Expectation propagation for signal detection in flat-fading channels

Yuan Qi
MIT Media Lab
Cambridge, MA, 02139 USA
yuanqi@media.mit.edu

Thomas Minka
CMU Statistics Department
Pittsburgh, PA 15213 USA
minka@stat.cmu.edu

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 hybrid dynamic system that has both continuous and discrete variables. Then, an expectation propagation algorithm is proposed to address the inference problem. As an extension of belief propagation, expectation propagation efficiently approximates a Bayesian estimation by iteratively propagating information between different nodes in the dynamic system and projecting exact messages into the exponential family. Compared to sequential Monte Carlo filters and smoothers, the new method has much 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.

1 Introduction

Due to the rapid progress of various wireless services in our daily life, signal detection in flat Rayleigh fading channels has become an important research topic. Different approaches have been proposed to bear on this problem [?, ?, ?, ?, ?] Among them, sequential Monte Carlo filters and smoothers have been used in [?, ?]. The authors formulate signal detection as an estimation problem in a hybrid dynamic system that has both continuous and discrete variables, and then apply sequential Monte-Carlo filters and smoothers , which draw samples from a subpace that is marginalized over continuous variables, to approximate the Bayesian estimation. Excellent simulation performance has been achieved in [?, ?]. However, despite their capability to obtain accurate estimation, Monte Carlo filters and smoothers require high computational complexity.

In this paper, we develop an expectation propagation (EP) algorithm for hybrid dynamic systems and apply it to signal detection in flat-fading channels. The general expectation propagation algorithm, a powerful extension of belief propagation, was proposed in the statistical machine learning community [?, ?]. Belief propagation has been widely used in digital communications, such as Turbo decoding [?]. However, belief propagation can handle only discrete distributions or continuous distributions in the exponential family. In contrast, expectation propagation not only iteratively propagates the information in the dynamic system, but also projects exact messages into the exponential family. As a result, expectation propagation can efficiently approximate Bayesian integrals for a variety of distributions. Compared to sequential Monte Carlo filters and smoothers, expectation propagation has much lower computational complexity since it makes analytically deterministic approximation, instead of Monte Carlo approximations. Our simulations demonstrate that the new EP receiver achieves accurate detection without the aid of any training symbols or decision feedbacks.

2 Problem Formulation

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

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

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 $\mathcal{N}_c(0, \sigma^2)$ respectively. The symbols s_t take values from a finite alphabet set $\mathcal{A} = \{a_i\}_{i=1}^M$. The fading coefficients α_t can be modeled by an complex autoregressive moving-average

(ARMA) process as follows:

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

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, we only consider the uncoded case, where each alphabet in $\mathcal{A}$ has an equal probability prior. For this case, the communication system can be rewritten as a state-space model:

$$\mathbf{x}_t = \mathbf{F}\mathbf{x}_{t-1} + \mathbf{g}_t v_t \quad (2)$$

$$y_t = s_t \mathbf{h}^H \mathbf{x}_t + w_t \quad (3)$$

where

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

$$\mathbf{h} = [\theta_0, \theta_1, \dots, \theta_\rho]^H,$$

and the dimension of $\mathbf{x}$ is $d = \rho + 1$. Note that H means hermitian transpose.

The signal detection problem can then be formulated as an inference problem in the dynamic system defined by (2) and (3). We address this problem by a Bayesian approach; that is, we update the posterior $p(s_t, \mathbf{x}_t | y_{1:t+L-1})$ based on the observations $y_{1:t+L-1} = [y_1, \dots, y_{t+L-1}]$. We set $L > 0$ for smoothing to improve the estimation accuracy.

If a dynamic system is linear and has Gaussian noises, then we can use Kalman filtering and smoothing to efficiently compute the posterior $p(s_t, \mathbf{x}_t | y_{1:t+L-1})$. Otherwise, we need to resort to other advanced techniques to approximate the posterior. Instead of using the sequential Monte Carlo method [?, ?], we utilize expectation propagation to efficiently approximate the posterior $p(s_t, \mathbf{x}_t | y_{1:t+L-1})$.

3 Expectation Propagation for Hybrid Dynamic System

In this section, we develop the expectation propagation algorithm for the hybrid dynamic system (2) and (3).

Expectation propagation exploits the fact that the likelihood is a product of simple terms. If we approximate each of these terms well, we can get a good approximation to the posterior. Expectation propagation chooses each approximation such that the posterior using the term exactly and the posterior using the term approximately are close in KL-divergence. This gives a system of coupled equations for the approximations which are iterated to reach a fixed-point.

