces complexity by considering only the M most likely states at each trellis section. Some improvements of the SOVA algorithm in the context of turbo decoding have been presented in [9-13]. Whereas these referenced papers make use of the approximate APPs inside the iterative decoder, we focus on using the approximate APPs only in the iterative estimation algorithm and use the standard Viterbi algorithm (which does not need symbol APPs) for decoding. Because an accurate approximation

*Correspondence: nico.aerts@telin.ugent.be
TELIN, UGent, Sint-Pietersnieuwstraat 42, Gent 9000, Belgium

of the true APPs is less critical for the proper operation of the EM algorithm, we propose a simpler approximation of the APP computation with roughly half the complexity of max-log MAP and with substantially less memory requirements. We compare the resulting EM algorithm in terms of estimation accuracy and FER of the Viterbi decoder, with the cases where the EM estimator uses either the true APPs, or the APPs resulting from SOMA, or the APPs that are computed under the simplifying assumption of uncoded transmission.

1.1 Notations

All vectors are row vectors and in boldface. The Hermitian transpose, statistical expectation, the m th element, the first m elements, and estimate of the row vector \mathbf{x} are denoted by \mathbf{x}^H , $E[\mathbf{x}]$, x_m , $\mathbf{x}_{1:m}$, and $\hat{\mathbf{x}}$, respectively.

2 System description

A source transmits to the destination a frame consisting of a sequence of K_p pilot symbols \mathbf{c}_p and a sequence of K data symbols \mathbf{c} ; the latter is obtained by applying K_b information bits \mathbf{b} to a trellis encoder [5]. We assume that the pilot symbols have a constant magnitude $\sqrt{E_s}$ and that the data symbols satisfy $E[|c_m|^2] = E_s$, with E_s denoting the average symbol energy at the transmitter. Considering a channel that is characterized by a channel gain h that is constant over the frame and a noise contribution \mathbf{n} , the signal received by the destination $\mathbf{r}_t = (\mathbf{r}_p, \mathbf{r})$ is defined by

$$\mathbf{r}_p = h\mathbf{c}_p + \mathbf{n}_p, \quad (1)$$

$$\mathbf{r} = h\mathbf{c} + \mathbf{n}. \quad (2)$$

The elements of \mathbf{n}_p and \mathbf{n} are independent zero mean circular symmetric complex Gaussian random variables with variance N_0 . The destination produces a channel gain estimate \hat{h} and uses a Viterbi decoder to obtain the ML information bit sequence decision $\hat{\mathbf{b}} = \arg \max_{\tilde{\mathbf{b}}} p(\mathbf{r}|\mathbf{b} = \tilde{\mathbf{b}}, h = \hat{h})$ with $\tilde{\mathbf{b}}$ belonging to the set of all 2^{K_b} information bit sequences of length K_b . If the estimate \hat{h} were equal to the actual channel gain h , the Viterbi decoder would minimize the FER given by $\Pr[\hat{\mathbf{b}} \neq \mathbf{b}]$.

As the operation of the Viterbi decoder is well documented in the literature [3], we only briefly recall its main features. The decoder makes use of the fact that the trellis encoder can be described as a finite-state machine. At the start of the m th symbol interval, the encoder accepts a vector \mathbf{u}_m of N_b information bits (the information bit sequence \mathbf{b} is partitioned as $\mathbf{b} = (\mathbf{u}_1, \dots, \mathbf{u}_K)$) and outputs a coded data symbol c_m which is given by the output equation $c_m = g(S_m, \mathbf{u}_m)$; here S_m denotes the encoder state at the start of the m th symbol interval. At the end of the m th symbol interval, the encoder has reached the state S_{m+1} given by the state equation $S_{m+1} = f(S_m, \mathbf{u}_m)$.

