Window-Based Expectation Propagation for Adaptive Signal Detection in Flat-Fading Channels

Yuan Qi and Thomas P. Minka

Abstract— **In this paper, we propose a new Bayesian receiver for signal detection in flat-fading channels. First, the detection problem is formulated as an inference problem in a graphical model that models a hybrid dynamic system with both continuous and discrete variables. Then, based on the expectation propagation (EP) framework, we develop a smoothing algorithm to address the inference problem and visualize this algorithm using factor graphs. As a generalization of loopy belief propagation, EP efficiently approximates Bayesian estimation by iteratively propagating information between different nodes in the graphical model and projecting the posterior distributions into the exponential family. We use window-based EP smoothing for online estimation as in the signal detection problem. Window-based EP smoothing achieves accuracy similar to that obtained by batch EP smoothing, as shown in our simulations, while reducing delay time. Compared to sequential Monte Carlo filters and smoothers, the new method has lower computational complexity since it makes analytically deterministic approximation instead of Monte Carlo approximations. Our simulations demonstrate that the new receiver achieves accurate detection without the aid of any training symbols or decision feedbacks. Furthermore, the new receiver achieves accuracy comparable to that achieved by sequential Monte Carlo methods, but with less than one-tenth computational cost.**

Index Terms— **Signal detection, Expectation propagation, Sequential estimation, Fading channels, Monte Carlo methods, Nonlinear or non-Gaussian systems, Bayesian inference.**

I. Introduction

SIGNAL detection in flat Rayleigh fading channels can be viewed as a statistical inference problem for the information bits while marginalizing over the unknown channel state. Many approaches, including per-survivor processing [1], pilot symbol usage, [2], and iterative estimation [3], have been proposed to address this inference problem. Computationally, the main challenge is representing and utilizing our uncertainty about the channel state, which is often continuous, high-dimensional, and time-varying. At one extreme, we could use the received signal to estimate the channel state and assume this estimate to be correct, thus discarding all uncertainty [4], [5]. At the other extreme, we could use a set of Monte Carlo samples to represent the distribution over channel states to arbitrary accuracy, but at much higher cost [6], [7].

Between these extremes, we have approximate methods which represent uncertainty about the channel state with analytically tractable distributions. For many systems, this approach provides the best cost/accuracy trade-off. For example, the canonical distribution method of [8] has been successfully applied to channels with low-dimensional state, e.g. a single unknown phase parameter [9]. This method is an extension of the sum-product algorithm for factor graphs, otherwise known as belief propagation in graphical models. For the signal detection problem, belief propagation is not feasible to apply directly, since it would require representing arbitrarily complex continuous distributions. The extension involves approximating each message in an ad-hoc manner, so that the distributions remain canonical.

In this paper, we utilize an alternative approximation scheme, called expectation propagation (EP), which should give better results than the canonical distribution method. This is for two reasons. First, we minimize divergence between variable distributions, rather than divergence between messages. This makes sense because the variable distributions are what we really care about in the problem. Second, EP minimizes a global error function [10], ensuring a proper fixed point for the message-passing. We apply this scheme to a flat-fading channel with 8-dimensional hidden state. Note that when the channel has higher-dimensional state, the more severely we need to approximate the uncertainty, which makes it especially important to choose the approximation well.

Our starting point is to formulate the signal detection problem as a Bayesian inference problem on a graphical model that models a hybrid dynamic system with both continuous and discrete variables. Then we develop a window-based expectation propagation (EP) algorithm for hybrid dynamic systems and apply it to signal detection in flat-fading channels.

The end result is a detection algorithm that passes messages whose parameters are governed by a divergence minimization rule. This rule ensures that each message discards the minimal amount of useful information, where "useful" is defined by the rest of the network. Compared to sequential Monte Carlo filters and smoothers, EP has much lower computational complexity since it makes deterministic analytic approximations, instead of Monte Carlo approximations. To make expectation propagation suitable for online estimation as in the signal detection problem, we propose window-based EP smoothing which is a trade-off between assumed-density filtering and traditional batch EP smoothing. As shown in our simulations for signal detection, window-based EP smoothing achieves accuracy similar to that obtained by batch EP smoothing.

In the rest of this paper, we first formulate the signal detection problem as a Bayesian inference problem on dynamic graphical models. Then we briefly review inference techniques

Manuscript received April 7, 2005; revised May 15, 2006 and September 12, 2006; accepted September 15, 2006. The associate editor coordinating the review of this paper and approving it for publication was G. Vitetta.

Y. Qi is with the MIT Computer Science and Artificial Intelligence Laboratory, Cambridge, Massachusetts, 02139 USA (email: alanqi@csail.mit.edu).

T. P. Minka is with Microsoft Research, 7 J J Thomson Avenue, Cambridge, CB3 0FB, UK (email: minka@microsoft.com).

Digital Object Identifier 10.1109/TWC.2006.05237.

for hybrid dynamic systems with nonlinear and non-Gaussian likelihoods. Section IV presents the EP algorithm for hybrid dynamic systems and visualize this algorithm by factor graphs [13]. Section V describes window-based EP smoothing for online estimation, followed by Section VI that compares the computational efficiency of window-based smoothing with sequential Monte Carlo methods. Our simulations in Section VII demonstrate that the new window-based EP receiver achieves accurate detection without the aid of any training symbols or decision feedbacks. Furthermore, the new receiver achieves accuracy comparable to that achieved by sequential Monte Carlo methods, with one-tenth the computational cost.

The following notational conventions will be used throughout the paper. Italics lower-case Greek letters (a) denote scalars, bold lower-case letters ($\mathbf{a}$) denote vectors, bold upper-case letters ($\mathbf{A}$) denote matrices, and italics upper-case letters (N) denote known constants. The superscript star, *, denotes the Hermitian transpose ($\mathbf{h}^\star$) of a vector ($\mathbf{h}$).

II. Signal Detection Problem

A wireless communication system with a fading channel can be modeled as [6]

$$y_t = s_t \alpha_t + w_t, \qquad t = 0, 1, \ldots \tag{1}$$

where y_t, s_t, α_t and w_t are the received signal, the transmitted symbol, the fading channel coefficient, and the complex Gaussian noise $\mathcal{N}_c(0, \sigma^2)$ at time t respectively. The symbols s_t are complex and take values from a finite alphabet of size M. The fading coefficients α_t can be modeled by a complex autoregressive moving-average (ARMA) process as follows:

$$\alpha_t = \sum_{i=0}^{\rho} \theta_i v_{t-i} - \sum_{i=1}^{\rho} \phi_i \alpha_{t-i} \tag{2}$$

where $\Theta = \{\theta_t\}$ and $\Phi = \{\phi_t\}$ are the ARMA coefficients, and v_t is the white complex Gaussian noise with unit variance. In this paper, the ARMA coefficients Θ and Φ are assumed to be known.

We assume that there is a prior distribution $p(s_t)$ on each symbol. This prior distribution $p(s_t)$ can come from the decoding module in joint iterative decoding and demodulation. Defining $\mathbf{h}$ as

$$\mathbf{h} \equiv [\theta_0, \theta_1, \ldots, \theta_\rho]^\star \tag{3}$$

and introducing latent variable $\mathbf{x}_t$ such that $\alpha_t = \mathbf{h}^\star \mathbf{x}_t$, we can rewrite the communication system as a state-space model:

$$\mathbf{x}_t = \mathbf{F}\mathbf{x}_{t-1} + \mathbf{g}_t v_t \tag{4}$$
$$y_t = s_t \mathbf{h}^\star \mathbf{x}_t + w_t \tag{5}$$

where

$$\mathbf{F} = \begin{pmatrix} -\phi_1 & -\phi_2 & \ldots & -\phi_\rho & 0 \\ 1 & 0 & \ldots & 0 & 0 \\ 0 & 1 & \ldots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \ldots & 1 & 0 \end{pmatrix}, \quad \mathbf{g} = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \tag{6}$$

and the dimension of $\mathbf{x}$ is $d = \rho + 1$. We can represent the state-space model defined by equations (4) and (5) as a graphical model (Figure 1).

