DISTRIBUTED RAO-BLACKWELLIZED POINT MASS FILTER FOR BLIND EQUALIZATION IN RECEIVER NETWORKS

Claudio J. Bordin Jr.

Universidade Federal do ABC
claudio.bordin@ufabc.edu.br

Marcelo G. S. Bruno

Instituto Tecnológico de Aeronáutica
bruno@ita.br

ABSTRACT

We describe new Bayesian algorithms for cooperative blind equalization in a network in which the signal broadcast by a single transmitter is received by multiple remote nodes through distinct frequency-selective channels. The algorithms are based on Rao-Blackwellized point mass filters, and approximate the posterior densities of the unknown channel parameters by Gaussian mixtures of fixed order. To keep computations treatable, density mixtures are reduced by a modified version of West's algorithm. A reduced complexity approach which employs a single mode approximation of some remote quantities is also considered. Via numerical simulations, we verify that the proposed algorithms outperform certain particle-filtering-based algorithms with comparable communication loads.

Index Terms— Blind Equalization, Distributed Algorithms, Bayesian Estimation, Point Mass Filter.

1. INTRODUCTION

Distributed estimation has attracted much interest due to the potential robustness and performance gains provided by those techniques. Most original distributed estimation methods [1, 2] were linear, but recently there has been a surge of interest in nonlinear methods [3–5]. Nonlinear methods generally lead to more accurate estimation results for nonlinear or non-Gaussian signal models at the cost of heavier computational and communication burden.

We consider in this paper the use of Rao-Blackwellized point mass filters (PMF) [6–8]. PMFs are deterministic, and recursively evaluate the posterior probability of *all possible* current states of the hidden (discrete) random variable given the set of observations. The complexity of PMFs is proportional to the number of possible state transitions and, in many setups, is much smaller than that of particle filters, whose complexity is proportional to the required number of particles. Contrary to particle filters, PMFs may need to approximate the posterior distributions of the unknown model parameters to keep computations feasible [7]. In the considered blind equalization scenario, those posteriors are Gaussian mixtures with exponentially growing size, which are approximated by mixtures with constant size by the mixture reduction algorithms [9].

In this paper, we develop novel distributed blind equalization approaches [3] based on PMFs. The proposed methods extend previous ones [6, 10] to a signal model that considers multiple remote receivers, and exhibit reduced complexity compared to [11]. The algorithms do not demand a fusion center, as each individual receiver processes its observations independently and exchange information to approximate the optimal joint estimate of the transmitted data given all observations. As we verify via numerical simulations, the proposed algorithms surpass certain approaches based on particle filters for comparable communication requirements.

The remainder of the text is organized as follows: in Section 2 we describe the considered problem setup and present in Section 3 the PMF. Section 4 introduces a distributed blind equalization algorithm based on PMF. Section 5, in turn, describes a second approach aimed at reducing communication complexity. In Section 6 we assess the performance of the proposed methods and finally draw our conclusions in Section 7.

2. PROBLEM SETUP

Denote by $x_t \in \{\pm 1\}$ the differentially encoded binary symbol transmitted at time instant t corresponding to $\{b_t\}$, an independent, identically distributed (i.i.d.) binary bit sequence. The observations $y_{r,t}$, $1 \leq r \leq R$, at the $r-$th node of a network of R receivers are assumed to be the output of the additive noise frequency-selective channel

$$y_{r,t} = \mathbf{h}_r^T \mathbf{x}_t + v_{r,t} \ , \tag{1}$$

where $\mathbf{x}_t \triangleq [x_t \dots x_{t-L+1}]^T$, L denotes the channel order, $v_{r,t}$ is an i.i.d. zero-mean Gaussian random process of known variance σ_r^2, and the *unknown, random* channel impulse response parameter vector $\mathbf{h}_r \in \mathbb{R}^{L\times 1}$ is assumed to be distributed a priori as $\mathbf{h}_r \sim \mathcal{N}(\mathbf{h}_r|\mathbf{0};\mathbf{I}/\varepsilon^2)$, where $\mathcal{N}$ denotes a (multivariate) Gaussian p.d.f., and ε is the model's hyper-parameter. The random quantities $\mathbf{x}_t, \mathbf{h}_1, \dots, \mathbf{h}_R, v_{1,t}, \dots, v_{R,t}$ are presumed to be mutually independent.

