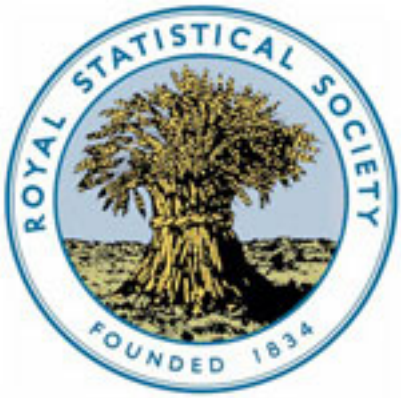

J. R. Statist. Soc. B (2000)
62, *Part* 3, *pp.* 493–508

Mixture Kalman filters

Rong Chen

Texas A&M University, College Station, USA

and Jun S. Liu

Stanford University, USA

[Received February 1999. Final revision November 1999]

Summary. In treating dynamic systems, sequential Monte Carlo methods use discrete samples to represent a complicated probability distribution and use rejection sampling, importance sampling and weighted resampling to complete the on-line 'filtering' task. We propose a special sequential Monte Carlo method, the mixture Kalman filter, which uses a random mixture of the Gaussian distributions to approximate a target distribution. It is designed for on-line estimation and prediction of conditional and partial conditional dynamic linear models, which are themselves a class of widely used non-linear systems and also serve to approximate many others. Compared with a few available filtering methods including Monte Carlo methods, the gain in efficiency that is provided by the mixture Kalman filter can be very substantial. Another contribution of the paper is the formulation of many non-linear systems into conditional or partial conditional linear form, to which the mixture Kalman filter can be applied. Examples in target tracking and digital communications are given to demonstrate the procedures proposed.

Keywords: Conditional dynamic linear models; Dynamic systems; Fading channels; Sequential Monte Carlo methods; State space models; Target tracking

1. Introduction

Dynamic systems are widely used in applied fields such as computer vision, economics and financial data analysis, feed-back control systems, mobile communication, radar or sonar surveillance systems, just to name a few. A main challenge to researchers in these fields is to find efficient methods for on-line (in realtime) estimation and prediction (filtering) of the ever-changing system characteristics, along with the continuous flow of the information (observations) from the system.

For a Gaussian linear system, Kalman (1960) provided an ingenious algorithm (the Kalman filter) for on-line filtering. To date, however, there has not been a universally effective algorithm for dealing with non-linear and non-Gaussian systems. Depending on the features of individual problems, some generalizations of the Kalman filter can be effective. A few well-known generalizations are the extended Kalman filters (Gelb, 1974), Gaussian sum filters (Anderson and Moore, 1979) and iterated extended Kalman filters (Jazwinski, 1970). Most of these methods are based on local linear approximations of the non-linear system. More recently, researchers have been attracted to a new class of filtering methods based on the sequential Monte Carlo approach.

Sequential Monte Carlo techniques achieve the filtering task by recursively generating

Address for correspondence: Rong Chen, Department of Statistics, 447 Blocker Building, Texas A&M University, College Station, TX 77843-3143, USA.
E-mail: chen@stat.tamu.edu

 1369–7412/00/62493

properly weighted Monte Carlo samples of the state variables or some other latent variables. The samples and their weights are then used to estimate various system characteristics. These Monte-Carlo-based methods are often more flexible in dealing with non-Gaussian models and more adaptive to features of the target system. Since the appearance of two such methods, the *bootstrap filter* (also called the particle filter) for non-linear state space models (Gordon *et al.*, 1993) and *sequential imputation* for general missing data problems (Kong *et al.*, 1994), Monte Carlo filtering techniques have caught the attention of researchers in many different areas that require dynamic modelling. Several modifications and improvements on the method have been suggested (Berzuini *et al.*, 1997; Carpenter *et al.*, 1997; Doucet *et al.*, 1999; Hürzeler and Künsch, 1998; Liu and Chen, 1995; Pitt and Shephard, 1999; Tanizaki, 1996). A *sequential importance sampling* framework has been proposed (Liu and Chen, 1998) to unify and generalize these techniques. In the following context, we refer to this class of methods applied to state space models as *Monte Carlo filters*.

Consider the general state space model of the following form: state equation—

$$x_{t+1} \sim f_t(\cdot|x_t); \tag{1a}$$

observation equation—

$$y_{t+1} \sim g_t(\cdot|x_{t+1}). \tag{1b}$$

Here the x_t are unobserved state variables and the y_t are observed signals. Let $\mathbf{y}_t = (y_1, \ldots, y_t)$ be the information that is available up to time t. Of interest in these systems are

(a) *estimation* of the state variable, say $E(x_t|\mathbf{y}_t)$, by using all available information;
(b) *prediction* of a future state, say $E(x_{t+1}|\mathbf{y}_t)$, and
(c) *revision* or *smoothing* of the previous state estimations given new information, e.g. $E(x_{t-s}|\mathbf{y}_t)$.

The main challenge is that these tasks need to be done on line, which makes it critical for a filtering method to be able to modify the estimations or predictions quickly as new observations come in.

In this paper, we focus on a special case of the state space model, the *conditional dynamic linear model* (CDLM), which is a direct generalization of the DLM (West and Harrison, 1989) and has been widely used in practice (see Shephard (1994) for more examples). An important feature of the CDLM, whose precise definition can be found in Section 2, is that, given the trajectory of an indicator variable (vector), the system is Gaussian and linear, for which the Kalman filter can be used. Thus, by using the *marginalization* technique for Monte Carlo computation (Rubinstein, 1981), we derive a Monte Carlo filter that focuses its full attention on the space of the indicator variable. We call this filter a *mixture Kalman filter* (MKF). By doing so we can achieve a much smaller Monte Carlo variation than that of a standard Monte Carlo filter applied directly to the state variables.

The MKF idea can also be applied to those systems that are only partially linear and conditional Gaussian, i.e. the systems whose state variable consists of a component that is conditionally linear and a component that is completely non-linear. By conditioning on an indicator and the value of the non-linear component of the state variable, the system (both the state equation and the observation equation) becomes linear and Gaussian. We call such a system a *partial CDLM* (PCDLM). In this case, the linear component of the state variable can be 'marginalized' out before running a Monte Carlo filter. The marginalization operation is again achieved by the Kalman filter operation. We call this method an extended MKF (EMKF).

Given the importance of the CDLM in system modelling, it is perhaps not surprising that approaches similar to the MKF described in this paper have been proposed earlier. Indeed, the earlier work of Ackerson and Fu (1970), Akashi and Kumamoto (1977) and Tugnait (1982) and recent work of Liu and Chen (1995) and Doucet *et al.* (1999) are all closely related. We shall provide a more detailed account on each of these approaches in Section 3.

The rest of the paper is organized as follows. In Section 2, we give a detailed description of CDLMs and the proposed MKF. Section 3 is devoted to the PCDLMs and the EMKFs. In Section 4, we give several applications of the proposed MKF and EMKF, including three examples in target tracking and one example in telecommunications. A brief summary is given in Section 5.

2. Model and method

2.1. Conditional dynamic linear models

A CDLM can be generally defined as follows:

$$\begin{aligned} x_t &= H_\lambda x_{t-1} + W_\lambda w_t, \\ y_t &= G_\lambda x_t + V_\lambda v_t \end{aligned} \qquad \text{if } \Lambda_t = \lambda \tag{2}$$

where $w_t \sim N(0, I)$, $v_t \sim N(0, I)$ and all coefficient matrices are known given λ. The Λ_t, which can be either continuous or discrete, is a latent indicator process with certain probabilistic structure. With discrete indicator variables, the model can be used to deal with outliers, sudden jumps, system failures, environmental changes and clutters. With carefully chosen continuous indicator variables, CDLMs can also accommodate DLMs with non-Gaussian innovations.

