Some System Identification Challenges and Approaches

Brett Ninness *

* *School of Electrical Engineering and Computer Science, University of Newcastle, Callaghan, NSW, 2308, Australia (E-mail: Brett.Ninness@newcastle.edu.au).*

Abstract: The field of control-oriented system identification is mature. Nevertheless, it is still very active. This is because there are many important unsolved challenges. Of these, this paper considers a selection. This involves considering the estimation of general nonlinear model structures, together with accurate error bounds, using methods that scale well to models of high dimension. A particular strength of the system identification field is that it has always actively sought to understand, embrace and develop ideas from other fields, such as statistics, mathematics and econometrics. This paper proposes a continuation of this successful strategy by proposing and profiling the adoption of new ideas originating in statistics, signal processing and statistical mechanics.

Keywords: Prediction Error Estimation, Maximum-Likelihood Methods, Bayesian Estimation.

1. INTRODUCTION

System identification is by now a very mature field. It offers solutions that are widely used and supported by a rich body of underlying theory. Furthermore, there are a range of substantial monographs [48, 76, 63, 14, 37, 35, 82] devoted to the subject, and comprehensive software packages available for its deployment [49, 1, 62, 85].

It is generally accepted that the origins of the field can be traced back at least to Gauss's development of the method of least squares [2] in 1795, and the Reverend Bayes' development of the idea of conditional probability prior to 1763 [17].

It is also widely recognised that current system identification practice is strongly based on methods and underpinning theory developed in the statistics and mathematics communities. Examples include the concept of likelihood and associated statistical inference [25, 26, 44], axiomatic probability theory [16], and the results of stochastic process and prediction theory [41, 84, 21].

This body of work was first applied in an engineering setting in the mid-1960's [6]. Progress from that point was swift, leading to the now famous observation in the 1971 survey paper [5] that for the then-new system identification field '*New methods are suggested en masse, and, on the surface, the field appears to look more like a bag of tricks than a unified subject*'.

The necessary unification was achieved over the ensuing decade (or so) with the development of the prediction error (PE) approach [50, 51] as a natural complement to the more classical maximum likelihood (ML) approach [4].

Arguably, the work arising from this effort now forms what can be considered the cornerstone theory and algorithms of current system identification practice. The following section 1.1 provides an overview of these foundations.

This is meant as a basis for the following section 1.2 that identifies some key open challenges, which are then addressed in the remainder of the work,

1.1 Foundations

A central principle of modern system identification is the separate recognition of the model structure being employed, the estimation method being used, and the implementing algorithm.

For example, a very wide class of model structures that are commonly used may be captured by the state-space description

$$x_{t+1} = f(x_t, u_t, \theta) + g(u_t, \nu_t, \theta), \tag{1}$$

$$y_t = h(x_t, u_t, \theta) + e_t. \tag{2}$$

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 at discrete integer time points t. Furthermore, the signals $\{\nu_t\}$ and $\{e_t\}$ model unmeasured corruptions as zero mean, finite variance, i.i.d. stochastic processes.

Finally, $\theta \in \mathbf{R}^{n_\theta}$ is a vector of (unknown) parameters that specifies the mappings $f(\cdot), g(\cdot), h(\cdot)$. Depending on the definition of these mappings, the above formulation can accommodate various situations. For example, it clearly encompasses the familiar linear state space structure

$$\begin{bmatrix} x_{t+1} \\ y_t \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} x_t \\ u_t \end{bmatrix} + \begin{bmatrix} \nu_t \\ e_t \end{bmatrix}. \tag{3}$$

It can also represent the equally familiar linear transfer function structure [48]

$$y_t = G(q, \theta)u_t + H(q, \theta)e_t \tag{4}$$

⋆ This work was supported by the Australian Research Council.

where q is the forward shift operator, and G, H are rational transfer functions in this operator. More complicated nonlinear structures can also be represented by the model structure (1),(2).

Given this specification of a model structure, the system identification problem is to determine an estimate $\widehat{\theta}$ of θ given the observations $Y_N = \{y_1, \cdots, y_N\}$, $U_N = \{u_1, \cdots, u_N\}$.

The prediction error (PE) estimation method provides a general solution to this problem given as

$$\widehat{\theta} = \arg\min_{\theta \in \mathbf{R}^{n_\theta}} V_N(\theta) \tag{5}$$

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

$$V_N(\theta) = \frac{1}{N}\sum_{t=1}^{N} \ell(\varepsilon_t(\theta)), \qquad \varepsilon_t(\theta) = y_t - \widehat{y}_{t|t-1}(\theta). \tag{6}$$

Here, with $\mathbf{E}_\theta\{\cdot\}$ denoting the statistical expectation operator with respect to a probability density function (pdf) dependent on θ,

$$\widehat{y}_{t|t-1}(\theta) \triangleq \mathbf{E}_\theta\{y_t \mid Y_{t-1}\} \tag{7}$$

is the mean square optimal one-step ahead predictor of y_t based on the model (1),(2) parametrized by θ and the past observations $Y_{t-1} = \{y_1, \cdots, y_{t-1}\}$. The function $\ell(\cdot)$ is an arbitrary and user-chosen positive mapping, with $\ell(x) = x^T x = \|x\|^2$ being a common situation.

Again returning to concrete examples, for the special linear state space model case (3) and Gaussian distributions for the stochastic components

$$\begin{bmatrix} \nu_t \\ e_t \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} Q & S \\ S^T & R \end{bmatrix} \right) \tag{8}$$

the required predictor can be computed using a Kalman filter [3], where $\mathcal{N}(\mu, P)$ denotes the multivariable Gaussian density of mean μ and variance P and $\cdot^T$ indicated matrix transpose.

In the scalar signal transfer function situation (4), the steady state Wiener filter

$$\widehat{y}_{t|t-1}(\theta) = H^{-1}(q,\theta)G(q,\theta)u_t + \left[1 - H^{-1}(q,\theta)\right] y_t \tag{9}$$

may be used provided that the parametrization is chosen so that $H(q,\theta)$ is monic (feedthrough term equal to one) for all values of θ [48].

This PE approach (5)-(7) draws inspiration from the classical maximum-likelihood (ML) technique pioneered by Fisher [25, 26] almost a century ago, and widely used today within statistics [44] and myriad other application areas, such as telecommunications [78] to name but one.

The ML involves maximising the joint θ-dependent density (likelihood) $p_\theta(Y_N)$ of the observations:

$$\widehat{\theta} = \arg\max_{\theta \in \mathbf{R}^{n_\theta}} p_\theta(y_1, \cdots, y_N). \tag{10}$$

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

$$p_\theta(y_1, \cdots, y_N) = p_\theta(y_1) \prod_{t=2}^{N} p_\theta(y_t | Y_{t-1}). \tag{11}$$

Therefore, since logarithm is a monotonic function, the maximisation problem (10) is equivalent to the minimisation problem

$$\widehat{\theta} = \arg\min_{\theta \in \mathbf{R}^{n_\theta}} -L_\theta(Y_N) \tag{12}$$

where $L_\theta(Y_N)$ is the log-likelihood which is given by:

$$L_\theta(Y_N) \triangleq \log p_\theta(Y_N) = \log p_\theta(y_1) + \sum_{t=2}^{N} \log p_\varepsilon(\varepsilon_t(\theta)) \tag{13}$$

where, with some abuse of notation, $p_\varepsilon(\cdot)$ refers to the pdf governing $\varepsilon_t(\theta)$. The ML estimate is therefore closely related to the PE estimate (5) with an appropriate choice of the function $\ell(\cdot)$. In particular, in the scalar Gaussian case where $\varepsilon_t \sim \mathcal{N}(0, \sigma^2)$ and are independent

$$-\log p_\varepsilon(\varepsilon_t(\theta)) = \log\sqrt{2\pi} + \log\sigma^2 + \frac{1}{\sigma^2}\varepsilon_t^2(\theta) \tag{14}$$

and hence, neglecting constant terms that do not depend on θ, the common scalar choice of $\ell(x) = x^2$ delivers

$$V_N(\theta) = \frac{\sigma^2}{N}\left[\log p_\theta(y_0) - L_\theta(Y_N)\right]. \tag{15}$$

This renders the prediction error estimate (5) asymptotically (as N grows) equal to the ML one.

This close connection is important, since the ML estimate generally (but not universally) possesses very attractive properties such as strong consistency, asymptotic normality, and asymptotic efficiency (asymptotic achievement of the Cramér–Rao lower bound) which then flow (with some modifications) to the prediction error estimate.

More specifically, under mild assumptions on the model structure (1),(2), the noise processes $\{\nu_t\}$, $\{e_t\}$ and the observed input $\{u_t\}$, for the PE estimate (5)

$$\lim_{N\to\infty} \widehat{\theta} = \theta_\star \triangleq \arg\min_\theta \lim_{N\to\infty} \mathbf{E}\{V_N(\theta)\} \tag{16}$$

with probability one [50]. Furthermore

$$\sqrt{N}\left[\widehat{\theta}_N - \theta_\star\right] \xrightarrow{\mathcal{D}} \mathcal{N}(0, P) \qquad \text{as } N \to \infty \tag{17}$$

where $P = R^{-1}QR^{-1}$, and with $\cdot'$ denoting differentiation $\mathrm{d}/\mathrm{d}\theta$ with respect to θ

$$R \triangleq \lim_{N\to\infty} \mathbf{E}\{V_N''(\theta_\star)\}, \; Q \triangleq \lim_{N\to\infty} N\mathbf{E}\left\{V_N'(\theta_\star)[V_N'(\theta_\star)]^T\right\} \tag{18}$$

Furthermore, in the situation that a true parameter value $\theta_\circ$ exists, it is straightforward to establish that $\theta_\circ$ satisfies the definition (12) for $\theta_\star$.

Additionally, an elementary result [44] in the theory of likelihood is that in this same case that $\theta_\circ$ exists

$$\mathbf{E}\left\{\frac{\mathrm{d}}{\mathrm{d}\theta}L_{\theta_\circ}(Y_N)\left[\frac{\mathrm{d}}{\mathrm{d}\theta}L_{\theta_\circ}(Y_N)\right]^T\right\} = -\mathbf{E}\left\{\frac{\mathrm{d}^2}{\mathrm{d}\theta\mathrm{d}\theta^T}L_{\theta_\circ}(Y_N)\right\}. \tag{19}$$

Therefore, in the Gaussian case where (15) applies, and neglecting the $N^{-1}p_\theta(y_0)$ term which disappears as N grows

$$\frac{N^2}{\sigma^4}\mathbf{E}\left\{V_N'(\theta_\circ)[V_N'(\theta_\circ)]^T\right\} = \mathbf{E}\left\{L_{\theta_\circ}'(Y_N)[L_{\theta_\circ}'(Y_N)]^T\right\} \tag{20}$$

$$= -\mathbf{E}\left\{L_{\theta_\circ}''(Y_N)\right\} = \frac{N}{\sigma^2}\mathbf{E}\left\{V_N''(\theta_\circ)\right\}. \tag{21}$$

As a consequence, via the definitions in (18) $Q = \sigma^2 R$ and hence again using (15)

$$P = \sigma^2 R^{-1} = \left[\lim_{N\to\infty}\frac{1}{N}\mathcal{I}_N(\theta_\circ)\right]^{-1} \tag{22}$$

where $\mathcal{I}_N(\theta_\circ)$ is the Fisher information matrix [44] associated with the log likelihood $L_\theta(Y_N)$

$$\mathcal{I}_N \triangleq -\mathbf{E}\left\{L''_{\theta_\circ}(Y_N)\right\}. \tag{23}$$

As a consequence, in this Gaussian case, the distributional result (17) implies that the approximation

$$\mathrm{Cov}\{\widehat{\theta}\} \approx \frac{1}{N}P \approx \mathcal{I}_N(\theta_\circ) \tag{24}$$

steadily improves as N increases. This further implies that the prediction error estimate (5) asymptotically achieves the Cramér–Rao lower bound [44]

$$\mathrm{Var}\{\widehat{\theta}\} \geq \mathcal{I}_N^{-1}(\theta_\circ). \tag{25}$$

This is not surprising, since in this same Gaussian case it has already been established that the PE estimate is asymptotically equal to the ML one.

More generally, under an assumption that $\varepsilon_t(\theta_\star)$ is an independent process, and that the choice of function $\ell(\cdot)$ used in (6) together with the distribution of $\varepsilon_t(\theta_\circ)$ satisfy

$$\mathbf{E}\left\{\frac{\mathrm{d}}{\mathrm{d}\varepsilon_t}\ell(\varepsilon_t(\theta_\circ)\right\} = 0 \tag{26}$$

then (17) holds with covariance matrix P given by

$$P = \sigma^2\left[\lim_{N\to\infty}\frac{1}{N}\sum_{t=1}^{N}\mathbf{E}\left\{\psi_t(\theta_\circ)\psi_t^T(\theta_\circ)\right\}\right]^{-1} \tag{27}$$

where

$$\psi_t(\theta) \triangleq -\frac{\mathrm{d}\widehat{y}_{t|t-1}(\theta)}{\mathrm{d}\theta}, \quad \sigma^2 \triangleq \frac{\mathbf{E}\{(\mathrm{d}\ell(\varepsilon_t(\theta_\star))/\mathrm{d}\varepsilon_t)^2\}}{\left[\mathbf{E}\{\mathrm{d}^2\ell(\varepsilon_t(\theta_\star))/(\mathrm{d}\varepsilon_t)^2\}\right]^2}. \tag{28}$$

In the common case $\ell(x) = x^2$, the expression for σ^2 simplifies to $\sigma^2 = \mathbf{E}\{\varepsilon_t^2(\theta_\star)\}$. Consequently, in the non-Gaussian and/or non-quadratic $\ell(x)$ situations, the asymptotic covariance P is a scalar multiple of the information matrix $\mathcal{I}_N$ defined by (23) that is achieved by the ML estimator in the Gaussian case. Vice-versa, the choice

$$\ell(x) = -\log p_\varepsilon(x) \tag{29}$$

where $p_\varepsilon(\cdot)$ is the pdf of $\varepsilon_t(\theta_\circ)$ implies that the PE estimate asymptotic covariance P given by (27) is equal to the inverse information matrix $\mathcal{I}_N^{-1}(\theta_\circ)$, and hence equal to the Cramér–Rao lower bound.