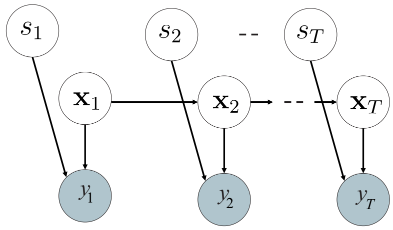


Fig. 1. Graphical model for adaptive signal detection. The shaded nodes $\{y_t\}_{t=1,\ldots,T}$, which represent the received signal, in the graph are the observed variables. The signal detection problem amounts to an inference problem on this graphical model, which couples continuous latent variables $\{\mathbf{x}_t\}_{t=1,\ldots,T}$, i.e., the channel coefficients, with discrete latent variables $\{s_t\}_{t=1,\ldots,T}$, i.e., the transmitted symbol.

The signal detection problem can then be formulated as an inference problem in this graphical model. We address this problem using a Bayesian approach. Specifically, for filtering, we update the posterior distribution $p(s_t, \mathbf{x}_t | y_{1:t})$ based on the observations from the beginning to the current time t, i.e., $y_{1:t} = [y_1, \ldots, y_t]$. For smoothing, we compute the posterior $p(s_t, \mathbf{x}_t | y_{1:T})$ based on the whole observation sequence, i.e., $y_{1:T} = [y_1, \ldots, y_T]$, where T is the length of the observation sequence. Since smoothing uses more information from the observations than filtering, smoothing generally achieves more accurate estimation than filtering.

Given the prior distribution $p(s_1, \mathbf{x}_1)$, the observation distributions $p(y_t | s_t, \mathbf{x}_t)$, and the transition distributions $p(s_{t+1}, \mathbf{x}_{t+1} | s_t, \mathbf{x}_t)$, the exact posterior distribution is proportional to their product:

$$p(y_t | s_t, \mathbf{x}_t) = \mathcal{N}_c(y_t | s_t \mathbf{h}^\star \mathbf{x}_t, \sigma^2) \tag{7}$$
$$p(s_{t+1}, \mathbf{x}_{t+1} | s_t, \mathbf{x}_t) = p(\mathbf{x}_{t+1} | \mathbf{x}_t) p(s_{t+1}) \tag{8}$$
$$= \mathcal{N}_c(\mathbf{x}_{t+1} | \mathbf{F}\mathbf{x}_t, \mathbf{g}\mathbf{g}^\star) p(s_{t+1}) \tag{9}$$

$$\begin{aligned} p(s_{1:T}, \mathbf{x}_{1:T} | y_{1:T}) &\propto p(s_1) p(\mathbf{x}_1) p(y_1 | s_1, \mathbf{x}_1) \cdot \\ &\cdot \prod_{t=2}^{T} p(s_t, \mathbf{x}_t | s_{t-1}, \mathbf{x}_{t-1}) p(y_t | s_t, \mathbf{x}_t) \\ &\propto p(s_1) p(\mathbf{x}_1) p(y_1 | s_1, \mathbf{x}_1) \cdot \\ &\cdot \prod_{t=2}^{T} p(\mathbf{x}_t | \mathbf{x}_{t-1}) p(s_t) p(y_t | s_t, \mathbf{x}_t) \end{aligned} \tag{10}$$

where $\mathbf{h}$, $\mathbf{F}$, and $\mathbf{g}$ are defined in equations (3) and (6), and the prior $p(\mathbf{x}_1)$ is a Gaussian distribution. Equation (8) holds because s_t's are independent of each other at different times in the dynamic model. As a switching-linear Kalman filtering model, the number of Gaussian components in the exact posterior distribution increases exponentially as the length of the observation sequence increases[14]. Apparently, the explosion of the number of the Gaussians makes the exact inference intractable and necessitates the use of approximate inference techniques. In the following section, we briefly review approximate inference techniques on hybrid dynamic graphical models with nonlinear/non-Gaussian likelihoods.

III. Approximate Inference on Hybrid Dynamic Models with Nonlinear/Non-Gaussian Likelihoods

For linear Gaussian or discrete dynamic models, we have efficient and elegant inference algorithms such as forward-backward algorithm, Kalman-filtering and smoothing. However, we encounter in practice many hybrid dynamic models with continuous and discrete state variables. For these kinds of dynamic models, more complicated approaches are needed to do the inference. Most of these approaches can be categorized into two classes: Monte Carlo and deterministic approximation.

Monte Carlo methods can generally achieve more accurate inference results than deterministic approximation methods, once having drawn a sufficiently large amount of samples. Markov Chain Monte Carlo methods, including Gibbs sampling and Metropolis-Hastings, have been applied to dynamic models to achieve accurate results [15], [16]. Also, resampled sequential Monte Carlo, i.e., particle filtering and smoothing has used to explore the Markovian property of dynamic models to do efficient inference [17], [18]. The sequential Monte Carlo method [6], [7] has been used to wireless signal detection and achieved excellent estimation accuracy. However, since the inference accuracy heavily depends on the number of samples, Monte Carlo methods are generally much slower than deterministic inference methods and make themselves less practical.

Deterministic approximation methods are generally more efficient, but less accurate than Monte Carlo methods. For example, extended Kalman filtering and smoothing [19], a popular deterministic approximation method, linearizes the process and measurement equations by a Taylor expansion about the current state estimate. After linearization, the classical Kalman filter and smoother are applied. However, the wireless signal detection model contains both continuous and discrete variables, so extended Kalman filtering is not directly applicable. Instead, we need another way of making the distributions Gaussian, so that the Kalman filter can be applied. For specific problems, ad-hoc Gaussian approximations can be devised, such as the one in [5]. But we prefer an approach that works well for hybrid dynamic models in general.

Therefore, based on the expectation propagation framework [10], we develop a new smoothing technique for hybrid dynamic systems. This new technique can be viewed as a generalization of Kalman smoothing and can achieve estimation accuracy comparable to that achieved by Monte Carlo methods at a much lower computational cost, as shown in the following sections. In related work, [20] have described a generic EP algorithm for dynamic Bayesian networks and applied it to a model with non-Gaussian state equations and linear-Gaussian observations. They also showed how to modify EP to improve convergence. Here we consider a different class of dynamic models and emphasize efficient implementation for this class.

IV. Expectation Propagation for Signal Detection

In this section, we develop the EP smoothing algorithm for the hybrid dynamic system defined by (4) and (5). As mentioned in the previous section, EP smoothing for dynamic systems generalizes traditional Kalman smoothing. On the one hand, both EP smoothing and Kalman smoothing have forward and backward passes to obtain the posterior distributions of the hidden states. On the other hand, EP smoothing keeps refining the approximate state posterior distributions by iterating forward and backward passes until convergence, while Kalman smoothing, as a special case of belief propagation [13], uses only one forward pass and one backward pass to obtain the exact state posterior distributions. The iteration of forward and backward passes in EP smoothing enables the refinement of observation approximations, which in turn leads to better approximation of the posterior distributions.

