

 Open access • Book Chapter • DOI:10.1007/978-1-84996-513-2_7

Wiener System Identification Using the Maximum Likelihood Method

— [Source link](#) 

Adrian Wills, Lennart Ljung

Institutions: University of Newcastle, Linköping University

Published on: 01 Jan 2010 - Lecture Notes in Control and Information Sciences (Springer)

Topics: System identification, Nonlinear system and Automatic control

Related papers:

- [System Identification: Theory for the User](#)
- [Identifying MIMO Wiener systems using subspace model identification methods](#)
- [Blind Identification of Wiener Models](#)
- [Blind Maximum-Likelihood Identification of Wiener Systems](#)
- [Blind Maximum-likelihood Identification of Wiener and Hammerstein Nonlinear Block Structures](#)

Share this paper:    

View more about this paper here: <https://typeset.io/papers/wiener-system-identification-using-the-maximum-likelihood-4u9cazr4l1>

Wiener system identification using the maximum likelihood method

Adrian Wills, Lennart Ljung

Division of Automatic Control

E-mail: Adrian.Wills@newcastle.edu.au, ljung@isy.liu.se

20th December 2010

Report no.: LiTH-isy-R-2990

Accepted for publication in Block-Oriented Nonlinear System Identification, Ed: F. Giri and E. W. Bai, Springer, 2010.

Address:

Department of Electrical Engineering

Linköpings universitet

SE-581 83 Linköping, Sweden

WWW: <http://www.control.isy.liu.se>

AUTOMATIC CONTROL
REGLERTEKNIK
LINKÖPINGS UNIVERSITET



Abstract

The Wiener model is a block oriented model where a linear dynamic system block is followed by a static nonlinearity block. The dominant method to estimate these components has been to minimize the error between the simulated and the measured outputs. This is known to lead to biased estimates if disturbances other than measurement noise are present. For the case of more general disturbances we present Maximum Likelihood expressions and provide algorithms for maximising them. This includes the case where disturbances may be coloured and as a consequence we can handle blind estimation of Wiener models. This case is accommodated by using the Expectation-Maximisation algorithm in combination with particles methods. Comparisons between the new algorithms and the dominant approach confirm that the new method is unbiased and also has superior accuracy.

Keywords: System Identification, Nonlinear models, maximum likelihood, Wiener models

Wiener system identification using the maximum likelihood method

Adrian Wills and Lennart Ljung

Abstract The Wiener model is a block oriented model where a linear dynamic system block is followed by a static nonlinearity block. The dominant method to estimate these components has been to minimize the error between the simulated and the measured outputs. This is known to lead to biased estimates if disturbances other than measurement noise are present. For the case of more general disturbances we present Maximum Likelihood expressions and provide algorithms for maximising them. This includes the case where disturbances may be coloured and as a consequence we can handle blind estimation of Wiener models. This case is accommodated by using the Expectation-Maximisation algorithm in combination with particles methods. Comparisons between the new algorithms and the dominant approach confirm that the new method is unbiased and also has superior accuracy.

Dedicated To Anna Hagenblad (1971 - 2009)

Much of the research presented in this chapter was initiated and pursued by Anna as part of her work towards a Ph.D thesis, which she sadly never had the opportunity to finish. Her interest in this research area spanned nearly ten years and her contributions were significant. She will be missed. We dedicate this work to the memory of Anna.

Adrian Wills

School of Electrical Engineering and Computer Science, University of Newcastle,
Callaghan, NSW, 2308, Australia e-mail: adrian.wills@newcastle.edu.au

Lennart Ljung

Division of Automatic Control, Linköpings universitet, SE-581 80 Linköping, Sweden,
e-mail: ljung@isy.liu.se

1 Introduction

Within the class of nonlinear system models, the so-called *block-oriented models* have gained wide recognition and attention by the system identification and automatic control community. Typically, these models are constructed by joining linear dynamic system blocks with static nonlinear mappings in various forms of interconnection.

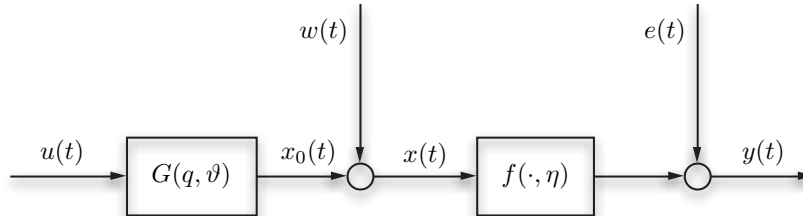


Fig. 1 The Wiener model. The input $u(t)$ and the output $y(t)$ are measurable, but not the intermediate signal $x(t)$. $w(t)$ and $e(t)$ are noise sources. $x_0(t)$ denotes the output of the linear dynamic system G . f is nonlinear and static (memoryless).

The Wiener model depicted in Figure 1 is one such block-oriented model, see, e.g. [2], [18] or [9]. It is typically comprised of two blocks, where the first one is linear and dynamic and the second is nonlinear and static.

From one perspective, these models are reasonable since they often reflect the physical realities of a system. Some examples of this include distillation columns [24], pH control processes [11], and biological examples [10]. More generally, they accurately model situations where the output of a linear system is obtained using a nonlinear measurement device.

From another perspective, if the blocks of a Wiener model are multi-variable, then it can be shown [3] that almost any nonlinear system can be approximated arbitrarily well using them. However, this is not the focus of the current chapter, where single input - single output systems are considered.

With this as motivation, in this chapter we are concerned with estimating Wiener models based on input and/or output measurements. To make these ideas more precise, we will adopt the notation used in Figure 1 here and throughout the remainder of this chapter. In particular, the input signal is denoted by $u(t)$, the output signal by $y(t)$ and $x(t)$ denotes the intermediate unmeasured signal. The disturbance term $w(t)$ is henceforth called the process noise and $e(t)$ is called the measurement noise as usual. These noise terms are assumed to be mutually independent.

Using this notation, the Wiener system can be described by the following equations.

$$\begin{aligned}
x_0(t) &= G(q, \vartheta)u(t) \\
x(t) &= x_0(t) + w(t) \\
y(t) &= f(x(t), \eta) + e(t)
\end{aligned} \tag{1}$$

Throughout this chapter it is assumed that f and G each belong to a parametrized model class. Typical classes for the nonlinear term f include basis function expansions such as polynomials, splines, or neural networks. The nonlinearity f may also be a piecewise linear function, such as a dead-zone or saturation function. Typical classes for the linear term G include rational transfer functions and linear state space models.

It is important to note that if the process noise w and the intermediate signal x are unknown, then the parametrization of the Wiener model is not unique. For example, scaling the linear block G via κG and scaling the nonlinear block f via $f(\frac{1}{\kappa} \cdot)$ will result in identical input–output behaviour. (It may necessary to scale the process noise variance with a factor κ .)

Based on the above description, the problem addressed in this chapter is to estimate the parameters ϑ within the model class for G and η within the model class for f that best match the measured output data from the system.

For convenience, we define a joint parameter vector θ as

$$\theta = [\vartheta^T, \eta^T]^T \tag{2}$$

which will be used throughout this chapter.

2 An Output-Error Approach

While there are several methods for identifying Wiener models proposed in the literature, the most dominant of these is to parametrize the linear and the nonlinear blocks, and then estimate the parameters from data by minimizing an output-error criterion (this has been used in [1], [21] and [22] for example).

In particular, if the process noise $w(t)$ in Figure 1 is ignored, then a natural criterion is to minimize

$$V_N(\theta) = \frac{1}{N} \sum_{t=1}^N \left(y(t) - f(G(q, \vartheta)u(t), \eta) \right)^2 \tag{3}$$

This approach is standardly used in software packages such as [23] and [12].

If it is true that the process noise $w(t)$ is zero, then (3) becomes the prediction-error criterion. Furthermore, if measurement noise is white and Gaussian, (3) is also the Maximum Likelihood criterion and the estimate is therefore consistent [13].