Specifically speaking, the exact posterior distribution is proportional to a product of observation densities and transition densities as follows

$$o(s_t, \mathbf{x}_t) = p(y_t | s_t, \mathbf{x}_t) \quad (4)$$

$$g_t(s_t, \mathbf{x}_t, s_{t+1}, \mathbf{x}_{t+1}) = p(s_{t+1}, \mathbf{x}_{t+1} | s_t, \mathbf{x}_t) \quad (5)$$

$$= p(\mathbf{x}_{t+1} | \mathbf{x}_t) \quad (6)$$

$$p(s_{1:T}, \mathbf{x}_{1:T} | y_{1:T}) \propto p(s_1, \mathbf{x}_1) o(s_1, \mathbf{x}_1) \cdot \cdot \prod_{i=1}^{T} g_{t-1}(s_{t-1}, \mathbf{x}_{t-1}, s_t, \mathbf{x}_t) o(s_t, \mathbf{x}_t) \quad (7)$$

Equation (6) holds because s_t's are independent to each other at different times in the dynamic model.

Then we approximate the posterior by the product of the independent terms:

$$p(s_{1:T}, \mathbf{x}_{1:T} | y_{1:y_T}) \approx \prod_{i=1}^{T} q(s_i, \mathbf{x}_i) \quad (8)$$

Correspondingly, the terms $o(s_t, \mathbf{x}_t)$ and $g_t(s_t, \mathbf{x}_t, s_{t+1}, \mathbf{x}_{t+1})$ in (7) are approximated by $\tilde{o}(s_t, \mathbf{x}_t)$ and $\tilde{g}_t(s_t, \mathbf{x}_t, s_{t+1}, \mathbf{x}_{t+1})$. To decouple the states in (7), we set

$$\tilde{g}_t(s_t, \mathbf{x}_t, s_{t+1}, \mathbf{x}_{t+1}) = \tilde{g}_t(s_t, \mathbf{x}_t) \tilde{g}_t(s_{t+1}, \mathbf{x}_{t+1}).$$

We can interpret these approximation terms as messages that propagate in the dynamic system: $\tilde{o}(s_t, \mathbf{x}_t)$ is an *observation message* from y_t to $(s_t, \mathbf{x}_t)$, $\tilde{g}_t(s_{t+1}, \mathbf{x}_{t+1})$ a *forward message* from $(s_t, \mathbf{x}_t)$ to $(s_{t+1}, \mathbf{x}_{t+1})$, and $\tilde{g}_t(s_t, \mathbf{x}_t)$ a *backward message* from $(s_{t+1}, \mathbf{x}_{t+1})$ to $(s_t, \mathbf{x}_t)$. After all of these approximations are made, each approximate state posterior is a product of three messages:

$$q(s_t, \mathbf{x}_t) = \tilde{g}_{t-1}(s_t, \mathbf{x}_t) \tilde{o}(s_t, \mathbf{x}_t) \tilde{g}_t(s_t, \mathbf{x}_t)$$
$$= (\text{forward})(\text{observation})(\text{backward})$$

In the following sections, we describe how to compute and incorporate these messages.

3.1 Moment matching and observation message update

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

Denote by $q^{\backslash o}(s_t, \mathbf{x}_t)$ the belief of the state $(s_t, \mathbf{x}_t)$ before incorporating the observation message. We assume $\mathbf{x}_t$ and s_t are both statistically independent and in the exponential family, so that

$$q^{\backslash o}(s_t, \mathbf{x}_t) = q^{\backslash o}(\mathbf{x}_t) q^{\backslash o}(s_t)$$
$$q^{\backslash o}(\mathbf{x}_t) \sim N_c(\mathbf{m}_t^{\backslash o}, V_t^{\backslash o})$$
$$q^{\backslash o}(s_t) \sim \text{Discrete}(p_{t,1}^{\backslash o}, p_{t,2}^{\backslash o}, \ldots, p_{t,M}^{\backslash o})$$

where $p_{t,i}^{\backslash o}$ is shorthand of $p(s_t = a_i)$.

