SCALING OF THE SAMPLING PERIOD IN NONLINEAR SYSTEM IDENTIFICATION

Torbjörn Wigren, *Senior Member, IEEE*

Abstract—**The paper presents a scaling algorithm for system identification, based on a nonlinear black box differential equation model. The model is discretized by an Euler forward numerical integration scheme. A scale factor is applied to the explicitly appearing *sampling period,* when iterating the discrete time state space model and the corresponding gradient recursion. The result is an exponential scaling of the state components of the model, and a scaling of the estimated parameter vector. The original parameter vector can be explicitly calculated from the scaled parameter vector using a diagonal matrix that is a function only of the scale factor. A new analysis of the effect of scaling on the Hessian, shows how the same diagonal matrix affects its eigenvalue distribution. A simulation study illustrates the beneficial effects on *e.g.* the condition number that can be obtained with the algorithm.**

I. INTRODUCTION

Scaling of state variables has been a central tool in optimization for a long period of time, see *e.g.* [1]. By scaling it is possible to affect the eigenvalues of the optimization problem, thereby influencing and improving convergence speed as well as other properties of nonlinear algorithms. Despite of this, scaling has not achieved that much of attention in system identification. One reason for this is perhaps the wide application of least-squares optimization, which is made possible by the wide use of linear models in the system identification field.

Recently, there has been an increasing interest in techniques for nonlinear system identification, see *e.g.* [2]. An important reason is of course the vast number of applications in many engineering fields, see *e.g.* [3]-[5] for a few examples. In nonlinear system identification, the use of least squares techniques is less wide spread and the set of methods is more scattered. Hence scaling may very well have much more to offer in nonlinear system identification. Scaling should be of particular interest in cases where steepest decent type algorithms are applied. Then the convergence speed is directly related to the eigenvalue spread of the Hessian [1]. It should be noted that steepest decent algorithm may be beneficial to use in tracking applications, where their performance is sometimes close to that of more advanced second order methods [6]. Even in situations where more advanced second order algorithms *e.g.* of Gauss-Newton type are applied, scaling can be expected to be effective during the initial transient stage, where the estimate is far from the true minimum point. Then the criterion function is unlikely to be even close to quadratic, with the consequence that the self-scaling property of second order algorithms deteriorates [1].

Among the methods available for identification of nonlinear systems, the simplest ones are perhaps the block-oriented models as described *e.g.* in [7]. More advanced approaches include grey-box techniques, where a model may first be constructed from physical principles. The unknown parameters of the model are then estimated with an optimization algorithm, often in combination with a numerical integration scheme, cf. *e.g.* [8]. Among black box methods, the NARMAX methods [9] have achieved quite a lot of attention. There a nonlinear discrete time difference equation is used as model. As discussed in [2], [10], also neural networks are common tools.

This paper is based on a MIMO black box nonlinear state space model, formulated in continuous time. The model is restricted by the use of *only one* component of the ODE to model the unknown right hand side function. The advantage of this restriction is that overparameterization is avoided. Further, as discussed in [11] using results in [12] *the model is still capable of modeling systems with more complicated right hand structure* (although the result is obtained in another co-ordinate system). The right hand side function of the ODE is then parameterized by the coefficients of a multi-variable polynomial in the states and inputs. The output measurement equation is assumed to be linear and known in this paper. A generalization appears in [13]. Recent applications to a solar heating system and an anaerobic digestion process are described in [14] and [15], respectively.

The contributions of the paper include a new analysis of the effect of the scaling of the sampling period, on the Hessian. The analysis assumes that a prediction error type criterion is used. A second contribution consists of a simulation study that verifies and illustrates the effect on

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.

scaling on i) the state variables, ii) the estimated parameters and iii) the eigenvalues and the condition number of the Hessian. Among other things the study indicates that large improvements of the condition number can be obtained by applying the proposed scaling algorithm. One other advantage of the proposed scaling algorithm is that prior knowledge of the range of parameters and/or state signals is not needed. Note also that the scaling algorithm, as well as the corresponding analysis of the paper, is applicable to any identification algorithm that is based on the proposed nonlinear dynamic model.

The paper is organized as follows. Section II defines the nonlinear state space model for which the scaling algorithm is defined. Section III presents an analysis of the effect of the scaling, while a corresponding simulation study is presented in section IV. The conclusions appear in section V.

II. THE NONLINEAR STATE SPACE MODEL

The nonlinear MIMO model to be defined here is used for estimation of an unknown parameter vector $\boldsymbol{\theta}$ from measured inputs $\mathbf{u}(t)$ and outputs $\mathbf{y}_m(t)$, given by

$$\mathbf{u}(t) = \left(u_1(t) \;\; \ldots \;\; u_1^{(n_1)}(t) \;\; \ldots \;\; u_k(t) \;\; \ldots \;\; u_k^{(n_k)}(t)\right)^T \qquad (1)$$
$$\mathbf{y}_m(t) = \left(y_{m,1}(t) \;\; \ldots \;\; y_{m,p}(t)\right)^T$$

