Katholieke Universiteit Leuven

Departement Elektrotechniek

ESAT-SISTA/TR 04-110

Model Structure Determination and Identification with Kernel Based Partially Linear Models[1]

Marcelo Espinoza [2], Johan A.K. Suykens, Bart De Moor

May 2004
Submitted to IEEE TAC Special Issue on Linear-Nonlinear Identification.

[1]This report is available by anonymous ftp from *ftp.esat.kuleuven.ac.be* in the directory *pub/sista/mespinoz/reports/04-110.pdf.gz*

[2]K.U.Leuven, Dept. of Electrical Engineering (ESAT), Research group SCD-SISTA, Kasteelpark Arenberg 10, 3001 Leuven, Belgium, Tel. 32/16/32 17 09, Fax 32/16/32 19 70, WWW: *http://www.esat.kuleuven.ac.be/scd*. E-mail: *marcelo.espinoza@esat.kuleuven.ac.be*.

Abstract

In nonlinear system identification, the black-box approach can often be applied successfully. However, if we know a priori that some part of the model is linear, then the estimation of a partially linear model can produce better identification results. In this paper, by using Least Squares Support Vector Machines (LS-SVMs) as a black-box technique, and the proposed kernel-based partially linear model, a simple algorithm can detect if a subset of the regressors is linear. The application of this methodology to artificial examples shows satisfactory results. A real-life example, based on the Silverbox data, also shows significant improvements in the generalization ability of the structured model with respect to the full black-box formulation.

Model Structure Determination and Identification with Kernel Based Partially Linear Models

Marcelo Espinoza, Johan A.K. Suykens, Bart De Moor

K.U. Leuven, ESAT-SCD-SISTA

Kasteelpark Arenberg 10, B-3001 Leuven, Belgium

{marcelo.espinoza,johan.suykens}@esat.kuleuven.ac.be

Abstract

In nonlinear system identification, the black-box approach can often be applied successfully. However, if we know a priori that some part of the model is linear, then the estimation of a partially linear model can produce better identification results. In this paper, by using Least Squares Support Vector Machines (LS-SVMs) as a black-box technique, and the proposed kernel-based partially linear model, a simple algorithm can detect if a subset of the regressors is linear. The application of this methodology to artificial examples shows satisfactory results. A real-life example, based on the Silverbox data, also shows significant improvements in the generalization ability of the structured model with respect to the full black-box formulation.

Keywords: Model Structure Determination, Nonlinear System Identification, Kernel, LS-SVM.

1 Introduction

Within the framework of nonlinear system identification, it is common practice to estimate a nonlinear black-box model starting from a set of observations. Generally speaking, the black-box model can take different structures [21], and it can be implemented using different techniques (Artificial Neural Networks[2, 13], (Least-Squares) Support Vector Machines (LS-SVM) [25, 17, 27, 26], Wavelets [31], basis expansions [21], splines [28], etc). In practice, once the model has been estimated and validated, it can be used for prediction, simulation and/or control.

However, there might be good reasons to restrict the nonlinear black-box model formulation in order to include a linear part in the model, especially if we are interested in obtaining more insights about the structure of the model. According to the golden rule of "do not estimate what you already know", we should define an ad-hoc structure to model a system for which we know in advance that it contains a linear part, as e.g. in a Hammerstein model. Moreover, stronger motivations to explicitly include a linear part in the model can arise from practical considerations, as to check whether the system can be described within a certain accuracy by imposing a linear structure (e.g. for control purposes). Furthermore, the goal of a specific problem can be to identify a linear part that is based on first-principles, and at the same time including a nonlinear black-box component in order to keep the overall model accuracy within satisfactory levels. In [21], another motivation for such type of models arises when considering a linear model between inputs and outputs in which the disturbance term is known only in terms of a spectral density. Therefore, we should be able to build a model that contains a nonlinear component which is treated like a black-box structure, and at the same time containing a linear parametric part. Such model has been called a "partially linear model".

Although partially linear models have been developed since the mid-80s within the sta-

tistical literature [22, 18, 12], they are based on (local) smoothing techniques, usually restricted to low dimensional input vectors [11]. A more powerful black-box technique can be used for the identification of the nonlinear part of the model, and we use the LS-SVM framework[25] for this purpose.In this way, it is possible to incorporate input vectors of arbitrary dimension into the nonlinear component thanks to the application of the Mercer's theorem [3] using a positive definite kernel. In recent work [6], it was shown that there exist close links between the traditional statistical techniques and the LS-SVM model extension with a linear parametric part.

However, the partially linear model can also be used as an empirical tool for model structure determination, as we propose in this paper. Our objective here is to present a simple methodology, using the kernel-based partially linear model as an estimation technique, that can give insights into the model structure from an empirical point of view. In this paper we show that, if the true model contains a linear part, then it is possible to improve an existing nonlinear black-box model; the linear part can be identified and the complexity of the model can be reduced. Intuitively, we should expect that although we can train a nonlinear black-box model to accurately describe the system, the estimation can be improved if we can identify the linear part directly.

This paper is structured as follows. Section 2 describes the setting for the identification problem on which we want to focus. Section 3 presents the kernel-based partially linear model estimator for system identification, and the methodology proposed for model structure determination. Section 5 applies the methodology for two artificial examples where we know the true model, and for a real-life problem where we do not know the true model. Section 5 concludes.

2 The Problem

In this section we introduce some notation in order to describe the problem that we want to address, within the system identification framework.

2.1 General Black-Box Model

The objective of nonlinear system identification is to estimate a relationship between a sequence of inputs $u(t)$ and outputs $y(t)$ generated by an unknown system, under the assumption that the output can be described by a possibly nonlinear function. In general, a model is built using past values of the inputs up to a certain lag q, past values of the output up to a certain lag p, past predictions from the same model and stochastic disturbances with some structure. If only past values of both inputs and outputs are considered to build the model, this corresponds to a NARX (Nonlinear AutoRegressive with eXogenous input) formulation. Other structures (NARMAX, NOE, etc.) are described in [21] based in analogous linear system formulations [15].

Throughout this paper we use only NARX formulations, thus not considering any moving average part. Let us denote by $\boldsymbol{z}(t)$ the regression vector corresponding to the output $y(t)$:

$$\boldsymbol{z}(t) = [y(t-1), \cdots y(t-p), u(t), u(t-1), \cdots, u(t-q)]^T \tag{1}$$

where the superscript $()^T$ denotes the transpose. Each component of the regression vector is called a *regressor* [21]. The nonlinear system identification task is to estimate a function g such that

$$y(t) = g(\boldsymbol{z}(t)) + e(t) \tag{2}$$

where $e(t)$ is a disturbance term, assumed to be white noise. Under the nonlinear black box setting, the function g can be estimated by means of different implementations (ANNs,

SVMs, LS-SVMs, etc.), and the identification of the estimated $\hat{g}$ involves some training and validation stages [25]. The one step ahead prediction is simply $\hat{y}(t)$; simulation $n-$steps ahead can be obtained by iteratively applying the prediction equation replacing future outputs by its predictions [15, 21].

2.2 Linear-Nonlinear Structure

The usual nonlinear black-box paradigm assumes that the function g is nonlinear. However, many systems that we find in applied work can be described with reasonable accuracy by means of linear techniques, thus suggesting that the nonlinearity is not affecting the entire system; rather a subset of the relevant variables may be affected by it. The nonlinear black-box models do improve over linear identification techniques, but at the expense of losing the insights that we get from a linear parametric model. For example, consider a system for which the true model can be described as

$$y(t) = 0.5y(t-1)^3 - 0.5y(t-2) + 0.3u(t) - 0.2u(t-1) + e(t). \tag{3}$$

For this model, the regression vector is defined by $\boldsymbol{z}(t) = [y(t-1), y(t-2), u(t), u(t-1)]^T$. If a linear parametric model is estimated, it will have a misspecification error as, in absence of any prior information, the nonlinear term may not be identified a priori by the user. However, by moving to a full nonlinear black-box technique, the linear parts are not identified as such, yet as a part of a (usually nonparametric) general nonlinear model. The nonlinear black-box model may show a much better performance than the linear model, but the model complexity will possibly be larger than required.

