Estimation of General Nonlinear State-Space Systems.

Brett Ninness, Adrian Wills and Thomas B. Schön

Abstract— **This paper presents a novel approach to the estimation of a general class of dynamic nonlinear system models. The main contribution is the use of a tool from mathematical statistics, known as Fishers' identity, to establish how so-called "particle smoothing" methods may be employed to compute gradients of maximum-likelihood and associated prediction error cost criteria.**

I. Introduction

The field of *linear* dynamic system identification is by now quite mature. A comprehensive, unified and effective framework has been developed for understanding the various approaches which have proven effective [1–3]. This involves noting the distinctions between model structure, estimation criterion, and employed algorithm. Central to this is the straightforward computability for linear systems of a mean square optimal one-step ahead predictor via a Kalman or Wiener filter.

The estimation of systems governed by *nonlinear* dynamics is a much more open research topic, which continues to attract significant attention [1, 4]. It is a challenging problem, for which it has proven to be effective to restrict attention to focused nonlinear structure subsets. Examples include work targeted at Hammerstein–Wiener [5], Volterra kernel [6], nonlinear ARMAX (NARMAX) [7] and neural network structures [8].

Underlying this approach is the fact that for general nonlinear model structures, the issue of computing one-step ahead predictors is much less straightforward than in the linear case. Targeting specific nonlinear sub-structures provides one means for managing this difficulty.

Of key relevance to this issue has been the recent development of sequential importance resampling (SIR) techniques, which are more colloquially known as "particle filters" [9–12]. These are techniques for obtaining approximate solutions to the time and measurement update equations for predictors for general nonlinear models.

This paper employs these SIR methods in order to address the topic of parameter estimation for a rather general class of nonlinear systems. The SIR techniques are used as an approach for computing the prediction error criterion cost for a given parameter value. Finding a parameter estimate then requires computing the minimum of this cost. In the linear system case, this is routinely solved via a gradient-based search approach [1, 13].

Unfortunately extending this technique to the nonlinear case using SIR methods leads to a fundamental difficulty. The gradient of the predictor with respect to the parameters is not readily obtained via SIR algorithms. To address this difficulty, in previous work the expectation–maximisation (EM) algorithm has been examined as one approach to avoiding the problem of computing predictor gradients [14, 15].

This paper proposes an alternate approach by noting that a result from statistics, known as Fisher's identity, provides a means where the required gradients may be computed using a particle *smoother* (as opposed to a particle filter).

II. Problem Formulation

This paper considers the following general nonlinear state-space model structure

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

$$y_t = h_t(x_t, u_t, \theta) + e_t(\theta). \tag{1b}$$

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 of arbitrary form, and hence nonlinear. Note that via the subscript t, the functions f_t and h_t may also be time varying.

Finally, v_t and e_t represent mutually independent vector i.i.d. processes described by the probability density functions (pdf's) $p_v(\cdot)$ and $p_e(\cdot)$. These are assumed to be of known form, but parameterized (e.g. mean and variance) by values that can be absorbed into θ for estimation if necessary.

The problem studied here is the formation of an estimate $\widehat{\theta}$ of the parameter vector θ based on N measurements

$$U_N \triangleq [u_1, \cdots, u_N], \quad Y_N \triangleq [y_1, \cdots, y_N], \tag{2}$$

of observed system input-output responses. In addressing this problem, it will prove important to note that the model (1) permits an alternative probabilistic description according to

$$x_{t+1} \sim p_\theta(x_{t+1} \mid x_t) = p_v(x_{t+1} - f_t(x_t, u_t, \theta)), \tag{3a}$$

$$y_t \sim p_\theta(y_t \mid x_t) = p_e(y_t - h_t(x_t, u_t, \theta)). \tag{3b}$$

Note that here we have adopted a common practice of streamlining notation by labeling different pdf's with the same identifier p_θ, with the understanding that the arguments determine what is intended.

A. Wills and Brett Ninness are with the School of Electrical Engineering and Computer Science, University of Newcastle, Callaghan, NSW 2308, Australia {Adrian.Wills, Brett.Ninness}@newcastle.edu.au

T. B. Schön is with the Division of Automatic Control, Linköping University, SE-581 83 Linköping, Sweden, E-mail: schon@isy.liu.se

This work was partly supported by the Australian Research Council, and partly 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.