978-0-9928626-3-3/15/$31.00 ©2015 IEEE

Under these assumptions, our objective is to develop a recursive method for obtaining maximum a posteriori estimates $\hat{b}_t = \arg\max_{b_t} p(b_t|Y_t)$, where $Y_t \triangleq \{y_{1,1:t}, \ldots, y_{R,1:t}\}$ and $y_{r,1:t} \triangleq \{y_{r,1}, \ldots, y_{r,t}\}$. In the sequel, we also employ the notation $y_t \triangleq y_{1:R,t}$ and $\mathbf{h} \triangleq \mathbf{h}_{1:R}$.

3. RAO-BLACKWELLIZED POINT MASS FILTER

The Rao-Blackwellized Point Mass Filter (PMF) [7] allows one to recursively evaluate the posterior discrete probability $p(\mathbf{x}_t|Y_t)$. Suppose that, at time instant $t-1$, we have $p(\mathbf{x}_{t-1}|Y_{t-1})$ and $p(\mathbf{h}|\mathbf{x}_{t-1}, Y_{t-1})$. For $t = 1$, these quantities reduce to the prior probabilities of symbol $p(\mathbf{x}_0)$ and the prior probability density function $p(\mathbf{h})$, respectively. The PMF first computes

$$\begin{aligned} p(y_t, \mathbf{x}_t, \mathbf{h}, \mathbf{x}_{t-1}|Y_{t-1}) = p(y_t|\mathbf{x}_t, \mathbf{h}, \mathbf{x}_{t-1}, Y_{t-1}) \times \\ p(\mathbf{x}_t|\mathbf{h}, \mathbf{x}_{t-1}, Y_{t-1}) p(\mathbf{h}|\mathbf{x}_{t-1}, Y_{t-1}) p(\mathbf{x}_{t-1}|Y_{t-1}). \end{aligned} \tag{2}$$

The updated conditional probability of the parameters $\mathbf{h}$ can then be evaluated as

$$p(\mathbf{h}|\mathbf{x}_t, Y_t) = \frac{\sum_{\mathbf{x}_{t-1}} p(y_t, \mathbf{x}_t, \mathbf{h}, \mathbf{x}_{t-1}|Y_{t-1})}{\sum_{\mathbf{x}_{t-1}} \int_{\mathbb{R}^{LR}} p(y_t, \mathbf{x}_t, \mathbf{h}, \mathbf{x}_{t-1}|Y_{t-1})\, d\mathbf{h}}, \tag{3}$$

and the updated posterior of the states $\mathbf{x}_t$ as

$$p(\mathbf{x}_t|Y_t) \propto \sum_{\mathbf{x}_{t-1}} \int_{\mathbb{R}^{LR}} p(y_t, \mathbf{x}_t, \mathbf{h}, \mathbf{x}_{t-1}|Y_{t-1})\, d\mathbf{h}. \tag{4}$$

For the recursion (2)-(4) to be of practical use, $\mathbf{x}_t$ must be discretely distributed on a finite number of points and the integral in $\mathbf{h}$ must be treatable. This generally requires that some form of approximation be applied to $p(\mathbf{h}|\mathbf{x}_t, Y_t)$.

4. PMF-BASED DISTRIBUTED BLIND EQUALIZATION ALGORITHM

Using Equation 1 and exploiting conditional independence relations, it follows that, for a given pair of possible states $\mathbf{x}_t^{(i)}$ and $\mathbf{x}_{t-1}^{(j)}$, we have

$$p(y_t|\mathbf{x}_t^{(i)}, \mathbf{h}, \mathbf{x}_{t-1}^{(j)}, Y_{t-1}) = \prod_{r=1}^{R} \mathcal{N}\left(y_{r,n}|(\mathbf{x}_t^{(i)})^T \mathbf{h}_r; \sigma_r^2\right), \tag{5}$$

$$p(\mathbf{x}_t^{(i)}|\mathbf{h}, \mathbf{x}_{t-1}^{(j)}, Y_{t-1}) = p(\mathbf{x}_t^{(i)}|\mathbf{x}_{t-1}^{(j)}). \tag{6}$$

For the integral in (3) to have closed analytical expression, we approximate $p(\mathbf{h}|\mathbf{x}_{t-1}^{(j)}, Y_{t-1})$ as the $M-$mode Gaussian product mixture

$$p(\mathbf{h}|\mathbf{x}_{t-1}^{(j)}, Y_{t-1}) \triangleq \sum_{m=1}^{M} w_{t-1}^{(j,m)} \prod_{r=1}^{R} \mathcal{N}\left(\mathbf{h}_r|\hat{\mathbf{h}}_{r,t-1}^{(j,m)}; \mathbf{\Sigma}_{r,t-1}^{(j,m)}\right), \tag{7}$$

where $w_{t-1}^{(j,m)}$ are positive weights such that $\sum_{m=1}^{M} w_{t-1}^{(j,m)} = 1, \forall j$.