Given $q^{\backslash o}(\mathbf{x}_t) q^{\backslash o}(s_t)$ and $o(s_t, \mathbf{x}_t)$, we can obtain the posterior $\hat{q}(s_t, \mathbf{x}_t)$:

$$\hat{q}(\mathbf{x}_t, s_t) = \frac{o(\mathbf{x}_t, s_t) q^{\backslash o}(\mathbf{x}_t, s_t)}{\int_{\mathbf{x}_t, s_t} o(\mathbf{s}_t) q^{\backslash o}(\mathbf{x}_t, s_t)} \quad (9)$$

Define $Z = \int_{\mathbf{x}_t, s_t} o(\mathbf{s}_t) q^{\backslash \mathbf{o}}(\mathbf{x}_t, s_t)$. Then it follows that

$$Z = \sum_{s_t \in \mathcal{A}} q^{\backslash o}(s_t) \int_{\mathbf{x}_t} \mathcal{N}_c(y_t | s_t h^H \mathbf{x}_t, \sigma^2) N_c(\mathbf{x}_t | \mathbf{m}_t^{\backslash o}, V_t^{\backslash o})$$
$$= \sum_{s_t \in \mathcal{A}} q^{\backslash o}(s_t) \mathcal{N}(y_t | \mathbf{m}_{y_t}, V_{y_t}) \quad (10)$$

where

$$\mathbf{m}_{y_t} = s_t h^H \mathbf{m}_t^{\backslash o}, \quad (11)$$
$$V_{y_t} = s_t h^H V_t^{\backslash o} h s_t^H + \sigma^2, \quad (12)$$

and $N_c(\cdot | \mathbf{m}_t^{\backslash o}, V_t^{\backslash o})$ is the probability density function of a Gaussian with mean of $\mathbf{m}_t^{\backslash o}$ and variance of $V_t^{\backslash o}$.

However, we cannot keep $\hat{q}(\mathbf{x}_t, s_t)$ as the new belief of $(s_t, \mathbf{x}_t)$ for message propagation. The reason is that $\hat{q}(\mathbf{x}_t, s_t)$ is not in the exponential family and, therefore, we cannot keep updating the state belief in the dynamic model analytically and efficiently. To solve this problem, we project $\hat{q}(\mathbf{x}_t, s_t)$ into an approximate distribution $q(\mathbf{x}_t, s_t)$ in the exponential family:

$$q(s_t, \mathbf{x}_t) = q(\mathbf{x}_t) q(s_t) \quad (13)$$
$$q(\mathbf{x}_t) \sim \mathcal{N}_c(\mathbf{m}_t, V_t) \quad (14)$$
$$q(s_t) \sim \text{Discrete}(p_{t,1}, p_{t,2}, \ldots, p_{t,M}) \quad (15)$$

The projection criterion is to minimize the KL divergence between $\hat{q}$ and q. To this end, we match the moments between $\hat{q}$ and q:

$$p_{t,i} = \frac{p_{t,i}^{\backslash o} \mathcal{N}(y_t | a_i h^H \mathbf{m}_t^{\backslash o}, V_{y_t})}{Z} \quad (16)$$
$$\mathbf{m}_t = \frac{\sum_{s_t \in \mathcal{A}} q^{\backslash o}(s_t) \mathcal{N}(y_t | \mathbf{m}_{y_t}, V_{y_t}) \mathbf{m}_{x_t|y_t}}{Z} \quad (17)$$
$$V_t = V_{x_t|y_t} - \mathbf{m}_t \mathbf{m}_t^H + + \sum_{s_t \in \mathcal{A}} q^{\backslash o}(s_t) \mathcal{N}(y_t | \mathbf{m}_{y_t}, V_{y_t}) \mathbf{m}_{x_t|y_t} \mathbf{m}_{x_t|y_t}^H / Z \quad (18)$$

where

$$\mathbf{m}_{x_t|y_t} = \mathbf{m}_t^{\backslash o} + K_{s_t}(y_t - s_t h^H \mathbf{m}_t^{\backslash o}) \quad (19)$$
$$V_{x_t|y_t} = V_t^{\backslash o} - K_{s_t} s_t h^H V_t^{\backslash o} \quad (20)$$
$$K_{s_t} = V_t^{\backslash o} h s_t^H V_{y_t}^{-1} \quad (21)$$

Then we compute the observation message as follows:

$$\tilde{o}(s_t, \mathbf{x}_t) = Z \frac{q(\mathbf{x}_t, s_t)}{q^{\backslash o}(\mathbf{x}_t, s_t)}. \quad (22)$$

