NONLINEAR DYNAMICS IDENTIFIED BY MULTI-INDEX MODELS [1]

David Lindgren and Lennart Ljung

Division of Automatic Control
Linköping University, SE 581 83 Linköping, Sweden,
`davlin@foi.se, ljung@isy.liu.se`

Abstract: We study nonlinear regression models of, for example, NARX-type, where the predicted output is determined as a nonlinear function of known past data. A particular structure of the nonlinear mapping is imposed, which confines the nonlinearities to a subspace of the regression space. Utilizing this structure simplifies the estimation problem and allows more efficient parameterizations as well as visualization of the nonlinearity. We show how the LS fit of *polynomials* and *piecewise affine functions* are used as criteria to find the projection that best describes the residual. A study of two particular nonlinear systems illustrates that the regressor can be projected down to 2 dimensions, and still yield a model simulation fit of around 99%. An electronic circuit can be accurately modeled with far less parameters than conventional, black-box models, of, say, neural network type. *Copyright ©2005 IFAC*

Keywords: System Identification, Nonlinear Models

1. INTRODUCTION

Consider the nonlinear regression model

$$y_t = f(\boldsymbol{\varphi}_t) + v_t, \tag{1}$$

where $\boldsymbol{\varphi}_t$ is a regression vector known at time t. For most of the discussion in this paper, the way $\boldsymbol{\varphi}$ is constructed from measurement is immaterial, but in order to fix ideas we may think of an underlying NARX (Nonlinear ARX) structure with

$$\boldsymbol{\varphi}_t = \begin{bmatrix} y_{t-1} \\ \vdots \\ y_{t-n_a} \\ u_{t-n_k} \\ \vdots \\ u_{t-n_b-n_k+1} \end{bmatrix} \in \mathbb{R}^n \tag{2}$$

where u and y are inputs and outputs of a system, possibly vectors. (For notational convenience, in this paper we will assume that f is scalar-valued, so multi-output models have to be treated by one output at a time.). The dimension of $\boldsymbol{\varphi}$ will be denoted by n.

We shall assume the following special structure for the mapping f:

$$f(\boldsymbol{\varphi}_t) = \boldsymbol{b}^T\boldsymbol{\varphi}_t + g(\boldsymbol{S}^T\boldsymbol{\varphi}_t), \tag{3}$$

where $\boldsymbol{S}$ is a $n \times k$ ($k < n$) matrix. This means that g is a mapping from $\mathbb{R}^k$ to $\mathbb{R}$, so the nonlinearity is confined to a k-dimensional space. Structures of this kind have been called *single-index* and *multi-index* structures in the statistical literature, see, e.g. (Carrol *et al.*, 1997). g can be thought of as the residual of the linear part $\boldsymbol{b}$, a *nonlinear* residual that is confined to the column space of the matrix $\boldsymbol{S}$. When $k << n$ this implies that the complexity of the parameterization and estimation of g will

[1] This work was supported by the *Swedish Research Council*. It is also part of the VISIMOD project supported by the *Swedish Foundation for Strategic Research*.

be greatly reduced. A nonlinear structure in, say, 10 dimensions have many parameters, and also as many as 10^6 observations (sample points) will be *very sparse* in $\mathbb{R}^{10}$. The multi-index structure allows the nonlinearity to be modeled in low dimensions, where the point density is larger and where functions may be defined by few parameters.

Of course, the structure (3) is a restriction. Sometimes one may realize on physical grounds that such a structure is at hand, in other cases one can simply try it and see how well it works out. One may think of feed-forward, sigmoidal ("ridge") neural networks with several terms $g_r(\boldsymbol{S}_r^T\boldsymbol{\varphi})$ with $k = 1$ as a way to build up a subspace (the joint range of $\boldsymbol{S}_r$, $r = 1, \ldots, d$) to which the nonlinearity is confined.

An important question is how to estimate $\boldsymbol{S}$. A very concrete way in case $k = 1$ would be to plot the residuals

$$y_t - \boldsymbol{b}^T\boldsymbol{\varphi}_t \tag{4}$$

against the regressor projection $\boldsymbol{S}^T\boldsymbol{\varphi}_t$. This plot would correspond to $g(\boldsymbol{S}^T\boldsymbol{\varphi}_t) + v_t$ and should thus look like a well defined curve, having a small "area". We should thus look for a matrix $\boldsymbol{S}$ that makes this projection of the data points $y_t, \boldsymbol{\varphi}_t$ appear to have a small area. This is a task that could be approached by visualization, cf (Johansson *et al.*, 2005). It was also investigated in (Lindgren and Ljung, 2004) using Delaunay triangulation to measure the area formed by the points. The success of this was limited.