Even for the case where there is process noise, it may still seem reasonable to use an output-error criterion like (3) to obtain an estimate. However,

$f(G(q, \vartheta)u(t), \eta)$ is not the true predictor in this case and it has been shown in [8] that this can result in biased estimates.

A further difficulty with this approach is that it cannot directly handle the case of blind Wiener model estimation where the process noise is assumed to be zero, but the input $u(t)$ is not measured. Related criteria to (3) have been derived for this case, but they assume that the nonlinearity is invertible and/or that the measurement noise is not present [20, 19].

By way of motivating the main tool in this chapter, namely Maximum Likelihood estimation, the next section provides conditions for the estimates of (3) to be consistent. It is shown by example that using the output-error criterion can produce biased estimates. These results appeared in [8].

2.1 Consistent Estimates

Consider a Wiener system in the form of Figure 1 and Equation (1) and assume we have measurements of the input and output according to some “true” parameters (ϑ_0, η_0) , i.e.

$$y(t) = f(G(q, \vartheta_0)u(t) + w(t), \eta_0) + e(t) \quad (4)$$

Based on the measured inputs and outputs, we would like to find an estimate of these parameter values, $(\hat{\vartheta}, \hat{\eta})$ say, that are *close* to the true parameters. A more precise way of describing this is to say that an estimate is *consistent* if the parameters converge to their true values as the number of data, N tends to infinity.

In order to make this idea concrete for the output-error criterion in (3) we write the true system (4) as

$$y(t) = f(G(q, \vartheta_0)u(t), \eta_0) + \tilde{w}(t) + e(t) \quad (5)$$

where

$$\tilde{w}(t) = f(G(q, \vartheta_0)u(t) + w(t), \eta_0) - f(G(q, \vartheta_0)u(t), \eta_0) \quad (6)$$

The new disturbance term $\tilde{w}(t)$ may be regarded as a (input-dependent) transformation of the process noise $w(t)$ to the output. This transformation will most likely distort the stochastic properties of $w(t)$, such as mean and variance, compared with $\tilde{w}(t)$.

By inserting the equation for y in (5) into the criterion (3), we receive the following expression.

$$\begin{aligned}
V_N(\boldsymbol{\theta}) &= \frac{1}{N} \sum_{t=1}^N \left(f_0 - f + \tilde{w}(t) + e(t) \right)^2 \\
&= \frac{1}{N} \sum_{t=1}^N \left(f_0 - f \right)^2 + \frac{1}{N} \sum_{t=1}^N \left(\tilde{w}(t) + e(t) \right)^2 + \frac{2}{N} \sum_{t=1}^N \left(f_0 - f \right) \left(\tilde{w}(t) + e(t) \right)
\end{aligned} \tag{7}$$

where

$$f_0 \triangleq f(G(q, \vartheta_0)u(t), \eta_0), \quad f \triangleq f(G(q, \vartheta)u(t), \eta). \tag{8}$$

Further, assume that all noise terms are ergodic, so that time averages tend to their mathematical expectations as N tends to infinity. Assume also that u is a (quasi)-stationary sequence [13], so that it also has well defined sample averages. Let, E denote both mathematical expectation and averaging over time signals (cf. \bar{E} in [13]). Using the fact that the measurement noise e is zero mean, and independent of the input u and the process noise w means that several cross terms will disappear. The criterion then tends to

$$\bar{V}(\boldsymbol{\theta}) = E \left(f_0 - f \right)^2 + E \bar{w}^2(t) + E e^2(t) + 2E \left(f_0 - f \right) \bar{w}(t) \tag{9}$$

The last term in this expression cannot necessarily be removed since the transformed process noise \tilde{w} need not be independent of u . The criterion (9) has a quadratic form, and the true values (ϑ_0, η_0) will minimize the criterion if and essentially only if

$$E \left(f(G(q, \vartheta_0)u(t), \eta_0) - f(G(q, \vartheta)u(t), \eta) \right) \bar{w}(t) = 0 \tag{10}$$

Typically, this will not hold due to the possible dependence between u and \tilde{w} . The parameter estimates will therefore be biased in general. To illustrate this, we provide an example below.

Example 1. Consider the following Wiener system, with linear dynamic part described by

$$\begin{aligned}
x_0(t) + 0.5x_0(t-1) &= u(t-1) \\
x(t) &= x_0(t) + w(t)
\end{aligned} \tag{11}$$

followed by a static nonlinearity described as a second order polynomial,

$$\begin{aligned}
f(x(t)) &= c_0 + c_1 x^2(t) \\
y(t) &= f(x(t)) + e(t)
\end{aligned} \tag{12}$$

The goal is to estimate the nonlinearity parameters denoted here by \hat{c}_0 and \hat{c}_1 .

In this case it is possible to provide expressions for the analytical minimum of criterion (3). Recall that in this case the process noise $w(t)$ is assumed to be zero. Therefore, the predicted output can be expressed as

$$\hat{y}(t) = f(G(q, \vartheta)u(t), \eta) = f(x_0(t), \eta) = \hat{c}_0 + \hat{c}_1 x_0^2(t) \quad (13)$$

Assume that all signals, noises as well as inputs, are Gaussian, zero mean and ergodic. Let λ_x denote the variance of x_0 , λ_w denote the variance of w , and λ_e denote the variance of e . As N tends to infinity, the criterion (3) tends to the limit (9)

$$\begin{aligned} \bar{V} &= E(y - \hat{y})^2 = E(c_0 + c_1(x_0 + w)^2 + e - \hat{c}_0 - \hat{c}_1 x_0^2)^2 \\ &= E((c_1 - \hat{c}_1)x_0^2 + c_0 - \hat{c}_0 + 2c_1 x_0 w + c_1 w^2 + e)^2 \end{aligned}$$

All the cross terms will be zero since the signals are Gaussian, independent and zero mean. The fourth order moments are $E x^4 = 3\lambda_x^2$ and $E w^4 = 3\lambda_w^2$. This leaves

$$\begin{aligned} \bar{V} &= 3(c_1 - \hat{c}_1)^2 \lambda_x^2 + (c_0 - \hat{c}_0)^2 + 4c_1 \lambda_x \lambda_w + 3c_1^2 \lambda_w^2 + \lambda_e \\ &\quad + 2(c_0 - \hat{c}_0) \times (c_1 - \hat{c}_1) \lambda_x + 2c_1(c_1 - \hat{c}_1) \lambda_x \lambda_w + 2c_1(c_0 - \hat{c}_0) \lambda_w \end{aligned}$$

From this expression it is possible to compute the gradient with respect to each \hat{c}_i and therefore find the minimum by solving

$$\begin{aligned} (c_0 - \hat{c}_0) + (c_1 - \hat{c}_1) + c_1 \lambda_w &= 0 \\ 3(c_1 - \hat{c}_1) \lambda_x^2 + (c_0 - \hat{c}_0) \lambda_x + 3c_1 \lambda_x \lambda_w &= 0 \end{aligned}$$

with the solution

$$\hat{c}_0 = c_0 + c_2 \lambda_w, \quad \hat{c}_1 = c_1.$$

Therefore, the estimate of c_0 is clearly biased.

Motivated by the above example, the next section investigates the use of the Maximum-Likelihood criterion to estimate the system parameters, which is known to produce consistent estimates under the assumptions of this chapter [13].

3 The Maximum Likelihood Method

The maximum likelihood method provides estimates of the parameter values θ based on an observed data set $Y_N = \{y(1), y(2), \dots, y(N)\}$ by maximizing a likelihood function. In order to use this method it is therefore necessary to first derive an expression for the likelihood function itself.

The likelihood function is the probability density function (PDF) of the outputs that is parametrized by θ . We shall assume for the moment that the input sequence $U_N = \{u(1), u(2), \dots, u(N)\}$ is a given, deterministic se-

quence (the case of blind Wiener estimation where the input is assumed to be stochastic is subsumed by the coloured process noise case in Section 3.2).

