Recursive Prediction Error Identification and Scaling of Non-linear Systems with Midpoint Numerical Integration

Soma Tayamon and Torbjörn Wigren*

March 19, 2010

Abstract

A new recursive prediction error algorithm (RPEM) based on a non-linear ordinary differential equation (ODE) model of black-box state space form is presented. The selected model is discretised by a midpoint integration algorithm and compared to an Euler forward algorithm. When the algorithm is applied, scaling of the sampling time is used to improve performance further. This affects the state vector, the parameter vector and the Hessian. This impact is analysed and described in three Theorems. Numerical examples are provided to verify the theoretical results obtained.

*The authors are with the division of Systems and Control, Department of Information Technology, Uppsala University, SE–75105 Uppsala, Sweden. (e-mail: soma.tayamon@it.uu.se; torbjorn.wigren@it.uu.se)

1 Introduction

The interest in techniques for non-linear system identification has increased due to its relevance in engineering applications. A few examples include solar heating systems [4], power system components [1] and pH-control systems [10]. The non-linearities of these applications present challenges in both modelling and control design. There are several modelling methods available. Physical modelling aims to describe the entire system based on its physical principles, but requires complete knowledge of the system. On the other hand gray-box modelling combines some available knowledge about the physical principles with the use of system identification. Physical modelling is applied to create a differential equation and the unknown parameters are then estimated using measured data. Hence each model is created for a specific application and therefore cannot be used in general cases. However, the complexity of the problem is reduced due to the reduction of number of unknown parameters, which also results in a higher accuracy [3].

In cases where the knowledge of the system is limited a more flexible black-box model is a useful tool. The main advantage is the wide variety of applications that can be modelled. Some of the commonly used black-box identification methods include neural networks [7], block-oriented algorithms [2] and non-linear models based on difference equations such as NARMAX [6], NARX and NFIR. For an overview of non-linear black-box modelling see [11].

In [14] and [12] a recursive prediction error method (RPEM) based on a MIMO black-box non-linear model in state space form is described. The algorithm uses a continuous time state space model with a restricted parameterisation, in that only one component of the right hand side of the ODE is used to model the function of the ODE. An Euler forward method is used for discretisation. A drawback is that the Euler method requires fast sampling to achieve

improved accuracy.

Given the above background, the purpose and first contribution of this paper is to modify the RPEM of [14] and [12] by the use of a more accurate integration algorithm. Even though the Euler method is simple and fast, it only uses a one-sided estimate of the derivative meaning that the estimation of the derivative is valid between each sample. The midpoint integration method applied here uses the point in between each sample to obtain a more correct alignment, [5]. With a more accurate derivative, a more precise parameter estimation can be expected.

The papers [14] and [13] also use scaling of the sampling period to improve the numerical properties of the RPEM based on the Euler method. A main consequence of the use of the midpoint method is that the state vector, parameter vector and the Hessian will be affected differently by scaling than in [14] and [13]. The second contribution of this paper is hence the analysis of the scaling of the sampling period for the RPEM based on the midpoint method.

This paper is organised as follows. Section II presents the model for which the RPEM is defined. In section III the analysis of the effects of the new integration method is presented. In section IV a simulation study is discussed and finally the conclusions are presented in section V.

2 The non-linear state space model and the algorithm

2.1 The model

The algorithm developed in this paper is based on a non-linear continuous time state space model. To describe the general model the input vector $\boldsymbol{u}(t)$ is

introduced

$$\boldsymbol{u}(t) = \left(u_1(t) \dots u_1^{(n_1)}(t) \dots u_k(t) \dots u_k^{(n_k)}(t)\right)^T, \tag{1}$$

together with the output vector $\boldsymbol{y}(t)$

$$\boldsymbol{y}(t) = \begin{pmatrix} y_1(t) & \dots & y_p(t) \end{pmatrix}^T. \tag{2}$$

Here, the superscript n_i denotes differentiation n_i times. The model can be described as

$$\boldsymbol{x}^{(1)} = \begin{pmatrix} x_1^{(1)} \\ \vdots \\ x_{n-1}^{(1)} \\ x_n^{(1)} \end{pmatrix} = \begin{pmatrix} x_2 \\ \vdots \\ x_n \\ f(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\theta}) \end{pmatrix} \equiv \bar{f}\left(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\theta}\right)$$

$$\begin{pmatrix} y_1 \\ \vdots \\ y_p \end{pmatrix} = \underbrace{\begin{pmatrix} c_{11} & \dots & c_{1n} \\ \vdots & \ddots & \vdots \\ c_{p1} & \dots & c_{pn} \end{pmatrix}}_{\mathbf{C}} \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}. \tag{3}$$

where $\boldsymbol{x} = \begin{pmatrix} x_1 & x_2 & \dots & x_{n-1} & x_n \end{pmatrix}^T$ is the state vector and $\boldsymbol{\theta}$ is the unknown parameter vector. As is shown in Theorem 1 in [14] the model described in (3) can be used to model also ODEs with general right-hand sides, locally in the state space. The model in (3) concentrates the non-linearity to one component of the equation, minimising the risk for overparametrisation. The

right-hand side function, $f(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\theta})$, is chosen to be a polynomial

$$
\begin{aligned}
&f(x_1, \ldots, x_n, u_1, \ldots, u_1^{(n_1)}, \ldots, u_k, \ldots, u_k^{(n_k)}, \boldsymbol{\theta}) \\
&= \sum_{i_{x_1}=0}^{I_{x_1}} \cdots \sum_{i_{x_n}=0}^{I_{x_n}} \sum_{i_{u_1}=0}^{I_{u_1}} \cdots \sum_{i_{u_1}{}^{(n_1)}=0}^{I_{u_1}{}^{(n_1)}} \cdots \sum_{i_{u_k}=0}^{I_{u_k}} \cdots \sum_{i_{u_k}{}^{(n_k)}=0}^{I_{u_k}{}^{(n_k)}} \\
&\boldsymbol{\theta}_{i_{x_1} \ldots i_{x_n} i_{u_1} \ldots i_{u_1{}^{(u_1)}} \ldots i_{u_k} \ldots i_{u_k{}^{(u_k)}}} \left(x_1\right)^{i_{x_1}} \ldots \left(x_n\right)^{i_{x_n}} \\
&\left(u_1\right)^{i_{u_1}} \ldots \left(u_1{}^{(n_1)}\right)^{i_{u_1}{}^{(n_1)}} \ldots \left(u_k\right)^{i_{u_k}} \ldots \left(u_k{}^{(n_k)}\right)^{i_{u_k}{}^{(n_k)}}.
\end{aligned}
$$

This polynomial can be written as a regressor vector generated from the state vector and the input vector multiplied by an unknown parameter vector as follows

$$
f(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\theta}) = \boldsymbol{\varphi}^T \left(\boldsymbol{x}(t, \boldsymbol{\theta}), \boldsymbol{u}(t)\right) \boldsymbol{\theta}. \tag{4}
$$

