Parameter Estimation for Discrete-Time Nonlinear Systems Using EM

Adrian Wills * **Thomas B. Schön** ** **Brett Ninness** *

* *School of Electrical Engineering and Computer Science, University of Newcastle, Callaghan, NSW 2308, Australia (e-mail: Adrian.Wills@newcastle.edu.au, and Brett.Ninness@newcastle.edu.au).*
** *Division of Automatic Control, Linköping University, SE-581 83 Linköping, Sweden (e-mail: schon@isy.liu.se).*

Abstract: In this paper we consider parameter estimation of general stochastic nonlinear state-space models using the Maximum Likelihood method. This is accomplished via the employment of an Expectation Maximisation algorithm, where the essential components involve a particle smoother for the expectation step, and a gradient-based search for the maximisation step. The utility of this method is illustrated with several nonlinear and non-Gaussian examples.

Keywords: Nonlinear estimation, maximum likelihood, system identification, expectation maximisation.

1. INTRODUCTION

One of the most general descriptions of dynamic system behaviour is a stochastic nonlinear state-space model of the form

$$x_{t+1} = f_t(x_t, u_t, w_t, \theta), \tag{1a}$$
$$y_t = h_t(x_t, u_t, v_t, \theta), \tag{1b}$$

where $x_t \in \mathbb{R}^{n_x}$ denotes the state variable, $u_t \in \mathbb{R}^{n_u}$ denotes the input signal (control signal), $y_t \in \mathbb{R}^{n_u}$ denotes the output signal (measurement), w_t and v_t denote the mutually independent i.i.d. process and measurement noise (respectively), and $\theta \in \mathbb{R}^{n_\theta}$ denotes model parameters.

Parametrization of the above model will largely depend on the situation at hand. Typically, however, this will involve a combination of physical insight in conjunction with more generic black-box structures [Ljung, 1999]. Given such a model structure, in this paper we are concerned with estimating the parameters θ based on the information about the system that is present in the observed inputs and outputs.

This is a nonlinear system identification problem [Ljung, 1999, Söderström and Stoica, 1989] and we will approach it using a maximum likelihood (ML) formulation, i.e. we seek an estimate of the parameter values $\widehat{\theta}$ via

$$\widehat{\theta} \triangleq \arg\max_{\theta} p_\theta(y_1, \ldots, y_N), \tag{2}$$

where $p_\theta(y_1, \ldots, y_N)$ denotes the joint likelihood of N output measurements, as postulated via model (1).

One approach for solving this maximum likelihood problem is to use an iterative gradient-based search procedure. This requires calculation of the likelihood and the predictor gradient $\frac{\partial}{\partial\theta} p(y_t|y_{1:t-1})$ for a given parameter value. This in turn requires the solution of a nonlinear filtering problem as shown in Coquelin et al. [2007], Poyiadjis et al. [2005], where some insight into approximating the filter gradients using Sequential Monte Carlo (SMC) methods was also provided.

In the present contribution we approach the maximum likelihood problem using an Expectation Maximisation (EM) algorithm. This has previously been considered for linear and bilinear systems by Gibson and Ninness [2005] and Gibson et al. [2005], respectively. A detailed discussion of EM as applied to Stochastic Volatility models can be found in [Kim, 2005]. However, the more general state-space model (1) is not considered in that work. Andrieu and Doucet [2003] have considered the case of on-line estimation using EM. They use a split-data likelihood criteria to avoid degeneration in the expectation step of EM, which shows very promising results.

In previous work by the authors [Schön et al., 2006], offline parameter estimation using the EM algorithm for nonlinear state-space models was also addressed. However, in contrast to the work here employing the very general state-space model (1) and placing (almost) no restriction on the distribution of $w_t.v_t$, in [Schön et al., 2006] the parametrization was required to be affine and the noise was required to be additive and Gaussian.

2. ML ESTIMATION AND THE EM ALGORITHM

The maximum likelihood problem in (2) can be restated in a more convenient form as the maximum log-likelihood problem

$$\widehat{\theta} \triangleq \arg\max_{\theta} L_\theta(Y), \qquad L_\theta(Y) \triangleq \log p_\theta(Y), \tag{3}$$

where $Y \triangleq \{y_1, \ldots, y_N\}$. The inherent difficulty in solving the above optimisation problem stems from the need to perform a nonlinear filtering operation in order to calculate $L_\theta(Y)$. This problem is amplified when gradient and Hessian information is also required. Nevertheless, this approach has been successfully applied by [**??**Coquelin et al., 2007, Poyiadjis et al., 2005]

Here, we take a different approach and employ the EM algorithm. As motivation for this, suppose that in addition to the output measurements Y we are also given measurements of the state $X \triangleq \{x_1, \ldots, x_{N+1}\}$, and based on all these measurements and the postulated system model in (1), we seek the maximum log-likelihood estimate of θ via

$$\widehat{\theta} \triangleq \arg\max_{\theta} L_\theta(X, Y), \tag{4}$$

$$L_\theta(X, Y) \triangleq \log p_\theta(X, Y). \tag{5}$$

Then, at least in principle, we could maximise L_θ using a similar iterative search procedure as mentioned above. This situation is not realistic, however, since the state is typically not measured. At the same time, we can regard this as the best possible scenario. That is to say, if solving the above problem is difficult, then we can expect solving (3) to become even more difficult.