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

$$\begin{pmatrix} x_1^{(1)} \\ \vdots \\ x_{n-1}^{(1)} \\ x_n^{(1)} \end{pmatrix} = \begin{pmatrix} x_2 \\ \vdots \\ x_n \\ f\left(x_1,\ldots,x_n,u_1,\ldots,u_1^{(n_1)},\ldots,u_k,\ldots,u_k^{(n_k)},\boldsymbol{\theta}\right) \end{pmatrix}$$
$$\begin{pmatrix} y_1 \\ \vdots \\ y_p \end{pmatrix} = \begin{pmatrix} c_{11} & \ldots & c_{1n} \\ \vdots & \ddots & \vdots \\ c_{p1} & \ldots & c_{pn} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}, \qquad (2)$$

where $\mathbf{x} = \left(x_1 \;\; \ldots \;\; x_{n-1} \;\; x_n\right)^T$ is the state vector. The following polynomial parameterization of the right hand side function of (2) is used

$$f\left(x_1,\ldots,x_n,u_1,\ldots,u_1^{(n_1)},\ldots,u_k,\ldots,u_k^{(n_k)},\boldsymbol{\theta}\right)$$
$$= \sum_{i_{x_1}=0}^{I_{x_1}} \cdots \sum_{i_{x_n}=0}^{I_{x_n}} \sum_{i_{u_1}=0}^{I_{u_1}} \cdots \sum_{i_{u_1^{(n_1)}}=0}^{I_{u_1^{(n_1)}}} \cdots \sum_{i_{u_k}=0}^{I_{u_k}} \cdots \sum_{i_{u_k^{(n_k)}}}^{I_{u_k^{(n_k)}}}$$
$$\theta_{i_{x_1}\ldots i_{x_n} i_{u_1}\ldots i_{u_1^{(n_1)}}\ldots i_{u_k}\ldots i_{u_k^{(n_k)}}} \left(x_1\right)^{i_{x_1}} \ldots \left(x_n\right)^{i_{x_n}} \left(u_1\right)^{i_{u_1}} \ldots$$
$$\ldots \left(u_1^{(n_1)}\right)^{i_{u_1^{(n_1)}}} \ldots \left(u_k\right)^{i_{u_k}} \ldots \left(u_k^{(n_k)}\right)^{i_{u_k^{(n_k)}}} = \boldsymbol{\varphi}^T(\mathbf{x},\mathbf{u})\boldsymbol{\theta} \,. \qquad (3)$$

Here

$$\boldsymbol{\theta} = \left(\theta_{0\ldots0} \;\; \ldots \;\; \theta_{0\ldots I_{u_k^{(n_k)}}} \;\; \theta_{0\ldots010} \;\; \ldots \;\; \theta_{0\ldots01I_{u_k^{(n_k)}}} \;\; \ldots\right.$$
$$\left.\theta_{0\ldots0I_{u_k^{(n_k-1)}}0} \;\; \ldots \;\; \theta_{0\ldots0I_{u_k^{(n_k-1)}}I_{u_k^{(n_k)}}} \;\; \ldots \;\; \theta_{I_{X_1}\ldots I_{u_k^{(n_k)}}}\right)^T$$
$$\boldsymbol{\varphi} = \left(1 \;\; \ldots \;\; \left(\left(u_k^{(n_k)}\right)^{I_{u_k^{(n_k)}}}\right) \;\; u_k^{(n_k-1)} \;\; \ldots\right.$$
$$\left(u_k^{(n_k-1)}\left(u_k^{(n_k)}\right)^{I_{u_k^{(n_k)}}}\right) \ldots \;\; \left(u_k^{(n_k-1)}\right)^{I_{u_k^{(n_k-1)}}} \;\; \ldots \qquad (4)$$
$$\left(\left(u_k^{(n_k-1)}\right)^{I_{u_k^{(n_k-1)}}}\left(u_k^{(n_k)}\right)^{I_{u_k^{(n_k)}}}\right) \;\; \ldots$$
$$\left.\left(\left(x_1\right)^{I_{x_1}} \ldots \left(x_n\right)^{I_{x_n}} \left(u_1\right)^{I_{u_1}} \ldots \left(u_k^{(n_k)}\right)^{I_{u_k^{(n_k)}}}\right)\right)^T .$$

In order to obtain a discrete time model that is suitable for scaling, the Euler integration method is applied to (2). A main reason for using the Euler method is that the sampling appears explicitly and linearly in the right hand side of the resulting difference equation model (5). The result of the discretization is

$$\begin{pmatrix} x_1(t+T_S,\boldsymbol{\theta}) \\ \vdots \\ x_{n-1}(t+T_S,\boldsymbol{\theta}) \\ x_n(t+T_S,\boldsymbol{\theta}) \end{pmatrix} = \begin{pmatrix} x_1(t,\boldsymbol{\theta}) \\ \vdots \\ x_{n-1}(t,\boldsymbol{\theta}) \\ x_n(t,\boldsymbol{\theta}) \end{pmatrix}$$
$$+ T_S \begin{pmatrix} x_2(t,\boldsymbol{\theta}) \\ \vdots \\ x_n(t,\boldsymbol{\theta}) \\ \boldsymbol{\varphi}^T\left(x_1(t,\boldsymbol{\theta}),\ldots,x_n(t,\boldsymbol{\theta}),u_1(t),\ldots,u_k^{(n_k)}(t)\right)\boldsymbol{\theta} \end{pmatrix} \qquad (5)$$

$$\begin{pmatrix} y_1(t,\boldsymbol{\theta}) \\ \vdots \\ y_p(t,\boldsymbol{\theta}) \end{pmatrix} = \begin{pmatrix} c_{11} & \ldots & c_{1n} \\ \vdots & \ddots & \vdots \\ c_{p1} & \ldots & c_{pn} \end{pmatrix} \begin{pmatrix} x_1(t,\boldsymbol{\theta}) \\ \vdots \\ x_n(t,\boldsymbol{\theta}) \end{pmatrix} = \mathbf{C}\mathbf{x}(t,\boldsymbol{\theta}) \,. \qquad (6)$$

It can be remarked that the Euler method may require fast sampling in order not to introduce significant discretization errors. This is fortunately a less important effect in system identification applications. The reason is that the minimization algorithm uses the parameters as instruments to fit the model output to the measured data, as expressed by the criterion function. Even if an additional bias would be introduced in the estimated parameters, the input output properties of the identified model should still resemble the behavior of the system. The reader is referred to [11], [13] for a further discussion of various properties of the model (5), (6).