The details of $\boldsymbol{\theta}$ and $\boldsymbol{\varphi}(\boldsymbol{x}\left(t, \boldsymbol{\theta}\right), \boldsymbol{u}(t))$ appear in (5) and (6)

$$
\begin{aligned}
\boldsymbol{\theta} = \Big(\; & \theta_{0\ldots0} \quad \ldots \quad \theta_{0\ldots I_{u_k}{}^{(n_k)}} \quad \theta_{0\ldots010} \quad \ldots \\
& \theta_{0\ldots01 I_{u_k}{}^{(n_k)}} \quad \ldots \quad \theta_{0\ldots0 I_{u_k}{}^{(n_k-1)}} \quad \ldots \\
& \theta_{0\ldots0 I_{u_k}{}^{(n_k-1)} I_{u_k}{}^{(n_k)}} \quad \ldots \quad \theta_{I_{x_1}\ldots I_{u_k}{}^{(n_k)}} \; \Big)^T.
\end{aligned} \tag{5}
$$

$$
\begin{aligned}
\boldsymbol{\varphi} = \Big(\; & 1 \quad \ldots \quad \left(\left(u_k{}^{(n_k)}\right)^{I_{u_k}{}^{(n_k)}}\right) \quad u_k{}^{(n_k-1)} \quad \ldots \\
& \left(u_k{}^{(n_k-1)} \left(u_k{}^{(n_k)}\right)^{I_{u_k}{}^{(n_k)}}\right) \quad \ldots \quad \left(u_k{}^{(n_k-1)}\right)^{I_{u_k}{}^{(n_k-1)}} \\
& \ldots \quad \left(\left(u_k{}^{(n_k-1)}\right)^{I_{u_k}{}^{(n_k-1)}} \left(u_k{}^{(n_k)}\right)^{I_{u_k}{}^{(n_k)}}\right) \quad \ldots \\
& \left(\left(x_1\right)^{I_{x_1}} \ldots \left(x_n\right)^{I_{x_n}} \left(u_1\right)^{I_{u_1}} \ldots \left(u_k^{(n_k)}\right)^{I_{u_k}{}^{(n_k)}}\right)\Big)^T.
\end{aligned} \tag{6}
$$

Here, I_{x_i} and $I_{u_j^{(n_j)}}$ denote the polynomial degree of each variable. Since output error identification is used the state vector $\boldsymbol{x}$ needs to be estimated.

2.2 Numerical integration, gradient and algorithm

The algorithm is based on the minimisation of the prediction errors, $\boldsymbol{\varepsilon}(t, \boldsymbol{\theta}) = \boldsymbol{y}_m(t, \boldsymbol{\theta}) - \boldsymbol{y}(t, \boldsymbol{\theta})$, using the criterion

$$V(\boldsymbol{\theta}) = \frac{1}{2} E\left[\boldsymbol{\varepsilon}^T(t, \boldsymbol{\theta}) \boldsymbol{\Lambda}^{-1}(t, \boldsymbol{\theta}) \boldsymbol{\varepsilon}\left(t, \boldsymbol{\theta}\right) + \det\left(\boldsymbol{\Lambda}\left(t, \boldsymbol{\theta}\right)\right)\right], \tag{7}$$

where $\boldsymbol{y}_m(t)$ is the measured output and where $\boldsymbol{\Lambda}(t, \boldsymbol{\theta})$ is the unknown covariance matrix.

In order to continue, the continuous time model in (3) needs to be discretised. In this paper the midpoint integration method, also known as the second order Runge-Kutta method is applied as dicretisation algorithm [5]. This integration method is a refinement of the Euler method. Contrary to the Euler method, the midpoint method is not linear in the sampling period. On the other hand, the midpoint method is not as sensitive to the sampling period and does not require the same fast sampling that is necessary for the original Euler based RPEM. This way a possibility to use the RPEM for wider applications and with a better accuracy is created. The discretised model, using the sampling period T and

the midpoint integration scheme, can be represented as

$$\begin{pmatrix} x_1(t+T,\boldsymbol{\theta}) \\ \vdots \\ x_{n-1}(t+T,\boldsymbol{\theta}) \\ x_n(t+T,\boldsymbol{\theta}) \end{pmatrix} = \begin{pmatrix} x_1(t,\boldsymbol{\theta}) \\ \vdots \\ x_{n-1}(t,\boldsymbol{\theta}) \\ x_n(t,\boldsymbol{\theta}) \end{pmatrix}$$
$$+ T \begin{pmatrix} x_2(t,\boldsymbol{\theta}) + \frac{T}{2} x_2(t,\boldsymbol{\theta}) \\ \vdots \\ x_n(t,\boldsymbol{\theta}) + \frac{T}{2} x_n(t,\boldsymbol{\theta}) \\ f\left(\boldsymbol{x}(t,\boldsymbol{\theta}) + \frac{T}{2}\bar{f}\left(x(t,\boldsymbol{\theta}), \boldsymbol{u}(t), \boldsymbol{\theta}\right), \boldsymbol{u}(t+\frac{T}{2}), \boldsymbol{\theta}\right) \end{pmatrix}$$
$$\boldsymbol{y}(t) = \mathbf{C}\boldsymbol{x}(t,\boldsymbol{\theta}). \tag{8}$$

where $\bar{f}(.)$ is defined by (3).

The gradient $\boldsymbol{\psi}(t) = \mathbf{C}\frac{d\boldsymbol{x}(t,\boldsymbol{\theta})}{d\boldsymbol{\theta}}$ plays an important role in the development of the RPEM. It is constructed from the derivative of the state, the latter being given by the equations

$$\frac{d\boldsymbol{x}(t+T,\boldsymbol{\theta})}{d\boldsymbol{\theta}} = \frac{d\boldsymbol{x}(t,\boldsymbol{\theta})}{d\boldsymbol{\theta}} + T \begin{pmatrix} \left(1+\frac{T}{2}\right)\frac{dx_2(t+T,\boldsymbol{\theta})}{d\boldsymbol{\theta}} \\ \vdots \\ \left(1+\frac{T}{2}\right)\frac{dx_n(t+T,\boldsymbol{\theta})}{d\boldsymbol{\theta}} \\ \frac{df(\bar{\boldsymbol{x}},\boldsymbol{u},\boldsymbol{\theta})}{d\boldsymbol{\theta}} \end{pmatrix}$$
$$\boldsymbol{\psi}(t+T) = \mathbf{C}\frac{d\boldsymbol{x}(t+T,\boldsymbol{\theta})}{d\boldsymbol{\theta}}. \tag{9}$$