Let us introduce now some notation. Let $\mathcal{Z} = \{\boldsymbol{x} : \boldsymbol{x}$ is a component of the vector $\boldsymbol{z}(t)\}$. Define an arbitrary partition $\mathcal{Z} = \mathcal{Z}^a \cup \mathcal{Z}^b$ with $\mathcal{Z}^a \cap \mathcal{Z}^b = \emptyset$. Now we can define a vector $\boldsymbol{z}^a(t)$ with regressors $\boldsymbol{x} \in \mathcal{Z}^a$, and a vector $\boldsymbol{z}^b(t)$ with regressors $\boldsymbol{x} \in \mathcal{Z}^b$. The superscript a represents the subset of regressors that enters linearly into the model; the superscript b

represents the subset of regressors that enters the model as a nonlinear component. In this way, the original regression vector can be written as

$$\boldsymbol{z}(t) = [\boldsymbol{z}^a(t), \boldsymbol{z}^b(t)]^T. \tag{4}$$

Thus, by defining $\boldsymbol{z}^a(t) = [y(t-2), u(t), u(t-1)]^T$ and $\boldsymbol{z}^b(t) = [y(t-1)]$ the model (3) can be written as

$$y(t) = \boldsymbol{\beta}^T \boldsymbol{z}^a(t) + g(\boldsymbol{z}^b(t)) + e(t), \tag{5}$$

which takes the form of a partially linear model as defined in the statistics literature [22, 18, 12, 11]. The nonlinear black-box model is now only a part of the model structure, allowing to identify a coefficient vector $\boldsymbol{\beta}$ for the linear part. The vectors $\boldsymbol{z}^a(t), \boldsymbol{z}^b(t)$ may contain any arbitrary selection over the original regressors definition. If we have, for instance, $\boldsymbol{z}^a(t)$ containing past outputs $y(t-k)$, and $\boldsymbol{z}^b(t)$ containing current and past inputs, the model will resemble a Hammerstein model[1]. The condition $\mathcal{Z}^a \cap \mathcal{Z}^b = \emptyset$ avoids the identifiability problem [18] that arises when a regressor is included simultaneously in the linear part and in the nonlinear component. Let us further denote by $\mathcal{M}_0$ the model class containing all models estimated by a nonlinear black-box model over the full set of regressors with orders (p, q) of the form (2). We denote by $\mathcal{M}_a$ the model class containing all models of the form (5). Clearly, class $\mathcal{M}_a$ is a more restrictive class of models than $\mathcal{M}_0$.

3 Kernel-Based Partially Linear Models

In this section, we describe an extension on the Least Squares Support Vector Machines (LS-SVM) nonlinear modelling technique [26, 24] in order to include a linear parametric term. For details about the original LS-SVM formulation, the interested reader is referred to [25].

[1] Although eventually more restrictions should be placed onto g, as $g = \sum_i b_i h(\boldsymbol{z}^b(t-i))$ for some static nonlinearity h.

3.1 Estimation of a kernel-based Partially Linear Model

Let us consider the model class $\mathcal{M}_a$:

$$y(t) = \boldsymbol{\beta}^T \boldsymbol{z}^a(t) + g(\boldsymbol{z}^b(t)) + e(t), \qquad t = 1, \ldots, N, \tag{6}$$

where $\boldsymbol{z}^a(t) \in \mathbb{R}^{N_a}, \boldsymbol{\beta} \in \mathbb{R}^{N_a}, \boldsymbol{z}^b(t) \in \mathbb{R}^{N_b}$, and $g : \mathbb{R}^{N_b} \to \mathbb{R}$ is an unknown nonlinear function. The terms $e(t)$ are assumed to be i.i.d. random errors with zero mean and constant (finite) variance. Using a regularized cost function and assuming that the function g takes the form $g(\boldsymbol{z}^b(t)) = \boldsymbol{w}^T \boldsymbol{\varphi}(\boldsymbol{z}^b(t))$, where $\boldsymbol{\varphi} : \mathbb{R}^{N_b} \to \mathbb{R}^{N_h}$ is a nonlinear mapping [25] from the input space to the so-called feature space (of dimension N_h which can be possibly infinite)[2], then a Partially Linear model can be formulated,

$$y(t) = \boldsymbol{\beta}^T \boldsymbol{z}^a(t) + \boldsymbol{w}^T \boldsymbol{\varphi}(\boldsymbol{z}^b(t)) + c + e(t), \qquad t = 1, \ldots, N, \tag{7}$$

where c is a constant (bias) term. The minimization problem (primal problem) can be formulated as

$$\begin{aligned} \min_{\boldsymbol{w},c,\boldsymbol{e},\boldsymbol{\beta}} \quad & \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \gamma\frac{1}{2}\boldsymbol{e}^T\boldsymbol{e} \\ \text{s.t.} \quad & \boldsymbol{y} = \boldsymbol{Z}\boldsymbol{\beta} + \boldsymbol{\Phi}\boldsymbol{w} + \mathbf{1}c + \boldsymbol{e}, \end{aligned} \tag{8}$$

with $\boldsymbol{Z} \in \mathbb{R}^{N \times N_a}$ the matrix of linear regressors $\boldsymbol{z}^a(t)$. This model has a unique solution if $\boldsymbol{Z}$ has full column rank [6]. As in standard LS-SVM, we build the Lagrangian $\mathcal{L}(\boldsymbol{w}, c, e, \boldsymbol{\beta}; \alpha) = \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \gamma\frac{1}{2}\boldsymbol{e}^T\boldsymbol{e} - \boldsymbol{\alpha}^T(\boldsymbol{y} - \boldsymbol{Z}\boldsymbol{\beta} - \boldsymbol{\Phi}\boldsymbol{w} - \mathbf{1}c - \boldsymbol{e})$, with $\boldsymbol{\alpha} \in \mathbb{R}^N$ the vector of Lagrange multipliers. The optimality conditions are obtained as follows:

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial \boldsymbol{w}} = 0 \;\; \to \;\; \boldsymbol{w} = \boldsymbol{\Phi}^T\boldsymbol{\alpha} \\ \frac{\partial \mathcal{L}}{\partial c} = 0 \;\; \to \;\; \boldsymbol{\alpha}^T\mathbf{1} = 0, \\ \frac{\partial \mathcal{L}}{\partial \boldsymbol{e}} = 0 \;\; \to \;\; \boldsymbol{\alpha} = \gamma\boldsymbol{e} \\ \frac{\partial \mathcal{L}}{\partial \boldsymbol{\beta}} = 0 \;\; \to \;\; \boldsymbol{Z}^T\boldsymbol{\alpha} = \mathbf{0}_{N_a \times 1} \\ \frac{\partial \mathcal{L}}{\partial \boldsymbol{\alpha}} = 0 \;\; \to \;\; \boldsymbol{y} = \boldsymbol{Z}\boldsymbol{\beta} + \boldsymbol{\Phi}\boldsymbol{w} + \mathbf{1}c + \boldsymbol{e}, \end{cases} \tag{9}$$

