RECURSIVE IDENTIFICATION BASED ON NONLINEAR STATE SPACE MODELS APPLIED TO DRUM-BOILER DYNAMICS WITH NONLINEAR OUTPUT EQUATIONS

Torbjörn Wigren, *Senior Member, IEEE*

Abstract—**The paper generalizes an RPEM based on a restricted nonlinear state space ODE model, to handle also nonlinear measurement equations. The coefficients of a multi-variable polynomial that models *one* right hand side component of the ODE are estimated, thereby avoiding overparameterization. It is discussed why this restricted model can also handle ODEs with more complicated right hand side structures. The paper also generalizes related results on scaling of the sampling period to the nonlinear output equation case. As an illustration, the RPEM was applied to simulated data obtained from a second order nonlinear drum-boiler model. A first order, four parameter, nonlinear model resulted in accurate modeling of the output power over a wide range of fuel flows.**

I. INTRODUCTION

Identification of nonlinear systems is of central importance in many engineering disciplines. Examples include power plant modeling and control [1]-[3], power grid modeling [4], modeling of chemical reactions [5], pulp digester modeling and control [6] as well as modeling of electric machinery [7], just to mention a few applications. A wide variety of methods have therefore been developed to solve nonlinear identification problems.

The available modeling methods are often classified as physical modeling, grey-box identification methods or black-box identification methods. Grey-box identification aims at combining physical modeling with system identification, see *e.g.* [8]. Physical modeling may then be applied to derive a model that contains unknown parameters, followed by a system identification step where the parameters are optimized against measured input and output data. Black-box identification methods apply a model that describes a certain class of nonlinear systems. In *e.g.* [9], a number of such identification methods are described within a unified framework. Important examples include *e.g.* the NARMAX class of methods [10] as well as different variants of neural networks [11].

In [12], [13] a recursive prediction error method (RPEM) based on a restricted black box parameterization of a state space ODE is described. The RPEM exploits a multi-variable polynomial that models *one* right hand side component of the ODE. The output (measurement) equation is assumed to be known and linear in [12], [13]. The model is discretized with an Euler numerical integration method and then used for derivation of the RPEM. The restricted model avoids overparameterization. Furthermore, results from [14] are used in [12], [13] to prove that the model can describe state space ODEs with more complicated right hand sides, albeit in other co-ordinate systems. The applicability of the method is hence quite wide. The use of an Euler integration scheme requires fast sampling, a fact that can cause ill-conditioned identification problems. This is however counteracted by the fact that the use of an Euler method allows an efficient scaling method to be developed. This method scales the sampling period applied in the identification algorithm. The effect is an exponential scaling of the states of the estimated model, [12], [13] and [15] for details.

One purpose of this paper is to present a generalization of the algorithm of [12], [13] to include a handling also of nonlinear measurement equations. Another purpose is to show that the results of [12], [13], regarding applicability of the model structure and the properties of the scaling method, still hold for nonlinear measurement equations. Finally, the paper applies the RPEM to simulated data from a simple drum-boiler model, using ideas and models from [1], [2].

The paper is organized as follows. Sections II and III presents the model structure and the resulting algorithm, respectively. Scaling properties follows in section IV, while section V discusses the results when the generalized RPEM is applied to simulated drum-boiler data. Conclusions follow in section VI.

II. THE MEASURED DATA AND THE MODEL

The purpose of the paper is to develop and apply an algorithm that estimates the parameter vector $\boldsymbol{\theta}$ of a nonlinear MIMO model from the measured input $\mathbf{u}(t)$ and output $\mathbf{y}_m(t)$, given by

Manuscript received September 27, 2004.

T. Wigren is with the Department of Information Technology, Uppsala University, P.O. Box 337, SE-75105, Uppsala, Sweden. phone: +46 18 4716226; e-mail: torbjorn.wigren@ it.uu.se.

0-7803-9098-9/05/$25.00 ©2005 AACC

$$\mathbf{u}(t)=\left(u_1(t)\;\;\ldots\;\;u_1^{(n_1)}(t)\;\;\ldots\;\;u_k(t)\;\;\ldots\;\;u_k^{(n_k)}(t)\right)^T$$

$$\mathbf{y}_m(t)=\left(y_{m,1}(t)\;\;\ldots\;\;y_{m,p}(t)\right)^T \tag{1}$$

The superscript $^{(k)}$ denotes differentiation k times. The starting point for the derivation of the model is the following n:th order state space ODE

$$\begin{pmatrix} x_1^{(1)} \\ \vdots \\ x_{n-1}^{(1)} \\ x_n^{(1)} \end{pmatrix} = \begin{pmatrix} x_2 \\ \vdots \\ x_n \\ f\left(x_1,\ldots,x_n,u_1,\ldots,u_1^{(n_1)},\ldots,u_k,\ldots,u_k^{(n_k)},\boldsymbol{\theta}\right) \end{pmatrix} \tag{2}$$