Substituting (5)-(7) into (2), it follows that

$$\begin{aligned} & p(y_t, \mathbf{x}_t^{(i)}, \mathbf{h}, \mathbf{x}_{t-1}^{(j)}|Y_{t-1}) = \\ & = \sum_{m=1}^{M} w_{t-1}^{(j,m)} \prod_{r=1}^{R} \left\{ \mathcal{N}\left(y_{r,n}|(\mathbf{x}_t^{(i)})^T \mathbf{h}_r; \sigma_r^2\right) \times \right. \\ & \left. \mathcal{N}\left(\mathbf{h}_r|\hat{\mathbf{h}}_{r,t-1}^{(j,m)}; \mathbf{\Sigma}_{r,t-1}^{(j,m)}\right)\right\} p(\mathbf{x}_t^{(i)}|\mathbf{x}_{t-1}^{(j)}) p(\mathbf{x}_{t-1}^{(j)}|Y_{t-1}) \qquad (8) \\ & = \sum_{m=1}^{M} w_{t-1}^{(j,m)} \prod_{r=1}^{R} \left\{ \mathcal{N}\left(y_{r,n}|(\mathbf{x}_t^{(i)})^T \hat{\mathbf{h}}_{r,t-1}^{(j,m)}; \gamma_{r,t}^{(i,j,m)}\right) \times \right. \\ & \left. \mathcal{N}\left(\mathbf{h}_r|\hat{\mathbf{h}}_{r,t}^{(i,j,m)}; \mathbf{\Sigma}_{r,t}^{(i,j,m)}\right)\right\} p(\mathbf{x}_t^{(i)}|\mathbf{x}_{t-1}^{(j)}) p(\mathbf{x}_{t-1}^{(j)}|Y_{t-1}), \end{aligned} \tag{9}$$

where, to obtain (9) from (8) we exploited standard Kalman filter theory results (see [12] for details), and [12]

$$\gamma_{r,t}^{(i,j,m)} = (\mathbf{x}_t^{(i)})^T \mathbf{\Sigma}_{r,t-1}^{(j,m)} \mathbf{x}_t^{(i)} + \sigma_r^2, \tag{10}$$

$$e_{r,t}^{(i,j,m)} = y_{r,t} - (\mathbf{x}_t^{(i)})^T \hat{\mathbf{h}}_{r,t-1}^{(j,m)}, \tag{11}$$

$$\hat{\mathbf{h}}_{r,t}^{(i,j,m)} = \hat{\mathbf{h}}_{r,t-1}^{(j,m)} + \mathbf{\Sigma}_{r,t-1}^{(j,m)} \mathbf{x}_t^{(i)} e_{r,t}^{(i,j,m)} / \gamma_{r,t}^{(i,j,m)}, \tag{12}$$

$$\mathbf{\Sigma}_{r,t}^{(i,j,m)} = \mathbf{\Sigma}_{r,t-1}^{(j,m)} - \mathbf{\Sigma}_{r,t-1}^{(j,m)} \mathbf{x}_t^{(i)} (\mathbf{x}_t^{(i)})^T \mathbf{\Sigma}_{r,t-1}^{(j,m)} / \gamma_{r,t}^{(i,j,m)}, \tag{13}$$

with $\hat{\mathbf{h}}_{r,0}^{(j,m)} = \mathbf{0}$ and $\mathbf{\Sigma}_{r,0}^{(j,m)} = \mathbf{I}/\varepsilon^2$. To integrate $\mathbf{h}$ out of (9), it suffices to discard the Gaussian densities in $\mathbf{h}_r$. Performing such operation and substituting the result into (4), we obtain that

$$p(\mathbf{x}_t^{(i)}|Y_t) \propto \sum_j \sum_{m=1}^{M} w_t^{(i,j,m)} \tag{14}$$

where

$$\begin{aligned} w_t^{(i,j,m)} \triangleq w_{t-1}^{(j,m)} \prod_{r=1}^{R} \left\{\lambda_{r,t}^{(i,j,m)}\right\} p(\mathbf{x}_t^{(i)}|\mathbf{x}_{t-1}^{(j)}) \times \\ p(\mathbf{x}_{t-1}^{(j)}|Y_{t-1}), \end{aligned} \tag{15}$$

$$\lambda_{r,t}^{(i,j,m)} \triangleq \mathcal{N}\left(y_{r,t}|(\mathbf{x}_t^{(i)})^T \hat{\mathbf{h}}_{r,t-1}^{(j,m)}; \gamma_{r,t}^{(i,j,m)}\right). \tag{16}$$