where

$$\bar{\boldsymbol{x}} = \boldsymbol{x}(t,\boldsymbol{\theta}) + \frac{T}{2}\bar{f}(\boldsymbol{x},\boldsymbol{u},\boldsymbol{\theta}) \tag{10}$$
$$\frac{df(\bar{\boldsymbol{x}},\boldsymbol{u},\boldsymbol{\theta})}{d\boldsymbol{\theta}} = \frac{\partial f}{\partial \bar{\boldsymbol{x}}}\Big|_{t+\frac{T}{2}} \frac{d\bar{\boldsymbol{x}}}{d\boldsymbol{\theta}} + \frac{\partial f}{\partial \boldsymbol{\theta}}\Big|_{t+\frac{T}{2}} \tag{11}$$

$$
\begin{aligned}
\frac{d\bar{\boldsymbol{x}}}{d\boldsymbol{\theta}} &= \frac{d}{d\boldsymbol{\theta}}\left(\boldsymbol{x}(t,\boldsymbol{\theta}) + \frac{T}{2}\bar{f}(\boldsymbol{x},\boldsymbol{u},\boldsymbol{\theta})\right) \\
&= \frac{d\boldsymbol{x}(t,\boldsymbol{\theta})}{d\boldsymbol{\theta}} + \frac{\partial f}{\partial \boldsymbol{\theta}}|_t \\
&+ \frac{T}{2}\begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & 0 & \dots & 0 & 1 \\ & \frac{\partial f}{\partial \boldsymbol{x}}|_t & & & \end{pmatrix} \frac{d\boldsymbol{x}(t,\boldsymbol{\theta})}{d\boldsymbol{\theta}}.
\end{aligned} \tag{12}
$$

Using the notation in (4) as a substitution of the function $f(\boldsymbol{x},\boldsymbol{u},\boldsymbol{\theta})$, it follows that

$$
\frac{\partial f}{\partial \boldsymbol{\theta}} = \boldsymbol{\varphi}(\boldsymbol{x}(t,\boldsymbol{\theta}),\boldsymbol{u}(t)) \tag{13}
$$

$$
\frac{\partial f}{\partial \boldsymbol{x}} = \boldsymbol{\theta}^T \frac{d\boldsymbol{\varphi}(\boldsymbol{x}(t,\boldsymbol{\theta}))}{d\boldsymbol{x}} \frac{d\boldsymbol{x}(t,\boldsymbol{\theta})}{d\boldsymbol{\theta}}. \tag{14}
$$

The derivative of the state vector can then be expressed as a function of the regressor and the parameter vector solely. Inserting (13) and (14) into equation (9) using (10) and (12) generates a description of the derivative of the state vector. See [15] for more details.

The development of the RPEM follows the standard approach of [9] and is described in detail in [14]. The gradient of the output prediction $\boldsymbol{\psi}(t,\boldsymbol{\theta})$ is used in the updating of the unknown parameters. The RPEM is formulated using the Gauss Newton minimisation algorithm of [9], as

$$
\begin{aligned}
\boldsymbol{\Lambda}(t) &= \boldsymbol{\Lambda}(t-T) + \frac{\mu(t)}{t}\left(\boldsymbol{\varepsilon}(t)\boldsymbol{\varepsilon}^T(t) - \boldsymbol{\Lambda}(t-T)\right), \\
\mathbf{R}(t) &= \mathbf{R}(t-T) + \frac{\mu(t)}{t}(t)\left(\boldsymbol{\psi}(t)\boldsymbol{\Lambda}^{-1}(t)\boldsymbol{\psi}^T(t) - \mathbf{R}(t-T)\right), \\
\hat{\boldsymbol{\theta}}(t) &= \left[\hat{\boldsymbol{\theta}}(t-T) + \frac{\mu(t)}{t}\mathbf{R}^{-1}(t)\psi(t)\boldsymbol{\Lambda}^{-1}(t)\boldsymbol{\varepsilon}(t)\right] D_{\mathcal{M}}.
\end{aligned} \tag{15}
$$

where $\frac{\mu(t)}{t}$ is the gain sequence and $\mathbf{R}(t)$ is the estimate of the Hessian. $D_{\mathcal{M}}$ is the model set defining the allowed values of the estimated parameters. To ensure stability, the model set is restricting the linearised model from becoming unstable. The updating is stopped if the parameter values are outside the model set. The resulting algorithm parallels (21) of [14]. The difference is limited to the use of (8) and (14) instead of the corresponding quantities of the Euler algorithm.

3 Analysis

To analyse the effect of the scaling of the sampling period on the new discretisation algorithm two models are defined, the original midpoint model and the scaled midpoint model. A new state vector $\boldsymbol{x}^s\left(t, \boldsymbol{\theta}^s\right)$ is defined for the scaled model with the corresponding parameter vector $\boldsymbol{\theta}^s$, where the superscript s denotes the scaled quantity. The scaled model is

$$\begin{pmatrix} x_1^s\left(t+T, \boldsymbol{\theta}^s\right) \\ \vdots \\ x_{n-1}^s\left(t+T, \boldsymbol{\theta}^s\right) \\ x_n^s\left(t+T, \boldsymbol{\theta}^s\right) \end{pmatrix} = \begin{pmatrix} x_1^s\left(t, \boldsymbol{\theta}^s\right) \\ \vdots \\ x_{n-1}^s\left(t, \boldsymbol{\theta}^s\right) \\ x_n^s\left(t, \boldsymbol{\theta}^s\right) \end{pmatrix} + T^s \begin{pmatrix} x_2^s\left(t, \boldsymbol{\theta}^s\right) + \frac{T^s}{2} x_2^s\left(t, \boldsymbol{\theta}^s\right) \\ \vdots \\ x_n^s\left(t, \boldsymbol{\theta}^s\right) + \frac{T^s}{2} x_n^s\left(t, \boldsymbol{\theta}^s\right) \\ \boldsymbol{\varphi}^T\left(\bar{\boldsymbol{x}}^s, \boldsymbol{u}\left(t+\frac{T}{2}\right)\right) \boldsymbol{\theta}^s \end{pmatrix}. \tag{16}$$

The model (16) is to be compared with the original model

$$\begin{pmatrix} x_1(t+T,\boldsymbol{\theta}) \\ \vdots \\ x_{n-1}(t+T,\boldsymbol{\theta}) \\ x_n(t+T,\boldsymbol{\theta}) \end{pmatrix} = \begin{pmatrix} x_1(t,\boldsymbol{\theta}) \\ \vdots \\ x_{n-1}(t,\boldsymbol{\theta}) \\ x_n(t,\boldsymbol{\theta}) \end{pmatrix} + T \begin{pmatrix} x_2(t,\boldsymbol{\theta}) + \frac{T}{2} x_2(t,\boldsymbol{\theta}) \\ \vdots \\ x_n(t,\boldsymbol{\theta}) + \frac{T}{2} x_n(t,\boldsymbol{\theta}) \\ \boldsymbol{\varphi}^T\left(\bar{\boldsymbol{x}}, \boldsymbol{u}\left(t+\frac{T}{2}\right)\right)\boldsymbol{\theta} \end{pmatrix}. \tag{17}$$

