

A FIXED-LAG PARTICLE SMOOTHING ALGORITHM FOR THE BLIND TURBO EQUALIZATION OF TIME-VARYING CHANNELS

Alberto Gaspar Guimarães*

Instituto Militar de Engenharia
Rio de Janeiro, Brazil
agaspar@ime.eb.br

Boujemaa Ait El Fquih, François Desbouvries

GET/INT/CITI and CNRS UMR 5157,
91011 Evry, France
{Boujemaa.Ait_elfquih,Francois.Desbouvries} @int-evry.fr

ABSTRACT

We introduce a novel sequential importance sampling (SIS) algorithm for the blind equalization of doubly selective channels. Our algorithm propagates a Monte Carlo (MC) approximation of the posterior fixed-lag smoothing distribution of the symbols. As we shall see, it is possible to sample particles from the *optimal* importance distribution and to update the smoothing importance weights accordingly. We next apply the developed method as a SISO (Soft Input Soft Output) equalizer in a turbo receiver framework. The performance evaluation of our algorithm is carried out under different fading scenarios, and the results are compared with a soft iterative channel estimation scheme available in the literature.

Index Terms— Blind Equalization, Turbo Equalization, Sequential Importance Sampling Methods.

1. INTRODUCTION

Equalizers aim at reducing the effect of intersymbol interference (ISI) in frequency selective channels. *Blind* equalizers do not use training data and thus avoid significant overheads and bandwidth-inefficient communication. A number of blind equalizers have been proposed for the case of time invariant channels. Some algorithms estimate the channel coefficients via EM, and are based either on the BCJR [1] or the Viterbi [2] algorithm. SIS algorithms (either filters [3] or fixed-lag smoothers [4]) have also been proposed. For time-varying channels, a blind equalization technique based on EM and the Viterbi algorithm has been proposed [5]; solutions based on the per survivor processing (PSP) method [6] or on particle filtering [7] are also available.

In this paper we propose a blind, fixed-lag, smoothing based equalizer for the case of time-varying channels. An important feature of our SIS algorithm is that one can draw the N particles from the optimal importance function. Since the target distribution (the posterior transmitted symbols distribution) is discrete and the channel response is analytically

marginalized out, only a few number of particles is required to attain good performance. Moreover, the fixed-lag is a parameter which can be adjusted to find a suitable trade-off between precision and complexity. By contrast, other methods such as EM or MCMC techniques process the entire batch of data, and thus do not present this flexibility.

Next, our algorithm can naturally be used as a SISO equalizer embedded in a turbo equalization setup. Turbo equalization systems were first proposed in [8] and further developed by several researchers (see e.g. [9] and references therein). They consist in an iterative receiver algorithm in which the equalizer and decoder exchange extrinsic soft information about the same batch of received data, until they achieve near-optimal bit error rate (BER) performance if the channel is known. Our method is applied to the scenario where the channel is both unknown and doubly-selective, performing channel estimation jointly with equalization.

2. STATE-SPACE MODEL

The binary data sequence is encoded with a Forward Error Correction (FEC) code and randomly interleaved before being mapped to a complex symbol x_n which belongs to a finite alphabet of size Q . The resulting data sequence is partitioned into blocks (frames) of length P and transmitted over a channel which introduces additive noise and ISI. So the receiver input y_n is related to the transmitted symbols via $y_n = \mathbf{x}_n^T \mathbf{h}_n + \omega_n$, where T (and later on H) stands for transpose (resp. Hermitian transpose), $\mathbf{x}_n = [x_n, \dots, x_{n-L+1}]^T$, and $\mathbf{h}_n = [h_{0,n}, \dots, h_{L-1,n}]^T$ is the baseband channel impulse response, assumed to be of finite length L . Under the turbo equalization setup, we assume that x_n ($n = 1, \dots, P$) are mutually independent and their probability mass function (pmf) $p(x_n)$ are known to the equalizer. ω_n are zero-mean circularly symmetric complex Gaussian variables with known variance Λ_n^ω , independent and independent of $\{x_n\}$. Finally

$$\mathbf{h}_{n+1} = \mathbf{F}_n \mathbf{h}_n + \mathbf{v}_n \quad (1)$$

where \mathbf{h}_0 and \mathbf{v}_n are Gaussian distributed with parameters $(\mathbf{0}, \Lambda_0^h)$ and $(\mathbf{0}, \Lambda_n^v)$, respectively, and \mathbf{v}_n are independent

*This work was performed during Dr Guimarães's sabbatical leave at INT, supported by CAPES-Brazil, under grant BEX 1203/06-1.

and independent of $\{x_n\}$, $\{\omega_n\}$ and \mathbf{h}_0 .

