Blind Identification of Wiener Models ⋆

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

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

Abstract: This paper develops and illustrates methods for the identification of Wiener model structures. These techniques are capable of accommodating the "blind" situation where the input excitation to the linear block is not observed. Furthermore, the algorithm developed here can accommodate a nonlinearity which need not be invertible, and may also be multivariable. Central to these developments is the employment of the Expectation Maximisation (EM) method for computing maximum likelihood estimates, and the use of a new approach to particle smoothing to efficiently compute stochastic expectations in the presence of nonlinearities.

Keywords: Wiener Model, Nonlinear Systems, Maximum Likelihood, System Identification, Parameter Estimation.

1. INTRODUCTION

A common framework for nonlinear models of dynamical systems is to work with combinations of memoryless nonlinear blocks and linear dynamic blocks. This is known as *block-oriented models*. The most common of these are *Hammerstein*, *Wiener*, *Hammerstein-Wiener* and *Wiener-Hammerstein* models. They correspond to a cascade combination of one or two linear dynamic models and one or two memoryless nonlinear blocks, e.g. Ljung (1999). Also more general, feedback variants have been discussed, e.g. by Schoukens et al. (2003) and Hsu et al. (2006).

Perhaps the most powerful and interesting example of simple block-oriented models is the Wiener model illustrated in Figure 1. The literature on estimation of Wiener models has been extensive; among many references we may mention Bai (2002); Westwick and Verhaegen (1996); Wigren (1993); Zhu (1999). It may be noted that most of the references deal with special cases of the general configuration in Figure 1. Typical restrictions imposed in prior work include: 1) The nonlinearity f is invertible; 2) The measurement noise e_t is absent. The first restriction excludes many common nonlinearities, such as dead-zones and saturations, and the second does not reflect most practical applications. These and other issues are discussed in detail in Hagenblad et al. (2008); Wills and Ljung (2010).

In many cases it may happen that the input w_t is not available for measurement. This is the case e.g. for vibration analysis (Peeters and Roeck, 2001). To be able to handle that, some assumption about the character of w must be introduced, typically that it is a stationary Gaussian stochastic process with unknown, but rational, spectral density. Merging such an input with the process noise gives a structure depicted in Figure 1. Formally we then have a case of *time-series modeling* with the output described as a stationary stochastic process with a spectral density to be determined, followed by a memoryless nonlinearity. In recent years, the term *blind identification*, cf Abed-Meraim et al. (1997), has been used for this situation, in analogy with "blind equalization", and also perhaps to stress that w in Figure 1 is a "physical signal", it is just that it cannot be measured.

While some of the general techniques for the estimation of Wiener models could be applied to the "blind identification of Wiener models" (like the one in Wills and Ljung (2010)), relatively few papers directly dealing with this problem have been published. An exception is the recent contribution by Vanbaylen et al. (2009), which defines a maximum likelihood method. It is however subject to the restriction (1) mentioned above.

The objective with the current paper is to give a treatment of blind identification of the Wiener model in Figure 1 allowing for nose and nonlinearities that may be multivariable and non-invertible.

2. PROBLEM FORMULATION

This paper addresses the problem of identifying the unknown parameter vector θ that specifies a Wiener model structure illustrated in Figure 1. Importantly, the paper considers the "blind" estimation problem in which only an output record $Y_N \triangleq \{y_1, \ldots, y_N\}$ is available, and the input w_t is not measured.

For this purpose, the following model structure is employed

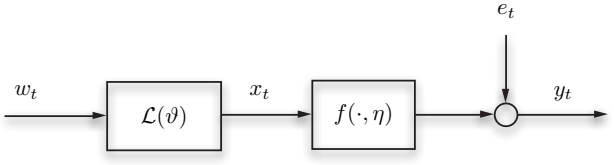

Fig. 1. *The Wiener model structure. A linear time-invariant system $\mathcal{L}$ followed by a memoryless nonlinearity f.*

⋆ This work was supported by: the Australian Research Council through their Discovery Project Program; and CADICS, a Linneaus Center funded by the Swedish Research Council and the project project Calibrating Nonlinear Dynamical Models (Contract number: 621-2010-5876) also funded by THE Swedish Research Council.

$$\xi_{t+1} = A\xi_t + w_t, \tag{1a}$$
$$x_t = C\xi_t, \tag{1b}$$
$$y_t = f(x_t, \eta) + e_t. \tag{1c}$$

Here $\xi_t \in \mathbb{R}^{n_\xi}$ is the state vector of a linear system $\mathcal{L}(\vartheta)$ driven by a Gaussian i.i.d. process $w_t \sim \mathcal{N}(0, Q)$. The system matrices $A \in \mathbb{R}^{n_\xi \times n_\xi}$, $C \in \mathbb{R}^{n_x \times n_\xi}$ and $Q \in \mathbb{R}^{n_\xi \times n_\xi}$ are fully parametrized (no elements are fixed) by the vector

$$\vartheta = [\mathrm{vec}\{A\}^T, \mathrm{vec}\{C\}^T, \mathrm{vec}\{Q\}]^T \tag{2}$$

where $\mathrm{vec}\{\cdot\}$ generates a vector from a matrix by stacking its columns on top of one another.