It follows that

$$\tilde{o}(s_t, \mathbf{x}_t) = \tilde{o}(\mathbf{x}_t) \tilde{o}(s_t) \quad (23)$$
$$\tilde{o}(\mathbf{x}_t) \sim \mathcal{N}(\tilde{\boldsymbol{\mu}}_t, \tilde{\Lambda}_t) \quad (24)$$
$$\tilde{o}(s_t) \sim \text{Discrete}(r_{t,1}, r_{t,2}, \ldots, r_{t,M}) \quad (25)$$

where

$$\tilde{\boldsymbol{\mu}}_t = \tilde{\Lambda}_t V_t^{-1} \mathbf{m}_t - \tilde{\Lambda}_t (V_t^{\backslash o})^{-1} \mathbf{m}_t^{\backslash o} \quad (26)$$
$$\tilde{\Lambda}_t = (V_t^{-1} - (V_t^{\backslash o})^{-1})^{-1} \quad (27)$$

Note that the observation message is not necessarily a valid probability distribution. For example, $\tilde{\Lambda}_t$ may not be positive definite, or some of its elements may even go to infinite. To avoid numerical problems, we change $\tilde{\boldsymbol{\mu}}_t$ and $\tilde{\Lambda}_t$ to the natural parameterization of the exponential family:

$$\boldsymbol{\mu}_t = \tilde{\Lambda}_t^{-1} \tilde{\boldsymbol{\mu}}_{tt} = V_t^{-1} \mathbf{m}_t - (V_t^{\backslash o})^{-1} \mathbf{m}_t^{\backslash o} \quad (28)$$
$$\Lambda_t = \tilde{\Lambda}_t^{-1} = V_t^{-1} - (V_t^{\backslash o})^{-1} \quad (29)$$
$$r_{t,j} = \frac{p_{t,j}}{p_{t,j}^{\backslash o}} \qquad \text{for } j = 1, \ldots, M \quad (30)$$

3.2 Incorporating forward, observation, and backward messages

In this section, we describe how to incorporate the messages to update $q(s_t, \mathbf{x}_t)$. Because s_t's at different times are independent, we consider only the belief update for $\mathbf{x}_t$ when incorporating forward and backward messages. Since all the mariginal messages and belief which are related to $\mathbf{x}_t$ are Gaussians, we can efficiently update them.

1. Compute and incorporate the forward message $\tilde{g}_{t-1}(\mathbf{x}_t)$. Set $q^{\backslash o}(\mathbf{x}_t) = \tilde{g}_{t-1}(\mathbf{x}_t)$ such that

$$\mathbf{m}_t^{\backslash o} = \mathbf{F} \mathbf{m}_{t-1} \quad (31)$$
$$V_t^{\backslash o} = \mathbf{F} V_{t-1} \mathbf{F}^H + \mathbf{g}\mathbf{g}^H. \quad (32)$$

$\mathbf{m}_0$ and V_0 are chosen as the prior. Also, set $P_t = V_t^{\backslash o}$, which will be used later when incorporating the backward message.

2. Incorporate the observation message $\tilde{o}(s_t, \mathbf{x}_t)$. In the previous section, we compute $\tilde{o}(s_t, \mathbf{x}_t)$ based on the update of $q(s_t, \mathbf{x}_t)$. On the other hand, given $\tilde{o}(s_t, \mathbf{x}_t)$ and $q_t^{\setminus o}(\mathbf{x}_t)$, we can update $q(s_t, \mathbf{x}_t)$ by rewriting (28) to (30) as follows:

$$\mathbf{m}_t = V_t(\boldsymbol{\mu}_t + (V_t^{\setminus o})^{-1}\mathbf{m}_t^{\setminus o}) \quad (33)$$
$$V_t = (P_t^{-1} + \Lambda_t)^{-1} \quad (34)$$
$$p_{t,j} = r_{t,j} p_{t,j}^{\setminus o} \quad (35)$$