[2]This is one of the key elements in kernel-related models [27, 25]

where $\mathbf{0}_{N_a \times 1}$ is a zero-valued vector of dimension $N_a \times 1$. After elimination of $\boldsymbol{w}$ and $\boldsymbol{e}$, we obtain the linear system (dual problem)

$$\left[\begin{array}{c|c|c} \boldsymbol{\Omega} + \gamma^{-1}\mathbf{I} & \mathbf{1} & \boldsymbol{Z} \\ \hline \mathbf{1}^T & 0 & \mathbf{0}_{1\times p} \\ \hline \boldsymbol{Z}^T & \mathbf{0}_{p\times 1} & \mathbf{0}_{p\times p} \end{array}\right] \left[\begin{array}{c} \boldsymbol{\alpha} \\ \hline c \\ \hline \boldsymbol{\beta} \end{array}\right] = \left[\begin{array}{c} \boldsymbol{y} \\ \hline 0 \\ \hline \mathbf{0}_{1\times p} \end{array}\right], \tag{10}$$

where, with the application of Mercer's theorem, we have $\boldsymbol{\varphi}(\boldsymbol{z}^b(i))^T\boldsymbol{\varphi}(\boldsymbol{z}^b(j)) = K(\boldsymbol{z}^b(i), \boldsymbol{z}^b(j))$ for a positive definite kernel K; and therefore it is not required to compute explicitly[3] the nonlinear mapping $\boldsymbol{\varphi}$. The kernel matrix $\boldsymbol{\Omega}$, contains all evaluations of $K(\boldsymbol{z}^b(i), \boldsymbol{z}^b(j))$ for $i, j = 1, \cdots, N$. For $K(\boldsymbol{z}^b(i), \boldsymbol{z}^b(j))$ there are usually the following choices: $K(\boldsymbol{z}^b(i), \boldsymbol{z}^b(j)) = \boldsymbol{z}^b(i)^T\boldsymbol{z}^b(j)$ (linear kernel); $K(\boldsymbol{z}^b(i), \boldsymbol{z}^b(j)) = (\boldsymbol{z}^b(i)^T\boldsymbol{z}^b(j) + r)^d$ (polynomial of degree d, with $r > 0$ a tuning parameter); $K(\boldsymbol{z}^b(i), \boldsymbol{z}^b(j)) = \exp(-||\boldsymbol{z}^b(i) - \boldsymbol{z}^b(j)||_2^2/\sigma^2)$ (RBF kernel), where σ is a tuning parameter. In general, let θ_h denote the hyperparameters for the problem (10). When using a RBF kernel, $\theta_h = [\sigma, \gamma]$; when working with a polynomial kernel, $\theta_h = [r, d, \gamma]$, etc.

The solutions of (10) are $\boldsymbol{\alpha}, c$ and[4] $\hat{\boldsymbol{\beta}}$. The final model becomes

$$\hat{y}(t) = \hat{\boldsymbol{\beta}}^T\boldsymbol{z}^a(t) + \sum_{i=1}^{N} \alpha_i K(\boldsymbol{z}^b(t), \boldsymbol{z}^b(i)) + c, \tag{11}$$

where the solution is expressed in terms of the dual variables $\boldsymbol{\alpha}$ and kernel evaluations. From (10), the explicit expression for $\hat{\boldsymbol{\beta}}$ is

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{Z}^T(I - \mathcal{S})\boldsymbol{Z})^{-1}\boldsymbol{Z}^T(I - \mathcal{S})(\boldsymbol{y} - \mathbf{1}c), \tag{12}$$

where $S = \boldsymbol{\Omega}(\boldsymbol{\Omega} + \gamma^{-1}\mathbf{I})^{-\mathbf{1}}$. The constant term c, as in any regression, has to do with the fact that the data may or may not have been centered around their mean. So far, no

[3]This is another key element in kernel-based modelling [25]

[4]The computed $\boldsymbol{\alpha}, c$ do not have a "hat" symbol as it is not the standard notation within the machine learning framework. Besides, $\boldsymbol{\alpha}$ is a dual variable (Lagrange multiplier) and not a parameter vector to be estimated in a parametric setting.

assumptions have been made in that respect. From the related kernel theoretical results [19, 25] it is known that if the variables in feature space $\boldsymbol{\varphi}(\boldsymbol{z}^b(t))$ are centered, then there is no need for the c term. Therefore, if one wishes not to have the c term in the above formulation, all $\boldsymbol{y}$, $\boldsymbol{Z}$ and $\boldsymbol{\Phi}$ have to be centered about their corresponding means. For more details about links with traditional statistical techniques, and a Monte Carlo assessment of the estimation properties of the model (8) over different model classes, the reader is referred to [6].

3.2 Extension to a problem with a large number of datapoints

The solution for the above system (8) is obtained in terms of the dual variables (Lagrange multipliers) $\boldsymbol{\alpha}$, after solving the Karush-Kuhn-Tucker conditions (9). When the sample size N is large, the system is not practical and it may become eventually unfeasible from a computational point of view. By exploiting the primal-dual structure of the problem, it has been motivated [25, 29] to explicitly approximate the nonlinear mapping $\boldsymbol{\varphi}$ by means of a Nyström technique.

In general, explicit expressions for approximations to $\boldsymbol{\varphi}$ can be obtained by means of an eigenvalue decomposition of the kernel matrix $\boldsymbol{\Omega}$ with entries $K(\boldsymbol{z}^b(i), \boldsymbol{z}^b(j))$. Given the integral equation $\int K(\boldsymbol{z}^b, \boldsymbol{z}^b(j))\phi_i p(\boldsymbol{z}^b)d\boldsymbol{z}^b = \lambda_i\phi_i$, with solutions λ_i and ϕ_i for a variable $\boldsymbol{z}^b$ with probability density $p(\boldsymbol{z}^b)$, we can write

$$\boldsymbol{\varphi} = [\sqrt{\lambda_1}\phi_1, \sqrt{\lambda_2}\phi_2, \ldots, \sqrt{\lambda}_{n_h}\phi_{n_h}]. \tag{13}$$

Given the dataset $\{\boldsymbol{z}^b(i), y_i\}_{i=1}^N$, it is possible to approximate the integral by a sample average [30]. This will lead to the eigenvalue problem (Nyström approximation)

$$\frac{1}{N}\sum_{k=1}^{N} K(\boldsymbol{z}^b(k), \boldsymbol{z}^b(j))u_i = \lambda_i^{(s)}u_i, \tag{14}$$

where the eigenvalues λ_i and eigenfunctions ϕ_i from the continuous problem can be approximated by the sample eigenvalues $\lambda_i^{(s)}$ and eigenvectors u_i as $\hat{\lambda}_i = \frac{1}{N}\lambda_i^{(s)}, \hat{\phi}_i = \sqrt{N}u_i$. Therefore, in a sample of N points, the approximation of $\boldsymbol{\varphi}$ will contain only N components. The $i-th$ component can be obtained as

$$\hat{\boldsymbol{\varphi}}_i(\boldsymbol{z}^b(t)) = \frac{1}{\sqrt{\lambda_i}} \sum_{k=1}^{N} u_{ki} K(\boldsymbol{z}^b(k), \boldsymbol{z}^b(t)). \tag{15}$$

In a large sample problem, it has been proposed [25] to select a subsample of size M, with $M \ll N$, build a "small" kernel matrix $\boldsymbol{\Omega}_M$ only with this subsample, and apply equation (15) (now with N in (15) replaced by M) to compute the approximation of $\boldsymbol{\varphi}$. In this case, the approximation will have M components. External criteria such as entropy maximization can be applied for an optimal selection of the subsample: given a fixed-size M, the aim is to select the support vectors that maximize the quadratic Renyi entropy [9]

$$H_R = -\log \int p(x)^2 dx \tag{16}$$

that can be approximated by

$$\int \hat{p}(x)^2 dx = \frac{1}{M^2} \mathbf{1}^T \boldsymbol{\Omega}_M \mathbf{1}. \tag{17}$$

The use of this active selection procedure can be quite important for large scale problems, as it is related to the underlying density distribution of the sample. In this sense, the optimality of this selection is related to the final accuracy that can be obtained in the modeling exercise. The interest reader is referred to [25, 7, 5, 14] for further details on this technique. For our purposes, it suffices to know that it is possible to approximate the (possibly infinite dimensional) nonlinear mapping $\boldsymbol{\varphi}$ by the M-dimensional $\hat{\boldsymbol{\varphi}}$ and, therefore, estimate explicitly the model (7).