For any m , S_m belongs to the set $\Sigma = \{\sigma_1, \dots, \sigma_L\}$, with L denoting the number of encoder states. The resulting trellis code has a rate of N_b information bits per coded symbol. The Viterbi algorithm recursively computes the sequences $\hat{\mathbf{c}}_{1:m}(\sigma_l)$ and their log-likelihood for all $\sigma_l \in \Sigma$, and $m = 1, \dots, K$; here, $\hat{\mathbf{c}}_{1:m}(\sigma_l)$ is the ML sequence consisting of m data symbols that yields $S_{m+1} = \sigma_l$. The log-likelihood of $\hat{\mathbf{c}}_{1:m}(\sigma_l)$ is given by $-\Lambda_m(\sigma_l)/N_0$, where the path metric $\Lambda_m(\sigma_l)$ satisfies the following recursion:

$$\Lambda_m(\sigma_l) = \min_{\sigma_j, \tilde{c}} (\Lambda_{m-1}(\sigma_j) + \lambda_m(\tilde{c})) \quad (3)$$

and $(S_{m+1}, S_m, c_m) = (\sigma_l, \sigma_j, \tilde{c})$ must be consistent with the encoder operation (i.e., a bit vector $\tilde{\mathbf{u}}$, satisfying both $\tilde{c} = g(\sigma_j, \tilde{\mathbf{u}})$ and $\sigma_l = f(\sigma_j, \tilde{\mathbf{u}})$, must exist). The quantity $\lambda_m(\tilde{c})$ denotes the branch metric corresponding to $c_m = \tilde{c}$, which is given by

$$\lambda_m(\tilde{c}) = |r_m - \hat{h}\tilde{c}|^2. \quad (4)$$

The value of \tilde{c} that results from (3) is $\hat{c}_m(\sigma_l)$, the last element of $\hat{\mathbf{c}}_{1:m}(\sigma_l)$. The recursion starts from $\Lambda_0(\sigma_l)$, which is determined by the *a priori* distribution of the initial state S_0 . The ML data sequence decision is given by $\hat{\mathbf{c}} = \hat{\mathbf{c}}_{1:K}(\hat{S}_{K+1})$, where $\hat{S}_{K+1} = \arg \min_{\sigma_l \in \Sigma} \Lambda_K(\sigma_l)$. The ML decision $\hat{\mathbf{b}}$ is the information bit sequence consistent with $\hat{\mathbf{c}}$, \hat{S}_{K+1} , and the encoder operation. The Viterbi decoder operation requires the storage of L data symbol sequences of length K . The above is straightforwardly extended to (i) multidimensional trellis coding, where a transition from state S_m to state S_{m+1} gives rise to multiple data symbols, and (ii) the presence of termination bits at the encoder input to impose a known final state S_{K+1} .

3 Estimation strategy

As mentioned before, the source transmits pilot symbols \mathbf{c}_p to assist the destination with the channel estimation process. The ML pilot-based estimate of h that uses only \mathbf{r}_p is given by [14]

$$\hat{h} = \arg \max_{\tilde{h}} p(\mathbf{r}_p|\tilde{h}) = \frac{\mathbf{r}_p \mathbf{c}_p^H}{K_p E_s}. \quad (5)$$

The estimate (5) gives rise to a mean-squared estimation error (MSE) that is equal to

$$E[|\hat{h} - h|^2] = \frac{N_0}{K_p E_s}. \quad (6)$$

Hence, for given E_s , the estimation accuracy is improved by increasing K_p .

When exploiting also the data part of the received signal, i.e., \mathbf{r} , the ML estimate of h is given by

$$\hat{h} = \arg \max_{\tilde{h}} p(\mathbf{r}_t | \tilde{h}) \quad (7)$$

$$= \arg \max_{\tilde{h}} \sum_c p(\mathbf{r}_t | c, \tilde{h}) \Pr[c]. \quad (8)$$

Because of the summation over all possible 2^{K_b} data sequences, obtaining from (8) the ML estimate for large K_b is computationally complex. Fortunately, the EM algorithm [1] allows to compute the ML estimate iteratively. For the problem at hand, the EM channel estimate $\hat{h}^{(i)}$ during the i th iteration is obtained as

$$\hat{h}^{(i)} = \frac{\mathbf{r}_p \mathbf{c}_p^H + \mathbf{r} (\boldsymbol{\mu}_m^{(i)})^H}{K_p E_s + \sum_{m=1}^K E_m^{(i)}}, \quad (9)$$

where $\boldsymbol{\mu}_m^{(i)}$ and $E_m^{(i)}$ denote the *a posteriori* mean of c_m and $|c_m|^2$, respectively:

$$\boldsymbol{\mu}_m^{(i)} = \sum_{\alpha} \alpha \Pr[c_m = \alpha | \mathbf{r}, \hat{h}^{(i-1)}], \quad (10)$$

$$E_m^{(i)} = \sum_{\alpha} |\alpha|^2 \Pr[c_m = \alpha | \mathbf{r}, \hat{h}^{(i-1)}]. \quad (11)$$

When the data symbols have a constant magnitude, the numerator of (9) reduces to $(K_p + K)E_s$. The iterations are initialized with the pilot-based estimate from (5), which we denote as $\hat{h}^{(0)}$.