3. A FIXED-LAG SIS SMOOTHING ALGORITHM

In this section we focus on the computation of a SIS approximation of $p(x_n|y_{1:n+M})$ where $M > 0$ is some fixed delay. Under the turbo equalization setup, we would then be able to forward to the decoder the (extrinsic) soft information about the symbols, or equivalently about the coded bits. So let us assume that the a posteriori joint pmf of symbols $x_{1:n-1}$ is approximated by

$$p(x_{1:n-1}|y_{1:n+M-1}) \approx \sum_{i=1}^N \lambda_{n-1}^i \delta(x_{1:n-1} - x_{1:n-1}^i), \quad (2)$$

where the samples $x_{1:n-1}^i$ are generated from an importance distribution $q(x_{1:n-1}|y_{1:n+M-1})$, and the importance weight λ_{n-1}^i associated to the i -th trajectory $x_{1:n-1}^i$ is given by

$$\lambda_{n-1}^i \propto \frac{p(x_{1:n-1}^i|y_{1:n+M-1})}{q(x_{1:n-1}^i|y_{1:n+M-1})}, \quad \sum_{i=1}^N \lambda_{n-1}^i = 1. \quad (3)$$

Let us see how to propagate (2) recursively. If we assume that $q(x_{1:n}|y_{1:n+M})$ factorizes as $q(x_n|x_{1:n-1}, y_{1:n+M}) \times q(x_{1:n-1}|y_{1:n+M-1})$, then for all i $[x_{1:n}^i] = [x_{1:n-1}^i, x_n^i]$, in which each particle x_n^i is drawn from $q(x_n|x_{1:n-1}^i, y_{1:n+M})$. As for the weights λ_{n-1}^i , we observe that $p(x_{1:n}|y_{1:n+M})$ factorizes as $p(x_{1:n}|y_{1:n+M}) = \bar{\lambda}_n^i p(x_{1:n-1}|y_{1:n+M-1})$ with

$$\bar{\lambda}_n^i = \frac{p(x_n|x_{1:n-1}, y_{1:n+M})p(y_{n+M}|x_{1:n-1}, y_{1:n+M-1})}{p(y_{n+M}|y_{1:n+M-1})}.$$

So the weights can be computed recursively as

$$\lambda_n^i \propto \frac{p(x_n^i|x_{1:n-1}^i, y_{1:n+M})p(y_{n+M}|x_{1:n-1}^i, y_{1:n+M-1})}{q(x_n^i|x_{1:n-1}^i, y_{1:n+M})} \times \underbrace{\frac{p(x_{1:n-1}^i|y_{1:n+M-1})}{q(x_{1:n-1}^i|y_{1:n+M-1})}}_{\propto \lambda_{n-1}^i}. \quad (4)$$

Finally $\sum_{i=1}^N \lambda_n^i \delta(x_{1:n} - x_{1:n}^i)$ approximates $p(x_{1:n}|y_{1:n+M})$, and thus $\sum_{i=1}^N \lambda_n^i \delta(x_n - x_n^i)$ approximates $p(x_n|y_{1:n+M})$.

Now, like in all SIS algorithms, it is important to resample from $\sum_{i=1}^N \lambda_n^i \delta(x_n - x_n^i)$ (either systematically or according to some strategy) and also to choose $q(x_n|x_{1:n-1}, y_{1:n+M})$ carefully. To that respect, the best choice is to sample the particles from the optimal conditional importance distribution [10]. In our model, the optimal distribution reads

$$q^{opt}(x_n|x_{1:n-1}, y_{1:n+M}) = p(x_n|x_{1:n-1}, y_{1:n+M}), \quad (5)$$

and under that choice (4) becomes

$$\lambda_n^i \propto \underbrace{p(y_{n+M}|x_{1:n-1}^i, y_{1:n+M-1})}_{\bar{\lambda}_n^i} \lambda_{n-1}^i. \quad (6)$$

Now, sampling from the optimal kernel is often impossible, so many efforts have been expended in order to approximate this distribution [10]. In our case however, it happens that one can compute q^{opt} exactly, and since this distribution is discrete sampling from it is straightforward. From now on, we thus focus on the exact computation of (5) (see §3.1) and of the incremental importance weight $\bar{\lambda}_n^i$ in (6) (see §3.2).