Similarly, it follows from (3) that

$$p(\mathbf{h}|\mathbf{x}_t^{(i)}, Y_t) \triangleq \sum_j \sum_{m=1}^{M} \tilde{w}_t^{(i,j,m)} \prod_{r=1}^{R} \mathcal{N}\left(\mathbf{h}_r|\hat{\mathbf{h}}_{r,t}^{(i,j,m)}; \mathbf{\Sigma}_{r,t}^{(i,j,m)}\right), \tag{17}$$

where $\tilde{w}_t^{(i,j,m)} \triangleq w_t^{(i,j,m)} / \sum_j \sum_{m=1}^M w_t^{(i,j,m)}$.

We wish to show that (17) has the same functional form of (7). To verify that, we introduce the unitary mapping $(i,j,m) \mapsto (i,m') \triangleq (i,(j-1)M+m)$, $1 \leq m' \leq MN_j$, where N_j denotes the number of possible predecessor states $\mathbf{x}_{t-1}^{(j)}$ given $\mathbf{x}_t^{(i)}$ (2 for binary systems). This allows (17) to be rewritten as

$$p(\mathbf{h}|\mathbf{x}_t^{(i)}, Y_t) \triangleq \sum_{m'=1}^{MN_j} \tilde{w}_t^{(i,m')} \prod_{r=1}^{R} \mathcal{N}\left(\mathbf{h}_r|\hat{\mathbf{h}}_{r,t}^{(i,m')}; \mathbf{\Sigma}_{r,t}^{(i,m')}\right), \tag{18}$$

which is similar to (7) except for the increased number of sum components. As a consequence, if no further approximations are made to (18), the complexity of PMFs grows exponentially with time.

4.1. Mixture Reduction

To keep the computational complexity of the PMF treatable, we follow [7] [6] and reduce the mixture in (18) to M components. This can be carried out by several algorithms (see [9] for a review), which differ in performance and computational complexity.

We propose to use a new version of West's algorithm [13] adapted to product mixtures. The algorithm is iterative: take a mixture as (18) with M' components. If $M' = M$, finish. Otherwise, if $M' > M$, determine

$$\tilde{m} \triangleq \arg\min_m w_t^{(i,m)}.$$

Then, for all $m \neq \tilde{m}$, evaluate the sum of symmetrized Kullback-Leibler divergences [14]

$$\begin{aligned}\mathcal{D}(\tilde{m}, m) &\triangleq \sum_{r=1}^{R} \mathcal{D}_{KL}(\phi_{r,t}^{i,\tilde{m}} || \phi_{r,t}^{i,m}) + \mathcal{D}_{KL}(\phi_{r,t}^{i,m} || \phi_{r,t}^{i,\tilde{m}}) \\ &= \sum_{r=1}^{R} \Big\{ \mathrm{tr}((\mathbf{\Sigma}_{r,t}^{(i,m)})^{-1} \mathbf{\Sigma}_{r,t}^{(i,\tilde{m})}) + \mathrm{tr}((\mathbf{\Sigma}_{r,t}^{(i,\tilde{m})})^{-1} \mathbf{\Sigma}_{r,t}^{(i,m)}) + \\ &\quad \|\hat{\mathbf{h}}_{r,t}^{(i,m)} - \hat{\mathbf{h}}_{r,t}^{(i,\tilde{m})}\|^2_{(\mathbf{\Sigma}_{r,t}^{(i,m)})^{-1} + (\mathbf{\Sigma}_{r,t}^{(i,\tilde{m}))})^{-1}} - 2L \Big\},\end{aligned} \tag{19}$$

where $\mathcal{D}_{KL}(\cdot||\cdot)$ denotes the Kullback-Leibler divergence [13] and $\phi_{r,t}^{i,m} \triangleq \mathcal{N}\left(\mathbf{h}_r|\hat{\mathbf{h}}_{r,t}^{(i,m)}; \mathbf{\Sigma}_{r,t}^{(i,m)}\right)$. Next, the index of the nearest component is determined as