The following derivation complements the results of [6], expressing a solution in primal space for a large-scale problem. The term $\boldsymbol{e}^T\boldsymbol{e}$ from the original cost function (8) can be

written directly as:

$$\begin{aligned} \boldsymbol{e}^T\boldsymbol{e} &= \boldsymbol{w}^T\hat{\boldsymbol{\Phi}}^T\hat{\boldsymbol{\Phi}}\boldsymbol{w} + 2\boldsymbol{w}^T\hat{\boldsymbol{\Phi}}^T\boldsymbol{Z}\boldsymbol{\beta} - 2\boldsymbol{w}^T\hat{\boldsymbol{\Phi}}^T\boldsymbol{y} \\ &\quad -2\boldsymbol{\beta}^T\boldsymbol{Z}^T\boldsymbol{y} + \boldsymbol{\beta}^T\boldsymbol{Z}^T\boldsymbol{Z}\boldsymbol{\beta} + \boldsymbol{y}^T\boldsymbol{y} \\ &\quad -2\boldsymbol{w}^T\hat{\boldsymbol{\Phi}}^T\mathbf{1}c - 2\boldsymbol{\beta}^T\boldsymbol{Z}^T\mathbf{1}c + \mathbf{1}^T\mathbf{1}c \end{aligned} \tag{18}$$

By minimizing the regularized cost function $\mathcal{J}(\boldsymbol{w}, c, \boldsymbol{\beta}) = \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \gamma\frac{1}{2}\boldsymbol{e}^T\boldsymbol{e}$ using (18) with respect to $\boldsymbol{\beta}, c, \boldsymbol{w}$ (parametric estimation in primal problem), we obtain the following normal equations for $\hat{\boldsymbol{\beta}}$ and the nonlinear part $(\hat{\boldsymbol{\Phi}}\boldsymbol{w} + \mathbf{1}c)$:

$$\begin{aligned} \hat{\boldsymbol{\beta}} &= (\boldsymbol{Z}^T\boldsymbol{Z})^{-1}\boldsymbol{Z}^T(\boldsymbol{y} - \hat{\boldsymbol{\Phi}}\boldsymbol{w} - \mathbf{1}c) \\ \hat{\boldsymbol{\Phi}}\boldsymbol{w} + \mathbf{1}c &= \hat{\boldsymbol{\Phi}}(\hat{\boldsymbol{\Phi}}^T\hat{\boldsymbol{\Phi}} + \gamma^{-1}\mathbf{I_M})^{-1}\hat{\boldsymbol{\Phi}}^{\mathbf{T}}(\boldsymbol{y} - \boldsymbol{Z}\hat{\boldsymbol{\beta}}). \end{aligned} \tag{19}$$

This system (19) has the same form as a general partially linear model estimator within the statistical literature [11].

It is important to stress that the solutions (12) and (19) show that the optimal solution for the linear coefficients $\hat{\boldsymbol{\beta}}$ in dual and primal formulations has to be estimated taking into account the effect of the nonlinear component $\hat{\boldsymbol{\Phi}}\boldsymbol{w}$. In this sense, the common approach of performing *first* a linear regression and *later* estimate a nonlinear model on the residuals of the first regression is suboptimal, and it is equivalent to (19) only if $\boldsymbol{Z}^T\hat{\boldsymbol{\Phi}}\boldsymbol{w} = 0$ (linear part completely uncorrelated with the nonlinear component), in which case the first equation from (19) gives the usual least-squares estimate for a linear regression.

3.3 Expressions for the Equivalent Smoother Matrix

It is useful to write the predictions of a model M in the form $\hat{\boldsymbol{y}} = \mathcal{S}_M\boldsymbol{y}$. The equivalent matrix $\mathcal{S}_M$ is called the smoother matrix [28]. With some algebra, and assuming the data have been centered, it is possible to express the equivalent smoother matrix for the full

black-box model estimated with LS-SVM M_0,

$$\mathcal{S}_{M_0} = \boldsymbol{\Omega}(\boldsymbol{\Omega} + \gamma^{-1} I)^{-1}, \tag{20}$$

where $\boldsymbol{\Omega}$ is computed using $\boldsymbol{z}(t)$. For the kernel-based partially linear model M_a in primal space, the smoother matrix takes the form

$$\mathcal{S}_{M_a} = \begin{bmatrix} \hat{\boldsymbol{\Phi}} & \boldsymbol{Z} \end{bmatrix} \left(\begin{bmatrix} \hat{\boldsymbol{\Phi}}^T \hat{\boldsymbol{\Phi}} & \hat{\boldsymbol{\Phi}}^T \boldsymbol{Z} \\ \boldsymbol{Z}^T \hat{\boldsymbol{\Phi}} & \boldsymbol{Z}^T \boldsymbol{Z} \end{bmatrix} + \Lambda \right)^{-1} \begin{bmatrix} \hat{\boldsymbol{\Phi}}^T \\ \boldsymbol{Z}^T \end{bmatrix}, \tag{21}$$

where Λ is a matrix containing the regularization term γ according to (8), as

$$\Lambda = \left[\begin{array}{c|c} \gamma^{-1}\mathbf{I} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} \end{array}\right] \tag{22}$$

with $\mathbf{I}$ the identity matrix of dimension d, where d is the number of columns in $\hat{\boldsymbol{\Phi}}$. Finally, the equivalent smoother matrix for the kernel-based partially linear model M_a estimated in dual space takes the form

$$\mathcal{S}_{M_a} = \mathcal{S} - \mathcal{S}\mathcal{W} + \mathcal{W} \tag{23}$$

with

$$\begin{aligned} \mathcal{S} &= \boldsymbol{\Omega}(\boldsymbol{\Omega} + \gamma^{-1} I)^{-1} \\ \mathcal{W} &= \boldsymbol{Z}(\boldsymbol{Z}^T (I - \mathcal{S}) \boldsymbol{Z})^{-1} \boldsymbol{Z}^T (I - \mathcal{S}), \end{aligned} \tag{24}$$

where $\boldsymbol{\Omega}$ in this case is computed using $\boldsymbol{z}^b(t)$ only.

With all these derivations, we have practical expressions to estimate the model (6) in dual space (11) or in primal space (19) depending on the sample size N, where the nonlinear component is captured by a kernel-induced feature map and the overall model is based on a least-squares formulation.

4 Using the Partially Linear Model for Structure Determination

In this Section the main contribution of the paper is presented. Based on empirical criteria, it is possible to define a simple constructive algorithm for structure determination via the kernel-based partially linear model.

4.1 Basic Elements

Given a sample $\{y(t), u(t)\}_{t=1}^{N}$, let the regression vector of orders (p, q) be defined as $\boldsymbol{z}(t) = [y(t-1), \cdots y(t-p), u(t), u(t-1), \cdots, u(t-q)]^T$. Denote by T an index set for the datapoints used for training a model, and denote by G the index set for the datapoints used to test the performance of the model, with $T \cap G = \{\emptyset\}$. A number R of different definitions of the sets (T, G) can be indexed by $\{T_j, G_j\}_{j=1}^{R}$. Let $\mathcal{Z}_* = \{\boldsymbol{x} : \boldsymbol{x}$ is a regressor contained in $\boldsymbol{z}_*(t)\}$ the set defining the regressors in the true model. Consider the partition $\mathcal{Z}_* = \mathcal{Z}_*^a \cup \mathcal{Z}_*^b$ with $\mathcal{Z}_*^a \cap \mathcal{Z}_*^b = \emptyset$. Define the vectors $\boldsymbol{z}_*^a(t)$ with regressors $\boldsymbol{x} \in \mathcal{Z}_*^a$ and $\boldsymbol{z}_*^b(t)$ with regressors $\boldsymbol{x} \in \mathcal{Z}_*^b$.