In the derivation, we do not assume the form of Kalman filtering as given, but rather derive it as a consequence of certain modeling assumptions. We approach the problem as a general problem in function approximation, where we want to approximate the distribution defined by the model with a simpler analytical form that minimizes information loss. Specifically, we choose to approximate the posterior distribution $p(s_{1:T}, \mathbf{x}_{1:T}|y_{1:T})$ by a factorized distribution $q(s_{1:T}, \mathbf{x}_{1:T})$:

$$q(s_{1:T}, \mathbf{x}_{1:T}) = \prod_{t=1}^{T} q(s_t, \mathbf{x}_t) = \prod_{t=1}^{T} q(s_t)q(\mathbf{x}_t) \tag{11}$$

where $q(s_t, \mathbf{x}_t) = q(s_t)q(\mathbf{x}_t)$ can be interpreted as the approximation of the state posterior $p(s_t, \mathbf{x}_t|y_{1:T})$, also known as the state belief in the machine learning community. The exact posterior $p(s_{1:T}, \mathbf{x}_{1:T}|y_{1:T})$ and its approximation are drawn as factor graphs in Figure 2.

Expectation propagation now provides a rule for optimizing q to fit p. It exploits the fact that p is a product of simple terms. If we approximate each of these terms to match the factorized form of q, then their product will also have the form of q. Let $q_\circ(s_t, \mathbf{x}_t)$ approximate the observation distribution $p(y_t|s_t, \mathbf{x}_t)$ and $q_{\triangleright\triangleleft}(\mathbf{x}_{t-1}, \mathbf{x}_t)$ approximate the transition distribution $p(\mathbf{x}_t|\mathbf{x}_{t-1})$. Then q is by definition

$$\begin{aligned} q(s_{1:T}, \mathbf{x}_{1:T}) = p(\mathbf{x}_1)p(s_1)q_\circ(s_1, \mathbf{x}_1)\cdot \\ \cdot \prod_{t=2}^{T} q_{\triangleright\triangleleft}(\mathbf{x}_{t-1}, \mathbf{x}_t)p(s_t)q_\circ(s_t, \mathbf{x}_t) \end{aligned} \tag{12}$$

To make (12) agree with (11), we need to define $q_\circ(s_t, \mathbf{x}_t)$ in (12) so that it is factorized:

$$q_\circ(s_t, \mathbf{x}_t) = q_\circ(s_t)q_\circ(\mathbf{x}_t) \tag{13}$$

Similarly, we need to define $q_{\triangleright\triangleleft}(\mathbf{x}_t, \mathbf{x}_{t+1})$ in terms of two decoupled approximation terms $q_\triangleleft(\mathbf{x}_t)$ and $q_\triangleright(\mathbf{x}_{t+1})$, i.e.,

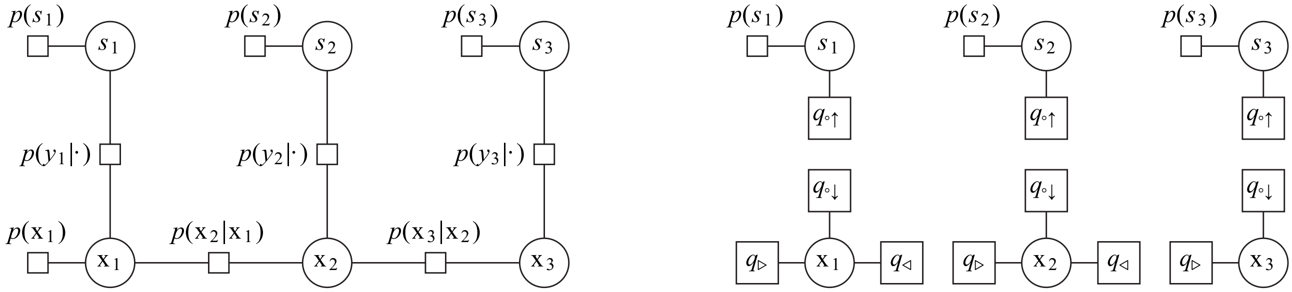
$$q_{\triangleright\triangleleft}(\mathbf{x}_t, \mathbf{x}_{t+1}) = q_\triangleleft(\mathbf{x}_t)q_\triangleright(\mathbf{x}_{t+1}) \tag{14}$$

Set $q_\triangleright(\mathbf{x}_1) \equiv p(\mathbf{x}_1)$ and $q_\triangleleft(\mathbf{x}_T) \equiv 1$. Then inserting (13) and (14) into (12) yields the following approximate joint posterior:

$$q(s_{1:T}, \mathbf{x}_{1:T}) \propto \prod_{t=1}^{T} q_\triangleright(\mathbf{x}_t)p(s_t)q_\circ(s_t)q_\circ(\mathbf{x}_t)q_\triangleleft(\mathbf{x}_t) \tag{15}$$

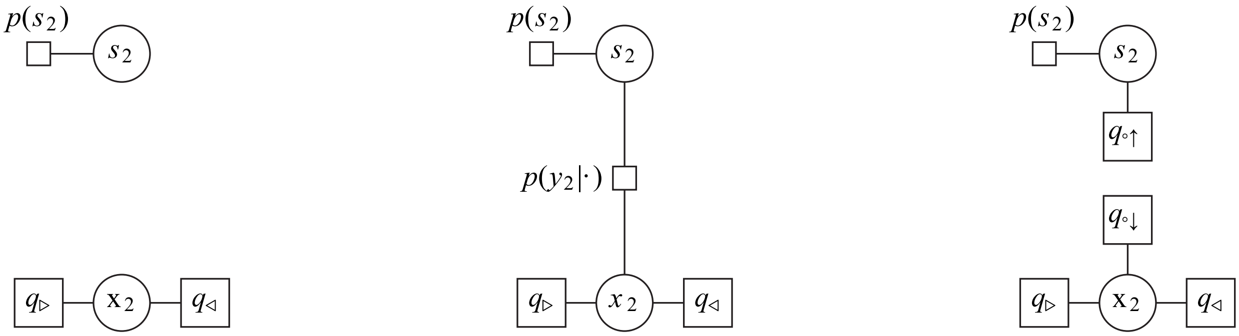
Clearly, the approximate state posterior $q(s_t, \mathbf{x}_t)$ is

$$q(s_t, \mathbf{x}_t) \propto q_\triangleright(\mathbf{x}_t)p(s_t)q_\circ(s_t)q_\circ(\mathbf{x}_t)q_\triangleleft(\mathbf{x}_t) \tag{16}$$



(a) Exact posterior $p(s_1, s_2, s_3, \mathrm{x}_1, \mathrm{x}_2, \mathrm{x}_3)$ (b) Approximate posterior $\prod_i q(s_i)q(\mathrm{x}_i)$

Fig. 2. Factor graph representations of exact and approximate posterior distributions. The posteriors of the variables (circles) are proportional to the product of factors (rectangles). Here $q_{\circ\uparrow} \equiv q_\circ(s_t)$, and $q_{\circ\downarrow} \equiv q_\circ(\mathbf{x}_t)$.



(a) $p(s_2)q_{\triangleright\triangleleft}(\mathrm{x}_2)$ (b) $p(y_2|s_2, \mathrm{x}_2)p(s_2)q_{\triangleright\triangleleft}(x_2)$ (c) $q(s_2)q(\mathrm{x}_2)$

Fig. 3. Illustration of the three steps when EP processes the observation distribution. The partial belief for $\mathbf{x}_2$ is $q_{\triangleright\triangleleft}(\mathbf{x}_2) = q_{\triangleright}(\mathbf{x}_2)q_{\triangleleft}(\mathbf{x}_2)$. The partial belief for s_2 is $p(s_2)$. The factors $q_{\circ\uparrow} \equiv q_\circ(s_t)$ and $q_{\circ\downarrow} \equiv q_\circ(\mathbf{x}_t)$ are *observation messages* from y_t to $(s_t, \mathbf{x}_t)$.

We visualize this approximation in Figure 2b, where the probability of each variable (circle) is the product of the approximation terms (boxes) connected to the variable.