2.1.1. Example 1

A special CDLM is the linear system with non-Gaussian errors. Suppose that

$$\begin{aligned} x_t &= Hx_{t-1} + Ww_t, \\ y_t &= Gx_t + Vv_t, \end{aligned}$$

where w_t and v_t are mixed Gaussian, i.e., conditional on the unobserved variable $\Lambda_t = (\eta_{1t}, \eta_{2t})$, the errors' distributions are $w_t|\eta_{1t} \sim N\{\boldsymbol{\mu}_1(\eta_{1t}), \Sigma_1(\eta_{1t})\}$ and $v_t|\eta_{2t} \sim N\{\boldsymbol{\mu}_2(\eta_{2t}), \Sigma_2(\eta_{2t})\}$, where $\boldsymbol{\mu}_1$, $\boldsymbol{\mu}_2$, Σ_1 and Σ_2 are functions of (η_{1t}, η_{2t}). This model is clearly a CDLM with Λ_t being its latent indicator processes. This class of error models includes, in addition to the discrete mixture of Gaussian distributions, the t-distributions, double-exponential distributions, the exponential power family and logistic distributions. Even if w_t and v_t are not mixed Gaussian, most of the time they can be satisfactorily approximated by a mixture of Gaussian distributions. In Section 5 we analyse several CDLMs in practice.

Engineers have dealt with special forms of the CDLM since the 1970s. In pioneering work, Ackerson and Fu (1970) considered a linear system operating in switching environments, which they formulate as the model in example 1 with the Λ_t being a finite discrete Markovian indicator process. To deal with the computational difficulty, they proposed an approximate filtering procedure in which the posterior probability of the indicator variable Λ_t (given y_t) is recursively updated under a conditional independence assumption, and then used in a Gaussian approximation of the posterior of x_t. Their approach can be easily generalized to update a segment $(\Lambda_{t-k}, \ldots, \Lambda_t)$ of the indicator process recursively (Tugnait, 1982). In

dealing with the same CDLM, Akashi and Kumamoto (1977) introduced essentially a Monte Carlo filter (i.e. a sequential importance sampler) for the indicator process in which an 'optimal' sampling distribution is used (Liu and Chen, 1998). Thus, Akashi and Kumamoto's algorithm is closest to the MKF proposed in this paper. However, the key resampling and rejection steps are missing in their method, which makes it perform much less satisfactorily. By formulating the MKF in a general sequential Monte Carlo framework, we can incorporate various Monte Carlo techniques, such as resampling, rejection control and the auxiliary variable approach, into the scheme and greatly extend the applicability of the method. More recently, the methods used in Svetnik (1986), Liu and Chen (1995) and Doucet *et al.* (1999) have all captured some attractive aspects of the CDLM and the MKF, but they are limited in scope.

Several Markov chain Monte Carlo algorithms for CDLMs and other state space models have been proposed (Carlin *et al.*, 1992; Carter and Kohn, 1994; Shephard, 1994). In particular, Shephard (1994) studied a class of 'partial non-Gaussian' models, a forerunner of CDLMs. He also suggested an efficient Monte Carlo algorithm for the posterior computation, in which the Gibbs sampler iterates between two big blocks, $\mathbf{\Lambda}_t$ and $\mathbf{x}_t$, to improve efficiency. Carter and Kohn (1994) presented another efficient Gibbs sampler for CDLMs in which the discrete indicator $\mathbf{\Lambda}_t$ is the only latent variable to be imputed and the state variable $\mathbf{x}_t$ is explicitly integrated out via a clever use of forward and backward Kalman filtering. A major problem with all the Markov chain Monte Carlo algorithms for dynamic systems, however, is that they cannot be effectively used for on-line estimation and prediction, whereas a fast and efficient on-line algorithm in problems such as target tracking and digital signal processing is essential.

A bootstrap filter can be directly applied to the CDLM for on-line estimation and prediction (see, for example, Avitzour (1995), Gordon *et al.* (1993) and Kitagawa (1996)). In such a procedure, Monte Carlo samples of the state variable x_t are recursively generated by the sampling–importance resampling technique (Rubin, 1987). In this paper we propose a more sophisticated algorithm by making further use of the conditional Gaussian structure, and, similar in spirit to West (1992), using a mixture of Gaussian distributions to approximate the target posterior distribution. In a CDLM, the mixture Gaussian distribution becomes an obvious choice because of the efficient Kalman filter.

2.2. The method of mixture Kalman filtering

Let $\mathbf{y}_t = (y_1, \ldots, y_t)$ and $\mathbf{\Lambda}_t = (\Lambda_1, \ldots, \Lambda_t)$. Let $\boldsymbol{\lambda}_t$ and λ_s be realizations of $\mathbf{\Lambda}_t$ and Λ_s respectively. We observe that

$$p(x_t|\mathbf{y}_t) = \int p(x_t|\boldsymbol{\lambda}_t, \mathbf{y}_t)\, p(\boldsymbol{\lambda}_t|\mathbf{y}_t)\, \mathrm{d}\boldsymbol{\lambda}_t,$$

where $p(x_t|\boldsymbol{\lambda}_t, \mathbf{y}_t) \sim N\{\boldsymbol{\mu}_t(\boldsymbol{\lambda}_t), \Sigma_t(\boldsymbol{\lambda}_t)\}$, in which $(\mu_t(\boldsymbol{\lambda}_t), \Sigma_t(\boldsymbol{\lambda}_t))$ can be obtained by running the Kalman filter with given trajectory $\boldsymbol{\lambda}_t$. The main idea of the MKF is to use a *weighted sample* of the indicators,

$$\mathcal{S}_t = \{(\boldsymbol{\lambda}_t^{(1)}, w_t^{(1)}), \ldots, (\boldsymbol{\lambda}_t^{(m)}, w_t^{(m)})\},$$

to represent the distribution $p(\mathbf{\Lambda}_t|\mathbf{y}_t)$, and then to use a random mixture of Gaussian distributions,

$$\frac{1}{W_t} \sum_{j=1}^{m} w_t^{(j)}\, N\{\boldsymbol{\mu}_t(\boldsymbol{\lambda}_t^{(j)}), \Sigma_t(\boldsymbol{\lambda}_t^{(j)})\},$$

where $W_t = \Sigma_{j=1}^m w_t^{(j)}$, to approximate the target distribution $p(x_t|\mathbf{y}_t)$. For any integrable function $h(\cdot)$, we approximate the quantity of interest $E\{h(x_t)|\mathbf{y}_t\}$ as

$$\hat{E}\{h(x_t)|\mathbf{y}_t\} = \frac{1}{W_t}\sum_{j=1}^{m} w_t^{(j)} \int h(x)\,\varphi\{x;\ \mu_t(\boldsymbol{\lambda}_t^{(j)})\ \ \Sigma_t(\boldsymbol{\lambda}_t^{(j)})\}\,\mathrm{d}x,$$

where φ is the Gaussian density function.

Whereas a straightforward Monte Carlo filter uses a weighted sample of the state variable, $\{(x_t^{(j)}, w_t^{(j)})\}_{j=1}^m$, to approximate $p(x_t|\mathbf{y}_t)$, the MKF operates in the indicator space, which is equivalent to marginalizing out the x_t. This approach has been shown to improve a Gibbs sampling algorithm (Liu *et al.*, 1994) and a standard importance sampling scheme (MacEachern *et al.*, 1999). No clear theory is available so far in the sequential Monte Carlo setting. Our limited experience shows that the efficiency gain of the MKF can be very significant. Intuitively, the usual Monte Carlo filter recursively approximates the posterior of x_t by a discrete sample, whereas the MKF approximates the posterior of x_t by a mixture of Gaussian distributions. Note that the true posterior of x_t in a CDLM is indeed a mixed Gaussian posterior, although the number of its components increases exponentially with t.