3. Incorporate the backward message $\tilde{g}_t(\mathbf{x}_t)$. Without explicitly computing the backward message $\tilde{g}_t(\mathbf{x}_t)$, we can directly incorporate $\tilde{g}_t(\mathbf{x}_t)$ into $q(\mathbf{x}_t)$ as Kalman smoothing:

$$\mathbf{J}_t = V_t^{\setminus \mathbf{b}} F^H P_t^{-1} \quad (36)$$
$$\mathbf{m}_t = \mathbf{m}_t^{\setminus \mathbf{b}} + \mathbf{J}_t(\mathbf{m}_{t+1} - \mathbf{F}\mathbf{m}_t^{\setminus o}) \quad (37)$$
$$V_t = V_t^{\setminus \mathbf{b}} + \mathbf{J}_t(V_{t+1}\mathbf{J}_t^H - \mathbf{F}(V_t^{\setminus o})^H) \quad (38)$$

   where $(\mathbf{m}_t^{\setminus \mathbf{b}},\ V_t^{\setminus \mathbf{b}})$ and $(\mathbf{m}_t,\ V_t)$ are the means and variances of the state belief before and after incorporating the backward message respectively.

3.3 Propagation iteration

For a linear Gaussian dynamic model or a discrete dynamic model, it will be sufficient to infer the exact posterior of any state given the whole observation sequence after propagating all the forward, observation, and backward messages once. However, the hybrid model of our interest is neither linear Gaussian nor completely discrete. As shown in [**?**], by iterating message propagation, expectation propagation will keep improving the approximation quality until it converges to a local minimum of its energy function; however, it is possible that expectation propagation does not converge as belief propagation, though it is rare in practice.

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. Initialize:

$$p_{t,j} = 1/M, \text{ for all } j,$$
$$\mathbf{m}_0 = [1, 1, \ldots, 1]^T,$$
$$V_0 = 10000\mathbf{I},$$
$$\boldsymbol{\mu}_t = [0, 0, \ldots, 0]^T,$$
$$\Lambda_t = \mathbf{0},$$

   where $\mathbf{I}$ and $\mathbf{0}$ are the identity and zero matrices.

2. Then loop $t = 1 : T$:
   (a) Set $q^{\setminus o}(\mathbf{x}_t)$ to the forward message from $\mathbf{x}_{t-1}$ via (31) and (32).
   (b) Update $q(s_t, \mathbf{x}_t)$ to match the moments of $o(s_t, \mathbf{x}_t)q^{\setminus o}(s_t, \mathbf{x}_t)$ via (16) to (18).
   (c) Compute $\tilde{o}(s_t, \mathbf{x}_t) \propto q(s_t, \mathbf{x}_t)/q^{\setminus o}(s_t, \mathbf{x}_t)$ via (28) to (30).

3. loop by increasing i until i equals n, or the convergence has been achieved:
   (a) loop $t = 1, \ldots, T$ (Skip on the first iteration)
      i. Set $q^{\setminus o}(\mathbf{x}_t)$ to the forward message from $\mathbf{x}_{t-1}$ via (31) and (32).
      ii. Set $q(s_t, \mathbf{x}_t)$ to the product $\tilde{o}(s_t, \mathbf{x}_t)q^{\setminus o}(s_t, \mathbf{x}_t)$ via (33) to (35).
   (b) loop $t = T, \ldots, 1$
      i. Set $\mathbf{m}_t^{\setminus \mathbf{b}} = \mathbf{m}_t$ and $V_t^{\setminus \mathbf{b}} = V_t$
      ii. Update $q(s_t, \mathbf{x}_t)$ by incorporating the backward message via (36) to (38) when $t < T$.
      iii. Update $q(s_t, \mathbf{x}_t)$ and $\tilde{o}(s_t, \mathbf{x}_t)$ as follows:
         A. Delete the current observation message from $q(s_t, \mathbf{x}_t)$. This is an important step, in order to avoid double-counting the observation message $\tilde{o}(s_t, \mathbf{x}_t)$:

$$q^{\setminus o}(s_t, \mathbf{x}_t) \propto q(s_t, \mathbf{x}_t)/\tilde{o}(s_t, \mathbf{x}_t).$$

         Then it follows that

$$\mathbf{m}_t^{\setminus o} = V_t^{\setminus o}(V_t^{-1}\mathbf{m}_t - \boldsymbol{\mu}_t), \quad (39)$$
$$V_t^{\setminus o} = (V_t^{-1} - \Lambda_t)^{-1}, \quad (40)$$
$$p_{t,j}^{\setminus o} = p_{t,j}/r_{t,j} \quad (41)$$

         B. Update $q(s_t, \mathbf{x}_t)$ to match the moments of $o(s_t, \mathbf{x}_t)q^{\setminus o}(s_t, \mathbf{x}_t)$ via (16) to (18).
         C. Compute via (28) to (30) $\tilde{o}(s_t, \mathbf{x}_t) \propto q(s_t, \mathbf{x}_t)/q^{\setminus o}(s_t, \mathbf{x}_t)$.

