System identification of nonlinear state-space models

Thomas Schön, Adrian Wills and Brett Ninness

Original Publication:

Thomas Schön, Adrian Wills and Brett Ninness, System identification of nonlinear state-space models, 2011, AUTOMATICA, (47), 1, 39-49.
http://dx.doi.org/10.1016/j.automatica.2010.10.013

System Identification of Nonlinear State-Space Models ⋆

Thomas B. Schön [a], Adrian Wills [b], Brett Ninness [b]

[a] *Division of Automatic Control, Linköping University, SE-581 83 Linköping, Sweden*

[b] *School of Electrical Engineering and Computer Science, University of Newcastle, Callaghan, NSW 2308, Australia*

Abstract

This paper is concerned with the parameter estimation of a general class of nonlinear dynamic systems in state-space form. More specifically, a Maximum Likelihood (ML) framework is employed and an Expectation Maximisation (EM) algorithm is derived to compute these ML estimates. The Expectation (E) step involves solving a nonlinear state estimation problem, where the smoothed estimates of the states are required. This problem lends itself perfectly to the particle smoother, which provide arbitrarily good estimates. The maximisation (M) step is solved using standard techniques from numerical optimisation theory. Simulation examples demonstrate the efficacy of our proposed solution.

Key words: System identification, nonlinear models, dynamic systems, Monte Carlo method, smoothing filters, expectation maximisation algorithm, particle methods.

1 Introduction

The significance and difficulty of estimating nonlinear systems is widely recognised [1, 31, 32]. As a result, there is very large and active research effort directed towards the problem. A key aspect of this activity is that it generally focuses on specific system classes such as those described by Volterra kernels [4], neural networks [37], nonlinear ARMAX (NARMAX) [29], and Hammerstein–Wiener [41] structures, to name just some examples. In relation to this, the paper here considers Maximum Likelihood (ML) estimation of the parameters specifying a relatively general class of nonlinear systems that can be represented in state-space form.

Of course, the use of an ML approach (for example, with regard to linear dynamic systems) is common, and it is customary to employ a gradient based search technique such as a damped Gauss–Newton method to actually compute estimates [30, 46]. This requires the computation of a cost Jacobian which typically necessitates implementing one filter derived (in the linear case) from a Kalman filter, for each parameter that is to be estimated. An alternative, recently explored in [17] in the context of bilinear systems is to employ the Expectation Maximisation algorithm [8] for the computation of ML estimates.

Unlike gradient based search, which is applicable to maximisation of any differentiable cost function, EM methods are only applicable to maximisation of likelihood functions. However, a dividend of this specialisation is that while some gradients calculations may be necessary, the gradient of the likelihood function is not required, which will prove to be very important in this paper. In addition to this advantage, EM methods are widely recognised for their numerical stability [28].

Given these recommendations, this paper develops and demonstrates an EM-based approach to nonlinear system identification. This will require the computation of smoothed state estimates that, in the linear case, could be found by standard linear smoothing methods [17]. In the fairly general nonlinear (and possibly non-Gaussian) context considered in this work we propose a "particle based" approach whereby approximations of the required smoothed state estimates are approximated by Monte Carlo based empirical averages [10].

It is important to acknowledge that there is a very significant body of previous work on the problems addressed here. Many approaches using various suboptimal nonlinear filters (such as the extended Kalman filter) to ap-

⋆ Parts of this paper were presented at the 14th IFAC Symposium on System Identification, Newcastle, Australia, March 2006 and at the 17th IFAC World Congress, Seoul, South Korea, July, 2008. Corresponding author: T. B. Schön. Tel. +46-13-281373. Fax +46-13-139282.

Email addresses: schon@isy.liu.se (Thomas B. Schön), Adrian.Wills@newcastle.edu.au (Adrian Wills), Brett.Ninness@newcastle.edu.au (Brett Ninness).

proximate the cost Jacobian have been proposed [5, 22, 27]. Additionally, there has been significant work [3, 12, 40] investigating the employment of particle filters to compute the Jacobian's necessary for a gradient based search approach.

There has also been previous work on various approximate EM-based approaches. Several authors have considered using suboptimal solutions to the associated nonlinear smoothing problem, typically using an extended Kalman smoother [13, 15, 19, 43].

As already mentioned, this paper is considering particle based approaches in order to solve the involved nonlinear smoothing problem. This idea has been partially reported by the authors in two earlier conference publications [45, 47].

An interesting extension, handling the case of missing data is addressed in [20]. Furthermore, in [26], the authors introduce an EM algorithm using a particle smoother, similar to the algorithm we propose here, but tailored to stochastic volatility models. The survey paper [3] is one of the earliest papers to note the possiblility of EM-based methods employing particle smoothing methods.

2 Problem Formulation

This paper considers the problem of identifying the parameters θ for certain members of the following nonlinear state-space model structure

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

Here, $x_t \in \mathbf{R}^{n_x}$ denotes the state variable, with $u_t \in \mathbf{R}^{n_u}$ and $y_t \in \mathbf{R}^{n_y}$ denoting (respectively) observed input and output responses. Furthermore, $\theta \in \mathbf{R}^{n_\theta}$ is a vector of (unknown) parameters that specifies the mappings $f_t(\cdot)$ and $h_t(\cdot)$ which may be nonlinear and time-varying. Finally, v_t and e_t represent mutually independent vector i.i.d. processes described by probability density functions (pdf's) $p_v(\cdot)$ and $p_e(\cdot)$. These are assumed to be of known form (e.g.,Gaussian) but parameterized (e.g.,mean and variance) by values that can be absorbed into θ for estimation if they are unknown.

Due to the random components v_t and e_t, the model (1) can also be represented via the stochastic description

$$x_{t+1} \sim p_\theta(x_{t+1} \mid x_t), \tag{2a}$$
$$y_t \sim p_\theta(y_t \mid x_t), \tag{2b}$$

where $p_\theta(x_{t+1} \mid x_t)$ is the pdf describing the dynamics for given values of x_t, u_t and θ, and $p_\theta(y_t \mid x_t)$ is the pdf describing the measurements. As is common practise, in (2) the same symbol p_θ is used for different pdf's that depend on θ, with the argument to the pdf denoting what is intended. Furthermore, note that we have, for brevity, dispensed with the input signal u_t in the notation (2). However, everything we derive throughout this paper is valid also if an input signal is present.

The formulation (1) and its alternative formulation (2) capture a relatively broad class of nonlinear systems and we consider the members of this class where $p_\theta(x_{t+1} \mid x_t)$ and $p_\theta(y_t \mid x_t)$ can be explicitly expressed and evaluated.

The problem addressed here is the formation of an estimate $\widehat{\theta}$ of the parameter vector θ based on N measurements $U_N = [u_1, \cdots, u_N]$, $Y_N = [y_1, \cdots, y_N]$ of observed system input-output responses. Concerning the notation, sometimes we will make use of $Y_{t:N}$, which is used to denote $[y_t, \cdots, y_N]$. However, as defined above, for brevity we denote $Y_{1:N}$ simply as Y_N. Hence, it is here implicitly assumed that the index starts at 1.

One approach is to employ the general prediction error (PE) framework [30] to deliver $\widehat{\theta}$ according to

$$\widehat{\theta} = \arg\min_{\theta \in \Theta} V(\theta), \tag{3}$$

with cost function $V(\theta)$ of the form

$$V(\theta) = \sum_{t=1}^{N} \ell(\varepsilon_t(\theta)), \qquad \varepsilon_t(\theta) = y_t - \widehat{y}_{t|t-1}(\theta). \tag{4}$$

and with $\Theta \subseteq \mathbf{R}^{n_\theta}$ denoting a compact set of permissible values of the unknown parameter θ. Here,

$$\widehat{y}_{t|t-1}(\theta) = \mathsf{E}_\theta\{y_t \mid Y_{t-1}\} = \int y_t \, p_\theta(y_t \mid Y_{t-1}) \, \mathrm{d}y_t \tag{5}$$

is the mean square optimal one-step ahead predictor of y_t based on the model (1). The function $\ell(\cdot)$ is an arbitrary and user-chosen positive function.

This PE solution has its roots in the Maximum Likelihood (ML) approach, which involves maximising the joint density (likelihood) $p_\theta(Y_N)$ of the observations:

$$\widehat{\theta} = \arg\max_{\theta \in \Theta} p_\theta(y_1, \cdots, y_N). \tag{6}$$

To compute this, Bayes' rule may be used to decompose the joint density according to

$$p_\theta(y_1, \cdots, y_N) = p_\theta(y_1) \prod_{t=2}^{N} p_\theta(y_t | Y_{t-1}). \tag{7}$$

Accordingly, since the logarithm is a monotonic function, the maximisation problem (6) is equivalent to the minimisation problem

$$\widehat{\theta} = \arg\min_{\theta \in \Theta} -L_\theta(Y_N), \tag{8}$$

where $L_\theta(Y_N)$ is the log-likelihood

$$L_\theta(Y_N) \triangleq \log p_\theta(Y_N) = \log p_\theta(y_1) + \sum_{t=2}^{N} \log p_\theta(y_t \mid Y_{t-1}). \tag{9}$$

The PE and ML approaches both enjoy well understood theoretical properties including strong consistency, asymptotic normality, and in some situations asymptotic efficiency. They are therefore both an attractive solution, but there are two important challenges to their implementation.