Each approximation term is chosen to be log-linear, e.g.,

$$q_\circ(\mathbf{x}_t) \propto \exp(\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{\varphi}(\mathbf{x}_t)) \tag{17}$$

where we call the vector $\boldsymbol{\beta}$ the natural parameter of $q_\circ(\mathbf{x}_t)$. Note that $q_\circ(\mathbf{x}_t)$ is not necessarily a probability density function. Since the approximate state posterior $q(s_t, \mathbf{x}_t)$ equals the product of the approximation terms, it is clearly log-linear too, i.e., it is in the exponential family. More specifically, as shown in the next section, $q(s_t, \mathbf{x}_t)$ is a product of a Gaussian distribution and a discrete distribution, which is in the exponential family.

We can interpret these approximation terms as messages that propagate in the dynamic system: $q_\circ(s_t)q_\circ(\mathbf{x}_t)$ is an *observation message* from y_t to $(s_t, \mathbf{x}_t)$, $q_{\triangleright}(\mathbf{x}_{t+1})$ is a *forward message* from $\mathbf{x}_t$ to $\mathbf{x}_{t+1}$, and $q_{\triangleleft}(\mathbf{x}_t)$ is a *backward message* from $\mathbf{x}_{t+1}$ to $\mathbf{x}_t$. Accordingly, the approximate state belief (16) is a product of incoming messages. Therefore the key of this approximation is the computation of these messages.

The iterative computation of $q_\circ(s_t)q_\circ(\mathbf{x}_t)$ as defined by EP is illustrated in Figure 3. First, compute the *partial belief* $p(s_t)q_{\triangleright\triangleleft}(\mathbf{x}_t)$ defined by: (Figure 3a)

$$q_{\triangleright\triangleleft}(\mathbf{x}_t) = q(\mathbf{x}_t)/q_\circ(\mathbf{x}_t) = q_{\triangleright}(\mathbf{x}_t)q_{\triangleleft}(\mathbf{x}_t) \tag{18}$$

(Note that $q(\mathbf{x}_t) \equiv q_{\triangleright\circ\triangleleft}(\mathbf{x}_t)$. For simplicity, we suppress this subscript. Also, the partial belief $q_{\triangleright\triangleleft}(\mathbf{x}_t)$ is analogous to a message from the variable $\mathbf{x}_t$ to the factor $p(y_t|s_t, \mathbf{x}_t)$ in the sum-product algorithm.) After removing the observation message $q_\circ(s_t)$, the partial belief for s_t is $p(s_t)$, since $q(s_t) = q_\circ(s_t)p(s_t)$. After putting back the exact term $p(y_t|s_t, \mathbf{x}_t)$ in the picture (Figure 3b), minimize the following KL divergence over the approximate posterior $q(s_t)q(\mathbf{x}_t)$ (Figure 3c):

$$\mathrm{KL}(p(y_t|s_t, \mathbf{x}_t)p(s_t)q_{\triangleright\triangleleft}(\mathbf{x}_t) \parallel q(s_t)q(\mathbf{x}_t)), \tag{19}$$

and obtain the new messages:

$$q_\circ(s_t)^{\text{new}} \propto q(s_t)/p(s_t) \quad q_\circ(\mathbf{x}_t)^{\text{new}} \propto q(\mathbf{x}_t)/q_{\triangleright\triangleleft}(\mathbf{x}_t) \tag{20}$$

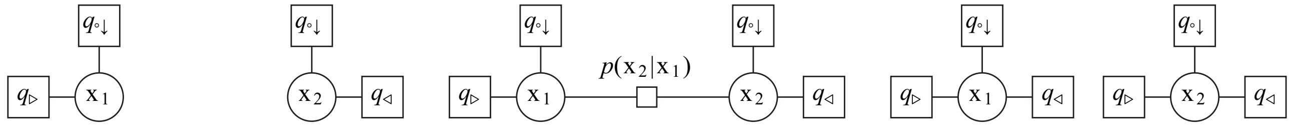
The KL minimization step is the same as assumed density filtering [11], [12]. Computationally, the KL minimization amounts to matching moments between $p(y_t|s_t, \mathbf{x}_t)p(s_t)q_{\triangleright\triangleleft}(\mathbf{x}_t)$ and $q(s_t)q(\mathbf{x}_t)$. Notice that we are minimizing divergence between variable distributions, rather than divergence between messages. This makes sense because the variable distributions are what we really care about in the problem.

The forward and backward messages are determined in a similar way, illustrated in Figure 4. First, compute the partial belief $q_{\triangleright\circ}(\mathbf{x}_{t-1})q_{\circ\triangleleft}(\mathbf{x}_t)$, defined as:

$$q_{\triangleright\circ}(\mathbf{x}_{t-1}) = q_{\triangleright}(\mathbf{x}_{t-1})q_\circ(\mathbf{x}_{t-1}) \quad q_{\circ\triangleleft}(\mathbf{x}_t) = q_\circ(\mathbf{x}_t)q_{\triangleleft}(\mathbf{x}_t) \tag{21}$$

Then minimize the KL divergence

$$\mathrm{KL}(q_{\triangleright\circ}(\mathbf{x}_{t-1})p(\mathbf{x}_t|\mathbf{x}_{t-1})q_{\circ\triangleleft}(\mathbf{x}_t) \parallel q(\mathbf{x}_{t-1})q(\mathbf{x}_t)) \tag{22}$$



(a) $q_{\triangleright\circ}(\mathrm{x}_1)q_{\circ\triangleleft}(\mathrm{x}_2)$ (b) $q_{\triangleright\circ}(\mathrm{x}_1)p(\mathrm{x}_2|\mathrm{x}_1)q_{\circ\triangleleft}(\mathrm{x}_2)$ (c) $q(\mathrm{x}_1)q(\mathrm{x}_2)$

Fig. 4. Illustration of the three steps when EP processes the transition distribution. The factors $q_{\triangleright}$ and $q_{\triangleleft}$ are *forward* and *backward messages*. The partial beliefs $q_{\triangleright\circ}(\mathbf{x}_1) = q_{\triangleright}(\mathbf{x}_1)q_{\circ}(\mathbf{x}_1)$ and $q_{\circ\triangleleft}(\mathbf{x}_2) = q_{\circ}(\mathbf{x}_2)q_{\triangleleft}(\mathbf{x}_2)$.

over $q(\mathbf{x}_{t-1})q(\mathbf{x}_t)$, giving the new forward message $q_{\triangleright}(\mathbf{x}_t)^{\text{new}} = q(\mathbf{x}_t)/q_{\circ\triangleleft}(\mathbf{x}_t)$ and the backward message $q_{\triangleleft}(\mathbf{x}_{t-1})^{\text{new}} = q(\mathbf{x}_{t-1})/q_{\triangleright\circ}(\mathbf{x}_{t-1})$. Since the transition probability distributions are Gaussians, the answer to the above KL minimization is the exact marginals of the pairwise joint. Instead of computing the forward and backward messages simultaneously as above, it is equally valid to run separate forward filtering and backward smoothing passes. Specifically, at time t in a forward pass, we compute only the forward message $q_{\triangleright}(\mathbf{x}_{t+1})$ and $q(\mathbf{x}_{t+1})$, while at time t in a backward pass, we compute only the backward message and the new $q(\mathbf{x}_t)$.

The following sections describe in detail how to incorporate the observation, forward, and backward messages.

A. Compute and Incorporate Observation Messages

First, consider how to update the state belief $q(s_t)q(\mathbf{x}_t)$ using the observation data and, correspondingly, how to generate the observation message $q_{\circ}(s_t, \mathbf{x}_t)$.