where $\mathbf{x}=\left(x_1\;\;\ldots\;\;x_{n-1}\;\;x_n\right)^T$ is the state vector. The linear measurement equation used in [12], [13] is here generalized to be the *known* nonlinear vector function

$$\begin{pmatrix} y_1 \\ \vdots \\ y_p \end{pmatrix} = \begin{pmatrix} c_1\left(x_1,\ldots,x_n,u_1,\ldots,u_k^{(n_k)}\right) \\ \vdots \\ c_p\left(x_1,\ldots,x_n,u_1,\ldots,u_k^{(n_k)}\right) \end{pmatrix}. \tag{3}$$

It should be noted that the generalization presented here, allows for compensation of known sensor nonlinearities. These may be substantial, a sensor insensitive to sign does *e.g.* correspond to an absolute value function.

Remark 1: The derivatives of the input signals may be required in (2). Practically, any need for input signal derivatives has consequences for input signal design and selection, a fact that may require further research. When input signal derivatives are required, a solution could be to first apply low pass filtering to the input signals.

The following polynomial parameterization of the right hand side function of (2) is then introduced,

$$\begin{aligned} & f\left(x_1,\ldots,x_n,u_1,\ldots,u_1^{(n_1)},\ldots,u_k,\ldots,u_k^{(n_k)},\boldsymbol{\theta}\right) \\ & =\sum_{i_{x_1}=0}^{I_{x_1}}\cdots\sum_{i_{x_n}=0}^{I_{x_n}}\sum_{i_{u_1}=0}^{I_{u_1}}\cdots\sum_{i_{u_1^{(n_1)}}=0}^{I_{u_1^{(n_1)}}}\cdots\sum_{i_{u_k}=0}^{I_{u_k}}\cdots\sum_{i_{u_k^{(n_k)}}}^{I_{u_k^{(n_k)}}} \\ & \theta_{i_{x_1}\ldots i_{x_n}i_{u_1}\ldots i_{u_1^{(n_1)}}\ldots i_{u_k}\ldots i_{u_k^{(n_k)}}}\left(x_1\right)^{i_{x_1}}\ldots\left(x_n\right)^{i_{x_n}}\left(u_1\right)^{i_{u_1}}\ldots \\ & \ldots\left(u_1^{(n_1)}\right)^{i_{u_1^{(n_1)}}}\ldots\left(u_k\right)^{i_{u_k}}\ldots\left(u_k^{(n_k)}\right)^{i_{u_k^{(n_k)}}}=\boldsymbol{\varphi}^T(\mathbf{x},\mathbf{u})\boldsymbol{\theta}. \end{aligned} \tag{4}$$

Here

$$\begin{aligned} \boldsymbol{\theta} = & \Big(\theta_{0\ldots0}\;\;\ldots\;\;\theta_{0\ldots I_{u_k^{(n_k)}}}\;\;\theta_{0\ldots010}\;\;\ldots\;\;\theta_{0\ldots01I_{u_k^{(n_k)}}}\;\;\ldots \\ & \theta_{0\ldots0I_{u_k^{(n_k-1)}}0}\;\;\ldots\;\;\theta_{0\ldots0I_{u_k^{(n_k-1)}}I_{u_k^{(n_k)}}}\;\;\ldots\;\;\theta_{I_{X_1}\ldots I_{u_k^{(n_k)}}}\Big)^T \\ \boldsymbol{\varphi} = & \Big(1\;\;\ldots\;\;\left(\left(u_k^{(n_k)}\right)^{I_{u_k^{(n_k)}}}\right)\;\;\;u_k^{(n_k-1)}\;\;\ldots \\ & \left(u_k^{(n_k-1)}\left(u_k^{(n_k)}\right)^{I_{u_k^{(n_k)}}}\right)\ldots\;\;\left(u_k^{(n_k-1)}\right)^{I_{u_k^{(n_k-1)}}}\;\;\ldots \\ & \left(\left(u_k^{(n_k-1)}\right)^{I_{u_k^{(n_k-1)}}}\left(u_k^{(n_k)}\right)^{I_{u_k^{(n_k)}}}\right)\;\;\ldots \\ & \left(\left(x_1\right)^{I_{x_1}}\ldots\left(x_n\right)^{I_{x_n}}\left(u_1\right)^{I_{u_1}}\ldots\left(u_k^{(n_k)}\right)^{I_{u_k^{(n_k)}}}\right)\Big)^T. \end{aligned} \tag{5}$$