The APP $\Pr[c_m = \alpha | \mathbf{r}, \hat{h}]$ (for notational convenience, we drop the iteration index) of c_m can be expressed as

$$\Pr[c_m = \alpha | \mathbf{r}, \hat{h}] \propto \sum_{c, c_m = \alpha} p(\mathbf{r} | c, \hat{h}) \Pr[c], \quad (12)$$

where \propto means equal within a normalization factor, and the summation is over all valid codewords with c_m equal to α . Making use of the finite-state description of the encoder, the APPs (12) can be computed efficiently for $m = 1, \dots, K$ by means of the BCJR algorithm [4]. However, its complexity is still about three times that of the Viterbi algorithm [5]. Hence, assuming that the EM algorithm converges after I iterations, the BCJR algorithm must be applied I times, after which, the Viterbi algorithm (with $\hat{h} = \hat{h}^{(I)}$) is used to detect the information bit sequence. The resulting complexity is $3I + 1$ times that of a single use of the Viterbi decoder.

The MSE resulting from (8) cannot be obtained in closed form. Therefore, we resort to the modified Cramer-Rao bound (MCRB) [15], which is a fundamental lower bound on the MSE performance of any unbiased estimate. For the observation model (1)-(2), the MCRB reduces to

$$\mathbb{E} [|\hat{h} - h|^2] \geq \frac{N_0}{(K + K_p) E_s} \quad (13)$$

Comparison of (6) with (13) indicates the possibility of substantially reducing the MSE when also including the data portion \mathbf{r} of the observation in the estimation process, especially when $K \gg K_p$.

4 Complexity reduction

In order to avoid the computational complexity associated with the BCJR algorithm (or the max-log MAP or SOMA approximations), we consider two reduced-complexity approximations for computing the APPs. In the first algorithm (A1), we do not exploit the code properties and compute the APPs as if the transmission was uncoded. The second algorithm (A2), which exploits the path metrics from the Viterbi decoder to approximate the APPs, represents our main contribution.

4.1 Algorithm A1

A trivial low-complexity approximation is obtained by simply ignoring the code constraint during the EM iterations. More specifically, it is assumed that the data symbols are statistically independent, which implies that the received samples r_k with $k = 1, \dots, K$ and $k \neq m$ do not contain any information about c_m . In (12), we can thus replace \mathbf{r} by \mathbf{r}_m . This yields

$$\Pr[c_m = \alpha | \mathbf{r}, \hat{h}] = \Pr[c_m = \alpha | \mathbf{r}_m, \hat{h}] \quad (14)$$

$$\propto \exp\left(\frac{-\lambda_m(\alpha)}{N_0}\right), \quad (15)$$

where $\lambda_m(\alpha)$ follows from (4).

4.2 Algorithm A2

Here, we present a new low-complexity approximation to the APP computation, that makes use only of the Viterbi decoder metrics $\{\Lambda_m(\sigma_l)\}$ of the surviving paths. The APP approximation makes use of the following simplifications:

- (i) We ignore future observations. More specifically, we approximate the APP $\Pr[c_m = \alpha | \mathbf{r}, \hat{h}]$ of a symbol c_m by conditioning on only the past and present observations $\mathbf{r}_{1:m}$. This APP is obtained by simply replacing in the right-hand side of (12) the vectors \mathbf{r} and \mathbf{c} by $\mathbf{r}_{1:m}$ and $\mathbf{c}_{1:m}$, respectively:

$$\Pr[c_m = \alpha | \mathbf{r}, \hat{h}] \approx \Pr[c_m = \alpha | \mathbf{r}_{1:m}, \hat{h}] \quad (16)$$

$$\propto \sum_{c_{1:m}, c_m = \alpha} p(\mathbf{r}_{1:m} | c_{1:m}, \hat{h}) \Pr[c] \quad (17)$$