First, both methods require knowledge of the prediction density $p_\theta(y_t \mid Y_{t-1})$. In the linear and Gaussian case, a Kalman filter can be employed. In the nonlinear case (1) an alternate solution must be found.

Second, the optimisation problems (3) or (8) must be solved. Typically, the costs $V(\theta)$ or $L_\theta(Y_N)$ are differentiable, and this is exploited by employing a gradient based search method to compute the estimate [30]. Unfortunately, these costs will generally possesses multiple local minima that can complicate this approach.

3 Prediction Density Computation

Turning to the first challenge of computing the prediction density, note that by the law of total probability and the Markov nature of (2)

$$p_\theta(y_t \mid Y_{t-1}) = \int p_\theta(y_t \mid x_t) p_\theta(x_t \mid Y_{t-1}) \, \mathrm{d}x_t, \tag{10}$$

where x_t is the state of the underlying dynamic system. Furthermore, using the Markov property of (2) and Bayes' rule we obtain

$$p_\theta(x_t \mid Y_t) = \frac{p_\theta(y_t \mid x_t) p_\theta(x_t \mid Y_{t-1})}{p_\theta(y_t \mid Y_{t-1})}. \tag{11}$$

Finally, by the law of total probability and the Markov nature of (2)

$$p_\theta(x_{t+1} \mid Y_t) = \int p_\theta(x_{t+1} \mid x_t) \, p_\theta(x_t \mid Y_t) \, \mathrm{d}x_t, \tag{12}$$

Together, (11), (10) are known as the "measurement update" and (12) the "time update" equations, which provide a recursive formulation of the required prediction density $p_\theta(y_t \mid Y_{t-1})$ as well as the predicted and filtered state densities $p_\theta(x_t \mid Y_{t-1})$, $p_\theta(x_t \mid Y_t)$.

In the linear and Gaussian case, the associated integrals have closed form solutions which lead to the Kalman filter [25]. In general though, they do not. Therefore, while in principle (10)-(12) provide a solution to the computation of $V(\theta)$ or $L_\theta(Y_N)$, there is a remaining obstacle of numerically evaluating the required n_x-dimensional integrals.

In what follows, the recently popular methods of sequential importance resampling (SIR, or particle filtering) will be employed to address this problem.

However, there is a remaining difficulty which is related to the second challenge mentioned at the end of section 2. Namely, if gradient-based search is to be employed to compute the estimate $\widehat{\theta}$, then not only is $p_\theta(y_t \mid Y_{t-1})$ required, but also its derivative

$$\frac{\partial}{\partial \theta} p_\theta(y_t \mid Y_{t-1}). \tag{13}$$

Unfortunately the SIR technique does not lend itself to the simple computation of this derivative. One approach to deal with this is to simply numerically evaluate the necessary derivative based on differencing. Another is to employ a search method that does not require gradient information. Here, there exist several possibilities, such as Nelder–Mead simplex methods or annealing approaches [44, 48].

This paper explores a further possibility which is known as the Expectation Maximisation (EM) algorithm, and is directed at computing an ML estimate. Instead of using the smoothness of L_θ, it is capable of employing an alternative feature. Namely, the fact that L_θ is the logarithm of a probability density $p_\theta(Y_N)$, which has unit area for all values of θ. How the EM algorithm is capable of utilising this simple fact to deliver an alternate search procedure is now profiled.

4 The Expectation Maximisation Algorithm

Like gradient based search, the EM algorithm is an iterative procedure that at the k'th step seeks a value θ_k such that the likelihood is increased in that $L_{\theta_k}(Y_N) > L_{\theta_{k-1}}(Y_N)$. Again like gradient based search, an approximate model of $L_\theta(Y_N)$ is employed to achieve this. However, unlike gradient based search, the model is capable of guaranteeing increases in $L_\theta(Y_N)$.

The essence of the EM algorithm [8, 33] is the postulation of a "missing" data set $X_N = \{x_1, \cdots, x_N\}$. In this paper, it will be taken as the state sequence in the model structure (1), but other choices are possible, and it can be considered a design variable. The key idea is then to consider the joint likelihood function

$$L_\theta(X_N, Y_N) = \log p_\theta(X_N, Y_N), \tag{14}$$

with respect to both the observed data Y_N and the missing data X_N. Underlying this strategy is an assumption that maximising the "complete" log likelihood $L_\theta(X_N, Y_N)$ is easier than maximising the incomplete one $L_\theta(Y_N)$.

As a concrete example, if the model structure (1) was linear and time-invariant, then knowledge of the state x_t would allow system matrices A, B, C, D to be estimated by simple linear regression. See [16] for more detail, and [34] for further examples.

The EM algorithm then copes with X_N being unavailable by forming an approximation $Q(\theta, \theta_k)$ of $L_\theta(X_N, Y_N)$. The approximation used, is the minimum variance estimate of $L_\theta(X_N, Y_N)$ given the observed available data Y_N, and an assumption θ_k of the true parameter value. This minimum variance estimate is given by the conditional mean [2]

$$Q(\theta, \theta_k) \triangleq \mathbf{E}_{\theta_k}\{L_\theta(X_N, Y_N) \mid Y_N\} \tag{15a}$$

$$= \int L_\theta(X_N, Y_N) p_{\theta_k}(X_N|Y_N)\, \mathrm{d}X_N. \tag{15b}$$

The utility of this approach depends on the relationship between $L_\theta(Y_N)$ and the approximation $Q(\theta, \theta_k)$ of $L_\theta(X_N, Y_N)$. This may be examined by using the definition of conditional probability to write

$$\log p_\theta(X_N, Y_N) = \log p_\theta(X_N \mid Y_N) + \log p_\theta(Y_N). \tag{16}$$

Taking the conditional mean $\mathbf{E}_{\theta_k}\{\cdot \mid Y_N\}$ of both sides then yields

$$Q(\theta, \theta_k) = L_\theta(Y_N) + \int \log p_\theta(X_N|Y_N) p_{\theta_k}(X_N|Y_N)\, \mathrm{d}X_N. \tag{17}$$

Therefore

$$L_\theta(Y_N) - L_{\theta_k}(Y_N) = Q(\theta, \theta_k) - Q(\theta_k, \theta_k) + \int \log \frac{p_{\theta_k}(X_N|Y_N)}{p_\theta(X_N|Y_N)} p_{\theta_k}(X_N|Y_N)\, \mathrm{d}X_N. \tag{18}$$

The rightmost integral in (18) is the Kullback-Leibler divergence metric which is non-negative. This follows directly upon noting that since for $x \geq 0$, $-\log x \geq 1 - x$

$$-\int \log \frac{p_\theta(X_N|Y_N)}{p_{\theta_k}(X_N|Y_N)} p_{\theta_k}(X_N|Y_N)\, \mathrm{d}X_N \geq \int \left(1 - \frac{p_\theta(X_N|Y_N)}{p_{\theta_k}(X_N|Y_N)}\right) p_{\theta_k}(X_N|Y_N)\, \mathrm{d}X_N = 0, \tag{19}$$

where the equality to zero is due to the fact that $p_\theta(X_N \mid Y_N)$ is of unit area for any value of θ. As a consequence of this simple fact

$$L_\theta(Y_N) - L_{\theta_k}(Y_N) \geq Q(\theta, \theta_k) - Q(\theta_k, \theta_k). \tag{20}$$

This delivers the key to the EM algorithm. Namely, choosing θ so that $Q(\theta, \theta_k) > Q(\theta_k, \theta_k)$ implies that the log likelihood is also increased in that $L_\theta(Y_N) > L_{\theta_k}(Y_N)$. The EM algorithm exploits this to deliver a sequence of values θ_k, $k = 1, 2, \cdots$ designed to be increasingly good approximations of the ML estimate (6) via the following strategy.

Algorithm 1 ***(EM Algorithm)***

(1) Set $k = 0$ and initialise θ_k such that $L_{\theta_k}(Y_N)$ is finite;

(2) ***(Expectation (E) Step):***

$$\textit{Calculate:} \qquad Q(\theta, \theta_k); \tag{21}$$

(3) ***(Maximisation (M) Step):***

$$\textit{Compute:} \qquad \theta_{k+1} = \arg\max_{\theta \in \Theta} Q(\theta, \theta_k); \tag{22}$$

(4) If not converged, update $k \mapsto k + 1$ and return to step 2.

The termination decision in step 4 is performed using a standard criterion such as the relative increase of $L_\theta(Y_N)$ or the relative increase of $Q(\theta, \theta_k)$ falling below a predefined threshold [9].

The first challenge in implementing the EM algorithm is the computation of $Q(\theta, \theta_k)$ according to the definition (15a). To address this, note that via Bayes' rule and the Markov property associated with the model structure (1)

$$L_\theta(X_N, Y_N) = \log p_\theta(Y_N|X_N) + \log p_\theta(X_N) = \log p_\theta(x_1) + \sum_{t=1}^{N-1} \log p_\theta(x_{t+1}|x_t) + \sum_{t=1}^{N} \log p_\theta(y_t|x_t). \tag{23}$$

When the model structure (1) is linear and the stochastic components v_t and e_t are Gaussian the $\log p_\theta$ terms are either linear or quadratic functions of the state x_t. Taking the conditional expectation (15a) in order to compute $Q(\theta, \theta_k)$ is then simply achieved by invoking a modification of a standard Kalman smoother [16, 24].