This likelihood will be denoted here by $p_\theta(Y_N)$ and the Maximum-Likelihood (ML) estimate is obtained via

$$\hat{\theta} = \arg \max_{\theta} p_\theta(Y_N) \quad (14)$$

This approach enjoys a long and fruitful history within the system identification community because of its statistical efficiency in producing consistent estimates (see e.g. [13]).

In the following sections we will provide expressions of the likelihood function for various Wiener models. In particular, we firstly consider the system depicted in Figure 1 and then consider a related one whereby the process noise is allowed to be coloured. Finally, we consider the case where the input signal is unknown (the is the so-called blind estimation problem).

Based on these expressions, Section 4 provides algorithms for computing the ML estimate. This includes the direct gradient-based approach for models in the form of Figure 1, which was presented in [8]. In addition, the Expectation-Maximisation approach is presented for the case of coloured process noise.

3.1 Likelihood Function for White Disturbances

For the Wiener model in Figure 1 we assume that the disturbance sequences $e(t)$ and $w(t)$ are each white noise. This means that for given input sequence U_N , $y(t)$ will also be a sequence of independent variables. This in turn implies that the PDF of Y_N will be the product of the PDF's of $y(t), t = 1, \dots, N$. Therefore, it is sufficient to derive the PDF of $y(t)$. To simplify notation we shall use $y(t) = y$, $x(t) = x$.

As a means to expressing this PDF, we firstly introduce an intermediate signal x (see Figure 1) as a nuisance parameter. The benefit of introducing this term is that the PDF of y given x is basically a reflection of the PDF of e since $y(t) = f(x(t)) + e(t)$ hence

$$p_y(y|x) = p_e(y - f(x, \eta)) \quad (15)$$

where p_e is the PDF of e . In a similar manner, the PDF of x given U_N can be obtained by noting that

$$x(t) = G(q, \vartheta)u(t) + w(t) = x_0(t, \vartheta) + w(t) \quad (16)$$

So that for a given U_N and ϑ , x_0 is a known, deterministic variable, and hence

$$p_x(x) = p_w(x - x_0(\vartheta)) = p_w(x - G(q, \vartheta)u(t)) \quad (17)$$

where p_w is the PDF of w .

Since $x(t)$ is not measured, then we must integrate over all $x \in \mathbf{R}$ in order to eliminate it from the expressions to receive

$$\begin{aligned} p_y(y) &= \int_{x \in \mathbf{R}} p_{x,y}(x,y) dx \\ &= \int_{x \in \mathbf{R}} p_y(y|x) p_x(x) dx \\ &= \int_{x \in \mathbf{R}} p_e(y - f(x, \eta)) p_w(x - G(q, \vartheta)u(t)) dx \end{aligned} \quad (18)$$

In order to proceed further, it is necessary to assume a PDF for e and w . Therefore, we assume that the process noise $w(t)$ and the measurement noise $e(t)$ are Gaussian, with zero means and variances λ_w and λ_e respectively, i.e.

$$p_e(\varepsilon) = \frac{1}{\sqrt{2\pi\lambda_e}} e^{-\frac{1}{2\lambda_e}\varepsilon^2} \quad \text{and} \quad p_w(v) = \frac{1}{\sqrt{2\pi\lambda_w}} e^{-\frac{1}{2\lambda_w}v^2} \quad (19)$$

The joint likelihood can be expressed as the product over all time instants since the noise is white, so that

$$p_\theta(Y_N) = \left(\frac{1}{2\pi\sqrt{\lambda_e\lambda_w}} \right)^N \prod_{t=1}^N \int_{-\infty}^{\infty} e^{-\frac{1}{2}\varepsilon(t,\theta)} dx(t) \quad (20)$$

where

$$\varepsilon(t, \theta) = \frac{1}{\lambda_e} \left(y(t) - f(x(t), \eta) \right)^2 + \frac{1}{\lambda_w} \left(x(t) - G(q, \vartheta)u(t) \right)^2 \quad (21)$$

Therefore, when provided with the observed data U_N and Y_N , we can calculate $p_\theta(Y_N)$ and its gradients for each θ . This means that the ML criterion (14) can be maximized numerically. This approach is detailed in Section 4.1.

The derivation of the Likelihood function appeared in [7] and [8].

3.2 Likelihood Function for Coloured Process Noise

If the process noise is coloured, we may represent the Wiener system as in Figure 2. In this case, equations for the output are given by

$$\begin{aligned} x(t) &= G(q, \vartheta)u(t) + H(q, \vartheta)w(t) \\ y(t) &= f(x(t), \eta) + e(t) \end{aligned} \quad (22)$$

By using the predictor form, see [13], we may write this as

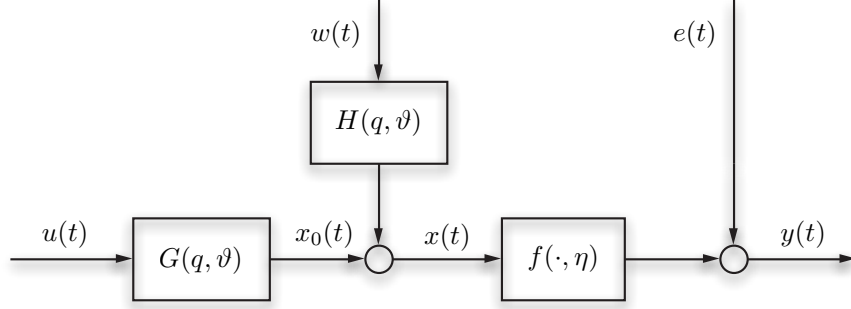


Fig. 2 Wiener model with colored process noise. Both $w(t)$ and $e(t)$ are white noise sources, but $w(t)$ is filtered through $H(q, \vartheta)$.

$$x(t) = \hat{x}(t|X_{t-1}, U_t, \vartheta) + w(t) \quad (23)$$

$$\hat{x}(t|X_{t-1}, U_t, \vartheta) \triangleq H^{-1}(q, \vartheta)G(q, \vartheta)u(t) + (1 - H^{-1}(q, \vartheta))x(t) \quad (24)$$

$$y(t) = f(x(t), \eta) + e(t) \quad (25)$$

In the above, X_{t-1} denotes the sequence $X_{t-1} = \{x(1), \dots, x(t-1)\}$ and similarly for U_t . The only stochastic parts are e and w , hence for a given sequence X_N , the joint PDF of Y_N is obtained in the standard way

$$p_{Y_N}(Y_N|X_N) = \prod_{t=1}^N p_e(y(t) - f(x(t), \eta)) \quad (26)$$

On the other hand, the joint PDF for X_N is given by (c.f. eq (5.74), Lemma 5.1, in [13])

$$p_{X_N}(X_N) = \prod_{t=1}^N p_w(x(t) - \hat{x}(t|X_{t-1}, U_t, \vartheta)) \quad (27)$$

The likelihood function for Y_N is thus obtained from (26) by integrating out the nuisance parameter X_N using its PDF (27)

$$p_{\theta}(Y_N) = \int \prod_{t=1}^N p_w(H^{-1}(q, \vartheta)[x(t) - G(q, \vartheta)u(t)]) p_e(y(t) - f(x(t), \eta)) dX_N \quad (28)$$

Unfortunately, in this case filtered versions of $x(t)$ enter the integral, which means that the integration is a true multidimensional integral over the entire sequence X_N . This is likely to be intractable using direct integration methods in practise, unless the inverse noise filters are short FIR filters.

Motivated by this, here we adopt another approach whereby the noise filter H is described in state-space form as

$$H(q, \vartheta) = C(\vartheta)(qI - A(\vartheta))^{-1}B(\vartheta). \quad (29)$$

where A , B , C are state-space matrices, and the state update is described via

$$\xi(t+1) = A(\vartheta)\xi(t) + B(\vartheta)w(t) \quad (30)$$

Therefore, according to Figure 2, the output can be expressed as

$$y(t) = f(C(\vartheta)\xi(t) + G(q, \vartheta)u(t), \eta) + e(t) \quad (31)$$

Equations (30) and (31) are in the form of a nonlinear state-space model, which has recently been considered in [17]. In that paper the authors use the Expectation-Maximisation algorithm in conjunction with particle methods to compute the ML estimate. We also adopt this technique here, which is detailed in Section 4.2.