3.1. Computing the optimal importance pmf

Let us address (5). At each time instant n , and for each trajectory i , we should sample a new particle x_n^i according to

$$p(x_n|x_{1:n-1}^i, y_{1:n+M}) = \frac{p(x_n)p(y_{n:n+M}|x_n, x_{1:n-1}^i, y_{1:n-1})}{\sum_{x_n} p(x_n)p(y_{n:n+M}|x_n, x_{1:n-1}^i, y_{1:n-1})}. \quad (7)$$

Pmf $p(x_n)$ is known, so it remains to compute

$$p(y_{n:n+M}|x_n, x_{1:n-1}^i, y_{1:n-1}) = \sum_{x_{n+1}} \dots \sum_{x_{n+M}} \left[\prod_{k=1}^M p(x_{n+k}) \right] p(y_{n:n+M}|\theta^i) \quad (8)$$

with $\theta^i \stackrel{def}{=} (x_{1:n-1}^i, x_{n:n+M}, y_{1:n-1}) \stackrel{def}{=} (\tilde{x}_{1:n+M}, y_{1:n-1})$, i.e. $\tilde{x}_k = x_k^i$ if $k \leq n-1$ and $\tilde{x}_k = x_k$ if $n \leq k \leq n+M$. Let also $\tilde{\mathbf{x}}_n = [\tilde{x}_n, \tilde{x}_{n-1}, \dots, \tilde{x}_{n-L+1}]^T$. One can show that $p(y_{n:n+M}|\theta^i)$ in (8) is Gaussian with parameters which can be computed recursively. Injecting into (7), we finally get

$$p(x_n|x_{1:n-1}^i, y_{1:n+M}) \propto p(x_n) \sum_{x_{n+1}} \dots \sum_{x_{n+M}} \left[\prod_{k=1}^M p(x_{n+k}) \right] \underbrace{p(y_{n:n+M}|\theta^i)}_{\mathcal{N}(y_{n:n+M}; \mu_{M+1}^{y_i}, \Sigma_{M+1}^{y_i})} \quad (9)$$

in which $\mu_{M+1}^{y_i}$ and $\Sigma_{M+1}^{y_i}$ are computed via the following recursions (the proof is omitted for want of space) :

1. *Initialization.* Compute the parameters of $p(\mathbf{h}_n|x_{1:n-1}^i, y_{1:n-1}) = \mathcal{N}(\mathbf{h}_n; \hat{\mathbf{h}}_{n|n-1}^i, \mathbf{\Lambda}_{n|n-1}^i)$ by the Kalman filter, and (μ_1^i, Σ_1^i) as

$$\begin{cases} \mu_1^i = \begin{bmatrix} \tilde{\mathbf{x}}_n^T \\ \mathbf{F}_n \end{bmatrix} \hat{\mathbf{h}}_{n|n-1}^i, \\ \Sigma_1^i = \begin{bmatrix} \tilde{\mathbf{x}}_n^T \\ \mathbf{F}_n \end{bmatrix} \mathbf{\Lambda}_{n|n-1}^i \begin{bmatrix} \tilde{\mathbf{x}}_n^T \\ \mathbf{F}_n \end{bmatrix}^H + \begin{bmatrix} \Lambda_n^\omega & \mathbf{0} \\ \mathbf{0} & \Lambda_n^\nu \end{bmatrix}; \end{cases} \quad (10)$$

2. *Recursion* $(k-1) \rightarrow k$, for all $k = 2, \dots, M$. Compute

$$\begin{cases} \mu_k^i = \mathbf{A}_{k-1}^i \mu_{k-1}^i \\ \Sigma_k^i = \mathbf{A}_{k-1}^i \Sigma_{k-1}^i \mathbf{A}_{k-1}^{iH} \\ \quad + \text{diag}(\mathbf{0}_{k-1}, \Lambda_{n+k-1}^\omega, \mathbf{\Lambda}_{n+k-1}^\nu) \end{cases}, \quad (11)$$