In the more general setting of this paper, the situation is more complicated and requires an alternative approach. To develop it, application of the conditional expectation operator $\mathbf{E}_{\theta_k}\{\cdot \mid Y_N\}$ to both sides of (23) yields

$$Q(\theta, \theta_k) = I_1 + I_2 + I_3, \tag{24}$$

where

$$I_1 = \int \log p_\theta(x_1) p_{\theta_k}(x_1|Y_N)\,\mathrm{d}x_1, \tag{25a}$$

$$I_2 = \sum_{t=1}^{N-1} \int\int \log p_\theta(x_{t+1}|x_t) p_{\theta_k}(x_{t+1}, x_t|Y_N)\,\mathrm{d}x_t\,\mathrm{d}x_{t+1}, \tag{25b}$$

$$I_3 = \sum_{t=1}^{N} \int \log p_\theta(y_t|x_t) p_{\theta_k}(x_t|Y_N)\,\mathrm{d}x_t. \tag{25c}$$

Computing $Q(\theta, \theta_k)$ therefore requires knowledge of densities such as $p_{\theta_k}(x_t|Y_N)$ and $p_{\theta_k}(x_{t+1}, x_t|Y_N)$ associated with a nonlinear smoothing problem. Additionally, integrals with respect to these must be evaluated. Outside the linear case, there is no hope of any analytical solution to these challenges. This paper therefore takes the approach of evaluating (25a)-(25c) numerically.

5 Computing State Estimates

The quantities I_1, I_2, I_3 in (25) that determine $Q(\theta, \theta_k)$ depend primarily on evaluating the smoothed density $p_{\theta_k}(x_t \mid Y_N)$ and expectations with respect to it.

To perform these computations, this paper employs sequential importance resampling (SIR) methods. These are often discussed under the informal title of "particle filters", and the main ideas underlying them date back half a century [35, 36]. However, it was not until 1993 that the first working particle filter was discovered by [21]. As will be detailed, this approach first requires dealing with the filtered density $p_\theta(x_t \mid Y_t)$, and hence the discussion will begin by examining this.

5.1 Particle Filtering

The essential idea is to evaluate integrals by a randomised approach that employs the strong law of large numbers (SLLN). For example, if it is possible to build a random number generator that delivers (suitably uncorrelated) realisations $\{x^i\}$ with respect to a given target probability density $\pi(x)$, then by the SLLN, for a given (measurable) function g

$$\frac{1}{M}\sum_{i=1}^{M} g(x^i) \approx \mathsf{E}\,\{g(x)\} = \int g(x)\pi(x)\,\mathrm{d}x, \tag{26}$$

with equality (with probability one) in the limit as $M \to \infty$.

Certainly, for some special cases such as the Gaussian density, random number generator constructions are well known. Denote by $q(x)$ the density for which such a random variable generator is available, and denote by $\tilde{x}^i \sim q(\tilde{x})$ a realisation drawn using this generator.

A realisation $x^j \sim \pi(x)$ that is distributed according to the target density $\pi(x)$ is then achieved by choosing the j'th realisation x^j to be equal to the value $\tilde{x}^i$ with a certain probability $w(\tilde{x}^i)$. More specifically, for $j = 1, \ldots, M$, a realisation x^j is selected as $\tilde{x}^i$ randomly according to

$$\mathrm{P}(x^j = \tilde{x}^i) = \frac{1}{\kappa}\, w(\tilde{x}^i) \tag{27}$$

where

$$w(\tilde{x}^i) = \frac{\pi(\tilde{x}^i)}{q(\tilde{x}^i)}, \qquad \kappa = \sum_{i=1}^{M} w(\tilde{x}^i). \tag{28}$$

This step is known as "resampling", and the random assignment is done in an independent fashion. The assignment rule (27) works, since by the independence, the probability that as a result x^j takes on the value $\tilde{x}^i$ is the probability $q(\tilde{x}^i)$ that $\tilde{x}^i$ was realised, times the probability $w(\tilde{x}^i)$ that x^j is then assigned this value. Hence, with $\tilde{x}^i$ viewed as a continuous variable, rather than one from a discrete set $\{\tilde{x}^1, \cdots, \tilde{x}^M\}$

$$\mathrm{P}(x^j = \tilde{x}^i) \propto q(\tilde{x}^i)\,\frac{\pi(\tilde{x}^i)}{q(\tilde{x}^i)} = \pi(\tilde{x}^i), \tag{29}$$

so that x^j is a realisation from the required density $\pi(x)$.

The challenge in achieving this is clearly the specification of a density $q(x)$ from which it is both feasible to generate realisations $\{\tilde{x}^i\}$, and for which the ratio $w(x)$ in (28) can be computed. To address this, consider the following selections:

$$\pi(x_t) = p_\theta(x_t \mid Y_t), \qquad q(\tilde{x}_t) = p_\theta(\tilde{x}_t \mid x_{t-1}). \tag{30}$$

This choice of proposal density q is feasible since a realisation $\tilde{x}_t^i \sim p_\theta(\tilde{x}_t \mid x_{t-1})$ may be obtained by simply generating a realisation $v_t^i \sim p_v$, and substituting it, a given x_{t-1}, a measured u_t and model-implied θ into f_t in (1a) in order to deliver a realisation $\tilde{x}_t^i$.

Furthermore, if x_{t-1} used in (30) is a realisation distributed as $x_{t-1} \sim p_\theta(x_{t-1} \mid Y_{t-1})$ then the unconditional proposal density q is given by the law of total probability as

$$q(\tilde{x}_t) = \int p_\theta(\tilde{x}_t \mid x_{t-1})\, p_\theta(x_{t-1} \mid Y_{t-1})\,\mathrm{d}x_{t-1} \tag{31}$$

and hence by the time update equation (12)

$$q(\tilde{x}_t) = p_\theta(\tilde{x}_t \mid Y_{t-1}). \tag{32}$$

As a result, the ratio $w = \pi/q$ implied by the choice (30)

can be expressed as

$$w(\tilde{x}_t^i) = \frac{p_\theta(\tilde{x}_t^i \mid Y_t)}{q(\tilde{x}_t^i)} = \frac{p_\theta(\tilde{x}_t^i \mid Y_t)}{p_\theta(\tilde{x}_t^i \mid Y_{t-1})} = \frac{p_\theta(y_t \mid \tilde{x}_t^i)}{p_\theta(y_t \mid Y_{t-1})} \tag{33}$$

where the measurement update equation (11) is used in progressing to the last equality.

According to the model (1), the numerator in this expression is simply the pdf of $g_t(x_t, u_t, e_t, \theta)$ for given $\tilde{x}_t^i, u_t, \theta$ and hence computable. Additionally, the denominator in (33) is independent of $\tilde{x}_t^i$, and hence simply a normalising constant to ensure unit total probability so that

$$w(\tilde{x}_t^i) = \frac{1}{\kappa} p_\theta(y_t \mid \tilde{x}_t^i), \qquad \kappa = \sum_{i=1}^{M} p_\theta(y_t \mid \tilde{x}_t^i). \tag{34}$$

This analysis suggests a recursive technique of taking realisations $x_{t-1}^i \sim p_\theta(x_{t-1} \mid Y_{t-1})$, using them to generate candidate $\tilde{x}_t^i$ via the proposal (30), and then resampling them using the density (34) to deliver realisations $x_t^i \sim p_\theta(x_t \mid Y_t)$. Such an approach is known as sequential importance resampling (SIR) or, more informally, the realisations $\{x_t^j\}$, $\{\tilde{x}_t^i\}$ are known as particles, and the method is known as particle filtering.

Algorithm 2 ***Basic Particle Filter***

(1) Initialize particles, $\{x_0^i\}_{i=1}^M \sim p_\theta(x_0)$ *and set* $t = 1$*;*
(2) Predict the particles by drawing M *i.i.d. samples according to*

$$\tilde{x}_t^i \sim p_\theta(\tilde{x}_t | x_{t-1}^i), \qquad i = 1, \ldots, M. \tag{35}$$

(3) Compute the importance weights $\{w_t^i\}_{i=1}^M$*,*

$$w_t^i \triangleq w(\tilde{x}_t^i) = \frac{p_\theta(y_t|\tilde{x}_t^i)}{\sum_{j=1}^M p_\theta(y_t|\tilde{x}_t^j)}, \qquad i = 1, \ldots, M. \tag{36}$$

(4) For each $j = 1, \ldots, M$ *draw a new particle* x_t^j *with replacement (resample) according to,*

$$P(x_t^j = \tilde{x}_t^i) = w_t^i, \qquad i = 1, \ldots, M. \tag{37}$$

(5) If $t < N$ *increment* $t \mapsto t + 1$ *and return to step* 2*, otherwise terminate.*

It is important to note that a key feature of the resampling step (37) is that it takes an independent sequence $\{\tilde{x}_t^i\}$ and delivers a dependent one $\{x_t^i\}$. Unfortunately, this will degrade the accuracy of approximations such as (26), since by the fundamental theory underpinning the SLLN, the rate of convergence of the sum to the integral decreases as the correlation in $\{x_t^i\}$ increases [38]. To address this, note that the proposal values $\{\tilde{x}_t^i\}$ are by construction independent, but distributed as $\tilde{x}_t^i \sim p_\theta(\tilde{x}_t \mid Y_{t-1})$. Using them, and again appealing to the law of large numbers

$$\frac{1}{M}\sum_{i=1}^{M} g(\tilde{x}_t^i) w(\tilde{x}_t^i) \approx \int g(\tilde{x}_t) w(\tilde{x}_t) p_\theta(\tilde{x}_t \mid Y_{t-1})\, \mathrm{d}\tilde{x}_t \tag{38a}$$

$$= \int g(\tilde{x}_t) \frac{p_\theta(\tilde{x}_t \mid Y_t)}{p_\theta(\tilde{x}_t \mid Y_{t-1})} p_\theta(\tilde{x}_t \mid Y_{t-1})\, \mathrm{d}\tilde{x}_t \tag{38b}$$

$$= \int g(\tilde{x}_t) p_\theta(\tilde{x}_t \mid Y_t)\, \mathrm{d}\tilde{x}_t = \mathbf{E}_\theta\{g(\tilde{x}_t) \mid Y_t\} \tag{38c}$$

where the transition from (38a) to (38b) follows by (33). Note that the expectation in (38c) is identical to that in (26) with $\pi(x_t) = p_\theta(x_t \mid Y_t)$. However, since the sum in (38a) involves independent $\{\tilde{x}_t^i\}$ rather than the dependent $\{x_t^i\}$ used in (26), it will generally be a more accurate approximation to the expectation.