Assumption 1 *There exists a unique pair $(\boldsymbol{\beta}_*, g_*)$ and a unique partition of the regression vector $\boldsymbol{z}^*(t)$ into $\boldsymbol{z}_*^a(t), \boldsymbol{z}_*^b(t)$ such that the true model can be defined by*

$$y(t) = \boldsymbol{\beta}_*^T \boldsymbol{z}_*^a(t) + g_*(\boldsymbol{z}_*^b(t)) + e(t). \tag{25}$$

Assumption 2 *The true model can be estimated with a model $M_0 \in \mathcal{M}_0$ (a fully nonlinear black-box model) using standard LS-SVM:*

$$y(t) = \hat{g}(\boldsymbol{z}(t)) + e(t) \tag{26}$$

with $\hat{g_{\theta_h}}(\boldsymbol{z}(t)) = \sum_{i \in T_j} \alpha_i K_{\theta_h}(\boldsymbol{z}(t), \boldsymbol{z}(i)) + c$*, and* θ_h *the vector of hyperparameters. If the estimated* $\hat{g_{\theta_h}}$ *minimizes the cross-validation mean squared error* $CV(\hat{g_{\theta_h}})$

$$\min_{\theta} CV(M_0) = CV(\hat{g_{\theta_h}}) = \frac{1}{R} \sum_{j=1}^{R} \left(\frac{1}{N_{G_j}} \sum_{i \in G_j} (y(i) - \hat{g}_{\theta_h}(\boldsymbol{z}(i)))^2 \right) \tag{27}$$

with R *the number of different definitions of training-validation datasets* $\{T_j, G_j\}_{j=1}^{R}$*, then the model has an optimal generalization ability among the class* $\mathcal{M}_0$*. Analogously, the true model can be estimated with a partially linear model* $M_a \in \mathcal{M}_a$ *with* $\mathcal{Z}^a \subseteq \mathcal{Z}^a_*(t)$ *using the kernel-based model (8). This model will have an optimal generalization ability in* $\mathcal{M}_a$ *if it minimizes the corresponding* $CV(M_a) = CV(\hat{\boldsymbol{\beta}}, \hat{g}_{\theta_h})$*.*

These assumptions can be motivated by empirical work and consistency results related to SVMs [23]. Identifiability of a partially linear model is a standard assumption [11, 8]. And the use of cross-validation [10] as a way to assess the generalization ability of any model is very common in practice; although the reader may use another empirical model selection criteria (VC dimension [27], GPE [16], BIC, AIC [1], etc.).

The following proposition is the central argument of the proposed algorithm.

Proposition 1 *A partially linear model* M_a *of the form*

$$y(t) = \boldsymbol{\beta}^T \boldsymbol{z}^a(t) + g(\boldsymbol{z}^b(t)) + e(t), \tag{28}$$

estimated as a minimizer of $CV(M_a) = CV(\hat{\boldsymbol{\beta}}, \hat{g}_{\theta_h})$ *for the hyperparameter* θ_h *is a better representation of the true model compared to a full nonlinear black box formulation* M_0 *of the form*

$$y(t) = g(\boldsymbol{z}(t)) + e(t), \tag{29}$$

estimated as a minimizer of $CV(M_0) = CV(\hat{g}_{\theta_h})$*, if* $CV(M_a) \leq CV(M_0)$*. If this is the case, under the assumptions above, then* $\mathcal{Z}^a(t) \subset \mathcal{Z}^a_*(t)$*.*

This proposition is based on empirical arguments, with Occam's razor principle that states that we should always prefer a simpler model. As M_a is a more restrictive model compared to M_0, model M_a contains a lower effective number of parameters than model M_0. Any observed reduction in the CV is generated by a genuine increase in accuracy, and also by the decrease in the effective number of parameters. As it was pointed out in [28, 4], for any estimator $\hat{f}$ of $y(t) = f(\boldsymbol{z}(t)) + e(t)$,

$$CV(\hat{f}) \approx \sum_{i \in T} \frac{(y(i) - \hat{f}(\boldsymbol{z}(i)))^2}{1 - \frac{\text{trace } \mathcal{S}_M}{N}} \tag{30}$$

where N is the number of datapoints used in the training set T, and $\mathcal{S}_M$ is the equivalent smoother matrix of $\hat{f}$. This means that even if two models can have the same in-sample accuracy, the one with the lower CV will be the model whose equivalent smoother matrix B has a lower trace. As the trace of the smoother matrix is equivalent with the effective number of parameters of the model [16], the model M_a should be preferable to M_0 even if both have the same in-sample accuracy. The last part of the proposition can be proved by contradiction. Suppose $\boldsymbol{z}^a(t)$ contains a regressor $\boldsymbol{x} \notin \mathcal{Z}_*^a(t)$. Then the restricted model M_a is misspecified, it does not include a nonlinear effect due to $\boldsymbol{x}$, therefore the generalization ability of M_a should be worse than the one from M_0, and therefore we should have $CV(M_a) > CV(M_0)$.

4.2 Structure Determination Algorithm

Our working algorithm can be described as follows. Given a sample $\{y(t), u(t)\}_{t=1}^N$:

1. Estimate a full nonlinear black-box model using LS-SVM, selecting the order (p, q) of the regression vector $\boldsymbol{z}(t)$ and the kernel parameters by means of cross-validation. Call this model M_0. The vector of regressors contains $(p + q)$ terms.

2. Initially, $\boldsymbol{z}^a(t) = \emptyset, \boldsymbol{z}^b(t) = \boldsymbol{z}(t)$.

3. Repeat for i=1,...,$(p+q)$:

   (a) Select the $i-$th regressor in $\boldsymbol{z}(t)$ and add it to $\boldsymbol{z}^a$. Define $\boldsymbol{z}^b(t)$ as containing the remaining regressors not in $\boldsymbol{z}^a(t)$.

   (b) Train a partially linear model M_a

$$y(t) = \boldsymbol{\beta}^T \boldsymbol{z}^a(t) + g(\boldsymbol{z}^b(t)) + e(t) \tag{31}$$

   as minimizer of the cross-validation error.

   (c) If $CV(M_a) \leq CV(M_0)$, then keep $\boldsymbol{z}^a(t)$ (the $i-$th regressor is linear); otherwise, remove it from $\boldsymbol{z}^a(t)$ (it can not be considered to be linear).

4. The regressors in $\boldsymbol{z}^a(t)$ have been identified as being linear.

5 Examples

In this Section, practical applications of the partially linear models in a model structure identification framework are presented. The first subsection contains artificial examples where the true system is known, and the second subsection shows a real-life example.

5.1 Example 1: The true model is known

Consider the following two systems:

$$\begin{aligned} \text{System 1: } y(t) &= (0.5y(t-1))^3 - 0.5y(t-2) + 0.3u(t) - 0.2u(t-1) + e(t) \\ \text{System 2: } y(t) &= 0.5y(t-1)^3 - 0.5y(t-2)u(t) + 0.3u(t) - 0.2u(t-1) + e(t). \end{aligned}$$

On each system, the input sequence $u(t)$ is generated by drawing a normal distribution of zero mean and unit variance; the error term $e(t)$ is i.i.d. normally distributed with

zero mean and standard deviation 0.1. Both systems were generated up to $N = 1000$ datapoints. The first 350 datapoints were discarded to avoid any transient effect; the training with cross-validation was done using the next 400 datapoinst; and the final 250 points were used as final test set.

According to our definitions, clearly System 1 contains $[y(t-2), u(t), u(t-1)]^T$ as a linear regression vector. Identifiability requires that a linear variable should not be contained also in the nonlinear component; therefore in System 2 only $u(t-1)$ is linear.

The application of the algorithms gives the following results. First, the overall order of the system is identified using a black-box NARX model M_0 trained with LS-SVM. Figure 1 shows the evolution of the $CV(M_0)$ over different model orders for System 1. The minimum is located at 2, which corresponds to the regression vector defined as $\boldsymbol{z}(t) = [y(t-1), y(t-2), u(t), u(t-1)]$. The same situation happens for System 2.