Here $\bar{\boldsymbol{x}}^s$, $\bar{\boldsymbol{x}}$ and the scaled sampling period T^s are defined as

$$\bar{\boldsymbol{x}}^s = \boldsymbol{x}^s(t,\boldsymbol{\theta}^s) + \frac{T^s}{2}\left(\boldsymbol{\varphi}^T\left(\boldsymbol{x}^s(t,\boldsymbol{\theta}^s), \boldsymbol{u}(t)\right)\boldsymbol{\theta}^s\right) \tag{18}$$

$$\bar{\boldsymbol{x}} = \boldsymbol{x}(t,\boldsymbol{\theta}) + \frac{T}{2}\left(\boldsymbol{\varphi}^T\left(\boldsymbol{x}(t,\boldsymbol{\theta}), \boldsymbol{u}(t)\right)\boldsymbol{\theta}\right) \tag{19}$$

$$T^s = \alpha T. \tag{20}$$

Remark 1: As discussed in [14] and [13], the idea is to apply the algorithm with a scaled value of the sampling period. The effects include improved numerical properties as shown in [13], and the present paper.

3.1 States

In order to analyse the scaling of the states, the assumptions (C1) and (C2) are introduced, for a detailed motivation of these assumptions, see [14].

(C1) The measured output $\boldsymbol{y}_m(t)$ is assumed to be equal to the states $x_1(t,\boldsymbol{\theta})$ and $x_1^s(t,\boldsymbol{\theta}^s)$.

(C2) The algorithm converges to an exact description of the input-output

properties of the system for (16) and (17). Hence

$$\boldsymbol{y}_m(t) = x_1(t, \boldsymbol{\theta}) = x_1^s(t, \boldsymbol{\theta}^s). \tag{21}$$

$$\mathbf{C} = \begin{pmatrix} 1 & 0 & \dots & 0 \end{pmatrix}. \tag{22}$$

From the equations (16), (17) and (20), (C1) and (C2) it directly follows that

$$x_2(t, \boldsymbol{\theta}) = \alpha \left(\frac{1 + \frac{\alpha T}{2}}{1 + \frac{T}{2}} \right) x_2^s(t, \boldsymbol{\theta}^s) \equiv \alpha\gamma x_2^s(t, \boldsymbol{\theta}^s). \tag{23}$$

Equation (23) implies that the relation between $x_2(t, \boldsymbol{\theta})$ and $x_2^s(t, \boldsymbol{\theta}^s)$ is dependent on the sampling period T and the scaling factor α. This relation can be extended up to the n^{th} state, repeating the argumentation above, the result is

Theorem 1 Consider the two models (16) and (17) and assume that (C1) and (C2) holds. It then follows that

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

$$\mathbf{A}(\alpha, T) = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & \alpha\gamma & 0 & \vdots \\ \vdots & 0 & \ddots & 0 \\ 0 & \dots & 0 & \alpha^{n-1}\gamma^{n-1} \end{pmatrix},$$

Remark 2: The scaled updated state vector discretised by the Euler integration method, see [14], differs from the scaled state vector obtained with the midpoint method. This result implies that the resulting parameters are also affected by this change in the algorithm.

3.2 Parameters

To perform an analysis of the effect on the parameters, the last components of (16) and (17) are compared. Let

$$\bar{\boldsymbol{\varphi}}\left(\boldsymbol{x}(t,\boldsymbol{\theta}),\boldsymbol{u},\boldsymbol{\theta}\right) \equiv \begin{pmatrix} x_2(t,\boldsymbol{\theta}) \\ \vdots \\ x_n(t,\boldsymbol{\theta}) \\ \boldsymbol{\varphi}^T\left(\boldsymbol{x}(t,\boldsymbol{\theta}),\boldsymbol{u}(t)\right)\boldsymbol{\theta} \end{pmatrix}. \tag{24}$$

It then follows from (18) and (19) that

$$\begin{aligned} &\frac{1}{\alpha^n\gamma^{n-1}}\boldsymbol{\varphi}^T\left(\boldsymbol{x}(t,\boldsymbol{\theta}) + \frac{T}{2}\bar{\boldsymbol{\varphi}}\left(\boldsymbol{x}(t,\boldsymbol{\theta}),\boldsymbol{u}(t)\right)\boldsymbol{\theta}\right)\boldsymbol{\theta} \\ =&\boldsymbol{\varphi}^T\left(\boldsymbol{x}^s(t,\boldsymbol{\theta}^s) + \frac{T^s}{2}\bar{\boldsymbol{\varphi}}\left(\boldsymbol{x}^s(t,\boldsymbol{\theta}^s),\boldsymbol{u}(t)\right)\boldsymbol{\theta}^s\right)\boldsymbol{\theta}^s. \end{aligned} \tag{25}$$

Inserting the result of Theorem 1 into (25) then leads to the final non-trivial equation, relating $\boldsymbol{\theta}$ and $\boldsymbol{\theta}^s$.

$$\begin{aligned} &\boldsymbol{\varphi}\left(\boldsymbol{x}(t,\boldsymbol{\theta}) + \frac{T}{2}\bar{\boldsymbol{\varphi}}^T\left(\boldsymbol{x}(t,\boldsymbol{\theta}),\boldsymbol{u}(t)\right)\boldsymbol{\theta}\right)\boldsymbol{\theta} = \\ &\alpha^n\gamma^{n-1}\boldsymbol{\varphi}\left(\mathbf{A}^{-1}(\alpha)\boldsymbol{x}(t,\boldsymbol{\theta})\right. \\ &\left.+\frac{\alpha T}{2}\begin{pmatrix} \alpha^{-1}\gamma^{-1}x_2(t,\boldsymbol{\theta}) \\ \vdots \\ \alpha^{-(n-1)}\gamma^{-(n-1)}x_n(t,\boldsymbol{\theta}) \\ \boldsymbol{\varphi}^T\left(\mathbf{A}^{-1}(\alpha)\boldsymbol{x}(t,\boldsymbol{\theta}),\boldsymbol{u}(t)\right)\boldsymbol{\theta}^s \end{pmatrix}\right)\boldsymbol{\theta}^s \end{aligned} \tag{26}$$

To find the relation between $\boldsymbol{\theta}$ and $\boldsymbol{\theta^s}$, each component of $\boldsymbol{\varphi}\left(\boldsymbol{x}^s\left(t,\boldsymbol{\theta}^s\right),\boldsymbol{u}\right)$ must be compared to the corresponding component in $\boldsymbol{\varphi}\left(\boldsymbol{x}\left(t,\boldsymbol{\theta}\right),\boldsymbol{u}\right)$. Contrary to [14] and [13], this results in more equations than the number of parameters. Additional terms of high degree appear and since the integration method is

only an approximation, it results in an inconsistent set of equations.