¹ $\mathcal{N}(\alpha; \mu, \Sigma)$ denotes a Gaussian pdf with argument α and parameters (μ, Σ) .

where $\text{diag}(\cdot)$ denotes a block-diagonal matrix, \mathbf{I}_{k-1} the $(k-1) \times (k-1)$ identity matrix, and

$$\mathbf{A}_{k-1}^i = \begin{bmatrix} \mathbf{I}_{k-1} & \mathbf{0} \\ \mathbf{0} & \tilde{\mathbf{x}}_{n+k-1}^T \\ \mathbf{0} & \mathbf{F}_{n+k-1} \end{bmatrix};$$

3. *Final step* $M \rightarrow (M+1)$. Compute

$$\begin{cases} \mu_{M+1}^{yi} = \mathbf{A}_M^{yi} \mu_M^i \\ \Sigma_{M+1}^{yi} = \mathbf{A}_M^{yi} \Sigma_M^i (\mathbf{A}_M^{yi})^H + \text{diag}(\mathbf{0}_M, \Lambda_{n+M}^w) \end{cases} \quad (12)$$

where

$$\mathbf{A}_M^{yi} = \begin{bmatrix} \mathbf{I}_M & \mathbf{0} \\ \mathbf{0} & \tilde{\mathbf{x}}_{n+M}^T \end{bmatrix}.$$

We remark that for each trajectory i , $p(y_{n:n+M}|\theta^i)$ and then (7) is computed by using only one Kalman filter.

3.2. Updating the Importance Weights

We now address the computation of $\tilde{\lambda}_n^i \stackrel{\text{def}}{=} p(y_{n+M}|x_{1:n-1}^i, y_{1:n+M-1})$ in (6). One can see easily that

$$\tilde{\lambda}_n^i = \sum_{x_n} \dots \sum_{x_{n+M}} \left[\prod_{k=0}^M p(x_{n+k}) \right] \int p(y_{n+M}|\mathbf{h}_{n+M}, \tilde{\mathbf{x}}_{n+M}) \times p(\mathbf{h}_{n+M}|x_{n:n+M}, x_{1:n-1}^i, y_{1:n+M-1}) d\mathbf{h}_{n+M}. \quad (13)$$

For a given i and $x_{n:n+M}$, computing the integral in (13) would require the evaluation of the parameters of the Gaussian density $p(\mathbf{h}_{n+M}|x_{n:n+M}, x_{1:n-1}^i, y_{1:n+M-1})$, which are given by a Kalman filter. A direct computation of $\tilde{\lambda}_n^i$ (for all i , with $1 \leq i \leq N$) via (13) would thus require NQ^{M+1} Kalman filters, which is prohibitive in most cases. Fortunately, it happens that the computational burden can be drastically reduced, and in particular that only one Kalman filter is required for a given trajectory $x_{1:n-1}^i$. To see this, let us rewrite $\tilde{\lambda}_n^i$ as

$$\tilde{\lambda}_n^i = \frac{p(y_{n:n+M}|x_{1:n-1}^i, y_{1:n-1})}{p(y_{n:n+M-1}|x_{1:n-1}^i, y_{1:n-1})}. \quad (14)$$

We recognize that the numerator of (14) is equal to the denominator of (7) which has been computed before. Let us now consider the denominator of (14). The term $p(y_{n:n+M-1}|\theta^i)$ is a Gaussian pdf with parameters $(\mu_M^{yi}, \Sigma_M^{yi})$, which have been computed before (at step $k = M$, see (11)); finally

$$\tilde{\lambda}_n^i = \frac{\sum_{x_n} \dots \sum_{x_{n+M}} \left[\prod_{k=0}^M p(x_{n+k}) \right] \mathcal{N}(y_{n:n+M}; \mu_{M+1}^{yi}, \Sigma_{M+1}^{yi})}{\sum_{x_n} \dots \sum_{x_{n+M-1}} \left[\prod_{k=0}^{M-1} p(x_{n+k}) \right] \mathcal{N}(y_{n:n+M-1}; \mu_M^{yi}, \Sigma_M^{yi})}, \quad (15)$$

in which μ_M^{yi} (resp. Σ_M^{yi}) is the first $M \times 1$ subvector of μ_{M+1}^{yi} (resp. the first $M \times M$ submatrix of Σ_{M+1}^{yi}) (see (12)). From a computational point of view, (15) suggests that the quadratic forms in $p(y_{n:n+M}|x_{1:n-1}^i, y_{1:n-1})$ should be computed recursively (along with recursions (11) and (12)) by using Woodbury's lemma.