Once the regression vector has been identified, the algorithm is applied. In order to illustrate the outcome of the algorithm on each stage, we show the CV of all estimated models, and also the effective number of parameters N_{eff} which can be computed simply as the trace of the equivalent smoother matrix for each model according to (20) or (23).

Table 1 shows the evolution of the algorithm over the 4 possible regressors. It is clear that if a variable is originally nonlinear, the error will increase with respect to the original M_0 model, and therefore it will not be selected as linear. We see that even if the original M_0 model was estimated optimally, it is still possible to improve the performance by using a partially linear model correctly specified. For System 1, we have $CV(M_0) = 2.32 \times 10^{-4}$, which is reduced to $CV(M_a) = 1.20 \times 10^{-4}$; for System 2, where only one regressor gets identified as linear,we have $CV(M_0) = 2.31 \times 10^{-4}$ reduced to $CV(M_a) = 1.67 \times 10^{-4}$. The effective number of parameters in System 1 is reduced from 41 to 5.5. For System 2, it is reduced from 39 to 24.

Once the final selection has been made, the generalization ability of the model is assessed using the test set. We compute the one-step-ahead error and the simulation error [15] for the original black-box model M_0 and the structured model M_a. The results are reported in Table 2. We see that both partially linear models obtain a better generalization performance in new data.

Finally, we also see that the partially linear model can identify the linear coefficients. For System 1 we obtain $\hat{\boldsymbol{\beta}} = [-0.4996, 0.3002, 0.2002]^T$. For System 2, the only linear coefficient is identified at $\hat{\boldsymbol{\beta}} = 0.1996$. The identified nonlinear part for System 1 is shown in Figure 2, which clearly is a cubic function of $y(t-1)$.

5.2 A Real Life Problem: The SilverBox Data

This problem is an interesting test case of black-box versus white-box nonlinear system identification. The dataset was generated in a real physical system (an electric circuit) by J. Schoukens [20]. According to the first-principles, this system is described by a differential equation in continuous time:

$$m\frac{d^2y(t)}{dt^2} + d\frac{dy(t)}{dt} + ay(t) + by(t)^3 = u(t). \tag{32}$$

A dataset of 130,000 points was generated, as it is shown in Figure 3. Several research groups working in system identification were given the dataset, in order to compare different approaches and modelling techniques [5]. What makes this problem particularly challenging is that any model was supposed to be trained using datapoints 40,000 to 85,000 (the first-half of the body of the "arrow"); the validation set was the second half of the body of the "arrow"(points 85,000 to 130,000), and finally the model was tested in simulation mode in the "head of the arrow" (points 1 to 40,000). The test set contains, therefore, a zone

[5]The outcome of this work is the basis of a Special Session in the NOLCOS 2004 conference.

where the amplitude of the signals is beyond the range that was used for training. This was a revealing way to test the generalization ability of any model.

For that purpose, we built a full black-box model using LS-SVM. The typical selection procedure gives an optimal NARX regression vector of 10 lags. The root mean squared error in the test set, in simulation mode, was found to be $RMSE = 3.24 \times 10^{-4}$ using a fixed-size version estimated in primal space, which was one of the best results achieved. For details on this particular implementation, see [5].

Starting from this result, we continue further exploring the structure determination of the model. According to the first-principles equation, this model should be linear in the outputs. We estimate a kernel-based partially linear model by using $\boldsymbol{z}^a(t)$ containing all past lags of the input $u(t)$, and $\boldsymbol{z}^b(t)$ contains all past values of the outputs $y(t)$.

Figure 4 shows the errors obtained in simulation mode when testing the generalization performance of the model in the "head of the arrow" for the full nonlinear black-box LS-SVM. It is clear that there are significant errors in the final part, which corresponds to the wider zone of the "head of the arrow"[6]. Using a kernel-based partially linear model as described above, the results are much better, as it is shown in Figure 5. This result did not improve further when including also past lags of $y(t)$ as linear regressors. Therefore we can not consider any of the lags of $y(t-1)$ to be linear.

Table 3 shows a comparison between both models in terms of their in-sample accuracy, their validation performance, the simulation accuracy and the complexity. It is a good result that by imposing a linear structure we decrease the simulation root mean squared error to 2.7×10^{-4}; however, this result is misleading because of the large number of points being averaged. When considering only the last quarter of the test data for the average, the

[6]Although this method had an excellent performance in comparison with the results from the other participants.

improvement is more important, as shown in Table 4. Moreover, using the full nonlinear black-box model, the maximum absolute error is 0.0081. When using the partially linear model, it reduces to 0.0037. The mean absolute error for the full linear black-box model is 2.3×10^{-4}; for the partially linear model, it is 2.02×10^{-4}. The effective number of parameters is reduced from 490 to 190.

6 Conclusion

In this paper, we showed that it is possible to use a kernel-based partially linear model to sucessfully identify a model containing a linear part and a nonlinear component, with better identification results than a full nonlinear black-box model. Moreover, we propose a simple methodology for model structure determination, that can detect if a subset of the regressors is linear. Practical expressions in primal and dual space are provided.

In examples where the true model is known, our methodology is able to determine correctly the model structure and identify the linear parameters involved. The structured model obtained shows a better generalization ability, and a lower complexity, than a full nonlinear black-box model. In a real-life example, the Silverbox data, an existing nonlinear black-box model can be further improved by imposing a linear structure, as it is illustrated in the simulation performance.

Acknowledgments This work was supported by grants and projects for the Research Council K.U.Leuven (GOA-Mefisto 666, GOA-Ambiorics, several PhD/Postdocs & fellow grants), the Flemish Government (FWO: PhD/Postdocs grants, projects G.0240.99, G.0407.02, G.0197.02, G.0141.03, G.0491.03, G.0120.03, G.0452.04,G.0499.04, ICCoS, ANMMM; AWI;IWT:PhD grants, GBOU(McKnow)Soft4s), the Belgian Federal Government (Belgian Federal Science Policy Office: IUAP V-22; PODO-II (CP/01/40), the EU(FP5-Quprodis;ERNSI, Eureka 2063-Impact;Eureka 2419-FLiTE) and Contracts Research/Agreements

(ISMC/IPCOS,Data4s,TML,Elia,LMS,IPCOS,Mastercard). J. Suykens and B. De Moor are an associated professor and a full professor at the K.U.Leuven, Belgium, respectively. The scientific responsibility is assumed by its authors.

References

[1] H. Akaike. Information measures and model selection. *Bulletin of the International Statistical Institute*, 50:277–290, 1983.

[2] C.M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[3] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines*. Cambridge University Press, 2000.

[4] A. Elisseeff and M. Pontil. Leave-one-out error and stability of learning algorithms with applications. In Suykens, J.A.K. et al., editor, *Advances in Learning Theory: Methods, Models and Applications*. IOS Press, 2003.

[5] M. Espinoza, K. Pelckmans, L. Hoegaerts, J.A.K. Suykens, and B. De Moor. A comparative study of LS-SVMs applied to the silverbox identification problem. Technical Report 04-17, ESAT-SCD-SISTA, K.U.Leuven, Belgium, 2004. NOLCOS 2004, to appear.

[6] M. Espinoza, J. Suykens, and B. De Moor. Partially linear models and least squares support vector machines. Technical Report 04-57, ESAT-SCD-SISTA, K.U.Leuven, Belgium, 2004. Submitted to CDC 2004.

[7] M. Espinoza, J.A.K. Suykens, and B. De Moor. Least squares support vector machines and primal space estimation. In *Proc. of the IEEE 42nd Conference on Decision and Control (CDC)*, pages 3451–3456, 2003.

[8] J. Gao. Semiparametric regression modelling of nonlinear time series. *Scand. J. Statistics*, 25:521–539, 1998.