Remark 2: Note that the selection of states is directly related to the nonlinear ODE

$$x^{(n)}=f\left(x,\ldots,x^{(n-1)},u_1,\ldots,u_1^{(n_1)},\ldots,u_k,\ldots,u_k^{(n_k)},\boldsymbol{\theta}\right). \tag{6}$$

Due to the selected parameterization no scale factor problems occur. The reason is that the transformation

$$z_n = kx_n \tag{7}$$

implies that

$$z_i = kx_i,\;\; i=1,\ldots,n. \tag{8}$$

Hence (7) transforms (2) to

$$\begin{pmatrix} z_1^{(1)} \\ \vdots \\ z_{n-1}^{(1)} \\ z_n^{(1)} \end{pmatrix} = \begin{pmatrix} z_2 \\ \vdots \\ z_n \\ kf\left(\frac{1}{k}z_1,\ldots,\frac{1}{k}z_n,u_1,,u_k^{(n_k)},\boldsymbol{\theta}\right) \end{pmatrix} \tag{9}$$

The polynomial model (4) can model the right hand side function of the last component of (9) equally well as the last component of the right hand side function of (2). This follows since the parameters are all free. Hence, the polynomial is capable of automatic scaling in response to any selected scaling of the output equation. In case only one output is present, the selected scale factor of the output equation is arbitrary, only the shape matters. Note that the output relation needs to be known a priori, any free scale factor also in the output equation would result in a singular problem.

Before the RPEM can be formulated, it remains to discretize the model. In order to achieve a discrete time model that is suitable for scaling, the Euler integration method is applied to (2) When the result is combined with (3) and (4) the following discrete time model is obtained

$$\begin{pmatrix} x_1(t+T_S,\boldsymbol{\theta}) \\ \vdots \\ x_{n-1}(t+T_S,\boldsymbol{\theta}) \\ x_n(t+T_S,\boldsymbol{\theta}) \end{pmatrix} = \begin{pmatrix} x_1(t,\boldsymbol{\theta}) \\ \vdots \\ x_{n-1}(t,\boldsymbol{\theta}) \\ x_n(t,\boldsymbol{\theta}) \end{pmatrix}$$

$$+T_S \begin{pmatrix} x_2(t,\boldsymbol{\theta}) \\ \vdots \\ x_n(t,\boldsymbol{\theta}) \\ \boldsymbol{\varphi}^T\left(x_1(t,\boldsymbol{\theta}),\ldots,x_n(t,\boldsymbol{\theta}),u_1(t),\ldots,u_k^{(n_k)}(t)\right)\boldsymbol{\theta} \end{pmatrix} \tag{10}$$

$$\begin{pmatrix} y_1(t,\theta) \\ \vdots \\ y_p(t,\theta) \end{pmatrix} = \begin{pmatrix} c_1\left(x_1(t,\theta),\ldots,x_n(t,\theta),u_1,\ldots,u_k^{(n_k)}\right) \\ \vdots \\ c_p\left(x_1(t,\theta),\ldots,x_n(t,\theta),u_1,\ldots,u_k^{(n_k)}\right) \end{pmatrix}. \tag{11}$$

Remark 3: Conditions are given in [12], [13] that imply the existence of a state transformation that transforms the ODE model

$$\begin{pmatrix} x_1^{(1)} \\ \vdots \\ x_{n-1}^{(1)} \\ x_n^{(1)} \end{pmatrix} = \begin{pmatrix} f_1(x_1,\ldots,x_n,u_1,\ldots,u_k,\boldsymbol{\theta}_1) \\ \vdots \\ f_{n-1}(x_1,\ldots,x_n,u_1,\ldots,u_k,\boldsymbol{\theta}_{n-1}) \\ f_n(x_1,\ldots,x_n,u_1,\ldots,u_k,\boldsymbol{\theta}_n) \end{pmatrix} \tag{12}$$

into (2). A closer study of the proof of this fact shows that the result still holds for the case where the nonlinear measurement equation (3) is restricted to

$$y_1 = c_1\left(x_1,u_1,\ldots,u_k^{(n_k)}\right). \tag{13}$$

The reader is referred to [12], [13] and [14] for details on regularity conditions and proofs. This results provides a theoretical foundation for the approach taken in this paper, since it follows that even in case the data is generated by a system with a structure according to (12) this system can still be identified with the model (2).