As a result it is preferable to use the left hand side of (38a) rather than the right hand side of (26). The former, due to use of the "weights" $\{w(\tilde{x}_t^i)\}$ is an example of what is known as "importance sampling" [42]. This explains the middle term in the SIR name given to Algorithm 2.

Of course, this suggests that the resampling step (37) is not essential, and one could simplify Algorithm 2 by removing it and simply propagating the weights $\{w_t^i\}$ for a set of particles $\{x_t^i\}$ whose positions are fixed. Unfortunately this extreme does not work over time since the resampling is critical to being able to track movements in the target density $p_\theta(x_t \mid Y_t)$.

Recognising that while resampling is necessary, it need not be done at each time step t, and recognising the possibility for alternatives to the choice (32) for the proposal density have lead to a range of different particle filtering methods [10]. All deliver values $\{w_t^i\}$, $\{\tilde{x}_t^i\}$, $\{x_t^i\}$ such that arbitrary integrals with respect to a target density $p_\theta(x_t \mid Y_t)$ can be approximately computed via sums such as (26) and (38a).

A mathematical abstraction, which is a useful way of encapsulating this deliverable, is the discrete Dirac delta approximation of $p_\theta(x_t \mid Y_t)$ given by

$$p_\theta(x_t \mid Y_t) \approx \widehat{p}_\theta(x_t \mid Y_t) = \sum_{i=1}^{M} w_t^i \delta(x_t - \tilde{x}_t^i). \tag{39}$$

Underlying this abstraction is the understanding that substituting $\widehat{p}_\theta$ for p_θ delivers finite sum approximations to integrals involving p_θ.

5.2 Particle Smoother

The stochastic sampling approach for computing expectations with respect to the filtered density $p_\theta(x_t \mid Y_t)$

can be extended to accommodate the smoothed density $p_\theta(x_t \mid Y_N)$. The same abstraction just introduced of

$$p_\theta(x_t \mid Y_N) \approx \widehat{p}_\theta(x_t \mid Y_N) = \sum_{i=1}^{M} w_{t|N}^i \delta(x_t - \tilde{x}_t^i) \quad (40)$$

will be used to encapsulate the resulting importance sampling approximations. To achieve this, note that using the definition of conditional probability several times

$$p_\theta(x_t \mid x_{t+1}, Y_N) = p_\theta(x_t \mid x_{t+1}, Y_t, Y_{t+1:N}), \quad (41a)$$
$$= \frac{p_\theta(x_t, x_{t+1}, Y_t, Y_{t+1:N})}{p_\theta(x_{t+1}, Y_t, Y_{t+1:N})} \quad (41b)$$
$$= \frac{p_\theta(Y_{t+1:N} \mid x_t, x_{t+1}, Y_t) p_\theta(x_t, x_{t+1}, Y_t)}{p_\theta(x_{t+1}, Y_t, Y_{t+1:N})} \quad (41c)$$
$$= \frac{p_\theta(Y_{t+1:N} \mid x_t, x_{t+1}, Y_t) p_\theta(x_t \mid x_{t+1}, Y_t) p_\theta(x_{t+1}, Y_t)}{p_\theta(x_{t+1}, Y_t, Y_{t+1:N})} \quad (41d)$$
$$= \frac{p_\theta(Y_{t+1:N} \mid x_t, x_{t+1}, Y_t) p_\theta(x_t \mid x_{t+1}, Y_t)}{p_\theta(Y_{t+1:N} \mid x_{t+1}, Y_t)} \quad (41e)$$
$$= p_\theta(x_t \mid x_{t+1}, Y_t), \quad (41f)$$

where the last equality follows from the fact that given x_{t+1}, by the Markov property of the model (1) there is no further information about $Y_{t+1:N}$ available in x_t and hence $p_\theta(Y_{t+1:N} \mid x_t, x_{t+1}, Y_t) = p_\theta(Y_{t+1:N} \mid x_{t+1}, Y_t)$.

Consequently, via the law of total probability and Bayes' rule

$$p_\theta(x_t \mid Y_N) = \int p_\theta(x_t \mid x_{t+1}, Y_t)\, p_\theta(x_{t+1} \mid Y_N)\, \mathrm{d}x_{t+1} \quad (42a)$$
$$= \int \frac{p_\theta(x_{t+1} \mid x_t) p_\theta(x_t \mid Y_t)}{p_\theta(x_{t+1} \mid Y_t)} p_\theta(x_{t+1} \mid Y_N)\, \mathrm{d}x_{t+1} \quad (42b)$$
$$= p_\theta(x_t \mid Y_t) \int \frac{p_\theta(x_{t+1} \mid x_t) p_\theta(x_{t+1} \mid Y_N)}{p_\theta(x_{t+1} \mid Y_t)}\, \mathrm{d}x_{t+1}. \quad (42c)$$

This expresses the smoothing density $p_\theta(x_t \mid Y_N)$ in terms of the filtered density $p_\theta(x_t \mid Y_t)$ times an x_t dependent integral. To compute this integral, note first that again by the law of total probability, the denominator of the integrand can be written as

$$p_\theta(x_{t+1} \mid Y_t) = \int p_\theta(x_{t+1} \mid x_t)\, p_\theta(x_t \mid Y_t)\, \mathrm{d}x_t. \quad (43)$$

As explained in the previous section, the particle filter (39) may be used to compute this via importance sampling according to

$$p_\theta(x_{t+1} \mid Y_t) \approx \sum_{i=1}^{M} w_t^i\, p_\theta(x_{t+1} \mid \tilde{x}_t^i). \quad (44)$$

To complete the integral computation, note that for the particular case of $t = N$, the smoothing density and the filtering density are the same, and hence the weights in (40) may be initialised as $w_{N|N}^i = w_N^i$ and likewise the particles $\tilde{x}_N^i$ are identical. Working backwards in time t then, we assume an importance sampling approximation (40) is available at time $t+1$, and use it and (44) to compute the integral in (42c) as

$$\int \frac{p_\theta(x_{t+1} \mid x_t) p_\theta(x_{t+1} \mid Y_N)}{p_\theta(x_{t+1} \mid Y_t)}\, \mathrm{d}x_{t+1} \approx \sum_{k=1}^{M} \frac{w_{t+1|N}^k\, p_\theta(\tilde{x}_{t+1}^k \mid x_t)}{\sum_{i=1}^{M} w_t^i\, p_\theta(\tilde{x}_{t+1}^k \mid \tilde{x}_t^i)}. \quad (45a)$$

The remaining $p_\theta(x_t \mid Y_t)$ term in (42c) may be represented by the particle filter (39) so that the smoothed density $p_\theta(x_t \mid Y_N)$ is represented by

$$p_\theta(x_t \mid Y_N) \approx \widehat{p}_\theta(x_t \mid Y_N) = \sum_{i=1}^{M} w_{t|N}^i \delta(x_t - \tilde{x}_t^i), \quad (46a)$$
$$w_{t|N}^i = w_t^i \sum_{k=1}^{M} w_{t+1|N}^k \frac{p_\theta(\tilde{x}_{t+1}^k | \tilde{x}_t^i)}{v_t^k}, \quad (46b)$$
$$v_t^k \triangleq \sum_{i=1}^{M} w_t^i p_\theta(\tilde{x}_{t+1}^k | \tilde{x}_t^i). \quad (46c)$$

These developments can be summarised by the following particle smoothing algorithm.

Algorithm 3 ***Basic Particle Smoother***

(1) Run the particle filter (Algorithm 2) and store the predicted particles $\{\tilde{x}_t^i\}_{i=1}^M$ *and their weights* $\{w_t^i\}_{i=1}^M$*, for* $t = 1, \ldots, N$*.*

(2) Initialise the smoothed weights to be the terminal filtered weights $\{w_t^i\}$ *at time* $t = N$*,*

$$w_{N|N}^i = w_N^i, \quad i = 1, \ldots, M. \quad (47)$$

and set $t = N - 1$*.*

(3) Compute the smoothed weights $\{w_{t|N}^i\}_{i=1}^M$ *using the filtered weights* $\{w_t^i\}_{i=1}^M$ *and particles* $\{\tilde{x}_t^i, \tilde{x}_{t+1}^i\}_{i=1}^M$ *via the formulae (46b), (46c).*

(4) Update $t \mapsto t-1$*. If* $t > 0$ *return to step 3, otherwise terminate.*

Like the particle filter Algorithm 2, this particle smoother is not new [11]. Its derivation is presented here so that the reader can fully appreciate the rationale and approximating steps that underly it. This is important since they are key aspects underlying the novel estimation methods derived here.