Let $\mathrm{KF}_t^{(j)} = (\boldsymbol{\mu}_t(\boldsymbol{\lambda}_t^{(j)}),\ \Sigma_t(\boldsymbol{\lambda}_t^{(j)}))$ which records the posterior mean and covariance matrix of x_t, conditional on $\mathbf{y}_t$ and a given trajectory $\boldsymbol{\lambda}_t^{(j)}$. This can be obtained by the Kalman filter. Then the MKF updating scheme consists of recursive applications of the following steps.

For $j = 1, \ldots, m$:

(a) generate $\lambda_{t+1}^{(j)}$ from a trial distribution $q_{t+1}(\lambda_{t+1}|\boldsymbol{\lambda}_t^{(j)}, \mathrm{KF}_t^{(j)})$;

(b) obtain $\mathrm{KF}_{t+1}^{(j)}$ by a one-step Kalman filter, conditional on $\{\mathrm{KF}_t^{(j)}, y_{t+1}, \lambda_{t+1}^{(j)}\}$,

$$\begin{aligned} P_{t+1} &= H_{\lambda_{t+1}}\Sigma_t H'_{\lambda_{t+1}} + W_{\lambda_{t+1}}W'_{\lambda_{t+1}},\\ S_{t+1} &= G_{\lambda_{t+1}}P_{t+1}G'_{\lambda_{t+1}} + V_{\lambda_{t+1}}V'_{\lambda_{t+1}},\\ \boldsymbol{\mu}_{t+1} &= H_{\lambda_{t+1}}\boldsymbol{\mu}_t + P_{t+1}G'_{\lambda_{t+1}}S_{t+1}^{-1}(y_{t+1} - G_{\lambda_{t+1}}H_{\lambda_{t+1}}\boldsymbol{\mu}_t),\\ \Sigma_{t+1} &= P_{t+1} - P_{t+1}G'_{t+1}S_{t+1}^{-1}G_{\lambda_{t+1}}P_{t+1}; \end{aligned} \tag{3}$$

(c) update the new weight as $w_{t+1}^{(j)} = w_t^{(j)} \times u_{t+1}^{(j)}$, where

$$u_{t+1}^{(j)} = \frac{p(\boldsymbol{\lambda}_t^{(j)}, \lambda_{t+1}^{(j)}|\mathbf{y}_{t+1})}{p(\boldsymbol{\lambda}_t^{(j)}|\mathbf{y}_t)\,q_{t+1}(\lambda_{t+1}^{(j)}|\boldsymbol{\lambda}_t^{(j)}, \mathrm{KF}_t^{(j)})};$$

(d) (resampling–rejuvenation) if the coefficient of variation of the w_{t+1} exceeds a threshold value, resample a new set of KF_{t+1} from $\{\mathrm{KF}_{t+1}^{(1)}, \ldots, \mathrm{KF}_{t+1}^{(m)}\}$ with probability proportional to the weights $w_{t+1}^{(j)}$.

When Λ_t takes values in a finite discrete set $\mathcal{I}$, then a reasonable trial distribution for Λ_{t+1} is its predictive distribution $q_{t+1}(\lambda_{t+1}|\boldsymbol{\lambda}_t, \mathrm{KF}_t) = p(\lambda_{t+1}|\boldsymbol{\lambda}_t, \mathrm{KF}_t, y_{t+1})$, which can be obtained by inspecting all the possible values of Λ_t. The incremental weight $u_{t+1}^{(j)}$ is then simplified as

$$u_{t+1}^{(j)} \propto p(y_{t+1}|\mathrm{KF}_t^{(j)}) = \sum_{i\in\mathcal{I}} p(y_{t+1}|\Lambda_{t+1} = i, \mathrm{KF}_t^{(j)})\,p(\Lambda_{t+1} = i|\boldsymbol{\lambda}_t^{(j)}).$$

Specifically, an MKF updating step in this case becomes, for $j = 1, \ldots, m$,

(i) for each $\Lambda_{t+1} = i$, $i \in \mathcal{I}$, run the Kalman filter to obtain

$$v_i^{(j)} \propto p(y_{t+1}|\Lambda_{t+1} = i, \mathrm{KF}_t^{(j)})\,p(\Lambda_{t+1} = i|\boldsymbol{\lambda}_t^{(j)}),$$

where $p(\Lambda_{t+1} = i|\boldsymbol{\lambda}_t^{(j)})$ is the prior transition probability for the indicator and $p(y_{t+1}|\Lambda_{t+1} = i, \mathrm{KF}_t^{(j)})$ is a by-product of the Kalman filter

$$p(y_{t+1}|\mathbf{y}_t, \boldsymbol{\lambda}_t, \lambda_{t+1}) \sim N(G_{\lambda_{t+1}} H_{\lambda_{t+1}} \boldsymbol{\mu}_t, S_{t+1}), \tag{4}$$

(ii) sample a $\lambda_{t+1}^{(j)}$ from the set $\mathcal{I}$, with probability proportional to $v_i^{(j)}$,
(iii) let $\mathrm{KF}_{t+1}^{(j)}$ be the one with $\Lambda_{t+1} = \lambda_{t+1}^{(j)}$,
(iv) the new weight is $w_{t+1}^{(j)} = w_t^{(j)} \Sigma_{i \in \mathcal{I}} v_i^{(j)}$.

Other choices of q_{t+1}, such as a 'delayed sampling' version

$$q_{t+1}(\lambda_{t+1}|\boldsymbol{\lambda}_t, \mathrm{KF}_t) = p(\lambda_{t+1}|\boldsymbol{\lambda}_t, \mathrm{KF}_t, y_{t+1}, \ldots, y_{t+d})$$

or its approximations, are also possible and sometimes more desirable.

Smith and Winter (1978) proposed a deterministic method called the *split track filter* for CDLMs with a finite discrete indicator variable. Their method has a similar flavour to the MKF. In a split track filter, we always keep m trajectories of the latent indicators. At a future time step, we evaluate the likelihoods of all possible one-step propagations from the m trajectories held previously and retain the m updated trajectories with the highest likelihood values. In contrast, our MKF selects the updated trajectories randomly, according to the weights (which is the predictive likelihood value), and uses the associated weights to measure how good each trajectory is. The important step of resampling is naturally built into the MKF which can overcome some weaknesses of the split track filter. More sophisticated sampling and estimation methods can also be incorporated. A comparison of the MKF and the split track filter in target tracking is presented in Section 5.

When Λ_t is a continuous random variable a simpler but less efficient algorithm is

(ii) sample a $\lambda_{t+1}^{(j)}$ from $p(\Lambda_{t+1}|\boldsymbol{\lambda}_t^{(j)})$, the prior structure of the indicator variable,
(iii) run one step of the Kalman filter on $\{\lambda_{t+1}^{(j)}, \mathrm{KF}_t^{(j)}, y_{t+1}\}$ to obtain $\mathrm{KF}_{t+1}^{(j)}$, using equation (3),
(iv) the new weight is $w_{t+1}^{(j)} = w_t^{(j)} p(y_{t+1}|\lambda_{t+1}^{(j)}, \mathrm{KF}_t^{(j)})$ using expression (4).

The methods of Berzuini *et al.* (1997) and Pitt and Shephard (1999) can be applied to improve the efficiency of this algorithm.

3. The extended mixture Kalman filters

3.1. Partial conditional dynamic linear models

Suppose that the state variable has two components: $x_t = (x_{t1}, x_{t2})$. The following system is called a PCDLM: state equations—

$$x_{t,1} = H_t(x_{t-1,2}, \Lambda_t)x_{t-1,1} + W_t(x_{t-1,2}, \Lambda_t)w_t,$$
$$x_{t,2} = g_t(x_{t-1,2}, \Lambda_t, \epsilon_t);$$

observation equation—

$$y_t = G_t(x_{t,2}, \Lambda_t)x_{t,1} + h_t(x_{t,2}, \Lambda_t) + V_t(x_{t,2}, \Lambda_t)v_t,$$

with $w_t \sim N(0, I)$ and $v_t \sim N(0, I)$. The matrices H_t, G_t, W_t and V_t are known given the values of $\{x_{t-1,2}, x_{t,2}, \Lambda_t\}$. The functions g_t and h_t are known and ϵ_t has a known distribution.