The memoryless nonlinearity component $f(x_t, \eta) : \mathbb{R}^{n_x} \to \mathbb{R}^{n_y}$ is parametrized by the vector η. This mapping and its parametrization may be quite general. Common situations such as basis function expansions (polynomials, splines, neural networks), piecewise linear expansions, or specific types such as dead-zones or saturations are all allowed.

Finally, the measurement noise e_t is a Gaussian i.i.d. process modeled as $e_t \sim \mathcal{N}(0, R)$ which is independent of w_t. The full Wiener model is then described by the parameter vector

$$\theta = [\vartheta^T, \eta^T, \mathrm{vec}\{R\}^T]^T. \tag{3}$$

In what follows, it will be useful to note that the Wiener model (1) may also be represented by the following probabilistic description

$$\xi_{t+1} \sim p(\xi_{t+1} \mid \xi_t) = p_w(\xi_{t+1} - A\xi_t), \tag{4a}$$
$$y_t \sim p(y_t \mid \xi_t) = p_e(y_t - f(C\xi_t, \eta)). \tag{4b}$$

This paper examines the formation of an estimate $\widehat{\theta}$ of θ via the maximum likelihood (ML) approach

$$\widehat{\theta} = \operatorname*{argmax}_\theta L_\theta(Y_N), \quad L_\theta(Y_N) \triangleq \log p_\theta(Y_N) \tag{5}$$

Here $p_\theta(Y_N)$ denotes the joint density of the measurements Y_N and via subscript makes explicit that according to the model (1) it will depend upon θ, and likewise for $L_\theta(Y_N)$. Note that since the logarithm is a monotonic function, the maximiser $\widehat{\theta}$ of the "log likelihood" $L_\theta(Y_N)$ is also the maximiser of the "likelihood" $p_\theta(Y_N)$.

The log-likelihood can by repeated use of Bayes' rule be written as

$$L_\theta(Y_N) = \sum_{t=1}^{N} \log p_\theta(y_t \mid Y_{t-1}), \quad p_\theta(y_1 \mid Y_0) \triangleq p_\theta(y_1) \tag{6}$$

which reduces the problem of computing $L_\theta(Y_N)$ to that of computing the prediction density $p_\theta(y_t \mid Y_{t-1})$. In the linear and time invariant case this can be simply computed using a Kalman filter.

However, due to the memoryless nonlinearity component $f(\cdot, \eta)$, computing $p_\theta(y_t \mid Y_{t-1})$ is far less straightforward for the Wiener model structure (1). One strategy that might suggest itself would be to use sequential importance resampling, more colloquially known as "particle filtering" (Doucet and Johansen, 2011) to numerically approximate the prediction density.

However, as discussed in Schön et al. (2011), this approach delivers an approximation $L_\theta(Y_N)$ which is not a continuous, much less differentiable, function of θ, which makes the problem of computing its maximiser $\widehat{\theta}$ intractable.

To address these difficulties, this paper takes the same approach as in Schön et al. (2011), and employs the expectation-maximisation (EM) algorithm (Dempster et al., 1977; McLachlan and Krishnan, 2008) to compute the maximiser $\widehat{\theta}$, since this approach avoids the need to directly compute $L_\theta(Y_N)$ or its derivatives. Sequential importance sampling methods are still employed, but critically this is by way of using particle *smoothers* as opposed to particle filters.

3. THE EXPECTATION MAXIMISATION (EM) ALGORITHM

The EM algorithm is a method for computing the ML estimate $\widehat{\theta}$ that is very general and addresses a wide range of applications. Key to both its implementation and theoretical underpinnings is the consideration of a joint log-likelihood function of both the measurements Y_N and "missing data" Z

$$L_\theta(Y_N, Z) \triangleq \log p_\theta(Y_N, Z). \tag{7}$$

The missing data Z consist of measurements that while not available, would be useful to the estimation problem. The choice of Z is a design variable in the deployment of the EM algorithm.

Importantly, the log-likelihood $\log p_\theta(Y_N)$ and the joint log likelihood $\log p_\theta(Y_N, Z)$ are related by the definition of conditional probability according to

$$\log p_\theta(Y_N) = \log p_\theta(Z, Y_N) - \log p_\theta(Z \mid Y_N). \tag{8}$$

Let θ_k denote an estimate of the likelihood maximiser $\widehat{\theta}$. Further, denote by $p_{\theta_k}(Z \mid Y_N)$ the conditional density of the missing data Z, given observations of the available data Y_N and depending on the choice θ_k.

These definitions allow the following expression, which is obtained by taking conditional expectations of both sides of (8) relative to $p_{\theta_k}(Z \mid Y_N)$.

$$\begin{aligned}\log\, p_\theta(Y_N) &= \int \log p_\theta(Z, Y_N) p_{\theta_k}(Z \mid Y_N) dZ \\ &- \int \log p_\theta(Z \mid Y_N) p_{\theta_k}(Z \mid Y_N) dZ \\ &= \underbrace{\mathrm{E}_{\theta_k}\left\{\log p_\theta(Z, Y_N) \mid Y_N\right\}}_{\triangleq \mathcal{Q}(\theta, \theta_k)} - \underbrace{\mathrm{E}_{\theta_k}\left\{\log p_\theta(Z \mid Y_N) \mid Y_N\right\}}_{\triangleq \mathcal{V}(\theta, \theta_k)}.\end{aligned} \tag{9}$$

Employing these newly defined $\mathcal{Q}$ and $\mathcal{V}$ functions, we can express the difference between the likelihood $L_\theta(Y_N) = \log p_\theta(Y_N)$ at an arbitrary value of θ and the likelihood $L_{\theta_k}(Y_N)$ at the estimate θ_k as

$$\begin{aligned}L_\theta(Y_N) - L_{\theta_k}(Y_N) &= (\mathcal{Q}(\theta, \theta_k) - \mathcal{Q}(\theta_k, \theta_k)) \\ &+ \underbrace{(\mathcal{V}(\theta_k, \theta_k) - \mathcal{V}(\theta, \theta_k))}_{\geq 0}.\end{aligned} \tag{10}$$

The positivity of the last term in the above equation can be established by noting that it is the Kullback–Leibler divergence metric between two densities (Gibson and Ninness, 2005). As a consequence if we obtain a new estimate θ_{k+1} such that $\mathcal{Q}(\theta_{k+1}, \theta_k) > \mathcal{Q}(\theta_k, \theta_k)$, then it follows that $L_{\theta_{k+1}}(Y_N) > L_{\theta_k}(Y_N)$. So that, by increasing the $\mathcal{Q}$ function we are also increasing the likelihood $L_\theta(Y_N)$.

This leads to the EM algorithm, which iterates between forming $\mathcal{Q}(\theta, \theta_k)$ and then maximising it with respect to θ to obtain a better estimate θ_{k+1}.

The Expectation and Maximisation steps are quite involved for the Wiener model considered in this work and

Algorithm 1 : Expectation Maximisation Algorithm

(1) Set $k = 0$ and initialize θ_0 such that $L_{\theta_0}(Y_N)$ is finite.
(2) **Expectation (E) step:** Compute
$$\mathcal{Q}(\theta, \theta_k) = \mathrm{E}_{\theta_k}\left\{\log p_\theta(Z, Y_N) \mid Y_N\right\}. \tag{11}$$
(3) **Maximisation (M) step:** Compute
$$\theta_{k+1} = \arg\max_\theta \mathcal{Q}(\theta, \theta_k). \tag{12}$$
(4) If not converged, update $k := k+1$ and return to step 2.

are therefor treated separately in Sections 4.1 and 4.2 below. The algorithm in its general form was first presented by Dempster et al. (1977), and we refer the reader to McLachlan and Krishnan (2008) as an excellent reference work on the method.

4. THE EM ALGORITHM FOR WIENER MODEL STRUCTURES

As mentioned earlier, the specification of the missing data Z is the principle design variable when employing the EM algorithm. In this work, it is taken as the record for $t \in [1, N]$ of the underlying state vector ξ_t in the model structure (1). That is
$$Z = \{\xi_1, \ldots, \xi_N\}. \tag{13}$$
Applying the EM algorithm then reduces to the consideration of how the E-step is computed (how $\mathcal{Q}(\theta, \theta_k)$ can be computed) and how the M-step is achieved (how the maximiser of $\mathcal{Q}(\theta, \theta_k)$ with respect to θ is calculated).

4.1 Expectation Step

The starting point for addressing the computation of $\mathcal{Q}(\theta, \theta_k)$ is its definition (9) for which the following decomposition is useful, and which is achieved by using Bayes' rule and the Markov property of the model structure (1).
$$\begin{aligned} L_\theta(Y_N, Z) &= \log p_\theta(Y_N|Z) + \log p_\theta(Z) \\ &= \sum_{t=1}^{N-1} \log p_\theta(\xi_{t+1}|\xi_t) + \sum_{t=1}^{N} \log p_\theta(y_t|\xi_t). \end{aligned} \tag{14}$$
Application of the conditional expectation operator $\mathrm{E}_{\theta_k}\{\cdot \mid Y_N\}$ to both sides of (14) then yields
$$\mathcal{Q}(\theta, \theta_k) = I_1 + I_2, \tag{15}$$
where
$$I_1 = \sum_{t=1}^{N-1} \int\int \log p_\theta(\xi_{t+1}|\xi_t)\, p_{\theta_k}(\xi_{t+1}, \xi_t|Y_N)\, \mathrm{d}\xi_t\, \mathrm{d}\xi_{t+1}, \tag{16a}$$
$$I_2 = \sum_{t=1}^{N} \int \log p_\theta(y_t|\xi_t)\, p_{\theta_k}(\xi_t|Y_N)\, \mathrm{d}\xi_t. \tag{16b}$$
In the Gaussian case, and in the absence of the memoryless nonlinearity $f(\cdot, \eta)$, the associated densities would be Gaussian, and they and the associated expectations could be computed by an optimal linear smoother (Gibson and Ninness, 2005).

In contrast, with $f(\cdot, \eta)$ present the situation is much less straightforward, and it seems the exact computation of the above expectations is intractable.

To address this difficulty, this paper will employ sequential importance resampling (SIR) methods, which are more colloquially known as "particle" techniques. Underpinning these approaches, is the central idea of generating a user chosen number M of random realisations (particles) ξ_t^i, $i = 1, \ldots, M$ from the smoothing density of interest $\xi_t^i \sim p(\xi_t \mid Y_N)$.

These realisations are then used to form the following approximation to multi-dimensional integrals that may involve an arbitrary (integrable) function $g(\cdot)$
$$\frac{1}{M}\sum_{i=1}^{M} g(\xi_t^i) \approx \int g(\xi_t)\, p(\xi_t \mid Y_N)\, \mathrm{d}\xi_t. \tag{17}$$
As the user chosen number of particles M tends to infinity, the approximation in (17) tends to equality with probability one, and hence the quality of approximation for finite M improves as M grows (Hu et al., 2008).

Generating random realisations from the smoothing density requires a preceding step of generating realisations ζ_t^i for $i = 1, \ldots, M$ from the *filtering* density $p(\xi_t \mid Y_t)$. The following algorithm for achieving this has now become a benchmark, although there are many variants on it (Doucet et al., 2001; Arulampalam et al., 2002; Ristic et al., 2004).

Algorithm 2 Particle Filter

1: Initialize particles, $\{\zeta_0^i\}_{i=1}^M \sim p_\theta(\zeta_0)$ and set $t = 1$;
2: Predict the particles by drawing M i.i.d. samples according to
$$\tilde{\zeta}_t^i \sim p_\theta(\tilde{\zeta}_t|\zeta_{t-1}^i), \qquad i = 1, \ldots, M. \tag{18}$$
3: Compute the importance weights $\{w_t^i\}_{i=1}^M$,
$$w_t^i \triangleq w(\tilde{\zeta}_t^i) = \frac{p_\theta(y_t|\tilde{\zeta}_t^i)}{\sum_{j=1}^M p_\theta(y_t|\tilde{\zeta}_t^j)}, \qquad i = 1, \ldots, M. \tag{19}$$
4: For each $j = 1, \ldots, M$ draw a new particle ζ_t^j with replacement (resample) according to,
$$\mathrm{P}(\zeta_t^j = \tilde{\zeta}_t^i) = w_t^i, \qquad i = 1, \ldots, M. \tag{20}$$
5: If $t < N$ increment $t \mapsto t+1$ and return to step 2, otherwise terminate.

The development of particle smoothing methods is much less mature. However, the recent work by Douc et al. (2010) has developed a new approach that is both computationally efficient, and has the great advantage of generating realisations from the complete *joint* smoothing density $p(\xi_1, \ldots, \xi_N \mid Y_N)$.

Central to this new work is the use of what is known as "rejection sampling" in order to generate realisations from a certain "target" density, which as established in Douc et al. (2010) should be taken as $p(\xi_{t+1} \mid \xi_t)$. Via the model (1), this may be expressed as
$$p(\xi_{t+1} \mid \xi_t) = (2\pi)^{-1/2}|Q|^{-n_\xi/2} f(\xi_{t+1}, \xi_t, \theta), \tag{21}$$
where
$$f(\xi_{t+1}, \xi_t, \theta) \triangleq \exp\left(-\frac{1}{2}(\xi_{t+1} - A\xi_t)^T Q^{-1}(\xi_{t+1} - A\xi_t)\right). \tag{22}$$
This latter function is then central to the following Algorithm 3 which is a rejection sampling based particle smoother, and a variant on the approach developed by Douc et al. (2010).

The realisations ξ_t^i generated by Algorithm 3 from the joint smoothing density $p(\xi_1, \ldots, \xi_N \mid Y_N)$ may then be used

Algorithm 3 Rejection Sampling Based Particle Smoother

1: Run the particle filter (Algorithm 2) and store all the generated particles ζ_t^i for $t = 1, \ldots, N$ and $i = 1, \ldots, M$;
2: Set $t = N$ and initialize the smoothed particles $\xi_N^i = \zeta_N^i$ for $i = 1, \ldots, M$;
3: **for** $i = 1 : M$ **do**
4: Draw an integer j randomly according to $j \sim \mathcal{U}([1, \ldots, M])$ where the later is the uniform distribution over the integers $1, \ldots, M$;
5: Draw a real number τ randomly according to $\tau \sim U([0, 1])$ where the latter is the uniform distribution over the real numbers in the interval $[0, 1]$;
6: **if** $\tau < f(\xi_t^i, \zeta_{t-1}^j, \theta)$ **then**
7: return to step 4;
8: **end if**
9: Set $\xi_{t-1}^i = \zeta_{t-1}^j$.
10: **end for**
11: **if** $t > 1$ **then**
12: Decrement $t \mapsto t - 1$. Return to step 4
13: **else**
14: Terminate;
15: **end if**