$$\hat{m} \triangleq \arg\min_m \mathcal{D}(\tilde{m}, m).$$

After that, components with indexes $\tilde{m}$ and $\hat{m}$ are merged via marginal *moment matching* [13]

$$\bar{w}_t^{(i,\hat{m})} = w_t^{(i,\hat{m})} + w_t^{(i,\tilde{m})}, \tag{20}$$

$$\bar{\hat{\mathbf{h}}}_{r,t}^{(i,\hat{m})} = (w_t^{(i,\hat{m})} \hat{\mathbf{h}}_{r,t}^{(i,\hat{m})} + w_t^{(i,\tilde{m})} \hat{\mathbf{h}}_{r,t}^{(i,\tilde{m})}) / \bar{w}_t^{(i,\hat{m})}, \tag{21}$$

$$\begin{aligned}\bar{\mathbf{\Sigma}}_{r,t}^{(i,\hat{m})} &= (w_t^{(i,\hat{m})} \mathbf{\Sigma}_{r,t}^{(i,\hat{m})} + w_t^{(i,\tilde{m})} \mathbf{\Sigma}_{r,t}^{(i,\tilde{m})}) / \bar{w}_t^{(i,\hat{m})} + \\ &+ \frac{w_t^{(i,\hat{m})} w_t^{(i,\tilde{m})}}{(\bar{w}_t^{(i,\hat{m})})^2} (\hat{\mathbf{h}}_{r,t}^{(i,\hat{m})} - \hat{\mathbf{h}}_{r,t}^{(i,\tilde{m})})(\hat{\mathbf{h}}_{r,t}^{(i,\hat{m})} - \hat{\mathbf{h}}_{r,t}^{(i,\tilde{m})})^T.\end{aligned} \tag{22}$$

After evaluating (20)-(22), the distinctive bar is dropped from the quantities on the left-hand size and the component $\tilde{m}$ is deleted. The resulting mixture has $M' - 1$ components.

4.2. Distributed Implementation

A distributed implementation of PMF-based blind equalizer requires that computations performed in the *filtering* step (Section 4) and the mixture reduction step (Section 4.1) be spread across the nodes.

To implement the filtering step, each node r runs a bank of $M \cdot 2^{L+1}$ Kalman filters that process the local observation $y_{r,t}$, update the statistics of $\mathbf{h}_r$, and evaluate $\lambda_{r,t}^{(i,j,m)}$ via (16). These quantities are then *broadcast* to the remaining nodes ($M \cdot 2^{L+1}$ real numbers), which evaluate $w_t^{(i,j,m)}$ (Equation 15) and the symbols' posterior probabilities via (14).

As for the mixture reduction step, the $r-$th node evaluates the $r-$th term of the sum (19) and broadcasts this quantity to the remaining nodes. All nodes then are able to evaluate (19) and determine the indexes of components to be merged. Moment matching operations (21)-(22) referring to the $r-$th component are performed with local information. To reduce a mixture from $2M$ to M components, a total of $(3M^2 - M)/2$ distances must be evaluated. As this procedure must be repeated for each possible state $\mathbf{x}_t^{(i)}$, a total of $(3M^2 - M)2^{L-1}$ real numbers must be broadcast.

5. THE DISTRIBUTED PMF (DPMF) ALGORITHM

To reduce the communication burden, we propose an alternative algorithm in which the general $r-$th node employs a *single mode* approximation to the remote marginal posteriors $p(\mathbf{h}_s|\mathbf{x}_{t-1}^{(j)}, Y_{t-1})$, $r \neq s$. By this new approach, the r-th node approximates (7) as[1]

$$\begin{aligned}p_r(\mathbf{h}|\mathbf{x}_{t-1}^{(j)}, Y_{t-1}) \triangleq\; & p_r(\mathbf{h}_r|\mathbf{x}_{t-1}^{(j)}, Y_{t-1}) \times \\ & \prod_{\substack{s=1 \\ s\neq r}}^{R} \mathcal{N}\left(\mathbf{h}_s|\tilde{\mathbf{h}}_{s,t-1}^{(j)}; \tilde{\mathbf{\Sigma}}_{s,t-1}^{(j)}\right),\end{aligned} \tag{23}$$

[1] We denote by $p_r(\cdot)$ and $w_{r,t-1}^{(j,m)}$ quantities that are specific to the $r-$th node.