III. RPEM

The construction of the RPEM of this paper follows the standard approach of [16]. The covariance matrix of the measurement disturbances is estimated on-line. An output error identification approach is used since an application of least squares techniques to the problem requires that the complete state vector is measurable. The difference here as compared to [12], [13] is the known nonlinear measurement equation. This difference manifests itself in two ways in the resulting algorithm. First, (3) replaces the linear relation of [12], [13] in the output equation of the algorithm. Secondly, the partial derivative of (3) replaces the corresponding measurement matrix in the calculation of the gradient. The resulting RPEM algorithm is given by

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

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

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

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

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

$$\mathbf{y}(t+T_S) = \begin{pmatrix} c_1\left(\mathbf{x}(t+T_s),\mathbf{u}(t+T_S)\right) \\ \vdots \\ c_n\left(\mathbf{x}(t+T_S),\mathbf{u}(t+T_S)\right) \end{pmatrix}$$

$$\frac{d\boldsymbol{\varphi}}{dx_i}(t) = \Big(\mathbf{0}^T \quad 1 \quad u_k^{(n_k)}(t) \quad \ldots \tag{14}$$

$$\left(\left(x_{i+1}(t)\right)^{I_{x_{i+1}}}\ldots\left(x_n(t)\right)^{I_{x_n}}\left(u_1(t)\right)^{I_{u_1}}\ldots\left(u_k^{(n_k)}(t)\right)^{I_{u_k^{(n_k)}}}\right)$$

$$2x_1(t) \quad 2x_1(t)u_k^{(n_k)} \quad \ldots\Big), \quad i=1,\ldots,n$$

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

$$\begin{pmatrix} \frac{dx_1}{d\boldsymbol{\theta}}(t+T_S) \\ \vdots \\ \frac{dx_{n-1}}{d\boldsymbol{\theta}}(t+T_S) \\ \frac{dx_n}{d\boldsymbol{\theta}}(t+T_S) \end{pmatrix} = \begin{pmatrix} \frac{dx_1}{d\boldsymbol{\theta}}(t) \\ \vdots \\ \frac{dx_{n-1}}{d\boldsymbol{\theta}}(t) \\ \frac{dx_n}{d\boldsymbol{\theta}}(t) \end{pmatrix} + \alpha T_S$$

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

$$\boldsymbol{\psi}(t+T_S)$$

$$= \begin{pmatrix} \frac{\partial c_1}{\partial x_1}(t+T_s) & \cdots & \frac{\partial c_1}{\partial x_n}(t+T_s) \\ \vdots & \ddots & \vdots \\ \frac{\partial c_p}{\partial x_1}(t+T_S) & \cdots & \frac{\partial c_p}{\partial x_n}(t+T_S) \end{pmatrix} \begin{pmatrix} \frac{dx_1}{d\boldsymbol{\theta}}(t+T_S) \\ \vdots \\ \frac{dx_n}{d\boldsymbol{\theta}}(t+T_S) \end{pmatrix}.$$

The details of the notation are as follows. $\boldsymbol{\varepsilon}(t)$ is the prediction error, $\boldsymbol{\Lambda}(t)$ is the running estimate of the covariance matrix of the measurement disturbance, $\mu(t)/t$ is the gain sequence, $\mathbf{R}(t)$ is the running estimate of the Hessian and $\boldsymbol{\psi}(t)$ is the gradient of the output prediction with respect to the parameter vector. Note that the gradient is computed by means of a dynamic recursion, where the dynamics is obtained from the linearized state space model of the system. In order to keep the prediction and the gradient recursions stable, projection into the set of stable models D_M is applied. In the present case the model is nonlinear and stability may be difficult to determine. A linearized model is therefore used in the projection

$$\left[\boldsymbol{\theta}(t+T_S)\right]_{D_M} = \begin{cases} \boldsymbol{\theta}(t+T_S), & \boldsymbol{\theta}(t+T_S) \in D_M \\ \boldsymbol{\theta}(t), & \boldsymbol{\theta}(t+T_S) \notin D_M \end{cases} \tag{15}$$

$$D_M = \left\{ \boldsymbol{\theta} \middle| \left| eig\left(\mathbf{I}_n + \alpha T_S \frac{d}{d\mathbf{x}} \begin{pmatrix} x_2 \\ \vdots \\ x_{n-1} \\ \boldsymbol{\varphi}^T(x_1,\ldots,x_n,\mathbf{u})\boldsymbol{\theta} \end{pmatrix}\right)\right| < 1-d \right\} \tag{16}$$

$$d > 0.$$

Note that the projection algorithm (15), (16) automatically guarantees that the gradient dynamics has all eigenvalues strictly inside the unit circle. A drawback may be that the parameter estimates may become stuck on a trajectory that leave D_M, cf. [16].

The parameter α is a scale factor applied to the sampling period. The effect of this scale factor is discussed in the next section.