Blind estimation

Note that if the linear term G was zero, then the above system will become a blind Wiener model, so that (31) becomes

$$y(t) = f(C(\vartheta)\xi(t), \eta) + e(t) \quad (32)$$

and the parameters in H and f must be estimated via the output measurements only. This case is profiled via a simulation example in Section 5.3.

4 Maximum Likelihood Algorithms

For the case of white Gaussian process and measurement noise described in Section 3.1, it was mentioned that numerical methods could be used to evaluate the likelihood integral in Equation (20). At the same time, these methods can be used to compute the gradient for use in a gradient based search procedure to find the maximum likelihood estimate. This is the approach outlined in Section 4.1 below and profiled in Section 5 by way of simulation examples.

While this method is very useful and practical, it does not handle the case of estimating parameters of a colouring filter for the case discussed in Section 3.2. Further, it does not handle the blind estimation case discussed in Section 3.2.

Therefore, we present an alternative method based on using the Expectation Maximisation (EM) approach in Section 4.2 below. A key point to note is that this method requires a nonlinear smoothing operation and this is achieved via particle methods. Again, the resulting algorithm is profiled in Section 5 by way of simulation studies.

4.1 Direct Gradient Based Search Approach

In this section we are concerned with maximising the likelihood function described in (20) and (21) via gradient based search. In order to avoid numerical conditioning issues, we consider the equivalent problem of maximising the log-likelihood function provided below.

$$\hat{\theta} = \arg \max_{\theta} L(\theta) \quad (33)$$

where

$$L(\theta) \triangleq \log(p_{\theta}(Y_N)) \quad (34)$$

$$= -N \log(2\pi) - \frac{N}{2} \log(\lambda_w \lambda_e) + \sum_{t=1}^N \log \left(\int_{-\infty}^{\infty} e^{-\frac{1}{2} \varepsilon(t, \theta)} dx \right) \quad (35)$$

and $\varepsilon(t, \theta)$ is given by Equation (21).

To solve (33) here we employ an iterative gradient based approach. Typically, this approach proceeds by computing a “search direction”, and then the function L is increased along the search direction to obtain a new parameter estimate. This search direction is usually determined so that it forms an acute angle with the gradient, since under these conditions it can be shown to increase the cost when added to the current estimate.

To be more precise, at iteration k , $L(\theta_k)$ is modeled locally as

$$L(\theta_k + p) \approx L(\theta_k) + g_k^T p + \frac{1}{2} p^T H_k^{-1} p, \quad (36)$$

where g_k is the derivative of L with respect to θ evaluated at θ_k and H_k^{-1} is a symmetric matrix. If a Newton direction is desired, then H_k^{-1} would be the inverse of Hessian matrix, but the Hessian matrix itself may be quite expensive to compute. However, the structure in (34) is directly amenable to using Gauss-Newton gradient based search [4], which provides a good approximation to the Hessian. Here, however, we employ a quasi-Newton method where H_k is updated at each iteration based on local gradient information so that it resembles the Hessian matrix in the limit. In particular, we use the well-known BFGS update strategy [15, Section 6.1], which can guarantee that H_k is negative definite and symmetric so that

$$p_k = -H_k g_k \quad (37)$$

maximizes (36). The new parameter estimate θ_{k+1} is then obtained by updating the previous one via

$$\theta_{k+1} = \theta_k + \alpha_k p_k, \quad (38)$$

where α_k is selected such that

$$L(\theta_k + \alpha_k p_k) > L(\theta_k). \quad (39)$$

Evaluating the cost $L(\theta_k)$ and its derivative g_k are essential to the success of the above approach. For the case of computing the cost, we see from (34) that this requires the evaluation of an integral. Similarly, note that the i 'th element of the gradient vector g_k , denoted $g_k(i)$, is given by

$$g_k(i) = \left. \left[\frac{N}{2} \frac{\partial \log(\lambda_w)}{\partial \theta(i)} + \frac{N}{2} \frac{\partial \log(\lambda_w)}{\partial \theta(i)} + \frac{1}{2} \sum_{t=1}^N \frac{\int_{-\infty}^{\infty} \frac{\partial \varepsilon(t, \theta)}{\partial \theta(i)} e^{-\frac{1}{2} \varepsilon(t, \theta)} dx}{\int_{-\infty}^{\infty} e^{-\frac{1}{2} \varepsilon(t, \theta)} dx} \right] \right|_{\theta=\theta_k} \quad (40)$$

so that computing the gradient vector also requires evaluation of an integral.

Evaluating the integrals in (34) and (40) will be achieved numerically in this chapter. In particular, we employ a fixed-interval grid over x and use the composite Simpson's rule to obtain the approximation [16, Chapter 4]. The reason for employing a fixed grid (it need not be of fixed-interval as used here) is that it allows straightforward computation of $L(\theta_k)$ and its derivative g_k at the same grid points. This is detailed in Algorithm 1 below and used in the simulations in Section 5.

4.2 Expectation Maximisation Approach

In this section we address the coloured process noise case introduced in Section 3.2. As mentioned in that section, the likelihood function as expressed in (28) involved the evaluation of a high dimensional integral, which is not tractable on desktop computers. To tackle this problem, the output $y(t)$ was expressed as a nonlinear state-space model via (31), (29) and (30).

In this form, the problem is directly amenable to the recently developed Expectation Maximisation (EM) algorithm described in [17]. This section will detail the EM approach as applied to the coloured process noise case. It is also directly applicable to the blind estimation case discussed in Section 3.2.

In keeping with the notation already defined in Section 4.1 above, the EM algorithm is a method for computing $\hat{\theta}$ in (33) 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 the so-called "missing data" Z

$$L_{Z, Y_N}(\theta) = \log p_{\theta}(Z, Y_N). \quad (41)$$

In some cases, the missing data is quite literally measurements that are absent for some reason. More generally though, the missing data Z consists of "measurements" that while not available, would be useful to the estimation

Algorithm 1 : Numerical computation of likelihood and derivatives

Given an odd number of grid points M , the parameter vector θ and the data U_N and Y_N , perform the following steps. (Note that after the algorithm has terminated, the cost $L \approx \bar{L}$ and gradient $g \approx \bar{g}$).

1. Simulate the system $x_0(t) = G(\vartheta, q)u(t)$.
2. Specify grid vector $\Delta \in \mathbb{R}^M$ as M equidistant points between the limits $[a \ b]$, so that $\Delta(1) = a$ and $\Delta(i+1) = \Delta(i) + (b-a)/M$ for all $i = 1, \dots, M-1$.
3. Set $\bar{L} = N \log(2\pi) + \frac{N}{2} \log(\lambda_w \lambda_e)$, and $\bar{g}(i) = 0$ for $i = 1, \dots, n_\theta$.
4. **for** $t=1:N$,
 - a. **for** $j=1:M$, compute

$$x = x_0(t) + \Delta(j), \quad \alpha = x - x_0(t), \quad \beta = y(t) - f(x, \eta)$$

$$\gamma_j = e^{-\frac{1}{2}(\alpha^2/\lambda_w + \beta^2/\lambda_e)}, \quad \delta_j(i) = \gamma_j \frac{\partial \mathcal{E}(t, \theta)}{\partial \theta(i)}, \quad i = 1, \dots, n_\theta,$$

end

- b. Compute

$$\kappa = \frac{(b-a)}{3M} \left(\gamma_1 + 4 \sum_{j=1}^{\frac{M-1}{2}} \gamma_{2j} + 2 \sum_{j=1}^{\frac{M-3}{2}} \gamma_{2j+1} + \gamma_M \right),$$

$$\pi(i) = \frac{(b-a)}{3M} \left(\delta_1(i) + 4 \sum_{j=1}^{\frac{M-1}{2}} \delta_{2j}(i) + 2 \sum_{j=1}^{\frac{M-3}{2}} \delta_{2j+1}(i) + \delta_M(i) \right), \quad i = 1, \dots, n_\theta,$$

$$\bar{L} = \bar{L} - \log(\kappa),$$

$$\bar{g}(i) = \bar{g}(i) + \frac{1}{2} \left(\frac{\partial \log(\lambda_w \lambda_e)}{\partial \theta(i)} + \frac{\pi(i)}{\kappa} \right), \quad i = 1, \dots, n_\theta,$$

end

problem. As such, the choice of Z is a design variable in the deployment of the EM algorithm. For the case in Section 3.2, this choice is naturally the missing state sequence

$$Z = \{\xi_1, \dots, \xi_N\}, \quad (42)$$

since if it were known or measured, then the problem would reduce to one in the form of (3), which is more readily soluble.

It is of vital importance to understand the connection between the joint likelihood in (41) and the likelihood (34) that we are trying to optimise. Accordingly, note that by the definition of conditional probability, the likelihood (34) and the joint likelihood (41) are related by

$$\log p_\theta(Y_N) = \log p_\theta(Z, Y_N) - \log p_\theta(Z | Y_N). \quad (43)$$

Let θ_k denote an estimate of the likelihood maximiser $\hat{\theta}$ in (33). Further, denote by $p_{\theta_k}(Z | Y_N)$ 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 (43) relative to $p_{\theta_k}(Z | Y_N)$.

$$\begin{aligned} \log p_{\theta}(Y_N) &= \int \log p_{\theta}(Z, Y_N) p_{\theta_k}(Z | Y_N) dZ - \int \log p_{\theta}(Z | Y_N) p_{\theta_k}(Z | Y_N) dZ \\ &= \underbrace{E_{\theta_k} \{ \log p_{\theta}(Z, Y_N) | Y_N \}}_{\triangleq Q(\theta, \theta_k)} - \underbrace{E_{\theta_k} \{ \log p_{\theta}(Z | Y_N) | Y_N \}}_{\triangleq V(\theta, \theta_k)}. \end{aligned} \quad (44)$$

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

$$L(\theta) - L(\theta_k) = (Q(\theta, \theta_k) - Q(\theta_k, \theta_k)) + \underbrace{(V(\theta_k, \theta_k) - V(\theta, \theta_k))}_{\geq 0}. \quad (45)$$

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 [5]. As a consequence if we obtain a new estimate θ_{k+1} such that $Q(\theta_{k+1}, \theta_k) > Q(\theta_k, \theta_k)$, then it follows that $L(\theta_{k+1}) > L(\theta_k)$. So that, by increasing the Q function we are also increasing the likelihood (34).

This leads to the EM algorithm, which iterates between forming $Q(\theta, \theta_k)$ and then maximising it with respect to θ to obtain a better estimate θ_{k+1} (for further information regarding the EM algorithm, the text [14] is an excellent reference).

Algorithm 2 : Expectation Maximisation Algorithm

1. Set $k = 0$ and initialize θ_0 such that $L(\theta_0)$ is finite.

2. **Expectation (E) step**: Compute

$$Q(\theta, \theta_k) = E_{\theta_k} \{ \log p_{\theta}(Z, Y_N) | Y_N \}. \quad (46)$$

3. **Maximisation (M) step**: Compute

$$\theta_{k+1} = \arg \max_{\theta} Q(\theta, \theta_k). \quad (47)$$

4. If not converged, update $k := k + 1$ and return to step 2.

The Expectation and Maximisation steps are treated separately in Sections 4.2.1 and 4.2.2 below.

4.2.1 Expectation Step

The first challenge in implementing the EM algorithm is the computation of $Q(\boldsymbol{\theta}, \boldsymbol{\theta}_k)$ according to (44). To address this, note that via Bayes' rule and the Markov property associated with the model in (30) and (31) and with the choice (42) for Z

$$\begin{aligned} L_\theta(Z, Y_N) &= \log p_\theta(Y_N|Z) + \log p_\theta(Z) \\ &= \sum_{t=1}^{N-1} \log p_\theta(\xi_{t+1}|\xi_t) + \sum_{t=1}^N \log p_\theta(y_t|\xi_t). \end{aligned} \quad (48)$$

Applying the conditional expectation operator $\mathbf{E}_{\theta_k}\{\cdot | Y_N\}$ to both sides of (48) yields

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}_k) = I_1(\boldsymbol{\theta}, \boldsymbol{\theta}_k) + I_2(\boldsymbol{\theta}, \boldsymbol{\theta}_k), \quad (49)$$

where

$$I_1(\boldsymbol{\theta}, \boldsymbol{\theta}_k) = \sum_{t=1}^{N-1} \int \int \log p_\theta(\xi_{t+1}|\xi_t) p_{\theta_k}(\xi_{t+1}, \xi_t | Y_N) d\xi_t d\xi_{t+1}, \quad (50a)$$

$$I_2(\boldsymbol{\theta}, \boldsymbol{\theta}_k) = \sum_{t=1}^N \int \log p_\theta(y_t|\xi_t) p_{\theta_k}(\xi_t | Y_N) d\xi_t. \quad (50b)$$

Hence, computing $Q(\boldsymbol{\theta}, \boldsymbol{\theta}_k)$ requires knowledge of densities such as $p_{\theta_k}(\xi_t | Y_N)$ and $p_{\theta_k}(\xi_{t+1}, \xi_t | Y_N)$ associated with a nonlinear smoothing problem. Unfortunately, due to the nonlinear nature of the Wiener model, these densities are unlikely to have analytical expressions. This chapter therefore takes a numerical approach of evaluating (50a)-(50b) via the use of particle methods, more formally known as sequential importance resampling (SIR) methods [6]. This will result in an approximation \widehat{Q} of Q via

$$\widehat{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}_k) = \widehat{I}_1(\boldsymbol{\theta}, \boldsymbol{\theta}_k) + \widehat{I}_2(\boldsymbol{\theta}, \boldsymbol{\theta}_k) \quad (51)$$

where \widehat{I}_1 and \widehat{I}_2 are approximations to (50a) and (50b). These approximations are provided by the particle smoothing Algorithm 3 below (see [17] for background and a more detailed explanation).

To use this algorithm, we require the ability to draw new samples from the distribution $p_{\theta_k}(\xi_t | \xi_{t-1}^i)$, but this is straightforward since ξ_t is given by a linear state-space equation in (30) with white Gaussian disturbance $w(t)$. Therefore, according to (30), for each ξ_{t-1}^i we can draw ξ_t^i via

$$\xi_t^i = A\xi_{t-1}^i + B\boldsymbol{\omega}^i \quad (60)$$

where $\boldsymbol{\omega}^i$ is a realization from the appropriate Gaussian distribution for $w(t)$.

Algorithm 3 : Particle Smoother

Given the current estimate θ_k , choose the number of particles M and complete the following steps.

1. Initialize particles, $\{\xi_0^i\}_{i=1}^M \sim p_{\theta_k}(\xi_0)$ and set $t = 1$;
2. Predict the particles by drawing M i.i.d. samples according to

$$\tilde{\xi}_t^i \sim p_{\theta_k}(\tilde{\xi}_t^i | \xi_{t-1}^i), \quad i = 1, \dots, M. \quad (52)$$

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

$$w_t^i \triangleq w(\tilde{\xi}_t^i) = \frac{p_{\theta_k}(y_t | \tilde{\xi}_t^i)}{\sum_{j=1}^M p_{\theta_k}(y_t | \tilde{\xi}_t^j)}, \quad i = 1, \dots, M. \quad (53)$$

4. For each $j = 1, \dots, M$ draw a new particle ξ_t^j with replacement (resample) according to,

$$P(\xi_t^j = \tilde{\xi}_t^i) = w_t^i, \quad i = 1, \dots, M. \quad (54)$$

5. If $t < N$ increment $t \mapsto t + 1$ and return to step 2, otherwise proceed to step 6.
6. Initialise the smoothed weights to be the terminal filtered weights $\{w_t^i\}$ at time $t = N$,

$$w_{N|N}^i = w_N^i, \quad i = 1, \dots, M. \quad (55)$$

and set $t = N - 1$.