where

$$p_r(\mathbf{h}_r|\mathbf{x}_{t-1}^{(j)}, Y_{t-1}) \triangleq \sum_{m=1}^{M} w_{r,t-1}^{(j,m)} \mathcal{N}\left(\mathbf{h}_r|\hat{\mathbf{h}}_{r,t-1}^{(j,m)}; \boldsymbol{\Sigma}_{r,t-1}^{(j,m)}\right) \tag{24}$$

is the local marginal posterior, and $\tilde{\mathbf{h}}_{s,t-1}^{(j)}$ and $\tilde{\boldsymbol{\Sigma}}_{s,t-1}^{(j)}$ are the moments of the remote posteriors $p_s(\mathbf{h}_s|\mathbf{x}_{t-1}^{(j)}, Y_{t-1})$ (defined as in Equation 24), which can be evaluated as

$$\tilde{\mathbf{h}}_{s,t-1}^{(j)} = \sum_{m=1}^{M} w_{s,t-1}^{(j,m)}\, \hat{\mathbf{h}}_{s,t-1}^{(j,m)}, \tag{25}$$

$$\begin{aligned}\tilde{\boldsymbol{\Sigma}}_{s,t-1}^{(j)} = \sum_{m=1}^{M} w_{s,t-1}^{(j,m)} \left[\hat{\mathbf{h}}_{s,t-1}^{(j,m)}(\hat{\mathbf{h}}_{s,t-1}^{(j,m)})^T + \boldsymbol{\Sigma}_{s,t-1}^{(j,m)}\right] \\ - \tilde{\mathbf{h}}_{s,t-1}^{(j)}(\tilde{\mathbf{h}}_{s,t-1}^{(j)})^T.\end{aligned} \tag{26}$$

Similarly to Section 3, plugging (23) into (2) yields

$$\begin{aligned}p_r(y_t, \mathbf{x}_t^{(i)}, \mathbf{h}, \mathbf{x}_{t-1}^{(j)}|Y_{t-1}) = p(\mathbf{x}_t^{(i)}|\mathbf{x}_{t-1}^{(j)})p_r(\mathbf{x}_{t-1}^{(j)}|Y_{t-1})\times \\ \sum_{m=1}^{M}\left\{w_{r,t-1}^{(j,m)}\lambda_{r,t}^{(i,j,m)}\mathcal{N}\left(\mathbf{h}_r|\hat{\mathbf{h}}_{r,t}^{(i,j,m)}; \boldsymbol{\Sigma}_{r,t}^{(i,j,m)}\right)\right\}\times \\ \prod_{\substack{s=1\\ s\neq r}}^{R}\left\{\tilde{\lambda}_{s,t}^{(i,j)}\mathcal{N}\left(\mathbf{h}_s|\tilde{\mathbf{h}}_{s,t}^{(i,j)}; \tilde{\boldsymbol{\Sigma}}_{s,t}^{(i,j)}\right)\right\},\end{aligned} \tag{27}$$

where $\tilde{\lambda}_{s,t}^{(i,j)} \triangleq \mathcal{N}\left(y_{s,n}|(\mathbf{x}_t^{(i)})^T\tilde{\mathbf{h}}_{s,t-1}^{(j)}; \tilde{\gamma}_{s,t}^{(i,j)}\right)$ and $\tilde{\gamma}_{s,t}^{(i,j)} \triangleq (\mathbf{x}_t^{(i)})^T\tilde{\boldsymbol{\Sigma}}_{s,t-1}^{(j)}\mathbf{x}_t^{(i)} + \sigma_s^2$. The expressions for $\tilde{\mathbf{h}}_{s,t}^{(i,j)}$ and $\tilde{\boldsymbol{\Sigma}}_{s,t}^{(i,j)}$ will be omitted since these variables do not need to be evaluated by the proposed algorithm.