IV. Scaling

Scaling is a standard technique in optimization theory, see *e.g.* [17]. Usually, when scaling is applied in optimization, a linear transformation is applied to the estimated quantities. In practice scaling is often limited to application of a diagonal transformation matrix, cf. [7], [17]. Obviously, some knowledge of the range of the estimated parameters is a requirement for success, something that may be difficult to achieve for the black-box models applied in the present paper. Furthermore, as discussed in [12], [13], the origin of the poor scaling is often to be found in the *relative size* of the state vector components. Exactly this is addressed by the scaling method discussed in [12], [13]. The purpose here is to generalize these results to the case with nonlinear measurement equations. Towards that end, the scale factor α and the scaled sampling period T_S^{Scaled} are introduced by

$$T_S^{Scaled} = \alpha T_S \,. \tag{17}$$

It is stressed that in the algorithm, the only thing needed is to iterate the model and the gradient with a modified sampling period.

In order to analyze the effects of this scaling the measurement equation of (11) is assumed to be one-dimensional and a function only of x_1, *i.e.*

A1) $y_1(t,\boldsymbol{\theta}) = c_1(x_1(t,\boldsymbol{\theta}))$.

Furthermore, the following model is introduced to be compared to (10) and (11) (^{s} denotes scaling), where (11) obeys A1).

$$\begin{pmatrix} x_1^s(t+T_S,\boldsymbol{\theta}^s) \\ \vdots \\ x_{n-1}^s(t+T_S,\boldsymbol{\theta}^s) \\ x_n^s(t+T_S,\boldsymbol{\theta}^s) \end{pmatrix} = \begin{pmatrix} x_1^s(t,\boldsymbol{\theta}^s) \\ \vdots \\ x_{n-1}^s(t,\boldsymbol{\theta}^s) \\ x_n^s(t,\boldsymbol{\theta}^s) \end{pmatrix}$$
$$+ T_S^{Scaled} \begin{pmatrix} x_1^s(t,\boldsymbol{\theta}^s) \\ \vdots \\ x_n^s(t,\boldsymbol{\theta}^s) \\ \boldsymbol{\varphi}^T\left(x_1^s(t,\boldsymbol{\theta}^s),\ldots,x_n^s(t,\boldsymbol{\theta}^s),u_1(t),\ldots,u_k^{(n_k)}(t)\right)\boldsymbol{\theta}^s \end{pmatrix} \tag{18}$$

$$y_1^s(t,\boldsymbol{\theta}) = c_1\left(x_1^s(t,\boldsymbol{\theta}^s)\right) \cdot \tag{19}$$

The following assumptions are then introduced

A2) $c_1(x_1)$ is strictly monotone and continuous.

A3) The algorithm converges to an exact description of the input - output properties of the system for (10), (11) as well as for (18), (19), *i.e.* $y_{m,1}(t) = y_1(t,\boldsymbol{\theta}) = y_1^s(t,\boldsymbol{\theta}^s) \neq 0$, $\forall t$.

Corollary 1 then follows

Corollary 1 [To Theorem 2, [12]]:
Consider the model (10), (11), and the corresponding scaled model (18), (19), where the scaling is performed as $T_S^{Scaled} = \alpha T_S$. Provided that A1) - A3) hold it follows that

$$\mathbf{x}^s(t,\boldsymbol{\theta}^s) = \mathbf{A}(\alpha)\ \mathbf{x}(t,\boldsymbol{\theta})$$
$$\mathbf{A}(\alpha) = \begin{pmatrix} 1 & 0 & \ldots & 0 \\ 0 & \alpha^{-1} & 0 & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \ldots & 0 & \alpha^{-(n-1)} \end{pmatrix}.$$

Proof: A1) and A2) imply that

$$c_1(x_1(t,\boldsymbol{\theta})) = c_1\left(x_1^s(t,\boldsymbol{\theta}^s)\right), \quad \forall t. \tag{20}$$

But since the known function c_1 is strictly monotone and continuous, the equation $c_1(x)=\gamma$ has exactly one solution provided that γ belongs to the range of c_1. The solution is the inverse c_1^{-1} of c_1. Applying this to (20) results in

$$x_1^s\left(t,\boldsymbol{\theta}^s\right)=x_1\left(t,\boldsymbol{\theta}\right),\ \ \forall t. \cdot \tag{21}$$

Equation (21) is equivalent to assumption C7 of Theorem 2 of [12]. A check shows that all other assumptions of that Theorem are fulfilled and Corollary 1 follows.

The relation between $\boldsymbol{\theta}$ and $\boldsymbol{\theta}^s$ in [12] is derived from Theorem 2 of [12], using the following additional assumption

A4) $\mathbf{0}<\delta_1\mathbf{I}<\dfrac{1}{N}\displaystyle\sum_{t=t_1}^{t_1+(N-1)T_S}\boldsymbol{\varphi}\left(\mathbf{x}(t,\boldsymbol{\theta}),\mathbf{u}(t)\right)\boldsymbol{\varphi}^T\left(\mathbf{x}(t,\boldsymbol{\theta}),\mathbf{u}(t)\right)$

$<\delta_2\mathbf{I}<\infty$, $\delta_1,\delta_2>0$, some finite $N\geq\dim(\boldsymbol{\theta})$, $\forall t_1$.