III. SCALING OF THE SAMPLING PERIOD

A. Scaling

The need for scaling arises whenever estimated quantities differ in size by large amounts. Linear scaling as applied to nonlinear system identification introduces a nonsingular linear transformation $\mathbf{T}$ of the identified parameters as

$$\tilde{\boldsymbol{\theta}} = \mathbf{T}\boldsymbol{\theta} \Leftrightarrow \boldsymbol{\theta} = \mathbf{T}^{-1}\tilde{\boldsymbol{\theta}} . \tag{7}$$

The transformation is then exploited in the criterion function and a minimization of the criterion is performed with respect to $\tilde{\boldsymbol{\theta}}$ instead of with respect to $\boldsymbol{\theta}$. In case a prediction error criterion is used, the following optimization problem then results

$$\tilde{\boldsymbol{\theta}} = \arg\min_{\tilde{\boldsymbol{\theta}}} \frac{1}{2} E\left[\boldsymbol{\varepsilon}^T\left(t, \mathbf{T}^{-1}\tilde{\boldsymbol{\theta}}\right)\boldsymbol{\varepsilon}\left(t, \mathbf{T}^{-1}\tilde{\boldsymbol{\theta}}\right)\right] \tag{8}$$

where $\boldsymbol{\varepsilon}(t,\boldsymbol{\theta})$ denotes the prediction error and where $E[.]$ denotes the expectation operator. In practice scaling transformations that are diagonal is usually what the prior knowledge allows, cf. [1].

A successful application of (7) requires knowledge of the expected range of all parameters, something that may be difficult to obtain for black box models. Further, the origin of the scaling problems may very well be the relative size of the state signals that are generated by the algorithm. Addressing this would require a corresponding scaling of the state variables. Also this would require prior knowledge of the range of these signals. As will be seen below, the proposed scaling of the sampling period almost completely avoids the need for prior knowledge of the range of parameters and state signals.

B. Scaling Algorithm

During development of the RPEM described in [13], [16] it was noticed that problems with convergence to false local minimum points of the criterion were 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 of the identified system. Hence the sampling period applied for measurement cannot be arbitrarily selected. However, since the sampling period appears *explicitly* in the model (5) and in the corresponding gradient difference equation, *it is straightforward to apply identification algorithms based on (5) with another, scaled value of the sampling period.* This idea affects the updating of the states, the gradient, and any projection algorithm that is used to control the stability of the model. A scale factor α appears before the multiplication with the sampling period T_S in those three quantities. To explain the details, the scale factor α and the scaled sampling period T_S^{Scaled} are first defined as

$$T_S^{Scaled} = \alpha T_S . \tag{9}$$

The model (5), (6), as applied in the identification algorithm is then transformed into

$$\begin{pmatrix} x_1^s\left(t+T_S, \boldsymbol{\theta}^s\right) \\ \vdots \\ x_{n-1}^s\left(t+T_S, \boldsymbol{\theta}^s\right) \\ x_n^s\left(t+T_S, \boldsymbol{\theta}^s\right) \end{pmatrix} = \begin{pmatrix} x_1^s\left(t, \boldsymbol{\theta}^s\right) \\ \vdots \\ x_{n-1}^s\left(t, \boldsymbol{\theta}^s\right) \\ x_n^s\left(t, \boldsymbol{\theta}^s\right) \end{pmatrix} + T_S^{Scaled} \begin{pmatrix} x_2^s\left(t, \boldsymbol{\theta}^s\right) \\ \vdots \\ x_n^s\left(t, \boldsymbol{\theta}^s\right) \\ f\left(x_1^s\left(t, \boldsymbol{\theta}^s\right), \ldots . x_n^s\left(t, \boldsymbol{\theta}^s\right), u_1(t), \ldots, u_k^{(n_k)}(t), \boldsymbol{\theta}^s\right) \end{pmatrix} \tag{10}$$

$$\begin{pmatrix} y_1^s\left(t, \boldsymbol{\theta}^s\right) \\ \vdots \\ y_p^s\left(t, \boldsymbol{\theta}^s\right) \end{pmatrix} = \begin{pmatrix} c_{11} & \ldots & c_{1n} \\ \vdots & \ddots & \vdots \\ c_{p1} & \ldots & c_{pn} \end{pmatrix} \begin{pmatrix} x_1^s\left(t, \boldsymbol{\theta}^s\right) \\ \vdots \\ x_n^s\left(t, \boldsymbol{\theta}^s\right) \end{pmatrix} = \mathbf{C}\mathbf{x}^s\left(t, \boldsymbol{\theta}^s\right) \tag{11}$$

where the superscript ^{s} denotes scaled quantities. Note that the original sampling period must be retained in all time arguments, so as to refer to the correct measurement times.

The gradient follows by differentiation of (10) and (11)

$$\frac{d\mathbf{x}^s\left(t+T_S, \boldsymbol{\theta}^s\right)}{d\boldsymbol{\theta}^s} = \frac{d\mathbf{x}^s\left(t, \boldsymbol{\theta}^s\right)}{d\boldsymbol{\theta}^s} + T_S^{Scaled} \begin{pmatrix} \frac{dx_2^s\left(t, \boldsymbol{\theta}^s\right)}{d\boldsymbol{\theta}^s} \\ \vdots \\ \frac{dx_n^s\left(t, \boldsymbol{\theta}^s\right)}{d\boldsymbol{\theta}^s} \\ \boldsymbol{\varphi}^T\left(\mathbf{x}^s\left(t, \boldsymbol{\theta}^s\right), \mathbf{u}(t)\right) \end{pmatrix} + T_S^{Scaled} \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \boldsymbol{\theta}^T \left(\frac{d\boldsymbol{\varphi}\left(\mathbf{x}^s\left(t, \boldsymbol{\theta}^s\right), \mathbf{u}(t)\right)}{d\mathbf{x}^s} \right) \left(\frac{d\mathbf{x}^s\left(t, \boldsymbol{\theta}^s\right)}{d\boldsymbol{\theta}^s} \right) \end{pmatrix} \tag{12}$$

$$\left(\boldsymbol{\psi}^s\left(t, \boldsymbol{\theta}^s\right)\right)^T = \frac{d\mathbf{y}^s\left(t, \boldsymbol{\theta}^s\right)}{d\boldsymbol{\theta}^s} = \mathbf{C}\frac{d\mathbf{x}^s\left(t, \boldsymbol{\theta}^s\right)}{d\boldsymbol{\theta}^s} . \tag{13}$$

Note that the above change from $\boldsymbol{\theta}$ to $\boldsymbol{\theta}^s$ is *not* to be treated as a change of variables in the differentiation leading to (12) and (13). The originally derived gradient is applied, but with a scaled sampling period.