4. SIMULATION RESULTS

This section aims at evaluating the performance of the blind SISO equalizer proposed in this paper, embedded in a turbo equalization receiver. The following setup is used. A set of 80 data bits is randomly generated and encoded using a 1/2-rate convolutional encoder with generator $(1 + D^2, 1 + D + D^2)$. Next the coded bits are interleaved and mapped to ± 1 symbols (BPSK). The frame with 160 symbols (plus the coder overhead) is then transmitted over a time-varying ISI channel of length 2 and with dynamics given by (1) with parameters $\mathbf{F}_n = \alpha^{1/2} \mathbf{I}_2$, $\Lambda_n^v = 0.5(1 - \alpha) \mathbf{I}_2$ and $\Lambda_n^w = N_0/2$ known at the receiver. Parameter α is set as 0.999 or 0.992. Defining the normalized fading rate f_d from $\int_0^{f_d} S(f) df \approx 0.98 \int_0^\infty S(f) df$, where $S(f)$ is the power spectrum density of the channel coefficients, $\alpha = 0.999$ corresponds to a fading rate of $f_d \approx 10^{-3}$ (slow-fading scenario), and $\alpha = 0.992$ yields $f_d \approx 10^{-2}$ (fast-fading scenario). At the receiver, the BCJR algorithm is used as the SISO decoder. The fixed-lag particle smoothing algorithm is implemented with $M = 3$ and $N = 30$ samples are used to approximate $p(x_n|y_{1:n+M})$. We resample from $\sum_{i=1}^N \lambda_n^i \delta(x_n - x_n^i)$ whenever the effective sample size ($N_{eff} \approx (\sum_{i=1}^N (\lambda_n^i)^2)^{-1}$) falls below $N/3$; this threshold is set empirically.

Our technique is compared with that of [11], which consists in iteratively estimating the channel by using the statistics (mean and variance) of each transmitted symbol, fed back from the decoder in the previous iteration. This turbo scheme with soft input channel estimator (SICE) is implemented under the same conditions as above, and a BCJR algorithm is used as the SISO equalizer. A preamble with 40 training symbols initializes the algorithm.

The BER performance of the turbo receivers (at first and fourth iterations) with either SICE or Fixed-Lag Particle Smoothing (FLPS) equalizers is displayed in Figs. 1(a) and 1(b). The results are averaged over 1000 random channel realizations. For comparison purposes, we also display the performance, after the fourth iteration, of a turbo scheme using a clairvoyant BCJR equalizer which processes the received symbol with perfect knowledge of the channel.

Figure 1(a) displays the performance of our FLPS-based receiver under slow fading ($f_d = 10^{-3}$). The significant gain over the iterations shows the efficiency of our algorithm as a SISO processing block. We observe a 1.5-2 dB gap between the FLPS receiver performance and the upper limit shown by the receiver with clairvoyant BCJR equalizer. Moreover, our

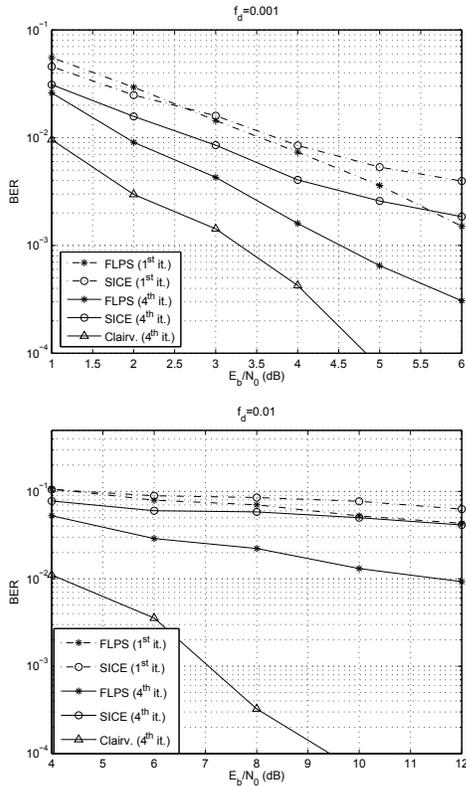


Fig. 1. BER performance of FLPS-based turbo receiver compared with the SICE-based and clairvoyant BCJR-based turbo receivers. (a) $f_d = 10^{-3}$, (b) $f_d = 10^{-2}$.