The utility of the distributional result (17) is that as a consequence

$$N[\widehat{\theta}-\theta_\star]^T P^{-1}[\widehat{\theta}-\theta_\star] \xrightarrow{\mathcal{D}} \chi^2_{n_\theta}. \tag{30}$$

where $\chi^2_{n_\theta}$ signifies the chi-squared density with n_θ degrees of freedom.

Therefore, if a threshold κ is chosen such that (say) 95% of realisations of a $\chi^2_{n_\theta}$ distributed random variable would be less than κ, then the ellipse

$$[\widehat{\theta}-\theta_\star]^T P^{-1}[\widehat{\theta}-\theta_\star] \leq \frac{\kappa}{N} \tag{31}$$

specifies a confidence region where, with probability 0.95, the limiting estimate $\theta_\star$ lies.

Of course, computing confidence regions this way depends on first computing the covariance matrix P defined by (27). To address this, when $\ell(x) = x^2$ the obvious approximation suggested by (27) of

$$P \approx \widehat{P} \triangleq \widehat{\sigma}^2\left[\frac{1}{N}\sum_{t=1}^{N}\mathbf{E}\left\{\psi_t(\widehat{\theta})\psi_t^T(\widehat{\theta})\right\}\right]^{-1}, \tag{32}$$

$$\widehat{\sigma}^2 \triangleq \frac{1}{N}\sum_{t=1}^{N}\varepsilon_t^2(\widehat{\theta}) \tag{33}$$

is commonly employed.

Often, it is of more interest to quantify the error in a function $\gamma(\widehat{\theta})$ of the parameter estimate. For example, the frequency response of a model parametrized by $\widehat{\theta}$. To address this requirement, a further approximating step is usually introduced whereby the functional relationship is reduced to a first order expansion, and Gauss' approximation formula [44] is then coupled with the asymptotic distributional result (17) to deliver ($\cdot^\star$ denotes conjugate transpose)

$$\sqrt{N}\left[\gamma(\widehat{\theta})-\gamma(\theta_\star)\right] \xrightarrow{\mathcal{D}} \mathcal{N}\left(0, \frac{\mathrm{d}\gamma(\widehat{\theta})}{\mathrm{d}\theta}^\star P\frac{\mathrm{d}\gamma(\widehat{\theta})}{\mathrm{d}\theta}\right). \tag{34}$$

Confidence regions on $\gamma(\widehat{\theta})$ are then obtained in the same way that (31) was obtained and again, the approximation (32),(33) is employed.

Finally, in terms of actually computing the estimate $\widehat{\theta}$, note that via (5) or (10) this requires the solution of an optimisation problem. Except for some special linearly parametrized cases, the associated cost $V_N(\theta)$ or $-L_\theta(Y_N)$ will be non-convex, and its minimizer $\widehat{\theta}$ will not be expressible in closed form.

To cope with this (taking the PE case (5) as an example) an iterative search based on the gradient $V'_N(\theta)$ of the cost is generally successful. This involves finding an initial approximation θ_0 of $\widehat{\theta}$, and then iteratively improving this approximation to a new one via the update

$$\theta_{k+1} = \theta_k - \mu_k J_k V'_N(\theta_k) \tag{35}$$

where μ_k is a scalar 'step-length' and J_k is a search direction modifying-matrix that may be chosen in various ways [20, 86].

1.2 Challenges and Potential Approaches

Of course, all this material is very well known. The point of this whirlwind tour of system identification fundamentals is to establish notation and emphasize the assumptions and approximations underpinning current practice.

This provides a foundation for the main purpose of the paper - to discuss some open and important challenges and profile some new approaches that, as will be illustrated, have potential. This promise derives in part from their proven success in dealing with related challenges in related research fields. Three key challenges and three non-standard approaches will be considered.

Estimation of General Non-linear models The first challenge is that of computing the parameter estimate $\widehat{\theta}$ for general nonlinear model structures such as (1), (2). While, as just profiled, much of the fundamental theory applies for such structures, actually solving the optimisation problem (5), or in fact even computing $V_N(\theta)$ is not without difficulty.

In what follows, the approach of using 'particle filtering' to compute $V_N(\theta)$ and the Expectation-Maximization (EM) algorithm to solve (5) and compute $\widehat{\theta}$ will be profiled.

Estimation of high dimensional systems The second challenge is again that of solving (5), but in the situation where the state dimension n_x coupled with the input-output dimensions n_u, n_y imply a high computational burden on standard gradient search based algorithms for solving (5). The Expectation-Maximization algorithm will again be employed as a potentially useful new approach.

Accurate Error Quantification The final challenge to be discussed here is that of computing error quantifications on parameter estimates. A key imposed requirement is the avoidance of asymptotic in data length theory in the interests of delivering results that are accurate for possibly 'short' data lengths. For this purpose, the use of Markov-Chain Monte–Carlo (MCMC) methods will be illustrated.

The identification of these challenges is certainly not novel to this paper. They, and other unresolved challenges are widely recognised [46, 47, 28]. As a result, over the last several years they have attracted significant research effort that has delivered important contributions.

For example, by drawing on principles from the field of stochastic realization, subspace-based estimation methods provide an effective solution to the estimation of high-dimension multivariable systems [77, 81].

As another example, effective techniques for hard-bounding estimation error for finite data length N have been developed [64, 58, 83, 59], which arguably have their roots in the computer-science community, particularly in the field of computational complexity.

Finally, the application of ideas from the machine learning community to nonlinear system identification problems has recently attracted significant interest [32, 45].

The work here is in the same theme of applying methods that are of proven effectiveness in other fields, but nevertheless novel in the system identification field.

It is important to emphasize that employment of these new approaches is by no means the current author's invention. Their first application problems relevant or related to system identification ones can in all cases be traced back to others.

Equally, these original contributions have largely occurred in literature outside the automatic control/system identification field, and hence with terminology and assumptions peculiar to other fields. The author believes this has presented somewhat of a barrier to their appreciation and understanding to system identification researchers. The paper at hand is therefore meant as a tutorial/survey contribution designed to diminish this obstacle.

2. NON-LINEAR MODELS AND PARTICLE FILTERING

The estimation of models for nonlinear systems has and continues to be a research topic attracting very significant attention. For example, conference sessions (such as [46]) directed specifically at this topic regularly form a very significant component of the system identification program streams at the major conferences.

Given this intense activity, the scope of existing solutions is too wide to be adequately summarised here. Nevertheless, it is arguable that current approaches primarily focus on specific specialized subclasses of nonlinear systems. Additionally, there is much attention on developing specialized estimation methods adapted to these subclasses, as opposed to employing general methods such as the PE and ML ones profiled in the introduction.

There are good reasons for this, and a primary one is that to employ a PE or ML method, either the conditional density $p_\theta(y_t \mid Y_{t-1})$ or the mean with respect to it $\widehat{y}_{t|t-1}(\theta) = \mathbf{E}_\theta\{y_t \mid Y_{t-1}\}$ is required to compute the log-likelihood $L_\theta(Y_N)$ or the PE cost $V_N(\theta)$. If the model structure is quite general, such as (1), (2), then computing the required density or mean is a very significant challenge.

To expand on this, note that by employing the Markov properties of the model (1), (2)

$$\begin{aligned} p_\theta(x_t \mid Y_t) &= \frac{p_\theta(x_t, Y_t)}{p_\theta(Y_t)} \\ &= \frac{p_\theta(y_t, Y_{t-1}, x_t)}{p_\theta(Y_{t-1}, x_t)} \cdot \frac{p_\theta(Y_{t-1}, x_t)}{p_\theta(Y_{t-1})} \cdot \frac{p_\theta(Y_{t-1})}{p_\theta(Y_t)} \\ &= \frac{p_\theta(y_t \mid x_t)\, p_\theta(x_t \mid Y_{t-1})}{p_\theta(y_t \mid Y_{t-1})} \end{aligned} \tag{36}$$

where by the law of total probability

$$p_\theta(y_t \mid Y_{t-1}) = \int p_\theta(y_t \mid x_t) p_\theta(x_t \mid Y_{t-1})\, \mathrm{d}x_t. \tag{37}$$

Additionally, again by the law of total probability and the Markov nature of (1),(2)

$$p_\theta(x_{t+1} \mid Y_t) = \int p_\theta(x_{t+1} \mid x_t)\, p_\theta(x_t \mid Y_t)\, \mathrm{d}x_t. \tag{38}$$

Together (36) and (38) are the general so-called 'measurement update' and 'time update' equations solving the general non-linear filtering problem, with (38) being an instance of the Chapman–Kolmogorov equation.

In principle, their solution provides the conditional probability (37) that further delivers the likelihood $L_\theta(Y_N)$ via (11). Furthermore, again with the probability density (37) in hand, the conditional expectation

$$\widehat{y}_{t|t-1}(\theta) = \mathbf{E}_\theta\{y_t \mid Y_{t-1}\} = \int y_t\, p_\theta(y_t \mid Y_{t-1})\, \mathrm{d}y_t \tag{39}$$

can be computed to then provide the PE cost (6).

Unfortunately, there are very few cases, such as the linear Gaussian, and the discrete time, discrete state Hidden Markov Model situation for which (36)-(39) have closed form solutions. The first one is widely known as the Kalman filter.

More generally then, it is necessary to numerically evaluate (36)-(39). This is a significant challenge, primarily since (37) and (38) imply the numerical evaluation of an n_x dimensional integral. Multidimensional integration is notoriously demanding of computer resources. For example, with M grid points in each dimension, M^{n_x} function evaluations are required to numerically approximate an n_x-dimensional integral using any straightforward method such as Simpson's rule.

An effective approach diminishing this complexity is a random sampling one, that relies on the strong law of large numbers (SLLN). The essential idea is that if a vector random number generator is available that generates realisations $x^i \sim \phi(x)$ from some 'target' density ϕ, then by the SLLN and for an arbitrary (measurable) function $\gamma(\cdot)$

$$\lim_{M\to\infty} \frac{1}{M}\sum_{i=1}^{M} \gamma(x^i) = \mathbf{E}\left\{\gamma(x)\right\} = \int \gamma(x)\phi(x)\,\mathrm{d}x \qquad (40)$$

with probability one. Hence the multidimensional integral on the right can be approximated as

$$\int \gamma(x)\phi(x)\,\mathrm{d}x \approx \frac{1}{M}\sum_{i=1}^{M}\gamma(x^i) \qquad (41)$$

for some finite M. Furthermore, by the Central Limit Theorem, the approximation error in the above will converge in distribution to a Gaussian density with variance proportional to $1/M$, and hence the error can be expected to decay like $1/\sqrt{M}$. These are established ideas, and the work [66] is a classic text on the topic.

The application of these techniques to the above filtering relations (36)-(39) is the essence of what has become informally known as 'particle filtering'. The remainder of this section is devoted to explaining the approach and it's application to the estimation of the nonlinear model structure (1),(2).

For this latter purpose, it will prove useful to recognize that this structure can also be represented by the stochastic description

$$x_{t+1} \sim p_\theta(x_{t+1} \mid x_t) \qquad (42)$$
$$y_t \sim p_\theta(y_t \mid x_t) \qquad (43)$$

where

$$p_\theta(x_{t+1} \mid x_t) = p_{g(u_t,\nu_t,\theta)}\left(x_{t+1} - f(x_t,u_t,\theta)\right) \qquad (44)$$
$$p_\theta(y_t \mid x_t) = p_e\left(y_t - h(x_t,u_t,\theta)\right). \qquad (45)$$

Here, as is common practice, 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.

2.1 Sequential Importance Resampling (Particle Filtering)

With this in mind, the challenge now is to build the random number generator that delivers the required realisations $\{x^i\}$ distributed according to a target density $\phi(x)$ equal to the filtering density $p_\theta(x_t \mid Y_t)$.

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

A realisation $x^j \sim \phi(x)$ that is distributed according to the target density $\phi(x)$ 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 $i,j = 1,\cdots,M$, the realisation x^j is selected as $\tilde{x}^i$ randomly according to

$$p(x^j = \tilde{x}^i) = w(\tilde{x}^i), \qquad w(x) = \frac{\phi(x)}{\lambda(x)}. \qquad (46)$$

This step is known as a 'resampling', and the random assignment is done in an independent fashion. The above assignment rule works, since by the independence, the probability that as a result x^j takes on the value $\tilde{x}^i$ is the probability $\lambda(\tilde{x}^i)$ that $\tilde{x}^i$ was realised, times the probability $w(\tilde{x}^i) = \phi(\tilde{x}^i)/\lambda(\tilde{x}^i)$ that x^j is then assigned this value. Since the $\lambda(\tilde{x}^i)$ terms in this multiplication cancel, $p(x^j = \tilde{x}^i) = \phi(\tilde{x}^i)$, so that x^j is a realisation from the required density $\phi(x)$.

An important advantage of this resampling approach, is that if the ratio $w(x) = \phi(x)/\lambda(x)$ can be computed, even if $\phi(x)$ is unknown, then realisations $\{x^j\}$ distributed according to $\phi(x)$ can still be generated.

The challenge in achieving this is clearly the specification of a density $\lambda(x)$ from which it is both feasible to generate realisations $\{\tilde{x}^i\}$, and for which the ratio $w(x)$ of the unknown target density $\phi(x)$ to this specified $\lambda(x)$ can be computed.

To address this, consider the target density $\phi(x_t) = p_\theta(x_t \mid Y_t)$ being the filtering density associated with the model structure (1),(2). We begin by concentrating on the state update component (1) and its stochastic equivalent (42). A realisations $\tilde{x}_t^i \sim p_\theta(\tilde{x}_t \mid x_{t-1})$ may be obtained by simply substituting a given x_{t-1} into f, then generating a realisation $\nu_t^i \sim p_\nu$, and then finally computing a realised $\tilde{x}_t^i$ according to the formula (1).

