Switched and PieceWise Nonlinear Hybrid System Identification

Fabien Lauer, Gérard Bloch

Switched and PieceWise Nonlinear Hybrid System Identification

Fabien Lauer and Gérard Bloch

Centre de Recherche en Automatique de Nancy (CRAN UMR 7039),
Nancy–University, CNRS, France,
fabien.lauer@esstin.uhp-nancy.fr, gerard.bloch@esstin.uhp-nancy.fr

Abstract. Hybrid system identification aims at both estimating the discrete state or mode for each data point, and the submodel governing the dynamics of the continuous state for each mode. The paper proposes a new method based on kernel regression and Support Vector Machines (SVM) to tackle this problem. The resulting algorithm is able to compute both the discrete state and the submodels in a single step, independently of the discrete state sequence that generated the data. In addition to previous works, nonlinear submodels are also considered, thus extending the class of systems on which the method can be applied from PieceWise Affine (PWA) and switched linear to PieceWise Smooth (PWS) and switched nonlinear systems with unknown nonlinearities. Piecewise systems with nonlinear boundaries between the modes are also considered with some preliminary results on this issue.

1 Introduction

Context. Hybrid systems are usually described by both a continuous state and a discrete state, where the vector field defining the evolution of the continuous state depends on the discrete state. In this framework, a system can be seen as switching between n different subsystems, which are usually modeled by AutoRegressive with eXogenuous inputs (ARX) models in the discrete-time case. Two types of identification problems may arise in this setting depending on whether the discrete state sequence that generated the data is known or not. If it is, then the problem can be simply recast as n common identification problems, each one using only the data for a given discrete state. However, in most cases this sequence is unknown and the problem becomes nontrivial.

Models of hybrid systems. The predicted output y_t of a hybrid model in ARX form is given as a function of the continuous state $\boldsymbol{x}_t = [u_{t-n_c} \ \dots \ u_{t-1}, y_{t-n_a} \ \dots \ y_{t-1}]^T$, containing the lagged inputs u_{t-j} and outputs y_{t-j}, and the discrete state λ_t. Considering n submodels f_j, the hybrid model is written as

$$y_t = f_{\lambda_t}(\boldsymbol{x}_t) + e_t, \tag{1}$$

where e_t is a noise term. Hybrid models can be classified with respect to the nature of the submodels f_j and of the evolution of the discrete state $\lambda_t \in \{1, \dots, n\}$.

Table 1. Nomenclature of the hybrid models in ARX form

ARX model	abbr.	models f_j	discrete state λ_t	domains S_j
PieceWise	PWARX	affine	function of $\boldsymbol{x}$	polyhedral
PieceWise Nonlinear	PWNARX	nonlinear	function of $\boldsymbol{x}$	polyhedral
Nonlinearly PieceWise	NPWARX	affine	function of $\boldsymbol{x}$	arbitrary
Nonlinearly PieceWise Nonlinear	NPWNARX	nonlinear	function of $\boldsymbol{x}$	arbitrary
Switched	SARX	affine	arbitrary	
Switched Nonlinear	SNARX	nonlinear	arbitrary	

Table 1 defines the nomenclature that will be used in the paper. SARX and SNARX models assume that the system is arbitrarily switched. On the other hand, PWARX models consider a dependency between the discrete state and the continuous state. They can thus be defined by PieceWise Affine (PWA) maps of the type $f(\boldsymbol{x}) = f_j(\boldsymbol{x})$, if $\boldsymbol{x} \in S_j = \{\boldsymbol{x} \ : \ \boldsymbol{H}_j[\boldsymbol{x}^T \ 1]^T \leq 0\}$, $j = 1, \ldots, n$, where the matrices $\boldsymbol{H}_j$ represent a set of hyperplanes that define the polyhedral domains S_j partitioning the continuous state space. Similarly, PWNARX models can be defined by PieceWise Smooth (PWS) maps, where f_j are smooth nonlinear functions instead of affine functions. Extensions of the PWARX and PWNARX models to "nonlinearly piecewise" models, where the domains S_j are no more constrained to be polyhedral, will be denoted NPWARX and NPWNARX.