This is a basic premise of the EM method; we should choose the *missing data* X such that if it were available, then solving (4) would be straightforward or at least easier than solving (3). The problem is that we don't have the missing data X. This motivates the first step (E-step) of the EM method, in which the joint likelihood $L_\theta(X, Y)$ is approximated using its *expected* value over the missing data X–based upon some current guess at the parameters θ'. That is, we approximate $L_\theta(X, Y)$ via its expected value $\mathbf{E}_{\theta'}\{L_\theta(X, Y) \mid Y\}$ conditional on the observed data Y and a current estimate θ' of the model parameters. This can be viewed as marginalisation of the missing data X,

$$L_\theta(X, Y) \approx \int L_\theta(X, Y) p_{\theta'}(X|Y) dX \triangleq Q(\theta, \theta'). \tag{6}$$

A remarkable feature of the EM algorithm is that maximising $Q(\theta, \theta')$ actually guarantees an increase of the likelihood $L_\theta(Y)$, which is our purpose in this paper. Indeed (see, for example Gibson and Ninness [2005])

$$Q(\theta, \theta') \triangleq L_\theta(Y) + \int \log p_\theta(X|Y) p_{\theta'}(X|Y) dX,$$

and therefore

$$L_\theta(Y) - L_{\theta'}(Y) =$$
$$Q(\theta, \theta') - Q(\theta', \theta') + \int \log \frac{p_{\theta'}(X|Y)}{p_\theta(X|Y)} p_{\theta'}(X|Y) dX. \tag{7}$$

Furthermore, the rightmost integral in (7) is the Kullback-Leibler divergence metric, which is therefore non-negative. Hence,

$$L_\theta(Y) - L_{\theta'}(Y) \geq Q(\theta, \theta') - Q(\theta', \theta'), \tag{8}$$

which implies that by increasing Q we in fact increase the likelihood $L_\theta(Y)$. It follows that at iteration k of the EM algorithm we proceed as follows

(1) (E-Step): Form the expected value of $L_\theta(X, Y)$ over the missing data X based on the current parameter estimate θ_k and the measurements Y via

$$Q(\theta, \theta_k) = \mathbf{E}_{\theta'}\{L_\theta(X, Y) \mid Y\} = \int L_\theta(X, Y) p_{\theta_k}(X|Y) dX. \tag{9}$$

(2) (M-Step): obtain a new estimate θ_{k+1} by maximising $Q(\theta, \theta_k)$ over θ, i.e.

$$\theta_{k+1} \triangleq \arg\max_{\theta} Q(\theta, \theta_k). \tag{10}$$

Iterating between these expectation and maximisation steps is known as the Expectation Maximisation (EM) algorithm [Dempster et al., 1977]. Clearly, its employment requires a mechanism for computing the expectation involved in $Q(\theta, \theta_k)$, and also a means for maximising $Q(\theta, \theta_k)$ over θ.

In terms of the expectation, there are very few situations where an exact and tractable solution exists (a well known exception is linear systems with additive Gaussian noise). As such, we will employ an approximation technique in this paper; namely we employ Sequential Monte Carlo methods to approximate the distribution $p_{\theta_k}(X|Y)$ in (9). The approximation consists of a sum of weighted delta functions, which allows us to convert the integral in $Q(\theta, \theta_k)$ into a finite sum as shown in the next section.

3. EXPECTATION STEP

The expectation step corresponds to computing $Q(\theta, \theta_k)$. This may be performed by first noting that via Bayes' rule and the Markov properties associated with the model structure (1)

$$p_\theta(X, Y) = p_\theta(Y \mid X) p_\theta(X) \tag{11}$$

$$= p_\theta(x_1) \prod_{t=1}^{N} p_\theta(x_{t+1} \mid x_t) p(y_t \mid x_t) \tag{12}$$