Integrating $\mathbf{h}$ out of (27) and substituting the result into (3) and (4), we get that

$$p_r(\mathbf{x}_t^{(i)}|Y_t) \propto \sum_j\sum_{m=1}^{M} w_{r,t}^{(i,j,m)}, \tag{28}$$

$$p_r(\mathbf{h}_r|\mathbf{x}_t^{(i)}, Y_t) = \sum_j\sum_{m=1}^{N} \tilde{w}_{r,t}^{(i,j,m)}\mathcal{N}\left(\mathbf{h}_r|\hat{\mathbf{h}}_{r,t}^{(i,j,m)}; \boldsymbol{\Sigma}_{r,t}^{(i,j,m)}\right), \tag{29}$$

where $\tilde{w}_{r,t}^{(i,j,m)} = w_{r,t}^{(i,j,m)}/\sum_j\sum_{m=1}^{M} w_{r,t}^{(i,j,m)}$, and

$$w_{r,t}^{(i,j,m)} \triangleq p(\mathbf{x}_t^{(i)}|\mathbf{x}_{t-1}^{(j)})p(\mathbf{x}_{t-1}^{(j)}|Y_{t-1})w_{r,t-1}^{(i,j,m)}\lambda_{r,t}^{(i,j,m)}\prod_{\substack{s=1\\ s\neq r}}^{R}\tilde{\lambda}_{s,t}^{(i,j)}. \tag{30}$$

By plugging (29) back into (23), one verifies that, in contrast to the algorithm of Section 4, the DPMF algorithm can perform mixture reduction (Section 4.1) *independently* for each local marginal posterior (Equation 29) without changing the functional form of (23) as t grows. To accomplish this, it suffices to replace $w_t^{(\cdot)}$ with $w_{r,t}^{(\cdot)}$ in (20)-(22) and drop the summation in (19).

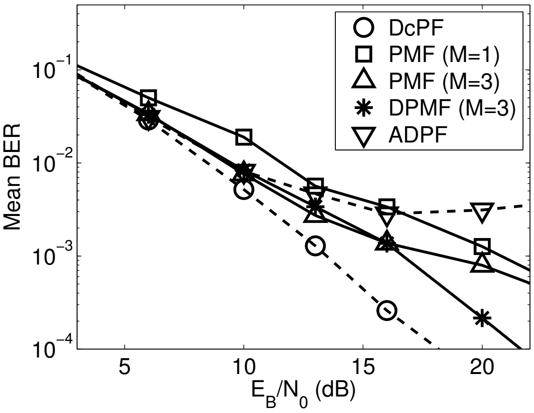

Fig. 1. Mean BER estimated in 3.000 Monte Carlo runs.

As a consequence, the $r-$th node needs only to broadcast $\tilde{\lambda}_{r,t}^{(i,j)}$ (2^{L+1} real numbers), which allow (30) and hence (28) and (29) to be evaluated by the remaining nodes.

6. SIMULATION RESULTS

We evaluated the steady-state performance of the proposed algorithms via Monte Carlo simulations consisting of 3.000 independent runs, in each of which the mean bit error rate (BER) was estimated as a function of E_B/N_0. In each realization, a random sequence of 250 i.i.d differentially encoded binary symbols was transmitted, being the first 150 bits discarded to allow for convergence.

The simulated system has $R = 4$ receivers. The transmission channels $\mathbf{h}_r$ have $L = 3$ coefficients, and were obtained by sampling independently for each receiver from a Gaussian p.d.f. $\mathcal{N}(\mathbf{0}; \mathbf{I})$ and normalizing so that $\|\mathbf{h}_r\|^2 = 1$. The noise variance was determined as $\sigma_r^2 = 1/(E_B/N_0)$. Figure 1 displays the performance of the PMF-based algorithm (PMF) and of the reduced complexity version (DPMF) as a function of E_B/N_0 and M, the number of modes. Note that, for $M = 1$, DPMF is equivalent to PMF. For comparison, we ran with the same setup the optimal joint particle-filter-based algorithm (DcPF) [3] and the particle filtering algorithm that employs parametric approximations (ADPF) [3], both using 300 particles and the prior importance function. As one may verify, the performance of the PMF and DPMF methods grow with the number of modes M, surpassing the performance of the ADPF for $M = 3$. Although the DPMF is outmatched by the optimal DcPF, the communication cost (Table 1) of the former for $M = 3$ is only 5.3% of the latter. Note that this remains true even for larger channel sizes L, as particle-filter-

Table 1. Comparison of Communication and Computational Costs (per node per symbol).

	real numbers broadcast (per symbol)	**operations**
DcPF	P	$\mathcal{O}(PL^2)$
ADPF	2^L	$\mathcal{O}(PL^2)$
PMF	$3(M^2+M)2^{L-1}$	$\mathcal{O}(2^{L+1}M^2L^3)$
DPMF	2^{L+1}	$\mathcal{O}(2^{L+1}M^2L^3)$

based equalizers demand a number of particles $P \gg 2^L$ for proper operation. The computational cost of the PMF and DPMF are dominated by the cost of the mixture reduction algorithm, and is comparable to that of particle filters for the considered parameters.