The last affected quantity of the algorithm is the projection algorithm that becomes (cf. [11])

$$\mathbf{S}^s\left(\boldsymbol{\theta}^s\right) = I_n + T_S^{Scaled}\begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & 0 & \cdots & 0 & 1 \\ \multicolumn{5}{c}{\left(\boldsymbol{\theta}^s\right)^T \dfrac{d\boldsymbol{\varphi}\left(\mathbf{x}^s\left(t,\boldsymbol{\theta}^s\right)\right)}{d\mathbf{x}^s}} \end{pmatrix} \qquad (14)$$

$$D_M = \left\{\boldsymbol{\theta}^s \;\middle|\; \left|eig\left(S^s\left(\boldsymbol{\theta}^s\right)\right)\right| < 1-\delta\right\},\ \delta > 0 \qquad (15)$$

$$\left[\hat{\boldsymbol{\theta}}^s(t)\right]_{D_M} = \begin{cases} \hat{\boldsymbol{\theta}}^s(t) & \hat{\boldsymbol{\theta}}^s(t) \in D_M \\ \hat{\boldsymbol{\theta}}^s\left(t-T_S\right) & \hat{\boldsymbol{\theta}}^s(t) \notin D_M \end{cases} \qquad (16)$$

In (14), $\mathbf{S}^s\left(\boldsymbol{\theta}^s\right)$ denotes the linearized system matrix of the model, D_M denotes the model set, here defined as the asymptotically stable models with a margin δ to the stability limit. The last equation stops the updating of the parameter vector in case the update would result in values outside the model set. Other details of an RPEM where the scaling algorithm is used can be found in [11] and [13].

When 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 scaling algorithm experimentally, significant improvements were 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. This is illustrated in section IV of this paper.

C. Analysis

In order to analyze the effect of scaling of the sampling period, the following models are introduced to describe the original model and the scaled model, respectively

$$\begin{pmatrix} x_1\left(t+T_S,\boldsymbol{\theta}\right) \\ \vdots \\ x_{n-1}\left(t+T_S,\boldsymbol{\theta}\right) \\ x_n\left(t+T_S,\boldsymbol{\theta}\right) \end{pmatrix} = \begin{pmatrix} x_1\left(t,\boldsymbol{\theta}\right) \\ \vdots \\ x_{n-1}\left(t,\boldsymbol{\theta}\right) \\ x_n\left(t,\boldsymbol{\theta}\right) \end{pmatrix} + T_S \begin{pmatrix} x_2\left(t,\boldsymbol{\theta}\right) \\ \vdots \\ x_n\left(t,\boldsymbol{\theta}\right) \\ f\left(x_1(t,\boldsymbol{\theta}),\ldots.x_n(t,\boldsymbol{\theta}),u_1(t),\ldots,u_k^{(n_k)}(t),\boldsymbol{\theta}\right) \end{pmatrix} \qquad (17)$$

$$\begin{pmatrix} x_1^s\left(t+T_S,\boldsymbol{\theta}^s\right) \\ \vdots \\ x_{n-1}^s\left(t+T_S,\boldsymbol{\theta}^s\right) \\ x_n^s\left(t+T_S,\boldsymbol{\theta}^s\right) \end{pmatrix} = \begin{pmatrix} x_1^s\left(t,\boldsymbol{\theta}^s\right) \\ \vdots \\ x_{n-1}^s\left(t,\boldsymbol{\theta}^s\right) \\ x_n^s\left(t,\boldsymbol{\theta}^s\right) \end{pmatrix} + T_S^{Scaled} \begin{pmatrix} x_2^s\left(t,\boldsymbol{\theta}^s\right) \\ \vdots \\ x_n^s\left(t,\boldsymbol{\theta}^s\right) \\ f\left(x_1^s\left(t,\boldsymbol{\theta}^s\right),\ldots.x_n^s\left(t,\boldsymbol{\theta}^s\right),u_1(t),\ldots,u_k^{(n_k)}(t),\boldsymbol{\theta}^s\right) \end{pmatrix} \qquad (18)$$

The following assumptions are then introduced

C1) The measurement $y(t)$ corresponds to the states $x_1\left(t,\boldsymbol{\theta}\right)$ and $x_1^s\left(t,\boldsymbol{\theta}^s\right)$ of (15) and (16).

C2) The algorithm converges to an exact description of the input-output properties of the system for (15) and (16), *i.e.* $y(t)-e(t) = x_1\left(t,\boldsymbol{\theta}\right) = x_1^s\left(t,\boldsymbol{\theta}^s\right) \neq 0, \forall t$, where $e(t)$ denotes a zero mean additive measurement disturbance.

Note that C1) corresponds to the choice

$$\mathbf{C} = \begin{pmatrix} 1 & 0 & \ldots & 0 \end{pmatrix}. \qquad (19)$$

Note also that in case the system is in the model set, then C2) can be exactly achieved *e.g.* by the RPEM of [11].