Related work. Five main approaches have been devised for hybrid system identification: the clustering-based approach [1], the mixed integer programming based approach [2], the Bayesian approach [3], the bounded error approach [4] and the algebraic approach [5, 6]. The first four focus on the problem of PieceWise Affine (PWA) system identification, where the discrete state depends on the continuous state. However, both the bounded error and Bayesian approaches can also be used to identify a broader class of systems, known as switched linear systems, where the discrete state evolves independently of the continuous state. The algebraic approach [5] focuses on this latter problem, but without taking the noise into account in its development. This leads to an algorithm very sensitive to noise, compared to the clustering-based or bounded error methods, as shown by [7]. Besides, the bounded error method [4] provides a convenient way of dealing with noisy data by looking for a model with a predefined accuracy. However, the hyperparameters of the method, such as the model accuracy that determines the number of modes, may be difficult to tune to get a prescribed structure, e.g. if prior knowledge on the number of modes is available [7].

Tools and proposed method. The paper proposes a new method for hybrid system identification based on kernel regression and Support Vector Machines (SVMs). Stemming from statistical learning theory, Support Vector Machines (SVMs) [8] quickly became a state-of-the-art tool for classification and are already com-

monly used, either in their original form or through closely related methods [9], in hybrid system identification to estimate the switching boundary between modes [1, 4]. Based on the same theoretical concepts, Support Vector Regression (SVR) retains properties of SVMs, such as a good generalization ability from few samples, and offers an interesting alternative both for regression and system identification [10–12]. SVR uses an ε-insensitive loss function which does not take into account errors that are less than ε [13]. This loss function ignoring errors below a predefined threshold is close in spirit to the bounded error approach. However, the origin is different. In learning theory, this effect is justified in order to minimize the generalization error of the model, whereas the bounded error approach was developed to allow the automatic determination of the number of linear submodels required to approximate a non-linear function with a given accuracy.

In the past decade, kernel methods have attracted much attention in a large variety of fields and applications: classification and pattern recognition, regression, density estimation, etc. Indeed, using kernel functions, many linear methods can be extended to the nonlinear case in an almost straightforward manner, while avoiding the curse of dimensionality by transposing the focus from the data dimension to the number of data.

The proposed method uses the SVR framework to estimate hybrid models with submodels in kernel expansion form. The resulting algorithm is able to compute both the discrete state and the submodels in a single step, independently of the discrete state sequence that generated the data. Nonlinear submodels with unknown types of nonlinearities can be easily treated, thus extending the class of systems on which the method can be applied from switched linear to switched nonlinear systems and from piecewise affine to piecewise smooth systems by considering models in SARX, SNARX, PWARX or PWNARX form. Nonlinearly PWA and PWS maps with nonlinear mode boundaries are also considered in the paper with some preliminary results using nonlinear SVM classifiers. The idea is that since the method estimates the discrete state without any assumption on the switching sequence, labeling data points generated from nonlinearly separable modes is possible.

Contribution. The paper proposes solutions for two problems that have not yet been extensively studied and solved in the literature: identification of hybrid systems switching between unknown nonlinear dynamics and identification of nonlinearly piecewise systems with nonlinear boundaries between the modes in the continuous state space.

Paper organization. The paper starts by some preliminaries on kernel functions and Support Vector Regression (Sect. 2.1) before using these to develop a hybrid system identification algorithm in Sect. 2.2. The problem of estimating nonlinear boundaries between modes is then discussed in Sect. 2.3 and Section 3 provides an interpretation of the method based on previous approaches from the literature. Finally, Section 4 gives some numerical examples of application.

Notations. All vectors are column vectors written in boldface and lowercase letters whereas matrices are boldface and uppercase. The vectors $\mathbf{0}$ and $\mathbf{1}$ are vectors of appropriate dimensions with all their components respectively equal to 0 and 1. For $\boldsymbol{A} \in \mathbb{R}^{d\times m}$ and $\boldsymbol{B} \in \mathbb{R}^{d\times n}$ containing d-dimensional sample vectors and the kernel function $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$, the "kernel" $\boldsymbol{K}(\boldsymbol{A}, \boldsymbol{B})$ maps $\mathbb{R}^{d\times m} \times \mathbb{R}^{d\times n}$ in $\mathbb{R}^{m\times n}$ with $\boldsymbol{K}(\boldsymbol{A}, \boldsymbol{B})_{i,j} = k(\boldsymbol{A}_i, \boldsymbol{B}_j)$, where $\boldsymbol{A}_i$ and $\boldsymbol{B}_j$ are the ith and jth columns of $\boldsymbol{A}$ and $\boldsymbol{B}$. In particular, if $\boldsymbol{x} \in \mathbb{R}^d$ is a column vector then $\boldsymbol{K}(\boldsymbol{x}, \boldsymbol{B})$ is a row vector in $\mathbb{R}^{1\times n}$. The matrix $\boldsymbol{X} \in \mathbb{R}^{N\times d}$ contains all the training samples $\boldsymbol{x}_i$, $i = 1, \ldots, N$, as rows. The vector $\boldsymbol{y} \in \mathbb{R}^N$ gathers all the target values y_i for these samples. The kernel matrix $\boldsymbol{K}(\boldsymbol{X}^T, \boldsymbol{X}^T)$ will be written $\boldsymbol{K}$ for short.