Theorem 2 Consider the two models (16) and (17). Provided that (C1) and (C2) hold, $\boldsymbol{\theta}$ can be estimated from the scaled parameter vector $\boldsymbol{\theta}^s$ using

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

where $g(\alpha, T, \boldsymbol{\theta}^s)$ denotes the least square solution obtained from (26).

This solution is obtained using the minimisation criterion, [8]

$$\phi(\boldsymbol{\theta}) = \frac{1}{2}\boldsymbol{r}(\boldsymbol{\theta})^T\boldsymbol{r}(\boldsymbol{\theta}) \tag{27}$$

where $r_i(\boldsymbol{\theta})$ is defined as

$$r_i(\boldsymbol{\theta}) = h_i^s(\boldsymbol{\theta}^s) - h_i(\boldsymbol{\theta}), \quad i = 1, \ldots, m,$$

and $h_i(.)$ and $h_i^s(.)$ are specific functions that follow from (26), cf. Example 3 below. Since the scaled parameters are given, $\boldsymbol{h}^s(\boldsymbol{\theta}^s)$ is known. The solution can then be obtained using a non-linear search algorithm applied to (27).

3.3 Hessian

To analyse the effect of the new discretisation algorithm on the Hessian of the identification algorithm, the steps of the method proposed in [13] is followed. First the state vector $\boldsymbol{x}(t, \boldsymbol{\theta}^s)$ is differentiated with respect to the parameter vector $\boldsymbol{\theta}^s$. Using Theorem 1, it follows that

$$\begin{aligned}\frac{d\boldsymbol{x}^s(t,\boldsymbol{\theta}^s)}{d\boldsymbol{\theta}^s} &= \mathbf{A}^{-1}(\alpha)\frac{d\boldsymbol{x}(t,\boldsymbol{\theta})}{d\boldsymbol{\theta}^s} \\ &= \mathbf{A}^{-1}(\alpha)\frac{d\boldsymbol{x}(t,\boldsymbol{\theta})}{d\boldsymbol{\theta}}\frac{d\boldsymbol{\theta}}{d\boldsymbol{\theta}^s} \\ &= \mathbf{A}^{-1}(\alpha)\frac{d\boldsymbol{x}(t,\boldsymbol{\theta})}{d\boldsymbol{\theta}}\frac{dg(\alpha,T,\boldsymbol{\theta}^s)}{d\boldsymbol{\theta}^s}. \end{aligned} \tag{28}$$

Introduce the assumption

(C3) Assume that the identification algorithm is based on the simplified criterion

$$V(\boldsymbol{\theta}) = \frac{1}{2}E\left[\varepsilon^T(t,\boldsymbol{\theta})\varepsilon(t,\boldsymbol{\theta})\right].$$

The Hessian of the identification algorithm is defined as the second derivative of the minimisation criterion and with the simplification of (C3) it can be calculated as follows

$$\begin{aligned}\mathbf{R}(\boldsymbol{\theta}) = \frac{d^2}{d\boldsymbol{\theta}\ d\boldsymbol{\theta}^T}V(\boldsymbol{\theta}) &= E\left[\boldsymbol{\psi}(t,\boldsymbol{\theta})\boldsymbol{\psi}^T(t,\boldsymbol{\theta})\right] \\ &+E\left[\left(\frac{d^2}{d\boldsymbol{\theta}\ d\boldsymbol{\theta}^T}\boldsymbol{\varepsilon}(t,\boldsymbol{\theta})\right)\boldsymbol{\varepsilon}(t,\boldsymbol{\theta})\right]. \end{aligned} \tag{29}$$

Provided that the system is in the model set and assuming that the assumption (C4) below holds, the last term of (29) can be neglected.

(C4) The measurement noise $e(t)$ is zero mean, and the regressor vector $\boldsymbol{\varphi}$ is generated only from the input signal.

This leads to the following simplification of the Hessian

$$\mathbf{R}(\boldsymbol{\theta}) = E\left[\boldsymbol{\psi}(t,\boldsymbol{\theta})\boldsymbol{\psi}^T(t,\boldsymbol{\theta})\right].$$

The effect of the scaling can now be analysed by using the gradient of the

measurement. The scaled Hessian then becomes

$$\mathbf{R}(\boldsymbol{\theta}^s) = E\left[\left(\frac{d\boldsymbol{x}^s(t,\boldsymbol{\theta}^s)}{d\boldsymbol{\theta}^s}\right)^T \mathbf{C}^T\mathbf{C}\frac{d\boldsymbol{x}^s(t,\boldsymbol{\theta}^s)}{d\boldsymbol{\theta}^s}\right] \tag{30}$$

and together with (28) and Theorem 1 it can be concluded that

$$\mathbf{R}(\boldsymbol{\theta}^s) = E\left[\left(\frac{dg(\alpha,T,\boldsymbol{\theta}^s)}{d\boldsymbol{\theta}^s}\right)^T \left(\frac{d\boldsymbol{x}(t,\boldsymbol{\theta})}{d\boldsymbol{\theta}}\right)^T \underbrace{\mathbf{A}^{-T}\mathbf{C}^T\mathbf{C}\mathbf{A}^{-1}}_{\mathbf{I}} \times \frac{d\boldsymbol{x}(t,\boldsymbol{\theta})}{d\boldsymbol{\theta}}\frac{dg(\alpha,T,\boldsymbol{\theta}^s)}{d\boldsymbol{\theta}^s}\right]$$

where the last step follows from the fact that the $(1,1)$ element of $\mathbf{A}^{-1} = 1$ and that $\mathbf{C}$ is given by (22). This proves

Theorem 3 Consider the two models (16) and (17) where T^s is the scaled sampling period and assume that the conditions (C1)-(C4) apply. The Hessian is then given by

$$\mathbf{R}(\boldsymbol{\theta}^s) = \left(\frac{dg(\alpha,T,\boldsymbol{\theta}^s)}{d\boldsymbol{\theta}^s}\right)^T \mathbf{R}(\boldsymbol{\theta})\frac{dg(\alpha,T,\boldsymbol{\theta}^s)}{d\boldsymbol{\theta}^s}.$$

where $g(\alpha,T,\boldsymbol{\theta}^s)$ is defined by Theorem 2.

Note that also for Theorem 3, numerical solution is required to relate the scaled Hessian to the unscaled one.

4 Simulated results

To study the results obtained in section III, a simulation study based on a system described in [14] was performed. The system is given by a second order

state space model.

$$\dot{\boldsymbol{x}}(t) = \begin{pmatrix} -x_1(t) - x_2(t) \\ x_1(t)\,(2 + u(t)) - u(t) \end{pmatrix}$$

$$\boldsymbol{y}(t) = \begin{pmatrix} 1 & 0 \end{pmatrix} \boldsymbol{x}(t) + e(t). \tag{31}$$

This model can easily be rewritten to correspond to the model structure (3) using following the calculations

$$x_2 = x_1(t) - \dot{x}_1(t),$$
$$\dot{x_2}(t) = \dot{x}_1(t) - \ddot{x}_1(t)$$

if $\dot{x}_1(t) = x_2(t)$, then the system can be written as

$$\dot{\boldsymbol{x}}(t) = \begin{pmatrix} x_2 \\ u(t) - (2 + u(t))\,x_1(t) - x_2(t) \end{pmatrix}$$

$$\boldsymbol{y}(t) = \begin{pmatrix} 1 & 0 \end{pmatrix} \boldsymbol{x}(t) + e(t). \tag{32}$$

For the simulation two different sets of data were generated, one generated using the unscaled midpoint algorithm, i.e. $\alpha = 1$, and the other generated by the scaled midpoint method. In both cases, the data length was 10000 samples with a sampling period of 0.1 seconds, i.e. $T = 0.1$. The input signal was chosen as a uniform PRBS-like signal with zero mean in the range of $[-1, 1]$, cf. [13].

Example 1 Identification was then perfomed with the corresponding algorithm. The algorithm was initialised with the following parameter vector

$$\hat{\boldsymbol{\theta}}^s(0) = \begin{pmatrix} 0 & 1 & -1 & 0 & -1 & 0 & 0 & 0 \end{pmatrix}^T$$

and $\mathbf{\Lambda}(0) = 0.1$, $\mathbf{R}(0) = 100\mathbf{I}$, furthermore the state vector was $\boldsymbol{x}^s(0) = \begin{pmatrix} 0.5 & -1 \end{pmatrix}^T$. In the first run, $\alpha = 2$ was chosen. The convergence of the parameters using the new discretisation algorithm is shown in Fig 1. In Fig 2, the convergence of the eigenvalues of the Hessian is depicted. As compared to the result in [14], the transient is improved.

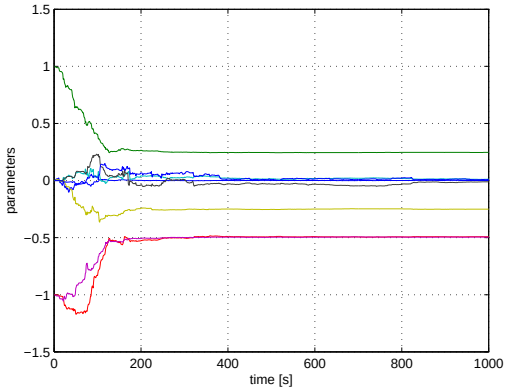

Figure 1: The convergence of the scaled parameters with the new algorithm using $\alpha = 2$.

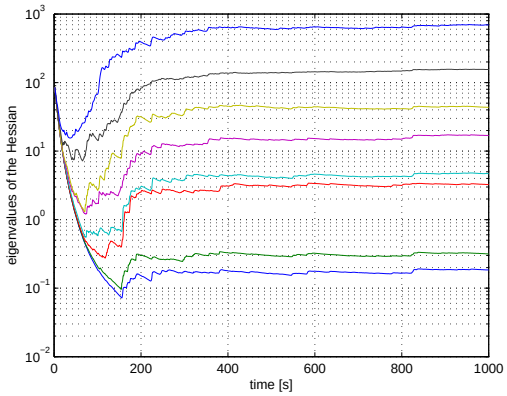

Figure 2: The convergence of the eigenvalues of the Hessian with the new discretisation algorithm using $\alpha = 2$.

Example 2 This example was performed to verify the results of Theorem 1. Since the state space model is of second order ($n = 2$), the matrix $\mathbf{A}(\alpha, T)$ as described in Theorem 1 is a 2×2 matrix represented as

$$\mathbf{A}(\alpha, T) = \begin{pmatrix} 1 & 0 \\ 0 & \alpha\gamma \end{pmatrix} \tag{33}$$

To analyse the accuracy of Theorem 1, the average root-mean square values of the states were used.

First, the state vector for the midpoint method using $\alpha = 1$ was estimated and stored. This was then repeated for the scaled midpoint method for different values of α. Finally, the value of α was calculated and compared with the real applied α. The results obtained were then normalised with respect to the first state of both methods. As seen in Fig. 3 the result of Theorem 1 agrees well with the experimental results.

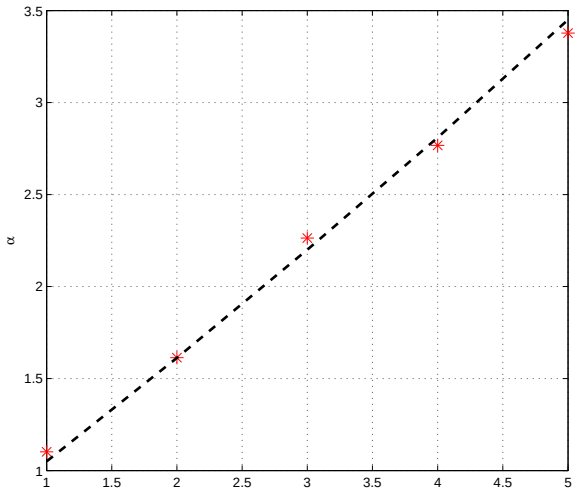

Figure 3: Experimentally calculated value of α using Theorem 1, (stars) and the real applied α, (dashed).

Example 3 This example aims to verify the results of Theorem 2. In this example, α was chosen in a range $1-4$. The parameter values were then calculated numerically as described in section III. The model described by equation (32) gave rise to the following 24 non-linear equations, from (26), describing the relation between $\boldsymbol{\theta}^s$ and $\boldsymbol{\theta}$.