There is in fact no absolute distinction between a PCDLM and a CDLM because, if we regard the 'non-linear' component $x_{t,2}$ of the state variable in a PCDLM as part of the

indicator variable, the system becomes a CDLM. However, unlike in CDLMs where we have no interest in the latent indicator, inference about the non-linear component of the state variable is often of great interest in a PCDLM. Note that in our model formulation the state propagation of the non-linear component does not depend on the linear component.

3.1.1. Example 2: fading channel

Many mobile communication channels can be modelled as Rayleigh flat fading channels, which have the following form: state equations —

$$\mathbf{x}_t = F\mathbf{x}_{t-1} + Ww_t,$$
$$\alpha_t = G\mathbf{x}_t,$$
$$s_t \sim p(\cdot|s_{t-1});$$

observation equation —

$$y_t = \alpha_t s_t + Vv_t,$$

where s_t are the input digital signals (symbols), y_t are the received complex signals and α_t are the unobserved (changing) fading coefficients. Both w_t and v_t are complex Gaussian with identity covariance matrices. This system is clearly a PCDLM. Given the input signals s_t, the system is linear in $\mathbf{x}_t$ and y_t. In Section 4 we show how to use the EMKF for extracting digital signals transmitted over such channels.

3.1.2. Example 3: blind deconvolution

Consider the following system in digital communication:

$$y_t = \sum_{i=1}^{q} \theta_i s_{t-i} + \epsilon_t,$$

where s_t is a discrete process taking values on a known set S. In a blind deconvolution problem, s_t is to be estimated from the observed signals $\{y_1, \ldots, y_t\}$, without knowing the channel coefficients θ_i. This system can be formulated as a PCDLM. Let $\boldsymbol{\theta}_t = (\theta_{t1}, \ldots, \theta_{tq})$ and $x_t = (s_t, \ldots, s_{t-q})'$. We can define state equations

$$\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1},$$
$$x_t = Hx_{t-1} + Ws_t$$

and observation equation

$$y_t = \boldsymbol{\theta}_t x_t + \epsilon_t,$$

where H is a $q \times q$ matrix with lower off-diagonal element 1 and all other elements 0 and $W = (1, 0, \ldots, 0)'$. In this case, the unknown system coefficients are part of the state variable and are linear conditional on the digital signal x_t. Liu and Chen (1995) studied this problem with a procedure which is essentially an EMKF as described in the next subsection. This PCDLM formulation can be easily extended to deal with a blind deconvolution problem with time-varying system coefficients.

3.2. The extended mixture Kalman filter

The main idea of the EMKF is to extract as many linear and Gaussian components from the

system as possible, and then to integrate these components out (marginalize) using the Kalman filter before running a Monte Carlo filter on the remaining components. Thus, in the EMKF we generate discrete samples in the joint space of the latent indicator and the non-linear state component. More intuitively, because of the fact that

$$p(x_{t1}, x_{t2}|\mathbf{y}_t) = p(x_{t1}|x_{t2}, \mathbf{y}_t)\, p(x_{t2}|\mathbf{y}_t),$$

the approximation of $p(x_{t1}, x_{t2}|\mathbf{y}_t)$ in the EMKF is decomposed as a Monte Carlo approximation of the marginal distribution $p(x_{t2}|\mathbf{y}_t)$ and an exact Gaussian conditional distribution $p(x_{t1}|x_{t2}, \mathbf{y}_t)$. Let $\mathbf{x}_{t,2} = (x_{1,2}, \ldots, x_{t,2})$. The EMKF algorithm is as follows: suppose that at time t we have a sample $(\boldsymbol{\lambda}_t^{(j)}, \mathbf{x}_{t,2}^{(j)}, \mathrm{KF}_t^{(j)}, w_t^{(j)})$, $j = 1, \ldots, m$, where $\mathrm{KF}_t^{(j)} = (\boldsymbol{\mu}_t(\boldsymbol{\lambda}_t^{(j)}, \mathbf{x}_{t,2}^{(j)}), \Sigma_t(\boldsymbol{\lambda}_t^{(j)}, \mathbf{x}_{t,2}^{(j)}))$ represents the mean and the covariance matrix of $p(x_{t1}|\boldsymbol{\lambda}_t^{(j)}, \mathbf{x}_{t2}^{(j)}, \mathbf{y}_t)$ obtained by the Kalman filter. The EMKF updating scheme consists of recursive application of the following steps.

For $j = 1, \ldots, m$,

(a) generate $(\lambda_{t+1}^{(j)}, x_{t+1,2}^{(j)})$ from a trial distribution $q_{t+1}(\lambda_{t+1}, x_{t+1,2}|\boldsymbol{\lambda}_t^{(j)}, x_{t,2}^{(j)}, \mathrm{KF}_t^{(j)})$,
(b) run one step of the Kalman filter conditioning on $(\lambda_{t+1}^{(j)}, x_{t+1,2}^{(j)}\, \mathrm{KF}_t^{(j)}, y_{t+1})$ and obtain $\mathrm{KF}_{t+1}^{(j)}$,
(c) calculate the incremental weight

$$u_{t+1}^{(j)} = \frac{p(\boldsymbol{\lambda}_t^{(j)}, x_{t+1,2}^{(j)}, \lambda_{t+1}^{(j)}|\mathbf{y}_{t+1})}{p(\boldsymbol{\lambda}_t^{(j)}, x_{t,2}^{(j)}|\mathbf{y}_t)\, q_{t+1}(\lambda_{t+1}^{(j)}, x_{t+1,2}^{(j)}|\lambda_t^{(j)}, x_{t,2}^{(j)}, \mathrm{KF}_t^{(j)})}$$

and update the new weight as $w_{t+1}^{(j)} = w_t^{(j)} u_{t+1}^{(j)}$,
(d) resample as in step (d) of the MKF updating scheme if necessary.

From the weighted sample obtained at each time t, we can estimate quantities of interest, e.g.

$$E\{h(x_t)|y_1, \ldots, y_t\} \approx W_t^{-1} \sum_{j=1}^{m} w_t^{(j)} \int h(x_1, x_{t,2}^{(j)})\, \varphi(x_1; \boldsymbol{\mu}_t^{(j)}, \Sigma_t^{(j)})\, \mathrm{d}x_1$$

where $W_t = \Sigma\, w_t^{(j)}$. In particular,

$$E\{h_1(x_{t,1})|y_1, \ldots, y_t\} \approx W_t^{-1} \sum_{j=1}^{m} w_t^{(j)} \int h_1(x_1)\, \phi(x_1; \boldsymbol{\mu}_t^{(j)}, \Sigma_t^{(j)})\, \mathrm{d}x_1,$$

$$E\{h_2(x_{t,2})|y_1, \ldots, y_t\} \approx W_t^{-1} \sum_{j=1}^{m} w_t^{(j)}\, h_2(x_{t,2}^{(j)}).$$

4. Some numerical examples

4.1. Target tracking

Designing a sophisticated target tracking algorithm is an important task for both civilian and military surveillance systems, particularly when a radar, sonar or optical sensor is operated in the presence of clutter or when innovations are non-Gaussian (Bar-Shalom and Fortmann, 1998). We show three examples of target tracking using the MKF:

(a) targets in the presence of random interference (clutter);
(b) targets with non-Gaussian innovations;
(c) targets with manoeuvring.