The messages $q_{\circ}(\mathbf{x}_t)$, $q_{\triangleright}(\mathbf{x}_t)$, and $q_{\triangleleft}(\mathbf{x}_t)$ are chosen to be Gaussian, while $q_{\circ}(s_t)$ is discrete. From (18), the partial belief $q_{\triangleright\triangleleft}(\mathbf{x}_t)$ is therefore Gaussian:

$$q_{\triangleright\triangleleft}(\mathbf{x}_t) = \mathcal{N}_c(\mathbf{x}_t|\mathbf{m}_{\mathbf{x}_t\triangleright\triangleleft}, \mathbf{V}_{\mathbf{x}_t\triangleright\triangleleft})$$

where $\mathcal{N}_c(\cdot|\mathbf{m}_{\mathbf{x}_t\triangleright\triangleleft}, \mathbf{V}_{\mathbf{x}_t\triangleright\triangleleft})$ is the probability density function of a complex Gaussian with mean of $\mathbf{m}_{\mathbf{x}_t\triangleright\triangleleft}$ and variance of $\mathbf{V}_{\mathbf{x}_t\triangleright\triangleleft}$.

Given $p(s_t)q_{\triangleright\triangleleft}(\mathbf{x}_t)$ and $p(y_t|s_t, \mathbf{x}_t)$, we compute an intermediate approximate posterior $\hat{q}(s_t, \mathbf{x}_t)$:

$$\hat{q}(s_t, \mathbf{x}_t) = \frac{p(y_t|s_t, \mathbf{x}_t)p(s_t)q_{\triangleright\triangleleft}(\mathbf{x}_t)}{\int_{s_t,\mathbf{x}_t} p(y_t|s_t, \mathbf{x}_t)p(s_t)q_{\triangleright\triangleleft}(\mathbf{x}_t)} \tag{23}$$

Define the normalization constant of $\hat{q}(s_t, \mathbf{x}_t)$ as z:

$$\begin{aligned} z &= \sum_{s_t\in\mathcal{A}} \int p(y_t|s_t, \mathbf{x}_t)p(s_t)q_{\triangleright\triangleleft}(\mathbf{x}_t)\mathrm{d}\mathbf{x}_t \\ &= \sum_{s_t\in\mathcal{A}} p(s_t) \int \mathcal{N}_c(y_t|s_t\mathbf{h}^\star\mathbf{x}_t, \sigma^2)\mathcal{N}_c(\mathbf{x}_t|\mathbf{m}_{\mathbf{x}_t\triangleright\triangleleft}, \mathbf{V}_{\mathbf{x}_t\triangleright\triangleleft})\mathrm{d}\mathbf{x}_t \\ &= \sum_{s_t\in\mathcal{A}} p(s_t)\mathcal{N}(y_t|m_{y_t}(s_t), v_{y_t}(s_t)) \end{aligned} \tag{24}$$

where

$$m_{y_t}(s_t) = s_t\mathbf{h}^\star\mathbf{m}_{\mathbf{x}_t\triangleright\triangleleft}, \qquad v_{y_t}(s_t) = s_t\mathbf{h}^\star\mathbf{V}_{\mathbf{x}_t\triangleright\triangleleft}\mathbf{h}s_t^\star + \sigma^2 \tag{25}$$

Since $\hat{q}(s_t, \mathbf{x}_t)$ is not in the exponential family and, therefore, it is difficult to keep updating the state belief in the dynamic model analytically and efficiently if we keep $\hat{q}(s_t, \mathbf{x}_t)$ as the new belief of $(s_t, \mathbf{x}_t)$. To solve this problem, we project $\hat{q}(s_t, \mathbf{x}_t)$ into $q(\mathbf{x}_t)q(s_t)$:

$$q(\mathbf{x}_t) = \mathcal{N}_c(\mathbf{x}_t|\mathbf{m}_{\mathbf{x}_t}, \mathbf{V}_{\mathbf{x}_t}) \tag{26}$$

$$q(s_t) = \text{Discrete}_M(s_t) \tag{27}$$

by minimizing the KL divergence (19).

For this minimization, we match the moments of $\hat{q}$ and q such that

$$q(s_t) = \frac{p(s_t)\mathcal{N}(y_t|s_t\mathbf{h}^\star\mathbf{m}_{\mathbf{x}_t\triangleright\triangleleft}, v_{y_t}(s_t))}{z} \tag{28}$$

$$\mathbf{m}_{\mathbf{x}_t} = \frac{\sum_{s_t\in\mathcal{A}} p(s_t)\mathcal{N}(y_t|m_{y_t}(s_t), v_{y_t}(s_t))\mathbf{m}_{\mathbf{x}_t}(s_t)}{z} \tag{29}$$

$$\begin{aligned} \mathbf{V}_{\mathbf{x}_t} &= \mathbf{V}_{\mathbf{x}_t|y_t} - \mathbf{m}_{\mathbf{x}_t}\mathbf{m}_{\mathbf{x}_t}^\star + \frac{1}{z}\cdot \\ &\cdot \sum_{s_t\in\mathcal{A}} p(s_t)\mathcal{N}(y_t|m_{y_t}(s_t), v_{y_t}(s_t))\mathbf{m}_{\mathbf{x}_t|s_t}(s_t)\mathbf{m}_{\mathbf{x}_t|y_t}(s_t)^\star \end{aligned} \tag{30}$$

where

$$\mathbf{m}_{\mathbf{x}_t|y_t}(s_t) = \mathbf{m}_{\mathbf{x}_t\triangleright\triangleleft} + \mathbf{k}(s_t)(y_t - s_t\mathbf{h}^\star\mathbf{m}_{\mathbf{x}_t\triangleright\triangleleft}) \tag{31}$$

$$\mathbf{V}_{\mathbf{x}_t|y_t} = \mathbf{V}_{\mathbf{x}_t\triangleright\triangleleft} - \mathbf{k}(s_t)s_t\mathbf{h}^\star\mathbf{V}_{\mathbf{x}_t\triangleright\triangleleft} \tag{32}$$

$$\mathbf{k}(s_t) = \mathbf{V}_{\mathbf{x}_t\triangleright\triangleleft}\mathbf{h}s_t^\star v_{y_t}(s_t)^{-1} \tag{33}$$

Note that $\mathbf{V}_{\mathbf{x}_t|y_t}$ is not a function of s_t since $s_t^\star s_t = 1$.

Then from (20), it follows that

$$q_{\circ}(\mathbf{x}_t) \propto \mathcal{N}(\mathbf{x}_t|\mathbf{m}_{\mathbf{x}_t\circ}, \mathbf{V}_{\mathbf{x}_t\circ}) \tag{34}$$

where

$$\mathbf{m}_{\mathbf{x}_t\circ} = \mathbf{V}_{\mathbf{x}_t\circ}(\mathbf{V}_{\mathbf{x}_t}^{-1}\mathbf{m}_{\mathbf{x}_t} - \mathbf{V}_{\mathbf{x}_t\triangleright\triangleleft}^{-1}\mathbf{m}_{\mathbf{x}_t\triangleright\triangleleft}) \tag{35}$$

$$\mathbf{V}_{\mathbf{x}_t\circ} = (\mathbf{V}_{\mathbf{x}_t}^{-1} - \mathbf{V}_{\mathbf{x}_t\triangleright\triangleleft}^{-1})^{-1} \tag{36}$$

Since the observation message is not necessarily a valid probability distribution, $\mathbf{\Lambda}_{\mathbf{x}_t\circ}$ can be near singular, such that inverting $\mathbf{\Lambda}_{\mathbf{x}_t\circ}$ loses numerical accuracy. To avoid numerical problems, we transform $\mathbf{m}_{\mathbf{x}_t\circ}$ and $\mathbf{V}_{\mathbf{x}_t\circ}$ to the natural parameters of $q_{\circ}(\mathbf{x}_t)$:

$$\boldsymbol{\mu}_{\mathbf{x}_t\circ} = \mathbf{V}_{\mathbf{x}_t\circ}^{-1}\mathbf{m}_{\mathbf{x}_t\circ} = \mathbf{V}_{\mathbf{x}_t}^{-1}\mathbf{m}_{\mathbf{x}_t} - \mathbf{V}_{\mathbf{x}_t\triangleright\triangleleft}^{-1}\mathbf{m}_{\mathbf{x}_t\triangleright\triangleleft} \tag{37}$$

$$\mathbf{\Lambda}_{\mathbf{x}_t\circ} = \mathbf{V}_{\mathbf{x}_t\circ}^{-1} = \mathbf{V}_{\mathbf{x}_t}^{-1} - \mathbf{V}_{\mathbf{x}_t\triangleright\triangleleft}^{-1} \tag{38}$$

B. Incorporating Forward and Backward Messages

Because s_t and s_{t+1} are not directly connected, forward and backward messages are only sent for $\mathbf{x}_t$.