The following result for the scaling of the state variables is implied by the above assumptions:

Theorem 1: Consider the two models (17) and (18) where T_S is the measurement sampling period and where $T_S^{Scaled} = \alpha T_S$ is the scaled sampling period that is applied when running the identification algorithm. Provided that C1) and C2) hold it follows that

$$\mathbf{x}^s\left(t,\boldsymbol{\theta}^s\right) = \mathbf{A}(\alpha)\ \mathbf{x}\left(t,\boldsymbol{\theta}\right)$$

$$\mathbf{A}(\alpha) = 0\begin{pmatrix} 1 & 0 & \ldots & 0 \\ 0 & \alpha^{-1} & 0 & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \ldots & 0 & \alpha^{-(n-1)} \end{pmatrix}.$$

Proof: See Appendix A.

When the scale factor $\alpha \neq 1$ is used other parameters result than when $\alpha = 1$ is applied. In case a relation to underlying physical continuous time parameters exist, this relation hence appears to be lost by the scaling of the sampling period. Fortunately, it turns out that an explicit relation between $\boldsymbol{\theta}$ and $\boldsymbol{\theta}^s$ can be derived, so that $\boldsymbol{\theta}$ can be computed from $\boldsymbol{\theta}^s$. This computation can be performed on-line or after the end of the identification run. In order to state this result the following assumption is needed

C3) $\mathbf{0} < \delta_1 \mathbf{I} < \frac{1}{N} \sum_{t=t_1}^{t_1+(N-1)T_s} \boldsymbol{\varphi}(\mathbf{x}(t,\boldsymbol{\theta}))\boldsymbol{\varphi}^T(\mathbf{x}(t,\boldsymbol{\theta})) < \delta_2 \mathbf{I} < \infty$,

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

Note that this condition bears a close resemblance with conditions for persistent excitation in linear system identification. Further work is needed to clarify the exact connection. The following result now holds for the scaling of the parameter vector:

Theorem2: Consider the two models (17) and (18) where T_S is the measurement sampling period and where $T_S^{Scaled} = \alpha T_S$ is the scaled sampling period that is applied when running the identification algorithm. Provided that C1), C2) and C3) hold it follows that the components of the scaled parameter vector $\boldsymbol{\theta}^s$ are related to $\boldsymbol{\theta}$ by

$$\boldsymbol{\theta} = \mathbf{T}(\alpha)\ \boldsymbol{\theta}^s$$

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

Proof: See Appendix B.

The ordering of diagonal elements in the transformation matrix of Theorem 3 follows the original ordering of components in $\boldsymbol{\theta}$, as defined by (3) and (4). The exponent of α follows from the exponents of the state variables that appear in the corresponding component of $\boldsymbol{\varphi}(\mathbf{x}(t,\boldsymbol{\theta}),\mathbf{u}(t))$.

An novel analysis of the impact of scaling on the Hessian of the identification algorithm can now be performed, using Theorem 1 and Theorem 2. By differentiation of the result of Theorem 1, it follows that

$$\frac{d\mathbf{x}^s(t,\theta^s)}{d\theta^s} = \mathbf{A}(\alpha)\ \frac{d\mathbf{x}(t,\theta)}{d\theta^s} = \mathbf{A}(\alpha)\ \frac{d\mathbf{x}(t,\boldsymbol{\theta})}{d\boldsymbol{\theta}}\ \frac{d\boldsymbol{\theta}}{d\boldsymbol{\theta}^s}$$
$$= \mathbf{A}(\alpha)\ \frac{d\mathbf{x}(t,\boldsymbol{\theta})}{d\boldsymbol{\theta}}\ \mathbf{T}(\alpha)\,. \tag{20}$$

In order to proceed the following assumption is needed

C4) The identification algorithm is based on minimization of $V(\boldsymbol{\theta}) = \frac{1}{2}E\left[\varepsilon^2\ (t,\boldsymbol{\theta})\right]$.

It follows that the Hessian of the criterion function of C4) can be calculated as follows

$$\frac{d^2}{d\boldsymbol{\theta}d\boldsymbol{\theta}^T}V(\boldsymbol{\theta}) = \frac{d}{d\boldsymbol{\theta}}E\left[\boldsymbol{\psi}(t,\boldsymbol{\theta})\ \varepsilon\ (t,\boldsymbol{\theta})\right]$$
$$= E\left[\boldsymbol{\psi}(t,\boldsymbol{\theta})\boldsymbol{\psi}^T(t,\boldsymbol{\theta})\right]$$
$$+ E\left[\left(\frac{d^2}{d\boldsymbol{\theta}d\boldsymbol{\theta}^T}\ \varepsilon\ (t,\boldsymbol{\theta})\right)\ \varepsilon\ (t,\boldsymbol{\theta})\right]. \tag{21}$$

By C2), the prediction error is uncorrelated with its derivatives with respect to the parameter vector in case either of the following assumptions hold.

C5a) The colored measurement disturbance $e(t)$ is zero mean, and the regression vector $\boldsymbol{\varphi}(\mathbf{x}(t,\boldsymbol{\theta}),\mathbf{u}(t))$ is generated solely from the input signal.

C5b) The measurement disturbance $e(t)$ is zero mean and white.

The assumption C5a) covers output error and instrumental variable type algorithms while C5b) is intended to cover algorithms of the least squares type.