- (ii) From all paths yielding $S_{m+1} = \sigma_l$ ($l = 1, 2, \dots, L$), we only keep the most likely path that corresponds to the symbol sequence $\hat{\mathbf{c}}_{1:m}(\sigma_l)$; the likelihood of the

other, non-surviving, paths is assumed to be zero.
This yields the approximation

$$\Pr[c_m = \alpha | \mathbf{r}, \hat{h}] \propto \sum_{\substack{\sigma_l \in \Sigma \\ \hat{c}_m(\sigma_l) = \alpha}} p(\mathbf{r}_{1:m} | \hat{\mathbf{c}}_{1:m}(\sigma_l), \hat{h}) \Pr[c] \quad (18)$$

- (iii) We replace in (18) the summation over the valid symbol sequences by a maximization and finally obtain the approximation

$$\Pr[c_m = \alpha | \mathbf{r}, \hat{h}] \propto \max_{\substack{\sigma_l \in \Sigma \\ \hat{c}_m(\sigma_l) = \alpha}} p(\mathbf{r}_{1:m} | \hat{\mathbf{c}}_{1:m}(\sigma_l), \hat{h}) \quad (19)$$

Hence, from all surviving sequences $\hat{\mathbf{c}}_{1:m}(\sigma_l)$ with $\hat{c}_m(\sigma_l) = \alpha$, only the sequence with the highest likelihood contributes to the APP approximation (19). Taking into account the relation between the Viterbi decoder path metric $\Lambda_m(\sigma_l)$ from (3) and the likelihood of $\hat{\mathbf{c}}_{1:m}(\sigma_l)$, we obtain from (19) the approximation

$$\Pr[c_m = \alpha | \mathbf{r}, \hat{h}] \propto \exp(-\Lambda_{m,\min}(\sigma_l)/N_0), \quad (20)$$

with

$$\Lambda_{m,\min}(\sigma_l) = \min_{\substack{\sigma_l \in \Sigma \\ \hat{c}_m(\sigma_l) = \alpha}} \Lambda_m(\sigma_l) \quad (21)$$

Hence, when the surviving path with the largest likelihood at the end of the m th trellis section has $c_m = \alpha$, our APP approximation for the symbol c_m is largest for $c_m = \alpha$. Approximating the APPs

$\Pr[c_m = \alpha | \mathbf{r}, \hat{h}]$ for $m = 1, \dots, K$ using (21) yields a complexity similar to that of the Viterbi algorithm. Hence, assuming that the EM algorithm converges

after I iterations, the complexity as compared to a single use of the Viterbi decoder is $I + 1$ times for algorithm A2, whereas it is $3I + 1$ times when the APP computation is according to the BCJR algorithm. Note that unlike the Viterbi algorithm, the computation of the APP (21) of c_m does not require to store the data symbol decisions $\hat{c}_n(\sigma_l)$ for $n < m$, so that algorithm A2 uses considerably less memory than the Viterbi algorithm does.

Whereas simplifications similar to (ii) and (iii) have also been applied to APP algorithms from literature (e.g., max-log MAP), this is not the case for simplification (i). As the APP algorithms from the literature also make use of future observations, the APP of c_m requires updating each time future observations r_{m+1}, r_{m+2}, \dots become available, yielding a higher computational complexity and more memory requirements. Hence, approximation (i) is crucial for obtaining a very simple APP computation.

5 Numerical results

We consider a trellis encoder consisting of an eight-state rate $1/2$ (15,17)₈ convolutional encoder with known initial and final states, followed by Gray mapping of the convolutional encoder output bits to 4-QAM symbols. Each frame contains $K_p = 5$ pilot symbols and $K = 200$ data symbols (including four termination symbols). We consider both an Additive white Gaussian noise (AWGN) channel with $h = 1$ and a Rayleigh fading channel with $\mathbb{E}[|h|^2] = 1$. We investigate the performance of the estimator and the Viterbi decoder by means of Monte-Carlo simulations, in terms of MSE and FER, respectively. The EM algorithm has essentially converged after only one iteration, i.e., $I = 1$; for $I = 1$, the complexity reduction obtained by computing the APPs using the new algorithm A2 instead of the BCJR algorithm is about a factor of 2. In the following, we consider the APP computation according to