7. CONCLUSIONS

This paper described two distributed Bayesian algorithms for cooperative blind equalization based on Rao-Blackwellized point mass filters. The PMF algorithm approximates the posterior densities of channels parameters by Gaussian mixtures of fixed order. The required mixture reductions are implemented via a distributed version of the West's algorithm [13]. To reduce communication among nodes, the DPMF algorithm performs mixture reductions only locally, presenting to the remaining nodes a single term Gaussian approximation. Via numerical simulations, we verify that the proposed algorithms outperform certain particle-filtering-based algorithms that incur in comparable communication and computational complexities.

Counterintuitively, for $M > 1$, the PMF performed worse than the DPMF for low noise levels. This can possibly be explained by the fact that the PMF deals with Gaussian mixtures in which the components are more distant (in the KL measure) than for the DPMF (since all terms in the product but one are equal), which exacerbates distortions caused by the mixture reduction process. It remains to be verified, as future work, if the use of more complex mixture reduction algorithms [9] can avoid this effect.

REFERENCES

[1] A. Ribeiro, G. B. Giannakis, and S. I. Roumeliotis, "SOI-KF: Distributed Kalman filtering with low-cost communications using the sign of innovations," *IEEE Transactions on Signal Processing*, vol. 54, no. 12, pp. 4782–4795, 2006.

[2] C. G. Lopes and A. H. Sayed, "Incremental adaptive strategies over distributed networks," *IEEE Transactions on Signal Processing*, vol. 55, no. 8, pp. 4064–4077, 2007.

[3] C. J. Bordin Jr. and M. G. S. Bruno, "Nonlinear distributed blind equalization using network particle filtering," in *Statistical Signal Processing Workshop (SSP), 2009 IEEE*, Cardiff - U.K., Aug. 2009, pp. 469–472.

[4] A. Mohammadi and A. Asif, "Consensus-based distributed unscented particle filter," in *Statistical Signal Processing Workshop (SSP), 2011 IEEE*, Nice - France, June 2011, pp. 237–240.

[5] O. Hlinka, F. Hlawatsch, and P. M. Djurić, "Distributed particle filtering in agent networks: A survey, classification, and comparison," *IEEE Signal Processing Magazine*, vol. 30, no. 1, pp. 61–81, 2013.

[6] F. Lehmann, "A Gaussian mixture approach to blind equalization of block-oriented wireless communications," *EURASIP Journal on Advances in Signal Processing*, vol. 2010, pp. 6, 2010.

[7] V. Šmídl and M. Gašperin, "Rao-Blackwellized point mass filter for reliable state estimation," in *Information Fusion (FUSION), 2013 16th International Conference on*, Istanbul - Turkey, July 2013, pp. 312–318.

[8] M. Raitoharju, S. Ali-Löytty, and R. Piché, "Binomial gaussian mixture filter," *EURASIP Journal on Advances in Signal Processing*, vol. 2015, no. 1, pp. 36, 2015.

[9] D. F. Crouse, P. Willett, K. Pattipati, and L. Svensson, "A look at Gaussian mixture reduction algorithms," in *Information Fusion (FUSION), 2011 14th International Conference on*, Chicago - IL, July 2011, pp. 1–8.

[10] Y. Zheng, Z. Lin, and Y. Ma, "Recursive blind identication of Non-Gaussian time-varying ar model and application to blind equalisation of time-varying channel," *IEE Proceedings: Vision, Image and Signal Processing*, vol. 148, no. 4, pp. 275–282, 2001.

[11] C. J. Bordin Jr. and M. G. S. Bruno, "Bayesian distributed blind equalization based on density-sum filters," in *Telecommunications Symposium (ITS), 2014 International*, São Paulo - Brazil, July 2014, pp. 1–5.

[12] M. G. S. Bruno, *Sequential Monte Carlo Methods for Nonlinear Discrete-Time Filtering*, vol. 6 of *Synthesis Lectures on Signal Processing*, Morgan & Claypool Publishers, Feb. 2013.

[13] M. West, "Approximating posterior distributions by mixtures," *Journal of the Royal Statistical Society: Series B*, vol. 55, no. 2, pp. 409–422, 1993.

[14] G. Kitagawa, "The two-filter formula for smoothing and an implementation of the Gaussian-sum smoother," *Annals of the Institute of Statistical Mathematics*, vol. 46, no. 4, pp. 605–623, 1994.