III. Maximum Likelihood and Prediction Error Estimation

To address this estimation problem, this paper considers the maximum-likelihood (ML) approach [1] that delivers $\widehat{\theta}$ as the value maximising the joint density (likelihood) $p_\theta(Y_N)$ of the observations:

$$\widehat{\theta} = \arg\max_{\theta\in\Theta} p_\theta(y_1, \cdots, y_N), \tag{4}$$

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

In this paper, the input is assumed observed in a noise free manner, and is not a random variable. Hence it does not appear as an argument to the above density p_θ, although the form of that density will depend on it.

To compute (4), Bayes' rule may be used to decompose the joint density according to

$$p_\theta(y_1, \cdots, y_N) = p_\theta(y_1) \prod_{t=2}^{N} p_\theta(y_t|Y_{t-1}). \tag{5}$$

Accordingly, since the logarithm is a monotonic function, the maximisation problem (4) is equivalent to the minimisation problem

$$\widehat{\theta} = \arg\min_{\theta\in\Theta} -L_\theta(Y_N), \tag{6}$$

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

$$L_\theta(Y_N) \triangleq \log p_\theta(Y_N) = \log p_\theta(y_1) + \sum_{t=2}^{N} \log p_\theta(y_t \mid Y_{t-1}). \tag{7}$$

The mean square optimal one-step ahead predictor of y_t based on the model (1) is given by the conditional mean

$$\widehat{y}_{t|t-1}(\theta) = \mathbf{E}_\theta\{y_t \mid Y_{t-1}\} = \int y_t \, p_\theta(y_t \mid Y_{t-1}) \, \mathrm{d}y_t \tag{8}$$

and is related to y_t according to

$$y_t = \widehat{y}_{t|t-1}(\theta) + \varepsilon_t, \tag{9}$$

where $\{\varepsilon_t\}$ is a zero mean uncorrelated process [16]. Accordingly (7) may be expressed as

$$L_\theta(Y_N) = \log p_\theta(y_1) + \sum_{t=2}^{N} \log p_\varepsilon(\varepsilon_t(\theta)) \tag{10}$$

where $p_\varepsilon(\cdot)$ is the density of ε_t, which we assume here to be strongly stationary.

As N grows, the second term will dominate this expression, which will allow the ML estimate to be seen as a case of the more general prediction error (PE) framework:

$$\widehat{\theta} = \arg\min_{\theta\in\Theta} V(\theta), \quad V(\theta) = \sum_{t=1}^{N} \ell(\varepsilon_t(\theta)), \tag{11}$$

where $\ell(\cdot)$ is an arbitrary and user-chosen positive function. The choice $\ell(\cdot) = -\log p_\epsilon(\cdot)$ then establishes the ML estimate as an instance of the PE one. Possibly the most common choice for $\ell(\cdot)$ is the "least squares" one

$$\ell(x) = x^T x = \|x\|^2 \tag{12}$$

which is also the one studied here. This situation (11)–(12) is also an example of the ML approach, when ε_t is Gaussian. Equally then, a PE method may be encompassed by an ML approach with appropriate choice for the densities in the model (1).

Both methods depend on 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 is required. This paper examines an approach based on sequential importance resampling.

IV. Sequential Importance Resampling (SIR)

By the law of total probability and the Markov nature of the nonlinear model (1), the prediction density $p_\theta(y_t \mid Y_{t-1})$ can be expressed as

$$p_\theta(y_t \mid Y_{t-1}) = \int p_\theta(y_t \mid x_t) p_\theta(x_t \mid Y_{t-1}) \, \mathrm{d}x_t. \tag{13}$$

Additionally, by the definition of conditional probability

$$\begin{aligned} p_\theta(x_t \mid Y_t) &= \frac{p_\theta(x_t, Y_t)}{p_\theta(Y_t)} \\ &= \frac{p_\theta(y_t, Y_{t-1}, x_t)}{p_\theta(Y_{t-1}, x_t)} \cdot \frac{p_\theta(Y_{t-1}, x_t)}{p_\theta(Y_{t-1})} \cdot \frac{p_\theta(Y_{t-1})}{p_\theta(Y_t)} \\ &= \frac{p_\theta(y_t \mid x_t) \, p_\theta(x_t \mid Y_{t-1})}{p_\theta(y_t \mid Y_{t-1})}. \end{aligned} \tag{14}$$

Furthermore, again by the law of total probability and the Markov nature of (1)

$$p_\theta(x_{t+1} \mid Y_t) = \int p_\theta(x_{t+1} \mid x_t) \, p_\theta(x_t \mid Y_t) \, \mathrm{d}x_t. \tag{15}$$

Together (14) and (15) are the general so-called 'measurement update' and 'time update' equations solving the general nonlinear filtering problem, with (15) being an instance of the Chapman–Kolmogorov equation.