Taking logarithms and conditional expectations as per the definition (5), (6) then delivers

$$\begin{aligned} Q(\theta, \theta_k) = &\int \log p_\theta(x_1) p_{\theta_k}(X|Y) dx_1 \\ &+ \sum_{t=1}^{N} \int \log p_\theta(x_{t+1}|x_t) p_{\theta_k}(x_{t+1}|Y) dx_t \\ &+ \sum_{t=1}^{N} \int \log p_\theta(y_t|x_t) p_{\theta_k}(x_t|Y) dx_t, \end{aligned} \tag{13}$$

which explains why we are interested in the marginal smoothing density $p(x_t|Y)$, rather than the complete joint density $p(X|Y)$. These marginal smoothing densities will be approximated using sequential Monte Carlo methods, resulting in (δ is the Dirac delta)

$$p(x_t|Y) \approx \sum_{i=1}^{M} \tilde{q}_{t|N}^{(i)} \delta(x_t - x_{t|N}^{(i)}). \tag{14}$$

The details regarding the computation of this approximation are given in the subsequent section. Substituting (14) in (13) provides the desired approximation $\widehat{Q}(\theta, \theta_k)$ of $Q(\theta, \theta_k)$

$$\begin{aligned} \widehat{Q}(\theta, \theta_k) = &\sum_{i=1}^{M} \tilde{q}_{1|N}^{(i)} \log p_\theta(x_{1|N}^{(i)}) \\ &+ \sum_{t=1}^{N} \sum_{i=1}^{M} \tilde{q}_{t+1|N}^{(i)} \log p_\theta(x_{t+1|N}^{(i)}|x_t) \\ &+ \sum_{t=1}^{N} \sum_{i=1}^{M} \tilde{q}_{t|N}^{(i)} \log p_\theta(y_t|x_{t|N}^{(i)}). \end{aligned} \tag{15}$$

In order to maximise $Q(\theta, \theta_k)$ we will typically need gradients and possibly Hessians. It is straightforward to approximate them using (14) as well. For the gradient we have

$$\begin{aligned} \frac{\partial Q}{\partial \theta} &= \frac{\partial}{\partial \theta} \int \log p_\theta(X, Y) p_{\theta_k}(X|Y) dX \\ &= \int \frac{\partial \log p_\theta(X, Y)}{\partial \theta} p_{\theta_k}(X|Y) dX. \end{aligned}$$

This is an expression which is exactly in the same form as (6). Hence, we can use (14) to approximate the gradients as well (similarly for the Hessian).

4. PARTICLE METHODS

Here it is worth noticing that, similar to the present contribution, Coquelin et al. [2007] and Poyiadjis et al. [2005] make use of the *marginal* density $p_\theta(x_t|Y)$, rather than the joint density $p_\theta(x_{1:N}|Y)$. Approaches based on the joint density will inevitably run into problems as the sample size N increases [Andrieu et al., 2004].

This section will describe how to obtain an estimate of $p_\theta(x_t|Y)$ in the form (14) using particle methods. Inspired by the work of Tanizaki [2001, 2004], let us consider the problem of generating random numbers distributed according to some *target density* $t(x)$, which potentially is rather complex. One way of doing this would be to employ an alternate density that is simple to draw from, say $s(x)$, referred to as the *sampling density*, and then calculate the probability that the sample was in fact generated from the target density. That is, a sample $x^{(i)} \sim s(x)$ is drawn, and then the following ratio is calculated

$$a(x^{(i)}) \propto t(x^{(i)})/s(x^{(i)}),$$

which indicates how probable it is that $x^{(i)}$ is in fact generated from the target density $t(x)$.

The probability of accepting $x^{(i)}$ as a sample from $t(x)$ is referred to as the *acceptance probability*, and typically it is computed via consideration of $a(x^{(i)})$. This is the case, for example, for all of the so-called "rejection sampling", "importance sampling/resampling" and "Metropolis–Hastings independence sampling" methods [Tanizaki, 2001, Liu, 1996]. This paper employs importance resampling.

4.1 Particle Filter

In the case of filtering, the target density referred to above becomes $t(x_t) = p(x_t|Y_t)$, and it is then necessary to also choose an appropriate sampling density $s(\cdot)$ and acceptance probability. This is in fact quite simple, since from Bayes' theorem and the Markov property

$$\begin{aligned} p(x_t|Y_t) = p(x_t|y_t, Y_{t-1}) &= \frac{p(y_t|x_t)p(x_t|Y_{t-1})}{p(y_t|Y_{t-1})} \\ &\propto p(y_t|x_t)p(x_t|Y_{t-1}) \end{aligned}$$

which suggests, since $t(x) \propto a(x)s(x)$, the following choices

$$\underbrace{p(x_t|Y_t)}_{t(x_t)} \propto \underbrace{p(y_t|x_t)}_{a(x_t)}\underbrace{p(x_t|Y_{t-1})}_{s(x_t)}.$$

Via the principle of importance resampling the acceptance probabilities, $\{\tilde{a}^{(i)}\}_{i=1}^M$, are calculated according to

$$\tilde{a}^{(i)} = \frac{a(x_{t|t-1}^{(i)})}{\sum_{j=1}^M a(x_{t|t-1}^{(j)})} = \frac{p(y_t|x_{t|t-1}^{(i)})}{\sum_{j=1}^M p(y_t|x_{t|t-1}^{(j)})},$$

where $x_{t|t-1}^{(i)} \sim p(x_t|Y_{t-1})$. That is, acceptance probabilities $\tilde{a}^{(i)}$ depend upon computation of $p(y_t|x_{t|t-1})$.