Using C5a) or C5b), the second term of (21) is zero and the Hessian simplifies to

$$\mathbf{R}(\boldsymbol{\theta}) \equiv \frac{d^2}{d\boldsymbol{\theta}d\boldsymbol{\theta}^T}V(\boldsymbol{\theta}) = E\left[\boldsymbol{\psi}(t,\boldsymbol{\theta})\boldsymbol{\psi}(t,\boldsymbol{\theta})\right]. \tag{22}$$

The analysis of the effect of the scaling of the sampling period on the Hessian can now be finalized. Using (22) in the scaled case, together with (11) results in

$$\mathbf{R}^s(\boldsymbol{\theta}^s) = E\left[\left(\frac{d\mathbf{x}^s(t,\boldsymbol{\theta}^s)}{d\boldsymbol{\theta}^s}\right)^T \mathbf{C}^T\mathbf{C}\frac{d\mathbf{x}^s(t,\boldsymbol{\theta}^s)}{d\boldsymbol{\theta}^s}\right]. \tag{23}$$

The relation (20) then gives

$$\mathbf{R}^s(\theta^s) =$$
$$E[\mathbf{T}^T(\alpha)\left(\frac{d\mathbf{x}(t,\boldsymbol{\theta})}{d\boldsymbol{\theta}}\right)^T \mathbf{A}^T(\alpha)\ \mathbf{C}^T\mathbf{C}\ \mathbf{A}(\alpha)\ \frac{d\mathbf{x}(t,\boldsymbol{\theta})}{d\boldsymbol{\theta}}\ \mathbf{T}(\alpha)]$$
$$= E[\mathbf{T}^T(\alpha)\left(\frac{d\mathbf{x}(t,\boldsymbol{\theta})}{d\boldsymbol{\theta}}\right)^T \mathbf{C}^T\mathbf{C}\ \frac{d\mathbf{x}(t,\boldsymbol{\theta})}{d\boldsymbol{\theta}}\ \mathbf{T}(\alpha)]$$
$$= \mathbf{T}^T(\alpha)\ \mathbf{R}(\boldsymbol{\theta})\ \mathbf{T}(\alpha)\,, \tag{24}$$

where the second last step follows from the facts that $\mathbf{C}$ is given by (19) and that the (1,1)-element of $\mathbf{A}(\alpha)$ equals 1. This proves

Theorem 3: Consider the two models (17) and (18) where T_S is the measurement sampling period and where $T_S^{Scaled} = \alpha T_S$ is the scaled sampling period that is applied when running the identification algorithm. Provided that C1), C2), C3), C4) and one of C5a) or C5b) hold, it follows that the Hessians obey

$$\mathbf{R}^s(\theta^s) = = \mathbf{T}^T(\alpha)\ \mathbf{R}(\boldsymbol{\theta})\ \mathbf{T}(\alpha)$$

where $\mathbf{T}(\alpha)$ is defined by Theorem 2.

IV. NUMERICAL RESULTS

A simulation study was performed in order to study the proposed scaling algorithm numerically. A specific purpose was the verification of the results of Theorem 1, Theorem 2 and Theorem 3. The RPEM described in [11] and [16] was used for this purpose, in combination with simulated data from the system

$$\begin{pmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{pmatrix} = \begin{pmatrix} x_2(t)(2+u(t))-u(t) \\ -x_1(t)-x_2(t) \end{pmatrix}. \qquad (25)$$
$$y(t) = x_2(t) + e(t)$$

This system can be exactly described by the model (2), since it can be re-written as the second order ODE

$$\ddot{x}_2(t) + \dot{x}_2(t) + (2+u(t))x_2(t) = u(t). \qquad (26)$$
$$y(t) = x_2(t) + e(t)$$

Data was generated by simulation with the same Euler method as applied in the discretization of (2), thereby securing that the system was in the model set. 10000 samples were simulated using a sampling period of $T_S = 0.10s$. The input signal was selected as a PRBS, multiplied by a uniformly distributed (in amplitude) random variable, with a mean of 0 and range $[-1,1]$. The clock period of the original PRBS was 3.0s. The measurement disturbance was white, zero mean with a standard deviation of 0.1. The RPEM was initialized and run to convergence for a first value of the scale factor α after which all interesting quantities were recorded. The scale factor was then shifted slightly and the RPEM restarted. The initial parameter vector was then selected as the parameter vector obtained at the end of the run for the previous value of the scale factor. All parameters that controlled the adaptation gain were however re-initialized to their original values. Since a major purpose was to study the effect of the scaling on the Hessian, care was taken to ensure that the number of samples was high enough to secure convergence of the eigenvalues of the Hessian.

The algorithm was first run with a scale factor $\alpha = 2$. The RPEM was initialized with the values

$$\hat{\boldsymbol{\theta}}(0) = \begin{pmatrix} 0 & 1 & -1 & 0 & -0.25 & 0 & 0 & 0 \end{pmatrix}^T. \qquad (27)$$

The convergence of the parameters from their initial values is depicted in Fig. 1. Convergence to the true parameter vector takes place. Fig. 2 displays the convergence of the eigenvalues of the Hessian and it can be seen that also the Hessian has converged at the end of the run.

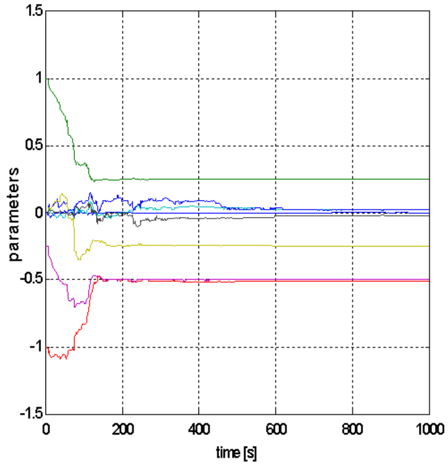

Fig. 1: Convergence of the parameter estimates for $\alpha = 2$.

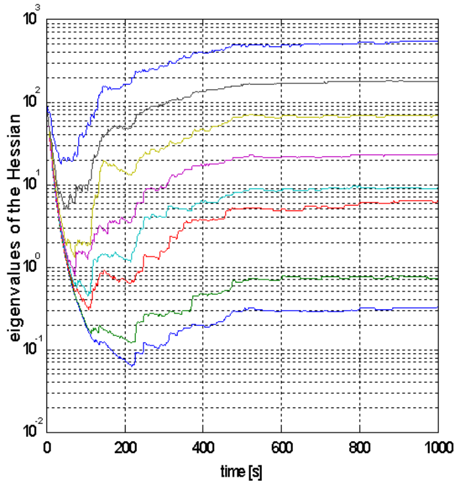

Fig. 2: Convergence of the eigenvalues of the Hessian for $\alpha = 2$.