1) Compute the forward message $q_{\triangleright}(\mathbf{x}_t)$ given $q_{\triangleright\circ}(\mathbf{x}_{t-1}) = \mathcal{N}_c(\mathbf{x}_{t-1}|\mathbf{m}_{\mathbf{x}_{t-1}\triangleright\circ}, \mathbf{V}_{\mathbf{x}_{t-1}\triangleright\circ})$, the partial belief of $\mathbf{x}_{t-1}$ before incorporating the backward message. Since the transition density is Gaussian, it is

easy to compute $q_{\triangleright}(\mathbf{x}_t) = \mathcal{N}_c(\mathbf{x}_t|\mathbf{m}_{\mathbf{x}_t\triangleright}, \mathbf{V}_{\mathbf{x}_t\triangleright})$ as in Kalman filtering:

$$\mathbf{m}_{\mathbf{x}_t\triangleright} = \mathbf{F}\mathbf{m}_{\mathbf{x}_{t-1}\triangleright\circ} \qquad \mathbf{V}_{\mathbf{x}_t\triangleright} = \mathbf{F}\mathbf{V}_{\mathbf{x}_{t-1}\triangleright\circ}\mathbf{F}^\star + \mathbf{g}\mathbf{g}^\star \tag{39}$$

where $\mathbf{m}_{\mathbf{x}_1\triangleright}$ and $\mathbf{V}_{\mathbf{x}_1\triangleright}$ are the parameters of the Gaussian prior $p(\mathbf{x}_1)$.

2) Incorporate the backward message $q_{\triangleleft}(\mathbf{x}_t)$ given the partial belief $q_{\triangleright\circ}(\mathbf{x}_t)$. Without explicitly computing the backward message $q_{\triangleleft}(\mathbf{x}_t)$, we can directly incorporate $q_{\triangleleft}(\mathbf{x}_t)$ into $q(\mathbf{x}_t)$ as in Kalman smoothing:

$$\mathbf{J}_t = \mathbf{V}_{\mathbf{x}_t\triangleright\circ}\mathbf{F}^\star\mathbf{V}^{-1}_{\mathbf{x}_{t+1}\triangleright} \tag{40}$$
$$\mathbf{m}_{\mathbf{x}_t} = \mathbf{m}_{\mathbf{x}_t\triangleright\circ} + \mathbf{J}_t(\mathbf{m}_{\mathbf{x}_{t+1}} - \mathbf{F}\mathbf{m}_{\mathbf{x}_t\triangleright\circ}) \tag{41}$$
$$\mathbf{V}_{\mathbf{x}_t} = \mathbf{V}_{\mathbf{x}_t\triangleright\circ} + \mathbf{J}_t(\mathbf{V}_{\mathbf{x}_{t+1}}\mathbf{J}_t^\star - \mathbf{F}\mathbf{V}^\star_{\mathbf{x}_t\triangleright\circ}) \tag{42}$$

Note that the forward and backward messages calculated as the above minimize the KL divergence (22). This suggests that the Kalman filtering and smoothing steps are the natural outcome of EP updates.

C. Algorithm Summary

Given the knowledge of how to incorporate different messages, we are ready to construct the whole expectation propagation algorithm by establishing the iteration mechanism.

1) Loop $t = 1 : T$:
   a) Compute the forward message $q_{\triangleright}(\mathbf{x}_t)$ via (39) and set the backward message to a constant, i.e., $q_{\triangleleft}(\mathbf{x}_t) \propto 1$.
   b) Update $q(s_t, \mathbf{x}_t)$ to match the moments of $\hat{q}(s_t, \mathbf{x}_t) = p(y_t|s_t, \mathbf{x}_t)p(s_t)q_{\triangleright}(\mathbf{x}_t)$ via (28) to (30).
   c) Compute $q_{\circ}(\mathbf{x}_t) \propto q(\mathbf{x}_t)/q_{\triangleright\triangleleft}(\mathbf{x}_t)$ via (37) to (38).

2) Loop until convergence or the maximal number of iterations n has been achieved:
   a) Loop $t = 1, \ldots, T$ (Skip on the first iteration)
      i) Compute the forward message $q_{\triangleright}(\mathbf{x}_t)$ via (39).
      ii) Compute the partial belief $q_{\triangleright\circ}(\mathbf{x}_t)$ given the forward and observation messages.:

$$q_{\triangleright\circ}(\mathbf{x}_t) = q_{\triangleright}(\mathbf{x}_t)q_{\circ}(\mathbf{x}_t)$$

      This can be easily accomplished as follows:

$$\mathbf{V}_{\mathbf{x}_t\triangleright\circ} = (\mathbf{V}^{-1}_{\mathbf{x}_t\triangleright} + \mathbf{\Lambda}_{\mathbf{x}_t\circ})^{-1} \tag{43}$$
$$\mathbf{m}_{\mathbf{x}_t\triangleright\circ} = \mathbf{V}_{\mathbf{x}_t\triangleright\circ}(\mathbf{V}^{-1}_{\mathbf{x}_t\triangleright}\mathbf{m}_{\mathbf{x}_t\triangleright} + \boldsymbol{\mu}_{\mathbf{x}_t\circ}) \tag{44}$$

   b) Loop $t = T-1, \ldots, 1$
      i) Compute $q(\mathbf{x}_t)$ by incorporating the backward message via (40) to (42) when $t < T$.
      ii) Update $q(s_t, \mathbf{x}_t)$ and $q_{\circ}(s_t, \mathbf{x}_t)$ as follows:
         A) Compute the partial belief $q_{\triangleright\triangleleft}(\mathbf{x}_t)$. From (18), it follows that

$$\mathbf{m}_{\mathbf{x}_t\triangleright\triangleleft} = \mathbf{V}_{\mathbf{x}_t\triangleright\triangleleft}(\mathbf{V}^{-1}_{\mathbf{x}_t}\mathbf{m}_{\mathbf{x}_t} - \boldsymbol{\mu}_{\mathbf{x}_t\circ}), \tag{45}$$
$$\mathbf{V}_{\mathbf{x}_t\triangleright\triangleleft} = (\mathbf{V}^{-1}_{\mathbf{x}_t} - \mathbf{\Lambda}_{\mathbf{x}_t\circ})^{-1} \tag{46}$$

         B) Update $q(s_t, \mathbf{x}_t)$ to match the moments of $p(y_t|s_t, \mathbf{x}_t)p(s_t)q_{\triangleright\triangleleft}(\mathbf{x}_t)$ via (28) to (30).
         C) Compute $q_{\circ}(\mathbf{x}_t)$ via (37) to (38).