The algorithm then proceeds by obtaining samples from $p(x_t|Y_t)$ by resampling the particles $\{x_{t|t-1}^{(i)}\}_{i=1}^M$ from the sampling density $p(x_t|Y_{t-1})$ according to the corresponding acceptance probabilities $\{\tilde{a}^{(i)}\}_{i=1}^M$. If this procedure is recursively repeated over time the following approximation

$$p(x_t|Y_t) \approx \sum_{i=1}^M \frac{1}{M}\delta(x_t - x_{t|t}^{(i)}) \tag{16}$$

is obtained, and we have in fact derived the *particle filter* algorithm, which is given below in Algorithm 1. It was first introduced by Gordon et al. [1993].

Algorithm 1. Particle filter

(1) Initialise the particles, $\{x_{0|-1}^{(i)}\}_{i=1}^M \sim p_{x_0}(x_0)$.

(2) Calculate weights $\{q_t^{(i)}\}_{i=1}^M$ according to
$$q_t^{(i)} = p(y_t|x_{t|t-1}^{(i)})$$
and normalize $\tilde{q}_t^{(i)} = q_t^{(i)}/\sum_{j=1}^M q_t^{(j)}$.

(3) Resample N particles according to
$$\Pr(x_{t|t}^{(i)} = x_{t|t-1}^{(j)}) = \tilde{q}_t^{(j)}$$

(4) For $i = 1, \ldots, M$, predict new particles according to $x_{t+1|t}^{(i)} \sim p(x_{t+1|t}|x_{t|t}^{(i)})$.

(5) Set $t := t + 1$ and iterate from step 2.

4.2 Particle Smoother

In solving the smoothing problem the target density becomes $t(x_{t+1}, x_t) = p(x_{t+1}, x_t|Y)$. Similarly to what was discussed in the previous section we have to find a suitable sampling density and the corresponding acceptance probabilities to solve the smoothing problem. Again, using Bayes' theorem we have

$$p(x_{t+1}, x_t|Y) = p(x_t|x_{t+1}, Y)p(x_{t+1}|Y) \tag{17}$$

where

$$\begin{aligned} p(x_t|x_{t+1}, Y) &= p(x_t|x_{t+1}, Y_t, Y_{t+1:N}) \\ &= \frac{p(Y_{t+1:N}|x_t, x_{t+1}, Y_t)p(x_t|x_{t+1}, Y_t)}{p(Y_{t+1:N}|x_{t+1}, Y_t)} \\ &= p(x_t|x_{t+1}, Y_t) = \frac{p(x_{t+1}|x_t)p(x_t|Y_t)}{p(x_{t+1}|Y_t)}. \end{aligned} \tag{18}$$

Inserting (18) into (17) gives

$$\underbrace{p(x_{t+1}, x_t|Y)}_{t(x_{t+1},x_t)} = \underbrace{\frac{p(x_{t+1}|x_t)}{p(x_{t+1}|Y_t)}}_{a(x_{t+1},x_t)}\underbrace{p(x_t|Y_t)p(x_{t+1}|Y)}_{s(x_{t+1},x_t)}.$$

At time t the sampling density can be used to generate samples. In order to find the acceptance probabilities $\{a^{(i)}\}_{i=1}^M$ we have to calculate

$$a(x_{t+1}, x_t) = \frac{p(x_{t+1}|x_t)}{p(x_{t+1}|Y_t)},$$

where $p(x_{t+1}|x_t)$ is calculated using the model (1), and $p(x_{t+1}|Y_t)$ can be approximated according to

$$\begin{aligned} p(x_{t+1}|Y_t) &= \int p(x_{t+1}|x_t)p(x_t|Y_t)dx_t \\ &\approx \sum_{j=1}^M \frac{1}{M}p(x_{t+1}|x_{t|t}^{(j)}), \end{aligned}$$

where (16) has been used. The particles can now be resampled according to the normalised acceptance probabilities

$\{\tilde{a}^{(i)}\}_{i=1}^M$ in order to generate samples from $p(x_{t+1}, x_t|Y)$. The above discussion can be summarised in the following algorithm (first introduced by Tanizaki [2001]).