If additionally x_{t-1} was a realisation distributed as $x_{t-1} \sim p_\theta(x_{t-1} \mid Y_{t-1})$, then by the law of total probability, the achieved realisation $\tilde{x}_t^i$ will have associated density

$$\lambda(\tilde{x}_t) = \int p_\theta(\tilde{x}_t \mid x_{t-1})\, p_\theta(x_{t-1} \mid Y_{t-1})\,\mathrm{d}x_{t-1}. \qquad (47)$$

However, according to the time update equation (38), this implies that the density in question is

$$\lambda(\tilde{x}_t) = p_\theta(\tilde{x}_t \mid Y_{t-1}). \qquad (48)$$

Furthermore, by the measurement update (37), the ratio $w = \phi/\lambda$ for this choice of ϕ and λ is

$$w(\tilde{x}_t^i) = \frac{p_\theta(\tilde{x}_t^i \mid Y_t)}{\lambda(\tilde{x}_t^i)} = \frac{p_\theta(\tilde{x}_t^i \mid Y_t)}{p_\theta(\tilde{x}_t \mid Y_{t-1})} = \frac{p_\theta(y_t \mid \tilde{x}_t^i)}{p_\theta(y_t \mid Y_{t-1})}. \qquad (49)$$

According to (45) the numerator in this expression is very simply computable as

$$p_\theta(y_t \mid \tilde{x}_t^i) = p_e\left(y_t - h(\tilde{x}_t^i, u_t, \theta)\right). \qquad (50)$$

Finally, note that the denominator in (49) is a constant independent of $\tilde{x}_t^i$ and that $w(\tilde{x}_t^i) = p(x_t^j = \tilde{x}_t^i)$ is a probability density function so that

$$1 = \sum_{i=1}^{M} w(\tilde{x}_t^i) = \frac{1}{p_\theta(y_t \mid Y_t)}\sum_{i=1}^{M} p_\theta(y_t \mid \tilde{x}_t^i) \qquad (51)$$

and hence

$$w(\tilde{x}_t^i) = \frac{1}{\kappa}\, p_\theta(y_t \mid \tilde{x}_t^i), \qquad \kappa = \sum_{i=1}^{M} p_\theta(y_t \mid \tilde{x}_t^i). \qquad (52)$$

Therefore, by using the general time and measurement update equations (36), (38), if realisations $x_{t-1}^i \sim p_\theta(x_{t-1} \mid Y_{t-1}\}$ are available, then the weight function w given by (52) may be used to compute realisations $x_t^i \sim p_\theta(x_t \mid Y_t)$ by the resampling approach (46).

This recursive approach for generating realisations $\{x_t^i\}$ from the filtering density $p_\theta(x_t \mid Y_t)$ is known as sequential

importance resampling (SIR). More informally, the realisations $\{x_t^j\}$, $\{\tilde{x}_t^i\}$ are known as particles, and the method is known as particle filtering [23].

Algorithm 1. **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

$$\tilde{x}_t^i \sim p_\theta(\tilde{x}_t | x_{t-1}^i), \qquad i = 1, \ldots, M. \tag{53}$$

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

$$w_t^i \triangleq w(\tilde{x}_t^i) = \frac{p_\theta(y_t|\tilde{x}_t^i)}{\sum_{i=1}^M p_\theta(y_t|\tilde{x}_t^i)}, \qquad i = 1, \ldots, M; \tag{54}$$

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

$$p(x_t^i = \tilde{x}_t^j) = w_t^j, \qquad j = 1, \ldots, M; \tag{55}$$

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

□

There are some aspects of this method which are important to its successful employment. For example, since Algorithm 1 delivers realisations $x_t^i \sim p_\theta(x_t^i \mid Y_t)$, this suggests that expected values with respect to $p_\theta(x_t \mid Y_t)$ can approximately computed as

$$\mathbf{E}_\theta\{g(x_t) \mid Y_t\} = \int g(x_t) p_\theta(x_t \mid Y_t)\, \mathrm{d}x_t \approx \frac{1}{M} \sum_{i=1}^M g(x_t^i) \tag{56}$$

where $g(\cdot)$ is an almost arbitrary function (it must be Lebesgue measurable). Here, as mentioned earlier, the strong law of large numbers (SLLN) is being appealed to, which states that if $\{x_t^i\}$ is suitably uncorrelated, then the right hand sum converges to the integral with probability one as the number of particles M tends to infinity.

However, a key feature of the resampling step (55) is that it takes an independent sequence $\{\tilde{x}_t^i\}$ and delivers a dependent one $\{x_t^i\}$. This is the price paid for achieving the distributional result $x_t^i \sim p_\theta(x_t \mid Y_t)$. It arises, because (55) draws with replacement from the 'alphabet' $\{\tilde{x}_t^j\}$ and hence several x_t^i may be equal to the same value $\tilde{x}_t^j$. For example, if $w_t^j = 0.1$, and $M = 100$, then approximately ten x_t^i will be equal to $\tilde{x}_t^j$.

Unfortunately, this will degrade the accuracy of the approximation (56), 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 [60].

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

$$\frac{1}{M} \sum_{i=1}^M g(\tilde{x}_t^i) w(\tilde{x}_t^i) \approx \int g(\tilde{x}_t) w(\tilde{x}_t) p_\theta(\tilde{x}_t \mid Y_{t-1})\, \mathrm{d}\tilde{x}_t \tag{57}$$

$$= \int g(\tilde{x}_t) \frac{p_\theta(\tilde{x}_t^i \mid Y_t)}{p_\theta(\tilde{x}_t \mid Y_{t-1})} p_\theta(\tilde{x}_t \mid Y_{t-1})\, \mathrm{d}\tilde{x}_t \tag{58}$$

$$= \int g(\tilde{x}_t) p_\theta(\tilde{x}_t \mid Y_t)\, \mathrm{d}\tilde{x}_t = \mathbf{E}_\theta\{g(\tilde{x}_t) \mid Y_t\} \tag{59}$$

where the transition from (57) to (58) follows by (49). Note that the expectation in (59) is identical to that in (56). However, since the sum in (57) involves independent $\{\tilde{x}_t^i\}$ rather than the dependent $\{x_t^i\}$ used in (56), it will generally be a more accurate approximation to the expectation.

As a result it is preferable to use the left hand side of (57) rather than the right hand side of (56). The former, due to use of the 'weights' $\{w(\tilde{x}_t^i)\}$ is an example of what is known as 'importance sampling' [66]. This explains the middle term in the SIR name given to Algorithm 1.

Of course, this also suggests that the resampling step (55) is not essential, and one could simplify Algorithm 1 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 support of 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 (48) for the proposal density have lead to a range of different particle filtering methods [23].

Nevertheless, they share a common thread. Namely, their deliverables are sets of values $\{\tilde{x}_t^i\}$, $\{w_t^i = w(\tilde{x}_t^i)\}$, $\{x_t^i\}$ 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 the left hand side of (57) (preferable) or right hand side of (56).

The techniques just profiled can be simply extended to also offer importance sampling approximations for expected values with respect to a 'smoothing' target density $\phi = p(x_t \mid Y_N)$ via the following algorithm [22].

Algorithm 2. **Basic Particle Smoother**

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

$$w_{N|N}^i = w_N^i, \quad i = 1, \ldots, M. \tag{60}$$

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 $\{\tilde{x}_t^i, \tilde{x}_{t+1}^i\}_{i=1}^M$ by using (44) in the expressions

$$w_{t|N}^i = w_t^i \sum_{k=1}^M w_{t+1|N}^k \frac{p_\theta(\tilde{x}_{t+1}^k|\tilde{x}_t^i)}{v_t^k}, \tag{61}$$

$$v_t^k \triangleq \sum_{j=1}^M w_t^j p_\theta(\tilde{x}_{t+1}^k|\tilde{x}_t^j); \tag{62}$$

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

□

On a historical note, the main ideas underlying the particle filter date back half a century [56, 55]. However, it was not until 1993 that the first working particle filter was discovered by Gordon *et al.* [36].

2.2 Application to Nonlinear Model Estimation

Returning now to the original motivation for discussing particle filters, they can be used to compute the predictor (7) required to compute the PE cost (6) according to

$$\widehat{y}_{t|t-1}(\theta) = \mathbf{E}_\theta\{h(x_t, u_t, \theta) \mid Y_t\} \approx \sum_{i=1}^{M} w(\tilde{x}_t^i) h(\tilde{x}_t^i, u_t, \theta). \tag{63}$$

With this available, it is natural to then turn to the usual gradient based search (35) to then compute a PE estimate (10).

For this purpose, in the scalar signal case and with $\ell(x) = x^2$ chosen, the required gradient $V_N'(\theta_k)$ is given by

$$V_N'(\theta_k) = -\frac{2}{N} \sum_{t=1}^{N} \varepsilon_t(\theta_k) \left(\frac{\mathrm{d}}{\mathrm{d}\theta} \widehat{y}_{t|t-1}(\theta) \Big|_{\theta=\theta_k} \right). \tag{64}$$

This presents a difficulty, since the particle filtering approach does not lend itself to simple computation of the derivative of $\widehat{x}_t(\theta)$ and hence derivative of $\widehat{y}_{t|t-1}(\theta)$ with respect to θ.

One approach to deal with this is to simply numerically evaluate the necessary derivative based on differencing. Another approach is to employ an alternative to the search method (35) that does not require gradient information. There exist several possibilities, such as Nelder–Mead simplex methods or annealing approaches [88, 70].

The following section profiles a further possibility, which while widely used as a non-gradient based search scheme in other fields, is perhaps not commonly appreciated as an option for system identification problems.

3. NONLINEAR MODELS AND THE EXPECTATION MAXIMISATION (EM) ALGORITHM

The Expectation-Maximisation (EM) algorithm is an iterative technique for computing maximum likelihood estimates. It is an alternative to the more commonly used gradient based search. The latter is an approach exploiting smoothness of the associated likelihood function, viewed as a cost to be maximised. The EM algorithm does not exploit this. Rather, it exploits the fact that the likelihood is the logarithm of a probability, which itself has unit area regardless of how it is parametrized.

The approach is widely used in the statistics community, with history stretching back to the founding work [19] in the late 1970's. However, use of the method in specialized forms can be traced back even earlier, such as the famous Baum–Welch method for estimating parameters of discrete valued, discrete state hidden Markov models [9].

The EM algorithm has also been employed in a wide variety of other fields as disparate as image processing and dairy science [12, 18], and there are by now many embellishments and extensions of the method [54]. Nevertheless, while the technique has certainly been applied in system identification settings [39, 34, 73, 33], the method and its properties are perhaps still not widely appreciated.

3.1 Ad-hoc Motivation

The EM algorithm is usually presented in an abstract setting that can be applied quite generally. Unfortunately, this has the potential to obscure the essence of the approach. In an attempt to avoid this, let us instead start with a concrete system identification problem, whereby the parameters of the linear state space model structure (3) are to be estimated.

This is not an open problem. For example, subspace-based estimation algorithms [81, 77] or prediction error approaches [48] can be employed.

Nevertheless, leaving these aside for the moment, one could naïvely observe that if the state x_t were measured, then estimation of the system matrices in (3) could be achieved by simple linear regression.

$$\begin{bmatrix} \widehat{A} & \widehat{B} \\ \widehat{C} & \widehat{D} \end{bmatrix} = \sum_{t=1}^{N} \begin{bmatrix} x_{t+1}x_t^T & x_{t+1}u_t^T \\ y_t x_t^T & y_t u_t^T \end{bmatrix} \left(\begin{bmatrix} x_t x_t^T & x_t u_t^T \\ u_t x_t^T & u_t u_t^T \end{bmatrix} \right)^{-1}. \tag{65}$$

Unfortunately, this is seldom feasible since the state is not measured. However, it can be estimated. For example, a Kalman smoother will deliver an estimate $\widehat{x}_{t|Y_N}$ with associated co-variance $P_{x_t|Y_N}$ that describes the conditional distribution of the state given the observations according to

$$x_t \mid Y_N \sim \mathcal{N}(\widehat{x}_{t|Y_N}, P_{x_t|Y_N}). \tag{66}$$

It is not unnatural to then consider substituting these estimates $\widehat{x}_{t|Y_N}$, $P_{x_t|Y_N} - \widehat{x}_{t|Y_N}\widehat{x}_{t|Y_N}^T$ for x_t, $x_t x_t^T$ (and so on) in (65).

The only difficulty in this seemingly ad-hoc process is that values for the system matrices that are being estimated in (65) are in fact also required to compute the Kalman smoothed estimates (66).

Therefore, some initial estimate of them is necessary. This suggests a further ad-hoc element of iterating, whereby estimates formed via the regression (65) are used to re-estimate the states via (66), which are in turn used back in (65) again.

While each step in this estimation process is not unreasonable, there would seem to be no good reason that it would necessarily deliver a reasonable estimate, or indeed that the iterations involved would necessarily improve the estimate at each step.

In fact, what was just described happens to be one example of the Expectation-Maximisation (EM) algorithm. As a result, via a very simple argument, the iterations just proposed are guaranteed to produce a sequence of non-decreasing likelihoods.

3.2 The EM Algorithm

To explain these properties, and with this concrete example in mind, we now turn to a more general description of the EM approach. The essence of the method is the postulation of a 'missing' data set $X = \{x_{t_1}, \cdots, x_{t_2}\}$. In the example just discussed, this was taken as the state sequence in the model structure (3) with $t_1 = 1, t_2 = N+1$, but other choices are possible, and it can be considered a design variable.

The idea is to then consider the likelihood function

$$L_\theta(X, Y_N) = \log p_\theta(X, Y_N) \tag{67}$$

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

The EM algorithm then copes with X being unavailable by forming an approximation $\mathcal{Q}(\theta, \theta_k)$ of $L_\theta(X, Y_N)$ using a current estimate θ_k of the parameter values. The approximation used is the minimum variance estimate of $L_\theta(X, Y_N)$ given the observed available data Y_N, and this assumption θ_k of the true parameter value. That is

$$\mathcal{Q}(\theta, \theta_k) \triangleq \mathbf{E}_{\theta_k}\{L_\theta(X, Y_N) \mid Y_N\} \tag{68}$$

$$= \int L_\theta(X, Y_N) p_{\theta_k}(X|Y_N)\, \mathrm{d}X. \tag{69}$$

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

$$\log p_\theta(X, Y_N) = \log p_\theta(Y_N) + \log p_\theta(X \mid Y_N). \tag{70}$$

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

$$\mathcal{Q}(\theta, \theta_k) = L_\theta(Y_N) + \int \log p_\theta(X|Y_N) p_{\theta_k}(X|Y_N)\, \mathrm{d}X. \tag{71}$$

Therefore

$$L_\theta(Y_N) - L_{\theta_k}(Y_N) = \mathcal{Q}(\theta, \theta_k) - \mathcal{Q}(\theta_k, \theta_k) + \int \log \frac{p_{\theta_k}(X|Y_N)}{p_\theta(X|Y_N)} p_{\theta_k}(X|Y_N)\, \mathrm{d}X. \tag{72}$$

The rightmost integral in (72) is the Kullback-Leibler divergence metric which is non-negative. This follows directly upon noting that since $-\log x \geq 1 - x$

$$-\int \log \frac{p_\theta(X|Y_N)}{p_{\theta_k}(X|Y_N)} p_{\theta_k}(X|Y_N)\, \mathrm{d}X \geq \tag{73}$$

$$\int \left(1 - \frac{p_\theta(X|Y_N)}{p_{\theta_k}(X|Y_N)}\right) p_{\theta_k}(X|Y_N)\, \mathrm{d}X = 0 \tag{74}$$

where the equality to zero is due to the fact that $p_\theta(X \mid Y_N)$ is of unit area for any value of θ. Consequently

$$L_\theta(Y_N) - L_{\theta_k}(Y_N) \geq \mathcal{Q}(\theta, \theta_k) - \mathcal{Q}(\theta_k, \theta_k). \tag{75}$$

This delivers the key to the EM algorithm. Namely, choosing θ so that $\mathcal{Q}(\theta, \theta_k) > \mathcal{Q}(\theta_k, \theta_k)$ 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 (4) via the following strategy.

