Working Paper

Model Structure Identification: Development and Assessment of a Recursive Prediction Error Algorithm

J. D. Stigter and M. B. Beck

WP-95-105
October 1995

IIASA International Institute for Applied Systems Analysis □ A-2361 Laxenburg □ Austria
Telephone: +43 2236 807 □ Fax: +43 2236 71313 □ E-Mail: info@iiasa.ac.at

Model Structure Identification: Development and Assessment of a Recursive Prediction Error Algorithm

J. D. Stigter and M. B. Beck

WP-95-105
October 1995

International Institute for Applied Systems Analysis □ A-2361 Laxenburg □ Austria
Telephone: +43 2236 807 □ Fax: +43 2236 71313 □ E-Mail: info@iiasa.ac.at

Abstract

The paper aims to develop a more systematic approach to the problem of model structure identification for continuous time (physically based) mathematical models with discrete observations. An introduction to the model structure identification problem is first presented. The approach to this problem is the application of a modified version of the extended Kalman filter, originally defined in [23]. This filter is tested using artificial data. The results obtained lead to a further discussion of the filter's stability properties and also to a metaphor for model structures. Further study of the numerical properties of the algorithm reveal that its stability can be improved. An alternative algorithm, the so called recursive prediction error algorithm, is modified to a Kalman-like algorithm in continuous-discrete formulation. This algorithm is also tested using artificial data. The RPE-type of filter has better stability properties and appears to be very robust to initial conditions. Its applicability to environmental case studies is set out through the application of the filter to a familiar case study. Applications of this type of filter is valuable for validation/verification of environmental and/or economical models that include a set of ordinary differential equations.

Acknowledgments

This project was funded with support from the International Task Force on Forecasting Environmental Change. The Dynamic Systems group at the International Institute of Applied Systems Analysis is kindly acknowledged for providing the facilities necessary to complete this work.

Contents

Model Structure Identification: Development and Assessment of a Recursive Prediction Error Algorithm

J. D. Stigter and M. B. Beck

1 Introduction

1.1 A Conceptual Framework

The problem of model structure identification as it is defined in environmental systems analysis is concerned with the definition of a model of the form,

$$\dot{x}(t) = f(x, u, \alpha; t) + \xi(t) \quad (1)$$

$$y(t_k) = h(x, u, \alpha; t_k) + \eta(t_k) \quad (2)$$

Here $x(t)$, $\alpha(t)$ and $u(t)$ are the state-, parameter-, and input-vector, having dimensions n, q, and r respectively. The vector $y(t_k)$ is the vector of measurements or observations ($\dim(y)$=m). Equation (2) includes the discrete time sampling process into the model.

Formally, it is required for the existence and uniqueness of the solution of (1) on the interval $[t_k, t_{k+1})$ that the function $f(x, u, \alpha)$ satisfies the Lipschitz continuity conditions and exhibits linear growth in x and α, [14]. In short the Lipschitz condition requires that there exists a $K > 0$ such that $\forall t \in [t_k, t_{k+1})$ and $u, v \in R$:

$$|f(t, u) - f(t, v)| \leq K|u - v|$$

where (u, v) can be (x_1, x_2), (u_1, u_2), or (α_1, α_2) and R is the vector space R^{n+1}, R^{q+1}, or R^{r+1}, respectively. To fully expand this assumption to the case of measurable functions, needed in the stochastic setting of the differential equation (1), the reader is referred to [14], chapter 3.

The model definition (1)-(2) is completed with a continuous-time noise process $\xi(t)$ and a discrete-time noise process $\eta(t_k)$. Both are assumed Gaussian white noise with first and second moments respectively defined by

$$E\{\xi(t)\} = 0 \quad (3)$$

$$E\{\eta(t_k)\} = 0 \quad (4)$$

$$E\{\xi(t)\xi^T(t)\} = Q(\tau)\delta(t - \tau) \quad (5)$$

$$E\{\eta(t_j)\eta^T(t_k)\} = R_1(t_k)\delta_{jk} \quad (6)$$

where $E\{\}$ is the expectation operator[1], $\delta(\tau)$ is the symbolic Dirac δ-operator, and δ_{jk} is the Kronecker δ-function. Also, an error in the input sequence

$$U(t_k) := \{u(t_0), u(t_1), ..., u(t_k)\}$$

[1]Note should be taken that in continuous time the expectation is an integral expression, while in discrete time it is a summation.

may be included that introduces an uncertainty. First and second order moments are defined by

$$E\{u(t_k)\} = \bar{u} \quad (7)$$

$$R_2(t_k) = E\{u(t_k) - \bar{u}\}\{u(t_k) - \bar{u}\}^T \quad (8)$$

The ordinary differential equation set (1) represents a very wide class of system models. *A priori* knowledge is commonly used as a starting position from which the system analyst attempts to define, validate, and improve a conceptual representation of the form (1) and (2). This process is called 'grey-box modeling' of an environmental system meaning that the model definition is based on physical and biochemical knowledge. We acknowledge the fact that distributed parameter systems, involving partial differential equations in the model structure, are preferred in some cases. In this working paper it is the aim to develop the theory for the field of identification of environmental systems. Here continuous time *lumped parameter* system models are used frequently. This is the most natural choice, since the data that have been collected are often sparse and also, many of the model definitions are based on dynamical laws that include lumped instead of distributed parameters. The model definition (1) and (2) in the above is then preferred (see e.g. [28]).

1.2 Model Structure Identification

1.2.1 Introduction

The physical and/or (bio)chemical quantities that are sampled from the system involve uncertainties. For example: A biochemical oxygen demand (state) variable is difficult to measure accurately in practice and introduces considerable uncertainty in the measurement equation (2). Also, it is typical for a system variable to fluctuate around a certain (time varying) mean value. Such fluctuations are included via a random process $\xi(t)$ in equation (1). Of course one must be careful about the assumptions on $\xi(t)$ and $\eta(t_k)$ that were introduced in section 1.1. Typically $\xi(t)$ is far from Gaussian- distributed, but exhibits a bias that can be related to an ill-defined model structure or structural incompleteness of $f(x, u, \alpha; t)$. If the sequence $\xi(t)$ would be truly Gaussian, then the parameter estimates generated by a combined state-parameter estimation algorithm such as the extended Kalman filter (EKF) would generate unbiased estimates of the state and parameter vectors given that the system model is *linear*. Both assumptions, i.e. *non-linearity of the dynamical description* $f(x, u, \alpha; t)$ and *non-Gaussian white noise of especially* $\xi(t)$ are important issues that need to be addressed. Also, it is typical that $\xi(t)$ can not be measured directly and its variance has to be specified 'a priori' through $Q(t_0)$, the system noise matrix. Alternative formulations of the Kalman filter are possible such that this matrix can be estimated *on-line* so that no 'a priori' value $Q(t_0)$ needs to be specified – this type of filtering is often referred to as adaptive filtering, e.g. [19].

The innovations-error $\epsilon(t_k)$ is measured more directly, through the difference between the observation $y(t_k)$ and the model's prediction of the same observation $\hat{y}(t_k)$. This error will be interpreted in terms of uncertainties. Four sources of uncertainty with their associated covariance matrices can be mentioned in this respect. They are:

1. model structure error ($Q(t)$)
2. measurement error ($R_1(t_k)$ and $R_2(t_k)$)
3. state uncertainty ($P_{xx}(t_k)$)

4. parameter uncertainty ($P_{\alpha\alpha}(t_k)$)

A major practical problem is that the first and fourth error source in the above are difficult to quantify *'a priori'*. Especially the earlier-mentioned specification of $Q(t)$, the system noise matrix, is a difficult task for the engineer, [3, 18].

1.3 A Formal Definition

The problem of model structure identification has quite a long history. It appeared in environmental systems analysis in the 1970's, [2, 3, 4]. In this field very often the aim is to reconstruct a continuous model $f(x, u, \alpha)$ from a historical data record and 'a priori' knowledge. The approach to a solution for the problem is a somewhat unconventional application of the extended Kalman filter. Instead of using the filter primarily for combined state-parameter estimation, here the results generated by the EKF (including the calculated innovations residuals $\{\epsilon(t_k), k = 0, 1, ...\}$) are used to infer conclusions on the validity or, as Young termed it, the credibility of the model of the badly defined system under study, [28]. Emphasis should be given to the fact that so often in environmental systems analysis the case studies are very poorly defined and it is difficult to deduce any structure in the sparsely collected data. Tools that are discussed in this report have the aim to help the system analyst to complete this task in a satisfying and more systematic way, i.e. less *ad hoc*, than originally in [2, 4, 6].

The model structure identification problem may be defined more specifically. If all the observed (historical) data up to and including time t_k are stacked into one set the following notation can be defined for the output and input sequence respectively:

$$Y(t_k) = \{y(t_0), y(t_1), ..., y(t_k)\} \quad (9)$$

$$U(t_k) = \{u(t_0), u(t_1), ..., u(t_k)\} \quad (10)$$

Further, let N denote the length of the data record. Now the following definition for the *model structure identification* problem can be given, [4]:

> Given the observations $\{Y(t_N), U(t_N)\}$ and given the the various sources of uncertainties, i.e. $\{P_{xx}(t_0)$, $P_{\alpha\alpha}(t_0)$, $R_1(t_0)$, $Q(t_0)$, $R_2(t_0)\}$, determine $[f, h; x, \alpha]$ and the accompanying uncertainties $\{P_{xx}(t_N), P_{\alpha\alpha}(t_N), Q(t_N)\}$.

In other words: The signals $\{Y(t_N), U(t_N)\}$ generated by the system are used to construct or *identify* a model. Very often the state and parameter uncertainties $P_{xx}(t_0)$ and $P_{\alpha\alpha}(t_0)$, as well as the value for the system noise matrix $Q(t_0)$, are not well known 'a priori', so that minimal information on the processes is available. As said before, this is typical for the field of environmental systems analysis.

2 Filtering Theory

2.1 Introduction

The history of filtering theory takes us back to the eminent scientist Carl Friedrich Gauss who developed the least squares method to predict the position and velocity of some of the asteroids that revolve around the sun between the orbits of Mars and Jupiter, [15]. This classical work found its counterpart in a modern approach published in the seminal work of Kalman, [20]. Filtering is an *on-line* estimation method that is used

frequently in signal processing problems. Numerous applications of some form of the Kalman filter, originally derived in the previously mentioned paper [20], have been reported, [2, 5, 9, 10, 11, 13, 16, 17, 18, 21, 27][2]. Typically one studies well defined objects of study that can be defined in 'grey-box' models that have a sound physical foundation. The primary interest in this kind of application is not so much one of model structure identification, but rather one of *combined* state/parameter estimation and this is why the *extended* Kalman filter plays such an important role. The basic (linear) filter was originally developed with emphasis on *state* estimation, [18]. One of the aims of this paper is to show that an algorithm developed in [24], is thought to be better suited for the *identification* of model structure in environmental systems analysis because it is especially developed for parameter estimation. The parameters in the model description (1)-(2) are focal elements that can reveal an incorrect model description when estimated through a recursive 'Kalman-like' algorithm. This technique is not new, [2], but needs a better developed tool that is more suited for the problem of identification. Also, the approach taken in [2] is rather *ad hoc* and needs to be modified so that more systematic identification of environmental models is possible.

2.2 State-Parameter Reconstruction

2.2.1 Introduction

In this introductory section a very brief summary of the extended Kalman filter algorithm will be presented. Other publications, e.g. [1, 16, 18, 24], give a much more detailed description of the mathematics involved. In section 3.2.2 a metaphor is presented that will help the reader in understanding the action of the filter clearly without too much loss of mathematical details.

2.2.2 Some Notation

The extended Kalman filter (EKF) recursively updates the most recent estimate of the so called augmented state vector

$$\hat{x}_a(t_{k+1}|t_k) = [\,\hat{x}^T(t_{k+1}|t_k) \quad \hat{\alpha}^T(t_{k+1}|t_k)\,]^T \tag{11}$$

A note should be made on the notation in (11). With the argument $(t_{k+1}|t_k)$ it is meant that an estimate of the augmented state vector is obtained at the discrete time instant t_{k+1} conditioned upon all the observations that were processed up to and including time instant t_k. The estimate $\hat{x}_a(t_{k+1}|t_k)$ is based on the previous estimate $\hat{x}_a(t_k|t_{k-1})$, and its uncertainty

[2]Note that this is only a small selection from the vast amount of literature that has been written on filtering theory.

$$
\begin{aligned}
P_a(t_k|t_{k-1}) &= E\{(\hat{x}_a(t_k|t_{k-1}) - x_a(t_k))\times \\
&\quad (\hat{x}_a(t_k|t_{k-1}) - x_a(t_k))^T\} \\
&= \begin{pmatrix} P_{xx}(t_k|t_{k-1}) & P_{x\alpha}(t_k|t_{k-1}) \\ P_{x\alpha}^T(t_k|t_{k-1}) & P_{\alpha\alpha}(t_k|t_{k-1}) \end{pmatrix}
\end{aligned} \tag{12}
$$

Here $P_{xx}(t_k|t_{k-1})$ is the $(n \times n)$-dimensional variance matrix of the state vector, $P_{x\alpha}$ $(t_k|t_{k-1})$ is the $(n \times q)$ dimensional covariance matrix of the states and parameters system, and $P_{\alpha\alpha}(t_k|t_{k-1})$ is the $(q \times q)$-dimensional variance matrix of the parameter vector. Also, the new observation $y(t_{k+1})$ that comes available from the data record is included in the estimate. It is easy to show that $P_a(t_k|t_{k-1})$ is positive semi-definite, i.e.

$$
\forall x_a \in \Re^{n+q} \backslash \{0\} : \quad x_a^T P_a(t_k|t_{k-1}) x_a \geq 0 \tag{13}
$$

This property is used for a factorization of the matrix $P_a(t_k|t_{k-1})$ which can be useful with regard to the stability of the filter and/or numerical roundoff errors in the computations involved.

2.2.3 A Modification

In this section a modified version of the extended Kalman filter will be introduced that forms our object of research. It has been shown to be useful in the context of identifiability, and especially an interesting extra element is included in the modified filter that can be useful for the detection of a *failure* of the model structure on a certain part of the data record $Y(t_N)$. The filter has been applied to a case study in [27] that was originally defined and studied in [2]. The case study involves data that were taken from the (non-tidal) lowland River Cam in the United Kingdom. The importance of this revisited case study is that a comparison can be made between the modified filter and the extended Kalman filter as defined in [18]. In [27] a discussion can be found of the results of the modified filter applied to the same data set. The modification is essentially due to Ljung, [23], and is rewritten in continuous-discrete formulation in [26].

Let us first introduce some notation. Let the model of the system not be given by (1)-(2), but by the *innovations form* representation of the system, i.e.

$$
\dot{x}(t) = f(x, u, \alpha; t) + \bar{K}(t)\eta(t) \tag{14}
$$

$$
y(t_k) = h(x, u, \alpha; t_k) + \eta(t_k) \tag{15}
$$

where $\bar{K}(t)$ is the so called *steady state* Kalman gain matrix. Incorporating the ideas of Ljung, [23], the augmented state vector is now augmented again, including the elements of the matrix $\bar{K}(t)$ in the parameter vector $\alpha(t)$. The number of elements of the matrix $\bar{K}(t)$ are nm, so that the augmented state vector (11) has now dimension $(n + q + nm)$. Now define

$$
\bar{K}_a(t_k) = \begin{pmatrix} \bar{K}(t_k) \\ L(t_k) \end{pmatrix} \tag{16}
$$

where $L(t_k)$ is the Kalman gain sub-matrix associated with the parameters of the system. It can be shown that $\bar{K}_a(t_k)$ includes a state transition matrix $\Phi_a(t_{k+1}, t_k)$ that has a special block structure. More specifically, the matrix $\Phi_a(t_{k+1}, t_k)$ is defined by

$$
\Phi_a(t_{k+1}, t_k) = e^{F\Delta t} \tag{17}
$$

with

$$F = \begin{pmatrix} F_{xx} & F_{x\alpha} \\ 0 & 0 \end{pmatrix} \tag{18}$$

$$F_{xx} = \frac{\partial f_i(x_a, u)}{\partial x_{aj}}, \quad i, j = 1, .., n \tag{19}$$

$$F_{x\alpha} = \frac{\partial f_i(x_a, u)}{\partial x_{aj}}, \quad \begin{matrix} i = 1, .., n \\ j = n+1, .., n+q+nm \end{matrix} \tag{20}$$

and $\Delta t = t_{k+1} - t_k$. This leads to the following block representation:

$$\Phi_a(t_{k+1}, t_k) = \begin{pmatrix} \phi_{11}(t_{k+1}, t_k) & \phi_{12}(t_{k+1}, t_k) \\ 0 & I_{q+nm} \end{pmatrix} \tag{21}$$

Further, the observation matrix $H(x, u, \alpha; t_k)$ can be partitioned as the $m \times (n + q + nm)$ matrix

$$H(x, u, \alpha; t_k) = (C(x, u, \alpha; t_k) \quad D(x, u, \alpha; t_k)) \tag{22}$$

where

$$C(x, u, \alpha; t_k) = \frac{\partial h_i(x, u, \alpha)}{\partial x_{aj}}, \quad \begin{matrix} i = 1, .., m \\ j = 1, .., n \end{matrix} \tag{23}$$

$$D(x, u, \alpha; t_k) = \frac{\partial h_i(x, u, \alpha)}{\partial x_{aj}}, \quad \begin{matrix} i = 1, .., m \\ j = n+1, .., n+q+nm \end{matrix} \tag{24}$$

The following algorithm can now be derived, [26]:

$$\hat{x}(t_{k+1}|t_k) = \hat{x}(t_k|t_{k-1}) + \bar{K}(t_k)\epsilon(t_k) + \int_{t_k}^{t_{k+1}} f(\hat{x}(\tau|t_{k-1}), \hat{\alpha}(t_k|t_{k-1}), u(t_k))d\tau \tag{25}$$

$$\hat{\alpha}(t_{k+1}|t_k) = \hat{\alpha}(t_k|t_{k-1}) + L(t_k)\epsilon(t_k) \tag{26}$$

$$\epsilon(t_k) = y(t_k) - h(\hat{x}(t_k|t_{k-1}), \hat{\alpha}(t_k|t_{k-1})) \tag{27}$$

$$L(t_k) = \left[P_{x\alpha}^T(t_k|t_{k-1})C^T(\hat{x}(t_k|t_{k-1}), \hat{\alpha}(t_k|t_{k-1})) + P_{\alpha\alpha}(t_k|t_{k-1})D^T(\hat{x}(t_k|t_{k-1}), \hat{\alpha}(t_k|t_{k-1}))\right] \times \Lambda^{-1}(t_k) \tag{28}$$

$$P_{x\alpha}(t_{k+1}|t_k) = \phi_{11}(t_{k+1}, t_k)P_{x\alpha}(t_k|t_{k-1}) + \phi_{12}(t_{k+1}, t_k) \times P_{\alpha\alpha}(t_k|t_{k-1}) - \bar{K}(t_k)\Lambda(t_k)L^T(t_k) \tag{29}$$

$$P_{\alpha\alpha}(t_{k+1}|t_k) = P_{\alpha\alpha}(t_k|t_{k-1}) - L(t_k)\Lambda(t_k)L^T(t_k) \tag{30}$$

$$\Lambda(t_k) = \frac{1}{t_k}\sum_{j=1}^{k} \epsilon(t_k)\epsilon^T(t_k) \tag{31}$$

It could be noted that, when there is no 'a priori' information on $\Lambda(t_k)$ (a matrix that gives an estimate of the variance of the *innovations sequence* $\{\epsilon(t_k), k = 0, 1, ..., N\}$) available, then it should be estimated by (31). Otherwise it is preferable to fix the value of this matrix to an initial value so that the number of free parameters is minimal. Further discussion of this so called *tuning* of the filter can be found in section 3.2.2. The author developed a MATLAB implementation of algorithm (25)-(31) as part of the project. The source code is included in appendix A.1 of this report.

2.3 Taking the Modification Further

2.3.1 Introduction

When applying algorithm (25)-(31), it appears that serious stability problems arise. Especially worth mentioning is that the main diagonal elements of the covariance matrix $P_{\alpha\alpha}$ become negative when the initial value $P_{\alpha\alpha}(t_0)$ is too large. This effect has also been reported in the literature, [8, 10, 24], and is usually due to round off errors in equation (30) that arise when subtracting two matrices that have entries close to zero. In [8, 10, 24] alternative formulations, namely

- a stabilized filter
- a square root formulation due to Potter
- a UDU^T factorization

are suggested to solve this problem. It is the positive definiteness of the variance covariance matrix $P_a(t_k|t_{k-1})$ that is utilized in these factorizations. However, factorization methods do only seem to be a solution for higher order systems and these kind of system are not used in the current analyses. When looking closer to equation (30), i.e. the propagation of the covariance matrix $P_{\alpha\alpha}$, it appears through numerical experiments that this matrix becomes unstable because of the high values of especially $L(t_k)$. This would also have happened in case of a factorization of the covariance matrix $P_a(t_k|t_{k-1})$. It was found useful to study the book [24] more extensively because it is here where problems related to the implementation of recursive algorithms are discussed in much more detail than in the paper [23].

In the following a stable algorithm that is specially suited for *parameter* estimation is presented. In the context of the model structure identification problem it has attractive potential. Let us emphasize here that the testing of a hypothesis via the interpretation of the parameter trajectories of a recursive identification algorithm is the main theme and our motivation for the research that is conducted in this report, [27]. It is the estimation trajectories of the *parameters* that play a crucial role in the verification of a model. Some of the parameters may be well known from 'a priori' knowledge and the related hypothesis can be defined in very bold and rigid terms. Also, the time invariance of the parameters can be used to identify the structure, [2]. It is therefore believed to be useful to develop a tool that is most suited for estimation of *parameters* in the context of identification. This view also has a sound foundation in the (Popperian) hypothetico-deductive philosophy of science, [22]. In general, the continuous testing of hypotheses is crucial to the progress of science. The model (1) can be seen as a constitutive assembly of hypotheses that need to be tested for their explanatory power. Filtering theory elegantly combines knowledge and data in one framework, [3].

2.3.2 A Unified Approach

In [24] it is shown in a revealing way that most of the recursive (Kalman like) estimation algorithms stem from one and the same *recursive prediction error* algorithm (RPE for short). Also, the numerical deficiencies of equation (30) are explained much better than in the paper [23] and some remedies are suggested to improve the stability of the filter. At this point it should be mentioned that many different algorithms for state-parameter estimation exist. Among these are the extended Kalman filter (EKF), [18], the instrumental variable method (IV), [25], and a Recursive Prediction Error algorithm (RPE), [24].

In the following a modified algorithm especially suited for continuous discrete systems, and more specifically applicable to environmental water quality systems, is developed and tested.

The RPE-algorithm as defined in [24] is not unique. For different tasks different sets of equations can be utilized that have their root in a recursive prediction error algorithm. This 'master' algorithm also needs some minor modifications to make it more applicable in the field of environmental systems analyses where ordinary differential equations are often used. Also, implementation considerations lead to a slightly different algorithm because the stability of the algorithm can be improved using a stabilized version of the variance-covariance equation (35). Some adjustments of the basic algorithm will be presented in the next section. After that, the resulting algorithm will be tested on an artificial set of data and also the 'real-life' Cam 1973 data set will be used for evaluation of the performance of the algorithm in the context of identifiability.

In [24], pg 127, a recursive prediction error (RPE) algorithm is developed starting with a (discrete) innovations form model of the form

$$x(t_{k+1}) = F_{xx}(t_k)x(t_k) + G(t_k)u(t_k) + K(t_k)\epsilon(t_k) \tag{32}$$

where $G(t_k)$ is the input matrix. The RPE-algorithm is summarized below for future reference and also, to introduce some notation.

$$\begin{aligned}
\epsilon(t_k) &= y(t_k) - \hat{y}(t_k) && (33)\\
\hat{\Lambda}(t_k) &= \hat{\Lambda}(t_{k-1}) + \gamma(t_k)\left[\epsilon(t_k)\epsilon^T(t_k) - \hat{\Lambda}(t_{k-1})\right] && (34)\\
R(t_k) &= R(t_{k-1}) + \gamma(t_k)[\psi(t_k)\hat{\Lambda}^{-1}(t_k)\psi^T(t_k) \\
&\quad -R(t_{k-1})] && (35)\\
\hat{\alpha}(t_k) &= \hat{\alpha}(t_{k-1}) + \gamma(t_k)R^{-1}(t_k)\psi(t_k)\Lambda^{-1}(t_k)\epsilon(t_k) && (36)\\
\hat{x}(t_{k+1}) &= F_{xx}(t_k)\hat{x}(t_k) + Gu(t_k) + K(t_k)\epsilon(t_k) && (37)\\
\hat{y}(t_{k+1}) &= H(t_k)\hat{x}(t_{k+1}) && (38)\\
W(t_{k+1}) &= [F_{xx} - K(t_k)H(t_k)]\,W(t_k) + \bar{M}(t_k) \\
&\quad -K(t_k)D(t_k) && (39)\\
\psi(t_{k+1}) &= W^T(t_{k+1})H^T(t_k) + D^T(t_k, \hat{x}(t_{k+1})) && (40)
\end{aligned}$$

where $\gamma(t_k)$ is a series of positive numbers that decreases with order $o(1/t_k)$[3], and

$$\begin{aligned}
\psi^T(t_k) &= \frac{d}{d\alpha}\left[H(\alpha)\hat{x}(t,\alpha)\right]\\
W(t_k) &= \frac{d}{d\alpha}\left[\hat{x}(t,\alpha)\right]\\
\bar{M}(t_k) &= \frac{d}{d\alpha}\left[F_{xx}(\alpha)x(t,\alpha) + G(\alpha)u(t) + \bar{K}(\alpha)\epsilon\right]_{\alpha=\hat{\alpha}(t_k)}
\end{aligned}$$

For an explanation of the matrix $R(t_k)$ the reader is referred to [24]. This is not of interest here, since the matrix is only a way to arrive at the variance-covariance matrix $P_{\alpha\alpha}$ in a later version of the algorithm. Further, some considerations for implementation of the algorithm should be presented that are of concern here. Because of the parametrization of the matrix $\bar{K}$ better convergence properties are guaranteed when compared to the (conventional) extended Kalman filter as defined in [18]. Further, if one uses the so called stabilized Kalman equation a very simple, however, computational more intensive,

[3]Here the decreasing series is taken as $\gamma(t_k) = 1/t_k$.

solution to the problem of stability of the variance covariance matrix $P_{\alpha\alpha}(t_k)$ is achieved. This stabilized equation is straightforward to derive. If one starts with equation (35) and defines

$$\begin{aligned} P_{\alpha\alpha}(t_k) &= \gamma(t_k)R^{-1}(t_k) && (41)\\ S(t_k) &= \psi^T(t_k)P_{\alpha\alpha}(t_{k-1})\psi(t_k) + \hat{\Lambda}(t_k) && (42)\\ L(t_k) &= \gamma(t_k)R^{-1}(t_k)\psi(t_k)\Lambda^{-1}(t_k) && (43) \end{aligned}$$

one can use the *matrix inversion lemma*[4] to arrive at the following equation for the propagation of the variance-covariance matrix:

$$\begin{aligned} P_{\alpha\alpha}(t_k) &= \left[I - L(t_k)\psi^T(t_k)\right] P_{\alpha\alpha}(t_{k-1}) \left[I - \psi(t_k)L^T(t_k)\right] \\ &\quad + L(t_k)\hat{\Lambda}(t_k)L^T(t_k) && (44) \end{aligned}$$

Finally, it could be noted that equation (40) can be modified so as to increase computational efficiency. Instead of calculating ψ it is more efficient to calculate ψ^T because then there is no need to transpose the matrices W, H, and D on the right hand side of (40). The following algorithm is now found:

$$\begin{aligned} \epsilon(t_k) &= y(t_k) - \hat{y}(t_k) && (45)\\ \hat{\Lambda}(t_k) &= \hat{\Lambda}(t_{k-1}) + \gamma(t_k)\left[\epsilon(t_k)\epsilon^T(t_k) - \hat{\Lambda}(t_{k-1})\right] && (46)\\ S(t_k) &= \psi^T(t_k)P_{\alpha\alpha}(t_{k-1})\psi(t_k) + \hat{\Lambda}(t_k) && (47)\\ L(t_k) &= P_{\alpha\alpha}(t_{k-1})\psi(t_k)S^{-1}(t_k) && (48)\\ P_{\alpha\alpha}(t_k) &= \left[I - L(t_k)\psi^T(t_k)\right] P_{\alpha\alpha}(t_{k-1}) \left[I - \psi(t_k)L^T(t_k)\right] \\ &\quad + L(t_k)\hat{\Lambda}(t_k)L^T(t_k) && (49)\\ \hat{\alpha}(t_k) &= \hat{\alpha}(t_{k-1}) + L(t_k)\epsilon(t_k) && (50)\\ \hat{x}(t_{k+1}) &= F_{xx}(t_k)\hat{x}(t_k) + Gu(t_k) + K(t_k)\epsilon(t_k) && (51)\\ \hat{y}(t_{k+1}) &= H(t_k)\hat{x}(t_{k+1}) && (52)\\ W(t_{k+1}) &= \left[F_{xx} - K(t_k)H(t_k)\right] W(t_k) + \bar{M}(t_k) \\ &\quad - K(t_k)D(t_k) && (53)\\ \psi^T(t_{k+1}) &= H(t_k)W(t_{k+1}) + D(t_k, \hat{x}(t_{k+1})) && (54) \end{aligned}$$

Although algorithm (45)-(54) is not new it does not appear to have caught the attention in the environmental sciences. A literature search confirmed this statement. The above algorithm has been implemented in MATLAB as part of the project (appendix A.2). It will be tested on a familiar ARMA time series in the next section.

3 Some Results

3.1 Introduction

Having introduced a filtering algorithm in section2.2.3, the discussion will continue with a test of algorithm (25)-(31) using artificial data. The reason for testing the algorithm more extensively is that one of the results of the (revisited) Cam '73 case study in [27] was that the algorithm showed unstable behavior whenever the main diagonal elements of

[4]This important lemma is stated in appendix B.

the matrices $\{P_{\alpha\alpha}(t_0), \Lambda(t_0)\}$ were specified too large. This was not really an issue at that time, because much of the Cam case study was well known through [2] and especially the 'a priori' values of the various uncertainty matrices and the augmented state vector were well tuned through these results. For future reference it is interesting to mention that the values of the uncertainties in the parameter vector ($P_{\alpha\alpha}(t_0)$) were taken to be very small (e.g. $P_{\alpha\alpha}(t_0) = 0.005$ for a (BOD) decay rate constant). It now appears that this small magnitude is a necessity for the algorithm to work. This will be shown in a later section.

The first objective is to test whether the algorithm indeed generates accurate estimates of both the states and parameters of the system. Hereto a model of the system is defined that is completely equivalent to the (artificial) system that generates the data. A later planned exercise will include simulations that have a structure $f(x, u, \alpha)$ *not* equivalent to the system, i.e. model structure errors ($Q(t)$) are imposed. The latter is indeed often the case in environmental case studies where systems are usually poorly defined.

3.2 Testing the LEKF

3.2.1 A Data Generating System

To test the LEKF for convergence the following very simple system definition is used:

$$\dot{x}(t) = \alpha_1 x(t) + u(t) \tag{55}$$

$$u(t) = \text{sign}(\omega(t_k)) \quad t_{k-1} \leq t < t_k \tag{56}$$

$$y(t_k) = x(t_k) + \epsilon(t_k) \tag{57}$$

where

$$\text{sign(v)} = \begin{cases} -1 & v \leq 0 \\ 1 & v > 0 \end{cases} \tag{58}$$

and $\omega(t_k)$ a (discrete) normally distributed random variable. The value of the measurement noise matrix $R_2(t_k)$ is taken as $R_2(t_k) = R_2 = 0.1$. The constant parameter vector α is defined as $\alpha = (0.8)$. An example of output generated by system (55)-(57) is given in figure 1.

As our main concern here is to test the convergence behavior of the filter the model $f(x, u, \alpha)$ is defined equivalent to the system so that the model reads:

$$\dot{x}(t) = \alpha_1 x(t) + u(t) \tag{59}$$

$$y(t_k) = x(t_k) + \epsilon(t_k) \tag{60}$$

Results of the LEKF are summarized in figures 2 and 3. It is apparent from figure 3 that the algorithm has difficulty in swiftly reconstruction the true values of α. When the 'a priori' estimate for α is too far away from the initial value the algorithm even generates very poor estimates that converge to a completely different value. It is therefore apparent that the only good results are obtained when the initial values for $\hat{\alpha}(t_0)$ are relatively close to the true values α. This does not mean that the LEKF is useless, since the estimation trajectory of the Kalman gain can be useful in finding an incorrectly specified model structure, [27]. The robustness of the filter is, however, rather poor. Further, something more can be said about the stability properties of the LEKF. Through numerical experiments it became clear that

1. The value of the variance matrix $P_{\alpha\alpha}(t_0)$ should be taken to be very small so as to define the model parameters very rigidly. This allows the model to 'survive' the transient effects at the beginning of the simulation.

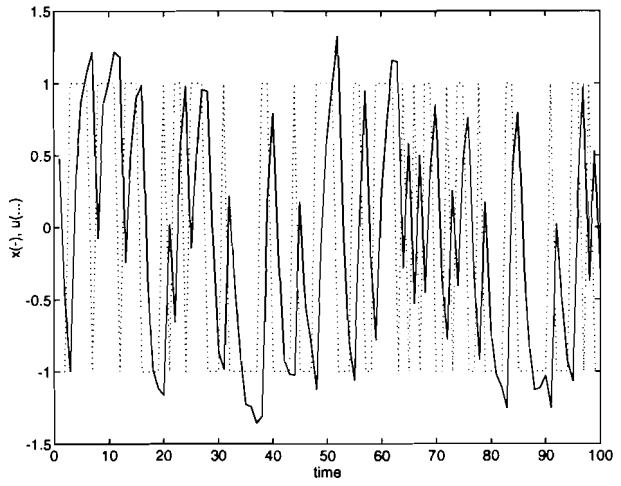

Figure 1: A N=100 size sample of the data generated by the system.

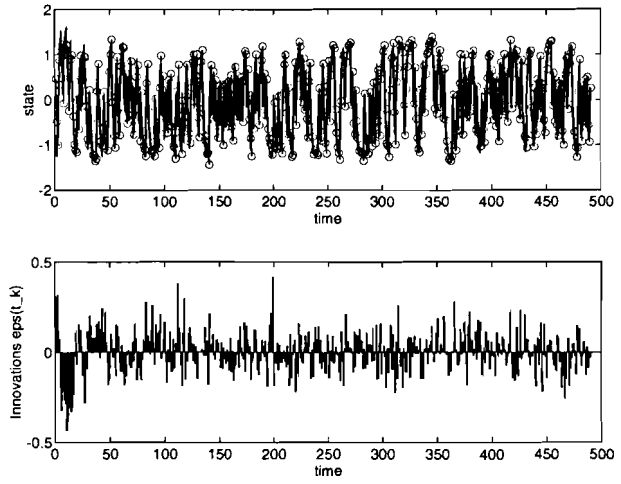

Figure 2: Results of the LEKF applied to the artificial data. Top: state estimates (—) and observations (o). Bottom: The innovations sequence $\epsilon(t_k)$.

2. The value of the innovations variance matrix $\Lambda(t_0)$ should be taken to be large with the result that the initial corrections to the parameter vector $\hat{\alpha}(t_0)$ are small.

3.2.2 A Metaphor

The results so far obtained may be explained using a simple metaphor that increases our insight into algorithm (25)-(31). Suppose the model (1)-(2) is visualized as a clay structure. Further, let us assume that different regions with different densities represent parameter/parameter-uncertainty pairs $\{(\hat{\alpha}_i(t_0), \hat{\sigma}_i(t_0)), i = 1, .., q\}$. The $\hat{\sigma}_i$ can be associated with the flexibility of the clay structure for every $\hat{\alpha}_i$. Hence, the structure has not a uniform flexibility, but rather a heterogeneous structure. A rigid, more brittle region can thus be associated with a highly certain parameter ($\hat{\sigma}_i(t_0) \approx 0.005$) while in the opposite case a highly uncertain parameter corresponds to a flexible part ($\hat{\sigma}_i(t_0) \approx 0.01$). Now imagine that the clay structure is continuously exhibited to stresses that cause the model to change its shape. These stresses are 'caused' by the data $\{Y(t_k)\}$, and more specifically, by the innovations sequence $\{\epsilon(t_k), k = 0, 1, ..., N\}$. Further, one can imagine a shield that surrounds the clay model and weakens the forces/stresses that cause the model to change

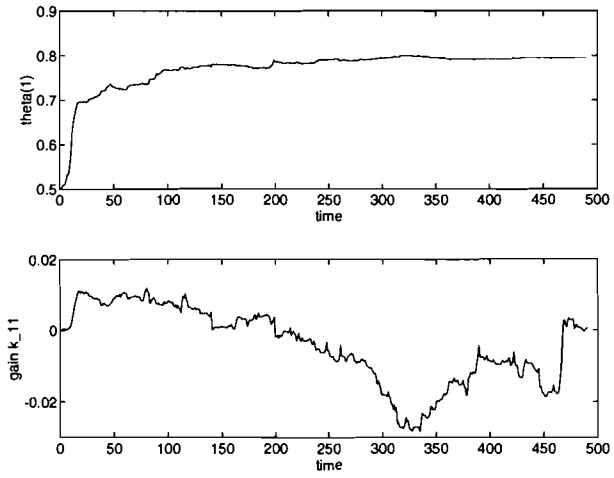

Figure 3: The parameter α_1 and Kalman gain k_{11} estimates.

its structure over time. The 'strength' of this shield can be specified through the value of the matrix $\Lambda(t_0)$. If the main diagonal elements of this matrix are high, this causes the stresses to change the structure weakly, while if the diagonal elements are small, it allows the data to change the shape of the clay structure strongly. Since $\hat{\Lambda}(t_k)$ is estimated on-line its value tends to decrease as the innovation $\epsilon(t_k)$ decreases through adjustment of the parameters so that the data are explained best. Thus, while the filter is learning the stresses acting on the model structure increase. These stresses are counterbalanced by a more rigid model definition through decreasing uncertainties of the parameters. In some cases it can even cause the model to crash, i.e. the stresses are so high that the model collapses and the values of the parameters 'explode' to infinity.

To be more specific, the update step of the parameter vector $\hat{\alpha}$ was previously defined in (26) as

$$\hat{\alpha}(t_{k+1}|t_k) = \hat{\alpha}(t_k|t_{k-1}) + L(t_k)\epsilon(t_k)$$

with $\epsilon(t_k)$ and $L(t_k)$ defined in equations (27) and (28) respectively. The clay-metaphor now helps to understand the behavior of the LEKF better. Note that in equation (28) the matrix $\hat{\Lambda}(t_k)$ is inverted, so that small values for $\hat{\Lambda}(t_k)$ cause *strong* corrections to the values of the parameter vector $\hat{\alpha}(t_{k+1}|t_k)$. In other words: When the stresses shaping the model structure $f(x, u, \alpha)$ are high, the structure changes substantially. The filter simultaniously reduces the uncertainty in the parameter estimates and the structure freezes, as it were, to a final shape that is very rigid. The set of parameters in the model, included in the ordinary differential equation set (1), has a very natural analogy in the metaphor presented in the last section. The 'explanation' for the unstable behavior of the LEKF is that the stresses that sculpture the predefined structure are so high and some of the parameters are so weakly defined that the complete structure crashes. It is especially the first period of the data record in which the model's shape can change dramatically. Initially, the flexibility of the structure is very high and it is as if a small 'earth-quake' shakes the model and changes its structure dramatically during the first iterations of the filter. These so called transient effects are vital to the explanation of the instability of algorithm (25)-(31).

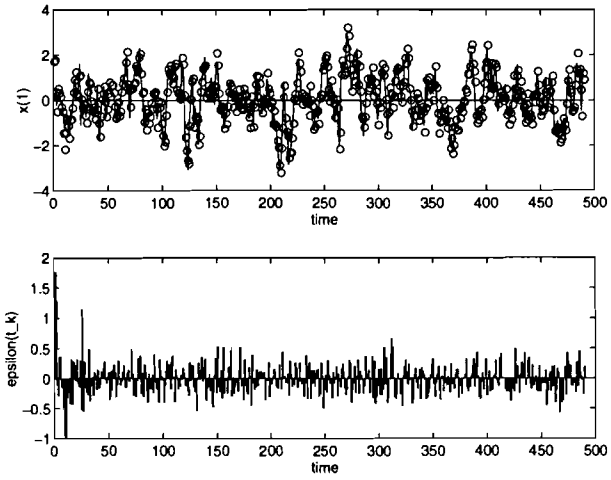

Figure 4: State estimation of an ARMA time series by the RPE-algorithm. Top: Observations (o) and predicted observations (—). Bottom: The innovations sequence.

3.3 Testing the RPE algorithm

3.3.1 A Data Generating System

The RPE-algorithm is tested on an ARMA time series of the following form

$$y(t_k) = \alpha_1 y(t_{k-1}) + \alpha_2 y(t_{k-2}) + \alpha_3 u(t_{k-1}) \tag{61}$$

with $\alpha = (\,1.0 \quad -0.2 \quad 0.6\,)$ and $R(t_k) = R = (\,0.2\,)$. Equation (61) can be rewritten in observable canonical form, i.e.

$$x(t_{k+1}) = \begin{pmatrix} -\alpha_1 & 1 \\ -\alpha_2 & 0 \end{pmatrix} x(t_k) + \begin{pmatrix} \alpha_3 \\ 0 \end{pmatrix} u(t_k) - \begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix} \epsilon(t_k) \tag{62}$$

$$y(t_k) = (\,1 \quad 0\,)\, x(t_k) + \epsilon(t_k) \tag{63}$$

The above representation is in state-space form which is preferable with regard to the RPE-algorithm as formulated in the above. The results for this simulation exercise are summarized in figures 4 and 5. The two figures show that the algorithm is really performing very well. The values for α are recovered fast and also, the initial high uncertainties (equal to one for all parameters) are decreasing rapidly. Also, the innovations error sequence shows white noise characteristics, with a mean close to zero ($\frac{1}{N}\sum_{k=1}^{N} \epsilon(t_k) = -0.0044$), except for the initial simulation interval which is due to initial transient effects. The above results are encouraging and suggest it may be possible to develop the algorithm further to a continuous-discrete formulation that is capable to attack models of the form (1)-(2). This will be done in the next section.