receiver clearly outperforms that using the SICE technique, mainly for high SNRs. We observe that at $SNR \geq 5$ dB the FLPS receiver achieves in the first iteration the same performance as the SICE receiver after four iterations.

In the fast fading channel ($f_d = 10^{-2}$) case (see Fig. 1(b)) the FLPS and SICE receivers both degrade when compared to the slow fading scenario, and the clairvoyant BCJR equalizer here is far superior. However, the FLPS receiver still compares favorably with the SICE one. We observe that performances of the FLPS receiver improve significantly over iterations, and after four iterations the FLPS receiver BER plot has a noticeably decreasing slope. By contrast, the SICE receiver performances do not really improve when the observations become less noisy, even after some turbo iterations.

5. CONCLUSIONS

In this paper we proposed a novel blind equalization method for doubly-selective channels. Our solution is based on a fixed-lag smoothing of the transmitted symbols using the SIS methodology. We showed that it is possible to sample particles from the *optimal* importance function, and that the time-variant channel response can be marginalized out by using

only one Kalman filter per particle trajectory. Furthermore, only a few number of particles is required in order to achieve good performance results. Computer simulation results were presented in order to validate our technique when it is used as a SISO equalizer embedded in a turbo equalization receiver. Our turbo receiver was shown to outperform another turbo scheme designed for operating under doubly-selective channels, especially under a fast fading scenario.

6. REFERENCES

- [1] G. Kawas-Kaleh and R. Vallet, "Joint parameter estimation and symbol detection for linear or nonlinear unknown channels," *IEEE Trans. Commun.*, vol. 42, no. 7, pp. 2406–2413, July 1994.
- [2] H.N. Nguyen and B.C. Levy, "The Expectation-Maximization Viterbi algorithm for blind adaptive channel equalization," *IEEE Trans. Commun.*, vol. 53, no. 10, pp. 1671–1678, 2005.
- [3] J. Miguez and P. M. Djuric, "Blind equalization of frequency-selective channels by sequential importance sampling," *IEEE Trans. Signal Processing*, vol. 52, no. 10, pp. 2738–2748, Oct. 2004.
- [4] T.C. Clapp and S.J. Godsill, "Fixed-lag smoothing using sequential estimation in digital communications using sequential importance sampling," Oxford, U.K., 1999, vol. 6 of *Bayes. Stat.*, Oxford University Press.
- [5] R. Perry, W.A. Berger, and K. Buckley, "EM algorithm for sequence estimation over Gauss-Markov ISI channels," in *Proc. ICC*, 18 - 22 June 2000, vol. 1, pp. 16–20.
- [6] N. Seshadri, "Joint data and channel estimation using blind trellis search techniques," *IEEE Trans. Commun.*, vol. 42, pp. 1000–1011, Mar. 1994.
- [7] E. Puskaya, C. Andrieu, A. Doucet, and W.J. Fitzgerald, "Particle filtering for demodulation in fading channels with non-gaussian additive noise," *IEEE Trans. Commun.*, vol. 49, no. 4, pp. 579–582, Apr. 2001.
- [8] C. Douillard et al., "Iterative correction of intersymbol interference : turbo-equalization," *Europ. Trans. on Telecomm.*, vol. 6, no. 5, pp. 507–512, Sept-Oct 1995.
- [9] M. Tuchler, R. Koetter, and A. Singer, "Turbo equalization," *IEEE Signal Processing Mag.*, pp. 67–80, 2004.
- [10] A. Doucet et al., "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statistics and Computing (Kluwer)*, vol. 10, pp. 197–208, July 2000.
- [11] S. Song, A. Singer, and K-M. Sung, "Soft input channel estimation for turbo equalization," *IEEE Trans. Signal Processing*, vol. 52, no. 10, pp. 2885–2894, Oct. 2004.