The relation between $\boldsymbol{\theta}$ and $\boldsymbol{\theta}^s$ follows as

Corollary 2 [To Theorem 3, [12]]:
Consider the model (10), (11), and the corresponding scaled model (18), (19), where the scaling is performed as $T_S^{Scaled}=\alpha T_S$. Provided that A1) - A3) hold it then follows that

$$\boldsymbol{\theta}=\mathbf{T}(\alpha)\ \boldsymbol{\theta}^s$$
$$\mathbf{T}(\alpha)=diag_{i_{x_1},\ldots}\left(\alpha^{n-i_{x_2}-2i_{x_3}-\ldots-(n-1)i_{x_n}}\right)$$

Proof: As in [12].

Note that $diag$ refers to each component of $\boldsymbol{\varphi}$. The indices $i_{x_1},\ldots$ for each component of $\boldsymbol{\varphi}$ then follow from the numbering of components defined in (4)-(5).

Corollary 1 shows that the states of the model are scaled exponentially by $(1/\alpha)^{i-1}$, which affects the conditioning of the identification problem. Obviously, the scaling changes the estimated parameter vector. The original parameters can however be re-calculated using Corollary 2. Further comments are available in the companion paper [15], which is dedicated to properties of the scaling algorithm.

V. IDENTIFICATION OF A DRUM-BOILER TURBINE UNIT

As discussed *e.g.* in [3] and the references therein, knowledge of drum-boiler dynamics is of central importance for the power industry. Both physical modeling and system identification have been applied to the problem. Depending on the purpose, physical modeling methods of different fidelity has been applied. [3] *e.g.* present a nonlinear fourth order state space model that describes essential parts of the system and is still suitable for controller design.

Identification of drum-boiler dynamics can be performed with many of the large number of available identification methods. The purpose of this section is to illustrate the applicability of the proposed RPEM to a drum-boiler turbine unit, and to illustrate the use of knowledge of nonlinear output equations. A relatively simple model, taken from [1], [2], [7], of the system displayed in Fig. 1 is used.

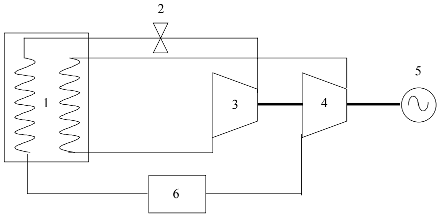

Fig. 1. Simplified block diagram of a drum-boiler turbin unit (from [7]). Steam is generated in the boiler 1. It then passes the control valve 2 and expands in the high-pressure turbine 3. After re-heating in the boiler 1, a further expansion takes place in the intermediate and low-pressure turbines 4, followed by condensation 6. Electricity is produced by the alternator, 5.

The following second order model can be derived, for the specific plant described in [7] .

$$\begin{aligned}\begin{pmatrix}\dot{x}_1(t)\\ \dot{x}_2(t)\end{pmatrix}&=\begin{pmatrix}0.014u_1(t)-0.0033(x_1(t))^{1.125}u_2(t)\\ 0.01408(x_1(t))^{1.125}u_2(t)-0.064x_2(t)\end{pmatrix}\\ y(t)&=0.2957(x_1(t))^{1.125}u_2(t)+3.456x_2(t)+e(t)\,.\end{aligned}\tag{22}$$

Here $x_1(t)$ is the drum pressure in kg/m^2, $x_2(t)$ is the re-heater pressure in kg/m^2, $u_1(t)$ is the fuel (oil) flow in $tons/hours$, $u_2(t)\in[0,1]$ is the control valve position, while $y(t)$ is the output power in MW. $N=2000$ samples of data was generated by integration of (22) with an Euler method. The valve position $u_2(t)$ was fixed at 0.8, while the fuel flow $u_1(t)$ was varied between 0 and 70 $tons/hours$. The sampling period was 10 s. A disturbance $e(t)$ with a variance of 1 MW was added.

Example 1: The algorithm (14) was applied to the data generated by (22), using the software described in [18]. A first order model was selected with the fuel flow as input and the total output power as output signal. $I_{x_1}=I_{u_1}=1$ was used. Reflecting (22), the measurement equation

$$y(t)=0.8(x_1(t))^{1.125} \tag{23}$$

was postulated. The projection algorithm used $1-d=0.975$. (14) was initiated with $\boldsymbol{\Lambda}=1$, $\mathbf{R}(0)=100\mathbf{I}$ and $\hat{\boldsymbol{\theta}}^{s}(0)=\begin{pmatrix}0.00 & 1.00 & -0.10 & 0.00\end{pmatrix}^{T}$.

To improve convergence the scale factor $\alpha=0.1$ was found to be suitable. The result of the identification is displayed in Fig. 2 and Fig. 3. The parameters at the end of the run were

$$\begin{aligned}\hat{\theta}^{s}(N)&=\begin{pmatrix}0.014 & 0.14 & -0.044 & -0.000080\end{pmatrix}^{T}\\ \hat{\theta}(N)&=\begin{pmatrix}0.0014 & 0.014 & -0.0044 & -0.0000080\end{pmatrix}^{T}\end{aligned} \qquad (24)$$

It can be noted that the agreement between the estimated model and the data is good. Since the fast mode of (22) is not particularly visible in the data, the algorithm tends to model the slow mode. The estimated parameters $\hat{\theta}_{01}(N)$ and $\hat{\theta}_{10}(N)$ are therefore close to the corresponding coefficients of the slow mode that is represented by the first state equation of (22). The nonlinear effect is noticeable in Fig. 4 that plots the model output with $\hat{\theta}_{11}(N)=0$.

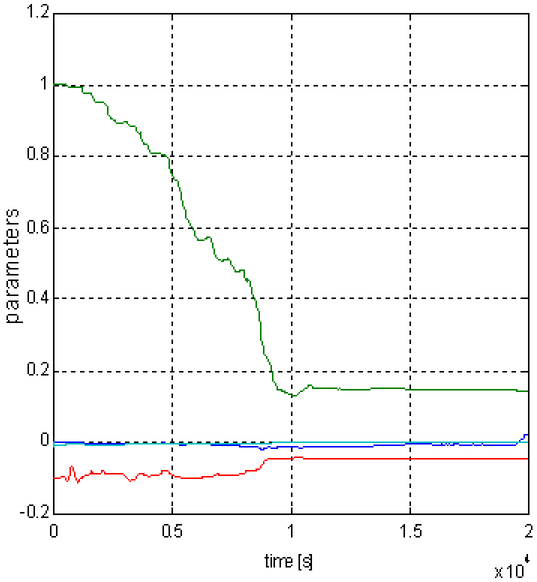

Fig. 2. Convergence of the parameter estimates.

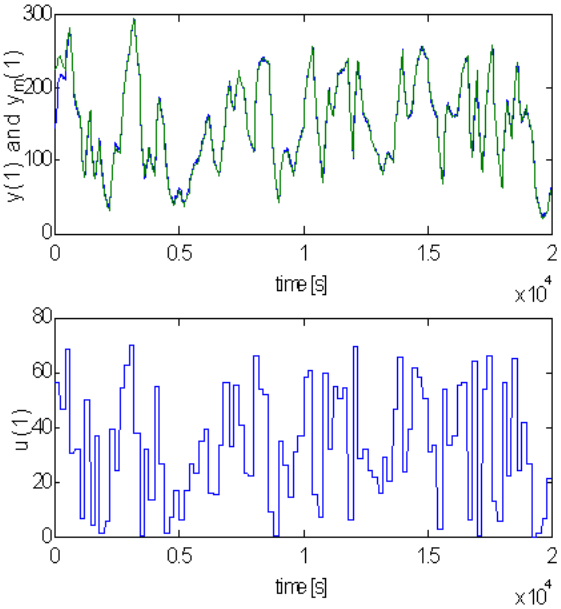

Fig. 3. The fuel flow input (bottom), together with the system and simulated model output.

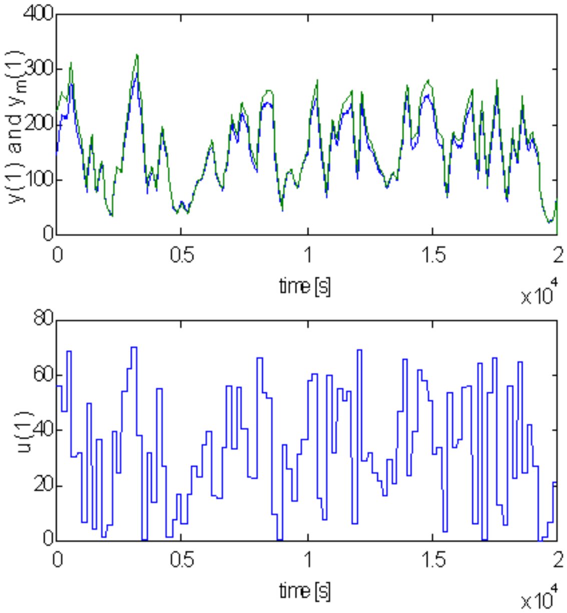

Fig. 4. The fuel flow input (bottom), together with the system and simulated model output.

Example 2: It may be argued that it is not realistic to assume knowledge of the exponent $^{1.125}$ in the output equation. For this reason, the experiments of Example 1 were repeated with (23) replaced by

$$y(t)=0.8x_{1}(t)\,. \qquad (25)$$

In this case the result was

$$\begin{aligned}\hat{\theta}^{s}(N) &= \left(-0.015 \quad 0.26 \quad -0.049 \quad -0.000065\right)^{T} \\ \hat{\theta}(N) &= \left(-0.0015 \quad 0.026 \quad -0.0049 \quad -0.0000065\right)^{T}\end{aligned} \qquad (26)$$

which appears to be a little further away from the parameters corresponding to the dominating mode of (22). The input output properties are however still comparable to those of Fig. 3.

VI. CONCLUSIONS

An RPEM based on a restricted black box state space model was generalized to allow a handling of known nonlinearities in the output equation. The paper also generalized previous theoretical result on the effect of scaling of the sampling period to this case. As an illustration the algorithm was applied to simulated data from a simplified model of a second order drum-boiler turbine unit. Good agreement was obtained between the simulated data and an identified first order nonlinear model, using only 4 parameters.

Some topics for further research include identification of more complicated higher order dynamics. An obvious step is to use identified nonlinear models for controller design. Further studies of applications, as in [19] and [20], would be very interesting. The software package [18], containing the algorithm and the support functions used in this paper, is available for free download from http://www.it.uu.se/research/reports/ for such purposes. Algorithms for initialization of the proposed RPEM are also needed, considering the risk for convergence to false local minimum points of the criterion function.

REFERENCES

[1] K. J. Åström and K. Eklund, ''A simplified non-linear model of a drum-boiler turbine unit'', *Int. J. Control,* vol. 16, pp. 145-169, 1972.

[2] K. J. Åström and K. Eklund, ''A simple non-linear drum boiler model'', *Int. J. Control,* vol. 22, pp. 739-740, 1975.

[3] K. J. Åström and R. D. Bell, ''Drum-boiler dynamics'', *Automatica,* vol. 36, pp. 363-378, 2000.

[4] P. Fairley, ''The unruly power grid'', *IEEE Spectrum,* vol. 41, pp. 16-21, 2004.

[5] G. A. Pajunen, ''Adaptive control of Wiener type nonlinear systems'', *Automatica,* vol. 28, pp. 781-785, 1992.

[6] J. Funkqvist, ''On modeling and identification of a continuous pulp digester'', *Proc. SYSID 1994,* Copenhagen, Denmark, 1994.

[7] L. Ekstam and T. Smed, ''Parameter estimation in dynamic systems with application to power engineering'', *Tekn. Lic. Thesis, UPTEC 8747R,* 1987.

[8] T. Bohlin, ''A case study of grey box identification'' , *Automatica,* vol. 30, pp. 307-318, 1994.

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

[10] L. Chen and S. A. Billings, ''Representation of nonlinear systems: the NARMAX model'', *Int. J. Control,* vol. 49, pp. 1013-1032, 1989.

[11] J. Sjöberg and L. Ljung, ''Overtraining, regularization, and searching for minimum in neural networks'', *Proc. of Symp. on Adaptive Systems in Control and Signal Processing,* Grenoble, Switzerland, 1992.

[12] T. Wigren, ''Recursive prediction error identification and scaling of nonlinear state space models using a restricted black box parameterization'', *submitted to Automatica,* 2004.

[13] T. Wigren, ''Recursive prediction error identification of nonlinear state space models'', *Technical Report 2004-004, Technical Reports from the Department of Information Technology,* Uppsala University, Uppsala, Sweden, 2004.

[14] H. Nijmeijer and A. J. van der Schaft, *Nonlinear Dynamic Control Systems.* New York: Springer Verlag, 1990.

[15] T. Wigren, "Scaling of the sampling period in nonlinear system identification", *in Proc. Of ACC 2005*, Portland, Oregon, U.S.A., June 8-10, 2005.

[16] L. Ljung and T. Söderström, *Theory and Practice of Recursive Identification.* Cambridge, MA: M. I. T. Press, 1983

[17] D. G. Luenberger, *Linear and Nonlinear Programming, 2:nd edition.* Reading, MA: Addison-Wesley, 1984.

[18] T. Wigren, ''MATLAB software for recursive identification and scaling using a structured nonlinear black-box model – Revision 1'', *Technical Report 2005-002, Technical Reports from the Department of Information Technology,* Uppsala University, Uppsala, January, 2005.

[19] L. Brus, "Nonlinear identification of a solar heating system", *submitted to CCA 2005,* Toronto, Canada, 2005.

[20] L. Brus, "Nonlinear identification of an anaerobic digestion process", *submitted to CCA 2005,* Toronto, Canada, 2005.