Unfortunately, there are very few cases, such as the linear Gaussian, and the discrete time, discrete state Hidden Markov model situation for which (13)–(15) have closed form solutions. The first one is widely known as the Kalman filter.

More generally then, it is necessary to numerically evaluate (13)–(15). This is a significant challenge, primarily since (13) and (15) imply the numerical evaluation of an n_x dimensional integral.

Sequential importance resampling (SIR) is an effective approach for approximately solving (13)–(15). The essential idea is to rely on the strong law of large numbers (SLLN). More specifically, SIR generates a set indexed by $i \in [1, M]$ of randomly distributed "particles" $\tilde{x}_t^i$ and associated "weights" w_t^i such that

$$\frac{1}{M} \sum_{i=1}^{M} g(\tilde{x}_t^i) w_t^i \approx \int g(\tilde{x}_t) p_\theta(\tilde{x}_t \mid Y_t) \, \mathrm{d}\tilde{x}_t, \tag{16}$$

where g is an arbitrary (Lebesgue measurable) function, and M is a user chosen number of particles. The approximation in (16) is based on the principle that by the SLLN, the sample average of the random variables on the left of (16) converges as $M \to \infty$ to the expected value on the right of (16) with probability one.

The required particles with appropriate random distribution and associated weights can be simply generated by the following particle filter algorithm [9–12].

Algorithm 4.1: **Particle Filter**

1) Initialize particles, $\{x_0^i\}_{i=1}^M \sim p_\theta(x_0)$ and set $t = 1$;
2) Generate new particles by drawing M i.i.d. samples according to
$$\tilde{x}_t^i \sim p_\theta(\tilde{x}_t | x_{t-1}^i), \qquad i = 1, \dots, M. \tag{17}$$
3) Compute the importance weights $\{w_t^i\}_{i=1}^M$,
$$w_t^i = \frac{p_\theta(y_t|\tilde{x}_t^i)}{\sum_{j=1}^M p_\theta(y_t|\tilde{x}_t^j)}, \qquad i = 1, \dots, M. \tag{18}$$
4) For each $j = 1, \dots, M$ draw a new particle x_t^j with replacement (resample) according to,
$$\mathrm{P}(x_t^j = \tilde{x}_t^i) = w_t^i, \qquad i = 1, \dots, M. \tag{19}$$
5) If $t < N$ increment $t \mapsto t + 1$ and return to step 2, otherwise terminate.

Note that the step (17) is simple to implement. Via the model (1a) it simply involves using an appropriate random number generator to deliver a realisation from the density p_v, and then adding this to the evaluation of $f_t(x_{t-1}^i, u_t, \theta)$.

Similarly, the step (18) is uncomplicated, since by (3b) is simply involves the evaluation of the function p_e with appropriately chosen argument.

Finally, the step (19) is also simple. It is achieved by drawing a realisation γ_j from a random number generator that is uniformly distributed in the interval $[0, 1]$, and then choosing x_t^j as the element $\tilde{x}_t^i$ corresponding to the largest w_t^i satisfying $w_t^i < \gamma_j$.

Algorithm 4.1 together with the principle (16) and the model (1) allows the one-step ahead predictor (8) to be approximated as

$$\widehat{y}_{t|t-1}(\theta) = \mathbf{E}_\theta\{h_t(x_t, u_t, \theta) \mid Y_t\} \approx \sum_{i=1}^M w_t^i h_t(\tilde{x}_t^i, u_t, \theta). \tag{20}$$

In turn, this allows the prediction error cost (11) or likelihood (10) to be approximated.

Concentrating on the PE approach for a moment, attention then turns to computing the minimiser $\widehat{\theta}$. In the linear system case, an iterative gradient-based search strategy is the standard approach. This has the general form, whereby an estimate θ_k of the minimiser $\widehat{\theta}$ is refined to a better one via the update

$$\theta_{k+1} = \theta_k + \mu_k p_k, \quad p_k = H_k g_k, \tag{21a}$$
$$g_k = V_N'(\theta_k) \triangleq \frac{\mathrm{d}}{\mathrm{d}\theta} V_N(\theta)\Big|_{\theta=\theta_k}. \tag{21b}$$

Here, H_k is a positive definite matrix that delivers a search direction p_k by modifying the gradient direction, and μ_k is a step length.