Algorithm 3. **(EM Algorithm)**

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

$$\text{Calculate:} \qquad \mathcal{Q}(\theta, \theta_k); \tag{76}$$

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

$$\text{Compute:} \qquad \theta_{k+1} = \arg\max_{\theta \in \Theta} \mathcal{Q}(\theta, \theta_k); \tag{77}$$

(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 decrease of $L_\theta(Y_N)$ or $\mathcal{Q}(\theta, \theta_k)$ falling below a pre-defined threshold [20].

3.3 Return to Motivating Example

To illustrate the application of this abstract formulation, consider again the linear system (3) discussed at the beginning of this section, and note that with the choice $X = \{x_1, \cdots, x_{N+1}\}$ and by Bayes' rule

$$p_\theta(X, Y_N) = p_\theta(x_1) \prod_{t=1}^{N} p_\theta(x_{t+1}, y_t \mid x_t). \tag{78}$$

Furthermore, with the Gaussian assumptions (8)

$$p_\theta(\xi_t|x_t) \sim \mathcal{N}(\Gamma z_t, \Pi), \quad \xi_t \triangleq \begin{bmatrix} x_{t+1} \\ y_t \end{bmatrix} \tag{79}$$

where

$$\Gamma \triangleq \begin{bmatrix} A & B \\ C & D \end{bmatrix}, \quad \Pi \triangleq \begin{bmatrix} Q & S \\ S^T & R \end{bmatrix}, \quad z_t \triangleq \begin{bmatrix} x_t \\ u_t \end{bmatrix}. \tag{80}$$

Therefore, with the further Gaussian assumption $x_1 \sim \mathcal{N}(\mu, P_1)$

$$-2 \log p_\theta(Y_N, X) = \log \det P_1 + (x_1 - \mu)^T P_1^{-1} (x_1 - \mu) + N \log \det \Pi + \sum_{t=1}^{N} (\xi_t - \Gamma z_t)^T \Pi^{-1} (\xi_t - \Gamma z_t). \tag{81}$$

Applying the conditional expectation operator $\mathbf{E}_\theta\{\cdot \mid Y_N\}$ to both sides of this expression then delivers

$$-2\mathcal{Q}(\theta, \theta_k) = \log \det P_1 + \mathrm{Tr}\left\{P_1^{-1} \mathbf{E}_{\theta_k}\{(x_1 - \mu)(x_1 - \mu)^T \mid Y_N\}\right\} + N \log \det \Pi + N\, \mathrm{Tr}\left\{\Pi^{-1}\left[\Phi - \Psi\Gamma^T - \Gamma\Psi^T + \Gamma\Sigma\Gamma^T\right]\right\} \tag{82}$$

where

$$\Phi \triangleq \frac{1}{N} \sum_{t=1}^{N} \mathbf{E}_{\theta_k}\{\xi_t \xi_t^T \mid Y_N\}, \quad \Psi \triangleq \frac{1}{N} \sum_{t=1}^{N} \mathbf{E}_{\theta_k}\{\xi_t z_t^T \mid Y_N\} \tag{83}$$

$$\Sigma \triangleq \frac{1}{N} \sum_{t=1}^{N} \mathbf{E}_{\theta_k}\{z_t z_t^T \mid Y_N\}. \tag{84}$$

Completing the square on the last term in (82) then delivers the maximiser of $\mathcal{Q}(\theta, \theta_k)$ with respect to Γ as

$$\Gamma = \Psi \Sigma^{-1}. \tag{85}$$

This is the value previously argued by applying the ad-hoc motivation of substituting Kalman smoothed estimated in (65). See [29] for a more detailed discussion.

3.4 A nonlinear system example

The problem of estimating nonlinear system models was the motivation in this paper for considering particle filtering methods. Via the consequent need for non-gradient

based optimisation methods, it was also the original motivation for considering the EM algorithm.

Accordingly, this paper now illustrates the potential of the particle filtering and EM methods in this nonlinear system estimation context by considering the following nonlinear system

$$x_{t+1} = ax_t + b\frac{x_t}{1+x_t^2} + c\cos(1.2t) + \nu_t, \quad (86)$$

$$y_t = dx_t^2 + e_t, \quad (87)$$

$$\begin{bmatrix}\nu_t \\ e_t\end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix}0 \\ 0\end{bmatrix}, \begin{bmatrix}\rho & 0 \\ 0 & r\end{bmatrix}\right) \quad (88)$$

where the true parameters in this case are

$$\theta_\circ = [a_\circ, b_\circ, c_\circ, d_\circ, \rho_\circ, r_\circ] = [0.5, 25, 8, 0.05, 0, 0.1]\,. \quad (89)$$

Note that the system (86)-(88) is also time varying. While not explicitly detailed in the interests of clarity, the previous particle filtering and EM-algorithm derivations do not depend on any time invariance, and hence are applicable in this time varying situation. The system (86)-(88) has been chosen due to it being acknowledged as a challenging estimation problem in several previous studies such as [22, 31].

The precise implementation details of how the particle filtering methods just presented are employed to compute the EM algorithm approximation $\mathcal{Q}(\theta, \theta_k)$ to the likelihood L_θ, and how $\mathcal{Q}(\theta, \theta_k)$ is then maximised at each step of the EM method are provided in [71].

The results for 104 different data realisations Y_N of length $N = 100$, with random initialisation of θ_0 in an interval equal to 50% of the the corresponding entry in the true parameter vector $\theta_\circ$, and employing $M = 100$ particles are presented in Table 1.

There, 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

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
ρ	0	$7.78 \times 10^{-5} \pm 7.6 \times 10^{-5}$
r	0.1	0.106 ± 0.015

Table 1. *True and estimated parameter values for the system* (86)-(88)*; mean value and standard deviations are shown for the estimates based on* 104 *Monte-Carlo runs.*

the 104 trials were not included in these calculations due to capture in local minima, which was defined according to the relative error test $|(\widehat{\theta^i} - \theta^i_\circ)/\theta^i_\circ)| > 0.1$ for any i'th component. Considering the random initialisation, this small number of required censorings 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 ρ parameters are to be estimated. In this case, the log-likelihood $L_\theta(Y_N)$ as a function of b with $\rho = \rho^\star = 0$ is shown as the solid line in Figure 1. Clearly the log-likelihood exhibits quite erratic behaviour with very many local maxima. This could reasonably be expected to create significant difficulties for iterative search methods (such as gradient based schemes) for seeking the global maximiser of $L_\theta(Y_N)$.

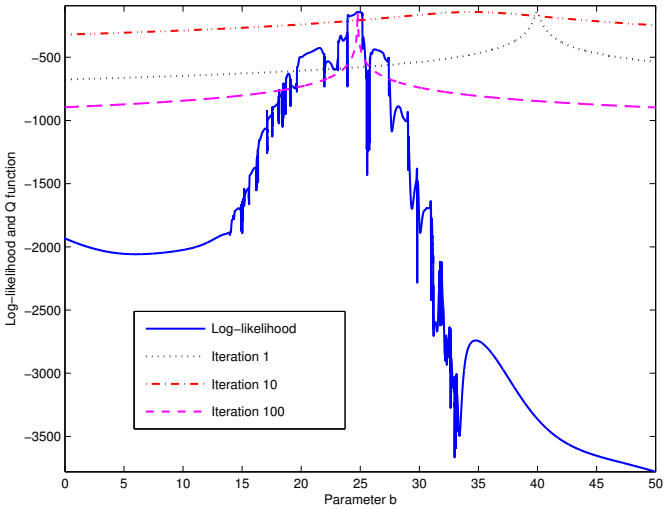


Fig. 1. *The true log-likelihood function is shown as a function of the b parameter. Superimposed onto this plot are three instances of the* $\mathcal{Q}(\theta, \theta_k)$ *function, defined in (68).*

Nevertheless, as illustrated in Table 1, the EM-based method seems quite robust against capture in these local maxima. The means whereby this is achieved are illustrated by profiling the function $\mathcal{Q}(\theta, \theta_k)$ initialised at $[b_0, \rho_0] = [40, 0.001]$ for $k = 1, 10$ and 100 as the dotted, dash-dotted and dashed lines (respectively) in Figure 1.

Clearly, in each case the $\mathcal{Q}(\theta, \theta_k)$ function is a much more straightforward maximisation problem than that of the log likelihood $L_\theta(Y)$. Furthermore, by virtue of the essential property (75), at each iteration directions of increasing $\mathcal{Q}(\theta, \theta_k)$ can be seen to co-incide 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 $\theta_k = [b_k, \rho_k]^T$ for this example are plotted in relation to the two dimensional log-likelihood surface $L_\theta(Y)$ in Figure 2. Clearly, the iterates have taken a path circumventing the highly irregular 'slice' at $\rho = 0$ illustrated in Figure 1. 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 capture in local minima is widely recognised and associated with the EM algorithm in the statistics literature [54]. Within this literature, there are broad explanations for this advantage, such as the fact that (75) implies that $\mathcal{Q}(\theta, \theta_k)$ forms a global approximation to the log likelihood $L_\theta(Y)$ as opposed to the local approximations that are implicit to gradient search based schemes. However, a detailed understanding of this phenomenon is an important open research question deserving further study.

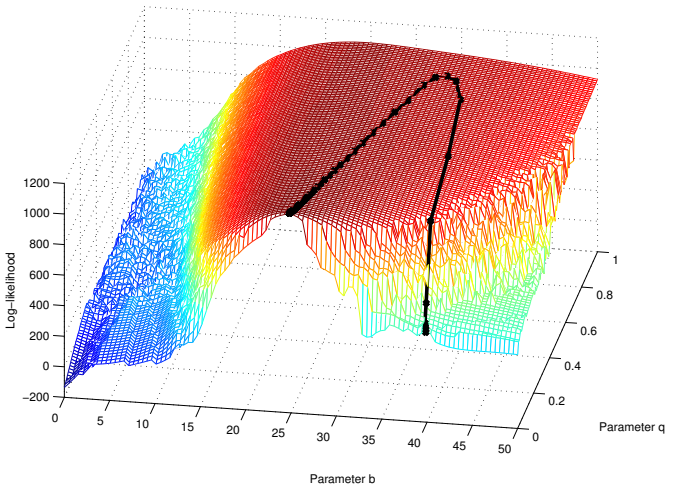


Fig. 2. *The log-likelihood is here plotted as a function of the two parameters b and ρ. Overlaying this are the parameter estimates* $\theta_k = [b_k, \rho_k]^T$ *produced by iterations of the EM algorithm.*

Readers seeking further detail on the application of the EM-algorithm to dynamic system estimation problems may find the papers [73, 39, 34, 29, 30, 87, 33] of interest.

4. HIGHER DIMENSION SYSTEMS AND THE EM ALGORITHM

The paper now turns to a further system identification challenge. Namely, the estimation of models that are of 'high' dimension in that the sizes n_u, n_y of the vector input $\{u_t\}$ and output $\{y_t\}$ in combination with the state dimension n_x imply a large number of parameters to be estimated.

4.1 A Gradient Based Search Example

As a concrete example, consider again the linear state space model (3). On the one hand, estimating the parameters in this model can be straightforwardly and efficiently solved using a subspace-based method [77, 81, 48], (almost) regardless of the dimensions n_u, n_y, n_x.

On the other hand, while there is some theory explaining the asymptotic properties of subspace-based estimates [15, 8, 7], it is arguably less complete than for the PE or ML estimation methods. As a result, quantifying error bounds on delivered subspace based estimates is less straightforward.

Additionally, it is generally held that while subspace-based method provide an excellent initialisation for the optimisation algorithm employed to compute PE or ML estimates, these latter estimates are commonly superior.

For these reasons, it is common to require that a PE or ML estimate be computed. As explained in the introduction, this will require an iterative search of the form (35), which in turn requires the computation of a cost gradient at each iteration. Again, to be concrete, in the PE case with $\ell(x) = x^T x = \|x\|^2$ the gradient $V_N'(\theta)$ given by (64) is required which depends on computing the predictor gradient

$$\frac{\mathrm{d}\widehat{y}_{t|t-1}(\theta)}{\mathrm{d}\theta}. \tag{90}$$

In order to facilitate this, it is common to replace the model structure (3) with its innovations form

$$\begin{bmatrix} x_{t+1} \\ y_t \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} x_t \\ u_t \end{bmatrix} + \begin{bmatrix} K \\ I \end{bmatrix} \epsilon_t. \tag{91}$$

(where $\{\epsilon_t\}$ is an i.i.d. zero mean finite variance process) since this allows the predictor $\widehat{y}_{t|t-1}(\theta)$ to be computed via the steady state Kalman filter

$$\begin{aligned} \widehat{x}_{t+1|t} &= (A - KC)\widehat{x}_{t|t-1} + (B - KD)u_t + Ky_t, \\ \widehat{y}_{t|t-1}(\theta) &= C\widehat{x}_{t|t-1} + Du_t. \end{aligned} \tag{92}$$

As a result, with θ^i denoting the i'th element of θ

$$\frac{\mathrm{d}\widehat{y}_{t|t-1}}{\mathrm{d}\theta^i} = \frac{\mathrm{d}C}{\mathrm{d}\theta^i}\widehat{x}_{t|t-1} + C\frac{\mathrm{d}\widehat{x}_{t|t-1}}{\mathrm{d}\theta^i} + \frac{\mathrm{d}D}{\mathrm{d}\theta^i}u_t \tag{93}$$

where

$$\begin{aligned} \frac{\mathrm{d}\widehat{x}_{t+1|t}}{\mathrm{d}\theta^i} = \frac{\mathrm{d}}{\mathrm{d}\theta^i}(A - KC)\widehat{x}_{t|t-1} + (A - KC)\frac{\mathrm{d}\widehat{x}_{t|t-1}}{\mathrm{d}\theta^i} \\ + \frac{\mathrm{d}}{\mathrm{d}\theta^i}(B - KD)u_t + \frac{\mathrm{d}K}{\mathrm{d}\theta^i}y_t. \end{aligned} \tag{94}$$

Consequently, a separate n_x dimensional filter must be run for each element θ^i of the parameter vector θ.