Instead of smoothing over the whole sequence as described above, we can use a sliding window, with or without overlapping, to reduce time delay for online estimation. It only requires some minor modifications of the time indexes in the above algorithm.

3.4 Computational complexity

The total computation time of incorporating the forward and observation messages via (31) to (35) is $O(d^3)$ (d is the dimension of $\mathbf{x}_t$), same as one-step Kalman filtering; incorporating the backward message via (36) to (38) takes $O(d^3)$ as Kalman smoothing; and finally updating $q(s_t, \mathbf{x}_t)$ and $\tilde{o}(s_t, \mathbf{x}_t)$ in step 3.(b).iii costs $O(d^3)$ too. Furthermore, since the estimation accuracy is not increasing after a few propagation iterations, the required number of iterations n is small. In practice, we set $n = 5$. In sum, if the length of the sliding window for smoothing is L, the computation takes $O(nLd^3)$.

In contrast, if we use m samples in stochastic mixture of Kalman filters, it takes $O(mMd^3)$ for one step update, and $O(mM^Ld^3)$ for L step smoothing [**?**].

Comparing the computation time, we can see the expectation propagation algorithm takes much less computation time than the sequential Monte Carlo filters and smoothers, especially when the window length L is large or a large number of samples need to be drawn.

4 Adaptive EP Receiver for Flat-fading Channels

For the signal detection problem in flat-fading channels, we apply the expectation algorithm algorithm developed in section 3 to the wireless communication system defined in (2) and (3), and decode the symbols s_t as

$$\hat{s}_t = \{a_i | \arg\max_i \{\hat{p}_{t,i}\}\}. \qquad (42)$$

where $\hat{p}_{t,i} areobtained after the convergence of the expectation$

5 Simulation

In this section, we demonstrate the high performance of the proposed 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 [**?**]:

$$\Phi = [-2.37409 \quad 1.92936 \quad -0.53208]$$
$$\Theta = 0.01 \times [0.89409 \quad 2.68227 \quad 2.68227 \quad 0.89409]$$
$$v_t \sim \mathcal{N}_c(0, 1).$$

With these parameters, we have $\mathrm{Var}\{\alpha_t\} = 1$. BPSK modulation is employed, that is, $s_t \in \{1, -1\}$. In addition, differential encoding and decoding are employed to resolve the phase ambiguity.

We test the new EP receiver with different smoothing window lengths $L = 1, 2, 4$, with $0, 1, 3$ overlap points correspondingly. In other words, the estimation time delay δ equals 0,1,and 3 repectively. Moreover, we run the EP receiver with smoothing over the whole 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 transimitted 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 the new observation sequence $\{\tilde{y}_t\}$. The symbols are then demodulated according to

$$\hat{s}_t = \mathrm{sign}(\mathcal{R}\{\hat{\alpha}_t^\star y_t\}) \qquad (43)$$

where $\star$ means conjugate. By obtaining the extra information from the genie, this detector supposes to achieve accurate detection results. For the differential detection, no attempt is made for channel estimation. It simply detects the phase difference between two consecutively observations y_{t-1} and y_t:

$$\hat{s_t} = \mathrm{sign}(\mathcal{R}\{\hat{y_t}^\star y_{t-1}\}). \qquad (44)$$

We run these detectors on 50,000 received signals multiple times. Each time, we randomly synthesize a new symbol sequence and a new observation sequence according to (2) and (3). The signal-noise ratio (SNR), defined as $10\log_{10}(\mathrm{Var}\{\alpha_t\}/\mathrm{Var}\{w_t\})$, increases each time. The bit-error rate (BER) performance of different detectors versus SNR is plotted in figure 1.