Algorithm 2. Particle smoother

(1) Run the particle filter (Algorithm 1) and store the filtered particles, $\{x_{t|t}^{(i)}\}_{i=1}^M$, $t = 1, \dots, N$.
(2) Initialise the smoothed particles and importance weights at time N according to $\{x_{N|N}^{(i)} = x_{N|N}^{(i)}, \tilde{q}_{N|N}^{(i)} = 1/M\}_{i=1}^M$ and set $t := t - 1$.
(3) Calculate weights $\{q_{t|N}^{(i)}\}_{i=1}^M$ according to
$$q_{t|N}^{(i)} = \frac{p(x_{t+1|N}^{(i)}|x_{t|t}^{(i)})}{\sum_{j=1}^M p(x_{t+1|N}^{(i)}|x_{t|t}^{(j)})}$$
and normalise $\tilde{q}_{t|N}^{(i)} = q_{t|N}^{(i)}/\sum_{j=1}^M q_{t|N}^{(j)}$.
(4) Resample the smoothed particles according to
$$\Pr(x_{t+1|N}^{(i)}, x_{t|N}^{(i)}) = (x_{t+1|N}^{(j)}, x_{t|t}^{(j)}) = \tilde{q}_{t|N}^{(j)}$$
(5) Set $t := t - 1$ and iterate from step 3.

5. MAXIMISATION STEP

Recall that the EM method comprises the expectation step as described in Sections 3 and 4, and the maximisation step, which is the subject of the current section. It was mentioned in Section 2 that the choice of missing data is often made so that maximising $Q(\theta, \theta_k)$ is straightforward. As such, it is difficult to prescribe an *efficient* method for solving the maximisation step since it will necessarily change on a case-by-case basis. Nevertheless, here we are interested in a general approach that is applicable whenever $Q(\theta, \theta_k)$ and its gradient with respect to θ exist, but makes no attempt to exploit any underlying structure of the specific problem at hand.

In order to use the EM method, we require the following steps to be performed (prior to calling the EM routine).

(1) Form the expression (i.e. write software) for $\widehat{Q}(\theta, \theta_k)$; this depends on θ, the weights $\tilde{q}_{t|N}^{(i)}$ and the smoothed particles $x_{t|N}^{(i)}$ (see Sections 3, 4.1 and 4.2).
(2) Form the expression (i.e. write software) for $\nabla_\theta \widehat{Q}(\theta, \theta_k)$
$$\nabla_\theta \widehat{Q}(\theta, \theta_k) \triangleq \sum_{i=1}^M \tilde{q}_{1|N}^{(i)} \frac{\partial \log p_\theta(x_{1|N}^{(i)})}{\partial \theta} + \sum_{t=1}^N \sum_{i=1}^M \tilde{q}_{t+1|N}^{(i)} \frac{\partial \log p_\theta(x_{t+1|N}^{(i)}|x_t)}{\partial \theta} + \sum_{t=1}^N \sum_{i=1}^M \tilde{q}_{t|N}^{(i)} \frac{\partial \log p_\theta(y_t|x_{t|N}^{(i)})}{\partial \theta}. \quad (19)$$

With this software available, then we can maximise $\widehat{Q}$ using any practical gradient-based search procedure. Note that for the numerical illustrations in Section 7, we have used a standard Quasi-Newton method (see e.g. [Nocedal and Wright, 2006, Dennis and Schnabel, 1983, Fletcher, 1987]). Furthermore, note that if the gradient of $\widehat{Q}$ is difficult to obtain, then numerical differentiation can be used instead.

6. THE ALGORITHM

The EM method described in this paper can be summarised by the following algorithm.

Algorithm 3. (EM algorithm). Given a model in the form of (1) and an initial parameter estimate θ_0, then set $k = 0$ and perform the following steps:

(1) Compute the filtered weights and particles $(\tilde{q}_{t|t}^{(i)}, x_{t|t}^{(i)})$ via Algorithm 1 based on θ_k.
(2) Compute the smoothed weights and particles $(\tilde{q}_{t|N}^{(i)}, x_{t|N}^{(i)})$ via Algorithm 2 based on $(\tilde{q}_{t|t}^{(i)}, x_{t|t}^{(i)})$ and θ_k.
(3) Using θ_k and $(\tilde{q}_{t|N}^{(i)}, x_{t|N}^{(i)})$, maximise $\widehat{Q}(\theta, \theta_k)$ as given in (15) via gradient based search to form
$$\theta_{k+1} \triangleq \arg\max_\theta \widehat{Q}(\theta, \theta_k). \quad (20)$$
(4) Set $k \leftarrow k + 1$ and goto Step 1.