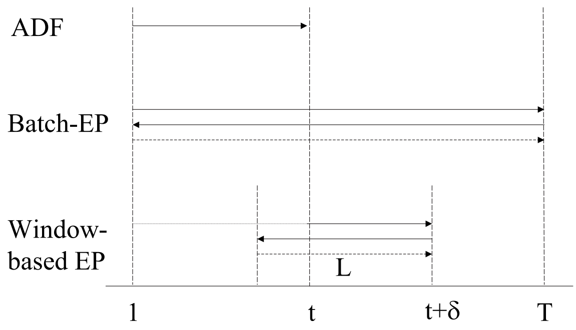


Fig. 5. Illustration of ADF, batch EP, and window-based EP. ADF sequentially processes the observation sequence to the current time t; batch EP smoothing uses the entire observation sequence from the beginning at time 1 to the end at time T; window-based EP uses the previous approximate posterior at time t as a prior distribution for filtering and performs smoothing in a sliding window with length L.

V. Window-Based EP Smoothing

For the signal detection problem, we need online processing of the data. This need cannot currently be satisfied by batch EP smoothing, which uses the entire observation sequence and therefore is computationally expensive.

To address this problem, we propose window-based EP smoothing as an alternative to batch EP smoothing. Window-based EP smoothing finds a trade-off between assumed-density filtering and batch EP smoothing. Instead of smoothing over the entire observation sequence, we use a sliding window with length L to approximate the posterior $p(s_t, \mathbf{x}_t|y_{1:t+\delta})$ based on the observations $y_{1:t+\delta} = [y_1, \ldots, y_{t+\delta}]$, where δ controls the delay for online estimation. Specifically, we first run ADF from time t to $t+\delta$, perform EP smoothing from $t+\delta$ to $t+\delta-L+1$, and then run ADF filtering from $t+\delta-L+1$ to $t+\delta$. We iterate the filtering and smoothing in the sliding window until a fixed number of iterations or the convergence has achieved. Essentially, window-based EP smoothing is a rescheduling of EP update orders for the online-estimation purpose. The difference between ADF, batch EP, and window-based EP is illustrated in Figure 5.

VI. Computational Complexity

In this section, we compare the efficiency of window-based EP smoothing with sequential Monte Carlo methods (table I). For window-based EP smoothing, the total computation time of incorporating the forward and observation messages via (39) to (44) is $O(d^2)$ (d is the dimension of $\mathbf{x}_t$), same as the cost of one-step Kalman filtering; incorporating the backward message via (40) to (42) takes $O(d^2)$ as Kalman smoothing; and finally updating $q(s_t, \mathbf{x}_t)$ and $q_{\circ}(s_t, \mathbf{x}_t)$ in step 2(b)ii costs $O(Md^2)$, M times as the cost of Kalman filtering, where M is the size of alphabet for symbols. Furthermore, since in general the estimation accuracy is not increasing after a few propagation iterations, the needed number of EP iterations n is small. In our experiments, we set $n = 5$.

In contrast, if we use m samples in a stochastic mixture of Kalman filters, also known as Rao-blackwellised particle smoothers, it takes $O(mMd^2)$ for a one-step update, and $O(mM^Ld^2)$ for L step smoothing, which is mM^L times the cost of one-step Kalman filtering and smoothing [6]. Another

TABLE I
COST OF EP VERSUS SEQUENTIAL MONTE CARLO.

EP	$O(nMLd^2T)$
[6]	$O(mM^Ld^2T)$
[18]	$O(mkMLd^2T)$
n	number of EP iterations
M	size of symbol alphabet
L	length of the smoothing window
d	dimension of $\mathbf{x}_t$
m	number of forward samples
k	number of backward samples
T	length of observation sequence

version of efficient particle smoothers [18], which is applied to audio processing, takes $O(mkMLd^2)$ where m and k are the numbers of forward and backward samples. To achieve accurate estimation, sequential Monte Carlo methods generally need a large number of samples, i.e., large m and k.

VII. EXPERIMENTAL RESULTS ON SIGNAL DETECTION

In this section, we apply the proposed window-based EP smoothing algorithm to the signal detection problem. The flat-fading channels are defined in (4) and(5), and the proposed algorithm decodes the symbols s_t as

$$\hat{s}_t = \operatorname{argmax} q(s_t) \tag{47}$$

where $q(s_t)$ is obtained after the convergence of the expectation propagation algorithm.

We demonstrate the high performance of the window-based EP receiver in a flat-fading channel with different signal noise ratios. We model the fading coefficients $\{\alpha_t\}$ by the following ARMA(3,3) model, as in [6]: $\Phi = [-2.37409 \quad 1.92936 \quad -0.53208]$, $\Theta = 0.01 \times [0.89409 \quad 2.68227 \quad 2.68227 \quad 0.89409]$, $v_t \sim \mathcal{N}_c(0,1)$. With these parameters, we have $\mathrm{Var}\{\alpha_t\} = 1$. BPSK modulation is employed ($s_t \in \{1, -1\}$). Also, differential encoding and decoding are employed to resolve the phase ambiguity. The prior $p(s_t) = 1/2$ for all t.

We test the window-based EP receiver with different window lengths $L = 1, 2, 4$, with $0, 1, 3$ overlap points, respectively. In other words, the estimation time delay δ equals 0,1,and 3, respectively. Note that the current detector with $\delta = 0$ performs ADF as a degenerate case of EP smoothing. Moreover, we run the batch EP receiver with smoothing over the entire data sequence.

For comparison, we test a genie-aided lower bound and a differential detector. For the genie-aided detection, an additional observation is provided, which is another transmitted signal where the symbol is always 1, i.e., $\tilde{y}_t = \alpha_t + w_t$. The receiver employs a Kalman filter to estimate the posterior mean $\hat{\alpha}_t$ of the fading process, based on the new observation sequence $\{\tilde{y}_t\}$. The symbols are then demodulated according to $\hat{s}_t = \mathrm{sign}(\mathcal{R}\{\hat{\alpha}_t^\star y_t\})$ where $\star$ means conjugate. By obtaining the extra information from the genie, this detector is expected to achieve accurate detection results. Also, the genie-aided detection is also lower bounded by the known channel bound. For the differential detection, no attempt is made for channel estimation. It simply detects the phase difference between two consecutive observations y_{t-1} and y_t: $\hat{s}_t = \mathrm{sign}(\mathcal{R}\{\hat{y}_t{}^\star y_{t-1}\})$.

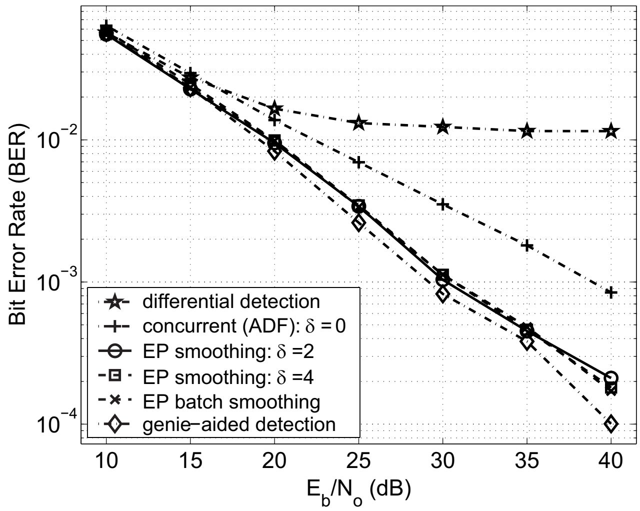


Fig. 6. BER demodulation performance of the EP receiver with different smoothing parameters, the genie-aided detector, and the differential detector in a fading channel with complex Gaussian noises.

We run these detectors on 10,000 received signals for different signal-to-noise ratios (SNRs), defined as $10\log_{10}(\mathrm{Var}\{\alpha_t\}/\mathrm{Var}\{w_t\})$. For each SNR, we randomly synthesize a new symbol sequence and a new observation sequence according to (4) and (5). We repeat this procedure 10 times. The averaged bit-error rate (BER) performance of different detectors versus SNR is plotted in Figure 6.