[9] M. Girolami. Orthogonal series density estimation and the kernel eigenvalue problem. *Neural Computation*, 10(6):1455–1480, 1998.

[10] G.H. Golub, M. Heath, and G. Wahba. Generalized cross-validation: a method for choosing a good ridge regression parameter. *Technometrics*, 21:215–223, 1979.

[11] W. Härdle. *Applied Nonparametric Regression*. Econometric Society Monographs. Cambridge University Press, 1989.

[12] W. Härdle, H. Liang, and J. Gao. *Partially Linear Models*. Physica-Verlag, Heidelberg, 2000.

[13] S. Haykin. *Neural Networks, A Comprehensive Foundation*. Macmillan, New York, 1994.

[14] L. Hoegaerts, J.A.K. Suykens, J. Vandewalle, and B. De Moor. Subset based least squares subspace regression in RKHS. *Neurocomputing*. In press.

[15] L. Ljung. *System Identification: Theory for the User*. Prentice Hall, New Jersey, 1987.

[16] J. Moody. Prediction risk and architecture selection for neural networks. In V. Cherkassy, J.H. Friedman, and H. Weschler, editors, *From Statistics to Neural Networks: Theory and Pattern Recognition Appications*, NATO ASI Series F. Springler-Verlag, 1994.

[17] T. Poggio and F. Girosi. Networks for approximation and learning. *Proceedings of the IEEE*, 78:1481–1497, 1990.

[18] P.M. Robinson. Root n-consistent Semiparametric Regression. *Econometrica*, 56(4):931–954, 1988.

[19] B. Schölkopf and A. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, 2002.

[20] J. Schoukens, G. Nemeth, Y. Crama, P. abd Rolain, and R. Pintelon. Fast approximate identification of nonlinear systems. *Automatica*, 39(7), 2003.

[21] J. Sjöberg, Q. Zhang, L. Ljung, A. Benveniste, B. Deylon, P. Glorennec, H. Hjalmarsson, and A. Juditsky. Nonlinear Black-box Modelling in System Identification: a Unified Overview. *Automatica*, 31:1691–1724, 1995.

[22] P. Speckman. Kernel smoothing in partial linear models. *Journal of the Royal Statistical Society, Series B*, 1988.

[23] I. Steinwart. On the influence of the kernel on the consistency of support vector machines. *Journal of Machine Learning Research*, 2:67–93, 2001.

[24] J.A.K. Suykens, J. De Brabanter, L. Lukas, and J. Vandewalle. Weighted least squares support vector machines: robustness and sparse approximation. *Neurocomputing*, 48(1-4):85–105, 2002.

[25] J.A.K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, and J. Vandewalle. *Least Squares Support Vector Machines*. World Scientific, Singapore, 2002.

[26] J.A.K. Suykens and J. Vandewalle. Least squares support vector machine classifiers. *Neural Processing Letters*, 9:293–300, 1999.

[27] V. Vapnik. *Statistical Learning Theory*. Wiley, New-York, 1998.

[28] G. Wahba. *Spline Models for Observational Data*. SIAM, Philadelphia, 1990.

[29] C.K.I Williams and M. Seeger. Using the nyström method to speed up kernel machines. In V.Tresp T.Leen, T.Dietterich, editor, *Proc. NIPS'2000*, volume 13, Vienna, Austria, 2000. MIT press.

[30] C.K.I Williams and M. Seeger. The effect of the input density distribution on kernel-based classifiers. In *Proceedings of the Seventeenth International Conference on Machine Learning (ICML 2000)*, 2001.

[31] Y. Yu and W. Lawton. Wavelet based modelling of nonlinear systems. In J.A.K. Suykens and J. Vandewalle, editors, *Nonlinear Modelling: Advanced Black Box Techniques*. Kluwer Academic Publishers, 1998.

Captions of Tables

Table 1:

Evolution of the algorithm applied to Systems 1 and 2. The black box model M_0 and each partially linear model M_a estimated with the corresponding $\boldsymbol{z}^a(t)$ are reported in terms of CV and N_{eff}.

Table 2:

Generalization performance of the models in the test set under simulation mode for Systems 1 and 2. The black box model M_0 and the selected partially linear model M_a are reported in terms of mean squared error.

Table 3:

Comparison between a black-box model and a partially linear model for the Silverbox data. For each model, the one-step-ahead validation error (RMSE_{val}), the one-step-ahead test error ($\text{RMSE}_{\text{test}}$),the simulation error in the test set (RMSE_{sim}) and the N_{eff} are reported.

Table 4:

Comparison between the simulation errors e of a black-box model and a partially linear model for the Silverbox data, over the full test set (Case I) or only in the last quarter of the test set (Case II).

Algorithm Results for System 1				
Step	Linear Regressors	$CV(\times 10^{-4})$	N_{eff}	Outcome
0	(full black-box)	2.32	41.13	-
1	$y(t-1)$	9.38	7.22	Nonlinear
2	$y(t-2)$	1.71	29.15	Keep as linear
3	$y(t-2), u(t)$	1.51	14.03	Keep as linear
4	$y(t-2), u(t), u(t-1)$	1.20	5.5	Keep as linear
Algorithm Results for System 1				
Step	Linear Regressors	$CV(\times 10^{-4})$	N_{eff}	Outcome
0	(full black-box)	2.31	39.86	-
1	$y(t-1)$	7.54	22.17	Nonlinear
2	$y(t-2)$	234.7	17.05	Nonlinear
3	$u(t)$	314.4	3.95	Nonlinear
4	$u(t-1)$	1.67	24.19	Keep as linear

Table 1

	Mean Squared Error in test set	
	Black-Box model	Partially Linear Model
System 1		
Prediction	1.44×10^{-4}	1.08×10^{-4}
Simulation	2.01×10^{-4}	1.31×10^{-4}
System 2		
Prediction	7.66×10^{-4}	2.50×10^{-4}
Simulation	0.0019	0.0011

Table 2

	Black-Box model	Partially Linear Model
RMSE_{val}	1.05×10^{-4}	0.45×10^{-4}
$\text{RMSE}_{\text{test}}$	1.75×10^{-4}	0.57×10^{-4}
RMSE_{sim}	3.24×10^{-4}	2.71×10^{-4}
N_{eff}	490	190

Table 3

	Black-Box model	Partially Linear Model
Case I: $i = 1, \cdots, 40,000$ (complete test set)		
$\max(\lvert e_i \rvert)$	0.0081	0.0037
$\text{mean}(\lvert e_i \rvert)$	2.30×10^{-4}	2.02×10^{-4}
$\text{RMSE}(e_i)$	3.24×10^{-4}	2.71×10^{-4}
Case II: $i = 30,000, \cdots, 40,000$ (last quarter of test set)		
$\max(\lvert e_i \rvert)$	0.0081	0.0037
$\text{mean}(\lvert e_i \rvert)$	3.72×10^{-4}	2.31×10^{-4}
$\text{RMSE}(e_i)$	5.86×10^{-4}	3.34×10^{-4}

Table 4

Captions of Figures

Figure 1:

Evolution of $CV(M_0)$ for System 1 over different orders of the system. The minimum $CV(M_0)$ is obtained with the regression vector $\boldsymbol{z}(t) = [y(t-1), y(t-2), u(t), u(t-1)]$.

Figure 2:

Estimated nonlinear component for System 1 using the kernel-based partially linear model. The true nonlinear function $g = (0.5y(t-1))^3$ (dotted line) is compared with the estimated $\hat{g}$ (dashed line). Training points ('o') are also indicated.

Figure 3:

Input and Output signals for the Silverbox data set, showing the zones for training, validation and testing.

Figure 4:

Simulation error obtained using a full black-box nonlinear model in the Silverbox study case.

Figure 5:

Simulation error obtained using a partially linear model for the Silverbox study case.