The above algorithm was trialled on several numerical examples, which are profiled in the following section.

7. NUMERICAL ILLUSTRATIONS

In this section we demonstrate the utility of Algorithm 3 through several simulation examples. The first example considers a linear time-invariant state-space model, which is known to have a exact solution (i.e. to machine precision) for both the expectation and maximisation steps Gibson and Ninness [2005]. The reason for including this example is to profile the exact solution against that obtained using Algorithm 3.

The remaining examples involve various nonlinearities in either the state-transition or measurement equations, which render parameter estimation a more challenging task.

7.1 Linear Gaussian System

Consider the following linear state-space model
$$x_{t+1} = ax_t + bu_t + w_t, \quad (21a)$$
$$y_t = cx_t + du_t + e_t, \quad (21b)$$
$$\begin{bmatrix} w_t \\ e_t \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} q & s \\ s & r \end{bmatrix} \right). \quad (21c)$$
Given measurements of the input u_t and the output y_t for $t = 1, \dots, N$, we would like to estimate the parameters $\theta^T = [a,\ b,\ c,\ d,\ q,\ s,\ r]$.

A realisation from (21) was obtained with the input selected as $u_t \sim \mathcal{N}(0, 1)$ and for $N = 1000$ samples and with $\theta^T = [0.9,\ 0.8,\ 0.5,\ 0.2,\ 0.01,\ 0,\ 0.01]$. Both the exact EM algorithm from Gibson and Ninness [2005] and Algorithm 3 were employed and the evolution of the parameter estimates are shown in Figure 1. The initial parameter estimates were selected as $\theta_0^T = [0.5,\ 0.5,\ 0.5,\ 0.5,\ 1,\ 0,\ 1]$ and Algorithm 3 was used with $M = 50$ particles.

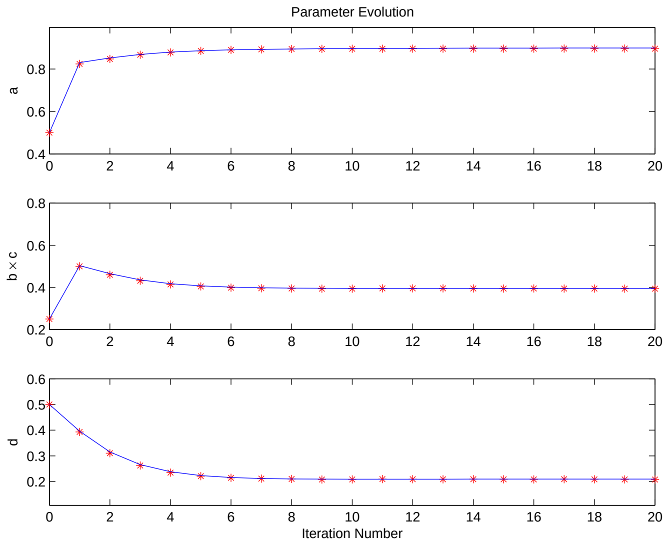

Fig. 1. Example 1: Evolution of the parameter values for the linear system (21). The exact EM (solid) is profiled against Algorithm 3 (shown as *'s).

Note that this parametrisation is not strictly identifiable since similarity transformations of the state will result in input-output equivalent systems; in fact, for this example only the (b, c) pair will not be identifiable. Therefore, we have plotted $b \times c$, which is independent of similarity transformations. The extremely close agreement between the Figure 1 results obtained via 'exact' Kalman smoothing, and approximate particle smoothing indicates gives some confidence that the approach will be viable in cases where an exact solution is not available. To such situations will now be profiled.

7.2 Stochastic Volatility System

With the favourable results from Section 7.1 we considered the more challenging problem of parameter estimation for a stochastic volatility model. This model is used, for example, to predict changes in the variance (or *volatility*) of stock prices and exchange rates. The stochastic volatility model can be described as

$$\begin{aligned} x_{t+1} &= a x_t + b w_t, && \text{(22a)} \\ y_t &= c e^{x_t/2} e_t, && \text{(22b)} \\ w_t &\sim \mathcal{N}(0,1), && \text{(22c)} \\ e_t &\sim \mathcal{N}(0,1), && \text{(22d)} \\ x_0 &\sim \mathcal{N}\left(0, b^2/(1-a^2)\right). && \text{(22e)} \end{aligned}$$

In this case the parameters to be estimated are $\theta^T = [a,\ b,\ c]$.