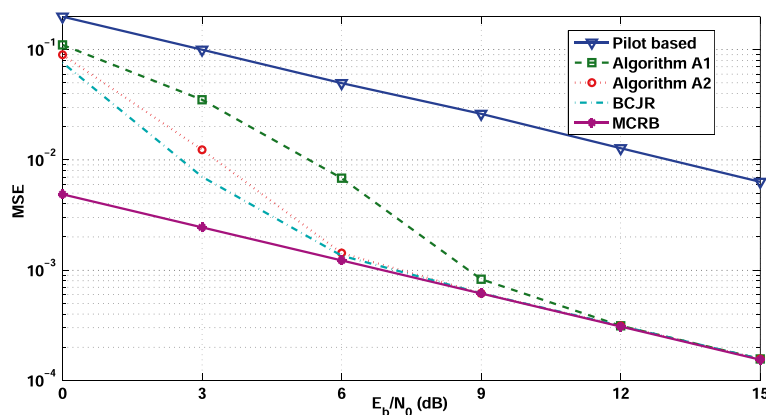


Figure 1 MSE for a (15,17)₈ convolutional code: AWGN channel.

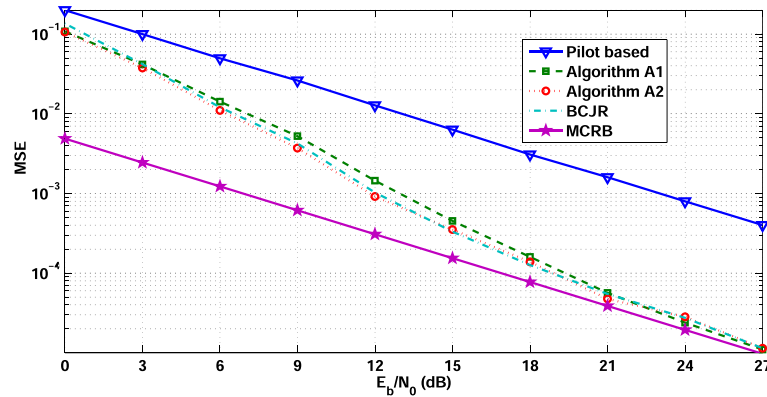


Figure 2 MSE for a (15,17)₈ convolutional code: Rayleigh channel.

the BCJR algorithm, the SOMA ($M = 4$) version of the exact APP algorithm from [16] and the above A1 and A2 algorithms.

The MSE performance for the AWGN channel and the Rayleigh fading channel is depicted in Figures 1 and 2, where the MCRB (13) is used as benchmark. The MSE performance of the EM-based channel estimators (a) converges to the MCRB with increasing signal-to-noise ratio (SNR) and (b) outperforms the system where only pilot symbols are used to estimate h . Ranking the APP algorithms according to the resulting MSE, we see that BCJR (which computes exact APPs) performs best, SOMA (which takes past, present, and future observations into account) is a very close second, and A2 (which ignores future observations) is only slightly worse than BCJR and SOMA; A1 (which uses only the present observation) is considerably worse than BCJR, SOMA, and A2 for low and medium SNRs.

Figures 3 and 4 show the FER of the Viterbi decoder for the AWGN channel and the Rayleigh fading channel

as a function of E_b/N_0 , with E_b denoting the energy per information bit. Hence,

$$\frac{E_s}{E_b} = N_b \cdot \frac{K - K_{\text{ter}}}{K + K_p}, \quad (22)$$

where K_{ter} is the number of termination symbols. As benchmark, we use the FER of a reference system with $(K, K_p) = (200, 0)$ where the channel coefficient h is known to the receiver. Hence, as compared to this reference system, the system with $(K, K_p) = (200, 5)$ suffers from an irreducible power efficiency loss of $10 \log_{10} \frac{205}{200} = 0.11$ dB because of the presence of pilot symbols; the actual degradation will exceed 0.11 dB because of channel estimation errors. We observe that (a) the A2, SOMA, and BCJR algorithms yield essentially the same FER performance and require, for given FER, about 0.11 dB more E_b/N_0 than the reference system: for these algorithms, the channel estimation is sufficiently accurate so that the degradation is mainly determined by the power efficiency