Subspace-based methods employ a 'full' parametrization in which every element of the system matrices in (91) are included in θ so that

$$\theta \triangleq \left[\mathrm{vec}\{A\}^T, \mathrm{vec}\{B\}^T, \mathrm{vec}\{C\}^T, \mathrm{vec}\{D\}^T, \mathrm{vec}\{K\}^T\right]^T \tag{95}$$

Here, the vec $\{\cdot\}$ operator is one which forms a vector from a matrix by stacking its columns on top of one another.

This full parametrization is an over-parametrization in that θ is of dimension $n_\theta = n_x^2 + n_x(n_u + 2n_y) + n_u n_y$ which is n_x^2 more than any minimal (injective) parametrization of dimension [52]

$$n_\beta = n_\theta - n_x^2 = n_x(n_u + 2n_y) + n_u n_y. \tag{96}$$

As established and developed in [52, 53, 10, 80, 43], such a minimal parametrization, dubbed 'Data Driven Local Co-ordinates' (DDLC) can be computed as a simple affine transformation of the full parametrization (95). Via this, it is straightforward to ensure that the gradient (90) is computed for only the minimum number of components θ^i, $i = 1, 2, \cdots, n_\beta$.

Unfortunately, this may still imply that a very substantial number of filters need to be run to compute (90) via (93), (94).

For example, suppose that the model (91) was to encompass an $n_u = 4$-input, $n_y = 4$-output system, for which the dynamics between any particular input/output pair were third order with no shared pole positions. This would imply a state dimension of $n_x = 3n_u n_y = 48$, and hence $n_\beta = 592$.

That is, for what might be considered a fairly modest 4-input, 4-output 3rd order dynamics situation, gradient based search for a prediction error estimate requires that $n_\beta = 592$ filters with state dimension $n_x = 48$ be run at each iteration. This is a substantial computational load, which will typically lead to long computation times.

4.2 The EM Algorithm Again

Continuing the discussion of this example, if the EM Algorithm were to instead be applied, as explained in section 3.3, the main computation load involves the computation of the $\mathcal{Q}(\theta, \theta_k)$ function given by (82). This dominates the load by way of the Kalman smoother must be run to obtain the quantities in (83), (84).

On the one hand, as opposed to the simple filters (93), (94) required when using gradient-based search, the Kalman smoother is significantly more complicated and computationally costly due to the need to compute the n_x^2 dimension state covariance matrix for each time point. On the other hand, only one smoother needs to be run, as opposed to n_β filters.

To understand these relative computational loads, a detailed audit of the floating point operations (FLOPS) required by the EM algorithm assuming a square-root implementation of the Kalman smoother, leads to a quantification of order $O(n_x^3 N)$ FLOPS per iteration [29].

Equally, since an n_x-state filter with no sparsity in the associated state matrices incurs an $O(n_x^2 N)$ FLOP load, a DDLC parametrized gradient based search incurs an order $O(\max(n_u, n_y) n_x^3 N)$ FLOP load at each iteration [29].

As a result, for multivariable linear systems, there is a computational load advantage of $\max(n_u, n_y)$ FLOPS per iteration in employing the EM algorithm relative to a gradient based search one.

While this advantage may seem modest in this linear system example, it can be much more significant in other situations. For example, suppose the innovations form linear system (91) is extended to represent bilinear systems according to

$$\begin{bmatrix} x_{t+1} \\ y_t \end{bmatrix} = \begin{bmatrix} A & F & B \\ C & G & D \end{bmatrix} \begin{bmatrix} x_t \\ u_t \otimes x_t \\ u_t \end{bmatrix} + \begin{bmatrix} K \\ I \end{bmatrix} \epsilon_t. \tag{97}$$

where the symbol $\otimes$ is the Kronecker tensor product of matrices, which is simply scalar multiplication if either argument is scalar [13].

The importance and utility of these sort of models is in part due to their ability to characterize a very wide range of chemical, biological, robotic and manufacturing processes [24, 79]. It also stems from their utility as approximators, or alternate representations for a range of other nonlinear systems [42, 69].

Importantly the bilinear model structure (97) can equally be represented as a time varying linear structure which is simply (91) but with the system matrices A and C replaced with time varying versions

$$A_t \triangleq A + F(u_t \otimes I_n), \qquad C_t \triangleq C + G(u_t \otimes I_n). \tag{98}$$

As a result, a Kalman smoother for this system is no more computationally difficult or costly than for the linear case (91). Hence, when employing the EM algorithm to estimate the parameters in the bilinear model structure (97), the previous audit value of $O(n_x^3 N)$ FLOPS per iteration still applies [30].

At the same time, as previously explained, a gradient search technique requires one filter per minimal parameter. Furthermore, the number of parameters in the bilinear model (97) is significantly higher than in the linear case (91). As a result, a detailed audit of the requirements of a gradient based search approach to estimating (97) lead to a quantification of $O(N n_x^4 n_u^2 n_y)$ FLOPS per iteration [30].

Compared to the EM algorithm, this is $O(n_x n_u^2 n_y)$ more operations per iteration, which can be significant. For example, in a $n_x = 10$'th order $n_u = n_y = 3$ input/output situation, an EM based approach involves a FLOP requirement per iteration that is less that 1/200'th of that required by the Gauss–Newton approach [30].

Of course, there are many other factors to consider such as computational load, memory requirements, suitability for caching, and of course the number of required iterations.

4.3 Using the $\mathcal{Q}(\theta, \theta_k)$ function to compute gradients of $L_\theta(Y_N)$

To this point, the EM algorithm has been presented as an alternative to gradient based search. In particular, since it does not require the computation of gradients, it was first introduced in this paper as a means for circumventing a problem in nonlinear system estimation.

Namely, while as detailed in section 2, particle-based methods can efficiently compute the predictor $\widehat{y}_{t|t-1}(\theta)$ of a nonlinear system via (63), they do not also straightforwardly provide the gradient $\mathrm{d}\widehat{y}_{t|t-1}(\theta)/\mathrm{d}\theta$.

Perhaps surprisingly, the $\mathcal{Q}(\theta, \theta_k)$ function (68) that is intrinsic to the EM algorithm, has a potential further application as a means for simply and efficiently computing these necessary gradients.

To explore this, note that the gradient of $\mathcal{Q}(\theta, \theta_k)$ may be decomposed according to

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

Evaluating both sides at $\theta = \theta_k$ then delivers

$$\frac{\mathrm{d}}{\mathrm{d}\theta}\mathcal{Q}(\theta, \theta_k)\Big|_{\theta=\theta_k} = \frac{\mathrm{d}}{\mathrm{d}\theta} L_\theta(Y_N)\Big|_{\theta=\theta_k} \tag{99}$$

which is known as Fisher's identity[54]. Therefore, in situations where the formulation of $L_\theta(Y_N)$ makes computing its derivative with respect to θ at some point $\theta = \theta_k$ impossible (as in the nonlinear system/particle filtering case) or computationally intensive (the 'high' dimension linear and bilinear system cases just profiled), a possible solution is to instead compute $\mathcal{Q}(\theta, \theta_k)$ and examine the computation of its gradient at $\theta = \theta_k$.

For example, with regard to the particle filtering/smoothing approach to estimating the nonlinear model structure (1),(2), as established in [71]

$$\frac{\mathrm{d}}{\mathrm{d}\theta}\mathcal{Q}(\theta,\theta_k) = \frac{\mathrm{d}I_1}{\mathrm{d}\theta} + \frac{\mathrm{d}I_2}{\mathrm{d}\theta} + \frac{\mathrm{d}I_3}{\mathrm{d}\theta}, \tag{100}$$

$$\frac{\mathrm{d}I_1}{\mathrm{d}\theta} = \sum_{i=1}^{M} w_{1|N}^i \frac{\mathrm{d}}{\mathrm{d}\theta}\log p_\theta(\tilde{x}_1^i), \tag{101}$$

$$\frac{\mathrm{d}I_2}{\mathrm{d}\theta} = \sum_{t=1}^{N}\sum_{i=1}^{M}\sum_{j=1}^{M} w_{t|N}^{ij} \frac{\mathrm{d}}{\mathrm{d}\theta}\log p_\theta(\tilde{x}_{t+1}^j|\tilde{x}_t^i) \tag{102}$$

$$\frac{\mathrm{d}I_3}{\mathrm{d}\theta} = \sum_{t=1}^{N}\sum_{i=1}^{M} w_{t|N}^{i} \frac{\mathrm{d}}{\mathrm{d}\theta}\log p_\theta(y_t|\tilde{x}_t^i) \tag{103}$$

where the $w_{t|N}^i$, $\tilde{x}_t^i$ terms are computed using the particle smoother Algorithm 2 applied to the model (1),(2).

Since the terms $p_\theta(y_t \mid x_t)$, $p_\theta(x_{t+1} \mid x_t)$ are given by (44) and (45), and hence eminently differentiable, a novel and simply solution for computing the gradient $L'_\theta(\theta_k)$ is therefore provided.

As another example of the use of $\mathcal{Q}'(\theta_k, \theta_k)$ to efficiently compute $L'_{\theta_k}(Y_N)$, recall the expression (82) for $\mathcal{Q}(\theta, \theta_k)$ when applied to the linear state space model (3). Note that by the definitions in (80), all the parameters in the system matrices A, B, C, D are collected in Γ. Hence the derivative of any of these parameters in the corresponding parameter vector θ may be studied by differentiation with respect to Γ.

According to (82), the only term in $\mathcal{Q}(\theta, \theta_k)$ to depend on Γ is the element

$$\mathrm{Tr}\left\{\Pi^{-1}\left[\Phi - \Psi\Gamma^T - \Gamma\Psi^T + \Gamma\Sigma\Gamma^T\right]\right\}. \tag{104}$$

Furthermore [11]

$$\frac{\mathrm{d}}{\mathrm{d}\Gamma}\mathrm{Tr}\{\Psi\Gamma^T\} = \frac{\mathrm{d}}{\mathrm{d}\Gamma}\mathrm{Tr}\{\Gamma\Psi^T\} = \Psi, \tag{105}$$

$$\frac{\mathrm{d}}{\mathrm{d}\Gamma}\mathrm{Tr}\{\Gamma\Sigma\Gamma^T\} = 2\Gamma\Sigma. \tag{106}$$

Therefore, if the m'th element θ^m of θ is the$[\Gamma]_{i,j}$'th element of Γ

$$\frac{\partial}{\partial\theta^m}\mathcal{Q}(\theta,\theta_k) = -N[(\Psi - \Gamma\Sigma)]_{i,j}. \tag{107}$$

As a consequence, if the elements in Γ are set to the same values as were used for the Kalman smoothing operations that delivered Ψ and Σ according to their definition (83),(83), then via (99), the elements in $L'_\theta(Y_N)$ may be obtained by the appropriate element of the right hand side of (99).

As opposed to the more common approach (93),(94), only one Kalman smoother operation is required rather than n_β Kalman filter computations. For large dimension systems, this can be a very significant computational advantage. The advantages of this method were first recognised in [72].

5. ERROR QUANTIFICATION AND MARKOV-CHAIN MONTE–CARLO METHODS

The final system identification challenge that this paper addresses is that of computing error bounds on system parameters θ *and functions of them* that are accurate for finite, and even short data lengths N.

As discussed in section 1.1, the current approach is to employ in the practical finite data length N situation, quantifications that actually apply only in the limit as data length N tends to infinity. This introduces imprecision, as does the need to substitute estimates for required covariance matrices, and rely on linear approximations to nonlinear functions γ of θ if error bounds on $\gamma(\widehat{\theta})$ are required.

In order to circumvent these difficulties, in this final section a Bayesian approach is examined as a means of quantifying the manner in which prior knowledge and data-based information are combined to yield posterior information about system properties.

5.1 A Bayesian Approach

Of course, even the casual reader in the area of estimation is aware of the sometimes passionate debates regarding the utility and philosophical underpinnings of a Bayesian perspective versus a frequentist (likelihood, prediction error) approach [67].

This paper does not seek to contribute or advocate in this unresolved discussion. Nevertheless, this final section does concentrate on proposing the computation of Bayesian statistics, and illustrate how this may be achieved.

Underlying this is the view that the system identification community are great empiricists, and primarily engineers either at heart or by training or both. Hence providing the means for this community to evaluate on real examples the consequences and relative advantages/disadvantages of a Bayesian approach is hoped to be of value and interest.

With these caveats and comments in mind, the posterior $p(\theta \mid Y_N)$ that is proposed as an error quantification may be computed using Bayes' rule according to

$$p(\theta \mid Y_N) = \frac{p(Y_N \mid \theta)p(\theta)}{p(Y_N)}. \tag{108}$$

Here $p(Y_N \mid \theta)$ is the previously considered likelihood, $p(\theta)$ is the a-priori distribution on θ reflecting any prior knowledge about it, and $p(Y_N)$ is a normalizing factor (constant with respect to θ) that simply ensures that $p(\theta \mid Y_N)$ is of unit area with respect to θ and which is formally given by

$$p(Y_N) = \int p(Y_N \mid \theta)p(\theta)\,\mathrm{d}\theta. \tag{109}$$

The computation of the likelihood $p(Y_N \mid \theta)$ has already been addressed in section 1.1. According to (11) it may be expressed in terms of the 'predictor' density via

$$p(Y_N \mid \theta) = p(y_1 \mid \theta)\prod_{t=2}^{N} p(y_t \mid Y_{t-1}, \theta). \tag{110}$$

Note that the notation $p(Y_N \mid \theta)$ used here for the likelihood is different to that of $p_\theta(Y_N)$ employed previously.