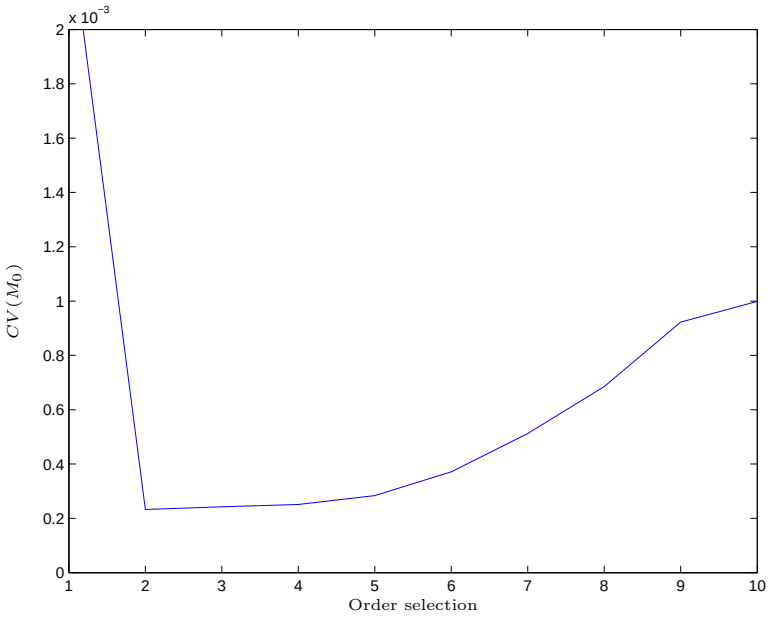


Figure 1:

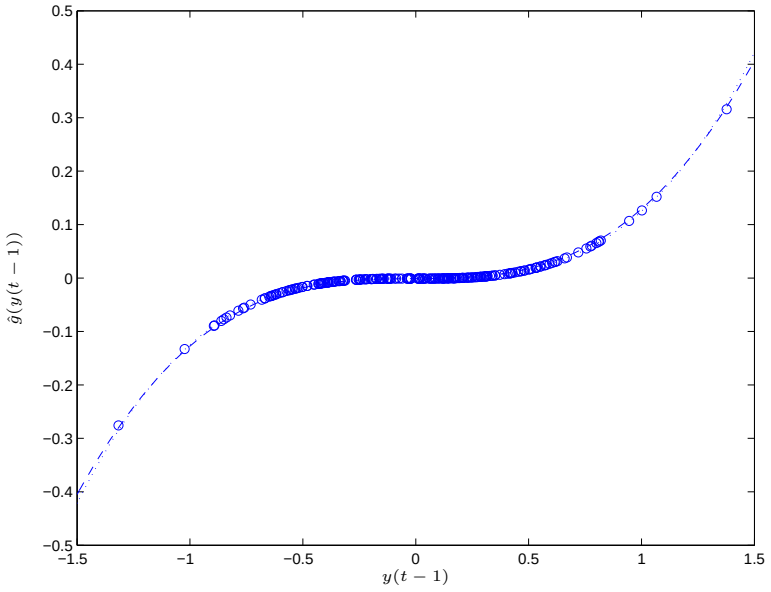


Figure 2:

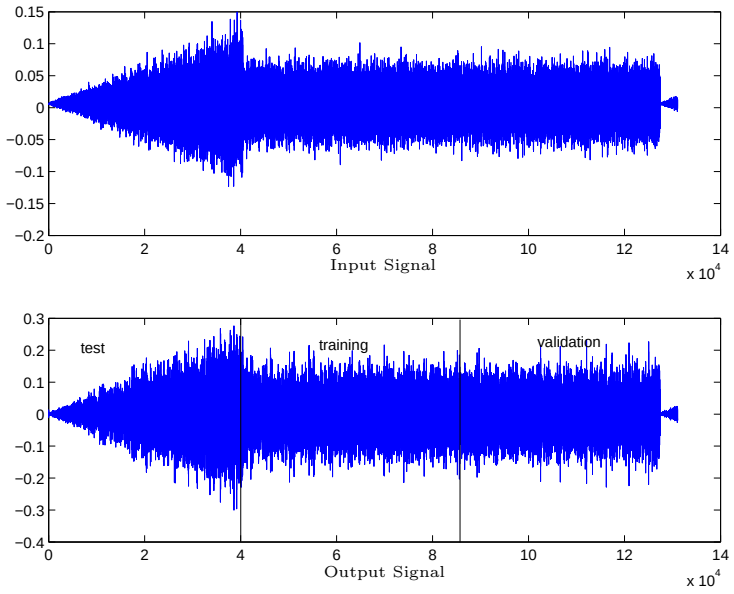


Figure 3:

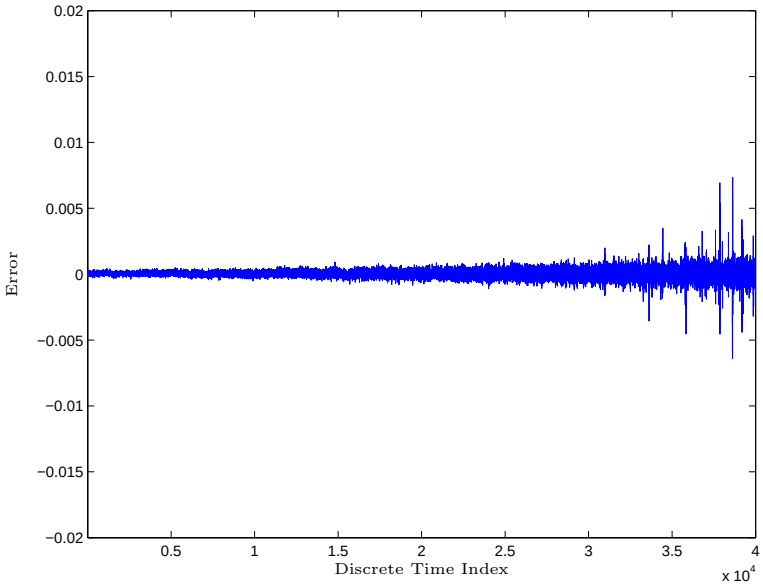


Figure 4:

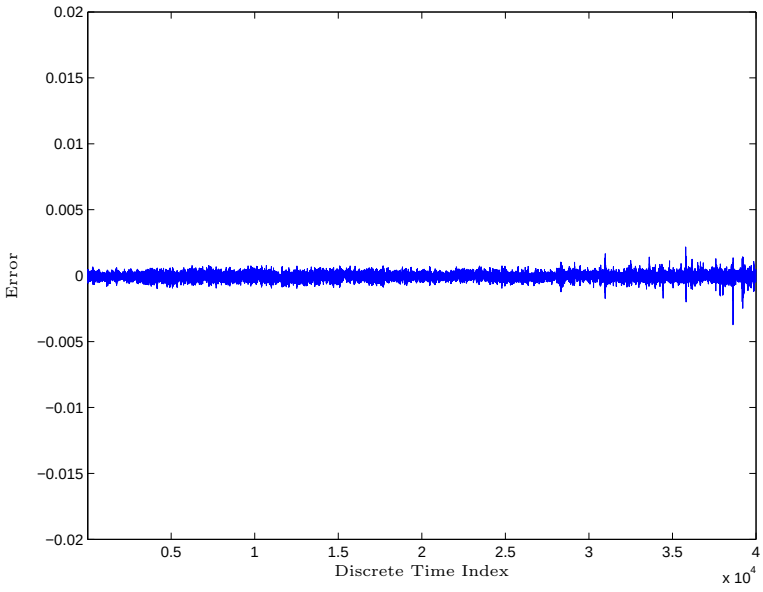

0.02
0.015
0.01
0.005
0
−0.005
−0.01
−0.015
−0.02
Error
0.5
1
1.5
2
2.5
3
3.5
4
Discrete Time Index
x 10^4

Figure 5: