Recursive Prediction Error Identification of Nonlinear State Space Models

Torbjörn Wigren

Systems and Control, Department of Information Technology, Uppsala University, PO Box 337, SE-75105 Uppsala, SWEDEN. torbjorn.wigren@it.uu.se

Key Words - Identification; non-linear systems; state-space models; prediction error methods; scaling; sampling.

Abstract

A recursive prediction error algorithm for identification of systems described by nonlinear ordinary differential equation (ODE) models is presented. The model is a MIMO ODE model, parameterized with coefficients of a multi-variable polynomial that describes *one component* of the right hand side function of the ODE. It is explained why such a parameterization is a key to obtain a well defined algorithm, that does not suffer from singularities and over-parameterization problems. Furthermore, it is proved that the selected model can *also* handle systems with more complicated right hand side structure, by identification of an input-output equivalent system in the coordinate system of the selected states. The linear output measurements can be corrupted by zero mean disturbances that are correlated between measurements and over time. The disturbance correlation matrix is estimated on-line and need not be known beforehand. The algorithm is applied to live data from a system consisting of two cascaded tanks with free outlets. It is illustrated that the identification algorithm is capable of producing a highly accurate nonlinear model of the system, despite the fact that the right hand structure of the system has two nontrivial nonlinear components. A novel technique based on scaling of the sampling period that significantly improves the numerical properties of the algorithm is also disclosed.

1. INTRODUCTION

System identification based on linear models is today well established in research and in practice, a fact that manifests itself in widely used software packages like the MATLAB System Identification Toolbox. Many methods have been extended to deal also with nonlinear models. Algorithms and available theoretical results are however more scattered in the nonlinear field. There are some strong reasons for this.

First, it is more difficult to find models with a wide validity, although general frameworks can be formulated (Ljung, 1997). One approach denoted gray-box modeling combines physical modeling, numerical integration and parameter estimation for estimation of physical parameters in differential equation models, ordinary (Bohlin, 1994) and partial (Funkquist, 1994). Algorithms based on general state space models are also discussed in Ljung and Glad (2004). In other situations black box identification, without direct physical model integration, is applied. Many models and methods have been proposed, see e.g. Sjöberg *et al.*, (1997) for a survey. The most simple methods are perhaps the block-oriented algorithms that are based on cascaded linear dynamic and static nonlinear blocks (Billings and Fakhouri, 1982). The Hammerstein and the Wiener model (Westwick and Verhagen, 1996) based algorithms belong to this class. Other common approaches for nonlinear system identification include the use of neural networks (Chen and Billings, 1992; Sjöberg and Ljung, 1992), fuzzy models and the non-linear dynamic models based on difference equations of e.g. NARMAX (Chen and Billings, 1989), NARX and NFIR type, cf. Ljung, (1997). The fundamental methods based on Wiener and Volterra series (Wiener, 1958; Schetzen, 1980) should also be mentioned here.

Given the model structure, the choice of algorithm (optimization algorithm) is often more complicated than in the linear case, where least squares techniques can often be applied. For nonlinear models least squares modeling can also sometimes be used, and then the recommendation is to try this approach first (Ljung, 1997). In gray-box modeling it is not uncommon to end up with models where the important parameters enter nonlinearly, e.g. due to underlying physical principles (Bohlin, 1994; Ekstam and Smed, 1987). All quantities needed for a least squares formulation may not

be measurable either, a situation that is often handled by the introduction of output error models. Other algorithmic issue that needs to be handled with some care include regularization (Sjöberg and Ljung, 1992; Ljung, 1997) and scaling (Ekstam and Smed, 1987). Input signal properties required for excitation of nonlinear systems are also more complicated than in the linear case (Wigren, 1993, 2003).

A third reason for the difficulty of nonlinear system identification is that fundamental properties, e.g. stability, is dramatically more difficult to handle when the model dynamics is nonlinear. This affects the performance and usefulness of algorithms as well as the theoretical analysis. Further complications on the theoretical side involve statistical properties, like stationarity, whose validity is crucial in attempts to apply theoretical analysis. Available results in this field are often based on extensions from linear system identification, (Ljung ,1977ab; Ljung and Söderström 1983). There are also general results like Ljung, (1975) available, but there application often require that fundamental properties like stability and statistical stationarity are either verified or assumed.

Naturally, a main driver for the development of algorithms for identification of nonlinear systems is the large number of economically important applications. This includes power system components (Ekstam and Smed, 1987), continuous pulp digesters (Funkqvist, 1994), diesel generators (Billings *et al.*, 1988), drum boilers (Åström and Eklund, 1975) and pH-control systems (Pajunen, 1992), just to mention a few.

With the above background the contributions of the present work can be stated. The report exploits a restricted parameterization of a nonlinear state space model, where only one component of the right hand side of the ODE is parameterized. It is explained that such a restricted parameterization is a key to obtain a nonlinear black box identification problem that reduces the risk for over-parameterization. The remaining components of the ODE are selected as a series of integrators, integrating the parameterized component. This choice of state is directly applicable to all multiple input single output (MISO) nonlinear ODEs formulated in input-output form. Physical modeling of electrical, mechanical and chemical systems often results in such models. However, when physical modeling is performed for the purpose of control it is often the case that processes that are

parts of the plant are modeled independently, cf. the cascaded tanks of section 5. In such a case a block right hand side of a nonlinear state space ODE is naturally obtained i.e. there are several components of the ODE that have nonlinear, nontrivial right hand sides, cf. Ljung and Glad (2004). At first glance it hence seems that the proposed restricted parameterization may not be general enough in practice.

The first main contribution of the report proves that the proposed restricted model structure is locally applicable for description of state space ODE models with a general right hand side structure. In other words, even if e.g. physical modeling suggests a complicated right hand side structure of a nonlinear ODE, the proposed parameterization of a single component of the ODE can be expected to provide useful identification results, albeit in another coordinate system than the one used for physical modeling.

The second contribution is the algorithm, which is of recursive prediction error (RPEM) type. It is based on a discretized version of the above output error ODE model. Discretization is performed by an Euler difference scheme. The advantage with this is that a direct connection to the continuous time physical parameters is retained, whenever such a relation is at hand. The parameterization of the right hand side component is based on a multi-variable polynomial in the states and inputs. The formulation of the scheme allows for multiple linear output measurements, i.e. the approach can be used for recursive identification of MIMO nonlinear state space models. Measurement noise levels and the noise correlation matrix are estimated on line and need not to be known beforehand.

A third contribution is constituted by a novel scheme utilizing scaling of the sampling period when running the algorithm. The scaling algorithms exploits the fact that the sampling period appears explicitly and linearly in proposed algorithm. The operation of the scheme is discussed and it is shown that it results in an exponential (in the model order) scaling of the states of the proposed model. This is attractive, since algorithmic problems often originate from poor scaling of the generated signals in the identification algorithm. The scheme results in significant performance improvements in experimental runs.

Advantages as compared to existing techniques includes the use of a smaller number of parameters than what is needed when some other black box models like Volterra series expansions and neural networks are applied. This is due to the structure imposed by the proposed black box parameterization. The structure also makes over-parameterization less likely and reduces the need for regularization that is a necessity when e.g. neural networks are applied to system identification (Sjöberg and Ljng, 1992). This is important, since parameterizations for nonlinear systems are often difficult to analyze and since it is often not known when a certain parameterization will result in a singular identification problem. Secondary benefits should include complexity advantages as well as possibly better accuracy as stated by the parsimony principle. The algorithm is also applicable to state space models with a general right hand side structure and can also be directly applied on-line. Note that continuous time nonlinear state space models is the basis for many nonlinear controller design methods. It is hence advantageous that the proposed model and algorithm provide estimates that are easy to transform to nonlinear MIMO continuous time state space models. Disadvantages include the risk to converge to local sub-optimal minimum points, a risk that always need to be assessed when output error modeling is used. Furthermore, structural prior information cannot be utilized since a black box approach is used.

The report is organized as follows. Section 2 presents the restricted model structure and proves results about its generality. Section 3 derives the algorithm while section 4 discusses scaling.. Section 5 proceeds with a discussion of the identification of a highly nonlinear double tank process where the algorithm is applied to measured data. The conclusions appear in section 6. Conditions on the model and the algorithm are introduced in the text when needed and they are denoted by C1), C2,…and so on.

2. MODEL STRUCTURAL ASPECTS

To motivate the proposed model structure, some properties of nonlinear ODE models are first addressed. Two main possibilities exist for the formulation of nonlinear ODE models. Either an input-output approach can be taken, or a general nonlinear state space model can be used. As discussed

below the two approaches differ when the input-output model is written in state space form in that only one right hand side nonlinear function needs to be parameterized, as compared to n for the general state space model. At first glance it hence may seem that the input-output based model is less general and that the general nonlinear state space model is the one to base algorithms on. However, there also appears to be advantages associated with the nonlinear input-output model. First, a reduction of the number of parameters seems to be possible. More importantly, a restriction of the general model structure appears to be a necessity in order to avoid over-parameterization and to have a well posed identification problem. This section analyses these issues and provides theoretical motivation for the proposed restricted state space model structure.

The starting point for the derivation of the model that forms the basis for the development of the report is the n :th order ordinary differential equation

$$x^{(n)} = f\left(x,\ldots,x^{(n-1)},u_1,\ldots,u_1^{(n_1)},\ldots,u_k,\ldots,u_k^{(n_k)},\boldsymbol{\theta}\right). \qquad (1)$$

The differential equation is parameterized in terms of the unknown parameter vector $\boldsymbol{\theta}$, consisting of $n_{\boldsymbol{\theta}}$ components (bold face characters denote vectors and matrices). The dependent variable is denoted by $x(t)$, while the input vector is given by

$$\mathbf{u}(t) = \left(u_1(t) \quad \ldots \quad u_1^{(n_1)}(t) \quad \ldots \quad u_k(t) \quad \ldots \quad u_k^{(n_k)}(t)\right)^T. \qquad (2)$$

Continuous time is denoted by t. Differentiation j times is denoted by the superscript $^{(j)}$ in the report.

In order to write (1) in state space form the following selection of states is made

$$x_i = x^{(i-1)}, \quad i = 1,\ldots,n. \qquad (3)$$

This results in the state space model

$$\begin{pmatrix} x_1^{(1)} \\ \vdots \\ x_{n-1}^{(1)} \\ x_n^{(1)} \end{pmatrix} = \begin{pmatrix} x_2 \\ \vdots \\ x_n \\ f\left(x_1,\ldots,x_n,u_1,\ldots,u_1^{(n_1)},\ldots,u_k,\ldots,u_k^{(n_k)},\boldsymbol{\theta}\right) \end{pmatrix}. \qquad (4)$$

Since the observed quantity of interest of (1) is x, the measurement equation is

$$y(t) = \begin{pmatrix} 1 & 0 & \cdots & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \tag{5}$$

where $y(t)$ denotes the measured quantity.

Obviously, the system (4) can describe arbitrary ODEs of the form (1), so it is a nonlinear model that can describe a relatively wide class of nonlinear systems. There are, however, related state space models that do not have the structure of (4), rather the more general structure

$$\begin{pmatrix} x_1^{(1)} \\ \vdots \\ x_{n-1}^{(1)} \\ x_n^{(1)} \end{pmatrix} = \begin{pmatrix} f_1(x_1,\ldots,x_n,u_1,\ldots,u_k,\boldsymbol{\theta}_1) \\ \vdots \\ f_{n-1}(x_1,\ldots,x_n,u_1,\ldots,u_k,\boldsymbol{\theta}_{n-1}) \\ f_n(x_1,\ldots,x_n,u_1,\ldots,u_k,\boldsymbol{\theta}_n) \end{pmatrix}, \tag{6}$$

applies, where the vector $\mathbf{u}(t)$ represents all input signal components to the system. The two main problems addressed in this section are then

P1) How are (4) and (6) related to over-parameterization ?

P2) How general is (4) as compared to the general description (6) ?

Note that when addressing the above problems, it is generally assumed that the objective is to obtain a black box model, i.e. no structure is assumed for the right hand side functions of the components (4) and (6).

The problem P1) can be analyzed by consideration of the linear state space model

$$\begin{pmatrix} x_1^{(1)} \\ \vdots \\ x_{n-1}^{(1)} \\ x_n^{(1)} \end{pmatrix} = \begin{pmatrix} -a_1 & -a_2 & & \cdots & -a_n \\ 1 & 0 & & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & & \vdots \\ 0 & \cdots & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ \\ \vdots \\ x_n \end{pmatrix} + \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix} u$$

$$y = \begin{pmatrix} b_1 & \cdots & b_{n-1} & b_n \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_{n-1} \\ x_n \end{pmatrix} \tag{7}$$

that corresponds to the strictly proper transfer function

$$Y(s) = \frac{b_1 s^{n-1} + \cdots b_{n-1} s + b_n}{s^n + a_1 s^{n-1} + \cdots + a_{n-1} s + a_n} U(s). \qquad (8)$$

Since (7) is a minimal realization in case there is no pole zero cancellation in (8), it is clear that black box parameterizations based on (6) would, in the linear special case, easily result in over-parameterization. As an example, a black box multi-variable polynomial parameterization based on (6) would in the linear special case have a system matrix filled with n^2 independent parameters, as compared to the n parameters of (7) needed to describe the dynamic modes. The dynamics of the restricted model (4) on the other hand, becomes identical with the controllable canonical form (7) in case of a multi-variable polynomial parameterization. Hence, it can be concluded that the model structure (4) should be used rather than (6) in order to avoid over-parameterization.

To analyze the problem P2), the remaining parts of this section sets out to prove that (4) can be locally used to describe the dynamics of (6), by applying a transformation to another coordinate system. The idea of the proof is to rewrite (6) as the input-output model (1), which in turn can be written in the state space form (4). The proof computes differentiated versions of the last component of (6), followed by an application of the implicit function theorem.

To begin, the last component of (6) is assumed to be the observed quantity of interest, i.e. the dependent variable to be used in (1) shall be x_n. This means that the following restriction needs to be imposed

C1) The system (6) is multiple input single output with observed signal x_n.

The last equation of (6) is then differentiated with respect to t. A first differentiation results in

$$x_n^{(2)} = \begin{pmatrix} \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_n} & \frac{\partial f_n}{\partial u_1} & \cdots & \frac{\partial f_n}{\partial u_k} \end{pmatrix} \begin{pmatrix} f_1 \\ \vdots \\ f_n \\ u_1^{(1)} \\ \vdots \\ u_k^{(1)} \end{pmatrix} \equiv f^1\left(x_1, \ldots, x_n, u_1, u_1^{(1)}, \ldots, u_k, u_k^{(1)}, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_n\right). \qquad (9)$$

After $n-1$ differentiation steps, the following relations are obtained

$$\begin{pmatrix} x_n^{(1)} \\ \vdots \\ x_n^{(n-1)} \\ x_n^{(n)} \end{pmatrix} = \begin{pmatrix} f_n\left(x_1,\ldots,x_n,u_1,\ldots,u_k,\boldsymbol{\theta}_1,\ldots,\boldsymbol{\theta}_n\right) \\ \vdots \\ f^{n-2}\left(x_1,\ldots x_n,u_1,\ldots,u_1^{(n-2)},\ldots,u_k,\ldots,u_k^{(n-2)},\boldsymbol{\theta}_1,\ldots,\boldsymbol{\theta}_n\right) \\ f^{n-1}\left(x_1,\ldots x_n,u_1,\ldots,u_1^{(n-1)},\ldots,u_k,\ldots,u_k^{(n-1)},\boldsymbol{\theta}_1,\ldots,\boldsymbol{\theta}_n\right) \end{pmatrix}$$

$$\equiv \begin{pmatrix} \mathbf{f}^{n-2}\left(x_1,\ldots,x_n,u_1,\ldots,u_1^{(n-1)},\ldots,u_k,\ldots,u_k^{(n-2)},\boldsymbol{\theta}_1,\ldots,\boldsymbol{\theta}_n\right) \\ f^{n-1}\left(x_1,\ldots,x_n,u_1,\ldots,u_1^{(n-1)},\ldots,u_k,\ldots,u_k^{(n-1)},\boldsymbol{\theta}_1,\ldots,\boldsymbol{\theta}_n\right) \end{pmatrix}. \qquad (10)$$

In order to prove that (10) can be written as (4), it is now sufficient to write the last equation of (10) as (1), which in turn can be written as (4) using (3). Towards that end, the state variables $x_1,\ldots,x_{n-1}$ need to be eliminated from the last component of (10). The next step of the proof therefore makes use of the implicit function theorem (Khalil, 1996). The implicit function theorem is given by

Lemma 1: [Implicit function theorem] Assume that $\mathbf{f}: R^n \times R^m \to R^n$ is continuously differentiable at each point $\left(\mathbf{y}^T \quad \mathbf{z}^T\right)^T$ of an open set $P \subset R^n \times R^m$. Let $\left(\mathbf{y}_0^T \quad \mathbf{z}_0^T\right)^T$ be a point in P for which $\mathbf{f}\left(\mathbf{y}_0,\mathbf{z}_0\right) = \mathbf{0}$ and for which the Jacobian matrix $\left[\partial\mathbf{f}/\partial\mathbf{y}\right]\left(\mathbf{y}_0,\mathbf{z}_0\right)$ is nonsingular. Then there exist neighborhoods $U \subset R^n$ of $\mathbf{y}_0$ and $V \subset R^m$ of $\mathbf{z}_0$ such that for each $\mathbf{z} \in V$ the equation $\mathbf{f}\left(\mathbf{y},\mathbf{z}\right) = \mathbf{0}$ has a unique solution $\mathbf{y} \in U$. Moreover, this solution can be given as $\mathbf{y} = \mathbf{g}\left(\mathbf{z}\right)$, where $\mathbf{g}$ is continuously differentiable at $\mathbf{z} = \mathbf{z}_0$.

Note that the notation $\mathbf{y}$ and $\mathbf{z}$ does not necessarily refer to the measurement in Lemma 1, they just denote variables used in the formulation of the result in Khalil, (1996)..

The idea is now to solve the nonlinear system of equations obtained from the first $n-1$ equations of (10) for $\left(x_1 \quad \ldots \quad x_{n-1}\right)^T$. The quantities relevant to the application of Lemma 1 are

$$\mathbf{f} = \mathbf{f}^{n-2} - \left(x_n^{(1)} \quad \ldots \quad x_n^{(n-1)}\right)^T : R^{n-1} \times R^{(n-1)+(n-1)k+\sum_{i=1}^{n} n_{\boldsymbol{\theta}_i}} \to R^{n-1} \qquad (11)$$

$$\mathbf{y} = \left(x_1 \quad \ldots \quad x_{n-1}\right)^T \in R^{n-1} \qquad (12)$$

$$\mathbf{z}=\begin{pmatrix} x_n^{(1)} & \dots & x_n^{(n-1)} & u_1 & \dots & u_1^{(n-2)} & \dots & u_k & \dots & u_k^{(n-2)} & \boldsymbol{\theta}_1^T & \dots & \boldsymbol{\theta}_n^T \end{pmatrix}^T \in R^{(n-1)+(n-1)k+\sum_{i=1}^{n} n_{\boldsymbol{\theta}_i}} \qquad (13)$$

By inspection, to apply the implicit function theorem and to obtain a continuously differentiable ODE written as (4) , the following conditions on (6) need to be introduced:

C2) The input signal is $\mathbf{u}(t)$ is $n-1$ times continuously differentiable.

C3) $f_n\left(x_1,\dots,x_n,u_1,\dots,u_k,\boldsymbol{\theta}_n\right)$ is n times continuously differentiable, for all $\begin{pmatrix} x_1 & \dots & x_n & u_1 & \dots & u_k & \boldsymbol{\theta}_n^T \end{pmatrix}^T \in R^{n+k+n_{\boldsymbol{\theta}_n}}$.

C4) $f_i\left(x_1,\dots,x_n,u_1,\dots,u_k,\boldsymbol{\theta}_i\right)$, $i=1,\dots,n-1$, are $n-1$ times continuously differentiable, for all $\begin{pmatrix} x_1 & \dots & x_n & u_1 & \dots & u_k & \boldsymbol{\theta}_i^T \end{pmatrix}^T \in R^{n+k+n_{\boldsymbol{\theta}_i}}$.

C5) $[\partial \mathbf{f}/\partial \mathbf{y}](\mathbf{y}_0,\mathbf{z}_0)$ defined by (11) is nonsingular for $\begin{pmatrix} \mathbf{y}_0^T & \mathbf{z}_0^T \end{pmatrix}^T$ defined by (12) and (13) .

Remark 1: Continuous differentiability is assumed to hold globally in C2), C3) and C4). Relaxed restrictions in line with Lemma 1 are possible, but would complicate the presentation and are therefore not included. The models of this report are polynomial, and hence global differentiability applies.

It follows from the construction of $\mathbf{f}^{n-2}\left(x_1,\dots,x_n,u_1,\dots u_1^{(n-2)},\dots,u_k,\dots,u_k^{(n-2)},\boldsymbol{\theta}_1,\dots,\boldsymbol{\theta}_n\right)$ and from C2), C3) and C4) that $\mathbf{f}^{n-2}\left(x_1,\dots,x_n,u_1,\dots u_1^{(n-2)},\dots,u_k,\dots,u_k^{(n-2)},\boldsymbol{\theta}_1,\dots,\boldsymbol{\theta}_n\right)-\begin{pmatrix} x_n^{(1)} & \dots & x_n^{(n-1)} \end{pmatrix}^T$ is continuously differentiable. For points $\begin{pmatrix} \mathbf{y}_0^T & \mathbf{z}_0^T \end{pmatrix}^T$ where C5) is fulfilled, there are therefore neighborhoods $U \subset R^{n-1}$ of $\mathbf{y}_0$ and $V \subset R^{(n-1)+(n-1)k+\sum_{i=1}^{n} n_{\boldsymbol{\theta}_i}}$ of $\mathbf{z}_0$ such that for each $\mathbf{z} \in V$ the equation

$$\mathbf{f}^{n-2}\left(x_1,\dots,x_n,u_1,\dots u_1^{(n-2)},\dots,u_k,\dots,u_k^{(n-2)},\boldsymbol{\theta}_1,\dots,\boldsymbol{\theta}_n\right)-\begin{pmatrix} x_n^{(1)} & \dots & x_n^{(n-1)} \end{pmatrix}^T=\mathbf{0} \qquad (14)$$

has the continuously differentiable solution

$$\begin{pmatrix} x_1 \\ \vdots \\ x_{n-2} \\ x_{n-1} \end{pmatrix} = \begin{pmatrix} g_1\left(x_n, x_n^{(1)}, \ldots, x_n^{(n-1)}, u_1, \ldots, u_1^{(n-2)}, u_k, \ldots, u_k^{(n-2)}, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_n\right) \\ \vdots \\ g_{n-2}\left(x_n, x_n^{(1)}, \ldots, x_n^{(n-1)}, u_1, \ldots, u_1^{(n-2)}, u_k, \ldots, u_k^{(n-2)}, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_n\right) \\ g_{n-1}\left(x_n, x_n^{(1)}, \ldots, x_n^{(n-1)}, u_1, \ldots, u_1^{(n-2)}, u_k, \ldots, u_k^{(n-2)}, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_n\right) \end{pmatrix}. \quad (15)$$

When the components of (15) are inserted into the last component of (10), the result can be expressed as the following ODE

$$x_n^{(n)} = f^{n-1}\left(g_1\left(x_n, \ldots, x_n^{(n-1)}, u_1, \ldots u_1^{(n-2)}, \ldots, u_k, \ldots, u_k^{(n-2)}, \boldsymbol{\theta}_1, \ldots \boldsymbol{\theta}_n\right), \ldots\right. \quad (16)$$

$$\left., g_{n-1}\left(x_n, \ldots, x_n^{(n-1)}, u_1, \ldots, u_1^{(n-2)}, \ldots, u_k, \ldots, u_k^{(n-2)}, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_n\right), x_n, u_1, \ldots, u_1^{(n-1)}, \ldots, u_k, \ldots, u_k^{(n-1)}, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_n\right)$$

which is of the form (1). Note that since the components of (15) are continuously differentiable, it follows by construction of (10), and by C2), C3) and C4), that the right hand side of (16) is continuously differentiable. This proves

Theorem 1: Assume that C1), C2), C3) and C4) hold and consider the ODE

$$\begin{pmatrix} x_1^{(1)} \\ \vdots \\ x_{n-1}^{(1)} \\ x_n^{(1)} \end{pmatrix} = \begin{pmatrix} f_1\left(x_1, \ldots, x_n, u_1, \ldots, u_k, \boldsymbol{\theta}_1\right) \\ \vdots \\ f_{n-1}\left(x_1, \ldots, x_n, u_1, \ldots, u_k, \boldsymbol{\theta}_{n-1}\right) \\ f_n\left(x_1, \ldots, x_n, u_1, \ldots, u_k, \boldsymbol{\theta}_n\right) \end{pmatrix}.$$

Then, for all points where C5) holds, there are neighborhoods $U \subset R^{n-1}$ of $\begin{pmatrix} x_1 & \ldots & x_{n-1} \end{pmatrix}^T$ and $V \subset R^{(n-1)+(n-1)k+\sum_{i=1}^{n} n_{\boldsymbol{\theta}_i}}$ of $\begin{pmatrix} x_n^{(1)} & \ldots & x_n^{(n-1)} & u_1 & \ldots & u_1^{(n-2)} & \ldots & u_k & \ldots & u_k^{(n-2)} & \boldsymbol{\theta}_1^T & \ldots & \boldsymbol{\theta}_n^T \end{pmatrix}^T$ as well as a coordinate transformation that transforms the ODE to the form

$$\begin{pmatrix} x_1^{(1)} \\ \vdots \\ x_{n-1}^{(1)} \\ x_n^{(1)} \end{pmatrix} = \begin{pmatrix} x_2 \\ \vdots \\ x_n \\ f\left(x_1, \ldots, x_n, u_1, \ldots, u_1^{(n_1)}, \ldots, u_k, \ldots, u_k^{(n_k)}, \boldsymbol{\theta}\right) \end{pmatrix}.$$

Remark 2: As stated above physical modeling often results in ODE:s of the form (6). The implication of Theorem 1 is that such systems can *anyway* be locally modeled by identification algorithms based on the structure (4), as in the remainder of this report.

Remark 3: In case the last state equation of (6) (the main observed quantity) depends on an input signal, the construction of the state transformation of Theorem 1 introduces derivatives of this input signal, a fact that may be practically troublesome. However, in case the last state equation of (6) does not contain any dependence of the input, no derivatives of inputs are introduced in the constructions leading to Theorem 1, cf. (9) - (10). In such a situation no derivatives of inputs are needed in the parameterization (19) - (21) provided that the remaining state equations of (6) does not contain any derivatives of input signals.

3. RECURSIVE PREDICTION ERROR ALGORITHM

In this section a recursive prediction error method (RPEM) of output error type is derived, based on the model (4). The derivation follows the standard approach of (Ljung and Söderström, 1983). The output error approach is needed since application of least squares techniques directly to (4) would require that the complete state vector would be measurable, which is a quite restrictive assumption.

3.1 Parameterization and Discretization

The recursive identification algorithm is assumed to operate on discrete time measured input and output signals, obtained from the system of interest. The input signal vector is given by (2), and the measured output signal is assumed to be

$$\mathbf{y}_m(t) = \begin{pmatrix} y_{m,1}(t) & \dots & y_{m,p}(t) \end{pmatrix}^T. \qquad (17)$$

The input signal derivatives contained in (2) that are sometimes needed (cf. Remark 3) may be measured or generated by numerical differentiation. In order to identify the system, the model (4) is used, together with a *known* output equation, i.e. the following model is used

$$\begin{pmatrix} x_1^{(1)} \\ \vdots \\ x_{n-1}^{(1)} \\ x_n^{(1)} \end{pmatrix} = \begin{pmatrix} x_2 \\ \vdots \\ x_n \\ f\left(x_n,\dots,x_1,u_1^{(n_1)},\dots,u_1,\dots,u_k^{(n_k)},\dots,u_k,\boldsymbol{\theta} \right) \end{pmatrix} \qquad (18)$$

$$\begin{pmatrix} y_1 \\ \vdots \\ y_p \end{pmatrix} = \begin{pmatrix} c_{11} & \cdots & c_{1n} \\ \vdots & \ddots & \vdots \\ c_{p1} & \cdots & c_{pn} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}.$$

Remark 4: In the basic multiple input single output setting, defined by C1), where the output signal corresponds to the dependent variable x, it follows that $k = p = 1$, $c_{11} = 1$ and $c_{1i} = 0$, $i \neq 1$. This is not an algorithmic requirement though, and multiple input multiple output models can be used, allowing for use of auxiliary measurements of derivatives of the output in the identification process. As stated above, the coefficients of the output relation are not estimated. The reason is that the cascade structure of the dynamics and the output signal relation would then result in a singular problem since only the product of the static small signal gain matters from an input-output point of view, cf. Wigren (1993).

The model (18) is parameterized in continuous time. The following polynomial parameterization is introduced for $f\left(x_1,\ldots,x_n,u_1,\ldots,u_1^{(n_1)},\ldots,u_k,\ldots,u_k^{(n_k)},\boldsymbol{\theta}\right)$

$$f\left(x_1,\ldots,x_n,u_1,\ldots,u_1^{(n_1)},\ldots,u_k,\ldots,u_k^{(n_k)},\boldsymbol{\theta}\right) = \sum_{i_{x1}=0}^{I_{x1}}\cdots\sum_{i_{xn}=0}^{I_{xn}}\sum_{i_{u1}=0}^{I_{u1}}\cdots\sum_{i_{u_1^{(n_1)}}=0}^{I_{u_1^{(n_1)}}}\cdots\sum_{i_{uk}=0}^{I_{uk}}\cdots$$

$$\cdots\sum_{i_{u_k^{(n_k)}}=0}^{I_{u_k^{(n_k)}}}\theta_{i_{x1}\ldots i_{xn}i_{u1}\ldots i_{u_1^{(n_1)}}\ldots i_{uk}\ldots i_{u_k^{(n_k)}}}\left(x_1\right)^{i_{x1}}\ldots\left(x_n\right)^{i_{xn}}\left(u_1\right)^{i_{u1}}\ldots\left(u_1^{(n_1)}\right)^{i_{u_1^{(n_1)}}}\ldots\left(u_k\right)^{i_{uk}}\ldots\left(u_k^{(n_k)}\right)^{i_{u_k^{(n_k)}}} = \boldsymbol{\varphi}^T(\mathbf{x},\mathbf{u})\boldsymbol{\theta}\,. \qquad (19)$$

where

$$\boldsymbol{\theta} = \left(\theta_{0\ldots0} \quad \ldots \quad \theta_{0\ldots I_{u_k^{(n_k)}}} \quad \theta_{0\ldots010} \quad \ldots \quad \theta_{0\ldots01I_{u_k^{(n_k)}}} \quad \ldots \quad \theta_{0\ldots0I_{u_k^{(n_k-1)}}0} \quad \ldots \quad \theta_{0\ldots0I_{u_k^{(n_k-1)}}I_{u_k^{(n_k)}}} \quad \ldots \quad \theta_{I_{X_1}\ldots I_{u_k^{(n_k)}}}\right)^T \qquad (20)$$

$$\boldsymbol{\varphi} = \left(1 \quad \ldots \quad \left(\left(u_k^{(n_k)}\right)^{I_{u_k^{(n_k)}}}\right) \qquad u_k^{(n_k-1)} \quad \ldots \quad \left(\left(u_k^{(n_k-1)}\right)^{I_{u_k^{(n_k-1)}}}\left(u_k^{(n_k)}\right)^{I_{u_k^{(n_k)}}}\right)\right.$$

$$\left.\ldots \quad \left(u_k^{(n_k-1)}\right)^{I_{u_k^{(n_k-1)}}} \quad \ldots \quad \left(\left(u_k^{(n_k-1)}\right)^{I_{u_k^{(n_k-1)}}}\left(u_k^{(n_k)}\right)^{I_{u_k^{(n_k)}}}\right) \quad \ldots \quad \left(\left(x_1\right)^{I_{x1}}\ldots\left(x_n\right)^{I_{xn}}\left(u_1\right)^{I_{u1}}\ldots\left(u_k^{(n_k)}\right)^{I_{u_k^{(n_k)}}}\right)\right)^T. \qquad (21)$$

and where capital I is used to denote the upper limit in the corresponding sum of (19), i.e. the *degree* of the variable in question.

In order to formulate a discrete time model the continuous time ODE model (18) needs to be discretized. This is done with an Euler forward discretization scheme with sampling period T_s. The result is the following discrete time model

$$\begin{pmatrix} x_1(t+T_S,\boldsymbol{\theta}) \\ \vdots \\ x_{n-1}(t+T_S,\boldsymbol{\theta}) \\ x_n(t+T_S,\boldsymbol{\theta}) \end{pmatrix} = \begin{pmatrix} x_1(t,\boldsymbol{\theta}) \\ \vdots \\ x_{n-1}(t,\boldsymbol{\theta}) \\ x_n(t,\boldsymbol{\theta}) \end{pmatrix} + T_S \begin{pmatrix} x_2(t,\boldsymbol{\theta}) \\ \vdots \\ x_n(t,\boldsymbol{\theta}) \\ \boldsymbol{\varphi}^T\left(x_1(t,\boldsymbol{\theta}),\ldots,x_n(t,\boldsymbol{\theta}),u_1(t),\ldots,u_1^{(n_1)}(t),\ldots,u_k(t),\ldots,u_k^{(n_k)}(t)\right)\boldsymbol{\theta} \end{pmatrix}$$

$$\mathbf{y}(t,\boldsymbol{\theta}) = \begin{pmatrix} y_1(t,\boldsymbol{\theta}) \\ \vdots \\ y_p(t,\boldsymbol{\theta}) \end{pmatrix} = \begin{pmatrix} c_{11} & \cdots & c_{1n} \\ \vdots & \ddots & \vdots \\ c_{p1} & \cdots & c_{pn} \end{pmatrix} \begin{pmatrix} x_1(t,\boldsymbol{\theta}) \\ \vdots \\ x_n(t,\boldsymbol{\theta}) \end{pmatrix} = \mathbf{C}\mathbf{x}(t,\boldsymbol{\theta}). \qquad (22)$$

3.2 Gradient

The gradient of the output predictor of (22) is of central importance in the RPEM. The gradient is

$$\boldsymbol{\psi}^T(t,\boldsymbol{\theta}) = \frac{d\mathbf{y}(t,\boldsymbol{\theta})}{d\boldsymbol{\theta}} = \mathbf{C}\frac{d\mathbf{x}(t,\boldsymbol{\theta})}{d\boldsymbol{\theta}}. \qquad (23)$$

The derivative of the state with respect to the parameter vector follows from a differentiation of the state equation of (22), exactly as in Ljung and Söderström, (1983).

$$\frac{d\mathbf{x}(t+T_S,\boldsymbol{\theta})}{d\boldsymbol{\theta}} = \frac{d\mathbf{x}(t,\boldsymbol{\theta})}{d\boldsymbol{\theta}} + T_S \begin{pmatrix} \frac{dx_2(t,\boldsymbol{\theta})}{d\boldsymbol{\theta}} \\ \vdots \\ \frac{dx_n(t,\boldsymbol{\theta})}{d\boldsymbol{\theta}} \\ \boldsymbol{\varphi}^T\left(\mathbf{x}(t,\boldsymbol{\theta}),\mathbf{u}(t)\right) + \boldsymbol{\theta}^T\left(\frac{d\boldsymbol{\varphi}\left(\mathbf{x}(t,\boldsymbol{\theta}),\mathbf{u}(t)\right)}{d\mathbf{x}}\right)\left(\frac{d\mathbf{x}(t,\boldsymbol{\theta})}{d\boldsymbol{\theta}}\right) \end{pmatrix}. \qquad (24)$$

Iteration of (24) then gives the gradient after multiplication according to (23) . The computation of the last factor of the last component of (24) remains to be defined. This follows by straightforward differentiation of (21). The details are outlined in the algorithm (28). The notation may look complex, however when coding the algorithm the structure of the underlying sums of (19) can be naturally exploited in a loop structure that efficiently computes all required quantities.

3.3 Algorithm

In order to derive a recursive prediction error identification algorithm the following criterion is introduced, cf. Ljung and Söderström, (1983), pp. 82-84

$$V(\boldsymbol{\theta}) = \frac{1}{2} E\left[\boldsymbol{\varepsilon}^T(t,\boldsymbol{\theta})\boldsymbol{\Lambda}^{-1}(t,\boldsymbol{\theta})\boldsymbol{\varepsilon}(t,\boldsymbol{\theta}) + \log\det\left(\boldsymbol{\Lambda}(t,\boldsymbol{\theta})\right)\right]. \qquad (25)$$

In (25), $E[\ ]$ denotes the expectation operator, $\boldsymbol{\Lambda}(t,\boldsymbol{\theta})$ denotes the (unknown) covariance matrix of the measurement disturbance and $\boldsymbol{\varepsilon}(t,\boldsymbol{\theta})$ denotes the prediction error. The covariance is dependent of the unknown parameter vector since it is *estimated* from the prediction errors obtained during the identification run. This is the reason for the second term of (25). In case of Gaussian prediction errors the criterion equals the maximum likelihood criterion, cf. Ljung and Söderström, (1983). The prediction error, $\boldsymbol{\varepsilon}(t,\boldsymbol{\theta})$, is given by

$$\boldsymbol{\varepsilon}(t,\boldsymbol{\theta}) = \mathbf{y}_m(t,\boldsymbol{\theta}) - \mathbf{y}(t,\boldsymbol{\theta}). \qquad (26)$$

The minimization of (25) is then performed with the stochastic Gauss-Newton method, cf. Ljung and Söderström (1983), pp. 88-93. That algorithm requires first and second derivatives of the criterion (25). The first derivative follows from (23) - (26). The second derivative follows by differentiation of (25) twice, utilizing (23) and (24), and by introduction of the conventional Gauss-Newton approximations. The resulting updating equations for $\boldsymbol{\theta}$ and $\boldsymbol{\Lambda}(t,\boldsymbol{\theta})$ then become. cf. Ljung and Söderström, (1983)

$$\begin{aligned}
\hat{\boldsymbol{\theta}}(t) &= \hat{\boldsymbol{\theta}}(t-T_S) + \frac{\mu(t)}{t}\,\mathbf{R}^{-1}(t)\boldsymbol{\psi}\left(t,\hat{\boldsymbol{\theta}}(t-T_S)\right)\boldsymbol{\Lambda}^{-1}\left(t,\hat{\boldsymbol{\theta}}(t-T_S)\right)\boldsymbol{\varepsilon}\left(t,\hat{\boldsymbol{\theta}}(t-T_S)\right) \\
\boldsymbol{\Lambda}(t) &= \boldsymbol{\Lambda}(t-T_S) + \frac{\mu(t)}{t}\left(\boldsymbol{\varepsilon}\left(t,\hat{\boldsymbol{\theta}}(t-T_S)\right)\boldsymbol{\varepsilon}^T\left(t,\hat{\boldsymbol{\theta}}(t-T_S)\right) - \boldsymbol{\Lambda}(t)\right)
\end{aligned} \qquad (27)$$

IT is shown in Ljung and Söderström, (1983), pp. 185-188, that this choice of updating direction for $\boldsymbol{\Lambda}$ leads to a minimization of (26). In (27), $\mu(t)/t$ is the gain sequence and $\mathbf{R}(t)$ is the approximation of the second derivative of the criterion of (25). The quantities $\boldsymbol{\varepsilon}\left(t,\hat{\boldsymbol{\theta}}(t-T_S)\right)$, $\boldsymbol{\psi}\left(t,\hat{\boldsymbol{\theta}}(t-T_S)\right)$ can be computed from (26) and (24), respectively, by replacement of $\boldsymbol{\theta}$ with the running estimate $\hat{\boldsymbol{\theta}}(t-T_S)$. The same procedure can be used for $\boldsymbol{\Lambda}$. The resulting algorithm is not recursive though, since all available data needs to be processed by the dynamic model, for fixed $\hat{\boldsymbol{\theta}}(t-T_S)$, in order to generate $\boldsymbol{\varepsilon}\left(t,\hat{\boldsymbol{\theta}}(t-T_S)\right)$, $\boldsymbol{\psi}\left(t,\hat{\boldsymbol{\theta}}(t-T_S)\right)$ and $\boldsymbol{\Lambda}\left(t,\hat{\boldsymbol{\theta}}(t-T_S)\right)$. However, exactly as in

Ljung and Söderström, (1983), this issue can be resolved by using the running estimate also in (22), to obtain approximations of $\hat{\mathbf{y}}\left(t,\hat{\boldsymbol{\theta}}(t-T_S)\right)$, $\boldsymbol{\varepsilon}\left(t,\hat{\boldsymbol{\theta}}(t-T_S)\right)$, $\boldsymbol{\psi}\left(t,\hat{\boldsymbol{\theta}}(t-T_S)\right)$ and $\boldsymbol{\Lambda}\left(t,\hat{\boldsymbol{\theta}}(t-T_S)\right)$. These approximations are generally accurate whenever the adaptation gain is sufficiently small and the system is exponentially stable. Before stating the complete algorithm it is noted that a projection algorithm is needed in order to keep the model stable, as indicated by the brackets in the $\hat{\boldsymbol{\theta}}$-recursion in (28). The projection algorithm is further discussed in the next subsection. The final result is now the following recursive algorithm

$$\boldsymbol{\varepsilon}(t)=\mathbf{y}_m(t)-\mathbf{y}(t)$$

$$\boldsymbol{\Lambda}(t)=\boldsymbol{\Lambda}(t-T_S)+\frac{\mu(t)}{t}\left(\boldsymbol{\varepsilon}(t)\boldsymbol{\varepsilon}^T(t)-\boldsymbol{\Lambda}(t-T_S)\right)$$

$$\mathbf{R}(t)=\mathbf{R}(t-T_S)+\frac{\mu(t)}{t}\left(\boldsymbol{\psi}(t)\boldsymbol{\Lambda}^{-1}(t)\boldsymbol{\psi}^T(t)-\mathbf{R}(t-T_S)\right)$$

$$\hat{\boldsymbol{\theta}}(t)=\left[\hat{\boldsymbol{\theta}}(t-T_S)+\frac{\mu(t)}{t}\,\mathbf{R}^{-1}(t)\boldsymbol{\psi}(t)\boldsymbol{\Lambda}^{-1}(t)\boldsymbol{\varepsilon}(t)\right]_{D_M}$$

$$\boldsymbol{\varphi}(t)=\left(1\quad\ldots\quad\left(\left(u_k^{(n_k)}(t)\right)^{I_{u_k^{(n_k)}}}\right)\qquad u_k^{(n_k-1)}(t)\quad\ldots\quad\left(\left(u_k^{(n_k-1)}(t)\right)^{I_{u_k^{(n_k-1)}}}\left(u_k^{(n_k)}(t)\right)^{I_{u_k^{(n_k)}}}\right)\quad\ldots\right.$$

$$\left.\ldots\quad\left(u_k^{(n_k-1)}(t)\right)^{I_{u_k^{(n_k-1)}}}\quad\ldots\quad\left(\left(u_k^{(n_k-1)}(t)\right)^{I_{u_k^{(n_k-1)}}}\left(u_k^{(n_k)}(t)\right)^{I_{u_k^{(n_k)}}}\right)\quad\ldots\quad\left(\left(x_1(t)\right)^{I_{x_1}}\ldots\left(x_n(t)\right)^{I_{x_n}}\left(u_1(t)\right)^{I_{u_1}}\ldots\left(u_k^{(n_k)}(t)\right)^{I_{u_k^{(n_k)}}}\right)\right)^T$$

$$\begin{pmatrix}x_1(t+T_S)\\ \vdots\\ x_{n-1}(t+T_S)\\ x_n(t+T_S)\end{pmatrix}=\begin{pmatrix}x_1(t)\\ \vdots\\ x_{n-1}(t)\\ x_n(t)\end{pmatrix}+T_S\begin{pmatrix}x_2(t)\\ \vdots\\ x_n(t)\\ \boldsymbol{\varphi}^T(t)\hat{\boldsymbol{\theta}}(t)\end{pmatrix}$$

$$\mathbf{y}(t+T_S)=\begin{pmatrix}y_1(t+T_S)\\ \vdots\\ y_p(t+T_S)\end{pmatrix}=\begin{pmatrix}c_{11}&\cdots&c_{1n}\\ \vdots&\ddots&\vdots\\ c_{p1}&\cdots&c_{pn}\end{pmatrix}\begin{pmatrix}x_1(t+T_S)\\ \vdots\\ x_n(t+T_S)\end{pmatrix}$$

$$\frac{d\boldsymbol{\varphi}}{dx_i}(t)=\left(\mathbf{0}^T\quad 1\quad u_k^{(n_k)}(t)\quad\ldots\quad\left(\left(x_{i-1}(t)\right)^{I_{x_{i-1}}}\ldots\left(x_n(t)\right)^{I_{x_n}}\left(u_1(t)\right)^{I_{u_1}}\ldots\left(u_k^{(n_k)}(t)\right)^{I_{u_k^{(n_k)}}}\right)\quad 2x_i(t)\quad 2x_i(t)u_k^{(n_k)}(t)\quad\ldots\right),$$

$$i=1,\ldots,n$$

$$\frac{d\boldsymbol{\varphi}}{d\mathbf{x}}(t)=\begin{pmatrix}\frac{d\boldsymbol{\varphi}}{dx_1}(t)\\ \vdots \\ \frac{d\boldsymbol{\varphi}}{dx_n}(t)\end{pmatrix}$$

$$\begin{pmatrix}\frac{dx_1}{d\boldsymbol{\theta}}(t+T_S)\\ \vdots \\ \frac{dx_{n-1}}{d\boldsymbol{\theta}}(t+T_S)\\ \frac{dx_n}{d\boldsymbol{\theta}}(t+T_S)\end{pmatrix}=\begin{pmatrix}\frac{dx_1}{d\boldsymbol{\theta}}(t)\\ \vdots \\ \frac{dx_{n-1}}{d\boldsymbol{\theta}}(t)\\ \frac{dx_n}{d\boldsymbol{\theta}}(t)\end{pmatrix}+T_S\begin{pmatrix}\frac{dx_2}{d\boldsymbol{\theta}}(t)\\ \vdots \\ \frac{dx_n}{d\boldsymbol{\theta}}(t)\\ \boldsymbol{\varphi}^T(t)+\hat{\boldsymbol{\theta}}^T(t)\left(\frac{d\boldsymbol{\varphi}}{d\mathbf{x}}(t)\right)\left(\left(\frac{dx_1}{d\boldsymbol{\theta}}(t)\right)^T \quad \cdots \quad \left(\frac{dx_n}{d\boldsymbol{\theta}}(t)\right)^T\right)^T\end{pmatrix}$$

$$\boldsymbol{\psi}(t+T_S)=\begin{pmatrix}c_{11} & \cdots & c_{1n}\\ \vdots & \ddots & \vdots \\ c_{p1} & \cdots & c_{pn}\end{pmatrix}\begin{pmatrix}\frac{dx_1}{d\boldsymbol{\theta}}(t+T_S)\\ \vdots \\ \frac{dx_n}{d\boldsymbol{\theta}}(t+T_S)\end{pmatrix}. \qquad (28)$$

3.4 Projection and Initialization

In the linear case, the application of projection algorithms (Ljung and Söderström, 1983) is a standard approach to keep the model asymptotically stable. The reason is that stability properties can be easily extracted from the pole locations. In the general nonlinear case treated here, no such straightforward method does exist. This is no reason to avoid the problem though, there is rather a strong need for stability monitoring methods in the nonlinear system identification field.

One (stringent) approach could be to apply Lyapunov stability theory to the model. A drawback with this approach is the fact that Lyapunov stability analysis can be restrictive and even inconclusive, providing results on instability that may not provide the required sharpness. An alternative could be to rely in linearization techniques in combination with an assumption on slow adaptation. The drawback with this approach is that the signal level variation may have to be kept relatively low, so as not to invalidate the linearization assumption too much. This would, in turn, lead to difficulties in the estimation of the non-linearities of the system, since the nonlinear effects are not reflected very much in the measured signals, cf. Wigren (2003).

The projection algorithm applied in (28) is based on a linearization of the model. The linearized system matrix becomes

$$S(\boldsymbol{\theta}) = I_n + T_S \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & 0 & \cdots & 0 & 1 \\ \hat{\boldsymbol{\theta}}^T(t)\dfrac{d\boldsymbol{\varphi}(t)}{d\mathbf{x}} & & & & \end{pmatrix}. \qquad (29)$$

The model set is defined according to

$$D_M = \{\boldsymbol{\theta} \;\; | \;\; |eig(S(\boldsymbol{\theta}))| \;\; < \;\; 1-\delta\}, \; \delta > 0. \qquad (30)$$

Since the system is not linear, δ may be selected with some margin to the linear stability limit. The following projection algorithm is then applied, i.e.

$$\left[\hat{\boldsymbol{\theta}}(t)\right]_{D_M} = \begin{cases} \hat{\boldsymbol{\theta}}(t) & \hat{\boldsymbol{\theta}}(t) \in D_M \\ \hat{\boldsymbol{\theta}}(t-T_S) & \hat{\boldsymbol{\theta}}(t) \notin D_M \end{cases}. \qquad (31)$$

The updating is hence stopped when the result is outside the model set, cf. Wigren, (1994).

As always when output error identification algorithms are used, initialization is of central importance in order to avoid convergence to sub-optimal minimum points of the criterion. For nonlinear models this is even more important, considering the more complicated dynamics.

The quantities that require initial values include $\mu(t)$, $\hat{\boldsymbol{\theta}}(t)$, $\boldsymbol{\Lambda}(t)$, $\mathbf{R}(t)$ and $d\mathbf{x}(t)/d\boldsymbol{\theta}$. One experience from the this work, is that it is advantageous for the algorithm to maintain a relatively low adaptation gain initially. This allows the search direction to build up from data without extensive initial parameter excursions perhaps towards an unstable model. The choice of the gain sequence $\mu(t)/t$ is hence a very important one. The initial values for $\boldsymbol{\Lambda}(t)$ and $\mathbf{R}(t)$ are both closely tied to the choice of gain sequence and theses choices are discussed below. The gain sequence of the algorithm is selected according to

$$\bar{\mu}(t+T_S) = \mu_0 \bar{\mu}(t) + 1 - \mu_0$$

$$\mu(t) = \frac{t}{t+\mu_1 T_S}\bar{\mu}(t). \qquad (32)$$

This supports the limitation of the initial adaptation gain, by selection of a high enough μ_1. At the same time the benefits of the conventional exponential decaying transient (Ljung and Söderström,

1983) can be retained. Typical values can be $\mu_0 = 0.999$, $\mu_1 = 100T_S$ together with $\bar{\mu}(0) = 10$. Note that the initial value for $\boldsymbol{\Lambda}(t)$ should not significantly underestimate the covariance matrix of the prediction errors since this tends to increase the gain of the parameter recursion. A similar argument applies to $\mathbf{R}(t)$ .Useful initial parameter estimates can often be obtained by applying linear identification methods. If this is not possible, the strategy of section 5 can be applied, i.e. to initiate $\hat{\boldsymbol{\theta}}(t)$ with parameters corresponding to asymptotically stable linear dynamics, well inside the model set.

4. SCALING

Scaling of estimated quantities is a standard technique in optimization theory (Luenberger, 1984). In the literature on linear identification scaling has not achieved a central status though. The reason is perhaps the strong position of least squares techniques that are not so sensitive to problems with slow convergence and local minimum points, that may be caused by poor scaling in nonlinear optimization problems.

In nonlinear identification scaling probably deserves a more central status.

Example 1: Consider identification of the following multi-variable polynomial nonlinear function $f(x_1, x_2) = \alpha_0 x_1 + \beta_0 x_1 x_2^2$ where the effect of the nonlinear term $\beta_0 x_1 x_2^2$ on $f(x_1, x_2)$ is assumed to be about the same as the effect of the linear term $\alpha_0 x_1$. In case the experiment condition would be such that x_2 would be roughly a factor of 100 greater than x_1 , it follows that the estimate β needs to be about $10^{-4}\alpha$ to explain the data. In case the achievable relative accuracy of α and β would be similar, it follows that the components of the covariance matrix would differ by a factor of 10^{-8} , i.e. conditioning is likely to be poor. In case gradient algorithms are applied convergence is likely to be slow since the convergence speed is negatively affected by the eigenvalue spread of the problem, cf. Luenberger, (1984). Far from the minimum point slow convergence is also likely to hamper more advanced methods. The reason is that the quadratic convergence that can be proved for advanced Newton search methods is only valid close to the minimum point of the criterion. In case of poor

scaling and conditioning the criterion function is likely to have narrow "valleys" where the algorithm may dwell for long periods of time.

Conventional linear scaling applied to nonlinear system identification introduces a nonsingular linear transformation of the identified parameters as

$$\tilde{\boldsymbol{\theta}} = \mathbf{T}\boldsymbol{\theta} \Leftrightarrow \boldsymbol{\theta} = \mathbf{T}^{-1}\tilde{\boldsymbol{\theta}} \qquad (33)$$

The transformation is then exploited in the criterion function (25) and a minimization of the criterion is performed with respect to $\tilde{\boldsymbol{\theta}}$ instead of with respect to $\boldsymbol{\theta}$, i.e.

$$\tilde{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \frac{1}{2} E\left[\boldsymbol{\varepsilon}^T\left(t, \mathbf{T}^{-1}\tilde{\boldsymbol{\theta}}\right)\boldsymbol{\Lambda}^{-1}\left(t, \mathbf{T}^{-1}\tilde{\boldsymbol{\theta}}\right)\boldsymbol{\varepsilon}\left(t, \mathbf{T}^{-1}\tilde{\boldsymbol{\theta}}\right) + \log\det\left(\boldsymbol{\Lambda}\left(t, \mathbf{T}^{-1}\tilde{\boldsymbol{\theta}}\right)\right)\right]. \qquad (34)$$

The details of the algorithmic consequences on (28) are not outlined here. In practice diagonal matrix scaling is usually all the prior knowledge allows, cf. Luenberger, (1984), Ekstam and Smed, (1987).

The application of (33) requires some knowledge of the expected range of the different parameters, something that may be difficult to obtain for black box models. Further, referring to Example 1, the origin of the scaling problems may very well be the relative size of the state signals that are generated in the algorithm.

It was experimentally noticed that scaling problems in (28) are often highly related to the selection of the sampling period. The sampling period of course needs to be short enough during measurement, in order to capture the essential dynamics. However, since the sampling period appears *explicitly* in the algorithm (28) it is straightforward to apply the algorithm with another, *scaled* value of the sampling period. Even if a scaled value of the sampling period is applied, the algorithm still attempts to minimize the criterion, thereby obtaining other minimizing parameter values than when the true sampling period is used. When testing the idea experimentally, dramatic improvements could be observed in the algorithmic behavior. Convergence speeds could be improved and initial values that lead to divergence and instability could be made to work well.

To explain this, introduce the following models (cf. (22)) that differ only in the applied sampling period (the superscript discriminates between (35) and (36))

$$\begin{pmatrix} x_1^1\left(t+T_S,\boldsymbol{\theta}^1\right) \\ \vdots \\ x_{n-1}^1\left(t+T_S,\boldsymbol{\theta}^1\right) \\ x_n^1\left(t+T_S,\boldsymbol{\theta}^1\right) \end{pmatrix} = \begin{pmatrix} x_1^1\left(t,\boldsymbol{\theta}^1\right) \\ \vdots \\ x_{n-1}^1\left(t,\boldsymbol{\theta}^1\right) \\ x_n^1\left(t,\boldsymbol{\theta}^1\right) \end{pmatrix} + T_S^1 \begin{pmatrix} x_2^1\left(t,\boldsymbol{\theta}^1\right) \\ \vdots \\ x_n^1\left(t,\boldsymbol{\theta}^1\right) \\ f\left(x_1^1\left(t,\boldsymbol{\theta}^1\right),\ldots x_n^1\left(t,\boldsymbol{\theta}^1\right),u_1(t),\ldots,u_k^{(n_k)}(t),\boldsymbol{\theta}^1\right) \end{pmatrix} \tag{35}$$

$$\begin{pmatrix} x_1^2\left(t+T_S,\boldsymbol{\theta}^2\right) \\ \vdots \\ x_{n-1}^2\left(t+T_S,\boldsymbol{\theta}^2\right) \\ x_n^2\left(t+T_S,\boldsymbol{\theta}^2\right) \end{pmatrix} = \begin{pmatrix} x_1^2\left(t,\boldsymbol{\theta}^2\right) \\ \vdots \\ x_{n-1}^2\left(t,\boldsymbol{\theta}^2\right) \\ x_n^2\left(t,\boldsymbol{\theta}^2\right) \end{pmatrix} + T_S^2 \begin{pmatrix} x_2^2\left(t,\boldsymbol{\theta}^2\right) \\ \vdots \\ x_n^2\left(t,\boldsymbol{\theta}^2\right) \\ f\left(x_1^2\left(t,\boldsymbol{\theta}^2\right),\ldots x_n^2\left(t,\boldsymbol{\theta}^2\right),u_1(t),\ldots,u_k^{(n_k)}(t),\boldsymbol{\theta}^2\right) \end{pmatrix}. \tag{36}$$

Note that the original sampling period must be retained in all time argument, so as to refer to the correct measurement times. Introduce the following assumptions

C6) The measurement $y(t)$ corresponds to the states $x_1^1\left(t,\boldsymbol{\theta}^1\right)$ and $x_1^2\left(t,\boldsymbol{\theta}^2\right)$ of (35) and (36) .

C7) The algorithm converges to an exact description of the input -output properties of the system for (35) and (36), i.e. $y(t) = x_1^1\left(t,\boldsymbol{\theta}^1\right) = x_1^2\left(t,\boldsymbol{\theta}^2\right) \neq 0, \forall t$.

It follows from C6) and C7) that

$$1 = \frac{x_1^1\left(t,\boldsymbol{\theta}^1\right)}{x_1^2\left(t,\boldsymbol{\theta}^2\right)} = \frac{x_1^1\left(t-T_S\right) + T_S^1 x_2^1\left(t-T_s,\boldsymbol{\theta}^1\right)}{x_1^2\left(t-T_S\right) + T_S^2 x_2^2\left(t-T_S,\boldsymbol{\theta}^2\right)}, \forall t$$

$$\Leftrightarrow x_1^2\left(t-T_S\right) + T_S^2 x_2^2\left(t-T_S,\boldsymbol{\theta}^2\right) = x_1^1\left(t-T_S\right) + T_S^1 x_2^1\left(t-T_s,\boldsymbol{\theta}^1\right), \forall t \tag{37}$$

Applying C7) once more results in

$$x_2^1\left(t-T_S,\boldsymbol{\theta}^1\right) = \left(\frac{T_S^2}{T_S^1}\right) x_2^2\left(t-T_S,\boldsymbol{\theta}^2\right), \forall t\,. \tag{38}$$

The argumentation can then be repeated starting with (38) . The result is

$$x_3^1\left(t-2T_S,\boldsymbol{\theta}^1\right) = \left(\frac{T_S^2}{T_S^1}\right)^2 x_3^2\left(t-2T_S,\boldsymbol{\theta}^2\right), \forall t\,. \tag{39}$$

Repetition of this process n times, and exploiting the fact that the relations are valid for all t proves

Theorem 2: Consider the two models

$$\begin{pmatrix} x_1^1\left(t+T_S,\boldsymbol{\theta}^1\right) \\ \vdots \\ x_{n-1}^1\left(t+T_S,\boldsymbol{\theta}^1\right) \\ x_n^1\left(t+T_S,\boldsymbol{\theta}^1\right) \end{pmatrix} = \begin{pmatrix} x_1^1\left(t,\boldsymbol{\theta}^1\right) \\ \vdots \\ x_{n-1}^1\left(t,\boldsymbol{\theta}^1\right) \\ x_n^1\left(t,\boldsymbol{\theta}^1\right) \end{pmatrix} + T_S^1 \begin{pmatrix} x_2^1\left(t,\boldsymbol{\theta}^1\right) \\ \vdots \\ x_n^1\left(t,\boldsymbol{\theta}^1\right) \\ f\left(x_1^1\left(t,\boldsymbol{\theta}^1\right),\ldots.x_n^1\left(t,\boldsymbol{\theta}^1\right),u_1(t),\ldots,u_k^{(n_k)}(t),\boldsymbol{\theta}^1\right) \end{pmatrix}$$

$$\begin{pmatrix} x_1^2\left(t+T_S,\boldsymbol{\theta}^2\right) \\ \vdots \\ x_{n-1}^2\left(t+T_S,\boldsymbol{\theta}^2\right) \\ x_n^2\left(t+T_S,\boldsymbol{\theta}^2\right) \end{pmatrix} = \begin{pmatrix} x_1^2\left(t,\boldsymbol{\theta}^2\right) \\ \vdots \\ x_{n-1}^2\left(t,\boldsymbol{\theta}^2\right) \\ x_n^2\left(t,\boldsymbol{\theta}^2\right) \end{pmatrix} + T_S^2 \begin{pmatrix} x_2^2\left(t,\boldsymbol{\theta}^2\right) \\ \vdots \\ x_n^2\left(t,\boldsymbol{\theta}^2\right) \\ f\left(x_1^2\left(t,\boldsymbol{\theta}^2\right),\ldots.x_n^2\left(t,\boldsymbol{\theta}^2\right),u_1(t),\ldots,u_k^{(n_k)}(t),\boldsymbol{\theta}^2\right) \end{pmatrix}$$

where T_S^1 is the measurement sampling period and where T_S^2 is the scaled sampling interval applied when running the algorithm (28). Provided that C6) and C7) holds it then follows that

$$x_i^2\left(t,\boldsymbol{\theta}^2\right) = \left(\frac{T_S^2}{T_S^1}\right)^{i-1} x_i^1\left(t,\boldsymbol{\theta}^1\right),\ i=1,\cdots,n\,.$$

Remark 5: The essence of this result is that the scaling of the sampling period results in state variable scaling according to Theorem 1, provided that the estimated model converges to a good enough input output model. Since powers of the state variables appear in the multi-variable polynomial used in the report, such scaling can be expected to have a significant effect on the criterion minimization problem.

5. EXPERIMENT

In this section a process consisting of a two cascaded tanks with free outlets fed by a pump is studied. The laboratory equipment is located in the control laboratory at Uppsala University and it is displayed in Fig. 1. The input signal to the process is the voltage applied to the pump and the output signals consist of measurements of the water levels of the two tanks. The process is controlled from a PC equipped with MATLAB and MATLAB interfaces to the A/D and D/A converters that provide commands to and measurements from the process. The process is relatively slow with time constants slightly less than one minute. The input to the process for the identification experiment is generated off line in MATLAB. A simple timed loop in MATLAB outputs the input signal to the pump and measures the levels of the tanks. The laboratory process is suitable for physical modeling. Application of Bernoulli's principle and conservation of mass results in

$$\begin{pmatrix} \frac{dh_1}{dt} \\ \frac{dh_2}{dt} \end{pmatrix} = \begin{pmatrix} -\frac{a_1\sqrt{2g}}{A_1}\sqrt{h_1} + \frac{1}{A_1}ku(t) \\ -\frac{a_2\sqrt{2g}}{A_2}\sqrt{h_2} + \frac{a_1\sqrt{2g}}{A_2}\sqrt{h_1} \end{pmatrix}. \qquad (40)$$

Here h_1 and h_2 denote the levels of the upper and the lower tank, respectively. The corresponding areas of the tanks are A_1 and A_2 while the effluent areas are denoted a_1 and a_2. The gravity is denoted by g, the voltage to input flow conversion constant by k and the applied voltage to the pump by $u(t)$. It can be noted that the model is highly nonlinear, that the structure corresponds to (6) rather than to (4), and that there is no explicit input dependence in the equation describing the level of the lower tank. Hence the result of Theorem 1 and Remark 3 can be illustrated in a real situation. The scaling scheme of Theorem 2 will also be illustrated. In the following example the input signal is the voltage applied to the pump and output signal is defined to be the level of the lower tank, i.e. (28) is used for identification of a second order nonlinear system. The water level measurements from the upper tank are not used.

Example 2: 2500 samples of data from the double tank process were collected. The sampling period was 5 s. In order to excite the system in frequency and in amplitude a PRBS down-sampled 30 times, modified by multiplication with a uniformly distributed random variable, was used as input signal, cf. Wigren, (2003). The input signal varied the applied control voltage to the pump between 0 and 2.4 V. The applied input signal appears in Fig. 2 together with the measured water level of the lower tank.

Identification was then performed with the algorithm (28). Note that the observed quantity equals the first state vector component in (28), contrary to the formulation in (40). Linear measurements were assumed, i.e. the following measurement equation was used

$$y_1 = x_1. \qquad (41)$$

The nonlinear state relation then needs to compensate for the complete input-output nonlinearity. The model orders of the right hand side function were selected as $I_{u_1} = I_{x_1} = I_{x_2} = 1$, i.e.

$$\boldsymbol{\theta} = \begin{pmatrix} \theta_{000} & \theta_{001} & \theta_{010} & \theta_{011} & \theta_{100} & \theta_{101} & \theta_{110} & \theta_{111} \end{pmatrix}^T$$

$$\boldsymbol{\varphi} = \begin{pmatrix} 1 & u_1 & x_1 & x_1 u_1 & x_2 & x_2 u_1 & x_2 x_1 & x_2 x_1 u_1 \end{pmatrix}^T . \qquad (42)$$

The identification algorithm was initialized with $\begin{pmatrix} x_1(0) & x_2(0) \end{pmatrix} = \begin{pmatrix} 1.0000 & 0.1000 \end{pmatrix}^T$, $\boldsymbol{\Lambda}(0) = 1.0000$, $\mathbf{R}(0) = 30\mathbf{I}$ and with $d\mathbf{x}/d\boldsymbol{\theta}(0) = \mathbf{0}$. Furthermore, $\mu_0 = 0.999$, $\mu_1 = 300$ and $\mu(0) = 5$ were used. The stability limit of the projection algorithm was set to $\delta = 0.025$. The initial parameter vector was selected as

$$\hat{\boldsymbol{\theta}}(0) = \begin{pmatrix} 0.0000 & 0.0000 & -0.2500 & 0.0000 & -0.2500 & 0.0000 & 0.0000 & 0.0000 \end{pmatrix}^T . \qquad (43)$$

This corresponds to asymptotically stable continuous time linear dynamics.

Initial experiments revealed a tendency to end up at the boundary of the model set. Experimenting traced the problem to the time scaling of the original system, and hence scaling of the sampling period was applied as described in section 4. The values T_S^1=5.0 s and T_S^2=0.5 s were used, i.e. the algorithm was applied with a scaled sampling period of 0.5 s. The scaling resolved the problems indicated above. At the end of the recursive identification run, the following parameters were obtained, corresponding to the scaled sampling period.

$$\hat{\boldsymbol{\theta}}(2500T_S) = \begin{pmatrix} 0.0650 & 0.0876 & -0.3020 & -0.0187 & -0.0896 & 0.0361 & -0.1666 & -0.0468 \end{pmatrix}^T \qquad (44)$$

The parameters obtained at the end of the run were used to simulate the output of the system. The result appears in Fig. 3 and it shows that a quite accurate model is obtained. The evolution of the parameter estimates during the run appears in Fig. 4. As a comparison to Fig. 2, the linear model that is obtained when the parameters corresponding to nonlinear terms are set to zero, was simulated. The result appears in Fig. 5 and this figure indicates that the nonlinear contribution to the output signal of the model is dominant. To validate the model the mean residual analysis method was applied (Wigren, 2003). That method sorts residuals in pre-determined intervals, corresponding to the interval of the measured output signal at the same time instant. This provides an estimate of the modeling accuracy as a function of the signal levels. The results appears in Fig. 5, which indicates that the accuracy is better for high signal levels. An inspection of Fig. 3 supports this fact. At the end of the run

$\Lambda(2500T_S) = 1.4204$ was obtained. It can be concluded that that the proposed method is capable of accurate modeling of the nonlinear double tank laboratory process. This verifies the algorithm, Theorem 1 as well as Theorem 2.

6. CONCLUSIONS

The report has discussed recursive identification based on nonlinear black box state space models. It was first explained why nonlinear state space models where the right hand sides of more than one component of the ODE model is parameterized, may easily lead to over-parameterization. Such model structures are therefore unsuitable when black box parameterizations of nonlinear state space models are used. The recursive prediction error identification algorithm proposed in the report was therefore based on a multi-variable polynomial parameterization of one single right hand side component of the nonlinear ODE. Many systems are however physically modeled by nonlinear ODE models with (structured) right hand sides with more than one nontrivial component. The applicability of the proposed RPEM to the identification of such systems was therefore analyzed. It was proved that a general state space model with more than one nontrivial right hand side component can be (locally) transformed to the coordinate system of the model used by the proposed RPEM. The algorithm is hence applicable to a quite general class of nonlinear systems. The method was applied successfully to measured data from a laboratory process consisting of two cascaded tanks with free outlets, thereby verifying all main results of the report experimentally.

Practically important algorithmic aspects include initialization since the output error formulation suffers from a risk of convergence to a sub-optimal minimum point of the criterion function. To improve algorithmic performance scaling turned out to be an important technique. In particular scaling of the sampling period applied in the algorithm was show to be an efficient method to improve the numerical performance of the algorithm. It was shown that when the algorithm converges, the method scales the states of the model according to the quotient between the scaled and true sampling periods. Since poor scaling between the states is a severe problem when powers of the states are exploited in the nonlinear model, the method *significantly* improves the operation of the

proposed algorithm. Conventional scaling of the assumed parameter values can of course also be applied. Further studies of scaling is recommended for the proposed and related methods. Other interesting research subjects include the design of projection algorithms for nonlinear systems as well as a closer study of convergence and identifiability properties of this and related methods.

REFERENCES

Åström, K. J. and Eklund, K (1975). A simple non-linear drum boiler model. *Int. J. Contr.*, **22**, 739-740.

Billings, S. A. and S. Y. Fakhouri (1982). Identification of systems containing linear dynamic and static nonlinear elements. *Automatica*, **18**, 15-26.

Billings, S. A., M. B. Fadzil, J. L. Sulley and P. M. Johnson (1988). Identification of a non-linear difference equation model of an industrial diesel generator. *Mech. Sys. Signal Processing*, **2,** 59-76.

Bohlin, T, (1994). A case study of grey box identification, *Automatica*, **30**, pp. 307-318.

Chen. S. and S. A. Billings (1989). Representation of nonlinear systems: the NARMAX model. *Int. J. Contr.*, **49**, 1013-1032.

Chen, S and S. A. Billings (1992). Neural networks for nonlinear dynamic system modelling and identification. *Int. J. Control*, **56**, 319-346.

Ekstam, L. and T. Smed (1987). Parameter estimation in dynamic systems with application to power engineering. *Technical Licentiate Thesisis*, UPTEC 8747R, Uppsala University, Uppsala, Sweden.

Funkquist, J. (1994). On modeling and identification of a continuous pulp digester. *Proc. SYSID 1994,* Copenhagen, Denmark.

Khalil, H. K. (1996). *Nonlinear Systems - second edition.* Prentice Hall, Upper Saddle River, NJ.

Ljung, L. (1975). Theorems for the asymptotic analysis of recursive stochastic algorithms. Report 7522, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.

Ljung, L. (1977a). On positive real transfer functions and the convergence of some recursive schemes. *IEEE Trans. Aut. Contr.*, **AC-22**, 539-551.

Ljung, L. (1977b). Analysis of recursive stochastic algorithms. *IEEE Trans. Aut. Contr.*, **AC-22**, 551-575.

Ljung, L. (1997). Non-linear black box models in system identification. *Proc. IFAC Symposium on Advanced Control of Chemical Processes, ADCHEM'97,* Banff, Canada, 1-13.

Ljung L.and T. Glad (2004). *Modellbygge och Simulering, 2:nd ed.* Studentlitteratur, Lund, Sweden (in Swedish).

Ljung, L. and T. Söderström (1983). *Theory and Practice of Recursive Identification*. M. I. T. Press, Cambridge, MA.

Luenberger, D. G. (1984). *Linear and Nonlinear Programming, 2:nd Ed.*. Addison-Wesley, Reading, MA.

Pajunen, G. A. (1992). Adaptive control of Wiener type nonlinear systems. *Automatica*, **28**, 781-785.

Schetzen, M. (1980). *The Volterra and Wiener Theories of Nonlinear Systems.* John Wiley, New York, NY.

Sjöberg, J. and L. Ljung (1992). Overtraining, regularization, and searching for minimum in neural networks. *Proc. Symp. On Adaptive systems in Control and Signal Processing.* Grenoble, Switzerland.

Sjöberg, J., Q. Zhang, L. Ljung, A. Benveniste, B. Delyon, P. Y. Glorennec, H. Hjalmarsson and A. Juditsky. Nonlinear black-box modeling in system identification: A unified approach. *Automatica*, **31**, pp. 1691-1724.

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

Wiener, N. (1958). *Nonlinear Problems in Random Theory.* The Technology Press M. I. T., and John Wiley and Sons, New York, NY.

Wigren, T. (1993). Recursive Prediction Error Identification Using the Nonlinear Wiener Model. *Automatica,* **29**, 1011-1025.

Wigren, T. (1994). Convergence analysis of recursive identification algorithms based on the nonlinear Wiener model. *IEEE Trans. Aut. Contr.,* **39**, 2191-2206.

Wigren, T. (2003). User choices and model validation in system identification using nonlinear Wiener models. *Proc. SYSID 2003,* Rotterdam, the Netherlands, August 27-29, 863-868.

Figures and Captions

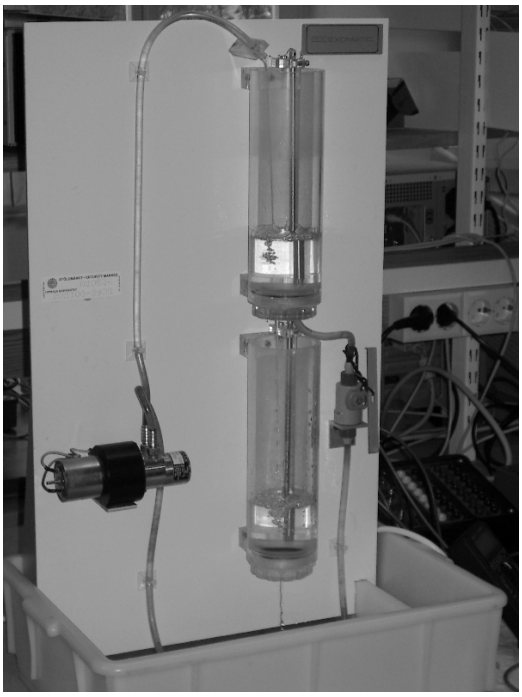

Figure 1: Photo of the cascaded tank laboratory process.

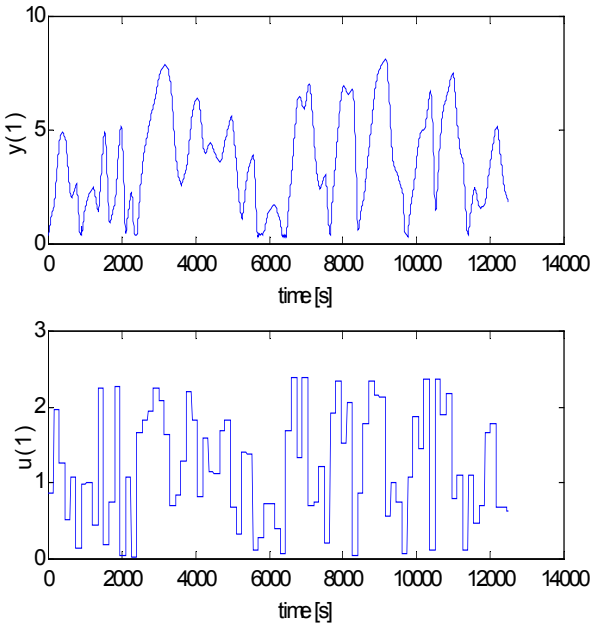

Figure 2: The input signal (bottom) and the output signal, i.e. the measured water level of the lower tank (top).

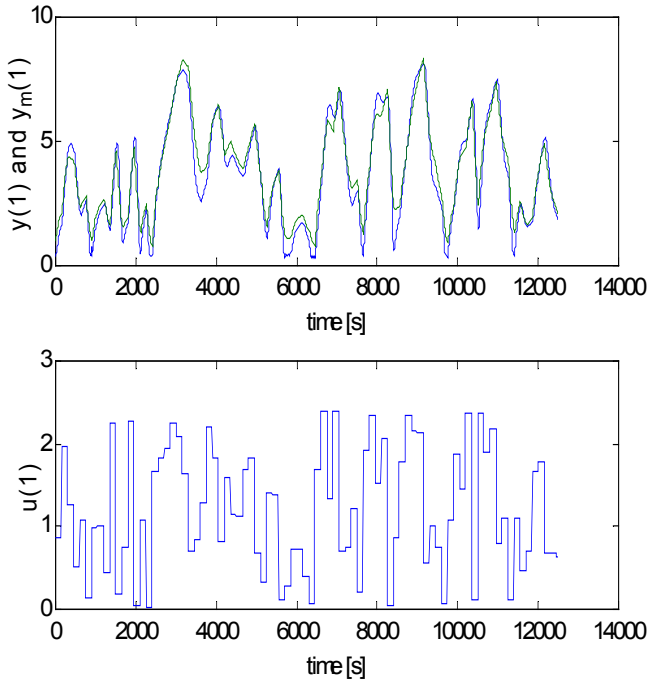

Figure 3: The output data plotted together with the simulated model, using parameters obtained at the end of the run. The lower subplot shows the input signal.

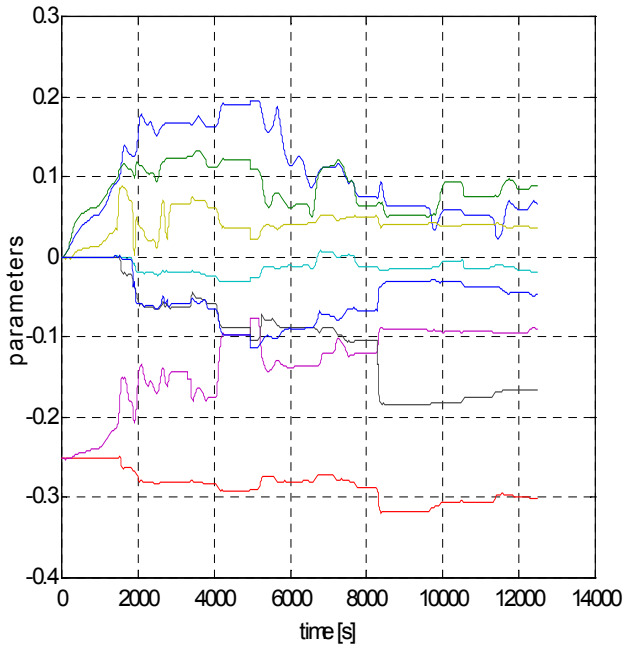

Figure 4: Evolution of the parameter estimates.

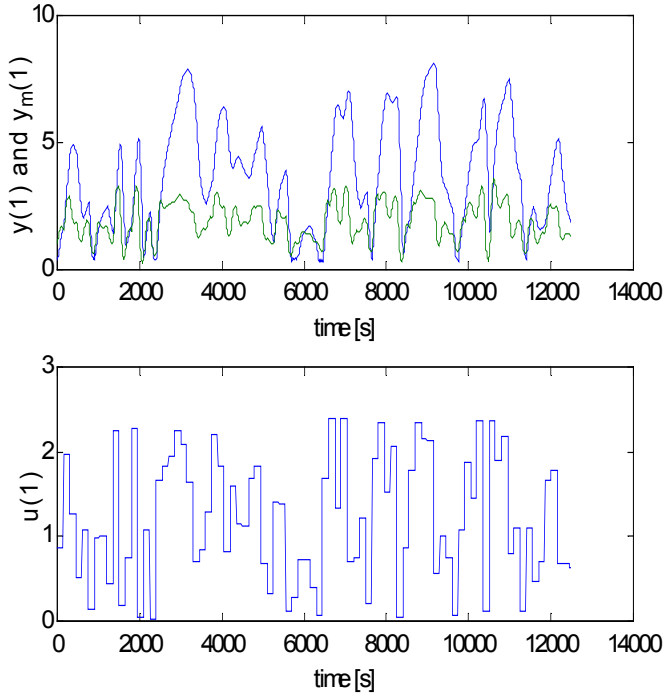

Figure 5: The data and the simulated model output, obtained with linear parameters. This is obtained by setting the fourth, sixth, seventh and eighth components of the parameter vector equal to zero.

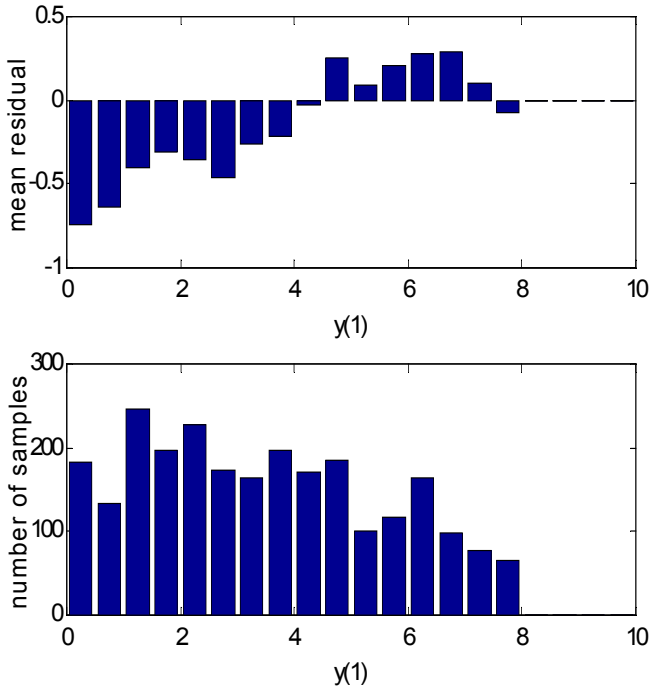

Figure 6: The result of validation with the mean residual analysis method.