4.1.1. Random (Gaussian) accelerated target in clutter

Suppose that the target follows a linear and Gaussian state space model

$$\begin{aligned} x_t &= Hx_{t-1} + Ww_t, \\ y_t &= Gx_t + Vv_t \end{aligned} \tag{5}$$

where x_t is the state variable (location and velocity) of the target and w_t and v_t are white noise with identity covariance matrix. For targets moving in a straight line, we have $x_t = (s_t, \nu_t)$ where s_t is the true target location and ν_t is its current velocity. In this case

$$H = \begin{pmatrix} 1 & T \\ 0 & T \end{pmatrix}, \qquad W = \sigma_w^2 \begin{pmatrix} T/2 \\ 1 \end{pmatrix}, \qquad G = (1, 0), \qquad V = \sigma_v^2, \tag{6}$$

where T is the time duration between two observations and the random acceleration is assumed to be constant in the period, with rate $\sigma_w^2 w_t / T$.

In a clutter environment, we observe m_t signals $\{z_{t1}, \ldots, z_{tm_t}\}$ at time t, with

$$m_t \sim \text{Bernoulli}(p_d) + \text{Poisson}(\lambda\Delta),$$

where p_d is the probability that a true signal y_t is detected, λ is the rate of a Poisson random field and Δ is the area of the surveillance region. In words, at time t the true signal has probability p_d to be detected, together with false signals, such as deceiving objects or electromagnetic interference, which are distributed as a Poisson point process in the space.

By letting Λ_t be the identifier of the target, Liu and Chen (1998) formulated the foregoing problem into a CDLM. More precisely, they let $\Lambda_t = 0$ if the target is not observed and $\Lambda_t = i$ if the ith observed object is the signal generated from the true target, i.e. $y_t = z_{ti}$. Then the system is linear and Gaussian with given Λ_t, and the remaining z signals bear no information. Some of their results are shown in Figs 1(a) and 1(b), which reveal the tracking errors (the differences between the estimated and true target locations) of 50 simulated runs of the tracking model, with $r^2 = 1.0$, $q^2 = 1.0$, $p_d = 0.9$ and $\lambda = 0.1$. 500 Monte Carlo samples were used for both the MKF and a standard Monte Carlo filter (i.e. a sampling–importance resampling algorithm with resampling applied to the state variable x_t). Here we also tested the split track filter (Fig. 1(c)), which, at each step, kept 500 trajectories with the highest likelihood values (recursively). The MKF performed much better than the other two algorithms in this problem.

4.1.2. Random (non-Gaussian) accelerated target (no clutter)

Consider again model (5), but with non-Gaussian errors w_t and v_t. Here we analyse the case when $w_t \sim t_{k_1}$ and $v_t \sim t_{k_2}$, but the same approach can be applied to other mixed Gaussian settings. By defining $\Lambda_t = (\Lambda_{t1}, \Lambda_{t2})$, where $\Lambda_{ti} \sim \chi^2_{k_i}$ independently, we can rewrite model (5) as

$$\begin{aligned} x_t &= Hx_{t-1} + (\sqrt{k_1}/\sqrt{\lambda_1})We_t, \\ y_t &= Gx_t + (\sqrt{k_2}/\sqrt{\lambda_2})V\epsilon_t, \end{aligned} \qquad \text{if } (\Lambda_{t1}, \Lambda_{t2}) = (\lambda_1, \lambda_2),$$

with $e_t \sim N(0, I)$ and $\epsilon_t \sim N(0, I)$.

Simulations were carried out with the matrices (6) with $T = 1$ and no interference and $w_t \sim t_3$ and $v_t \sim t_3$. Table 1 shows a comparison of the MKF and a standard Monte Carlo filter in terms of the number of times that the target was lost ($|x_t - \hat{x}_t| > 1200$) and the central processor unit (CPU) time for 100 simulated runs.

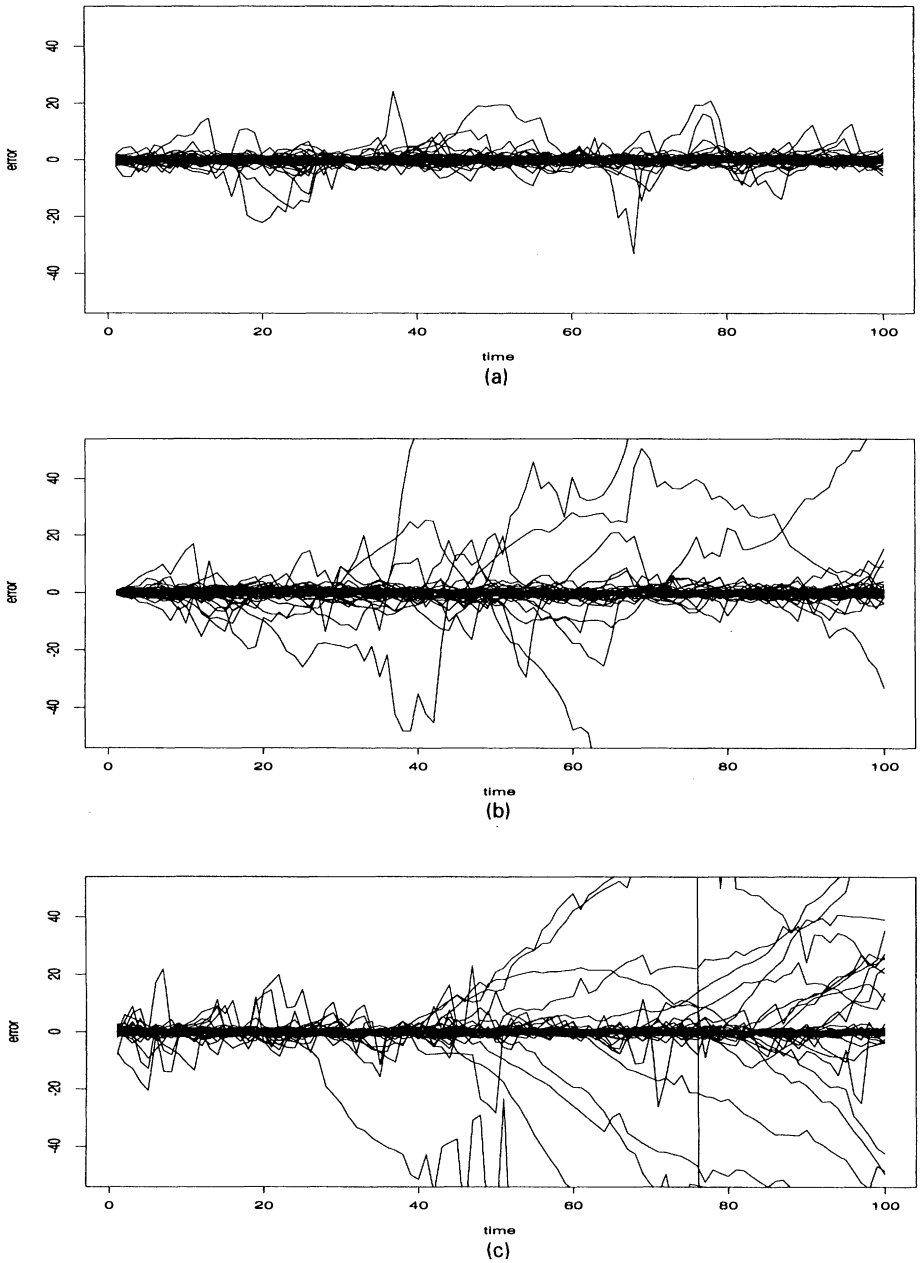

Fig. 1. Tracking errors of 50 runs of (a) the MKF, (b) a standard Monte Carlo filter and (c) the split track filter for a simulated one-dimensional target moving system

Fig. 2 shows the tracking mean-squared error (MSE), after the lost tracks have been eliminated. We observe that, although it takes about twice as much CPU time as the standard Monte Carlo filter with the same m, the MKF performs much more efficiently in the same CPU time.