3.3.2 A Continuous-Discrete Formulation

If one is pragmatic about the assumption of white Gaussian noise of $\xi(t)$ then it is not too difficult to include a continous time model into the filtering algorithm. Instead of using a discrete (linear) formulation of the model through discretization of the state propagation equation (1), see e.g. [12], one can include the continuous model through a simple Runge-Kutta procedure that numerically integrates the set of ordinary differential equations. The formulation of the innovations model, however, leads to a problem. Because $\epsilon(t)$ in equation (14) is continuous-time white noise it is expected to drop out of the equation

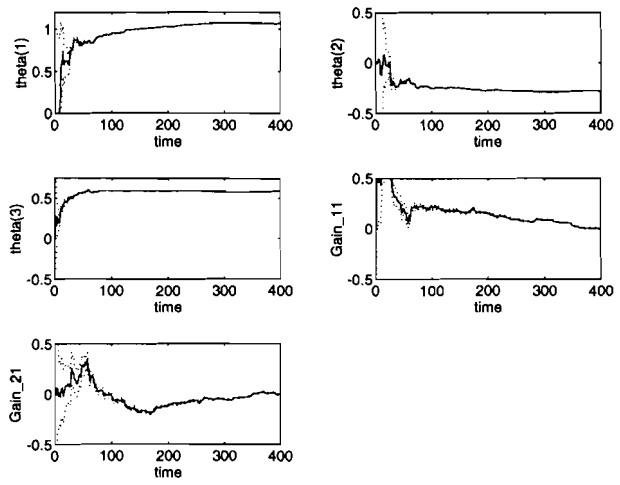

Figure 5: The parameter estimates for α_1, α_2, and α_3, (—) and the Kalman gain elements k_{11} and k_{12}, including estimated uncertainties $P_{\alpha\alpha}(t_k)$ (...). For layout-reasons only the first 400 iterations are included.

when calculating the estimate $\hat{x}(t_k)$ through applying an expectation operator to the original equation (1), see e.g. [18]. This is circumvented pragmatically here and $\epsilon(t_k)$ is therefore taken constant over the simulation interval $[t_k, t_{k+1})$. The following equation now appears for the propagation of the state vector:

$$\begin{aligned} \hat{x}(t_{k+1}|t_k) &= \hat{x}(t_k|t_{k-1}) + \bar{K}(t_k)\epsilon(t_k) + \\ &\int_{t_k}^{t_{k+1}} f(\hat{x}(\tau|t_{k-1}), \hat{\alpha}(t_k|t_{k-1}), u(t_k))d\tau \end{aligned} \tag{64}$$

Finally, the algorithm is completed by substituting $\phi_{11}(t_{k+1}, t_k)$ for F_{xx} and $\phi_{12}(t_{k+1}, t_k)$ for $\bar{M}(t_k)$ following the arguments in [26][5]. Since the algorithm is completely recursive we omit the notation $\cdot\,(t_k|t_{k-1})$ and abbreviate this to t_k only. This yields the following algorithm:

$$\epsilon(t_k) = y(t_k) - \hat{y}(t_k) \tag{65}$$

$$\hat{\Lambda}(t_k) = \hat{\Lambda}(t_{k-1}) + \gamma(t_k)\left[\epsilon(t_k)\epsilon^T(t_k) - \hat{\Lambda}(t_{k-1})\right] \tag{66}$$

$$S(t_k) = \psi^T P(t_{k-1})\psi + \Lambda(t_k) \tag{67}$$

$$L(t_k) = P(t_{k-1})\psi(t_k)S^{-1}(t_k) \tag{68}$$

$$\begin{aligned} P(t_k) &= \left[I - L(t_k)\psi^T(t_k)\right] P(t_{k-1}) \left[I - \psi(t_k)L^T(t_k)\right] \\ &\quad + L(t_k)\hat{\Lambda}(t_k)L^T(t_k) \end{aligned} \tag{69}$$

$$\hat{\alpha}(t_k) = \hat{\alpha}(t_{k-1}) + L(t_k)\epsilon(t_k) \tag{70}$$

$$\begin{aligned} \hat{x}(t_{k+1}) &= \hat{x}(t_k) + K(t_k)\epsilon(t_k) + \\ &\int_{t_k}^{t_{k+1}} f(\hat{x}(\tau), \hat{\alpha}(t_k), u(t_k))d\tau \end{aligned} \tag{71}$$

$$\hat{y}(t_{k+1}) = H(t_k)\hat{x}(t_{k+1}) \tag{72}$$

$$\begin{aligned} W(t_{k+1}) &= \left[\phi_{11}(t_{k+1}, t_k) - K(t_k)H(t_k)\right] W(t_k) + \phi_{12}(t_{k+1}, t_k) \\ &\quad - K(t_k)D(t_k) \end{aligned} \tag{73}$$

$$\psi(t_{k+1}) = W^T(t_{k+1})H^T(t_k) + D^T(t_k, \hat{x}(t_{k+1})) \tag{74}$$

[5]Note that ϕ_{11} and ϕ_{12} were defined in equation (21).

Algorithm (65)-(74) has been tested on artificial data and it was found that the results indeed recover the parameters very well. Because the behavior of the continuous-discrete RPE algorithm is similar to the previous test case the test results for the continuous discrete algorithm have been omitted in this report.

3.4 Application to the Cam-1973 data

3.4.1 Introduction

As a final exercise the continuous-discrete algorithm (65)-(74) is applied to a familiar case study. This case study includes a collection of water quality data taken from the river Cam (UK) in 1973 as part of [2]. The data have been discussed extensively in this reference, and also in e.g. [7]. In short the question is how to validate a set of candidate models that describe the processes that influence the downstream BOD concentration of a predefined 4.7 km strech of the river Cam. This set of models is increasingly more sophisticated and includes a mass-balance, a decay of BOD due to satisfaction of BOD modeled as a first order chemical decay process, a persistent bias in the BOD concentration due to e.g. BOD increase caused by chemical reactions in the suspended solids, and an increase of BOD due to growth of an algae population in the river water during long sunny weather conditions. How these processes are included in a differential equation of the form (1)-(2) is discussed extensively in e.g. [2]. The main interest here is to study the behavior of the recursive (continuous-discrete) prediction error algorithm. This is also to make a comparison between the results in [27] and the new type of filter.

3.4.2 BOD Model I

The first model is known to be rather poorly defined. It includes a mass balance over the sytem and a decay of BOD is assumed that behaves as a first order chemical decay reaction with decay rate constant k_1. The model reads

$$\dot{x}(t) = -(k_1 + \frac{Q(t)}{V})x(t) + \frac{Q(t)}{V}u(t) \tag{75}$$

where $Q(t)$ is the flow rate in the river stretch, V is the volume of the body of water in the defined system (assumed constant over time), $u(t)$ is the BOD concentration upstream (mg/l), and $x(t)$ is the BOD concentration at the downstream border of the stretch (mg/l). It is assumed for simplicity that

$$\forall t \in [t_k, t_{k+1}) : \quad u(t) = u(t_k)$$

The results summarized in figures 5-6 show clearly that the model is badly defined over the middle part of the simulation period. It is here where the gain matrix $\bar{K}(t_k)$ suddenly increases to a constant value of approximately 0.8 (see figure 7), implying an 'injection' of uncertainty in the model structure definition. Also, it is clear from the first parameter estimation trajectory that k_1 is far from constant, but deflects to a substantially lower value than estimated initially.

3.4.3 BOD Model II

The discussion is continued with the definition of a second model that includes a so called low-pass filter that simulates the increase of BOD due to an algae bloom in the river water

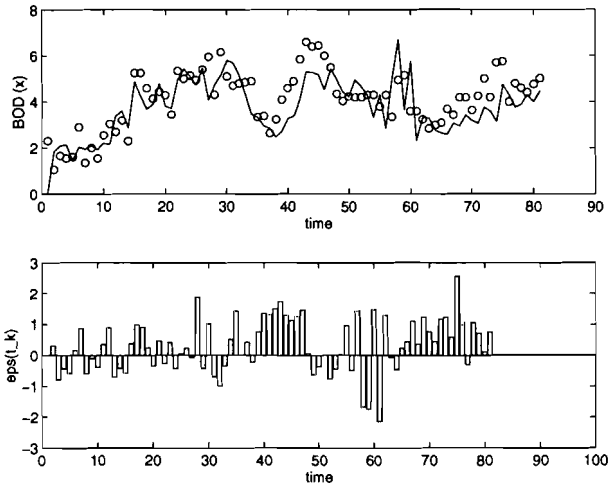

Figure 6: Estimates of observed and predictied BOD concentration, including the innovations sequence $\epsilon(t_k)$ for BOD model I.

during long sunny weather conditions. The model reads

$$\begin{aligned} \dot{x}(t) &= -(k_1 + \frac{Q(t)}{V})x(t) + \frac{Q(t)}{V}u(t) \\ &\quad +a_1 * (I(t_k) - \bar{I}) \qquad (76) \\ I(t_k) &= I(t_{k-1}) + \frac{1}{\tau}(u_2(t_k)\frac{(\alpha - \bar{\alpha})}{\bar{\alpha}} - I(t_{k-1})) \qquad (77) \\ I(t_k) - \bar{I} &= 0 \text{ if } I(t_k) < \bar{I} \qquad (78) \end{aligned}$$

where $I(t_k)$ is derived from the low-pass filter (77) and reflects the addition of BOD due to growth/death of an algae population, see also [27]. Further a_1 is a 'sunlight'-parameter associated with the process of growth/decay of algae. It is estimated on line by the filter and assumed constant. $\bar{I}$ is specified 'a prior' and reflects a threshold level above which the process becomes truly active. Results for the model are summarized in figures 8 and 9. The conclusion that the model better simulates the processes involved in the BOD dynamics in the river water is confirmed through this estimation process. The reader is referred to e.g. [7] for a more detailed discussion of this model validation process. Suffice to say here is that the filter clearly shows an improvement of model structure. The gain estimates (figure 9) are more constant and closer to zero during the whole simulation interval of 80 days when compared to the results of model II (see 7). Also the parameter α_1 shows truly constant behavior which confirms our expectation and contributes to a credible model structure definition. The results discussed are of preliminary nature. One should realize that the innovations sequence also plays an important role in the judgement of the model structure(s) that have been proposed. Current research is elaborating on this and attempt to include these results in a systematic approach to the problem of model structure identification.

4 Conclusions

In this working paper three filtering algorithms have been tested, namely

- A modified continuous-discrete extended Kalman filter as defined in [26].

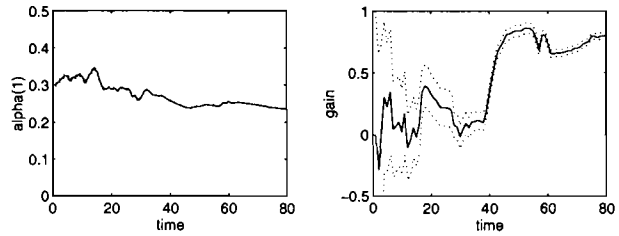

Figure 7: Estimates of parameters α_1 and the Kalman gain $\bar{k}_{11}$ for the BOD model I.

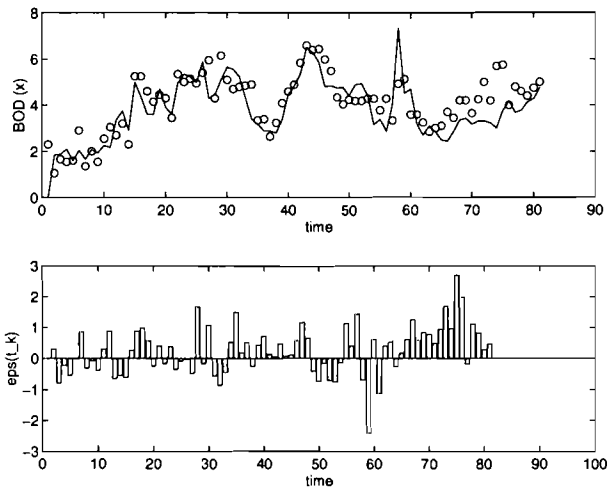

Figure 8: Estimates of observed and predictied BOD concentration, including the innovations sequence $\epsilon(t_k)$ for BOD model II.

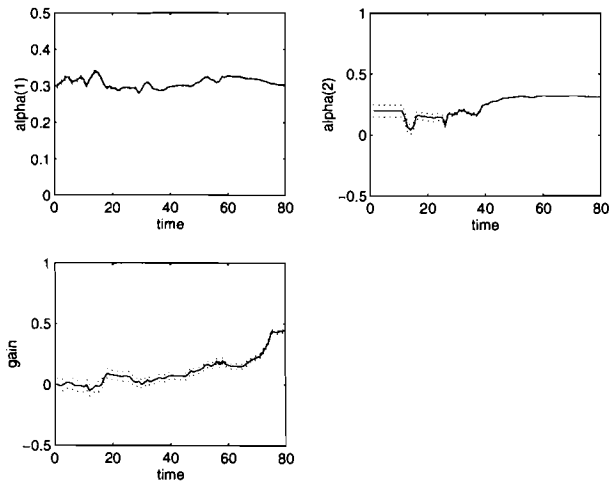

Figure 9: Estimates of parameters α_1, a_1, and the Kalman gain $\bar{k}_{11}$.

- A discrete recursive prediction error algorithm based on an innovations representation of the system, [24].
- A continous-discrete prediction error algorithm, also based on an innovations representation of the system.

First, the continuous-discrete LEKF as defined in [27] was implemented in MATLAB. The stability problems with this type of filter have been resolved using an alternative recursive prediction error algorithm that has excellent stability properties. Also this type of algorithm is well suited for the identification of environmental models because it is especially written for parameter estimation that can be related to the hypotheses that are included in the model structure $f(x, u, \alpha)$. This progress is of practical importance because not only the algorithm is capable of estimating the parameters in the model, but also, the 'a priori' information necessary to iterate the algorithm is kept to a minimum so that the system analyst only needs to specify the most necessary information, i.e. $\hat{x}_a(t_0)$,$f(x, u, \alpha)$, and possibly $\hat{\Lambda}(t_0)$ and $P(t_0)$. Obviously, the specification of $f(x, u, \alpha)$ is most important in the context of the model structure identification problem and the question arises whether the 'true' model structure is specified. It is believed that through the results generated by the RPE continuous-discrete algorithm this question can be solved heuristically. A suggestion for further research is to address more case studies as to observe how this algorithm achieves in the context of identifiability.

A Two MATLAB programs

This appendix includes two algorithms that were programmed in MATLAB. One should notice the efficiency and the shortness of the source code due to the compact representation of the matrices. The core of the program is almost a word to word repetition of the mathematical symbols.

A.1 The LEKF Algorithm in MATLAB

```
% Load the data file
load 'cam.dat';
% Initialization of the filter
% n1 == length of the state vector
% n2 == length of the model parameter vector (alpha)
%  m == length of the observation vector
% n3 == n2 + n1*m, i.e. the length of the parameter vector
%       (alpha prime) that now includes the Kalman gain
%       matrix elements K(t_k).
%  n == n1 + n3 length of the augmented state vector,
%       including the gain matrix elements
n1=1;n2=2;m=1;
% calculate the rest
n3=n2+n1*m;n=n1+n3;

% size of the data file
i=size(cam);i=i(1,1);tN=cam(i,1);clear i;

% Fixed contstants that are the same
% for all of the simulation time
V=1.51;
% 'A priori' Statistics for t=tstart
xhat =[2.3]';
theta=[0.31 0 0]';

% variance covariance matrices
P2=zeros(n1,n3);
P3=diag([0.005,0.25,0.05]);
R =diag([0.4]);
K =theta(n2+1:length(theta));
C =eye(n1);
D =zeros(n1,n3);

i=size(cam);timeend=i(1,1);
% specification of the initial inputs
data1=cam(1,:);t1=data1(1);

% Declare these variables global for the ode23 function
global V Q theta u;

state=fopen('state.out','w');
param=fopen('param.out','w');

% Start of the Filter Loop
for i=1:timeend-1;

% Shift the data vector data0 and update
data0=data1;
data1=cam(i+1,:);
```

```
Q=data0(3);
u=data0(5);
y=data1(8);
t0=t1;t1=data1(1);

% The Runge-Kutta integration (prediction) of the system model
[T,xhat] = ode23('cam2',t0,t1,xhat); xhat=xhat(length(xhat));

%epsilon is the difference between model output and observation
epsilon=y-xhat;

% Update the Jacoby matrix
% This file specifies the F11 and F12 blocks
% derived from the function f(x,u,alpha;t)
F11=[-(theta(1) + Q/V)];
F12=[-xhat 1 epsilon];
% Calculate the State Transition Matrix
Phi=expm([F11 F12; zeros(n3,n)]);
phi11=Phi(1:n1,1:n1);
phi12=Phi(1:n1,n1+1:n);

L=(P2'*C'+P3*D')*inv(R);
LT=L'; % (Save on computation time)
xhat=xhat+K*epsilon;
theta=theta+L*epsilon;
P2=phi11*P2+phi12*P3-K*R*LT;
P3=P3-L*R*LT;
% R=R+1/(t1+1)*(epsilon*epsilon'-R)+0.005*eye(m);
K=theta(n2+1:length(theta));

% write estimates to file
fprintf(state,'%12.8f\n',xhat);
fprintf(param,'%12.8f %12.8f %12.8f\n',theta);
% close the for loop
end;
fclose('all');
```

A.2 The RPE Algorithm in MATLAB

```
% Recursive Prediction Error Identification Algorithm
% Load the data file
load 'cam.dat';
load 'sunlight.dat';
% Initialization of the filter
% n1 == length of the state vector
% n2 == length of the model parameter vector (alpha)
%  m == length of the observation vector
% n3 == n2 + n1*m, i.e. the length of the parameter vector
%       (alpha prime) that now includes the Kalman gain
%       matrix elements K(t_k).
%  n == n1 + n3 length of the augmented state vector,
%       including the gain matrix elements
n1=1;n2=2;m=1;
% calculate the other dimensions
n3=n2+n1*m;n=n1+n3;

% 'A priori' Statistics for t=tstart
xhat  =[2]';
```

```
theta =[0.3 0.2 0]';%zeros(n3,1);
K     =[theta(n2+1:n3)];
C     =eye(n1);
D     =zeros(1,n3);
Lambda=diag([0.5]);
P     =diag([0.005,0.05,0.05]);
psi   =zeros(n3,m);
W     =zeros(n1,n3);

% Define any global constants here
V=1.51;Ibar=6.0;
global theta u Q V sun;

% Find the lenght of the columns in the data file for later use
timeend=length(cam(:,1));
tN=cam(timeend);

% Initialize memory storage
state =zeros(timeend,n1);
param =zeros(timeend-1,n3);
pvar  =zeros(timeend-1,n3);
mnoise=zeros(timeend,m);
inno  =zeros(timeend,m);

% specification of the initial inputs
yhat=xhat(1);

% forgetting factor
lambda=1;

% Start of the Filter Loop
for i=2:timeend;

% Define u and y
y=[cam(i-1,8)];
u=[cam(i,5)];
Q=cam(i,3);
sun=sunlight(i-1)-6.0; if sun<0, sun=0;end;

gamma=1/i;

% epsilon is the difference between model output and observation
epsilon=y-yhat;
Lambda=Lambda+gamma*(epsilon*epsilon'-Lambda);
S=psi'*P*psi+lambda*Lambda;
L=P*psi*inv(S);
U=eye(n3)-L*psi'; % Save on computations
P=U*P*U'/lambda+L*Lambda*L';
theta=theta+L*epsilon';
K=[theta(n2+1:n3)];

% Predict the value of xhat
[T,xhat]=ode23('cam3',cam(i-1,1),cam(i,1),xhat);
xhat=xhat(length(xhat))+K*epsilon;
F11=[-(theta(1)+Q/V)];
F12=[-xhat sun epsilon];
Phi=expm([F11 F12;zeros(n3,n)]);
phi11=Phi(1:n1,1:n1);
phi12=Phi(1:n1,n1+1:n);
```

```
yhat=C*xhat;
W=(phi11-K*C)*W+phi12-K*D;
psi=(C*W+D)';

% store results
state(i,1)=yhat;
param(i-1,:)=theta';
mnoise(i)=Lambda;
pvar(i-1,:)=diag(P)';
inno(i,:)=epsilon';

% close the for loop
end;

% show the results
show;
```

B The Matrix Inversion Lemma

The following lemma is taken from [24].

> **Lemma 1** (Matrix Inversion Lemma): Let A, B, C, and D be matrices of compatible dimension, so that the product BCD and the sum $A+BCD$ exist. Then
>
> $$[A+BCD]^{-1} = A^{-1} - A^{-1}B\left[DA^{-1}B + C^{-1}\right]^{-1} DA^{-1}$$

References

[1] A. Bagchi. *Optimal Control of Stochastic Systems.* Prentice Hall, London, UK, 1993.

[2] M.B. Beck. *The Application of Control and Systems Theory to the Problems of River Water Pollution.* PhD thesis, Kings College, University of Cambridge, 1974.

[3] M.B. Beck. Water Quality Modeling: A Review of the Analysis of Uncertainty. *Water Resources Research*, 23(8):1393–1442, 1990.

[4] M.B. Beck. Understanding Uncertain Environmental Systems. In J. Grasman and G. van Straaten, editors, *Predictabilty and Non-Linear Modeling in Natural Sciences and Economy.* Kluwer, Dordrecht, 1994.

[5] M.B. Beck and E. Halfon. Uncertainty, Identifiability and the Propagation of Prediciton Error: A Case Study of Lake Ontario. *Journal of Forecasting*, 10:135–161, 1990.

[6] M.B. Beck, F.M. Kleissen, and H.S. Wheater. Identifying Flow Paths in Models of Surface Water Acidification. *Reviews of Geophysics*, 28(2):207–230, 1990.

[7] M.B. Beck and P.C. Young. Systematic Identification of DO-BOD Model Structure. *Journal of the Environmental Engineering Division*, 102, EE5:902–927, 1976.

[8] G.J. Bierman. *Factorization Methods for Discrete Sequential Estimation.* Mathematics in Science and Engineering, Academic Press, Orlando, Florida, 1977.

[9] W.P. Budgell. Stochastic Filtering of Linear Shallow Water Wave Processes. *Siam J. Sci. Stat. Comput.*, 8(2):152–170, 1987.

[10] C.L. Chiu. *Applications of Kalman Filtering to Hydrology, Hydraulics, and Water Resourses.* University of Pittsburgh, 1978.

[11] B.J. Cosby and G.M. Hornberger. Identification of Photosynthesis-light Models for Aquatic Systems. I. Theory and Simulations. *Ecological Modelling*, 23:1–24, 1984.

[12] R.C. Dorf. *Time-Domain Analysis and Design of Control Systems.* Addison-Wesley, Reading, Mass., 1965.

[13] U. Eigbe. *State and Parameter Estimation in Groundwater Flow Problems by Kalman Filtering.* PhD thesis, Dept. of Civil Engineering, Imperial College, London, England, 1994.

[14] T.C. Gard. *Introduction to Stochastic Differential Equations* volume 114, 234 pp., 1988.

[15] C.F. Gauss. *Theoria Motus Corporum Coelestium*, volume xi. 1809.

[16] A. Gelb. *Applied Optimal Estimation.* M.I.T. Press, Massachusetts, Cambridge, 1974.

[17] W.D. Graham and D.B. McLaughlin. A Stochastic Model of Solute Transport in Groundwater: Application to the Borden, Ontario, Tracer Test. *Water Resources Research*, 27(6):1345–1359, 1991.

[18] A.H. Jazwinski. *Stochastic Processes and Filtering Theory.*, volume 64. Mathematics in Science and Engineering, Academic Press, NY, 1970.

[19] A.H. Jazwinski. Nonlinear and Adaptive Estimation in Re-entry. *AIAA*, 11, No. 7:922–926, 1973.

[20] R.E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *American Society of Mechanical Engineering Transactions, Series D: Journal of Basic Engineering*, 82:35–45, 1960.

[21] H. Koppelman and T. Schilperoort. Identification and Parameter Estimation of Phosphate Balance Models in Drinking Water Reservoirs. In O.D. Anderson, J.K. Ord, and E.A. Robinson, editors, *Time Series Analysis: Theory and Practice 6.* North-Holland, 1985.

[22] T.S. Kuhn. *The Structure of Scientific Revolutions.* University of Chicago Press, United States of America, second edition, 1970.

[23] L. Ljung. Asymptotic Behaviour of the Extended Kalman Filters as a Parameter Estimator for Linear Systems. *IEEE Transactions on Automatic Control*, 24:36–50, 1979.

[24] L. Ljung and T. Söderström. *Theory and Practice of Recursive Identification.* M.I.T. Press, Massachusetts, 1983.

[25] T. Söderström and P.G. Stoica. *Instrumental Variable Methods for System Idenfification.* Springer-Verlag, 1983.

[26] J.D. Stigter and M.B. Beck. An Innovations Form of Filter for Continuous-Discrete Systems. Technical report, Daniel B. Warnell School of Forest Resources, University of Georgia, Athens, Georgia, 1994a.

[27] J.D. Stigter and M.B. Beck. A New Approach to the Identification of Model Structure. *Environmetrics*, 5(3):315–333, September, 1994b.

[28] P.C. Young. The Validity and Credibility of Models for Badly Defined Systems. In M.B. Beck and G. van Straaten, editors, *Uncertainty and Predictabilty of Environmental Systems.* Wiley InterScience, 1982.