In accordance with the literature for this problem, we simulated the system for $\theta^T = [0.85,\ 0.35,\ 0.65]$ and recorded $N = 10000$ samples of the output y_t. Starting from the initial guess of $\theta_0^T = [0.5,\ 0.5,\ 0.5]$ we used Algorithm 3, save for the fact that we used a different particle smoother for this example. Rather than using Algorithm 2, we used the particle smoother proposed by Godsill et al. [2004]. Here it is important to note that it is straightforward to change smoothing algorithm, due to the general nature of Algorithm 3. Again $M = 50$ particles were used, and the parameter estimates are shown in Figure 2.

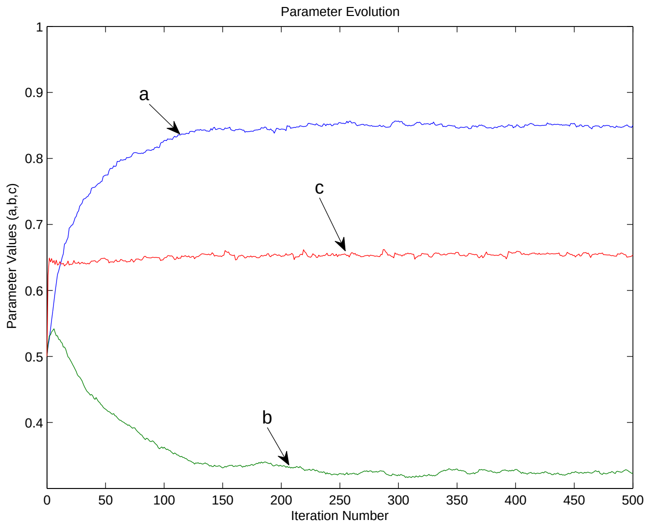

Fig. 2. Example 2: Evolution of the parameter values for the stochastic volatility model (22).

Despite using only 50 particles, the parameters converged towards to "true" values. However, we are using a large number of data points and there appears to be a bias on the b parameter for this data set. It was speculated that this effect was due to the small number of particles being used. Therefore, we increased the number to $M = 100$ and ran the algorithm again with the same data set as before, but observed only marginally better results.

To investigate this further, we performed a Monte Carlo test with 100 simulations as described above, and the results are summarised in Table 1. This shows that the EM method of Algorithm 3 appears to produce inconsistent estimates for this example under the above conditions.

Table 1. True and estimated parameter values for Example 7.2; mean value and standard deviations are shown for the estimates based on 100 Monte Carlo runs.

Parameters	True Values	Estimates
a	0.8500	0.8496 ± 0.0119
b	0.3500	0.3280 ± 0.0165
c	0.6500	0.6602 ± 0.0085

It is difficult to gauge this bias against that (if any) produced by competing off-line methods since, to the authors' knowledge, no such Monte Carlo tests have been reported.

7.3 A Synthetic Nonlinear System

As a final example we considered the synthetic state-space model

$$\begin{aligned} x_{t+1} &= a x_t + \frac{x_t}{b + x_t^2} + u_t + w_t, && \text{(23a)} \\ y_t &= c x_t + d x_t^2 + e_t, && \text{(23b)} \end{aligned}$$

where u_t is a known input signal that was selected as a sequence of random numbers, each distributed according to $\mathcal{N}(0, 1)$. Here, the parameters to be estimated are $\theta^T = [a,\ b,\ c,\ d,\ q,\ r]$, where q and r are the covariance of w_t and e_t, respectively. For brevity we report only (a, b, c, d) here.

This system was chosen because both the state transition and measurement equations are nonlinear. In addition, the parameters do not appear linearly in the model since b appears in the denominator. We simulated this system with $\theta^T = [0.7,\ 0.6,\ 0.5,\ 0.4,\ 0.01,\ 0.01]$ using $N = 1000$ samples. Algorithm 3 was employed with initial guess $\theta_0^T = [0.2,\ 0.2,\ 0.2,\ 0.2,\ 1,\ 1]$ and using $M = 50$ particles. The parameter values are shown in Figure 3.

It appears that Algorithm 3 performs quite well on this example for this data set. To examine the algorithm further, we performed a Monte Carlo test with 100 simulations as described above. The results are given in Table 2, which shows that the algorithm produces consistent estimates for this example.