to approximately compute the integrals I_1 and I_2 given in (16) as follows,

$$I_1 \approx \widehat{I}_1 = \frac{1}{M} \sum_{t=1}^{N-1} \sum_{i=1}^{M} \log p_\theta(\xi_{t+1}^i \mid \xi_t^i), \tag{23a}$$

$$I_2 \approx \widehat{I}_2 = \frac{1}{M} \sum_{t=1}^{N} \sum_{i=1}^{M} \log p_\theta(y_t | \xi_t^i). \tag{23b}$$

4.2 Maximisation Step

As mentioned above, the second step of the EM algorithm, called the M-step, involves maximising the $Q(\theta, \theta_k)$ function over θ. Note that according to Section 4.1 it is not tractable to work with the true $\mathcal{Q}$ function. Rather, as outlined above in Section 4.1, this paper considers an approximation obtained using particle methods, which is given by (23a)–(23b) and restated here for reference

$$\widehat{\mathcal{Q}}(\theta, \theta_k) = \widehat{I}_1(A, Q) + \widehat{I}_2(C, \eta, R). \tag{24}$$

Since these two terms are parametrized independently, then maximising $\widehat{\mathcal{Q}}$ can be achieved by independently maximising $\widehat{I}_1$ over A and Q, and $\widehat{I}_2$ over C, η and R.

Maximising $\widehat{I}_1$: according to (1a) and (4a) and the fact that w_t is Gaussian distributed via $w_t \sim \mathcal{N}(0, Q)$, the term $\widehat{I}_1$ can be expressed as

$$\begin{aligned}
\widehat{I}_1(A, Q) &= \frac{1}{M} \sum_{t=1}^{N-1} \sum_{i=1}^{M} \log p_\theta(\xi_{t+1}^i \mid \xi_t^i) \\
&= \kappa - \frac{(N-1)M}{2M} \log\det(Q) \\
&\quad - \frac{1}{2M} \sum_{t=1}^{N-1} \sum_{i=1}^{M} (\xi_{t+1}^i - A\xi_t^i)^T Q^{-1} (\xi_{t+1}^i - A\xi_t^i) \\
&= \kappa - \frac{(N-1)}{2} \log\det(Q) \\
&\quad - \frac{1}{2} \mathrm{Trace}\left\{ Q^{-1} (\Phi - \Psi A^T - A\Psi^T + A\Sigma A^T) \right\},
\end{aligned} \tag{25}$$

where κ is a constant term, and

$$\Phi \triangleq \frac{1}{M} \sum_{t=1}^{N-1} \sum_{i=1}^{M} \xi_{t+1}^i (\xi_{t+1}^i)^T, \tag{26a}$$

$$\Psi \triangleq \frac{1}{M} \sum_{t=1}^{N-1} \sum_{i=1}^{M} \xi_{t+1}^i (\xi_t^i)^T, \tag{26b}$$

$$\Sigma \triangleq \frac{1}{M} \sum_{t=1}^{N-1} \sum_{i=1}^{M} \xi_t^i (\xi_t^i)^T. \tag{26c}$$

It can be shown that (26c) can be maximised by the following choices for A and Q

$$A = \Psi\Sigma^{-1}, \qquad Q = \frac{1}{N-1} \left[\Phi - \Psi\Sigma^{-1}\Psi^T \right]. \tag{27}$$

Maximising $\widehat{I}_2$: again, according to (1c) and (4b) and using the assumption that $e_t \sim \mathcal{N}(0, R)$, the second term $\widehat{I}_2$ can be expressed as

$$\begin{aligned}
\widehat{I}_2(C, \eta, R) &= \frac{1}{M} \sum_{t=1}^{N} \sum_{i=1}^{M} \log p_\theta(y_t \mid \xi_t^i) \\
&= \gamma - \frac{NM}{2M} \log\det(R) \\
&\quad - \frac{1}{2M} \sum_{t=1}^{N} \sum_{i=1}^{M} (y_t - f(C\xi_t^i, \eta))^T R^{-1} (y_t - f(C\xi_t^i, \eta)) \\
&= \gamma - \frac{N}{2} \log\det(R) - \frac{1}{2} \mathrm{Trace}\left\{ R^{-1} F(C, \eta) \right\},
\end{aligned} \tag{28}$$

where γ is a constant and

$$F(C, \eta) \triangleq \frac{1}{M} \sum_{t=1}^{N} \sum_{i=1}^{M} (y_t - f(C\xi_t^i, \eta))(y_t - f(C\xi_t^i, \eta))^T. \tag{29}$$

The choice

$$R = \frac{1}{N} F(C, \eta) \tag{30}$$

maximises $\widehat{I}_2(C, \eta, R)$ over R. Inserting this solution into $\widehat{I}_2$ provides

$$\widehat{I}_2(C, \eta) = \gamma - \frac{N}{2} \log\det(\frac{1}{N} F(C, \eta)) - \frac{1}{2} \mathrm{Trace}\left\{ NI \right\} \tag{31}$$

Hence, it remains to maximise $\widehat{I}_2$ in (31) over C and η. The function $\widehat{I}_2$ is nonlinear in the parameters and, in general, it is not feasible to obtain an analytical solution for its maximum. Hence, we employ the standard approach of maximising $\widehat{I}_2$ via a gradient based search, which is now outlined. It is convenient to define a joint parameter vector

$$\beta = [\mathrm{vec}\left\{C\right\}^T \;\; \eta^T]^T. \tag{32}$$

The gradient based search approach updates the parameter β via

$$\beta \leftarrow \beta + \alpha\rho. \tag{33}$$

Here the vector ρ is given by the Gauss-Newton search direction (Dennis and Schnabel, 1983) defined as

$$\rho = H(\beta)^{-1} g(\beta), \tag{34}$$

where the j'th element of the gradient vector g is given by

$$g_j(\beta) \triangleq \frac{\partial \widehat{I}_2(\beta)}{\partial \beta_j} = -N \sum_{i=1}^{M} r_t^i(\beta)^T F^{-1}(\beta) \frac{\partial r_t^i(\beta)}{\partial \beta_j}, \tag{35a}$$

$$r_t^i(\beta) \triangleq y_t - f(C\xi_t^i, \eta). \tag{35b}$$

and the (p,m)'th element of the scaling matrix H is given by

$$H_{(p,m)}(\beta) = N \sum_{i=1}^{M} \frac{\partial r_t^i(\beta)^T}{\partial \beta_p} F^{-1}(\beta) \frac{\partial r_t^i(\beta)}{\partial \beta_m}. \tag{36}$$

Based on this choice for ρ, it can be shown that there exists an $\alpha > 0$ so that $\widehat{I}_2(\beta + \alpha g(\beta)) > \widehat{I}_2(\beta)$, which we achieve using a backstepping line search in this paper.

In terms of computing the gradient vector g and the scaling H, it is necessary to compute the derivatives

$$\frac{\partial r_t^i(\beta)}{\partial \beta_j} = -\frac{\partial f(C\xi_t^i, \eta)}{\partial \beta_j}. \tag{37}$$

For the C parameters it follows via the chain rule that

$$\frac{\partial f(C\xi_t^i, \eta)}{\partial \mathrm{vec}\,\{C\}_j} = \left.\frac{\partial f(x,\eta)}{\partial x}\right|_{x=C\xi_t^i} \frac{\partial C\xi_t^i}{\partial \mathrm{vec}\,\{C\}_j}. \tag{38}$$

The derivatives of the first term on the right hand side of (38) will be case dependent and it is difficult to say anything in general. This is also the case for the η parameters. A summary of the M-step is provided in Algorithm 4.

Algorithm 4 M-step

Given the current parameter values θ_k and a positive scalar ϵ, perform the following:

1: Compute A and Q via (27), (26a), (26b) and (26c);
2: Initialise $\beta = [\mathrm{vec}\,\{C_k\}\ \eta_k^T]^T$.
3: **while** $\|g(\beta)\| < \epsilon$ **do**
4: Compute $\rho = H(\beta)^{-1} g(\beta)$;
5: Set $\alpha = 1$;
6: **while** $\widehat{I}_2(\beta + \alpha\rho) < \widehat{I}_2(\beta)$ **do**
7: Update $\alpha \leftarrow \alpha/2$;
8: **end while**
9: Set $\beta \leftarrow \beta + \alpha\rho$;
10: **end while**
11: Set $[\mathrm{vec}\left\{C_{k+1}\right\}\ \eta_{k+1}^T]^T = \beta$.
12: Compute R_{k+1} via (30), using the new estimates C_{k+1} and η_{k+1} just obtained.

5. SIMULATION EXAMPLE

Here we consider a Wiener system with two measurements, as depicted in Figure 2. The linear dynamic block is a 2'nd order resonant system, whose transfer function H is given by

$$H(q) = \frac{q^{-1} + 0.3q^{-2}}{1 - q^{-1} + 0.9q^{-2}}. \tag{39a}$$

The true nonlinearities f_1 and f_2 are given by a saturation function and a deadzone function, respectively.

$$f_1(x) = \begin{cases} \eta_1 & : x > \eta_1 \\ x & : \eta_2 \le x \le \eta_1 \\ \eta_2 & : x < \eta_2 \end{cases} \tag{39b}$$

$$f_2(x) = \begin{cases} x - \eta_3 & : x > \eta_3 \\ 0 & : \eta_3 \le x \le \eta_4 \\ x - \eta_4 & : x < \eta_4 \end{cases} \tag{39c}$$

where the true parameter values are given by

$$\eta_1 = 0.1, \quad \eta_2 = -1.3, \quad \eta_3 = 0.8, \quad \eta_4 = -0.2. \tag{39d}$$

In terms of the estimation model structure, we used a 2'nd order model for the linear dynamic system and the nonlinearities were modeled as a saturation and deadzone, respectively.

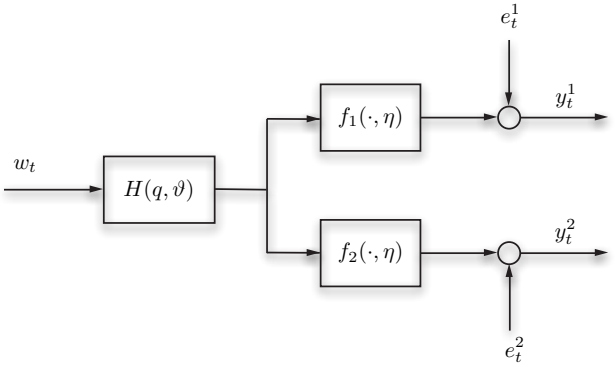

Fig. 2. Block diagram of blind Wiener model with two outputs.

For the purposes of estimation, $N = 1000$ samples of the outputs were simulated using (39) with the noise source $w_t \sim \mathcal{N}(0,1)$. The measurements were corrupted by Gaussian noise $e_t \sim \mathcal{N}(0, 0.1 \times I_2)$.

The initial values for $\widehat{\eta}$ were chosen as $\widehat{\eta}_i = \frac{\eta_i}{10}$ to reflect that they are unknown. The parameters for the linear dynamic block were initialised by estimating a 2'nd order state-space model using a subspace algorithm based on the measurements $\{y_1^2, \ldots, y_N^2\}$ from the deadzone nonlinearity.

Using the above initial parameter values, the EM method was employed to provide ML estimates based on $M = 100$ particles. The EM algorithm was terminated after just 100 iterations. The results of 100 Monte Carlo runs are shown in Figures 3–5. For each run, different noise realisations were used according to the distributions specified above.

These figures demonstrate the utility of the proposed algorithm in that the estimates appear to be good, even though the initial estimates are clearly far from accurate. These results are encouraging, especially given the very modest number of $M = 100$ of particles employed.

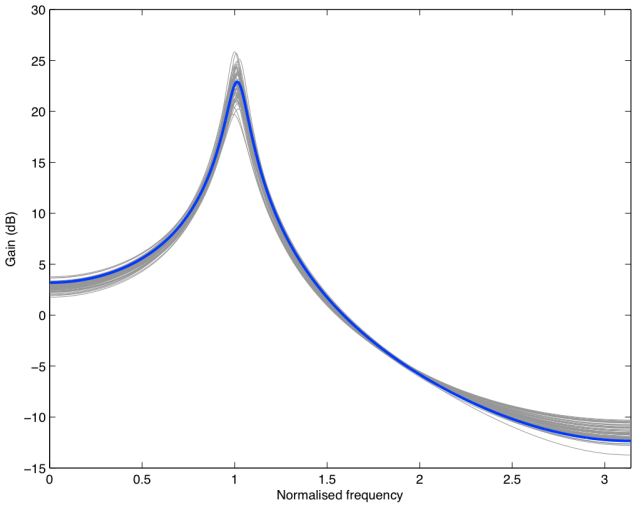

Fig. 3. Bode plot of estimated (grey) and true (blue) systems for the example studied in Section 5.

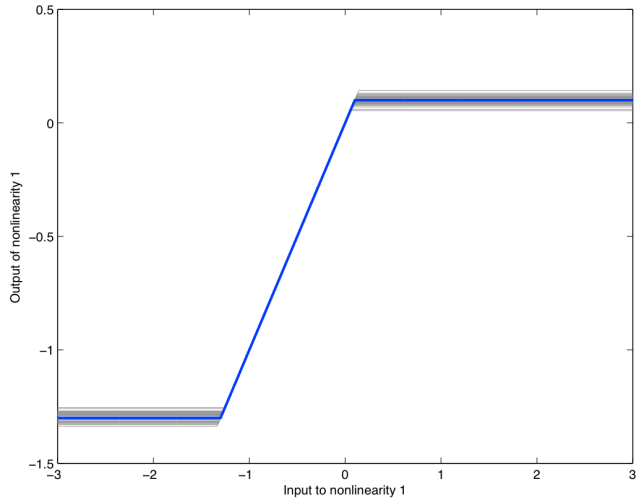

Fig. 4. Estimated (grey) and true (blue) memoryless nonlinearities for the example studied in Section 5.

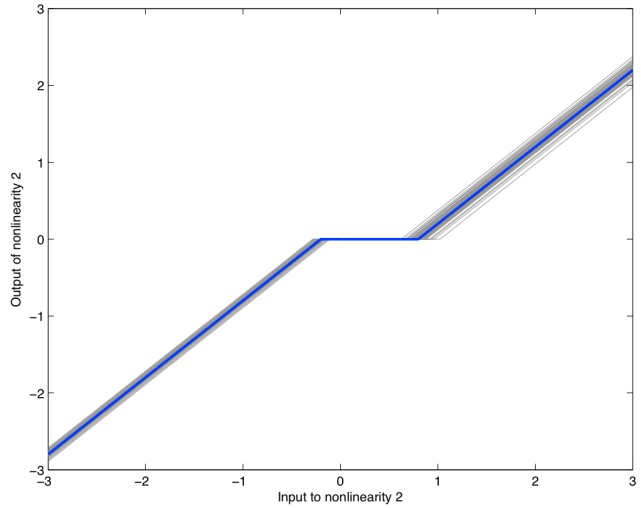

Fig. 5. Estimated (grey) and true (blue) memoryless nonlinearities for the example studied in Section 5.

6. CONCLUSION

This paper has considered the problem of identifying parameter values for Wiener systems where the input signal is not known, and the measurements are corrupted by noise. The proposed method caters for systems with potentially multiple outputs, and importantly, the static nonlinearities associated with the Wiener system are allowed to be quite general. Specifically, they do not need to be invertible.

This identification problem was specified using a maximum likelihood formulation, which depends on an underlying prediction density. The key technical difficulty in solving this problem is that the prediction density cannot be straightforwardly characterized. The impact is that the likelihood function cannot be straightforwardly evaluated, let alone maximised.

Against this, the paper employs the expectation maximisation (EM) algorithm, which does not need to evaluate the likelihood nor directly maximise it. The results of this new approach were profiled on several examples and the performance is very promising.

ACKNOWLEDGEMENTS

We would like to thank Dr Jimmy Olsson for pointing us to the particle smoother introduced in Douc et al. (2010) and Fredrik Lindsten for providing us with useful comments on the final manuscript.

REFERENCES

Abed-Meraim, K., Qiu, W., and Hua, Y. (1997). Blind system identification. *Proceedings of the IEEE*, 85(8), 1310–1322.

Arulampalam, M.S., Maskell, S., Gordon, N., and Clapp, T. (2002). A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2), 174–188.

Bai, E.W. (2002). A blind approach to the Hammerstein-Wiener model identification. *Automatica*, 38(6), 967–979.

Breiman, L. (1993). Hinging hyperplanes for regression, classification, and function approximation. *IEEE Transactions on Information Theory*, 39(3), 999–1013.

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

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

Douc, R., Garivier, A., Moulines, E., and Olsson, J. (2010). Sequential Monte Carlo smoothing for general state space hidden Markov models. *Submitted to Annals of Applied Probability*.

Doucet, A., de Freitas, N., and Gordon, N. (eds.) (2001). *Sequential Monte Carlo Methods in Practice*. Springer Verlag.

Doucet, A. and Johansen, A.M. (2011). *The Oxford Handbook of Nonlinear Filtering*, chapter A tutorial on particle filtering and smoothing: fifteen years later, D. Crisan and B. Rozovsky (eds.), 656–704. Oxford University Press.

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

Hagenblad, A., Ljung, L., and Wills, A. (2008). Maximum likelihood identification of Wiener models. *Automatica*, 44(11), 2697 – 2705.

Hsu, K., Vincent, T., and Poolla, K. (2006). A kernel based approach to structured nonlinear system identification part I: Algorithms, part II: Convergence and consistency. In *Proceeding of the 14th IFAC Symposium on System Identification*. Newcastle, Australia.

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

Ljung, L. (1999). *System Identification - Theory for the User*. Prentice-Hall, Upper Saddle River, N.J., 2nd edition.

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

Peeters, B. and Roeck, G.D. (2001). Stochastic system identification for operational modal analysis: A review. *Journal of Dymamic Systems Measument and Control*, 123(4), 659–667.

Ristic, B., Arulampalam, S., and Gordon, N. (2004). *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech house, Boston, MA, USA.

Schön, T.B., Wills, A., and Ninness, B. (2011). System identification of nonlinear state space models. *Automatica*, 47(1), 39–49.

Schoukens, J., Nemeth, J.G., Crama, P., Rolain, Y., and Pintelon, R. (2003). Fast approximate identification of nonlinear systems. *Automatica*, 39(7), 1267–1274. July.

Vanbaylen, L., Pintelon, R., and Schoukens, J. (2009). Blind maximum-likelihood identification of Wiener systems. *IEEE Transactions on Signal Processing*, 57(8), 3017–3029.

Westwick, D. and Verhaegen, M. (1996). Identifying MIMO Wiener systems using subspace model identification methods. *Signal Processing*, 52, 235–258.

Wigren, T. (1993). Recursive prediction error identifi- cation using the nonlinearwiener model. *Automatica*, 29(4), 1011–1025.

Wills, A. and Ljung, L. (2010). Wiener system identification using the maximum likelihood method. In F. Giri and E.W. Bai (eds.), *Block-Oriented Nonlinear System Identification, Lecture Notes in Control and Information Science no 404*, 404. Springer.

Zhu, Y. (1999). Parametric Wiener model identification for control. In *Proceedings of the 14th IFAC World Congress*, 37–42. Beijing, China.