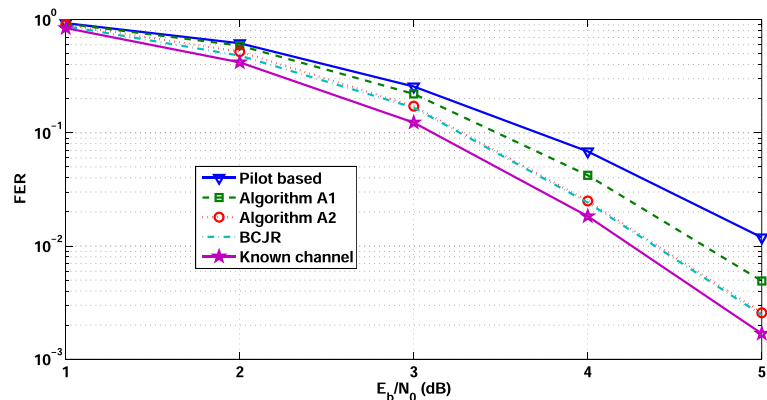


Figure 3 FER for a (15,17)₈ convolutional code: AWGN channel.

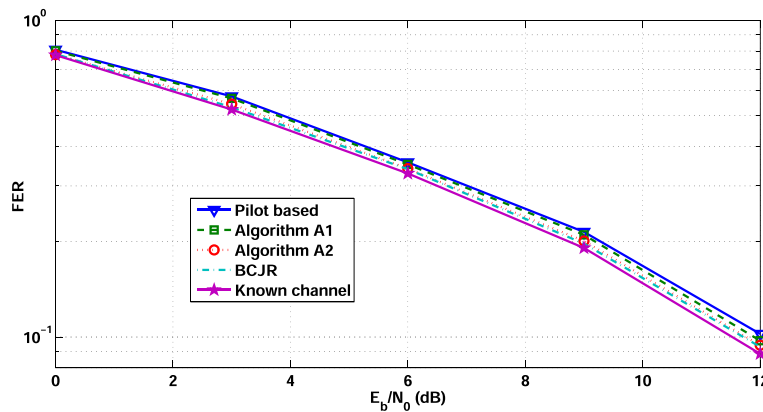


Figure 4 FER for a (15,17)₈ convolutional code: Rayleigh channel.

loss caused by the pilot symbol insertion. (b) The A1 algorithm performs worse than the A2, SOMA, and BCJR algorithms because ignoring the code constraints when computing the APPs yields less accurate channel estimates. (c) The FER performance is worst when only pilot symbols are used to estimate h . Hence, from a computational complexity and memory requirement point of view, it is advantageous to compute the trellis-coded symbol APPs in the EM algorithm by means of the new algorithm A2 (that ignores future observations) rather than the considered APP algorithms from the literature (that take also future observations into account).

6 Conclusions

EM-based channel estimation in the presence of trellis-coded modulation requires the use of the BCJR algorithm to efficiently compute the exact symbol APPs. As the computational complexity of the BCJR algorithm is about three times that of the Viterbi algorithm which we use for decoding, we have proposed a new approximation to the APP computation that beats the main APP algorithms from the literature in terms of computational complexity and memory requirements. By means of computer simulations, we have pointed out that when using the new APP computation instead of the exact APPs from the BCJR algorithm, the resulting Viterbi decoder FER performances are essentially the same. Hence, this motivates the use of the new APP approximation in the context of EM channel estimation for trellis-coded modulation.

So far, we have limited our attention to a time-invariant channel, i.e., the channel gain takes a constant value h during a frame. For time-varying channels, a similar approximation of the symbol APPs can be derived. Denoting by h_m the channel gain associated with the data symbol c_m , the branch metric corresponding to $c_m = \tilde{c}$ becomes

$$\lambda_m(\tilde{c}) = |r_m - \hat{h}_m \tilde{c}|^2, \quad (23)$$

Note that (23) is obtained by replacing in (4) \hat{h} by \hat{h}_m , which denotes the estimate of h_m . Collecting the channel gain estimates at the data symbol positions into the vector $\hat{\mathbf{h}}$, the likelihood $p(\mathbf{r}|\mathbf{c}, \hat{\mathbf{h}})$ can be decomposed as

$$p(\mathbf{r}|\mathbf{c}, \hat{\mathbf{h}}) = \prod_{m=1}^K p(r_m|c_m, \hat{h}_m), \quad (24)$$

where $p(r_m|c_m, \hat{h}_m) \propto \exp(-\lambda_m(c_m)/N_0)$, with $\lambda_m(\tilde{c})$ given by (23). Hence, the symbol APP approximations for time-varying channels are simply obtained by replacing in the APP approximations A_1 and A_2 for time-invariant channels the quantity $\lambda_m(\tilde{c})$ by the right-hand side of (23) instead of (4). The resulting approximated symbol APPs can be used in, for instance, a MAP EM channel estimation algorithm that exploits the correlation between the time-varying channel gains [17].