7. Compute the following smoothed weights

$$w_{t|N}^i = w_t^i \sum_{k=1}^M w_{t+1|N}^k \frac{p_{\theta_k}(\tilde{\xi}_{t+1}^k | \tilde{\xi}_t^i)}{v_t^k}, \quad (56)$$

$$v_t^k \triangleq \sum_{i=1}^M w_t^i p_{\theta_k}(\tilde{\xi}_{t+1}^k | \tilde{\xi}_t^i). \quad (57)$$

$$w_{t|N}^{ij} \triangleq \frac{w_t^i w_{t+1|N}^j p_{\theta_k}(\tilde{\xi}_{t+1}^j | \tilde{\xi}_t^i)}{\sum_{l=1}^M w_t^l p_{\theta_k}(\tilde{\xi}_{t+1}^l | \tilde{\xi}_t^i)} \quad (58)$$

8. Update $t \mapsto t - 1$. If $t > 0$ return to step 7, otherwise proceed to step 9.
9. Compute the approximations

$$\hat{I}_1(\theta, \theta_k) \triangleq \sum_{t=1}^N \sum_{i=1}^M \sum_{j=1}^M w_{t|N}^{ij} \log p_{\theta}(\tilde{\xi}_{t+1}^j | \tilde{\xi}_t^i), \quad (59a)$$

$$\hat{I}_2(\theta, \theta_k) \triangleq \sum_{t=1}^N \sum_{i=1}^M w_{t|N}^i \log p_{\theta}(y_t | \tilde{\xi}_t^i). \quad (59b)$$

In addition, we require the ability to evaluate the probabilities $p_{\theta_k}(y_t|\tilde{\xi}_t^j)$ and $p_{\theta_k}(\tilde{\xi}_{t+1}^k|\tilde{\xi}_t^i)$. Again, this is straightforward in the Wiener model case described by (29)–(31) since

$$p_{\theta_k}(y_t|\tilde{\xi}_t^j) = p_e(y_t - f(C\tilde{\xi}_t^i + G(q)u_t)), \quad (61)$$

$$p_{\theta_k}(\tilde{\xi}_{t+1}^k|\tilde{\xi}_t^i) = p_w(B^\dagger[\tilde{\xi}_{t+1}^k - A\tilde{\xi}_t^i]) \quad (62)$$

where B^\dagger is the Moore-Penrose psuedo inverse of B .

4.2.2 Maximisation Step

With an approximation $\widehat{Q}(\theta, \theta_k)$ of the function $Q(\theta, \theta_k)$ made available, attention now turns to the maximisation step (47). This requires that the approximation $\widehat{Q}(\theta, \theta_k)$ is maximised with respect to θ in order to compute a new iterate θ_{k+1} of the maximum likelihood estimate.

In general, a closed form maximiser of \widehat{Q} will not be available. As such, this section again employs a gradient based search technique as already utilised in Section 4.1. For this purpose, note that via (51) and (59) the gradient of $\widehat{Q}(\theta, \theta_k)$ with respect to θ is simply computable via

$$\frac{\partial}{\partial \theta} \widehat{Q}(\theta, \theta_k) = \frac{\partial \widehat{I}_1(\theta, \theta_k)}{\partial \theta} + \frac{\partial \widehat{I}_2(\theta, \theta_k)}{\partial \theta}, \quad (63a)$$

$$\frac{\partial \widehat{I}_1(\theta, \theta_k)}{\partial \theta} = \sum_{t=1}^N \sum_{i=1}^M \sum_{j=1}^M w_{t|N}^{ij} \frac{\partial \log p_\theta(\tilde{\xi}_{t+1}^j|\tilde{\xi}_t^i)}{\partial \theta}, \quad (63b)$$

$$\frac{\partial \widehat{I}_2(\theta, \theta_k)}{\partial \theta} = \sum_{t=1}^N \sum_{i=1}^M w_{t|N}^i \frac{\partial \log p_\theta(y_t|\tilde{\xi}_t^i)}{\partial \theta}. \quad (63c)$$

In the above, we require partial derivatives of $p_{\theta_k}(y_t|\tilde{\xi}_t^j)$ and $p_{\theta_k}(\tilde{\xi}_{t+1}^j|\tilde{\xi}_t^i)$ with respect to θ . To that end, we may obtain these derivatives via simple calculus on the expressions provided in (61) and (62).

Note that for a given θ_k , the particle smoother algorithm will provide the particles $\tilde{\xi}_t^i$ and all the weights required to calculate the above gradients (and indeed \widehat{Q} itself). Importantly, these particles and weights are valid while ever θ_k remains the same (which it does throughout the Maximisation step).

With this gradient available, we can employ the same strategy that was presented in Section 4.1 for maximising L , to the case of maximising \widehat{Q} . Indeed, this was used in the simulations in Section 5.

5 Simulation Examples

In this section we profile three different algorithms on various simulation examples. To streamline the presentation it is helpful to provide each algorithm with an abbreviation. Therefore, output error approach outlined in Section 2 is denoted by OE. Secondly, the direct gradient based search method of Section 4.1 is denoted by ML-DGBS. Thirdly, the expectation maximisation method of Section 4.2 is labelled ML-EM.

For the implementation of ML-DGBS we chose the limits for the integration $[a, b]$ (see Algorithm 1) as $\pm 6\sqrt{\lambda_w}$, where λ_w is the variance of the process noise $w(t)$. This corresponds to a confidence interval of 99.9999 % for the signal $x(t)$ if the process noise is indeed Gaussian and white. The number of grid points was chosen to be 1001.

5.1 Example 1: White Process and Measurement Noise

The first example is a second order system with complex poles for the linear part $G(\vartheta, q)$, followed by a deadzone function for the nonlinear part $f(\cdot, \eta)$. The input u and process noise w are Gaussian, each with zero mean and variance 1, while the measurement noise e is Gaussian with zero mean and variance 0.1. The system is given by

$$\begin{aligned}
 x_0(t) + a_1x_0(t-1) + a_2x_0(t-2) &= u(t) + b_1u(t-1) + b_2u(t-2) \\
 x(t) &= x_0(t) + w(t) \\
 f(x(t)) &= \begin{cases} x(t) - c_1 & \text{for } x(t) < c_1 \\ 0 & \text{for } c_1 \leq x(t) \leq c_2 \\ x(t) - c_2 & \text{for } c_2 > x(t) \end{cases} \quad (64) \\
 y(t) &= f(x(t)) + e(t)
 \end{aligned}$$

Here, we estimate the parameters $a_1, a_2, b_1, b_2, c_1, c_2$.

A Monte-Carlo simulation with 1000 data sets was generated, each using $N = 1000$ samples. The true values of the parameters, and the results of the OE approach (see Section 2) and ML-DGBS method (see Section 3.1) are summarized in Table 1. The estimates of the deadzone function $f(x(t))$ from Equation (69) are plotted in Figure 3.

This simulation confirms that the output error approach provides biased estimates. On the other hand, the Maximum Likelihood method provides consistent estimates, including noise variances.

Par	True	OE	ML-DGBS
a_1	0.6000	0.5486 ± 0.0463	0.6017 ± 0.0444
a_2	-0.6000	-0.5482 ± 0.0492	-0.6015 ± 0.0480
b_1	-0.6000	-0.6002 ± 0.0146	-0.6002 ± 0.0141
b_2	0.6000	0.6006 ± 0.0130	0.6007 ± 0.0126
c_1	-0.3000	-0.1600 ± 0.0632	-0.3064 ± 0.0610
c_2	0.5000	0.3500 ± 0.0652	0.5061 ± 0.0641
λ_w	1.0000	n.e.	0.9909 ± 0.0634
λ_e	0.1000	n.e.	0.1033 ± 0.0273

Table 1 Parameter estimates with standard deviations for Example 1, using OE and ML-DGBS methods. The mean and standard deviations are computed over 1000 runs. The notation n.e. stands for “not estimated” as the noise variances are not estimated with output error method.