In this paper we shall investigate criteria where $\boldsymbol{S}$ and g are estimated simultaneously. See e.g. (Hastie *et al.*, 2001) for a comprehensive text on nonlinear modeling, which contains many relevant insights. In (Carrol *et al.*, 1997) the estimation of the parameters in (3) with the constraint $\boldsymbol{S}^T\boldsymbol{S} = 1$ is discussed in the framework of generalized linear models with quasi-likelihood criteria.

Example: Drained Water Tank

Theoretically, a water tank with an outlet hole in the bottom obeys the nonlinear differential equation

$$\frac{d}{dt}y(t) = -\sqrt{y(t)} + u(t). \tag{5}$$

Here, $y(t)$ is the water level (system output) and $u(t)$ some inlet flow (system input). For convenience, the physical dimensions of the tank are here designed in a way that makes all coefficients in the differential equation equal unity. Sampling data from the tank gives a set $\{y_t, u_t\}_1^N$. This data set is also studied in (Lindgren and Ljung, 2004) and (Lindgren, 2005). As a regression vector we use (2) with $n_a = 3, n_b = 3, n_k = 1$ (which makes $n = 6$) and we use $k = 1$ (the dimension of the nonlinearity subspace, so that $\boldsymbol{S}$ is a 6-dimensional vector).

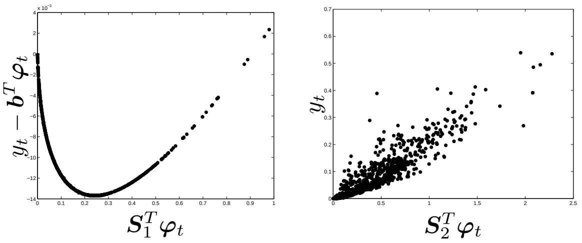

Fig. 1. Regressor projections of tank data. To the left $\boldsymbol{S}_1$ from a multi-index model, to the right $\boldsymbol{S}_2$ from partial least squares (PLS), see (6).

To illustrate the approaches to estimate $\boldsymbol{S}$ mentioned above, we plot the residuals (4) against $\{\boldsymbol{S}^T\boldsymbol{\varphi}_t\}$ in Fig. 1. In the plot to the left, $\boldsymbol{S}_1$ and $\boldsymbol{b}$ are the estimated parameters of a multi-index model (3) of the tank. The algorithm is the one described in Section 4.2 below. In the plot to the right, on the other hand, the first latent variable $\boldsymbol{S}_2$ of *partial least squares* is plotted versus y_t. (That is, we have no preliminary linear model: $\boldsymbol{b} = 0$.) Partial least squares finds a projection that maximizes the sample covariance with y_t and is a well-established linear technique to calculate regressor projections. The method could be interpreted as finding the projection that gives the smallest area of the projected points, if *size is measured as the area of the covariance ellipsoid of the points.* The projections in Fig. 1 are

$$\begin{aligned}\boldsymbol{S}_1 &= \begin{bmatrix}0.98 & -0.2 & 0.025 & 0.052 & -0.013 & 0.001\end{bmatrix}^T \\ \boldsymbol{S}_2 &= \begin{bmatrix}0.22 & 0.22 & 0.21 & 0.53 & 0.54 & 0.55\end{bmatrix}^T\end{aligned} \tag{6}$$

It is interesting to observe that it actually exists a 1-dimensional linear projection that totally isolates the nonlinearity. Evidently, it is not trivial to find it, however. The benefit of knowing $\boldsymbol{S}$ in the subsequent modeling is clear. Not only can g be parameterized in low dimensions. Visualization like the plot in Fig. 1 (left) may give the user a clue which nonlinear function family to be used.

2. MODEL PARAMETERIZATION

There are two issues involved in parameterizing the nonlinear part $g(\boldsymbol{S}^T\boldsymbol{\varphi}_t)$: the linear projection $\boldsymbol{S}$ from $\mathbb{R}^n$ to $\mathbb{R}^k$ and the nonlinear function g from $\mathbb{R}^k$ to $\mathbb{R}$. In this paper we only treat the case $k = 1$. Below are described the parameterizations mainly considered to this date: linear projection parameterized by Givens rotations and nonlinearity by polynomial or piecewise affine function.

2.1 Linear Projection $\boldsymbol{S}$

For identifiability, $\boldsymbol{S}$ is required to be orthonormal, $\boldsymbol{S}^T\boldsymbol{S} = I$. For the case we study where $k = 1$, this means that $\boldsymbol{S}$ is required to be a vector with length 1. This vector is parameterized by a product of $n-1$ Givens rotations, see (Golub and Loan, 1996, pp. 226) or (Lindgren, 2005, pp. 35). Thus, $\boldsymbol{S} = \boldsymbol{S}(\boldsymbol{p})$, where $\boldsymbol{p}$ is a new parameter vector with $n-1$ entries. For every $\boldsymbol{p}$ holds that $\|\boldsymbol{S}(\boldsymbol{p})\|_2 = 1$.

2.2 Polynomial g

A polynomial parameterization of the nonlinear part of (3) takes the form

$$g^{(p)}(x;\boldsymbol{c}) = \begin{bmatrix}1 \ x^2 \ x^3 \dots x^l\end{bmatrix}\boldsymbol{c}. \qquad (7)$$

Note that (7) is an ordinary polynomial except the linear term x. The linear term is already accounted for by the parameter $\boldsymbol{b}$ in (3).

A well-known drawback modeling dynamics by high-order polynomials is that they usually behave bad outside the support of the estimation data. However, they are very easy both to fit to data, and to differentiate. In this application we may also consider the parameterization of g a tool used to estimate the projection $\boldsymbol{S}$. Once $\boldsymbol{S}$ is at hand, a more robust function family can be used.

2.3 Piecewise Affine g

A rather general way to model a nonlinear curve is to approximate it with a number of line segments. This means that in a (small) region of the curve, an affine function (a constant plus linear term) is fitted. The more complicated curve we model, the more affine functions are of course needed to get a good approximation. Formally the piecewise function takes the form

$$g^{(a)}(x;\boldsymbol{c}) = \sum_{i=1}^{l}(\alpha_i + \beta_i x)w_i(x), \qquad (8)$$

where

$$\boldsymbol{c} = \begin{bmatrix}\alpha_1 \ \alpha_2 \cdots \alpha_l \ \beta_1 \ \beta_2 \cdots \beta_l \ z_1 \ z_2 \cdots z_l\end{bmatrix}^T$$

represents the parameter vector. The weight functions $w_i(\cdot)$ allocate every x in $\mathbb{R}$ to a local sum of affine functions. The affine functions are localized by knots (center points) z_i, one for each affine function. The weight functions are, given z_i, induced by an interpolator $\mathcal{P}$:

$$w_i(x) = \begin{cases}\mathcal{P}\left(\dfrac{x-z_i}{z_{i+1}-z_i}\right) & \text{if } x > z_i,\\ \mathcal{P}\left(\dfrac{z_i - x}{z_i - z_{i-1}}\right) & \text{else.}\end{cases} \qquad (9)$$

Here, $z_0 = -\infty$ and $z_{l+1} = \infty$. The interpolator is trigonometric, and gives a smooth transition from one affine function to the next,

$$\mathcal{P} = \begin{cases}\cos^2 \frac{\pi}{2}t & \text{if } t < 1,\\ 0 & \text{else.}\end{cases} \qquad (10)$$

This is very similar to splines, see (de Boor, 1978), but the trigonometric basis functions are somewhat different. The knots are chosen with respect to the distribution of x, see below.

Compared to the polynomial, the piecewise affine function is much more complicated to estimate and handle. However, it gives a good balance between adaptivity and stability/variance near the boundaries of the estimation set support.

2.3.1. Placing the Knots As described above, the weight functions are induced by a set of knots z_i, and an important task is to locate these knots. Assume we are given a set of real points $\{x_t\}_1^N$ that should be mapped by g. A simple solution to the problem is to place the knots on a uniform grid over the interval where there is support of points x_t. The model parameters are somewhat better utilized, however, if relatively more knots are allocated to dense regions of $\boldsymbol{x}$ (regions where many x_t are located).

It is known from splines theory that it is difficult to locate knots optimally, see (de Boor, 1978). A fast *ad hoc* procedure has been used, recursively dividing dense regions into subregions in a way that avoids too skew distributions within the subregions. Then a knot is allocated to each region.

3. MODEL VALIDATION

The one step ahead predictor for a given model $\{\boldsymbol{S}, \boldsymbol{b}, g\}$ is

$$\hat{y}_t = \boldsymbol{b}^T\boldsymbol{\varphi}_t + g(\boldsymbol{S}^T\boldsymbol{\varphi}_t) \qquad (11)$$

for $\boldsymbol{\varphi}_t$ defined in (2). Comparing this model output to the sampled output defines the model one step ahead prediction errors

$$e_t = y_t - \hat{y}_t. \qquad (12)$$

The root mean square (RMS) error for a data set $\{y_t, u_t\}_t^N$ is

$$E = \sqrt{\frac{1}{N}\sum_{t=1}^{N} e_t^2}. \qquad (13)$$

The RMS prediction error E is one way to validate a model – small error means that the model fits data well. The dataset used for model validation ("validation data") should be different from that used for parameter estimation ("estimation data").

The one step ahead predictor uses system output samples up to time $t-1$ in the regression vector $\boldsymbol{\varphi}_t$. The model may be validated in *another* sense by withholding this information, and only supply the predictor with the initial system state at time $t=0$ and the input. The so-obtained prediction, denoted $\hat{y}_{t|0}$, corresponds to the output sequence of a *simulation*, feeding the model with the sampled input data u_t (only). $\hat{y}_{t|0}$ is defined like (11), but with y_{t-r} in (2) replaced by $y_{t-r|0}$. $e_{t|0}$ and E_0 are defined analogously to e_t and E. Note that if $n_a = 0$ in(2) (i.e. we have an NFIR model), then there is no difference between E and E_0.

The prediction error E and the simulation error E_0 indeed validate a model in different senses. Validation by E_0 is generally more demanding and revealing, since the model dynamics are fully tested here. The computation of E_0 is however rather complex, since it is founded on a model simulation. In contrast, E can often be calculated mainly by linear matrix operations.

4. PARAMETER ESTIMATION

Given data $\{y_t, u_t\}_1^N$ and a model structure $\{n_a, n_b, n_k, g(\cdot;\cdot)\}$, the least squares errors defined above are indeed functions of the model parameters, $E = E(\boldsymbol{p}, \boldsymbol{b}, \boldsymbol{c})$ and $E_0 = E_0(\boldsymbol{p}, \boldsymbol{b}, \boldsymbol{c})$. The least squares estimates of the model parameters are

$$\{\hat{\boldsymbol{p}}, \hat{\boldsymbol{b}}, \hat{\boldsymbol{c}}\} = \arg\min_{\boldsymbol{p},\boldsymbol{b},\boldsymbol{c}} E(\boldsymbol{p}, \boldsymbol{b}, \boldsymbol{c}), \tag{14}$$

and analogously for E_0. A model that minimizes E is often termed *prediction error model* and one that minimizes E_0 *output error model.*

In this section we discuss how to solve (14) for the suggested functions $g^{(p)}$ and $g^{(a)}$. The techniques are based on numerical local optimization programs. It should be said immediately that such programs can only be guaranteed to converge to a local minimum of E and E_0.

4.1 Prediction Error Model, Polynomial (7)

Minimizing the prediction error $E(\boldsymbol{p}, \boldsymbol{b}, \boldsymbol{c})$ is equivalent to minimizing

$$V(\boldsymbol{S}, \boldsymbol{b}, \boldsymbol{c}) = \sum_{t=1}^{N} \left[y_t - \boldsymbol{b}^T\boldsymbol{\varphi}_t - g(\boldsymbol{S}^T\boldsymbol{\varphi}_t; \boldsymbol{c})\right]^2 \tag{15}$$

with the constraint $\boldsymbol{S}^T\boldsymbol{S} = 1$. Here we assume that $g(\cdot)$ is differentiable, for instance the polynomial $g^{(p)}$ in (7).

Given some initial value $\boldsymbol{S}^{(0)}$, V is minimized locally by repeating until convergence ($j = 1, 2, \dots$):

(1) Let $\boldsymbol{b}^{(j)}$ and $\boldsymbol{c}^{(j)}$ solve $\min_{\boldsymbol{b},\boldsymbol{c}} V(\boldsymbol{S}^{(j-1)}, \boldsymbol{b}, \boldsymbol{c})$
(2) Let $\widetilde{\boldsymbol{S}}^{(j)}$ solve $\min_{\boldsymbol{S}} V(\boldsymbol{S}, \boldsymbol{b}^{(j)}, \boldsymbol{c}^{(j)})$.
(3) Let $\boldsymbol{S}^{(j)} = \widetilde{\boldsymbol{S}}^{(j)} / \parallel \widetilde{\boldsymbol{S}}^{(j)} \parallel_2$.

4.1.1. Initial Value The initial value $\boldsymbol{b}^{(0)}$ is calculated as the solution to the linear least squares problem

$$\min_{\boldsymbol{b}} E(0, \boldsymbol{b}, 0),$$

corresponding to a linear ARX model ($g(0;0) = 0$). $\boldsymbol{S}^{(0)}$ is calculated as the first singular vector of the residual of this ARX model.

4.1.2. Calculate $\boldsymbol{b}$ *and* $\boldsymbol{c}$ *given* $\boldsymbol{S}$ Assume that the projection $\boldsymbol{S}$ is given (fix) and we should calculate the minimizing parameters $\boldsymbol{b}$ and $\boldsymbol{c}$. For notational purposes, the iteration superscripts are here dropped. Let

$$X = \begin{bmatrix} \boldsymbol{\varphi}_1^T \\ \boldsymbol{\varphi}_2^T \\ \vdots \\ \boldsymbol{\varphi}_N^T \end{bmatrix}, \quad \boldsymbol{Y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}. \tag{16}$$

Then $\boldsymbol{b}$ and $\boldsymbol{c}$ are given as the solution to the linear least squares problem

$$\min_{\boldsymbol{b},\boldsymbol{c}} \left\| \left[\boldsymbol{X}\ 1\ (\boldsymbol{XS})^{\cdot 2} \cdots (\boldsymbol{XS})^{\cdot l}\right] \begin{bmatrix} \boldsymbol{b} \\ \boldsymbol{c} \end{bmatrix} - \boldsymbol{Y} \right\|_2. \tag{17}$$

By $(\boldsymbol{XS})^{\cdot r}$ is denoted the rth power of every vector entry.

4.1.3. Calculate $\boldsymbol{S}$ *given* $\boldsymbol{b}$ *and* $\boldsymbol{c}$ Assume now instead that $\boldsymbol{b}$ and $\boldsymbol{c}$ are given (fix) and that the minimizing projection $\boldsymbol{S}$ should be calculated. Consider the first two terms in the Taylor expansion at some location in the regressor space $\boldsymbol{S} = \boldsymbol{S}_i$:

$$\begin{aligned} & g(\boldsymbol{S}^T\boldsymbol{\varphi}_t; \boldsymbol{c}) \approx \\ & \quad \approx g(\boldsymbol{S}_i^T\boldsymbol{\varphi}_t; \boldsymbol{c}) + g'(\boldsymbol{S}_i^T\boldsymbol{\varphi}_t; \boldsymbol{c})(\boldsymbol{S} - \boldsymbol{S}_i)^T\boldsymbol{\varphi}_t \quad (18) \\ & \quad = u(\boldsymbol{S}_i, \boldsymbol{c}, \boldsymbol{\varphi}_t) + \boldsymbol{S}^T\boldsymbol{r}(\boldsymbol{S}_i, \boldsymbol{c}, \boldsymbol{\varphi}_t), \end{aligned}$$

where u and $\boldsymbol{r}$ have obvious definitions. Let

$$\boldsymbol{U}_i = \begin{bmatrix} u(\boldsymbol{S}_i, \boldsymbol{c}, \boldsymbol{\varphi}_1) \\ u(\boldsymbol{S}_i, \boldsymbol{c}, \boldsymbol{\varphi}_2) \\ \vdots \\ u(\boldsymbol{S}_i, \boldsymbol{c}, \boldsymbol{\varphi}_N) \end{bmatrix}, \boldsymbol{R}_i = \begin{bmatrix} \boldsymbol{r}(\boldsymbol{S}_i, \boldsymbol{c}, \boldsymbol{\varphi}_1)^T \\ \boldsymbol{r}(\boldsymbol{S}_i, \boldsymbol{c}, \boldsymbol{\varphi}_2)^T \\ \vdots \\ \boldsymbol{r}(\boldsymbol{S}_i, \boldsymbol{c}, \boldsymbol{\varphi}_N)^T \end{bmatrix}.$$

Then the *Gauss-Newton* update of $\boldsymbol{S}_i$ is given as

$$\boldsymbol{S}_{i+1} = \arg\min_{\boldsymbol{S}} \|\boldsymbol{R}_i\boldsymbol{S} - \boldsymbol{Y} - \boldsymbol{Xb} - \boldsymbol{U}_i\|_2. \tag{19}$$

Starting at $\boldsymbol{S}_0 = \boldsymbol{S}^{(j)}$, this inner loop ($i = 0, 1, \dots$) is repeated until convergence.

4.2 Prediction Error Method with Piecewise Affine Function (8)

Consider estimation with the prediction error method when the nonlinearity is parameterized with the piecewise affine function in (8). Define the weight matrix $\boldsymbol{W}$ in $\mathbb{R}^{N\times l}$ in which the entry on row t and column i (w_{ti}) is the weight that observation t has for the affine function i. The piecewise affine function in the projection S is then

$$g(\boldsymbol{S}^T\boldsymbol{\varphi}_t;\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{W}) = \sum_{i=1}^{l}(\alpha_i + \beta_i\boldsymbol{S}^T\boldsymbol{\varphi}_t)w_{ti} \quad (20)$$

and the objective function is

$$V(\boldsymbol{S},\boldsymbol{b},\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{W}) = \\ = \sum_{t=1}^{N}\left[y_t - \boldsymbol{b}^T\boldsymbol{\varphi}_t - g(\boldsymbol{S}^T\boldsymbol{\varphi}_t;\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{W})\right]^2. \quad (21)$$

Here, the parameter vector $\boldsymbol{c}$ is divided into

$$\boldsymbol{\alpha} = \begin{bmatrix}\alpha_1 \ \alpha_2 \cdots \alpha_l\end{bmatrix}^T, \quad \boldsymbol{\beta} = \begin{bmatrix}\beta_1 \ \beta_2 \cdots \beta_l\end{bmatrix}^T.$$

A complication here is that any reasonable policy for choosing the knots z_i, that in turn induce the weight functions, must take the distribution of the projected points $\boldsymbol{XS}$ into account. This was briefly described in Section 2.3.1.

Given some initial values $\boldsymbol{S}^{(0)}$, $\boldsymbol{b}^{(0)}$, and some initial weight matrix, $\boldsymbol{W}^{(0)}$, the minimization program repeats until convergence ($j = 1, 2, \dots$):

(1) Let $\boldsymbol{\alpha}^{(j)}$ and $\boldsymbol{\beta}^{(j)}$ solve
$$\min_{\boldsymbol{\alpha},\boldsymbol{\beta}} V(\boldsymbol{S}^{(j-1)},\boldsymbol{b}^{(j-1)},\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{W}^{(j-1)})$$
(2) Let $\widetilde{\boldsymbol{S}}^{(j)}$ and $\widetilde{\boldsymbol{b}}^{(j)}$ solve
$$\min_{\boldsymbol{S},\boldsymbol{b}} V(\boldsymbol{S},\boldsymbol{b},\boldsymbol{\alpha}^{(j)},\boldsymbol{\beta}^{(j)},\boldsymbol{W}^{(j)})$$
and $\boldsymbol{S}^{(j)}$,$\boldsymbol{b}^{(j)}$ be the normalized solution.
(3) Calculate the weights $\boldsymbol{W}^{(j)} = \boldsymbol{W}(\boldsymbol{XS}^{(j)})$.

4.2.1. Initial Values The initial value $\boldsymbol{b}^{(0)}$ is calculated as the solution to the linear least squares problem $\min_{\boldsymbol{b}} V(0,\boldsymbol{b},0,0,0)$, corresponding to a linear ARX model. $\boldsymbol{S}^{(0)}$ is calculated as the first singular vector of the residual of this ARX model. $\boldsymbol{W}^{(0)} = \boldsymbol{W}(\boldsymbol{Xb})$ (described below).

4.2.2. Calculate $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ Assume that $\boldsymbol{S}$, $\boldsymbol{W}$, $\boldsymbol{b}$ are given (fix) and that we seek α and β that minimize (21). Let

$$\boldsymbol{U} = \begin{bmatrix}\boldsymbol{W}_1^T\boldsymbol{S}^T\boldsymbol{\varphi}_1\\ \boldsymbol{W}_2^T\boldsymbol{S}^T\boldsymbol{\varphi}_2\\ \vdots\\ \boldsymbol{W}_N^T\boldsymbol{S}^T\boldsymbol{\varphi}_N\end{bmatrix}, \quad (22)$$

where $\boldsymbol{W}_t$ is the t:th row of $\boldsymbol{W}$. Then $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are given as the solution to the linear least squares problem

$$\min_{\alpha,\beta}\left\|\begin{bmatrix}\boldsymbol{W} \ \boldsymbol{U}\end{bmatrix}\begin{bmatrix}\boldsymbol{\alpha}\\ \boldsymbol{\beta}\end{bmatrix} - \boldsymbol{Y} - \boldsymbol{Xb}\right\|_2. \quad (23)$$

4.2.3. Calculate $\boldsymbol{S}$ and $\boldsymbol{b}$ Assume now instead that α and β are given and that we seek $\boldsymbol{S}$ and $\boldsymbol{b}$ that minimize (21). Then $\boldsymbol{S}$ and $\boldsymbol{b}$ are given as the solution to the linear least squares problem

$$\min_{S,b}\left\|\begin{bmatrix}\boldsymbol{\beta}^T\boldsymbol{W}^T\boldsymbol{X} \ \boldsymbol{X}\end{bmatrix}\begin{bmatrix}\boldsymbol{S}\\ \boldsymbol{b}\end{bmatrix} - \boldsymbol{Y} - \boldsymbol{W\alpha}\right\|_2. \quad (24)$$

Note that it is here possible to calculate the projection $\boldsymbol{S}$ directly by solving a simple linear least squares problem. With the polynomial we had to use Gauss-Newton iterates.

If necessary, $\boldsymbol{S}$ is normalized to unit length. Then $\boldsymbol{b}$ is recalculated as the solution to

$$\min_{b}\|\boldsymbol{Xb} - \boldsymbol{Y} - \boldsymbol{W\alpha} - U\beta\|_2. \quad (25)$$

4.3 Output Error Method

The output error method estimates the model by minimizing the model output error $E_0(\boldsymbol{p},\boldsymbol{b},\boldsymbol{c})$, see Section 3. This is a more complex problem compared to minimizing the prediction error, since the output error is calculated by a model simulation. We will not discuss how to implement this method efficiently here. In our preliminary experiments we have used simple standard programs for unconstrained minimization without derivatives, see for instance (Brent, 1973).

The nonlinear function g can of course be parameterized like $g^{(p)}$ in (7) or $g^{(a)}$ in (8) or by any other appropriate nonlinear parameterization.

5. NUMERICAL EXPERIMENTS

In the introductory example on page 2 the dynamics of a simple water tank were studied. In fact, the projection depicted in Fig. 1 were found by using the piecewise affine parameterization and the parameter estimation suggested in Section 4.2.

Now an *electronic circuit* will be modeled by a multi-index model with polynomial parameterization and the parameters estimated as proposed in Section 4.1. The data set used is known from literature as the *Silver Box Data.* The data are described in (Pintelon and Schoukens, 2001) and have been studied in a special session at the NOLCOS symposium 2004, (NOLCOS, 2004).

In theory, the electronic circuit obeys the nonlinear differential equation

$$m\frac{d^2y(t)}{dt} + d\frac{dy(t)}{dt} + ay(t) + by(t)^3 = u(t). \quad (26)$$

The cubic nonlinearity enters additively, so in the differential equation it is trivial to separate the nonlinear dependency by linear operation (projection). The question is if this can be done also for the sampled data.

Table 1. Results for the Silver box model. See the text for an explanation.

Model	n_a n_b n_k l	Prms	$10^3 E_0$, PE	$10^3 E_0$,OE
Lin	2 3 0 -	5	14	15
M-I	2 3 0 3	12	5.0	0.44
M-I	5 5 0 3	22	2.5	0.44
ANN	2 3 0 10	82	4.2	2.6
ANN	2 3 0 75	617	0.52	0.46

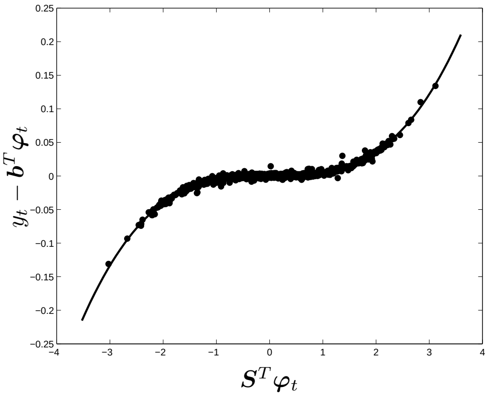

Fig. 2. Regressor projection versus linear model residual (subset with $N = 2000$ points). The model polynomial of degree 3 is also drawn. The model orders are $n_a = n_b = 5$, so the regression vector is in $\mathbb{R}^{10}$.

The sampling interval is $0.0016384s$. $N = 86916$ samples are available for estimation and 40000 for model validation.

Table 1 gives the validation errors for some different models structures fitted to the Silver box data, including linear, artificial neural network and multi-index models. The table shows the simulation fit to the validation data set (which is the "toughest" test – same as used in the NOLCOS session) both for models that were fitted using the prediction error (PE) and the output error (OE) approaches – see eq (14) and the text following. The table also shows the orders used in (2) and l, the polynomial order/number of hidden nodes. The total number of parameters (Prms) is also shown.

The multi-index model (3) with polynomial g fits very well. Using the prediction error criterion E, the multi-index model with 12 parameters has comparable error with an ANN model with 82 parameters. Using E_0 as criterion, there were no significant difference in error, even when comparing with the 617 parameter ANN model.

Fig. 2 depicts a regressor projection obtained from a multi-index model fitted with polynomial of degree $l = 3$. Here, $n_a = n_b = 5$, so the regressor space is 10-dimensional, cf. (NOLCOS, 2004). Since the points follow the curve well, the model is appropriate.

6. CONCLUSIONS

We have found that for some systems, the nonlinear dynamics can be well modeled in a low-dimensional linear projection of the regressor. These systems are with advantage modeled with a *multi-index structure* with two terms: one linear and one nonlinear.

For real life data sampled from an electronic circuit (the silver box data), it was seen that the nonlinearity could be well modeled in 2 dimensions with a model total of 12 parameters. This (output error) model performed as well as an artificial neural network with one hidden layer and 75 nodes (617 parameters), as measured by the simulation error for validation data.

REFERENCES

Brent, R. (1973). *Algorithms for minimization without derivatives.* Prentice-Hall, Englewood Cliffs, NJ

Carrol, R. J., J. Fan, I. Gijbels and M. P. Wand (1997). Generalized partially linear single-index models. *J Am Stat Assoc* **92**, 477–489.

de Boor, Carl (1978). *A Practical Guide to Splines.* Springer-Verlag. New York.

Golub, G. H. and C. F. Van Loan (1996). *Matrix Computations, 3 ed.* Johns Hopkins Univ. Press. Baltimore.

Hastie, T., Robert Tibshirani and Jerome Friedman (2001). *The Elements of Statistical Learning.* Springer. New York.

Johansson, J., D. Lindgren, M. Cooper and L. Ljung (2005). Interactive visualization as a tool for analysing time-varying and nonlinear systems. In: *Proc. 16th IFAC World Congress.* Prague, Czech Republic.

Lindgren, D. (2005). Projection Techniques for Classification and Identification. Dissertation no. 915. Department of Electrical Engineering, Linköping University. Linköping, Sweden.

Lindgren, D. and L. Ljung (2004). Nonlinear dynamics isolated by Delaunay triangulation criteria. In: *Proceedings of the 43rd IEEE Conference on Decision and Control.* Paradise Island, Bahamas. Paper ThC06.5

NOLCOS (2004). Special session on identification of nonlinear systems: The silver box study. In: *Proc. NOLCOS 2004 - IFAC Symposium on Nonlinear Control System.* Stuttgart, Germany.

Pintelon, R. and J. Schoukens (2001). *System Identification – A Frequency Domain Approach.* IEEE Press. New York.