Note also that there are alternatives to this algorithm for providing stochastic sampling approximations to func-

tions of the smoothed state densities [6, 7, 14, 18, 39]. The new estimation methods developed in this paper are compatible with any method the user chooses to employ, provided it is compatible with the approximation format embodied by (40). The results presented in this paper used the method just presented as Algorithm 3.

6 The E Step: Computing $Q(\theta, \theta_k)$

These importance sampling approaches will now be employed in order to compute approximations to the terms I_1, I_2 and I_3 in (25) that determine $Q(\theta, \theta_k)$ via (24). Beginning with I_1 and I_3, the particle smoother representation (46) achieved by Algorithm 3 directly provides the importance sampling approximations

$$I_1 \approx \widehat{I}_1 \triangleq \sum_{i=1}^{M} w_{1|N}^i \log p_\theta(\tilde{x}_1^i), \tag{48a}$$

$$I_3 \approx \widehat{I}_3 \triangleq \sum_{t=1}^{N} \sum_{i=1}^{M} w_{t|N}^i \log p_\theta(y_t|\tilde{x}_t^i). \tag{48b}$$

A vital point is that when forming these approximations, the weights $\{w_{t|N}^i\}$ are computed by Algorithms 2 and 3 run with respect to the model structure (1), (2) parameterised by θ_k.

Evaluating I_2 given by (25b) is less straightforward, due to it depending on the joint density $p_\theta(x_{t+1}, x_t|Y_N)$. Nevertheless, using the particle filtering representation (39) together with the smoothing representation (46a) leads to the following importance sampling approximation.

Lemma 6.1 *The quantity I_2 defined in* (25b) *may be computed by an importance sampling approximation $\widehat{I}_2$ based on the particle filtering and smoothing representations* (39), (44) *that is given by*

$$I_2 \approx \widehat{I}_2 \triangleq \sum_{t=1}^{N-1} \sum_{i=1}^{M} \sum_{j=1}^{M} w_{t|N}^{ij} \log p_\theta(\tilde{x}_{t+1}^j \mid \tilde{x}_t^i), \tag{49}$$

where the weights $w_{t|N}^{ij}$ are given by

$$w_{t|N}^{ij} = \frac{w_t^i w_{t+1|N}^j p_{\theta_k}(\tilde{x}_{t+1}^j \mid \tilde{x}_t^i)}{\sum_{l=1}^{M} w_t^l p_{\theta_k}(\tilde{x}_{t+1}^j \mid \tilde{x}_t^l)}. \tag{50}$$

PROOF. First, by the definition of conditional probability

$$p_\theta(x_{t+1}, x_t|Y_N) = p_\theta(x_t|x_{t+1}, Y_N) p_\theta(x_{t+1}|Y_N). \tag{51}$$

Furthermore, by (41a)-(41f)

$$p_\theta(x_t|x_{t+1}, Y_N) = p_\theta(x_t|x_{t+1}, Y_t). \tag{52}$$

Substituting (52) into (51) and using Bayes' rule in conjunction with the Markov property of the model (1) delivers

$$p_\theta(x_{t+1},x_t|Y_N) = p_\theta(x_t|x_{t+1}, Y_t) p_\theta(x_{t+1}|Y_N) \tag{53a}$$

$$= \frac{p_\theta(x_{t+1}|x_t)\, p_\theta(x_t|Y_t)}{p_\theta(x_{t+1}|Y_t)} p_\theta(x_{t+1}|Y_N). \tag{53b}$$

Therefore, the particle filter and smoother representations (39), (46a) may be used to deliver an importance sampling approximation to I_2 according to

$$\int\int \log p_\theta(x_{t+1}|x_t) p_{\theta_k}(x_{t+1}, x_t|Y_N)\, \mathrm{d}x_t\, \mathrm{d}x_{t+1} =$$

$$\int \frac{p_{\theta_k}(x_{t+1}|Y_N)}{p_{\theta_k}(x_{t+1}|Y_t)} \left[\int \log p_\theta(x_{t+1}|x_t) p_{\theta_k}(x_{t+1}|x_t) \times \right.$$

$$\left. p_{\theta_k}(x_t|Y_t)\, \mathrm{d}x_t \right] \mathrm{d}x_{t+1} \approx$$

$$\sum_{i=1}^{M} w_t^i \int \frac{p_{\theta_k}(x_{t+1}|Y_N)}{p_{\theta_k}(x_{t+1}|Y_t)} \log p_\theta(x_{t+1}|\tilde{x}_t^i) p_{\theta_k}(x_{t+1}|\tilde{x}_t^i)\, \mathrm{d}x_{t+1}$$

$$\approx \sum_{i=1}^{M} \sum_{j=1}^{M} w_t^i w_{t+1|N}^j \frac{p_{\theta_k}(\tilde{x}_{t+1}^j \mid \tilde{x}_t^i)}{p_{\theta_k}(\tilde{x}_{t+1}^j|Y_t)} \log p_\theta(\tilde{x}_{t+1}^j|\tilde{x}_t^i).$$

Finally, the law of total probability in combination with the particle filter (39) provides an importance sampling approximation to the denominator term given by

$$p_{\theta_k}(\tilde{x}_{t+1}^j|Y_t) = \int p_{\theta_k}(\tilde{x}_{t+1}^j|x_t) p_{\theta_k}(x_t|Y_t)\, \mathrm{d}x_t \tag{54a}$$

$$\approx \sum_{l=1}^{M} w_t^l p_{\theta_k}(\tilde{x}_{t+1}^j \mid \tilde{x}_t^l). \tag{54b}$$

■

Again, all weights and particles in this approximation are computed by Algorithms 2 and 3 run with respect to the model structure (1), (2) parametrised by θ_k.

Using these importance sampling approaches, the function $Q(\theta, \theta_k)$ given by (24), (25) may be approximately computed as $\widehat{Q}_M(\theta, \theta_k)$ defined by

$$\widehat{Q}_M(\theta, \theta_k) = \widehat{I}_1 + \widehat{I}_2 + \widehat{I}_3, \tag{55}$$

where $\widehat{I}_1$, $\widehat{I}_2$ and $\widehat{I}_3$ are given by (48a), (49) and (48b), respectively. Furthermore, the quality of this approximation can be made arbitrarily good as the number M of particles is increased.

7 The M Step: Maximisation of $\widehat{Q}_M(\theta, \theta_k)$

With an approximation $\widehat{Q}_M(\theta, \theta_k)$ of the function $Q(\theta, \theta_k)$ required in the E step (21) of the EM Algorithm 1 available, attention now turns to the M step (22). This requires that the approximation $\widehat{Q}_M(\theta, \theta_k)$ is maximised with respect to θ in order to compute a new iterate θ_{k+1} of the maximum likelihood estimate.

In certain cases, such as when the nonlinearities f_t and h_t in the model structure (1) are linear in the parameter vector θ, it is possible to maximise $\widehat{Q}_M(\theta, \theta_k)$ using closed-form expressions. An example of this will be discussed in Section 10.

In general however, a closed form maximiser will not be available. In these situations, this paper proposes a gradient based search technique. For this purpose, note that via (55), (48) and (49) the gradient of $\widehat{Q}(\theta, \theta_k)$ with respect to θ is simply computable via

$$\frac{\partial}{\partial\theta}\widehat{Q}_M(\theta,\theta_k) = \frac{\partial\hat{I}_1}{\partial\theta} + \frac{\partial\hat{I}_2}{\partial\theta} + \frac{\partial\hat{I}_3}{\partial\theta}, \tag{56a}$$

$$\frac{\partial\hat{I}_1}{\partial\theta} = \sum_{i=1}^{M} w_{1|N}^{i} \frac{\partial \log p_\theta(\tilde{x}_1^i)}{\partial\theta}, \tag{56b}$$

$$\frac{\partial\hat{I}_2}{\partial\theta} = \sum_{t=1}^{N-1}\sum_{i=1}^{M}\sum_{j=1}^{M} w_{t|N}^{ij} \frac{\partial \log p_\theta(\tilde{x}_{t+1}^j|\tilde{x}_t^i)}{\partial\theta}, \tag{56c}$$

$$\frac{\partial\hat{I}_3}{\partial\theta} = \sum_{t=1}^{N}\sum_{i=1}^{M} w_{t|N}^{i} \frac{\partial \log p_\theta(y_t|\tilde{x}_t^i)}{\partial\theta}. \tag{56d}$$

With this gradient available, there are a wide variety of algorithms that can be employed to develop a sequence of iterates $\theta = \beta_0, \beta_1, \cdots$ that terminate at a value β_* which seeks to maximise $\widehat{Q}_M(\theta, \theta_k)$.

A common theme in these approaches is that after initialisation with $\beta_0 = \theta_k$, the iterations are updated according to

$$\beta_{j+1} = \beta_j + \alpha_j p_j, \quad p_j = H_j g_j, \quad g_j = \frac{\partial}{\partial\theta}\widehat{Q}_M(\theta,\theta_k)\Big|_{\theta=\beta_j} \tag{57}$$

Here H_j is a positive definite matrix that is used to deliver a search direction p_j by modifying the gradient direction. The scalar term α_j is a step length that is chosen to ensure that $\widehat{Q}_M(\beta_j + \alpha_j p_j, \theta_k) \geq \widehat{Q}_M(\beta_j, \theta_k)$. The search typically terminates when incremental increases in $\widehat{Q}_M(\beta, \theta_k)$ fall below a user specified tolerance. Commonly this is judged via the gradient itself according to a test such as $|p_j^T g_j| \leq \epsilon$ for some user specified $\epsilon > 0$.