Clearly, the window-based EP receiver outperforms the concurrent detector and the differential detector. The concurrent detector is based on ADF, which has been shown to achieve more accurate estimation than extended Kalman filtering in general [11]. For SNRs from 10 dB to 40 dB, the new receiver does not have the error floor, while the differential detector does. Compared with the results obtained by [6], the performance of the window-based EP receiver is comparable to that of a sequential Monte Carlo receiver. However, the EP receiver is much more efficient. [6] use 50 samples in their sequential Monte Carlo receiver. With window length of 3, the window-based EP receiver is 13.33 times faster $(50 \cdot 2^3/(5 \cdot 2 \cdot 3) = 13.33)$ than the sequential Monte Carlo receiver. Furthermore, the cost of the new receiver increases linearly with the window length, while the cost of the sequential Monte Carlo receiver explodes exponentially. For example, when the window length increases to 5, the new receiver becomes 32 times faster $(50 \cdot 2^5/(5 \cdot 2 \cdot 5) = 32)$ than the sequential Monte Carlo receiver.

With a delay $\delta = 2$, the performance of the window-based EP receiver is almost as good as that of the batch EP receiver, which performs smoothing over each entire data sequence, while the window-based EP receiver is more efficient and results in much shorter estimation delay than the batch EP receiver. Moreover, the performance of the window-based EP receiver with a delay $\delta = 3$ is close to the genie-aided detector.

VIII. Discussion

This paper has presented a window-based EP smoothing algorithm for online estimation and applied it to adaptive signal detection in fading channels. We first propose the EP smoothing algorithm for hybrid dynamic systems, and then present window-based EP smoothing to achieve a trade-off between assumed density filtering and batch EP smoothing in terms of accuracy and efficiency. For the signal detection problem, the window-based EP receiver significantly outperformed both the classical differential detector and the concurrent adaptive Bayesian receiver, which is based on ADF, under different signal-to-noise ratios. Moreover, the window-based EP receiver performs comparably to the batch EP receiver, which uses the entire data sequence for smoothing and therefore leads to more estimation delay and larger computational cost. The performance of the window-based EP receiver is also close to the genie-aided detection, a performance upper bound. This performance similarity demonstrates the high quality of the window-based EP receiver. Compared to the sequential Monte Carlo receiver, the window-based EP receiver obtains the comparable estimation accuracy with less than one-tenth computational complexity. In short, for the signal detection problem, window-based EP improves the estimation accuracy over ADF, enhances the efficiency over batch EP without sacrificing accuracy, and achieves comparable accuracy as the sequential Monte Carlo methods with much lower cost.

The window-based EP receiver can be used for many online estimation problems in wireless digital communications, such as signal detection with unknown phase. The window-based EP receiver for wireless signal detection is just one example. Joint decoding and channel estimation would be a natural extension of this work. For dynamic systems with nonlinear and non-Gaussian likelihoods different from the signal detection problem, the only thing we need to modify in the algorithm is the moment matching step, defined by equations (28) to (30). Based on a different likelihood, we will have a different moment matching step. When exact moments are difficult to compute, we can use approximate techniques, for example, unscented Kalman filtering [21], to approximate the moment matching step.

In this paper, we chose a fully-factorized approximation so that a belief propagation-like algorithm would result. However, EP can generate algorithms different from belief propagation, that propagate Markov-chain structures [22].

References

[1] G. M. Vitetta and D. P. Taylor, “Maximum likelihood decoding of uncoded and coded PSK signal sequences transmitted over Rayleigh flat-fading channels,” *IEEE Trans. Commun.*, vol. 43, pp. 2750–2758, 1995.

[2] A. N. D’Andrea, A. Diglio, and U. Mengali, “Symbol-aided channel estimation with nonselective Rayleigh fading channels,” *IEEE Trans. Veh. Technol.*, vol. 44, pp. 41–48, Feb. 1995.

[3] P. Hoeher and J. H. Lodge, “Turbo DPSK: iterative differential PSK domodulation and channel decodings,” *IEEE Trans. Commun.*, vol. 47, pp. 837–843, June 1999.

[4] A. W. Eckford, “Channel estimation in block fading channels using the factor graph EM algorithm,” in *Proc. 22nd Biennial Symposium on Communications 2004*.

[5] R. Otnes and M. Tüchler, “Iterative channel estimation for turbo equalization of time-varying frequency-selective channels,” *IEEE Trans. Wireless Commun.*, vol. 3, no. 6, pp. 1918–1923, 2004.

[6] R. Chen, X. Wang, and J. S. Liu, “Adaptive joint detection and decoding in flat-fading channels via mixture Kalman filtering,” *IEEE Trans. Inf. Theory*, vol. 46, no. 6, pp. 2079–2094, Sept. 2000.

[7] X. Wang, R. Chen, and D. Guo, “Delayed-pilot sampling for mixture Kalman filter with application in fading channels,” *IEEE Trans. Signal Processing*, special issue on Monte Carlo methods for statistical signal processing, vol. 50, no. 2, pp. 241–254, Feb. 2002.

[8] A. P. Worthen and W. E. Stark, “Unified design of iterative receivers using factor graphs,” *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 843–849, 2001.

[9] G. Colavolpe, A. Barbieri, and G. Caire, “Algorithms for iterative decoding in the presence of strong phase noise,” *IEEE J. Sel. Areas Commun.*, vol. 23, no. 9, pp. 1748–1757, 2005.

[10] T. P. Minka, “Expectation propagation for approximate Bayesian inference,” in *Uncertainty in Artificial Intelligence*, 2001.

[11] H. J. Kushner and A. S. Budhiraja, “A nonlinear filtering algorithm based on an approximation of the conditional distribution,” *IEEE Trans. Autom. Control*, vol. 45, pp. 580–585, 2000.

[12] X. Boyen and D. Koller, “Tractable inference for complex stochastic processes,” in *Uncertainty in Artificial Intelligence*, 1998, vol. 11.

[13] B. J. Frey, *Graphical Models for Machine Learning and Digital Communication*. Cambridge, MA: MIT Press, 1998.

[14] R. H. Shumway and D. S. Stoffer, “Dynamic linear models with switching,” *J. The American Statistical Association*, vol. 86, pp. 763–769, 1991.

[15] C. Cargnoni, P. Mueller, and M. West, “Bayesian forecasting of multinomial time series through conditionally Gaussian dynamic models,” *J. American Statistical Association*, vol. 92, pp. 587–606, 1997.

[16] H. Tanizaki, “Nonlinear and non-Gaussian state-space modeling with Monte Carlo techniques: a survey and comparative study,” in *Handbook of Statistics: Stochastic Processes: Modeling and Simulation*, C. R. Rao and D. N. Shanbhag, Eds., chapter 22. North-Holland, 2000.

[17] A. Doucet, N. de Freitas, and N. Gordon, Eds., *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, 2001.

[18] W. Fong, S. Godsill, A. Doucet, and M. West, “Monte Carlo smoothing with application to audio signal enhancement,” *IEEE Trans. Signal Processing*, vol. 50, no. 2, Feb 2003.

[19] G. Welch and G. Bishop, “An introduction to the Kalman filter,” Tech. Report, Department of Computer Science and Engineering, University of North Carolina at Chapel Hill, March 2002.

[20] T. Heskes and O. Zoeter, “Expectation propagation for approximate inference in dynamic Bayesian networks,” in *Proc UAI’02*.

[21] E. A. Wan, R. van der Merwe, and A. T. Nelson, “Dual estimation and the unscented transformation,” in *NIPS 12*, 2000, pp. 666–672.

[22] T. P. Minka and Y. Qi, “Tree-structured approximations by expectation propagation,” in *NIPS*, 2003.