5.2 Example 2: Coloured Process Noise

The second example considers the Wiener model in Figure 2. It is similar to the first example in that the linear part G is a second order system with complex, but different in that we have replaced the deadzone function with a saturation function for the nonlinear part $f(\cdot, \eta)$, and different in that the process noise is coloured by

$$H(q) = \frac{q^{-1}}{1 - h_1 q^{-1}} \quad (65)$$

which corresponds to the state-space system

$$\xi(t+1) = h_1 \xi(t) + w(t). \quad (66)$$

Therefore, overall Wiener system is then given by

$$\begin{aligned}
 x_0(t) + a_1 x_0(t-1) + a_2 x_0(t-2) &= u(t) + b_1 u(t-1) + b_2 u(t-2) \\
 f(x) &= \begin{cases} c_1 & \text{for } x \leq c_1 \\ x & \text{for } c_1 < x \leq c_2 \\ c_2 & \text{for } c_2 < x \end{cases} \quad (67) \\
 y(t) &= f(\xi(t) + x_0(t)) + e(t)
 \end{aligned}$$

The goal is to estimate the parameters $a_1, a_2, b_1, b_2, c_1, c_2, h_1$ based on input and output measurements. In this case, three different algorithms were employed, namely the OE method from Section 2, the ML-DGBS approach from Section 4.1, and the ML-EM particle based method from Section 4.2. It should be mentioned that the former two algorithms do not cater for estimating the filter parameter h_1 . It is interesting nonetheless to observe their performance based on the wrong assumptions that each make about the process noise, i.e. it doesn't exist in the first case, and it is assumed white in the second.

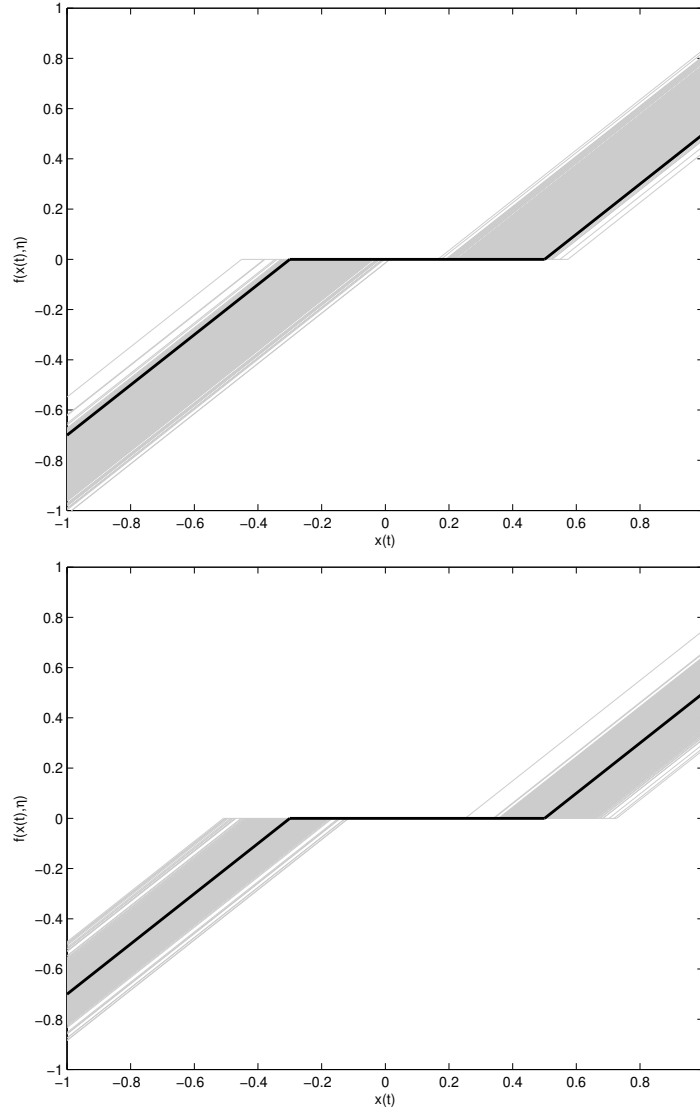


Fig. 3 Example 1: The true deadzone function as a thick black line and the 1000 estimated deadzones, appearing in grey. Above: OE. Below: ML-DGBS.

As before, we ran a Monte-Carlo simulation with 1000 runs and in each we generated a new data set with $N = 1000$ points. The signals $u(t)$, $w(t)$ and $e(t)$ were generated in the same way as for Example 1. For the EM approach, we used $M = 200$ particles in approximating Q (see (51)).

The results are summarized in Table 2. It can be observed that the output error approach again provides biased estimates of the nonlinearity param-

ters. The direct gradient based search procedure seems to produce reasonable results, but the expectation maximisation approach produces slightly more accurate results (this is perhaps surprising given that only $M = 200$ particles were used).

Par	True	OE	ML-DGBS	ML-EM
a_1	0.6000	0.5683 ± 0.2424	0.6163 ± 0.1798	0.5874 ± 0.1376
a_2	-0.6000	-0.5677 ± 0.2718	-0.6258 ± 0.2570	-0.5820 ± 0.1649
b_1	-0.6000	-0.5995 ± 0.0642	-0.5989 ± 0.0510	-0.5980 ± 0.0392
b_2	0.6000	0.6027 ± 0.0545	0.6022 ± 0.0403	0.6017 ± 0.0333
c_1	-0.5000	-0.3032 ± 0.0385	-0.4974 ± 0.0278	-0.5000 ± 0.0184
c_2	0.3000	0.1108 ± 0.0397	0.2991 ± 0.0250	0.3003 ± 0.0173
h_1	0.9000	n.e.	n.e.	0.8986 ± 0.0227
λ_w	1.0000	n.e.	5.4671 ± 1.8681	0.9765 ± 0.2410
λ_e	0.1000	n.e.	0.1000 ± 0.0069	0.1000 ± 0.0054

Table 2 Parameter estimates with standard deviations for Example 2 with colored noise, using the OE, ML-DGBS and ML-EM methods.

It is worth asking if the consistency of the ML-DGBS approach for colored process noise is surprising or not. It is well known from linear identification that the Output Error approach gives consistent estimates, even when the output error disturbance is colored, and thus an erroneous likelihood criterion is used, [13].

The Wiener model resembles the output error model in that, in essence, it is a static model, i.e. for given input u noise is added to the deterministic variable $\beta(t) = G(q)u(t)$ as $\beta(t) + e(t)$ (linear output error) or as $f(\beta(t) + w(t)) + e(t)$ (Wiener model). The spectrum or time correlation of the noises do not seem essential. However, a formal proof of this does not appear to be straightforward in the Wiener case.

Therefore, given the relative simplicity of implementing the ML-DGBS method compared with the EM approach, and given that the estimates for both approaches are comparable, it is worth asking whether or not the noise model really needs to be estimated.

On the other hand, if it is essential that the noise model be identified, then the output error and ML-DGBS methods are not really suitable since they do not handle this case. In line with this, the next section discusses the blind estimation problem where identifying the noise filter is essential.

5.3 Example 3: Blind Estimation

In the third simulation, we again consider the Wiener model depicted in Figure 2 but with $G = 0$. This can be interpreted as a blind Wiener model estimation problem, where the unknown input signal $w(t)$ is first passed through a

filter $H(q)$ and then secondly mapped through a static nonlinearity f . Finally, the measurements are corrupted by the disturbance $e(t)$ to provide $y(t)$.

In particular, we assume as in Example 2 that the process noise is coloured by

$$H(\vartheta, q) = \frac{q^{-1}}{1 - h_1 q^{-1}} \quad (68)$$

and the resulting signal is then mapped through a saturation nonlinearity, so that the overall Wiener system is given by

$$\begin{aligned} y(t) &= f(\xi(t)) + e(t) \\ \xi(t+1) &= h_1 \xi(t) + w(t) \\ f(\xi(t)) &= \begin{cases} c_1 & \text{for } \xi(t) \leq c_1 \\ \xi(t) & \text{for } c_1 < \xi(t) \leq c_2 \\ c_2 & \text{for } c_2 < \xi(t) \end{cases} \end{aligned} \quad (69)$$

Here we are trying to estimate the parameters h_1, c_1, c_2 and the variance parameters λ_w, λ_e of the process noise $w(t)$ and $e(t)$, respectively. This is to be done based on the output measurements alone. The EM method described in Section 4.2 is directly applicable to this case, and was employed here.

As usual, we ran a Monte-Carlo simulation with 1000 runs and in each we generated a new data set with $N = 1000$ points. The signals $w(t)$ and $e(t)$ were generated as Gaussian random numbers with variance 1 and 0.1, respectively. In this case, we used only $M = 50$ particles in approximating \mathcal{Q} .

The results are summarized in Table 3. Even with a modest number of particles used, $M = 50$, the estimates are consistent and appear to be accurate.

Par	True	ML-EM
b_2	0.9000	0.8995 ± 0.0237
c_1	-0.5000	-0.4967 ± 0.0204
c_2	0.3000	0.2968 ± 0.0193
λ_w	1.0000	1.0293 ± 0.1744
λ_e	0.1000	0.1019 ± 0.0063

Table 3 Parameter estimates with standard deviations for Example 3, using the EM method.

6 Conclusion

The dominant approach for estimating Wiener models is to parametrize the linear and nonlinear parts and then minimise, with respect to these parameters, the squared error between the measured output and a simulated one

from the Wiener model. This approach implicitly assumes that no process noise is present. It was confirmed that this leads to biased estimates if the assumption is wrong.

To overcome this problem, the chapter presents two algorithms for providing maximum likelihood estimates of Wiener models that include both process and measurement noise. The first is based on the assumption that the process noise is white, and the second assumes that the process noise has been coloured by a linear filter. In the latter case, the likelihood function involves the evaluation of a high dimension integral, which is not tractable using traditional numerical integration techniques.


Motivated by this, the chapter casts the Wiener model in the form of a nonlinear state-space model, which is directly amenable to a recently developed Expectation Maximisation algorithm. Of vital importance is that the expectation step can be approximated using sequential importance resampling (or particle) methods, which are easily implemented using standard desktop computing. This approach was profiled for the case of coloured process noise with very promising results.

Finally, the case of blind Wiener model estimation can be directly handled using the expectation maximisation method presented here. The efficacy of this method was demonstrated via a simulation example.

References

1. Er-Wei Bai. Frequency domain identification of Wiener models. *Automatica*, 39(9):1521–1530, 2003.
2. S. A. Billings. Identification of non-linear systems - a survey. *IEE Proc. D*, 127:272–285, 1980.
3. Stephen Boyd and Leon O. Chua. Fading memory and the problem of approximating nonlinear operators with Volterra series. *IEEE Transactions on Circuits and Systems*, CAS-32(11):1150–1161, November 1985.
4. J. E. Dennis, Jr and R. B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Prentice-Hall, Englewood Cliffs, New Jersey, USA, 1983.
5. S. Gibson and B. Ninness. Robust maximum-likelihood estimation of multivariable dynamic systems. *Automatica*, 41(10):1667–1682, 2005.
6. 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.
7. Anna Hagenblad and Lennart Ljung. Maximum likelihood estimation of wiener models. In *Proc. 39:th IEEE Conf. on Decision and Control*, pages 2417–2418, Sydney, Australia, Dec 2000.
8. Anna Hagenblad, Lennart Ljung, and Adrian Wills. Maximum likelihood identification of wiener models. *Automatica*, 44(11):2697–2705, November 2008.
9. Kenneth Hsu, Tyrone Vincent, and Kameshwar Poolla. A kernel based approach to structured nonlinear system identification part i: Algorithms, part ii: Convergence and consistency. In *Proc. IFAC Symposium on System Identification*, Newcastle, March 2006.

10. I. W. Hunter and M. J. Korenberg. The identification of nonlinear biological systems: Wiener and Hammerstein cascade models. *Biological Cybernetics*, 55:135–144, 1986.
11. A. Kalafatis, N. Arifin, L. Wang, and W. R. Cluett. A new approach to the identification of pH processes based on the Wiener model. *Chemical Engineering Science*, 50(23):3693–3701, 1995.
12. L. Ljung, R. Singh, Q. Zhang, P. Lindskog, and A. Juditski. Developments in Mathworks system identification toolbox. In *Proc. 15th IFAC Symposium on System Identification*, Saint-Malo, France, July 2009.
13. Lennart Ljung. *System Identification, Theory for the User*. Prentice Hall, Englewood Cliffs, New Jersey, USA, second edition, 1999.
14. G. McLachlan and T. Krishnan. *The EM Algorithm and Extensions (2nd Edition)*. John Wiley and Sons, 2008.
15. J. Nocedal and S. J. Wright. *Numerical Optimization, Second Edition*. Springer-Verlag, New York, 2006.
16. W. H. Press, S. A. Teukolsky, W. A. Vetterling, and B. P. Fannery. *Numerical Recipes in C, the Art of Scientific Computing, Second Edition*. Cambridge University Press, Cambridge, 1992.
17. Thomas Schön, Adrian Wills, and Brett Ninness. System identification of nonlinear state-space models. *Automatica (provisionally accepted)*, November 2009.
18. J. Schoukens, J. G. Nemeth, P. Crama, Y. Rolain, and R. Pintelon. Fast approximate identification of nonlinear systems. *Automatica*, 39(7):1267–1274, 2003. July.
19. L. Vanbaylen, R. Pintelon, and P. de Groen. Blind maximum likelihood identification of wiener systems with measurement noise. In *Proc. 15th IFAC Symposium on System Identification*, pages 1686–1691, Saint-Malo, France, July 2009.
20. L. Vanbaylen, R. Pintelon, and J. Schoukens. Blind maximum-likelihood identification of wiener systems. *IEEE Transactions on Signal Processing*, 57(8):3017–3029, August 2009.
21. David Westwick and Michel Verhaegen. Identifying MIMO Wiener systems using subspace model identification methods. *Signal Processing*, 52:235–258, 1996.
22. Torbjörn Wigren. Recursive prediction error identification using the nonlinear Wiener model. *Automatica*, 29(4):1011–1025, 1993.
23. A. G. Wills, A. J. Mills, and B. Ninness. A matlab software environment for system identification. In *Proc. 15th IFAC Symposium on System Identification*, Saint-Malo, France, July 2009.
24. Yucai Zhu. Distillation column identification for control using Wiener model. In *1999 American Control Conference*, Hyatt Regency San Diego, California, USA, June 1999.

	Avdelning, Institution Division, Department Division of Automatic Control Department of Electrical Engineering	Datum Date 2010-12-20
	Språk Language <input type="checkbox"/> Svenska/Swedish <input checked="" type="checkbox"/> Engelska/English <input type="checkbox"/> _____	Rapporttyp Report category <input type="checkbox"/> Licentiatavhandling <input type="checkbox"/> Examensarbete <input type="checkbox"/> C-uppsats <input type="checkbox"/> D-uppsats <input checked="" type="checkbox"/> Övrig rapport <input type="checkbox"/> _____
URL för elektronisk version http://www.control.isy.liu.se		LiTH-ISY-R-2990
Titel Wiener system identification using the maximum likelihood method Title		
Författare Adrian Wills, Lennart Ljung Author		
Sammanfattning Abstract <p>The Wiener model is a block oriented model where a linear dynamic system block is followed by a static nonlinearity block. The dominant method to estimate these components has been to minimize the error between the simulated and the measured outputs. This is known to lead to biased estimates if disturbances other than measurement noise are present. For the case of more general disturbances we present Maximum Likelihood expressions and provide algorithms for maximising them. This includes the case where disturbances may be coloured and as a consequence we can handle blind estimation of Wiener models. This case is accommodated by using the Expectation-Maximisation algorithm in combination with particles methods. Comparisons between the new algorithms and the dominant approach confirm that the new method is unbiased and also has superior accuracy.</p>		
Nyckelord Keywords System Identification, Nonlinear models, maximum likelihood, Wiener models		