Example 1: In order to study the result of Theorem 1, the estimated model was simulated using the parameters obtained at the end of the run. The root mean-square levels of the first and second state signals were computed as

$$\bar{x}_1^s = \sqrt{\frac{1}{10000}\sum_{t=1}^{10000} x_1^s\left(t,\hat{\boldsymbol{\theta}}(10000)\right)}$$
$$\bar{x}_2^s = \sqrt{\frac{1}{10000}\sum_{t=1}^{10000} x_2^s\left(t,\hat{\boldsymbol{\theta}}(10000)\right)}. \qquad (28)$$

The values obtained for $\alpha = 1$ were used to define a nominal relation between $\bar{x}_1^s$ and $\bar{x}_2^s$. The measured values were $\bar{x}_1^s = 0.3122$ and $\bar{x}_2^s = 0.2545$. The expected change of root mean-square signal levels, for varying values of α, can then both be predicted by Theorem 1 and measured by renewed computations similar to (28). In particular an experimentally obtained value $\bar{\alpha}$ can be computed and compared to the one that was applied, using

$$\bar{\alpha} = \left(\left(\frac{\bar{x}_1^s}{\bar{x}_2^s} \right)_{\alpha=1} \right)^{-1} \left(\frac{\bar{x}_1^s}{\bar{x}_2^s} \right)_{\alpha} \qquad (29)$$

The result appears in Table 1. The agreement between Theorem 1 and the measured results is excellent.

α	$\bar{\alpha}$	α	$\bar{\alpha}$
0.3000	0.3013	1.7500	1.7465
0.4000	0.4018	2.0000	1.9940
0.5000	0.5005	2.5000	2.5485
0.6000	0.6005	3.0000	3.0567
0.7500	0.7503	3.5000	3.5638
1.2500	1.2496	4.0000	4.0720
1.5000	1.4981	4.5000	4.5838

Table 1: Comparison between the applied and the experimentally measured scale factor.

Example 2: This example aims at verifying the result of Theorem 2. This time the expected scaled parameter vectors were computed by inversion of the result of Theorem2 as $\boldsymbol{\theta}^s = \mathbf{T}^{-1}(\alpha)\,\boldsymbol{\theta}^0$, for the values of the scale factor used in Example 1 and for the true parameter vector ($\alpha = 1$).

$$\boldsymbol{\theta}^0 = \begin{pmatrix} 0 & 1 & -1 & 0 & -2 & -1 & 0 & 0 \end{pmatrix}^T . \qquad (30)$$

The absolute values of the parameters as obtained from Theorem 2 are plotted in Fig. 3 (as lines), together with the absolute values of the parameters obtained at the end of each identification run (circles). Absolute values were used in order to be able to use a logarithmic plot. All signs were observed to be correct. The agreement between Theorem 2 and the experimental results appears to be excellent.

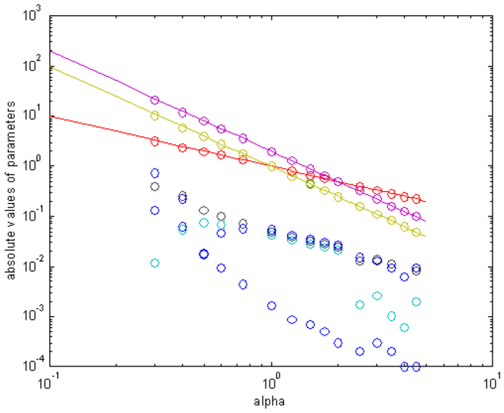

Fig. 3. Experimentally observed scaled parameter values (circles) plotted with calculated scaled parameter values according to Theorem 2 (lines). Note that only 3 lines are visible. This is because the absolute value of two of the parameters, as well as their scaling, coincide. The circles at the bottom of the plot represent true parameters that equal 0.

Example 3: This example aims at verifying the results of Theorem 3, by an evaluation of the condition number (the ratio of the largest and smallest singular values) of the Hessian. The condition number of the Hessian obtained for $\alpha = 1$ was recorded and used for prediction for other values of α according to Theorem 3. The experimentally obtained condition numbers obtained at the end of each identification run are plotted in Fig. 4 to compare to the theory.

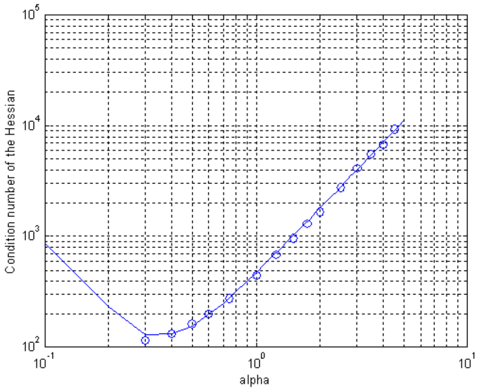

Fig. 4. The condition number of the Hessian as a function of α, obtained from Theorem 3 (lines) and from identification experiments (circles).

The agreement between Theorem 3 and the experiments is excellent. It can be observed that the scaling affects the condition number by several orders of magnitude. Further, an optimal condition number seems to exist. One practical application of Theorem 3 would be to tune the scaling for optimal condition number, given a successful initial identification run. The algorithm could then be used with the optimal scaling for tracking, or for renewed identifications under slightly different conditions. The benefit is expected to be a more well behaved algorithm.

V. CONCLUSIONS

New results on scaling for a class of nonlinear state space models have been presented. The main idea is to apply a scaled value of the sampling period in the identification algorithm. The effect of scaling was analyzed and relations between original and scaled values were obtained for the state signals, the estimated parameters and the most importantly, the Hessian. A simulation study verified all results and provided experimental support for observed improvements of algorithmic behavior that were observed during the development of the method. The software package [16], containing the algorithm and the support functions used in this paper, is available for free download from http://www.it.uu.se/research/reports/.