$$
\begin{aligned}
h_1(\boldsymbol{\theta}) &= \theta_{000} + \frac{T}{2}\theta_{000}\theta_{010} \\
&= \alpha^2\gamma\left(\theta^s_{000} + \frac{\alpha T}{2}\theta^s_{000}\theta^s_{010}\right) = h^s_1(\boldsymbol{\theta}^s) \\
h_2(\boldsymbol{\theta}) &= \theta_{001} + \frac{T}{2}\left(\theta_{001}\theta_{010} + \theta_{000}\theta_{011}\right) \\
&= \alpha^2\gamma\left(\theta^s_{001} + \frac{\alpha T}{2}\left(\theta^s_{001}\theta^s_{010} + \theta^s_{000}\theta^s_{011}\right)\right) = h^s_2(\boldsymbol{\theta}^s) \qquad (34) \\
h_3(\boldsymbol{\theta}) &= \theta_{010} + \frac{T}{2}\left(\theta_{010}\theta_{010} + \theta_{100}\right) + \frac{T}{4}\theta_{110}\theta_{000} \\
&= \alpha\left(\theta^s_{010} + \frac{\alpha T}{2}\left(\theta^s_{010}\theta^s_{010} + \theta_{100^s}\right) + \frac{(\alpha T)}{4}\theta_{110^s}\theta^s_{000}\right) = h^s_3(\boldsymbol{\theta}^s) \\
h_4(\boldsymbol{\theta}) &= \theta_{011} + \frac{T}{2}\left(2\theta_{011}\theta_{010} + \theta_{101}\right) + \frac{T}{4}\left(\theta_{110}\theta_{001} + \theta_{111}\theta_{000}\right) \\
&= \alpha\left(\theta^s_{011} + \frac{\alpha T}{2}\left(2\theta^s_{011}\theta^s_{010} + \theta^s_{101}\right) + \frac{(\alpha T)}{4}\left(\theta^s_{110}\theta^s_{001} + \theta^s_{111}\theta^s_{000}\right)\right) = h^s_4(\boldsymbol{\theta}^s) \\
h_5(\boldsymbol{\theta}) &= \theta_{100} + \frac{T}{2}\left(\theta_{100}\theta_{010} + \theta_{110}\theta_{000}\right) \\
&= \alpha^2\gamma\left(\theta^s_{100} + \frac{\alpha T}{2}\left(\theta^s_{100}\theta^s_{010} + \theta^s_{110}\theta^s_{000}\right)\right) = h^s_5(\boldsymbol{\theta}^s)
\end{aligned}
$$

$$
\begin{aligned}
h_6(\boldsymbol{\theta}) &= \theta_{101} + \frac{T}{2}\left(\theta_{101}\theta_{010} + \theta_{101}\theta_{011} + \theta_{110}\theta_{001} + \theta_{111}\theta_{000}\right) \\
&= \alpha^2\gamma\left(\theta^s_{101} + \frac{\alpha T}{2}\left(\theta^s_{101}\theta^s_{010} + \theta^s_{101}\theta^s_{011} + \theta^s_{110}\theta^s_{001} + \theta^s_{111}\theta^s_{000}\right)\right) = h^s_6(\boldsymbol{\theta}^s) \\
h_7(\boldsymbol{\theta}) &= \theta_{110} + T\theta_{110}\theta_{010} + \frac{T^2}{4}\theta_{110}\theta_{100} \\
&= \alpha\left(\theta^s_{110} + \alpha T\theta^s_{110}\theta^s_{010} + \frac{(\alpha T)^2}{4}\theta^s_{110}\theta^s_{100}\right) = h^s_7(\boldsymbol{\theta}^s) \\
h_8(\boldsymbol{\theta}) &= \theta_{111} + T\left(\theta_{111}\theta_{010} + \theta_{110}\theta_{011}\right) + \frac{T^2}{4}\left(\theta_{110}\theta_{101} + \theta_{111}\theta_{100}\right) \\
&= \alpha\left(\theta^s_{111} + \alpha T\left(\theta^s_{111}\theta^s_{010} + \theta^s_{110}\theta^s_{011}\right) + \frac{(\alpha T)^2}{4}\left(\theta^s_{110}\theta^s_{101} + \theta^s_{111}\theta^s_{100}\right)\right) \\
h_9(\boldsymbol{\theta}) &= \frac{T}{2}\theta_{001}\theta_{011} = \alpha^2\gamma\frac{\alpha T}{2}\theta^s_{001}\theta^s_{011} = h^s_9(\boldsymbol{\theta}^s) \\
h_{10}(\boldsymbol{\theta}) &= \frac{T}{2}\theta_{011}\theta_{011} + \frac{T^2}{4}\theta_{111}\theta_{001} = \alpha\left(\frac{\alpha T}{2}\theta^s_{011}\theta^s_{011} + \frac{(\alpha T)^2}{4}\theta^s_{111}\theta^s_{001}\right) = h^s_{10}(\boldsymbol{\theta}^s) \\
h_{11}(\boldsymbol{\theta}) &= \frac{T}{2}\left(\theta_{101}\theta_{011} + \theta_{111}\theta_{001}\right) = \alpha^2\gamma\left(\frac{\alpha T}{2}\left(\theta^s_{101}\theta^s_{011} + \theta^s_{111}\theta^s_{001}\right)\right) = h^s_{11}(\boldsymbol{\theta}^s) \\
h_{12}(\boldsymbol{\theta}) &= T\theta_{111}\theta_{011} + \frac{T^2}{4}\theta_{111}\theta_{101} = \alpha^2\left(T\theta^s_{111}\theta^s_{011} + \frac{\alpha T^2}{4}\theta^s_{111}\theta^s_{101}\right) = h^s_{12}(\boldsymbol{\theta}^s) \\
h_{13}(\boldsymbol{\theta}) &= \frac{T}{2}\theta_{110}\theta_{100} = \alpha^2\gamma\frac{\alpha T}{2}\theta^s_{110}\theta^s_{100} = h^s_{13}(\boldsymbol{\theta}^s) \\
h_{14}(\boldsymbol{\theta}) &= \frac{T}{2}\left(\theta_{110}\theta_{101} + \theta_{111}\theta_{100}\right) = \alpha^2\gamma\frac{\alpha T}{2}\left(\theta^s_{110}\theta^s_{101} + \theta^s_{111}\theta^s_{100}\right) = h^s_{14}(\boldsymbol{\theta}^s) \\
h_{15}(\boldsymbol{\theta}) &= \frac{T}{2}\theta_{110}\theta_{110} = \alpha^2\frac{T}{2}\theta_{110}\theta_{110} = h^s_{15}(\boldsymbol{\theta}^s) \\
h_{16}(\boldsymbol{\theta}) &= T\theta_{110}\theta_{111} = \alpha^2 T\theta^s_{110}\theta^s_{111} = h^s_{16}(\boldsymbol{\theta}^s) \\
h_{17}(\boldsymbol{\theta}) &= \frac{T}{2}\theta_{110} + \frac{T^2}{4}\theta_{110}\theta_{010} = \frac{\alpha}{\gamma}\left(\frac{T}{2}\theta_{110} + \frac{\alpha T^2}{4}\theta_{110}\theta_{010}\right) = h^s_{17}(\boldsymbol{\theta}^s) \\
h_{18}(\boldsymbol{\theta}) &= \frac{T}{2}\theta_{111}\theta_{101} = \frac{\alpha}{\gamma}\frac{T}{2}\theta^s_{111}\theta^s_{101} = h^s_{18}(\boldsymbol{\theta}^s) \\
h_{19}(\boldsymbol{\theta}) &= \frac{T}{2}\theta_{111}\theta_{111} = \frac{\alpha}{\gamma}\frac{T}{2}\theta^s_{111}\theta^s_{111} = h^s_{19}(\boldsymbol{\theta}^s) \\
h_{20}(\boldsymbol{\theta}) &= \frac{T}{2}\theta_{111} + \frac{T^2}{4}\left(\theta_{111}\theta_{010} + \theta_{110}\theta_{011}\right) \\
&= \frac{\alpha}{\gamma}\left(\frac{T}{2}\theta_{111} + \frac{\alpha T^2}{4}\left(\theta^s_{111}\theta^s_{010} + \theta^s_{110}\theta^s_{011}\right)\right) = h^s_{20}(\boldsymbol{\theta}^s)
\end{aligned}
$$

Table 1: The percentage error between the calculated parameters and the true parameters for different values of α.

α	percentage error [%]
1	0.081
1.25	0.055
1.5	0.043
1.75	0.038
2	0.039
2.25	0.047
2.5	0.068
2.75	0.088
3	0.118
3.25	0.148
3.5	0.181
3.75	0.181
4	0.254

$$h_{21}(\boldsymbol{\theta}) = \frac{T^2}{4}\theta_{110}\theta_{110} = \frac{\alpha^2}{\gamma}\frac{T^2}{4}\theta^s_{110}\theta^s_{110} = h^s_{21}(\boldsymbol{\theta}^s)$$
$$h_{22}(\boldsymbol{\theta}) = \frac{T^2}{2}\theta_{111}\theta_{110} = \frac{\alpha^2}{\gamma}\frac{T^2}{2}\theta^s_{111}\theta^s_{110} = h^s_{22}(\boldsymbol{\theta}^s)$$
$$h_{23}(\boldsymbol{\theta}) = \frac{T^2}{4}\theta_{111}\theta_{011} = \frac{\alpha^2}{\gamma}\frac{T^2}{4}\theta^s_{110}\theta^s_{011} = h^s_{23}(\boldsymbol{\theta}^s)$$
$$h_{24}(\boldsymbol{\theta}) = \frac{T^2}{4}\theta_{111}\theta_{111} = \frac{\alpha^2}{\gamma}\frac{T^2}{4}\theta^s_{111}\theta^s_{111} = h^s_{24}(\boldsymbol{\theta}^s)$$

From the relationship between h_{15} and h^s_{15}, for example, a theoretical relationship between θ_{110} and θ^s_{110} can be derived and used for derivation of the relationships between the other parameters. Nevertheless, these result show to be incosistent. Therefore, the numerical least squares solution is necessary for finding the nonscaled parameters. The results for the least square solution of the set of equations (34) are shown in table 1.

Example 4 To verify the results of Theorem 3, the condition number of the Hessian was estimated using Theorem 3 and compared to the condition number obtained by numerical simulation. The results are illustrated in Fig *4*.

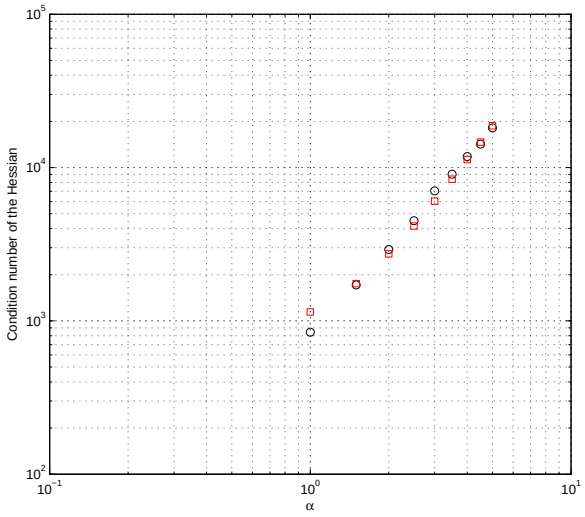

Figure 4: The condition number of the Hessian as a function of α, where squares represent the calculated condition number and circles represent the simulated results.

5 Conclusion

This paper has discussed an improvement of the RPEM described in [14] by a change of the dicretisation algorithm used. The effects of this new method on the state vector, the parameter vector and the Hessian were discussed and verified by numerical examples.

In the future, it would be of great interest to analyse the influence of general discretisation algorithm on the RPEM and to apply the present algorithm to challenging real world examples.

6 Acknowledgement

The work by the first author has been financially supported by *Swedish Energy Agency* (project 32299-1) which is greatfully acknowledged.

References

[1] K. J. Åström and R. D. Bell. Drum boiler dynamics. *Automatica*, 36:363–378, 2000.

[2] S. A. Billings and S. Y. Fakhouri. Identification of systems containing linear dynamic and static nonlinear elements. *Automatica*, 18:15–26, 1982.

[3] T. Bohlin. A case study of grey box identification. *Automatica*, 30:307–318, 1994.

[4] L. Brus, T. Wigren, and D. Zambrano. Mpc of a nonlinear solar collector plant with varying delays. *to appear in IET J. Contr. Theory and Applications*, 2010.

[5] Steven C. Chapra. *Applied Numerical Methods with MATLAB for Engineers and Scientists*. McGraw-Hill Science/Engineering/Math, 2006.

[6] S. Chen and S. A. Billings. Representation of nonlinear systems: the NARMAX model. *Int. J. Contr.*, 49:1013–1032, 1989.

[7] S. Chen and S. A. Billings. Neural networks for nonlinear dynamic system modelling and identification. *Int. J. of Contr.*, 56:319–346, 1992.

[8] M. T. Heath. *Scientific Computing: an Introductory Survey*. McGraw-Hill, New York, NY, 1997.

[9] L. Ljung and T. Söderström. *Theory and Practice of Recursive Identification*. MIT Press, Cambridge, MA, 1983.

[10] G. A. Pajunen. Adaptive control of Wiener type nonlinear systems. *Automatica*, 28:781–785, 1992.

[11] J. Sjöberg, 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 overview. *Automatica*, 31(12):1691–1724, 1995.

[12] T. Wigren. Recursive identification based on nonlinear state space models applied to drum-boiler dynamics with nonlinear output equations. In *Proceedings of American Control Conference*, pages 5066–5072, Portland, Oregon, U.S.A., June 2005.

[13] T. Wigren. Scaling of the sampling period in nonlinear system identification. In *Proceedings of American Control Conference*, pages 5058–5065, Portland, Oregon, U.S.A., June 2005.

[14] T. Wigren. Recursive prediction error identification and scaling of nonlinear state space models using a restricted black box parameterization. *Automatica*, 42(1):159–168, 2006.

[15] T. Wigren, L. Brus, and S. Tayamon. Matlab software for recursive identification and scaling using a structured nonlinear black-box model - revision 5. Technical report, Division of Systems and Control, Uppsala University, Uppsala, Sweden, January 2010. http://www.it.uu.se/research/publications/reports/2010-002/NRISSoftwareRev5.zip.