We also tested the idea of using a finite mixture of Gaussian distributions to approximate

Table 1. Comparison of the MKF and a standard Monte Carlo filter

Noise variance	*Monte Carlo size m*	*Results for the Monte Carlo filter*		*Results for the MKF*	
		CPU time (s)	*Number of misses*	*CPU time (s)*	*Number of misses*
	20	9.49843	72	19.4277	1
	50	20.1622	20	51.6061	1
$\sigma_w^2 = 16.00$	200	80.3340	7	181.751	1
$\sigma_v^2 = 1600$	500	273.369	4	500.157	1
	1500	1063.36	3	2184.67	1

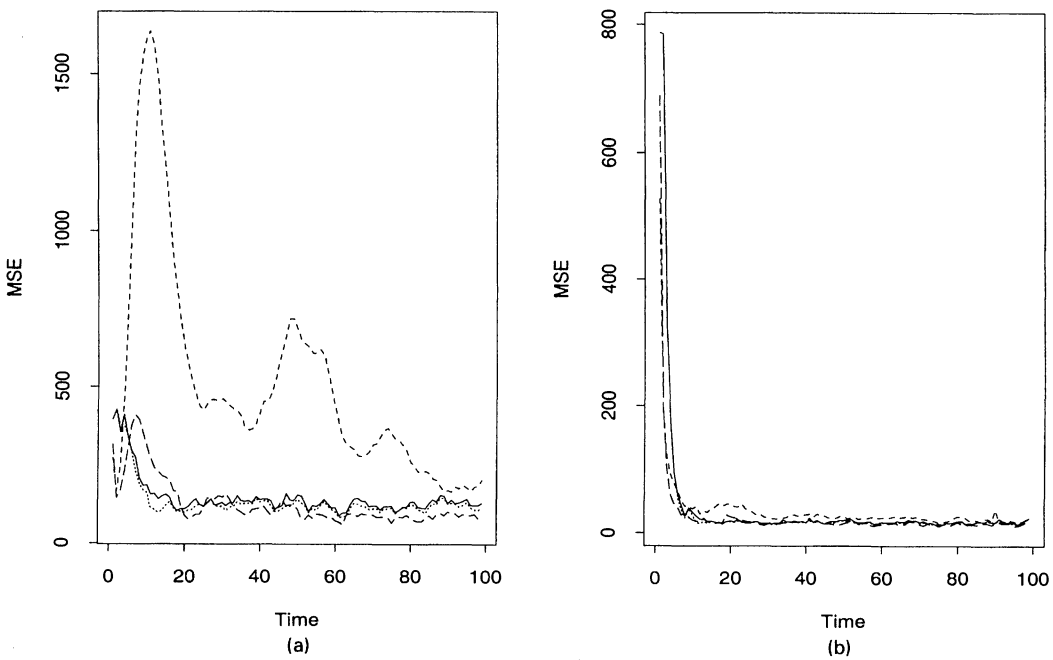

Fig. 2. MSEs of (a) location and (b) speed of 50 runs of the MKF and a standard Monte Carlo filter for a simulated one-dimensional target moving system with different Monte Carlo sample sizes (———, MKF, sample size 20, ·····, MKF, sample size 200; - - - - -, Monte Carlo filter, sample size 50; — — —, Monte Carlo filter, sample size 500)

the t-distribution, i.e. approximating t_3 with $\Sigma_{i=1}^{k} p_i N(0, \sigma_i^2)$. Similar results were obtained. The advantage of this approach is that a more efficient MKF can be used for discrete indicators. However, the approximation causes some biases.

4.1.3. Manoeuvred target (no clutter)

A manoeuvred target in a clean environment can be modelled as follows:

$$x_t = Hx_{t-1} + Fu_t + Ww_t,$$
$$y_t = Gx_t + Vv_t$$

where u_t is the manoeuvring acceleration. Here we consider an example of Bar-Shalom and

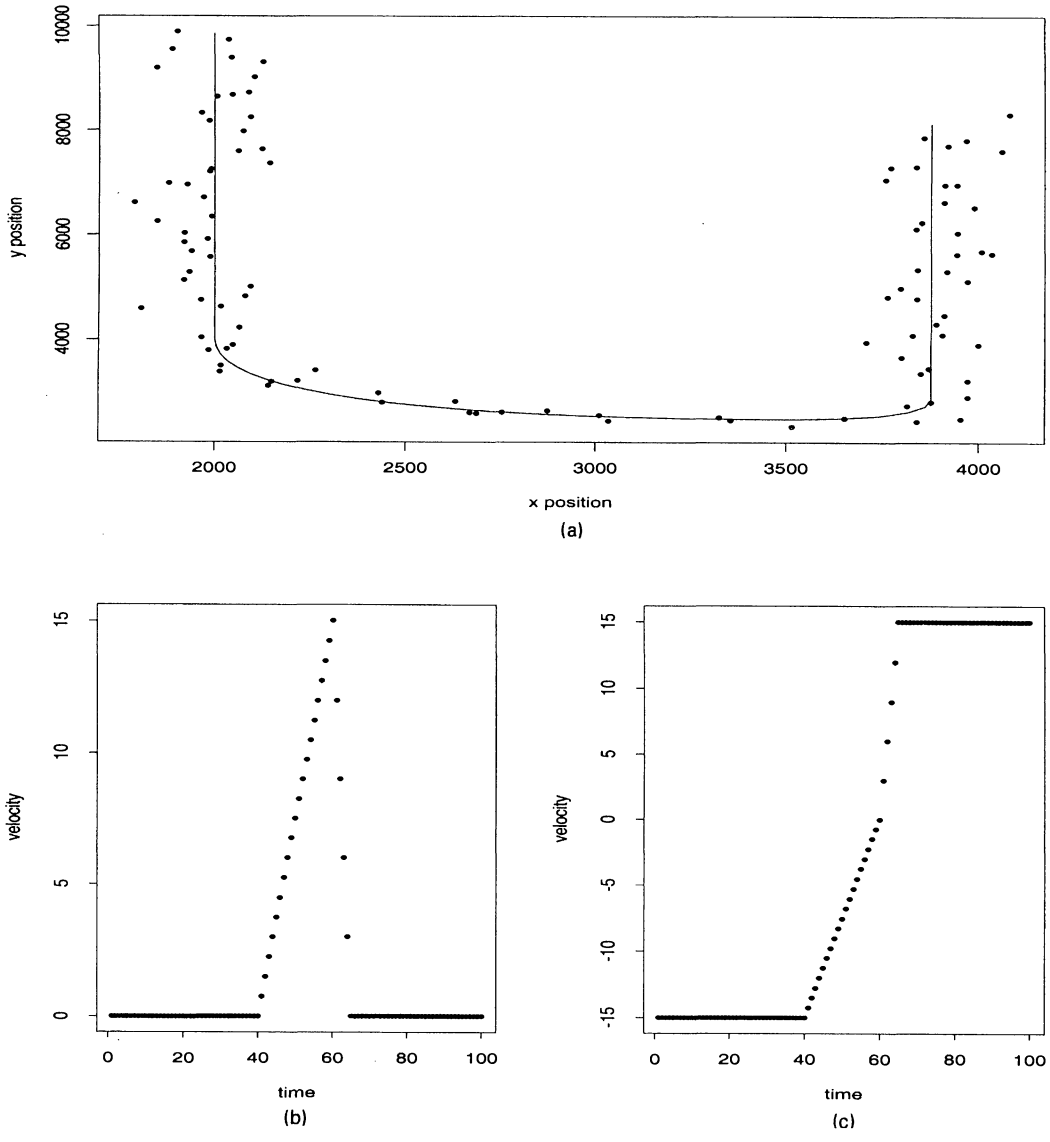

Fig. 3. (a) Position, (b) x-velocity and (c) y-velocity of a simulated two-dimensional manoeuvring target