Interesting topics for future research include applications as well as development of more advanced methods, possible self-scaling ones.

APPENDIX A

Note that this is a short version of the proof of [11].

It follows from C1) and C2) that

$$1=\frac{x_1(t,\boldsymbol{\theta})}{x_1^s(t,\boldsymbol{\theta}^s)}=\frac{x_1(t-T_S)+T_S x_2(t-T_S,\boldsymbol{\theta})}{x_1^s(t-T_S)+T_S^{Scaled}x_2^s(t-T_S,\boldsymbol{\theta}^s)},\forall t$$

$$\Leftrightarrow$$

$$x_1^s(t-T_S)+T_S^{Scaled}x_2^s(t-T_S,\boldsymbol{\theta}^s)$$

$$=x_1(t-T_S)+T_S x_2(t-T_S,\boldsymbol{\theta}),\forall t$$

$$\Leftrightarrow$$

$$x_2(t-T_S,\boldsymbol{\theta})=\left(\frac{T_S^{Scaled}}{T_S}\right)x_2^s(t-T_S,\boldsymbol{\theta}^s),\forall t \qquad (31)$$

after a renewed application of C2) in the last step. Repeating the argumentation for the next state component, utilizing (31) gives the result for the third state component. After repeating the argumentation n times and noting that the result is valid for all values of t proves the theorem.

APPENDIX B

Note that this is a short version of the proof of [11].

Applying Theorem 1 to the last component of (18) and rearranging results in

$$\begin{aligned}&x_n(t+T_S,\boldsymbol{\theta})=x_n(t,\boldsymbol{\theta})\\&+\alpha^n\ T_S f\left(x_1^s(t,\boldsymbol{\theta}^s),\ldots,x_n^s(t,\boldsymbol{\theta}^s),u_1,\ldots,u_k^{(n_k)},\boldsymbol{\theta}^s\right),\forall t\end{aligned} \qquad (32)$$

Comparing (32) to the last component of (17) and utilizing (3), then shows that

$$\boldsymbol{\varphi}^T(\mathbf{x}(t,\boldsymbol{\theta}),\mathbf{u}(t))\boldsymbol{\theta}=\alpha^n\ \boldsymbol{\varphi}^T(\mathbf{x}^s(t,\boldsymbol{\theta}^s),\mathbf{u}(t))\boldsymbol{\theta}^s,\ \forall t. \qquad (33)$$

To proceed, one arbitrary component of the LHS and RHS versions of (33) is studied, *i.e.* the following quantities

$$\begin{aligned}&\varphi_{i_{x_1},\ldots}(\mathbf{x}(t,\boldsymbol{\theta}),\mathbf{u}(t))\\&=(x_1)^{i_{x_1}}\ldots(x_n)^{i_{x_n}}(u_1)^{i_{u_1}}\ldots\left(u_k^{(n_k)}\right)^{i_{u_k^{(n_k)}}}\\&\varphi_{i_{x_1},\ldots}^s(\mathbf{x}^s(t,\boldsymbol{\theta}^s),\mathbf{u}(t))\\&=(x_1^s)^{i_{x_1}}\ldots(x_n^s)^{i_{x_n}}(u_1)^{i_{u_1}}\ldots\left(u_k^{(n_k)}\right)^{i_{u_k^{(n_k)}}}.\end{aligned} \qquad (34)$$

By application of Theorem 1, $\varphi_{i_{x_1}}^s(\mathbf{x}^s(t,\boldsymbol{\theta}^s),\mathbf{u}(t))$ can be transformed to depend only on $x_{1,}\ldots,x_n,u_1,\ldots,u_k^{(n_k)}$. The result is

$$\begin{aligned}&\varphi_{i_{x_1},\ldots}^s(\mathbf{x}^s(t,\boldsymbol{\theta}^s),\mathbf{u}(t))\\&=\alpha^{-i_{x_2}-2i_{x_3}-\ldots-(n-1)i_{x_n}}\ \varphi_{i_{x_1},\ldots}(\mathbf{x}(t,\boldsymbol{\theta}),\mathbf{u}(t)),\ \forall t.\end{aligned} \qquad (35)$$

Inserting (35) in (33) gives

$$\begin{aligned}&\boldsymbol{\varphi}^T(\mathbf{x}(t,\boldsymbol{\theta}),\mathbf{u}(t))\boldsymbol{\theta}\\&=\boldsymbol{\varphi}^T(\mathbf{x}(t,\boldsymbol{\theta}),\mathbf{u}(t))\ diag_{i_{x_1},\ldots}\left(\alpha^{n-i_{x_2}-2i_{x_3}-\ldots-(n-1)i_{x_n}}\right)\boldsymbol{\theta}^s,\ \forall t.\end{aligned} \qquad (36)$$

A multiplication of (36) with $\boldsymbol{\varphi}(\mathbf{x}(t,\boldsymbol{\theta}),\mathbf{u}(t))$ from the left, followed by a summation over time according to C3) then results in an equation that has the result of Theorem 2 as the unique solution.

REFERENCES

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

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

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

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

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

[6] L. Guo and L. Ljung, ''Performance analysis of general tracking algorithms'', *IEEE Trans. Automat. Contr.,* vol. 40, pp. 1388-1402, 1995.

[7] S. A. Billings and S. Y. Fakhouri, ''Identification of systems containing linear dynamic and static nonlinear elements'', *Automatica,* vol. 18, pp. 15-26, 1992.

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

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

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

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

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

[13] T. Wigren, ''Recursive identification based on nonlinear state space models applied to drum-boiler dynamics with nonlinear output equations'', *in Proc. of ACC 2005,* Portland, Oregon, USA, June 8-10, 2005.

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

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

[16] 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.