It is natural to consider applying this same strategy in the nonlinear case. This requires computation of the gradient

$$V_N'(\theta_k) = -\sum_{t=1}^N \frac{\mathrm{d}}{\mathrm{d}\varepsilon_t(\theta)} \ell(\varepsilon_t(\theta_k)) \, \frac{\mathrm{d}}{\mathrm{d}\theta} \widehat{y}_{t|t-1}(\theta)\Big|_{\theta=\theta_k}. \tag{22}$$

Unfortunately, this raises a fundamental difficulty with the particle filtering approach, since it involves a predictor (20) for which the derivative with respect to θ is not computable.

This is because the particles $\tilde{x}_t^i$ depend on θ via (17)-(19) in a random manner. The same problem in determining the gradient $L_\theta'(Y_N)$ arises in seeking a maximum likelihood estimate via an equivalent argument.

The main contribution of this paper is to examine how despite these difficulties, the likelihood gradient, and associated prediction error cost gradient can, in fact, be computed by particle methods. The key step is to employ what is known as Fisher's identity, which establishes that employing a particle smoother rather than a filter allows the required gradient to be evaluated.

V. Fisher's Identity

Fisher's identity is a result from mathematical statistics [19]. It is relevant to the study of likelihood functions $L_\theta(X, Y_N)$ that involve the postulated availability of extra data X, in addition to the available data Y_N. Usually, the rationale for considering this is that the maximum likelihood estimation problem would be easier to solve if X was available.

Since in fact, X is not available, as a fallback to resorting to $L_\theta(Y_N)$, another possibility is to use an approximation $\mathcal{Q}(\theta, \theta_k)$ for $L_\theta(X, Y_N)$ which is formed by taking its conditional mean:

$$\mathcal{Q}(\theta, \theta_k) \triangleq \mathbf{E}_{\theta_k}\{L_\theta(X, Y_N) \mid Y_N\} \tag{23a}$$
$$= \int L_\theta(X, Y_N) p_{\theta_k}(X|Y_N) \, \mathrm{d}X. \tag{23b}$$

Fisher's identity establishes equality between the gradient of this approximation $\mathcal{Q}(\theta, \theta_k)$ and the gradient of the available likelihood $L_\theta(Y_N)$.

Lemma 5.1:

$$\frac{\mathrm{d}}{\mathrm{d}\theta} \mathcal{Q}(\theta, \theta_k)\Big|_{\theta=\theta_k} = \frac{\mathrm{d}}{\mathrm{d}\theta} L_\theta(Y_N)\Big|_{\theta=\theta_k}. \tag{24}$$

Proof: The proof is available in the literature [19], but is reproduced here in the interests of a self contained presentation.

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}\theta}\mathcal{Q}(\theta,\theta_k) &= \frac{\mathrm{d}}{\mathrm{d}\theta}\int \log p_\theta(X,Y_N)p_{\theta_k}(X \mid Y_N)\mathrm{dX} \\
&= \int \frac{p'_\theta(X,Y_N)}{p_\theta(X,Y_N)}p_{\theta_k}(X \mid Y_N)\mathrm{dX} \\
&= \int \frac{\mathrm{d}}{\mathrm{d}\theta}[p_\theta(Y_N)p_\theta(X \mid Y_N)]\frac{p_{\theta_k}(X \mid Y_N)}{p_\theta(X,Y_N)}\mathrm{dX} \\
&= \int p'_\theta(Y_N)p_\theta(X \mid Y_N)\frac{p_{\theta_k}(X \mid Y_N)}{p_\theta(X,Y_N)}\mathrm{dX} + \\
&\quad \int p_\theta(Y_N)p'_\theta(X \mid Y_N)\frac{p_{\theta_k}(X \mid Y_N)}{p_\theta(X,Y_N)}\mathrm{dX} \\
&= \int \frac{p'_\theta(Y_N)}{p_\theta(Y_N)}p_{\theta_k}(X \mid Y_N)\mathrm{dX} + \\
&\quad \int \frac{p'_\theta(X \mid Y_N)}{p_\theta(X \mid Y_N)}p_{\theta_k}(X \mid Y_N)\mathrm{dX} \\
&= \frac{\mathrm{d}}{\mathrm{d}\theta}\log p_\theta(Y_N) + \\
&\quad \int \frac{\mathrm{d}}{\mathrm{d}\theta}p_\theta(X \mid Y_N)\frac{p_{\theta_k}(X \mid Y_N)}{p_\theta(X \mid Y_N)}\mathrm{dX}.
\end{aligned}$$

Evaluating both sides at $\theta = \theta_k$ then delivers the result. ■ The result is typically used for theoretical analysis of the Expectation-Maximisation (EM) algorithm [19].