Fortmann (1988) in which a two-dimensional target's position is sampled every $T = 10$ s. The target moves in a plane with constant course and speed until $k = 40$ when it starts a slow 90° turn which is completed in 20 sampling periods. A second, fast, 90° turn starts at $k = 61$ and is completed in five sampling times. Fig. 3 shows the trajectory of the target and its x-direction and y-direction velocity in one simulated run. Consequently, the matrices in this example are

$$H = \begin{pmatrix} 1 & 0 & 10 & 0 \\ 0 & 1 & 0 & 10 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \qquad G = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}, \qquad F = \begin{pmatrix} 5 & 0 \\ 0 & 5 \\ 1 & 0 \\ 0 & 1 \end{pmatrix},$$

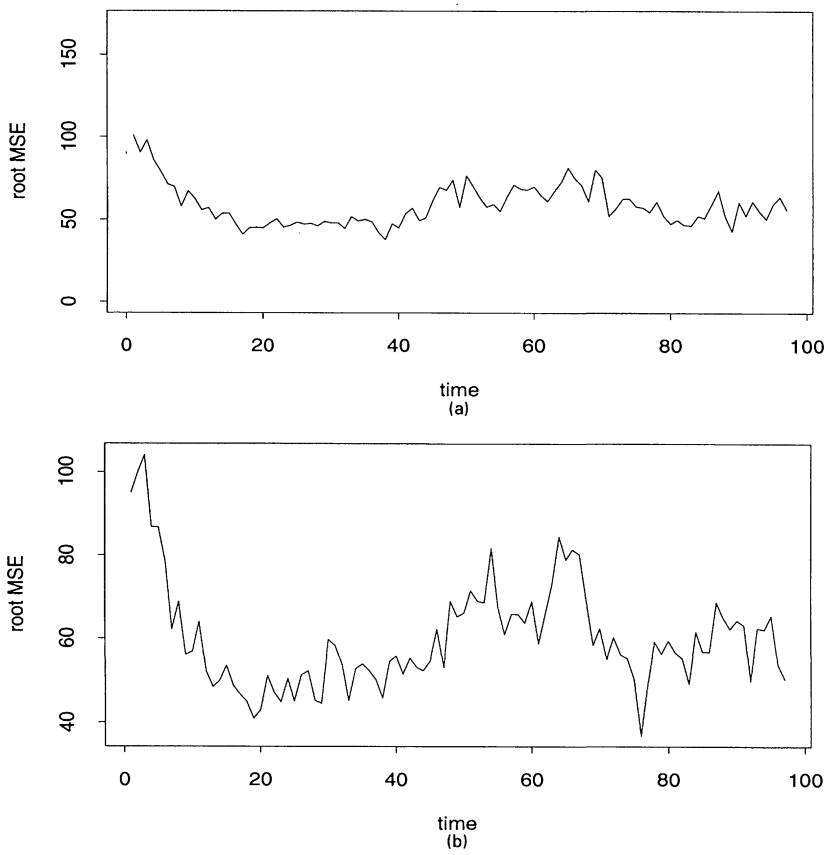

Fig. 4. Root-mean-squared errors of (a) the *x*-position and (b) the *x*-direction velocity of 50 runs of the MKF for a simulated two-dimensional target moving system with manoeuvring

$$W = \sigma_w^2 \begin{pmatrix} 5 & 0 \\ 0 & 5 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}, \qquad V = \sigma_v^2 \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

The slow turn is the result of acceleration inputs $u_t^x = u_t^y = 0.075$ $(40 < t \leqslant 60)$, and the fast turn is from $u_t^x = -u_t^y = -0.3$ $(61 < t \leqslant 65)$. Other u_ts are 0 (i.e. no manoeuvring).

To apply the MKF to this application, we need to specify the prior structure of u_t. First, we assume that manoeuvring can be classified into several categories, indicated by an indicator. In particular, we assume a three-level model: $I_t = 0$ indicates no manoeuvring ($u_t = 0$) and $I_t = 1$ and $I_t = 2$ indicate slow and fast manoeuvring respectively ($u_t \sim N(0, \sigma_i^2)$, $\sigma_1^2 < \sigma_2^2$). In this study we used $\sigma_1^2 = 1$ and $\sigma_2^2 = 36$. We also specify transition probabilities $P(I_t = j | I_{t-1} = i) = p_{ij}$ for the manoeuvring status. Specifically, we assume $p_{ii} = 0.8$ and $p_{ij} = 0.1$ for $i \neq j$ (i.e. the object is more likely to stay in a particular manoeuvring state than to change the manoeuvring state). Second, there are different ways of modelling the serial correlation of the u_t. Here we assume a multilevel white noise model, as in Bar-Shalom and Fortmann (1988),

where the u_t are assumed independent, given the indicator. This is the easiest but not a very realistic model. Other possible models are currently under investigation.

In Fig. 4 we present the root-mean-square errors of the MKF estimates of the target position for 50 simulated runs. Comparing our result with that of Bar-Shalom and Fortmann (1988), page 143, who used the traditional detection-and-switching method, we see a clear advantage of the proposed MKF.

4.2. *Digital signal extraction in fading channels*

Consider example 2 in Section 3.1 with binary input signals $s_t = \{1, -1\}$. The fading coefficient takes complex values, with independent real and imaginary parts following the same state equation. Simulations were done with the configurations

$$F = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0.9391 & -2.8763 & 2.9372 \end{pmatrix}, \qquad G' = 10^{-4} \begin{pmatrix} 0.0376 \\ 0.1127 \\ 0.1127 \\ 0.0376 \end{pmatrix},$$

$$W = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \qquad V = r,$$

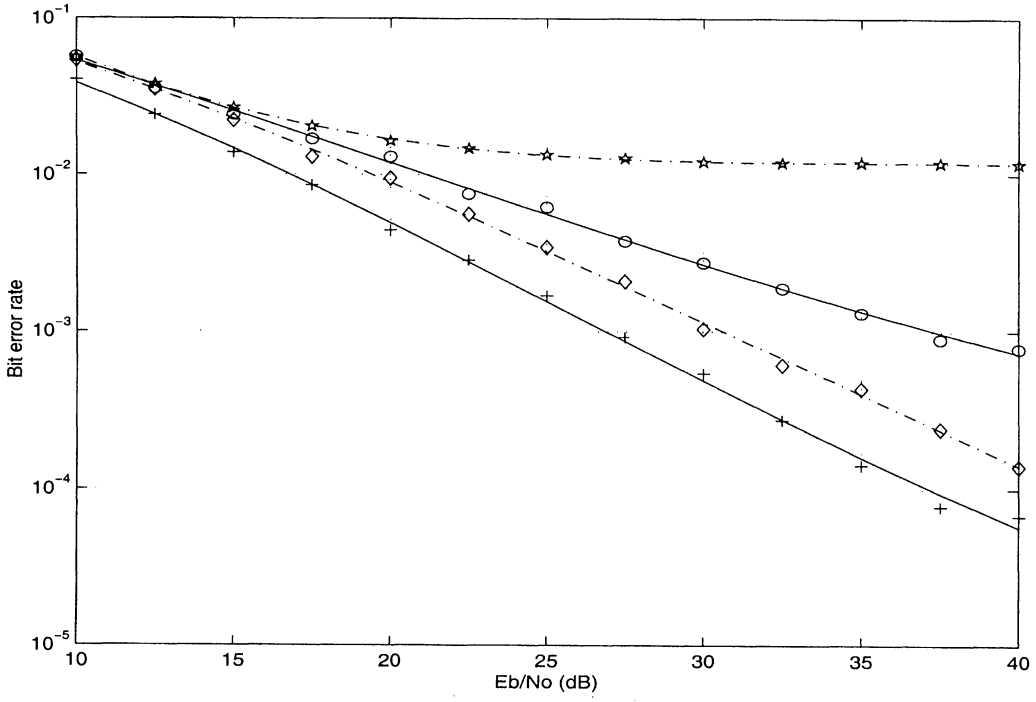

Fig. 5. Bit error rate of extracting differential binary signals from a fading channel by using differential detection (☆), the MKF (○) and the delayed MKF (◇): a lower bound that assumes exact knowledge of the fading coefficients is also shown (+)

i.e. both the real and the imaginary parts of α_t follow an autoregressive moving average ARMA(3, 3) process

$$\alpha_t - 0.9391\alpha_{t-1} + 2.8763\alpha_{t-2} - 2.9372\alpha_{t-3} = 0.0376e_t + 0.1127e_{t-1} + 0.1127e_{t-2} + 0.0376e_{t-3}$$

where $e_t \sim N(0, 0.01^2)$. In the communications literature, this is called a (low pass) Butterworth filter of order 3 with cut-off frequency 0.01. It is normalized to have a stationary variance 1.

We are interested in estimating the differential code $d_t = s_t s_{t-1}$. Fig. 5 shows the bit error rate of different signal-to-noise ratios, using the EMKF, the differential detection $\hat{d}_t = \text{sgn}\{\text{real}(y_t y_{t-1}^*)\}$ and a lower bound. The lower bound is obtained by using the true fading coefficients α_t and $\hat{d}_t = \text{sgn}\{\text{real}(\alpha_t^* y_t y_{t-1}^* \alpha_{t-1})\}$. The Monte Carlo sample size m was 100 for the MKF. We also include the result of a delayed estimation, in which s_t is estimated using the samples $s_t^{(j)}$ generated by the MKF, and the weight $w_{t+1}^{(j)}$ at time $t+1$ (Liu and Chen, 1998). This delayed estimation can utilize the substantial information contained in the future information y_{t+1}, and hence is more accurate because of the strong memory in the fading channel.

We can see that the simple differential detection works very well in low signal-to-noise cases and no significant improvement can be expected. However, it has an apparent bit error rate floor for high signal-to-noise cases. The MKF managed to break that floor, by using the structure of the fading coefficients.

5. Discussion

We have proposed the MKF for on-line estimation and prediction in CDLMs. The method was further extended to deal with PCDLMs. The MKF is a sequential Monte Carlo technique in which a *marginalization* operation is employed for improving its efficiency. All our numerical examples showed that the MKF approach gains significantly over the earlier sequential Monte Carlo approaches, e.g. the bootstrap filter and sequential imputation. Furthermore, we note that other Monte Carlo techniques such as the auxiliary variable method (Pitt and Shephard, 1999), Markov chain Monte Carlo updates (MacEachern *et al.*, 1999), delayed estimation (Liu and Chen, 1995) and the fixed lag filter (in an early version of Pitt and Shephard (1999)) can all be combined with the MKF to improve its effectiveness further. The development in this paper is thus an example to show that a *sequential Monte Carlo method* is a powerful platform for designing efficient non-linear filtering algorithms.

Acknowledgements

Rong Chen's research was partially supported by National Science Foundation (NSF) grant DMS-9626113 and the Terman fellowship from Stanford University. Jun Liu's research was partially supported by NSF grant DMS-9803649 and National Institutes of Health grant R01 HG01257-01. Part of the manuscript was written while Rong Chen was visiting the Department of Statistics, Stanford University. The authors thank Professor Xiaodong Wang for providing the fading channel example and two referees for helpful suggestions.

References

Ackerson, G. A. and Fu, K. S. (1970) On state estimation in switching environments. *IEEE Trans. Autom. Control*, **15**, 10–17.

Akashi, H. and Kumamoto, H. (1977) Random sampling approach to state estimation in switching environments. *Automatica*, **13**, 429–434.

Anderson, B. D. O. and Moore, J. B. (1979) *Optimal Filtering*. Englewood Cliffs: Prentice Hall.
Avitzour, D. (1995) A stochastic simulation Bayesian approach to multitarget tracking. *IEE Proc. Radar, Sonar Navign*, **142**, 41–44.
Bar-Shalom, Y. and Fortmann, T. E. (1988) *Tracking and Data Association*. Boston: Academic Press.
Berzuini, C., Best, N. G., Gilks, W. R. and Larizza, C. (1997) Dynamic conditional independence models and Markov chain Monte Carlo methods. *J. Am. Statist. Ass.*, **92**, 1403–1412.
Carlin, B. P., Polson, N. G. and Stoffer, D. S. (1992) A Monte Carlo approach to nonnormal and nonlinear state-space modeling. *J. Am. Statist. Ass.*, **87**, 493–500.
Carpenter, J., Clifford, P. and Fearnhead, P. (1997) An improved particle filter for non-linear problems. *Technical Report*. Department of Statistics, Oxford University, Oxford.
Carter, C. K. and Kohn, R. (1994) On Gibbs sampling for state space models. *Biometrika*, **81**, 541–553.
Doucet, A., Godsill, S. J. and Andrieu, C. (1999) On sequential Monte Carlo sampling methods for Bayesian filtering. *Statist. Comput.*, to be published.
Gelb, A. (1974) *Applied Optimal Estimation*. Boston: Massachusetts Institute of Technology Press.
Gordon, N. J., Salmond, D. J. and Smith, A. F. M. (1993) A novel approach to nonlinear/non Gaussian Bayesian state estimation. *IEE Proc. Radar Signal Process.*, **140**, 107–113.
Hürzeler, M. and Künsch, H. R. (1998) Monte Carlo approximations for general state-space models. *J. Comput. Graph. Statist.*, **7**, 175–193.
Jazwinski, A. (1970) *Stochastic Processes and Filtering Theory*. New York: Academic Press.
Kalman, R. E. (1960) A new approach to linear filtering and prediction problems. *J. Basic Engng*, **82**, 35–45.
Kitagawa, G. (1996) Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. *J. Comput. Graph. Statist.*, **5**, 1–25.
Kong, A., Liu, J. S. and Wong, W. H. (1994) Sequential imputations and Bayesian missing data problems. *J. Am. Statist. Ass.*, **89**, 278–288.
Liu, J. S. and Chen, R. (1995) Blind deconvolution via sequential imputations. *J. Am. Statist. Ass.*, **90**, 567–576.
——— (1998) Sequential Monte Carlo methods for dynamic systems. *J. Am. Statist. Ass.*, **93**, 1032–1044.
Liu, J. S., Wong, W. H. and Kong, A. (1994) Covariance structure of the Gibbs sampler with applications to the comparisons of estimators and augmentation schemes. *Biometrika*, **81**, 27–40.
MacEachern, S. N., Clyde, M. A. and Liu, J. S. (1999) Sequential importance sampling for non-parametric Bayes models: the next generation. *Can. J. Statist.*, **27**, 251–267.
Pitt, M. K. and Shephard, N. (1999) Filtering via simulation: auxiliary particle filters. *J. Am. Statist. Ass.*, **94**, 590–599.
Rubin, D. B. (1987) A noniterative sampling/importance resampling alternative to the data augmentation algorithm for creating a few imputations when fractions of missing information are modest: the SIR algorithm. *J. Am. Statist. Ass.*, **52**, 543–546.
Rubinstein, R. Y. (1981) *Simulation and the Monte Carlo Method*. New York: Wiley.
Shephard, N. (1994) Partial non-Gaussian state space. *Biometrika*, **81**, 115–131.
Smith, M. C. and Winter, E. M. (1978) On the detection of target trajectories in a multi-target environment. In *Proc. 17th Institution of Electrical and Electronics Engineers Conf. Decision Control, San Diego*. New York: Institution of Electrical and Electronics Engineers.
Svetnik, V. B. (1986) Applying the Monte Carlo method for optimum estimation in systems with random structure. *Autom. Remote Control*, **47**, 818–827.
Tanizaki, H. (1996) *Nonlinear Filters: Estimation and Applications*. New York: Springer.
Tugnait, J. K. (1982) Detection and estimation for abruptly changing systems. *Automatica*, **18**, 607–615.
West, M. (1992) Mixture models, Monte Carlo, Bayesian updating and dynamic models. *Comput. Sci. Statist.*, **24**, 325–333.
West, M. and Harrison, J. (1989) *Bayesian Forecasting and Dynamic Models*. New York: Springer.