This is to reflect that in the Bayesian setting, the parameter vector θ is viewed as a random variable, and not as a fixed quantity.

In the linear or bilinear Gaussian state space cases (3),(97) the density $p(y_t \mid Y_{t-1}, \theta)$ is simply computed (for the linear case) as

$$y_t \mid Y_{t-1} \sim \mathcal{N}\left(C\widehat{x}_{t|t-1} + Du_t, C^T P_{t|t-1} C + R\right) \quad (111)$$

where $\widehat{x}_{t|t-1}$, $P_{t|t-1}$ are the state estimate and associated covariance computed via a standard Kalman filter.

In the linear transfer function case (4), via (9) it is also simple to compute the required conditional density as

$$p(y_t \mid Y_{t-1}, \theta) = p_e(\varepsilon_t(\theta)), \quad (112)$$

$$\varepsilon_t(\theta) = H^{-1}(q)\left[y_t - G(q,\theta)u_t\right]. \quad (113)$$

In these and other examples where $p(Y_N \mid \theta)$ can be computed, it may then be substituted together with a prior $p(\theta)$ into (108) to then obtain the required posterior $p(\theta \mid Y_N)$.

While this does provide a simple solution, unfortunately it is usually the case that it is a function of $p(\theta \mid Y_N)$ that is actually required. For example, if error bounds on a particular parameter θ^i are sought, then this may be provided by the marginal density $p(\theta^i \mid Y_N)$. Unfortunately, this demands the computation of the multidimensional integral

$$p(\theta^i \mid Y_N) = \int p(\theta \mid Y_N) \mathrm{d}\theta^1 \cdots \mathrm{d}\theta^{i-1} \mathrm{d}\theta^{i+1} \cdots \mathrm{d}\theta^n. \quad (114)$$

For all but very simple models with very few parameters, the numerical evaluation of this is infeasible.

Furthermore, it is also common that it is not the parameters θ themselves, but in fact a function $\gamma(\theta)$ of the parameters that is of interest. A simple and common example is the frequency response $G(e^{j\omega}, \theta)$ when employing the transfer function model (4). A less common, but still reasonable example is the phase margin $\phi_m(K)$ achieved for a given closed loop controller K.

In these and other situations it is not at all clear how one might tractably compute the required posterior density $p(\gamma(\theta) \mid Y_N)$.

5.2 A Markov-Chain Monte-Carlo solution

The solution to these difficulties examined here is of the same theme underlying the particle filtering solution to the nonlinear state estimation problem discussed in section 2.1.

Namely, this section examines the strategy of numerically computing required densities by first generating a random sequence of realisations $\{\theta_k\}$ that are distributed according to the posterior density of interest. That is $\theta_k \sim p(\theta \mid Y_N)$.

As in the case of particle filtering, the strong law of large numbers (SLLN), which states that with probability one

$$\lim_{M\to\infty} \frac{1}{M} \sum_{k=1}^{M} \gamma(\theta_k) = \mathbf{E}\left\{\gamma(\theta) \mid Y_N\right\} = \int \gamma(\theta) p(\theta \mid Y_N)\, \mathrm{d}\theta \quad (115)$$

is then employed to deliver the finite M approximation

$$\int \gamma(\theta) p(\theta \mid Y_N)\, \mathrm{d}\theta \approx \frac{1}{M} \sum_{k=1}^{M} \gamma(\theta_k). \quad (116)$$

This then permits the computation of rather arbitrary posterior densities via sample histograms:

$$p(\gamma(\theta) \in A \mid Y_N) \approx \frac{1}{M} \sum_{k=1}^{M} I_{\gamma^{-1}(A)}(\theta_k). \quad (117)$$

Here, with X denoting a set, $I_X(\xi)$ is the indicator function for $\xi \in X$ defined as

$$I_X(\xi) = \begin{cases} 1 \ ; \xi \in X \\ 0 \ ; \text{Otherwise} \end{cases}. \quad (118)$$

Furthermore A is any γ-measurable set, and $\gamma^{-1}(A)$ denotes the pre-image

$$\gamma^{-1}(A) = \{\theta : \gamma(\theta) \in A\}. \quad (119)$$

While, this may seem like a reasonable approach in principle, it may also appear to be practically infeasible due to the implied requirement of a vector random number generator with given joint density $p(\theta \mid Y_N)$.

However, again like the particle filtering case, construction of the required generator is made possible by exploiting the structure of the problem. In this case, the essential ingredient is that via (108),(110) the task of of simply *evaluating* the function $p(\theta \mid Y_N)$ for a given θ is commonly straightforward.

This permits the construction of a Markov-chain $\{\theta_k\}$ which can be established to have limiting density

$$\lim_{k\to\infty} p(\theta_k = \theta \mid \theta_0) = p(\theta \mid Y_N) \qquad \forall \theta_0 \in \mathbf{R}^{n_\theta}. \quad (120)$$

The idea is to then allow for a 'burn in' period in which the above convergence occurs, and then subsequent realisations from the chain can then be used as realisations from $p(\theta \mid Y_N)$.

Again as per the particle filtering case, this vector random generator takes realisations from a convenient density $\lambda(\cdot)$ from which it is straightforward to draw realisations, and then modifies them to provide realisations distributed according to $p(\theta \mid Y_N)$.

The technique used to achieve this is known as the 'Metropolis–Hastings' algorithm, which was developed in [55] and generalized in [38] to the following form.

Algorithm 4. **(Metropolis–Hastings Sampler)**

(1) Initialise θ_0 at some value such that $p(\theta_0 \mid Y_N) > 0$ and set $k = 1$;
(2) At iteration k, consider a candidate value ξ_k for θ_k which is drawn from a ***proposal*** density $\lambda(\xi_k \mid \theta_{k-1})$. That is, find a possible realisation for θ_k as
$$\xi_k \sim \lambda(\cdot \mid \theta_{k-1}); \quad (121)$$
(3) Compute the acceptance probability
$$\alpha(\xi_k \mid \theta_{k-1}) = \min\left\{1, \frac{p(\xi_k \mid Y_N)}{p(\theta_{k-1} \mid Y_N)} \cdot \frac{\lambda(\theta_{k-1} \mid \xi_k)}{\lambda(\xi_k \mid \theta_{k-1})}\right\}; \quad (122)$$
(4) Accept the proposed ξ_k and set $\theta_k = \xi_k$ with probability $\alpha(\xi_k \mid \theta_{k-1})$, otherwise leave θ_k unchanged by setting $\theta_k = \theta_{k-1}$;
(5) Increment k and return to step 2. □

Note that step 4 may be simply implemented by drawing a random variable $z \sim U_{[0,1]}(\cdot)$ from a uniform distribution on $[0,1]$ and setting $\theta_k = \xi_k$ if $z < \alpha(\xi_k \mid \theta_{k-1})$.

This algorithm, with its roots in statistical mechanics [55], is also widely used in statistics, physics, chemistry and biology, as profiled in [40] where it is listed at first place in a survey of 'great algorithms of scientific computing'.

5.3 The proposal density

The only design variable in the Metropolis–Hastings algorithm is the choice of the proposal density $\lambda(\xi \mid \theta)$. Note that the notation is chosen to reflect that the density from which a proposal ξ is drawn may depend on a previous realisation θ.

To appreciate the importance of this, note that perhaps the most natural way to compute a proposed ξ_k at iteration k, is as a perturbation on the previous realisation θ_{k-1} according to

$$\xi_k = \theta_{k-1} + v_k. \tag{123}$$

In the special, but common, case in which the mean of v_k is zero so that the probability density $p_v(\cdot)$ governing v_k is symmetric ($p_v(x) = p_v(-x)$), then clearly

$$\lambda(\xi \mid \theta) = p_v(\xi - \theta) = p_v(\theta - \xi) = \lambda(\theta \mid \xi). \tag{124}$$

As a result

$$\frac{\lambda(\theta_{k-1} \mid \xi_k)}{\lambda(\xi_k \mid \theta_{k-1})} = 1, \tag{125}$$

and hence the acceptance probability (122) simplifies to

$$\alpha(\xi_k \mid \theta_{k-1}) = \min\left\{1, \frac{p(\xi_k \mid Y_N)}{p(\theta_{k-1} \mid Y_N)}\right\}. \tag{126}$$

In this case, Algorithm 4 is known as the *Metropolis Algorithm*.

A further specialization of Algorithm 4 occurs in the situation where only one component θ^i of θ is updated at a time by using the proposal scheme

$$\xi_k = \xi_k^i \cup \theta_{k-1}^{-i} \qquad \xi_k^i \sim p(\theta^i \mid \theta_{k-1}^{-i}, Y_N). \tag{127}$$

Here x^i denotes an i-th sub-block of the vector x, x^{-i} denotes the complement of this (namely everything *except* the i-th sub-block), and the $\cup$ notation denotes forming a new vector via concatenation of the two arguments.

An important consequence of this choice is that a simple calculation using Bayes' rule establishes that the associated acceptance probability $\alpha(\xi_k \mid \theta_{k-1}) = 1$, and hence in this case, the term in the acceptance probability (122) affected by this choice becomes,

$$\frac{p(\xi_k \mid Y_N)}{p(\theta_{k-1} \mid Y_N)} \cdot \frac{\lambda(\theta_{k-1} \mid \xi_k)}{\lambda(\xi_k \mid \theta_{k-1})} = \frac{p(\xi_k^i \mid \xi_k^{-i}, Y_N) p(\xi_k^{-i} \mid Y_N)}{p(\theta_{k-1}^i \mid \theta_{k-1}^{-i}, Y_N) p(\theta_{k-1}^{-i} \mid Y_N)} \cdot \frac{p(\theta_{k-1}^i \mid \theta_{k-1}^{-i}, Y_N)}{p(\xi_k^i \mid \xi_k^{-i}, Y_N)} = 1. \tag{128}$$

In this calculation, the definition of conditional probability $p(A,B) = p(A|B)p(B)$ and the fact that by design $\xi_k^{-i} = \theta_{k-1}^{-i}$ have both been used. This implies that the acceptance probability $\alpha(\xi_k|\ \theta_{k-1})$ in (126) is one, and hence the proposals drawn from the density $p(\xi_k^i \mid \theta_{k-1}^{-i}, Y_N)$ are always retained. In this special case, Algorithm 4 becomes an instance of the *Gibbs sampling* algorithm. Treatment of this method in [27] is often cited as a seminal moment in the birth of the now widespread interest in MCMC methods in the statistics community.

At a more general level, the choice of proposal $\gamma(\xi \mid \theta)$ involves a tradeoff between convergence rate and complexity. For example, as the correlation between the realisations $\{\theta_k\}$ decreases, the convergence rate of the SLLN (115) increases and hence the accuracy of the approximation (116) improves [60].

However, as correlation is minimized, algorithm complexity may increase. At one end of the scale, substituting the choice $\gamma(\xi \mid \theta) = p(\theta \mid Y_N)$ into (122) implies an acceptance probability $\alpha(\xi \mid \theta) = 1$, so that realisations of Algorithm 4 are independent realisations from $p(\theta \mid Y_N)$.

This is clearly infeasible, since the entire premise of the Markov chain Monte–Carlo method is that it is employed because sampling from $p(\theta \mid Y_N)$ is computationally impossible. Therefore, it is necessary to consider suboptimal choices for $\gamma(\xi \mid \theta)$ that are reasonable, while not overly sacrificing convergence rate.

One of the simplest proposal choices is the random walk (123) with $v_k \sim \mathcal{N}(0, \sigma_v^2 I)$, which leaves the variance σ_v^2 as the single design variable. If chosen too small, then almost all proposals will be accepted, and the correlation between samples will be very high. This will manifest in realisations $\{\theta_k\}$ very slowly trawling the range where $p(\theta \mid Y_N)$ is non-zero, resulting in very slow convergence. Vice versa, if σ_v^2 is chosen too high, then overly large jumps in θ_k will be proposed that are rarely in regions where $p(\theta \mid Y_N)$ is significant, and hence rarely accepted. Again, the correlation between samples will be very high (realisations θ_k will remain the same over long periods of rejected proposals) and convergence will be slow.

In consideration of this, the author has found it effective to adaptively modulate σ_v^2 in order for an observed acceptance rate a_L until a given target $\alpha_\star$ is approximately achieved, at which point the adaptation is halted. Here, the rate α_L is defined as the sample average proportion of acceptances over a window of width L (below δ is the Kronecker delta):

$$a_L \triangleq \frac{1}{L} \sum_{k=1}^{L} \delta(\theta_k - \xi_k). \tag{129}$$

Under the strong simplifying assumption of the components θ^i of θ being conditionally (on Y_N) independent, theoretical analysis is available [68] to conclude that $\alpha_\star = 0.234$ provides optimal convergence rate. In practice, the author has found that using $\alpha_\star = 0.3$ gives generally good results.

5.4 Supporting Theory

It is natural to question why the Metropolis–Hastings Algorithm 4 should generate a sequence of realisations $\{\theta_k\}$ that satisfies (120), and (115). To address this, a brief synopsis of the underpinning theory supporting Algorithm 4 will now be provided.

Central to this is the recognition that Algorithm 4, generates a new sample θ_k as a time-homogeneous Markov chain with transition density $K(\theta_k \mid \theta_{k-1})$. Furthermore,

the probability $K(\theta_k \mid \theta_{k-1})$ is determined as the product of the probability $\gamma(\xi \mid \theta)$ of proposing a move ξ, times the probability $\alpha(\xi \mid \theta)$ of accepting it:

$$K(\theta_k = \xi_k \mid \theta_{k-1}) = \alpha(\xi_k \mid \theta_{k-1})\gamma(\xi_k \mid \theta_{k-1}) I_{\mathcal{X}_{\theta_{k-1}}}(\xi_k) + \delta(\xi_k - \theta_{k-1}) r(\theta_{k-1}) \quad (130)$$

where $\mathcal{X}_{\theta_{k-1}} = \{\xi : \xi \neq \theta_{k-1}\}$ and

$$r(\theta_{k-1}) = 1 - \int_{\mathcal{X}_{\theta_{k-1}}} \alpha(\xi \mid \theta_{k-1})\gamma(\xi \mid \theta_{k-1})\, \mathrm{d}\xi$$

is the probability of no change in the value of θ_k from one iteration to another. In (130) the delta function is of the Dirac type.