However, in this paper it provides a means for computing the gradient $L'(\theta)$ by providing an alternative formulation for it.

VI. Gradient Computation via Particle Smoothing

In this paper, the unavailable data X will be taken as a record of the underlying state vector in the model (1)

$$X = X_N \triangleq [x_1, x_2, \cdots, x_N]. \tag{25}$$

Using this, Bayes' rule and the Markov property results in

$$\begin{aligned}
L_\theta(X_N, Y_N) &= \log p_\theta(Y_N|X_N) + \log p_\theta(X_N) \\
&= \log p_\theta(x_1) + \sum_{t=1}^{N-1}\log p_\theta(x_{t+1}|x_t) + \sum_{t=1}^{N}\log p_\theta(y_t|x_t).
\end{aligned} \tag{26}$$

Taking the conditional expectation of both sides of this expression then delivers the function $\mathcal{Q}(\theta,\theta_k)$ defined by (23a) according to

$$\mathcal{Q}(\theta,\theta_k) = I_1 + I_2 + I_3, \tag{27}$$

where

$$I_1 = \int \log p_\theta(x_1)p_{\theta_k}(x_1|Y_N)\,\mathrm{d}x_1, \tag{28a}$$

$$I_2 = \sum_{t=1}^{N-1}\int\int \log p_\theta(x_{t+1}|x_t)p_{\theta_k}(x_{t+1},x_t|Y_N)\,\mathrm{d}x_t\,\mathrm{d}x_{t+1}, \tag{28b}$$

$$I_3 = \sum_{t=1}^{N}\int \log p_\theta(y_t|x_t)p_{\theta_k}(x_t|Y_N)\,\mathrm{d}x_t. \tag{28c}$$

Computing $\mathcal{Q}(\theta,\theta_k)$ 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.

These requirements can be met by the following particle smoothing algorithm [20].

Algorithm 6.1: **Basic Particle Smoother**

1) Run the particle filter (Algorithm 4.1) and store the filtered particles $\{\tilde{x}_t^i\}_{i=1}^M$ and their weights $\{w_t^i\}_{i=1}^M$, for $t = 1,\dots,N$.
2) Initialise the smoothed weights to be the terminal filtered weights $\{w_t^i\}$ at time $t = N$,

$$w_{N|N}^i = w_N^i, \quad i = 1,\dots,M. \tag{29}$$

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

$$w_{t|N}^i = w_t^i\sum_{k=1}^M w_{t+1|N}^k\frac{p_\theta(\tilde{x}_{t+1}^k|\tilde{x}_t^i)}{v_t^k}, \tag{30}$$

$$v_t^k \triangleq \sum_{i=1}^M w_t^i\,p_\theta(\tilde{x}_{t+1}^k|\tilde{x}_t^i). \tag{31}$$

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

This delivers particles and weights, which via the same SLLN based argument outlined in relation to the particle filter and leading to (16), allows the approximation

$$\frac{1}{M}\sum_{i=1}^M g(\tilde{x}_t^i)w_{t|N}^i \approx \int g(\tilde{x}_t)p_\theta(\tilde{x}_t \mid Y_N)\,\mathrm{d}\tilde{x}_t. \tag{32}$$

Applied to (27)–(28), this principle provides the approximation

$$\frac{\mathrm{d}\mathcal{Q}(\theta,\theta_k)}{\mathrm{d}\theta} = \frac{\mathrm{d}I_1}{\mathrm{d}\theta} + \frac{\mathrm{d}I_2}{\mathrm{d}\theta} + \frac{\mathrm{d}I_3}{\mathrm{d}\theta}, \tag{33}$$

where

$$\frac{\mathrm{d}I_1}{\mathrm{d}\theta} \approx \sum_{i=1}^M w_{1|N}^i\frac{\mathrm{d}}{\mathrm{d}\theta}\log p_\theta(\tilde{x}_1^i), \tag{34a}$$

$$\frac{\mathrm{d}I_3}{\mathrm{d}\theta} \approx \sum_{t=1}^N\sum_{i=1}^M w_{t|N}^i\frac{\mathrm{d}}{\mathrm{d}\theta}\log p_\theta(y_t|\tilde{x}_t^i), \tag{34b}$$

$$\frac{\mathrm{d}I_2}{\mathrm{d}\theta} \approx \sum_{t=1}^N\sum_{i=1}^M\sum_{j=1}^M w_{t|N}^{ij}\frac{\mathrm{d}}{\mathrm{d}\theta}\log p_\theta(\tilde{x}_{t+1}^j|\tilde{x}_t^i). \tag{34c}$$