In relation to this, it is important to appreciate that it is in fact not necessary to find a global maximiser of $\widehat{Q}(\theta, \theta_k)$. All that is necessary is to find a value θ_{k+1} for which $Q(\theta_{k+1}, \theta_k) > Q(\theta_k, \theta_k)$ since via (20) this will guarantee that $L(\theta_{k+1}) > L(\theta_k)$. Hence, the resulting iteration θ_{k+1} will be a better approximation than θ_k of the maximum likelihood estimate (8).

8 Final Identification Algorithm

The developments of the previous sections are now summarised in a formal definition of the EM-based algorithm this paper has derived for nonlinear system identification.

Algorithm 4 ***(Particle EM Algorithm)***

(1) Set $k = 0$ and initialise θ_k such that $L_{\theta_k}(Y)$ is finite;

(2) ***(Expectation (E) Step):***

(a) Run Algorithms 2 and 3 in order to obtain the particle filter (39) *and particle smoother* (46a) *representations.*

(b) Use this information together with (48a), (48b) *and* (49) *to*

$$\textit{Calculate:} \qquad \widehat{Q}_M(\theta,\theta_k) = \widehat{I}_1 + \widehat{I}_2 + \widehat{I}_3. \tag{58}$$

(3) ***(Maximisation (M) Step):***

$$\textit{Compute:} \qquad \theta_{k+1} = \arg\max_{\theta\in\Theta} \widehat{Q}_M(\theta,\theta_k) \tag{59}$$

explicitly if possible, otherwise according to (57).

(4) Check the non-termination condition $Q(\theta_{k+1}, \theta_k) - Q(\theta_k, \theta_k) > \epsilon$ for some user chosen $\epsilon > 0$. If satisfied update $k \mapsto k + 1$ and return to step 2, otherwise terminate.

It is worth emphasising a point made earlier, that while the authors have found the simple particle and smoothing Algorithms 2 and 3 to be effective, the user is free to substitute alternatives if desired, provided the results they offer are compatible with the representations (39), (46a).

It is natural to question the computational requirements of this proposed algorithm. Some specific comments relating to this will be made in the example section following. More generally, it is possible to identify the computation of $\widehat{I}_2$ given by (49) and its gradient (56c) as a dominating component of both the E and M steps. As is evident, it requires $O(NM^2)$ floating point operations.

This indicates that the computing load is sensitive to the number M of particles employed. Balancing this, the experience of the authors has been that useful results can be achieved without requiring M to be prohibitively large. The following simulation section will provide an example illustrating this point with $M = 100$, and 1000 iterations of Algorithm 4 requiring approximately one minute of processor time on a standard desktop computing platform.

9 Convergence

It is natural to question the convergence properties of this iterative parameter estimation procedure. These will derive from the general EM algorithm 1 on which it is based, for which the most fundamental convergence property is as follows.

If the EM algorithm terminates at a point θ_{k+1} because it is a stationary point of $Q(\theta, \theta_k)$, then it is also a stationary point of the log likelihood $L(\theta)$. Otherwise, the likelihood is increased in progressing from θ_k to θ_{k+1}.

Lemma 9.1 *Let θ_{k+1} be generated from θ_k by an iteration of the EM Algorithm (21),(22). Then*

$$L(\theta_{k+1}) \geq L(\theta_k) \qquad \forall k = 0, 1, 2, \ldots, \tag{60}$$

Furthermore, equality holds in this expression if and only if both

$$Q(\theta_{k+1}, \theta_k) = Q(\theta_k, \theta_k), \tag{61}$$

and

$$p_{\theta_{k+1}}(X_N \mid Y_N) = p_{\theta_k}(X_N \mid Y_N), \tag{62}$$

hold for almost all (with respect to Lebesgue measure) X_N.

PROOF. See Theorem 5.1 in [16]. ■

An important point is that the proof of this result only depends on $Q(\theta_{k+1}, \theta_k) \geq Q(\theta_k, \theta_k)$ being non-decreasing at each iteration. It does not require that θ_{k+1} be a maximiser of $Q(\theta, \theta_k)$.

This provides an important theoretical underpinning for the EM method foundation of Algorithm 4 developed here. Its application is complicated by the fact that only an approximation $\widehat{Q}_M(\theta, \theta_k)$ of $Q(\theta, \theta_k)$ is available. However, this approximation is arbitrarily accurate for a sufficiently large number M of particles.

Lemma 9.2 *Consider the function $Q(\theta, \theta_k)$ defined by (24)-(25c) and its SIR approximation $\widehat{Q}_M(\theta, \theta_k)$ defined by (48a)-(49) and (55) which is based on M particles. Suppose that*

$$p_\theta(y_t \mid x_t) < \infty, \quad p_\theta(x_{t+1} \mid x_t) < \infty, \tag{63}$$

$$\mathsf{E}\left\{|Q(\theta, \theta_k)|^4 \mid Y_N\right\} < \infty, \tag{64}$$

hold for all $\theta, \theta_k \in \Theta$. Then with probability one

$$\lim_{M \to \infty} \widehat{Q}_M(\theta, \theta_k) = Q(\theta, \theta_k), \quad \forall \theta, \theta_k \in \Theta. \tag{65}$$

PROOF. By application of Corollary 6.1 in [23]. ■

Together, Lemmas 9.1 and 9.2, do not establish convergence of Algorithm 4, and are not meant to imply it.

Indeed, one drawback of the EM algorithm is that except under restrictive assumptions (such as convex likelihood), it is not possible to establish convergence of the iterates $\{\theta_k\}$, even when exact computation of the E-step is possible [34, 49].

The point of Lemma 9.1 is to establish that any algorithmic test that $Q(\theta, \theta_k)$ has not decreased (such as step (4) of Algorithm 4) guarantees a non-decrease of $L(\theta)$. Hence EM is capable of matching the guaranteed non cost-decreasing property of gradient based search.

Of course, this depends on the accuracy with which $Q(\theta, \theta_k)$ can be calculated. The point of Lemma 9.2 is to establish that the particle-based approximant $\widehat{Q}_M(\theta, \theta_k)$ used in this paper is an arbitrarily accurate approximation of $Q(\theta, \theta_k)$. Hence Lemma 9.2 establishes a scientific basis for employing $\widehat{Q}_M(\theta, \theta_k)$.

10 Numerical Illustrations

In this section the utility and performance of the new Algorithm 4 is demonstrated on two simulation examples. The first is a linear time-invariant Gaussian system. This is profiled since an exact solution for the expectation step can be computed using the Kalman smoother [16]. Comparing the results obtained by employing both this, and the particle based approximations used in Algorithm 4 therefore allow the effect of the particle approximation on estimation accuracy to be judged.

The performance of Algorithm 4 on a second example involving a well studied and challenging nonlinear system is then illustrated.

10.1 Linear Gaussian System

The first example to be considered is the following simple linear time series

$$\begin{aligned} x_{t+1} &= ax_t + v_t \\ y_t &= cx_t + e_t \end{aligned} \qquad \begin{bmatrix} v_t \\ e_t \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} q & 0 \\ 0 & r \end{bmatrix} \right) \tag{66a}$$

with the true parameters given by

$$\theta^\star = [a^\star, c^\star, q^\star, r^\star] = [0.9, 0.5, 0.1, 0.01]\,. \tag{66b}$$

The estimation problem is to determine just the $\theta = a$ parameter on the basis of the observations Y_N. Using EM methods it is straightforward to also estimate the

c, q and r parameters as well [16]. However, this example concentrates on a simpler case in order to focus attention on the effect of the particle filter/smoother approximations employed in Algorithm 4.

More specifically, via Algorithm 4, a particle based approximation $\widehat{Q}_M(a, a_k)$ can be expressed as

$$\widehat{Q}_M(a, a_k) = -\gamma(a_k)a^2 + 2\psi(a_k)a + d, \tag{67}$$

where d is a constant term that is independent of a and $\psi(\cdot)$ and $\gamma(\cdot)$ are defined as

$$\psi(a_k) = \sum_{t=1}^{N-1}\sum_{i=1}^{M}\sum_{j=1}^{M} w_{t|N}^{ij}\tilde{x}_{t+1}^{j}\tilde{x}_t^i, \tag{68a}$$

$$\gamma(a_k) = \sum_{t=1}^{N}\sum_{i=1}^{M} w_{t|N}^{i}(\tilde{x}_t^i)^2. \tag{68b}$$

Since $\widehat{Q}_M(a, a_k)$ in (67) is quadratic in a, it is straightforward to solve the M step in closed form as

$$a_{k+1} = \frac{\psi(a_k)}{\gamma(a_k)}. \tag{69}$$

Furthermore, in this linear Gaussian situation $Q(\theta, \theta_k)$ can be computed exactly using a modified Kalman smoother [16]. In this case, the exact $Q(a, a_k)$ is again of the quadratic form (67) after straightforward redefinitions of ψ and γ, so the "exact" M step also has the closed form solution (69).

This "exact EM" solution can then be profiled versus the new particle filter/smoother based EM method (67)-(69) of this paper in order to assess the effect of the approximations implied by the particle approach.

This comparison was made by conducting a Monte Carlo study over 1000 different realisations of data Y_N with $N = 100$. For each realisation, ML estimates $\widehat{a}$ were computed using the exact EM solution provided by [16], and via the approximate EM method of Algorithm 4. The latter was done for two cases of $M = 10$ and $M = 500$ particles. In all cases, the initial value a_0 was set to the true value $a^\star$.

The results are shown in Figure 1. There, for each of the 1000 realisations, a point is plotted with x co-ordinate the likelihood value $L(\widehat{a})$ achieved by 100 iterations of the exact EM method, and y co-ordinate the value achieved by 100 iterations of Algorithm 4.

Clearly, if both approaches produced the same estimate, all the points plotted in this manner should lie on the solid $y = x$ line shown in Figure 1. For the case of $M = 500$ particles, where the points are plotted with a cross '`x`', this is very close to being the case. This illustrates that with sufficient number of particles, the use of the approximation $\widehat{Q}_M$ in Algorithm 4 can have negligible detrimental effect on the final estimate produced.

Also plotted in Figure 1 using an '`o`' symbol, are the results obtained using only $M = 10$ particles. Despite this being what could be considered a very small number of particles, there is still generally reasonable, and often good agreement between the associated approximate and exact estimation results.

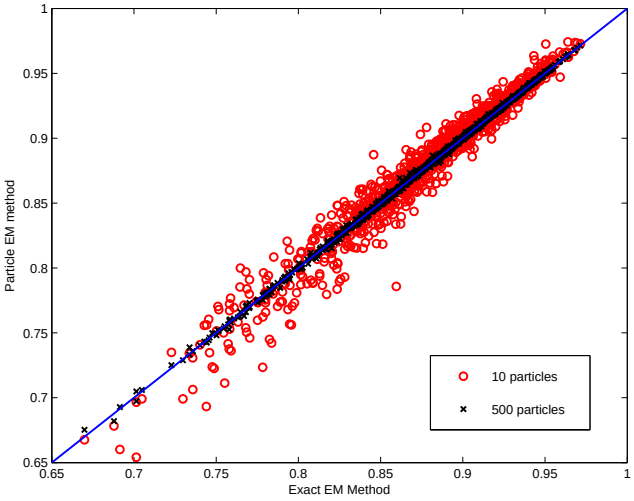

Fig. 1. *Comparison of the likelihood values for the final estimates after* 100 *iterations of the exact EM method and the particle EM method given in Algorithm 4 using both* $M = 10$ *and* $M = 500$ *particles.*

10.2 A Nonlinear and Non-Gaussian System

A more challenging situation is now considered that involves the following nonlinear and time-varying system

$$x_{t+1} = ax_t + b\frac{x_t}{1 + x_t^2} + c\cos(1.2t) + v_t, \tag{70a}$$

$$y_t = dx_t^2 + e_t, \tag{70b}$$

$$\begin{bmatrix} v_t \\ e_t \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} q & 0 \\ 0 & r \end{bmatrix}\right) \tag{70c}$$

where the true parameters in this case are

$$\theta^\star = [a^\star, b^\star, c^\star, d^\star, q^\star, r^\star] = [0.5, 25, 8, 0.05, 0, 0.1]. \tag{71}$$

This has been chosen due to it being acknowledged as a challenging estimation problem in several previous studies in the area [11, 18, 21].

To test the effectiveness of Algorithm 4 in this situation, a Monte Carlo study was again performed using 104 different data realisations Y_N of length $N = 100$. For each

of these cases, an estimate $\widehat{\theta}$ was computed using 1000 iterations of Algorithm 4 with initialisation θ_0 being chosen randomly, but such that each entry of θ_0 lay in an interval equal to 50% of the corresponding entry in the true parameter vector $\theta^\star$. In all cases $M = 100$ particles were used.

Using these choices, each computation of $\widehat{\theta}$ using Algorithm 4 took 58 seconds to complete on a 3 GHz quad-core Xeon running Mac OS 10.5.

The results of this Monte Carlo examination are provided in Table 1, where the rightmost column gives the sample mean of the parameter estimate across the Monte Carlo trials plus/minus the sample standard deviation. Note that 8 of the 104 trials were not included

Table 1
True and estimated parameter values for (70)*; mean value and standard deviations are shown for the estimates based on* 104 *Monte Carlo runs.*

Parameter	**True**	**Estimated**
a	0.5	0.50 ± 0.0019
b	25.0	25.0 ± 0.99
c	8.0	7.99 ± 0.13
d	0.05	0.05 ± 0.0026
q	0	$7.78 \times 10^{-5} \pm 7.6 \times 10^{-5}$
r	0.1	0.106 ± 0.015

in these calculations due to capture in local minima, which was defined according to the relative error test $|(\widehat{\theta}_i - \theta_i^\star)/\theta_i^\star)| > 0.1$ for any i'th component. Considering the random initialisation, this small number of required censoring and the results in Table 1 are considered successful results.

It is instructive to further examine the nature of both this estimation problem and the EM-based solution. For this purpose consider the situation where only the b and q parameters are to be estimated. In this case, the log-likelihood $L_\theta(Y)$ as a function of b with $q = q^\star = 0$ is shown as the solid line in Figure 2. Clearly the log-likelihood exhibits quite erratic behaviour with very many local maxima. These could reasonably be expected to create significant difficulties for iterative search methods, such as gradient based search schemes for maximising $L_\theta(Y)$.

However, in this simplified case, the EM-based method of this paper seems quite robust against capture in these local maxima. For example, the trajectory of the parameter estimates over 100 iterations of Algorithm 4 and over 100 different length $N = 100$ data realisations, and 100 random initialisations for the b parameter, with the q parameter initialised at $q = 0.001$ are shown in Figure 3. Here, $M = 50$ particles were employed, and in all cases, an effective estimate of the true parameter value $b = 25$ was obtained.

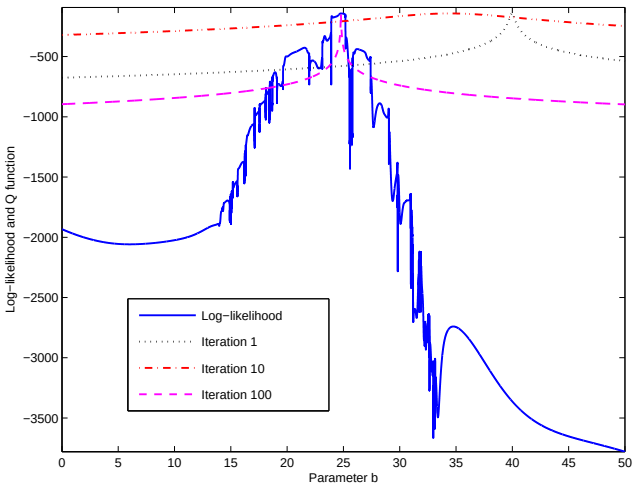

Fig. 2. *The true log-likelihood function is shown as a function of the b parameter. Superimposed onto this plot are three instances of the $Q(\theta, \theta_k)$ function, defined in (15a). Clearly, as the EM algorithm evolves, then locally around the global maximiser, the approximation $Q(\theta, \theta_k)$ resembles the log-likelihood $L_\theta(Y)$ more closely.*

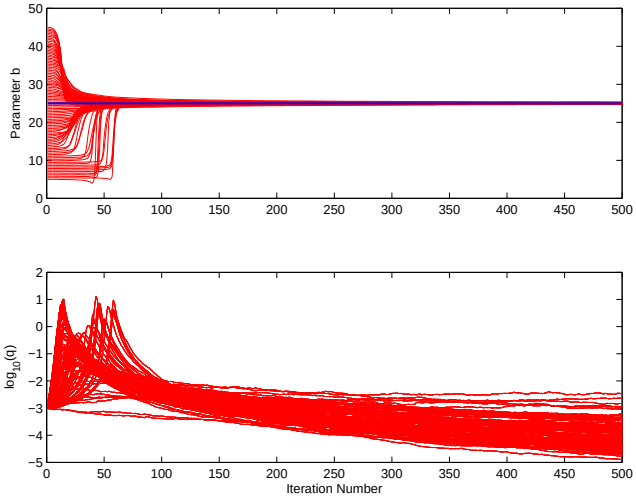

Fig. 3. *Top: parameter b estimates as a function of iteration number (horizontal line indicates the true parameter value at $b = 25$). Bottom: $\log_{10}(q)$ parameter estimates as a function of iteration number.*

The means whereby the EM-based Algorithm 4 achieves this are illustrated by profiling the function $Q(\theta, \theta_k)$ initialised at $[b_0, q_0] = [40, 0.001]$ for $k = 1, 10$ and 100 as the dotted, dash-dotted and dashed lines, respectively. Clearly, in each case the $Q(\theta, \theta_k)$ function is a much more straightforward maximisation problem than that of the log likelihood $L_\theta(Y)$. Furthermore, by virtue of the essential property (20), at each iteration directions of increasing $Q(\theta, \theta_k)$ can be seen to coincide with directions of increasing $L_\theta(Y)$. As a result, difficulties associated with the local maxima of $L_\theta(Y)$ are avoided.