Suppose now that θ_{k-1} is distributed according to the pdf $\pi_{k-1}(\theta)$. Then clearly, the pdf $\pi_k(\theta)$ for an ensuing element θ_k in this Markov chain is given by the law of total probability via

$$\pi_k(\theta_k) = \int K(\theta_k \mid \theta_{k-1})\pi_{k-1}(\theta_{k-1})\, \mathrm{d}\theta_{k-1}. \quad (131)$$

Therefore, if the realisations $\{\theta_k\}$ generated by Algorithm 4 are to converge in a distributional sense to realisations having some constant density $\pi(\theta)$, then that density must satisfy

$$\pi(\theta) = \int K(\theta \mid \xi)\pi(\xi)\, \mathrm{d}\xi \quad (132)$$

in which case $\pi(\theta)$ is termed [57] an *invariant* (or stationary) density with respect to the transition kernel $K(\theta_k \mid \theta_{k-1})$.

The essence of the design of the design of the Metropolis–Hastings Algorithm 4 is that the acceptance probability $\alpha(\xi \mid \theta)$ is crafted so that (132) is achieved for $\pi(\theta) = p(\theta \mid Y_N)$.

To establish this, consider first the almost ubiquitous situation of $\xi \neq \theta$ in which case (subscripts will be dropped momentarily to enhance readability)

$$\begin{aligned} p(\theta \mid Y)K(\xi \mid \theta) &= p(\theta \mid Y)\gamma(\xi \mid \theta) \times \\ &\quad \min\left\{1, \frac{p(\xi \mid Y)}{p(\theta \mid Y)} \cdot \frac{\gamma(\theta \mid \xi)}{\gamma(\xi \mid \theta)}\right\} \\ &= \min\left\{p(\theta \mid Y)\gamma(\xi \mid \theta), p(\xi \mid Y)\gamma(\theta \mid \xi)\right\}. \end{aligned} \quad (133)$$

Similarly,

$$\begin{aligned} p(\xi \mid Y)K(\theta \mid \xi) &= p(\xi \mid Y)\gamma(\theta \mid \xi) \times \\ &\quad \min\left\{1, \frac{p(\theta \mid Y)}{p(\xi \mid Y)} \cdot \frac{\gamma(\xi \mid \theta)}{\gamma(\theta \mid \xi)}\right\} \\ &= \min\left\{p(\xi \mid Y)\gamma(\theta \mid \xi), p(\theta \mid Y)\gamma(\xi \mid \theta)\right\}. \end{aligned} \quad (134)$$

Therefore, comparing (133) and (134) and noting that the $\min\{\cdot,\cdot\}$ operation is symmetric implies that algorithm 4 yields a Markov chain for which the so-called 'reversibility condition'

$$p(\theta \mid Y)K(\xi \mid \theta) = p(\xi \mid Y)K(\theta \mid \xi) \quad (135)$$

holds when $\xi \neq \theta$. Similarly, considering now the case of $\xi = \theta$, then (135) trivially holds simply by substitution of $\xi = \theta$ into the definition (135) of reversibility.

Therefore, (135) holds for all possible transitions. Substituting $p(\cdot \mid Y)$ for $\pi(\cdot)$ into the right hand side of (132) and using (135) then implies

$$\begin{aligned} \int K(\theta \mid \xi)p(\xi \mid Y)\, \mathrm{d}\xi &= \int K(\xi \mid \theta)p(\theta \mid Y)\, \mathrm{d}\xi \\ &= p(\theta \mid Y) \int K(\xi \mid \theta)\, \mathrm{d}\xi \\ &= p(\theta \mid Y) \end{aligned} \quad (136)$$

where the transition to the last line follows since $K(\xi \mid \theta)$ is a probability density function and hence integrates to one.

Therefore, the desired posterior density $p(\theta \mid Y)$ is a candidate for any density that realisations of Algorithm 4 might converge to. This line of thinking was the original basis for the design of the algorithm[55].

To establish further properties, such as $p(\theta \mid Y_N)$ being the only density satisfying the invariance property (132), that having satisfied this necessary condition for the distributional convergence (120), it actually does occur, and that the SLLN result (115) holds all requires much more analysis than can be described here.

Indeed, the basic theory underpinning these convergence results was established only relatively recently in the seminal work on the topic [75], which depends itself on relatively recent results in the theory of Markov chains on uncountable spaces [57].

Fortunately, the results of this analysis are that under the mild conditions that the support of the proposal $\gamma(\xi \mid \theta)$ is at all times larger than the support of the posterior $p(\theta \mid Y_N)$ which itself is a connected set, and that both of these densities are bounded, then all the desired convergence properties just mentioned can be established [61].

5.5 An example

To illustrate the application of these ideas, this section considers the case of a linear and time invariant system model of the output error (OE) form

$$y_t = G(q,\theta)u_t + e_t, \qquad G(q,\theta) = \frac{B(q,\theta)}{A(q,\theta)}, \quad (137)$$

where

$$A(q,\theta) = 1 + a_1 q^{-1} + a_2 q^{-2} + \cdots + a_{m_a} q^{-m_a}, \quad (138)$$

$$B(q,\theta) = b_0 + b_1 q^{-1} + b_2 q^{-2} + \cdots + b_{m_b} q^{-m_b}, \quad (139)$$

$$\theta^T = [a_1, \cdots, a_{m_a}, b_0, \cdots, b_{m_b}]. \quad (140)$$

In this case, to evaluate the likelihood (12) the required predictor is given by (9) as simply $\widehat{y}_{t|t-1}(\theta) = G(q,\theta)u_t$ so that via (5) and (12) the posterior $p(\theta \mid Y_N)$ can be evaluated for any value of θ as

$$p(\theta \mid Y_N) = k \cdot p(\theta) \prod_{t=1}^{N} p_e\left(y_t - G(q,\theta)u_t\right). \quad (141)$$

Here k is a constant independent of θ, that will not be required an any subsequent calculations (since it will cancel), but is included in (141) to ensure it has unit total probability.

Furthermore, in the example to follow, e_t of variance $\mathrm{Var}\{e_t\} = \sigma^2$ will have a uniform distribution $e_t \sim \mathcal{U}[-1.5\sigma^{2/3}, 1.5\sigma^{2/3}]$ so that $p_e(\cdot)$ will be the indicator function $I_{[-1.5\sigma^{2/3}, 1.5\sigma^{2/3}]}(\cdot)$ and hence the product term

in (141) will be either one or zero. Similarly, in what is to follow, the prior $p(\theta)$ will be uniform and hence zero or not.

A further specialization in the example presented in this section is the employment (in the MCMC Algorithm 4) of the random walk proposal (123)

$$\xi_k = \theta_{k-1} + v_k, \qquad v_k \sim \mathcal{N}(0, \sigma_v^2 I) \tag{142}$$

whereby the previous iteration θ_{k-1} is perturbed by a Gaussian distributed random amount v_k. Recall, that as explained via (124), in this situation the acceptance probability $\alpha(\xi \mid \theta)$ simplifies to the Metropolis form (126). As a result, in the example we present here, the MCMC Algorithm 4 is implemented as follows.

Algorithm 5. **(MCMC for OE Model)**

(1) Initialise θ_0 at some value such that according to (141) the probability $p(\theta_0 \mid Y_N) > 0$, and set $k = 1$;
(2) At iteration k, generate a candidate value ξ_k computed according to the random walk proposal (142);
(3) Substitute the ξ_k obtained in step 2 and the θ_{k-1} from the previous iteration into (141) in order to compute the acceptance probability; viz.

$$\alpha(\xi_k \mid \theta_{k-1}) = \min\left\{1, \frac{p(\xi_k \mid Y_N)}{p(\theta_{k-1} \mid Y_N)}\right\}; \tag{143}$$

(4) Generate a realisation $z \sim \mathcal{U}[0,1]$, where $\mathcal{U}[0,1]$ represents a uniform distribution on the interval $[0,1]$.
(5) Set $\theta_k = \xi_k$ if

$$z < \alpha(\xi_k \mid \theta_{k-1}), \tag{144}$$

otherwise set $\theta_k = \theta_{k-1}$.
(6) Increment k and return to step 2.

□

Within this setting, the case studied here is the simplest possible first order one of

$$m_a = 1, \quad m_b = 0, \quad a_1 = -0.8, \quad b_0 = 0.2 \tag{145}$$

with $\{e_t\}$ a zero mean i.i.d. uniformly distributed process of variance $\sigma^2 = \mathbf{E}\{e_t^2\} = 0.01$. It is then supposed that the available data from this system consists of only $N = 20$ samples of $\{y_t\}$ and $\{u_t\}$ being a sampled step response transiting $1 \mapsto 0$ at the data record midpoint. This is illustrated in Figure 3, where the solid line is the noise free response, and the samples around this line are the noise corrupted data assumed to be available.

In the case where the density $p_e(\cdot)$ governing e_t is uniform, and with prior distribution on $\theta = [a_1, b_0]$ being one that assigns zero weight to $b_0 < 0$ and $|a_1| > 1$, then the posterior distributions for these parameters given the data realisation shown in Figure 3 are illustrated in Figure 4.

There, the solid line shows the marginal posterior density for b_0 and a_1 computed via Algorithm 4 with the random walk proposal (123) using perturbations $\{v_k\}$ which are i.i.d. zero mean Gaussian with variance tuned to deliver an empirical acceptance rate $\alpha_L \approx 0.3$. These marginals were produced from histograms based on 10^5 iterations of Algorithm 4, which were then smoothed using standard kernel density estimation methods [74].

By way of comparison, the marginals obtained by numerical integration using Simpson's rule over 220 bins to evaluate (30) by 'brute force' are shown as a dashed line in Figure 4. They are virtually indistinguishable from the solid line, indicating the accuracy of the MCMC approach in this example.

As further comparison, the error quantification (17) associated with a PE estimate obtained from the data in Figure 3 and using the asymptotic in N approximation (17), (32),(33) is shown as the dash-dot Gaussian curves in Figure 4.

While these quantifications are not strictly comparable to the posterior densities, since they evaluate different quantities, it would still seem interesting to compare the two in terms of their utility in informing a user of what system information can be extracted from the available data, particularly in view of the very widespread use of the PE method and (asymptotic based) associated error quantification.

In relation to this, note that since $p_e(\cdot)$ is uniform in this example, then via (12) the likelihood $p_\theta(Y_N)$ is constant on the region Θ defined by

$$\Theta = \bigcap_{t=1}^{N} \left\{\theta : y_t - \widehat{y}_{t|t-1}(\theta) \in \Delta\right\}. \tag{146}$$

and hence has no uniquely defined maximum, so that a ML estimator does not exist and hence cannot be profiled.

This domain Θ is precisely the 'feasible parameter set' that is studied (albeit via different motivation) in the 'bounded error' estimation literature, and for which (depending on model structure) numerous efficient methods for computation have been developed [65, 64, 58, 83]. Since when intersected with the support of the prior $p(\theta)$, this is also the support of $p(\theta \mid Y_N)$ there are clear links between the two approaches.

To illustrate further potential uses of the Markov chain methods proposed here, suppose that it is necessary to design a closed loop PI controller $K(q)$ for the system responsible for the observations in Figure 3 and under the hypothesis that the system is first order.

The choice

$$K(q) = 2 + \frac{0.1}{q-1} \tag{147}$$

achieves a phase margin $\phi_m = 99.3°$ and gain margin $g_m = 5.56$ on the afore-mentioned PE estimate.

However, it is of course important to gauge the likely performance of the controller (147) on the real system. As argued in this paper, a Bayesian approach addresses this question by computing the posteriors

$$p(\phi_m \mid Y_N), \qquad p(g_m \mid Y_N). \tag{148}$$

Due to the implicit way in which ϕ_m and g_m are defined (i.e. they do not obey a closed form formula) this would be a daunting (if not impossible) task if approached from an analytical point of view, or via standard asymptotic approximation methods such as (34).

In contrast, it is straightforward to compute the required marginals (148) using the Monte–Carlo approach of this paper. Each realisation of $\{\theta_k\}$ provided by Algorithm 4, implies an associated ϕ_m and g_m which is straightforward to compute. Furthermore, they may each be thought of as

arising from a bounded mapping $\gamma : \theta \to \mathbf{R}$ so that the approach (117) can be employed.

The results of this strategy are shown in Figure 5. Their accuracy is ensured by the demonstrated accuracy of the distribution of the marginals of $p(\theta \mid Y_N)$ in Figure 4. Clearly, there seems to be good evidence from the data that the the controller (147) will achieve a phase margin greater than 95° and a gain margin greater than 3.7.

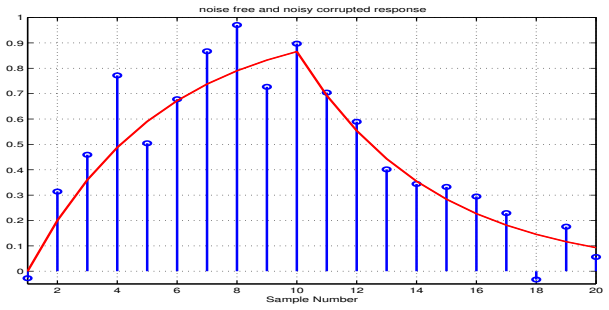
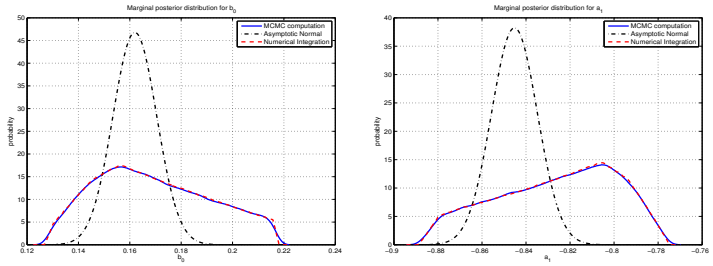
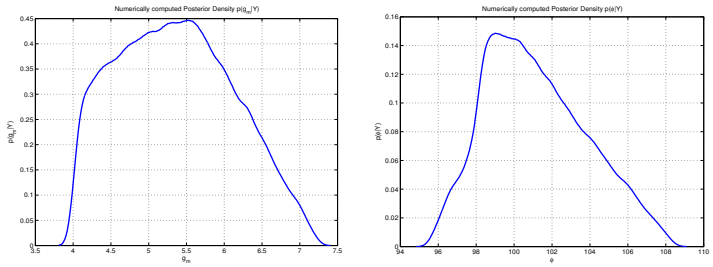


Fig. 3. *First order system response: Solid line is noise free, sampled dots are the noise corrupted measurements available.*



(a) Posterior $p(b_0 \mid Y_N)$ (b) Posterior $p(a_1 \mid Y_N)$

Fig. 4. *Posterior marginal densities of parameters computed via Algorithm 4 shown as a solid line together with (dashed line) another evaluation of the posterior marginals computed via numerical integration of the joint posterior, and (dash-dot line), the parameter information that would be inferred from the data via a PE approach with the asymptotic in N distribution approximation derived from* (17).