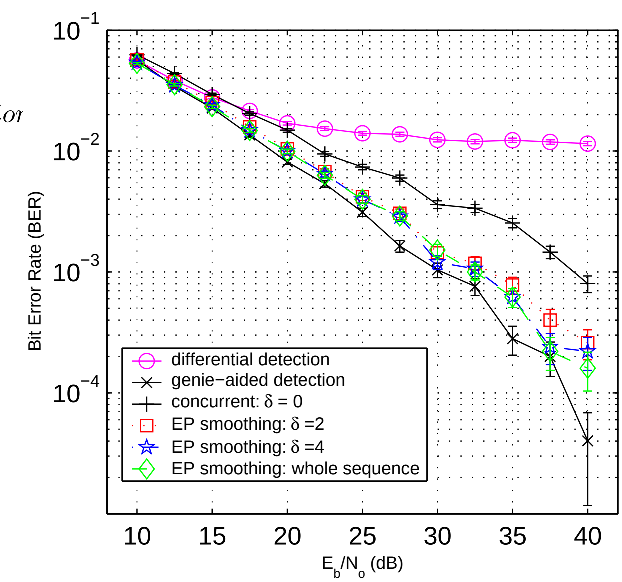

Figure 1: BER demodulation performance of the EP receiver with different smoothing parameters, the genie-aided detector, and the differential detector in a fading channel with complex Gaussian noises. The unit error bars, defined as $\sqrt{\frac{\mathrm{BER}(1-\mathrm{BER})}{T}}$, are also shown in the figure.

As shown in the figure, the proposed EP receiver with smoothing clearly outperforms the concurrent detector and the differential detector. The EP receiver does not have the error floor, as does the differential detector.

With a delay $\delta = 3$, the performance of the EP receiver is almost as good as that of the EP receiver with smoothing over the whole data sequences, and it is close to the performance of the genie-aided detector. Considering the range indicated by the error bars, the performance of the EP receiver is comparable to that of sequential Monte Carlo receivers as reported in [**?**]. In contrast, the EP receiver has much less computational complexity as discussed in section 3.4.

The EP receiver works in a robust way in the practice; it always converges to an accurate Bayesian estimation in all of our simulations.

6 Conclusion and future work

In this paper, we have developed the expectation propagation receiver for signal detection in fading channels. As shown by the simulations, the EP receiver with short-time smoothing clearly outperforms the differential detector and the concurrent adaptive Bayesian receiver under different signal-noise ratios. Moreover, its performance is close to the so-called genie-aided detection. Compared to sequential Monte Carlo filtering methods, the EP receiver has much lower computational complexity.

As to the future work, we plan to extend this work to the case that transmitted symbols are coded by convolutional codes. In such case, we will exploit the correlation between coded symbols at different times.

Expectation propagation, as an extension of belief propagation, allows the use of more complicated models in digital communications. It could be applied to many areas of digital communications such as iterative decoding, iterative equalization, and iterative demodulation. The EP receiver for wireless signal detection is only one example.

References

[Chen et al., 2000] Chen, R., Wang, X., and Liu, J. S. (2000). Adaptive joint detection and decoding in flat-fading channels via mixture Kalman filtering. *IEEE Transactions on Information Theory*, 46(6):2079–2094.

[D'Andrea et al., 1995] D'Andrea, A. N., Diglio, A., and Mengali, U. (1995). Symbol-aided channel estimation with nonselective Rayleigh fading channels. *IEEE Transactions on Communications*, 44:41–48.

[Hoeher and Lodge, 1999] Hoeher, P. and Lodge, J. H. (1999). Turbo DPSK:Iterative differential PSK domodulation and channel decodings. *IEEE Transactions on Communications*, 47:837–843.

[McEliece et al., 1998] McEliece, R. J., MacKay, D. J. C., and Cheng, J.-F. (1998). Turbo decoding as an instance of Pearl's 'Belief Propagation' algorithm. *IEEE Journal on Selected Areas in Communications*, 16(2):140–152.

[Minka, 2001a] Minka, T. (2001a). *A family of algorithms for approximate Bayesian inference*. PhD thesis, MIT. www.media.mit.edu/~tpminka/papers/learning.html.

[Minka, 2001b] Minka, T. P. (2001b). Expectation propagation for approximate Bayesian inference. In *Uncertainty in AI'01*. `http://www.stat.cmu.edu/~minka/papers/ep/`.

[Vitetta and Taylor, 1995] Vitetta, G. M. and Taylor, D. P. (1995). Maximum likelihood decoding of uncoded and coded PSK signal sequences transmitted over Rayleigh flat-fading channels. *IEEE Transactions on Communications*, 43:2750–2758.

[Wang et al., 2002] Wang, X., Chen, R., and Guo, D. (2002). Delayed-pilot sampling for mixture Kalman filter with application in fading channels. *IEEE Transactions on Signal Processing, Special Issue on Monte Carlo Methods for Statistical Signal Processing,*, 50(2):241–254.