Competing interests

The authors declare that they have no competing interests.

Acknowledgements

This research has been funded by the Interuniversity Attraction Poles Programme initiated by the Belgian Science Policy Office and is also supported by the FWO project G.0022.11 'Advanced multi-antenna systems, multi-band propagation models and MIMO/cooperative receiver algorithms for high data rate V2X communications'.

Received: 21 May 2014 Accepted: 2 September 2014

Published: 17 September 2014

References

1. Dempster, N Laird, DB Rubin, Maximum likelihood from incomplete data via the EM algorithm. *J. Roy. Stat. Soc.* **39**(1), 1–38 (1977)
2. J Choi, *Adaptive and Iterative Signal Processing in Communications*. (Cambridge University Press, New York, 2006)
3. AJ Viterbi, Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Trans. Inform. Theor.* **13**, 260–269 (1967)
4. L Bahl, J Cocke, F Jelinek, J Raviv, Optimal decoding of linear codes for minimizing symbol error rate. *IEEE Trans. Inform. Theor.* **IT-20**(2), 284–287 (1974)

5. S Lin, D Costello, *Error Control Coding*, 2nd edn. (Pearson Education Inc., Upper Saddle River, 2004)
6. J Hagenauer, A Viterbi algorithm with soft-decision output and its applications. *GLOBECOM*. **3**, 1680–1686 (1989)
7. M Mansour, A Bahai, Complexity based design for iterative joint equalization and decoding. *Proc. IEEE VTC*. **4**, 1699–1704 (2002)
8. R Mehlman, H Meyr, Soft output M-algorithm equalizer and trellis-coded modulation for mobile radio communication. *Proc. IEEE VTC*. **2**, 586–591 (1992)
9. S Papaharalabos, P Sweeney, BG Evans, PT Mathiopoulos, Improved performance SOVA turbo decoder. *IEE Proc. Commun.* **153**(5), 586–590 (2006)
10. AA Hamad, Performance enhancement of SOVA based decoder in SCCC and PCCC schemes. *SciRes. Wireless Engineering and Technology*. **4**, 40–45 (2013)
11. R Motwani, T Souvignier, Reduced-complexity soft-output Viterbi algorithm for channels characterized by dominant error events, in *IEEE Global Telecommunications Conference* (Miami, FL, USA, 6–10 Dec 2010), pp. 1–5
12. DW Yue, HH Nguyen, Unified scaling factor approach for turbo decoding algorithms. *IET Commun.* **4**(8), 905–914 (2010)
13. CX Huang, A Ghayeb, A simple remedy for the exaggerated extrinsic information produced by SOVA algorithm. *IEEE Trans. Wireless Commun.* **5**(5), 996–1002 (2006)
14. JG Proakis, *Digital Communications*, (McGraw-Hill, New York, 1995)
15. AD Andrea, U Mengali, R Reggiannini, The modified Cramer-Rao bound and its applications to synchronization problems. *IEEE Trans. Commun.* **42**(234), 1391–1399 (1994)
16. L Lee, Real-time minimum-bit-error probability decoding of convolutional codes. *IEEE Trans. Commun.* **COM-22**, 146–151 (1974)
17. N Aerts, M Moeneclaey, SAGE-based estimation algorithms for time-varying channels in amplify-and-forward cooperative networks, in *International Symposium on Information Theory and its Applications (ISITA)* (Honolulu, HI, USA, 28–31 Oct 2012), pp. 672–676

doi:10.1186/1687-1499-2014-153

Cite this article as: Aerts et al.: Low-complexity *a posteriori* probability approximation in EM-based channel estimation for trellis-coded systems. *EURASIP Journal on Wireless Communications and Networking* 2014 **2014**:153.

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com