(a) Gain Margin Posterior (b) Phase Margin Posterior

Fig. 5. *Posterior distributions for phase margin ϕ_m and gain margin g_m for a given PI controller.*

6. CONCLUSION

This paper has profiled some selected and acknowledged open system identification problems, and profiled some potential avenues of attack that are based on ideas from other fields, such as signal processing (particle filtering), statistics (EM algorithm) and statistical mechanics (Metropolis–Hastings algorithm).

In considering only a selection of areas, the author is very conscious of having ignored many areas of very active interest such as experiment design, identification for control, frequency domain identification, continuous time identification, errors in variables problems and many more. This exclusion is only due to space constraints and, more importantly, lack of expertise by the author on these topics.

The existence of so many open areas and the vigor by which they are pursued is a testament to the vitality and relevance of system identification research. This has, and continues to involve the adoption and development of ideas from very many fields. Indeed, it is probably not unfair to observe that to work in the system identification field requires expertise in probability theory, time series analysis, statistics, systems theory, linear algebra, functional analysis and numerical optimization algorithms as a minimum requirement.

The new(ish) approaches surveyed in this paper can therefore be considered just another part of accepted system identification practice whereby new ideas from other fields are evaluated, developed, and employed if appropriate. While the author believes that the methods proposed here have promise, it may well be the case that other techniques prove superior.

Whatever the situation, if history is a reliable guide (the prediction error method we regularly rely on seeks to minimise discrepencies from it!), then system identification researchers will successfully develop effective solutions to the open problems surveyed here, and these will involve an open-minded strategy of understanding and adapting effective techniques from other fields.

Acknowledgments

The material in this paper is the result of very many hours of work, discussion, and coffee breaks with colleagues. While the author accepts responsibility for errors or obfuscation in the current paper, whatever is useful is due to a collective effort.

In particular, the ideas and expertise of Adrian Wills, Thomas Schön, Håkan Hjalmarsson, Tomas McKelvey, Stuart Gibson and Soren Henriksen are central to this paper.

REFERENCES

[1] *Frequency Domain System Identification Toolbox for MATLAB.* `http://elecwww.vub.ac.be/fdident/index.html`.

[2] A. Abdulle and G. Wanner. 200 years of the least squares method. *Elemente der Mathematik*, 57:45–60, 2002.

[3] B.D.O. Anderson and J.B. Moore. *Optimal Filtering.* Prentice Hall, 1979.

[4] K.J. Åström. Maximum likelihood and prediction error methods. *Automatica*, 16:551–574, 1980.

[5] K.J. Åström and Peter Eykhoff. System identification - a survey. *Automatica*, 7:123–162, 1971.

[6] K.L. Åström and T. Bohlin. Numerical identification of linear dynamic systems from normal operating records. In *Proceedings of the IFAC Symposium on Self-Adaptive Systems, Teddington, UK*, pages 96–111, 1965.

[7] D. Bauer, M. Deistler, and W. Scherrer. Consistency and asymptotic Normality of some subspace algorithms for systems without observed inputs. *Automatica*, 35:1243–1254, 1999.

[8] Dietmar Bauer and Magnus Jansson. Analysis of the asymptotic properties of the moesp type of subspace algorithms. *Automatica*, 36(4):497–509, April 2000.

[9] L.E. Baum, T. Petrie, G. Soules, and N. Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *Annals of Mathematical Statistics*, 41:164–171, 1970.

[10] Niek H. Bergboer, Vincent Verdult, and Michel H.G. Verhaegen. An efficient implementation of Maximum Likelihood identification of LTI state-space models by local gradient search. In *Proceedings of the 41st IEEE CDC, Las Vegas, USA*, December 2002.

[11] Dennis S. Bernstein. *Matrix Mathematics*. Princeton University Press, 2005.

[12] M.J. Borran and B Aazhang. EM-based multiuser detection in fast fading multipath environments. In *Proc. EURASIP*, volume 8, pages 787–796, 2002.

[13] John W. Brewer. Kronecker products and matrix calculus in system theory. *IEEE Transactions on Circuits and Systems*, 25(9):772–781, September 1978.

[14] P.E. Caines. *Linear Stochastic Systems*. John Wiley and Sons, New York, 1988.

[15] A. Chiuso and G. Picci. Asymptotic variance of subspace estimates. *Journal of Econometrics*, 118(1-2):257–291, 2004.

[16] K.L. Chung. *A Course in Probability Theory*. Harcourt, Brace and World Inc., 1968.

[17] Rev.Mr.Bayes (communicated by Mr.Price). An essay towards solving a problem in the doctrine of chances. *Philosophical Transactions of the Royal Society of London*, 53:370–418, 1763.

[18] Matthew Crouse, Robert Nowak, and Richard Baraniuk. Wavelet-based statistical signal processing using hidden Markov models. *IEEE Transactions on Signal Processing*, 46(4):886–902, April 1998.

[19] 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.

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

[21] J.L. Doob. *Stochastic Processes*. John Wiley and Sons, London, 1953.

[22] 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.

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

[24] Wouter Favoreel, Bart De Moor, and Peter Van Overschee. Subspace identification of bilinear systems subject to white inputs. *IEEE Trans. Automat. Control*, 44(6):1157–1165, 1999.

[25] R.A. Fisher. On an absolute criterion for fitting frequency curves. *Mess. Math.*, 41:155, 1912.

[26] R.A. Fisher. On the mathematical foundations of theoretical statistics. *Philosophical Transactions of the Royal Society, London, Series A*, (222):309–368, 1922.

[27] A.E. Gelfand and A.F.M Smith. Sampling based approaches to calculating marginal densities. *Journal of the American Statistical Association*, 85:398–409, 1990.

[28] Michel Gevers. A personal view of the development of system identification. *IEEE Control Systems Magazine*, pages 93–105, December 2006.

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

[30] Stuart Gibson, Adrian Wills, and Brett Ninness. Maximum-likelihood parameter estimation of bilinear systems. *IEEE Transactions on Automatic Control*, 50(10):1581–1596, 2005.

[31] 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.

[32] I. Goethals, K. Pelckmans, J. A. K. Suykens, and B. De Moor. Supspace Identification of Hammerstein Systems Using Least Squares Support Vector Machines. *IEEE Transactions on Automatic Control*, 50(10):pp1509–1519, 2005.

[33] 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.

[34] G.C. Goodwin and A. Feuer. Estimation with missing data. *Mathematical and Computer Modelling of Dynamical Systems*, 5(3):220–244, 1999.

[35] G.C. Goodwin and R.L. Payne. *Dynamic System Identification*. Academic Press, 1977.

[36] 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.

[37] E.J. Hannan and Manfred Deistler. *The Statistical Theory of Linear Systems*. John Wiley and Sons, New York, 1988.

[38] W.K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57:97–109, 1970.

[39] Alf J. Isaksson. Identification of ARX-models subject to missing data. *IEEE Trans. Automat. Control*, 38(5):813–819, 1993.

[40] J.Dongarra and F.Sullivan (eds). The top ten algorithms - The Metropolis algorithm. *Computing in Science and Engineering*, 2(1):65–69, 2000.

[41] A.N. Kolmogorov. Interpolation und extrapolation von stationären zufälligen folgen. *Bulletin de*

l'académie des Sciences de U.S.S.R., 5:3–14, 1941.

[42] Arthur J. Krener. Bilinear and nonlinear realizations of input-output maps. *SIAM J. Control*, 13:827–834, 1975.

[43] L. H. Lee and K. Poolla. Identification of linear parameter-varying systems using nonlinear programming. *Journal of Dynamic Systems, Management, and Control*, 121:71–78, March 1999.

[44] E.L. Lehmann. *Theory of Point Estimation*. John Wiley & Sons, 1983.

[45] L. Ljung and A. Vicino (Ed). Special issue on system identification: Linear vs nonlinear. *IEEE Transaction on Automatic Control*, 2005.

[46] L. Ljung, G. Goodwin, J. Skoukens, D. Westwick, K. Keesman, and Hong Zhao. Experiences and challenges of nonlinear systems. Panel Discussion Session, at 14th IFAC Symposium on System Identification, 2006.

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

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

[49] Lennart Ljung. *MATLAB System Identification Toolbox Users Guide, Version 6*. The Mathworks, 2004.

[50] L.Ljung. Convergence analysis of parametric identification methods. *IEEE Transactions on Automatic Control*, AC-23(5):770–783, 1978.

[51] L.Ljung and P.E.Caines. Asymptotic Normality of prediction error estimators for approximate system models. *Stochastics*, 3:29–46, 1979.

[52] T. McKelvey, A. Helmersson, and T. Ribarits. Data driven local coordinates for multivariable linear systems and their application to system identification. *Automatica*, 40:1629–1635, 2004.

[53] Tomas McKelvey and Anders Helmersson. A dynamical minimal parametrization of multivariable linear systems and its application to optimization and system identification. In *Proc. of the 14th World Congress of IFAC*, volume H, pages 7–12, Beijing, P. R. China, 1999.

[54] 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.

[55] 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.

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

[57] S. P. Meyn and R. L. Tweedie. *Markov chains and stochastic stability*. Springer-Verlag, London, 1993.

[58] M. Milanese and A. Vicino. Optimal inner bounds of feasible parameter set in linear estimation with bounded noise. *IEEE Transactions on Automatic Control*, 36(6):759, 1991.

[59] M. Milanese and A. Vicino. Information based complexity and nonparametric worst-case system identification. *Journal of Complexity*, 9(4):427–446, December 1993.

[60] Brett Ninness. Strong laws of large numbers under weak assumptions with applications. *IEEE Transactions on Automatic Control*, 45(11):2117–2122, November 2000.

[61] Brett Ninness and Soren Henriksen. Bayesian system identification via markov chain monte carlo techniques. *Provisionally accepted as a full paper, Automatica*, 2009.

[62] Brett Ninness and Adrian Wills. An identification toolbox for profiling novel techniques. In *14th IFAC Symposium on System Identification*, pages 301–307, mar 2006.

[63] John P. Norton. *An Introduction to Identification*. Academic Press, 1985.

[64] J.P. Norton. Identification and application of bounded parameter models. *Automatica*, 23:497–507, 1987.

[65] J.P. Norton. Identification of parameter bounds of ARMAX models from records with bounded noises. *International Journal of Control*, 42:375–390, 1987.

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

[67] Christian P. Robert. *The Bayesian Choice*. Springer Verlag, 2 edition, 2001.

[68] Gareth O. Roberts and Jeffrey S. Rosenthal. Optimal scaling for various Metropolis–Hastings algorithms. *Statistical Science*, 16(4):351–367, November 2001.

[69] Wilson J. Rugh. *Nonlinear System Theory: The Volterra/Wiener Approach*. The Johns Hopkins University Press, Baltimore, MD, 1981.

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

[71] Thomas Schön, Adrian Wills, and Brett Ninness. System identification of nonlinear state space models. *Submitted to Automatica*, 2009.

[72] Mordechai Segal and Ehud Weinstein. A new method for evaluating the log-likelihood gradient (score) of linear dynamic systems. *IEEE Transactions on Automatic Control*, 33(8):763–766, 1988.

[73] R. H. Shumway and D. S. Stoffer. An approach to time series smoothing and forecasting using the EM algorithm. *Journal of time series analysis*, 3(4):253–264, 1982.

[74] B. W. Silverman. *Density estimation for statistics and data analysis*. Monographs on Statistics and Applied Probability. Chapman & Hall, London, 1986.

[75] Luke Tierney. Markov chains for exploring posterior distributions. *Ann. Statist.*, 22(4):1701–1762, 1994. With discussion and a rejoinder by the author.

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

[77] Peter van Overschee and Bart de Moor. N4SID:Subspace algortihms for the identification of combined deterministic-stochastic systems. *Automatica*, 30(1):75–93, 1994.

[78] Sergio Verdu. *Multiuser Detection*. Cambridge University Press, 1998.

[79] V. Verdult and M. Verhaegen. Subspace identification of MIMO bilinear systems. In *Proceedings of the European Control Conference*, Karlsruhe, Germany, 1999.

[80] Vincent Verdult, Niek Bergboer, and Michel Verghaegen. Maximum Likelihood identification of multivari-

able bilinear state-space systems by projected gradient search. In *Proceedings of the 41st IEEE CDC, Las Vegas, USA*, December 2002.

[81] M. Verhaegen. Identification of the deterministic part of MIMO state space models in innovations form from input-output data. *Automatica*, 30(1):61–74, January 1994.

[82] Michel Verhaegen and Vincent Verdult. *Filtering and System Identification*. Cambridge University Press, 2007.

[83] E. Walter and H.Piet-Lahanier. Exact recursive polyhedral description of the feasible parameter set for bounded-error models. *IEEE Transactions on Automatic Control*, AC-34:911–914, 1989.

[84] Norbert Wiener. *Extrapolation, Interpolation and Smoothing of Stationary Time Series*. M.I.T. Press, 1949.

[85] Adrian Wills, Adam Mills, and Brett Ninness. A MATLAB software environment for system identification. In *Proceeding of the 15th IFAC Symposium on System Identification*, 2009.

[86] Adrian Wills and Brett Ninness. On gradient-based search for multivariable system estimates. *IEEE Trans. Automat. Control*, 53(1):298–306, 2008.

[87] Adrian Wills, Brett Ninness, and Stuart Gibson. Maximum likelihood estimation of state space models from frequency domain data. *IEEE Transactions on Automatic Control*, 54(1):19–33, 2009.

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