An essential point here is that the particle smoother is used to compute an expectation with respect to densities parameterized by θ_k. Hence, the weights $w_{t|N}^i$ and the particles $\tilde{x}_t^i$ depend on θ_k, but they *do not* depend on θ. Hence forming the derivative involves computing *only* the derivative of the functional form of p_θ with respect to θ.

More specifically, via (3)

$$\frac{\mathrm{d}}{\mathrm{d}\theta}\log p_\theta(\tilde{x}^j_{t+1}|\tilde{x}^i_t) = \frac{\mathrm{d}}{\mathrm{d}\theta}\log p_v(\tilde{x}^j_{t+1} - f_t(\tilde{x}^i_t, u_t, \theta)), \tag{35a}$$

$$\frac{\mathrm{d}}{\mathrm{d}\theta}\log p_\theta(y_t|\tilde{x}^i_t) = \frac{\mathrm{d}}{\mathrm{d}\theta}\log p_e(y_t - h_t(\tilde{x}^i_t, u_t, \theta)). \tag{35b}$$

For commonly occurring densities p_v and p_e, these derivatives will have simple forms. For example, for the Gaussian situation $v_t \sim \mathcal{N}(0, \sigma_v^2)$, $e_t \sim \mathcal{N}(0, \sigma_e^2)$, the gradients (35) reduce to (in the most simple case when σ_v^2, σ_e^2 are known and hence not included in θ)

$$\frac{\mathrm{d}}{\mathrm{d}\theta}\log p_\theta(\tilde{x}^j_{t+1}|\tilde{x}^i_t) = \frac{1}{\sigma_v^2}\frac{\mathrm{d}}{\mathrm{d}\theta}f_t(\tilde{x}^i_t, u_t, \theta), \tag{36a}$$

$$\frac{\mathrm{d}}{\mathrm{d}\theta}\log p_\theta(y_t|\tilde{x}^i_t) = \frac{1}{\sigma_e^2}\frac{\mathrm{d}}{\mathrm{d}\theta}h_t(\tilde{x}^i_t, u_t, \theta). \tag{36b}$$

In this situation, the functional forms of f and h in the nonlinear model structure (1) then dictate the simplicity or otherwise of determining the cost function gradient.

VII. Algorithm Definition

The developments of the previous sections are summarised in the resulting algorithm.

Algorithm 7.1: **Gradient-based Estimation Algorithm**

1) Set $k = 0$, and initialise θ_0 as an estimate for $\widehat{\theta}$ satisfying (6);
2) Using the current iteration θ_k to parametrize the nonlinear system model (1), run the associated particle smoother Algorithm 6.1, which in turn requires the particle filter Algorithm 4.1.
3) Use the associated particles $\{\tilde{x}^i_t\}$ and weights $\{w^i_{t|N}\}$ to compute the gradient

$$p_k = -\left.\frac{\mathrm{d}}{\mathrm{d}\theta}L_\theta(Y_N)\right|_{\theta=\theta_k} = -\left.\frac{\mathrm{d}}{\mathrm{d}\theta}\mathcal{Q}(\theta, \theta_k)\right|_{\theta=\theta_k} \tag{37}$$

via (33)–(35);

4) Use this gradient to update the estimate θ_k of $\widehat{\theta}$ via the user-chosen variant of the gradient based search (21a);
5) Compute the associated cost increment

$$\Delta_k = |L_{\theta_{k+1}}(Y_N) - L_{\theta_k}(Y_N)| \tag{38}$$

via the particle filtering approximation (20) substituted into (6)–(7);

6) If $\Delta_k > \delta$, with $\delta > 0$ being some user defined threshold, then set $k \mapsto k+1$ are return to step 2. Otherwise, terminate and deliver the estimate $\widehat{\theta} = \theta_k$.

VIII. Simulation Example

This section considers estimation of the following nonlinear and time varying system.

$$x_{t+1} = ax_t + b\frac{x_t}{1+x_t^2} + c\cos(1.2t) + \nu_t, \tag{39a}$$

$$y_t = dx_t^2 + e_t, \tag{39b}$$

$$\begin{bmatrix}\nu_t\\ e_t\end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}q & 0\\ 0 & r\end{bmatrix}\right) \tag{39c}$$

where the true parameters are

$$\theta^\star = [a^\star, b^\star, c^\star, d^\star, q^\star, r^\star] = [0.5, 25, 8, 0.05, 0, 0.1]\,. \tag{40}$$

This example has been chosen for study due to it being acknowledged as a challenging estimation problem in previous studies [20, 21].

To test the effectiveness of Algorithm 7.1 in estimating this system, a Monte Carlo study was performed using 160 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 7.1 with initialisation θ_0 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 7.1 took an average of 9 seconds to complete on a 3 GHz quad-core Xeon running Mac OS 10.6.

The results of this Monte Carlo examination are provided in Table I, where the rightmost column gives the sample mean of the parameter estimate across the Monte Carlo trials plus/minus the sample standard deviation.

Parameter	True	Estimated
a	0.5	0.50 ± 0.0018
b	25.0	25.0 ± 0.66
c	8.0	8.00 ± 0.086
d	0.05	0.05 ± 0.0017
q	0	$5.96 \times 10^{-5} \pm 9.1 \times 10^{-5}$
r	0.1	0.10 ± 0.0005

TABLE I

True and estimated parameter values for the system (39) *using Algorithm 7.1; The right column shows that mean value and standard deviations using* 160 *Monte–Carlo runs, of which* 47 *were discarded due to capture in local minima.*

While this indicates that Algorithm 7.1 yields a consistent estimator of (39) it is important to note that 47 of the 160 trials were not included in Table I due to capture in local minima. This was defined according to the relative error test $|(\widehat{\theta}_i - \theta^\star_i)/\theta^\star_i)| > 0.1$ for any i'th component. The authors consider $47/160$ a relatively high proportion of failures.

In relation to this, the authors have previously studied the estimation of (39) in [15]. That work considers an alternate estimation technique based on the Expectation Maximisation (EM) algorithm. The performance of that approach on the same data sets, and with the same initialisations θ_0 is profiled in Table II.

These results are also indicative of a consistent estimator. Importantly though, only 23 of the 160 trials were not included in the calculations of Table II due to capture in local minima as defined above. This illustrates some robustness of the EM algorithm to local minima capture, which has been observed in other applications of that algorithm [22, 23].

However, compared to Algorithm 7.1, the EM algorithm is more computationally demanding, and required on average 58 seconds to compute an estimate using the same hardware.

The relationship between gradient search iterates θ_k delivered by Algorithm 7.1 and the EM algorithm iterates are further profiled in Figure 1. There it is assumed that all except the b and q parameters are known so that $\theta = [b, q]^T$. This allows the likelihood $L_\theta(Y_N)$ to be plotted as a surface, on which is overlaid the trajectory of the iterates θ_k obtained by Algorithm 7.1 in blue, and the EM algorithm in black.

In this case, both arrive at the global maximum of the likelihood function, but by different paths. Interestingly, the EM algorithm is more aggressive in early stages, and both methods eventually approach the global maximum via nearly identical paths and step lengths.

These aspects of differing robustness, differing computational load, yet similar final paths suggest a hybrid approach, in which the EM algorithm is initially employed, hopefully to obtain robustness against local minima capture, and then Algorithm 7.1 is employed in the latter stages to lower the overall computational load. It is worth noting that the properties of hybrid EM/gradient search approaches is a topic of study within the statistics literature [24].

Parameter	True	Estimated
a	0.5	0.50 ± 0.0024
b	25.0	24.9 ± 1.17
c	8.0	7.98 ± 0.15
d	0.05	0.05 ± 0.003
q	0	$3.0 \times 10^{-8} \pm 2.5 \times 10^{-8}$
r	0.1	0.12 ± 0.0003

TABLE II

True and estimated parameter values for the system (39) *using the EM-algorithm based method developed in [15]; The right column shows that mean value and standard deviations using* 160 *Monte–Carlo runs, of which* 23 *were discarded due to capture in local minima.*

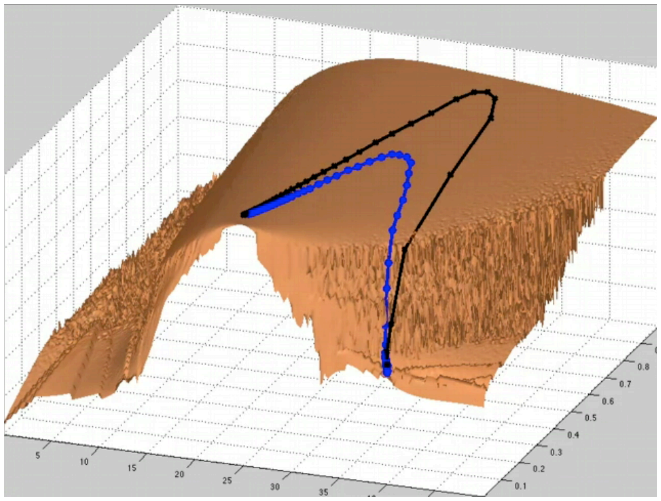

Fig. 1. *The log-likelihood plotted as a function of the two parameters b and q. Overlaying this are the parameter estimates $\theta_k = [b_k, q_k]^T$ produced by iterations of the EM algorithm (black) and the gradient search based Algorithm 7.1 (blue).*

IX. Conclusion

This paper has developed an algorithm for the estimation of a general class of non-linear systems using gradient based search. Initial simulation results suggest that its main utility may lie as the final stage of a hybrid approach that couples it with the EM algorithm.

References

[1] L. Ljung, *System Identification: Theory for the User, (2nd edition).* New Jersey: Prentice-Hall, Inc., 1999.

[2] T.Söderström and P.Stoica, *System Identification*. New York: Prentice Hall, 1989.

[3] R. Pintelon and J. Schoukens, *System Identification: A Frequency Domain Approach*. IEEE Press, 2001.

[4] L. Ljung and A. V. (Editors), "Special issue 'system identification: Linear vs nonlinear'," *IEEE Transactions of Automatic Control*, 2005.

[5] S. Rangan, G. Wolodkin, and K. Poolla, "New results for Hammerstein system identification," in *Proce. 34th IEEE Conference on Decision and Control*, New Orleans, USA, December 1995, pp. 697–702.

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

[7] I. Leontaritis and S. Billings, "Input-output parametric models for non-linear systems. part ii: stochastic non-linear systems," *International Journal of Control*, vol. 41, no. 2, pp. 329–344, 1985.

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

[9] 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*, vol. 140, 1993, pp. 107–113.

[10] P. Djuric, J. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M. Bugallo, and J. Miguez, "Particle filtering," *IEEE Signal Processing Magazine*, vol. 20, no. 5, pp. 19–38, September 2003.

[11] A. Doucet and A. M. Johansen, *Oxford Handbook of Nonlinear Filtering*. Oxford Uni. Press, 2009, ch. A tutorial on particle filtering and smoothing: fifteen years later, D. Crisan and B. Rozovsky (eds.).

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

[13] A. Wills and B. Ninness, "On gradient-based search for multivariable system estimates," *IEEE Trans. Automat. Control*, vol. 53, no. 1, pp. 298–306, 2008.

[14] R. B. Gopaluni, "A particle filter approach to identification of nonlinear processes under missing observations," *The Canadian Journal of Chemical Engineering*, vol. 86, no. 6, pp. 1081–1092, Dec. 2008.

[15] A. Wills, T. Schon, and B. Ninness, "Parameter estimation for discrete-time nonlinear systems using em," in *Proceedings of the 17th IFAC World Congress, Seoul, Korea*, 2008.

[16] B. Anderson and J. Moore, *Optimal Filtering*. Prentice Hall, 1979.

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

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

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

[20] A. Doucet, S. J. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statistics and Computing*, vol. 10, no. 3, pp. 197–208, 2000.

[21] S. J. Godsill, A. Doucet, and M. West, "Monte Carlo smoothing for nonlinear time series," *Journal of the American Statistical Association*, vol. 99, no. 465, pp. 156–168, Mar. 2004.

[22] S. Gibson and B. Ninness, "Robust maximum-likelihood estimation of multivariable dynamic systems," *Automatica*, vol. 41, no. 10, pp. 1667–1682, October 2005.

[23] S. Gibson, A. Wills, and B. Ninness, "Maximum-likelihood parameter estimation of bilinear systems," *IEEE Trans. Automat. Control*, vol. 50, no. 10, pp. 1581–1596, 2005.

[24] M. Jamshidian and R. I. Jennrich, "Acceleration of the EM algorithm by using quasi-newton methods," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 59, no. 3, pp. 569–587, 1997.