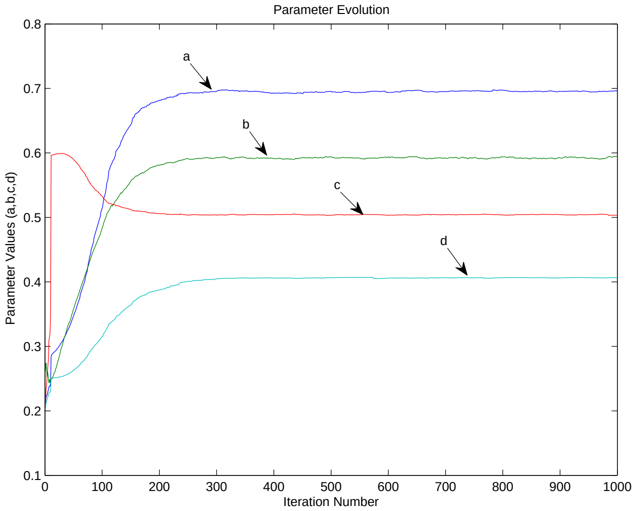

Fig. 3. Example 3: Evolution of the parameter values for state-space model (23).

Table 2. True and estimated parameter values for Example 7.3; mean and standard deviations are shown for the estimates based on 100 Monte Carlo runs.

Parameters	True Values	Estimates
a	0.7000	0.7010 ± 0.0072
b	0.6000	0.6007 ± 0.0057
c	0.5000	0.4999 ± 0.0027
d	0.4000	0.4052 ± 0.0085

8. CONCLUSION

The contribution in this paper is an EM algorithm for solving the parameter estimation problem in general stochastic nonlinear state-space models. Our experience from using the proposed algorithm is that it would probably benefit from an improved smoothing step. The plug-and-play nature of Algorithm 3 implies that it is straightforward to use it with a different smoother. Most particle smoothing algorithms (if not all) use the particles from the filtering step and just recompute the weights. However, it would be interesting to derive an algorithm that changes the support (i.e. the particles) as well.

REFERENCES

C. Andrieu and A. Doucet. Online expectation-maximization type algorithms for parameter estimation in general state space models. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 6, pages 69–72, Hong Kong, April 2003.

C. Andrieu, A. Doucet, S. S. Singh, and V. B. Tadić. Particle methods for change detection, system identification, and contol. *Proceedings of the IEEE*, 92(3):423–438, March 2004.

C. Andrieu, A. Doucet, and V. B. Tadić. On-line parameter estimation in general state-space models. In *Proceedings of the 44th Conference on Decision and Control*, pages 332–337, Seville, Spain, December 2005.

P.-A. Coquelin, R. Deguest, and R. Munos. Numerical methods for sensitivity analysis of Feynman-Kac models. Technical Report INRIA-00125427, INRIA, France, January 2007.

A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977.

J. E. Dennis and R. B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations.* Prentice Hall, 1983.

R. Fletcher. *Practical Methods of Optimization.* John Wiley & Sons, Chichester, UK, second edition, 1987.

S. Gibson and B. Ninness. Robust maximum-likelihood estimation of multivariable dynamic systems. *Automatica*, 41(10):1667–1682, 2005.

S. Gibson, A. Wills, and B. Ninness. Maximum-likelihood parameter estimation of bilinear systems. *IEEE Transactions on Automatic Control*, 50(10):1581–1596, 2005.

S. J. Godsill, A. Doucet, and M. West. Monte Carlo smoothing for nonlinear time series. *Journal of the American Statistical Association*, 99(465):156–168, March 2004.

N. J. Gordon, D. J. Salmond, and A. F. M. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. In *IEE Proceedings on Radar and Signal Processing*, volume 140, pages 107–113, 1993.

J. Kim. *Parameter Estimation in Stochastic Volatility Models with Missing Data Using Paticle Methodss and the EM Algorithm.* PhD thesis, Department of Statistics, University of Pittsburgh, 2005.

J. S. Liu. Metropolized independent sampling with comparison to rejection ampling and importance sampling. *Statistics and Computing*, 6:113–119, 1996.

L. Ljung. *System identification, Theory for the user.* System sciences series. Prentice Hall, Upper Saddle River, NJ, USA, second edition, 1999.

J. Nocedal and S. J. Wright. *Numerical Optimization.* Springer Series in Operations Research. Springer, New York, USA, 2 edition, 2006.

G. Poyiadjis, A. Doucet, and S. S. Singh. Maximum likelihhod parameter estimation in general state-space models using particle methods. In *Proceedings of the American Statistical Association*, Minneapolis, USA, August 2005.

T. B. Schön, A. Wills, and B. Ninness. Maximum likelihood nonlinear system estimation. In *Proceedings of the 14th IFAC Symposium on System Identification*, pages 1003–1008, Newcastle, Australia, March 2006.

T. Söderström and P. Stoica. *System identification.* Systems and Control Engineering. Prentice Hall, 1989.

H. Tanizaki. Nonlinear and non-Gaussian state space modeling using sampling techniques. *Annals of the Institute of Statistical Mathematics*, 53(1):63–81, 2001.

H. Tanizaki. *Computational Methods in Statistics and Economics.* Marcel Dekker, New York, NY, USA, 2004.