To study this further, the trajectory of EM-based estimates $\theta_k = [b_k, q_k]^T$ for this example are plotted in relation to the two dimensional log-likelihood surface $L_\theta(Y)$ in Figure 4. Clearly, the iterates have taken a path cir-

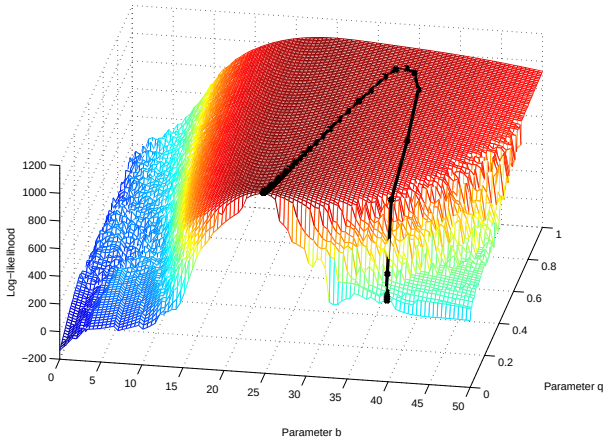

Fig. 4. *The log-likelihood is here plotted as a function of the two parameters b and q. Overlaying this are the parameter estimates $\theta_k = [b_k, q_k]^T$ produced by Algorithm 4.*

cumventing the highly irregular "slice" at $q = 0$ illustrated in Figure 2. As a result, the bulk of them lie in much better behaved regions of the likelihood surface.

This type of behaviour with associated robustness to get captured in local minima is widely recognised and associated with the EM algorithm in the statistics literature [34]. Within this literature, there are broad explanations for this advantage, such as the fact that (20) implies that $Q(\theta, \theta_k)$ forms a global approximation to the log likelihood $L_\theta(Y)$ as opposed to the local approximations that are implicit to gradient based search schemes. However, a detailed understanding of this phenomenon is an important open research question deserving further study.

A further intriguing feature of the EM-algorithm is that while (20) implies that local maxima of $L_\theta(Y)$ may be fixed points of the algorithm, there may be further fixed points. For example, in the situation just studied where the true $q^\star = 0$, if the EM-algorithm is initialised with $q_0 = 0$, then all iterations θ_k will be equal to θ_0, regardless of what the other entries in θ_0 are.

This occurs because with $v_t = 0$ in (1a), the smoothing step delivers state estimates completely consistent with θ_0 (a deterministic simulation arises in the sampling (2a)), and hence the maximisation step then delivers back re-estimates that reflect this, and hence are unchanged. While this is easily avoided by always initialising q_0 as non-zero, a full understanding of this aspect and the question of further fixed points are also worthy of further study.

11 Conclusion

The contribution in this paper is a novel algorithm for identifying the unknown parameters in general stochastic nonlinear state-space models. To formulate the problem a maximum likelihood criterion was employed, mainly due to the general statistical efficiency of such an approach. This problem is then solved using the expectation maximisation algorithm, which in turn required a nonlinear smoothing problem to be solved. This was handled using a particle smoothing algorithm. Finally, the utility and performance of the new algorithm was demonstrated using two simulation examples.

Acknowledgements

This work supported by: the strategic research center MOVIII, funded by the Swedish Foundation for Strategic Research (SSF) and CADICS, a Linneaus Center funded by be Swedish Research Council; and the Australian Reseach Council.

References

[1] *Bode Lecture: Challenges of Nonlinear System Identification*, December 2003.

[2] B. D. O. Anderson and J. B. Moore. *Optimal Filtering*. Prentice-Hall, New Jersey, 1979.

[3] 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.

[4] J.S. Bendat. *Nonlinear System Analysis and Identification from Random Data*. Wiley Interscience, 1990.

[5] T. Bohlin. *Practical Grey-box Process Identification: Theory and Applications*. Springer, 2006.

[6] Y. Bresler. Two-filter formulae for discrete-time non-linear Bayesian smoothing. *International Journal of Control*, 43(2):629–641, 1986.

[7] M. Briers, A. Doucet, and S. R. Maskell. Smoothing algorithms for state-space models. *Annals of the Institute of Statistical Mathematics (to appear)*, 2009.

[8] 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.

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

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

[11] A. Doucet, S. J. Godsill, and C. Andrieu. On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10(3):197–208, 2000.

[12] A. Doucet and V. B. Tadić. Parameter estimation in general state-space models using particle methods. *Annals of the Institute of Statistical Mathematics*, 55:409–422, 2003.

[13] S. Duncan and M. Gyöngy. Using the EM algorithm to estimate the disease parameters for smallpox in 17th century London. In *Proceedings of the IEEE international conference on control applications*, pages 3312–3317, Munich, Germany, October 2006.

[14] P. Fearnhead, D. Wyncoll, and J. Tawn. A sequential smoothing algorithm with linear computational cost. Technical report, Department of Mathematics and Statistics, Lancaster University, Lancaster, UK, May 2008.

[15] Z. Ghaharamani and S. T. Roweis. Learning nonlinear dynamical systems using an EM algorithm. In *Advances in Neural Information Processing Systems*, volume 11, pages 599–605. MIT Press, 1999.

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

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

[18] 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.

[19] G. C. Goodwin and J. C. Agüero. Approximate EM algorithms for parameter and state estimation in nonlinear stochastic models. In *Proceedings of the 44th IEEE conference on decision and control (CDC) and the European Control Conference (ECC)*, pages 368–373, Seville, Spain, December 2005.

[20] R. B. Gopaluni. A particle filter approach to identification of nonlinear processes under missing observations. *The Canadian Journal of Chemical Engineering*, 86(6):1081–1092, December 2008.

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

[22] S. Graebe. *Theory and Implementation of Gray Box Identification*. PhD thesis, Royal Institute of Technology, Stockholm, Sweden, June 1990.

[23] X.-L. Hu, T. B. Schön, and L. Ljung. A basic convergence result for particle filtering. *IEEE Transactions on Signal Processing*, 56(4):1337–1348, April 2008.

[24] A. H. Jazwinski. *Stochastic processes and filtering theory*. Mathematics in science and engineering. Academic Press, New York, USA, 1970.

[25] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME, Journal of Basic Engineering*, 82:35–45, 1960.

[26] J. Kim and D. S. Stoffer. Fitting stochastic volatility models in the presence of irregular sampling via particle methods and the em algorithm. *Journal of time series analysis*, 29(5):811–833, September 2008.

[27] N. R. Kristensen, H. Madsen, and S. B. Jorgensen. Parameter estimation in stochastic grey-box models. *Automatica*, 40(2):225–237, February 2004.

[28] Kenneth Lange. A gradient algorithm locally equivalent to the em algorithm. *Journal of the Royal Statistical Society*, 57(2):425–437, 1995.

[29] I.J. Leontaritis and S.A. Billings. Input-output parametric models for non-linear systems. part ii: stochastic non-linear systems. *International Journal of Control*, 41(2):329–344, 1985.

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

[31] L. Ljung and A. Vicino, editors. Special Issue 'System Identification: Linear vs Nonlinear'. *IEEE Transactions on Automatic Control*, 2005.

[32] Lennart Ljung. Perspectives on system identification. In *Plenary Talk at the 17th IFAC World Congress, Seoul, Korea, July 6–11 2008*.

[33] G. McLachlan and T. Krishnan. *The EM Algorithm and Extensions*. Whiley Series in Probability and Statistics. John Wiley & Sons, New York, USA, 2 edition, 2008.

[34] G. McLachlan and T. Krishnan. *The EM Algorithm and Extensions (2nd Edition)*. John Wiley and Sons, 2008.

[35] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equations of state calculations by fast computing machine. *Journal of Chemical Physics*, 21(6):1087–1092, 1953.

[36] N. Metropolis and S. Ulam. The Monte Carlo method. *Journal of the American Statistical Association*, 44(247):335–341, 1949.

[37] K. Narendra and K. Parthasarathy. Identification and control of dynamical systems using neural networks. *IEEE Transactions on Neural Networks*, 1:4–27, 1990.

[38] B. Ninness. Strong laws of large numbers under weak assumptions with application. *IEEE Trans. Automatic Control*, 45(11):2117–2122, 2000.

[39] G. Pillonetto and B. M. Bell. Optimal smoothing of non-linear dynamic systems via Monte Carlo Markov Chains. *Automatica*, 44(7):1676–1685, July 2008.

[40] 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.

[41] S. Rangan, G. Wolodkin, and K. Poolla. New results for Hammerstein system identification. In *Proceedings of the 34th IEEE Conference on Decision and Control*, pages 697–702, New Orleans, USA, December 1995.

[42] B.D. Ripley. *Stochastic Simulation*. Wiley, 1987.

[43] S. T. Roweis and Z. Ghaharamani. *Kalman filtering and neural networks*, chapter 6. Learning nonlinear dynamical systems using the expectation maximization algorithm, Haykin, S. (ed), pages 175–216. John Wiley & Sons, 2001.

[44] P. Salamon, P. Sibani, and R. Frost. *Facts, conjectures, and Improvements fir Simulated Annealing*. SIAM, Philadelphia, 2002.

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

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

[47] A. G. Wills, T. B. Schön, and B. Ninness. Parameter estimation for discrete-time nonlinear systems using em. In *Proceedings of the 17th IFAC World Congress*, Seoul, South Korea, July 2008.

[48] M. H. Wright. Direct search methods: once scorned, now respectable. In *Numerical analysis 1995 (Dundee, 1995)*, pages 191–208. Longman, Harlow, 1996.

[49] C. Wu. the convergence properties of the EM algorithm, 1983.