2 Nonlinear Hybrid System Identification

This section presents a new method based on kernel regression and support vector machines (SVMs) for hybrid system identification. The basics of nonlinear function approximation by kernel methods are first recalled, before describing the proposed method itself. The section ends with a discussion on piecewise systems with nonlinear boundaries between modes.

2.1 Kernels and Support Vector Regression

A simple method to approximate a nonlinear function is to first map the data to a higher dimensional feature space and then perform linear regression in that space. This approach usually suffers from the so-called *curse of dimensionality*, which can however be avoided thanks to the "kernel trick" depicted below.

First, consider the nonlinear mapping Φ that maps the data $\boldsymbol{x}$ from the input space $\mathcal{X} \subset \mathbb{R}^p$ to a vector $\boldsymbol{\Phi}(\boldsymbol{x})$ in a feature space $\mathcal{F}$. Assume now that the function f is given by an expansion based on the N training samples $\boldsymbol{x}_i \in \mathbb{R}^p$ in that feature space, i.e. $f(\boldsymbol{x}) = \sum_{i=1}^N \alpha_i \boldsymbol{\Phi}(\boldsymbol{x})^T \boldsymbol{\Phi}(\boldsymbol{x}_i) + b$. Clearly, though being a nonlinear function in the input space, f is a linear function in $\mathcal{F}$. Note that in order to compute $f(\boldsymbol{x})$, it is not necessary to explicitly compute the images $\boldsymbol{\Phi}(\boldsymbol{x}_i)$ of the points but only the result of their inner product. This is the "kernel trick" which replaces the inner products between images of points by a *kernel function* $k(\boldsymbol{x}, \boldsymbol{x}_i) = \boldsymbol{\Phi}(\boldsymbol{x})^T \boldsymbol{\Phi}(\boldsymbol{x}_i)$. In kernel regression, the training data $(\boldsymbol{x}_i, y_i)$, $i = 1, \ldots, N$, stacked in the matrix $\boldsymbol{X}$ and the vector $\boldsymbol{y}$, are thus approximated by a kernel expansion

$$f(\boldsymbol{x}) = \sum_{i=1}^{N} \alpha_i k(\boldsymbol{x}, \boldsymbol{x}_i) + b = \boldsymbol{K}(\boldsymbol{x}, \boldsymbol{X}^T)\boldsymbol{\alpha} + b \ , \tag{2}$$

where $\boldsymbol{\alpha} = [\alpha_1 \ \ldots \ \alpha_i \ \ldots \ \alpha_N]^T$ and b are the parameters of the model and $k(.,.)$ is the kernel function. Typical kernel functions are the linear ($k(\boldsymbol{x}, \boldsymbol{x}_i) = \boldsymbol{x}^T \boldsymbol{x}_i$), Gaussian RBF ($k(\boldsymbol{x}, \boldsymbol{x}_i) = \exp(-\|\boldsymbol{x} - \boldsymbol{x}_i\|_2^2 / 2\sigma^2)$ and polynomial ($k(\boldsymbol{x}, \boldsymbol{x}_i) = (\boldsymbol{x}^T \boldsymbol{x}_i + 1)^d$) kernels. The kernel function defines the feature space $\mathcal{F}$ in which

the data are implicitly mapped. The higher the dimension of $\mathcal{F}$ is, the higher the approximation capacity of the function f is, up to the universal approximation capacity obtained for an infinite feature space, as with Gaussian RBF kernels. It is also possible to build kernel functions from prior knowledge on the task at hand, see for instance [14] for the properties of kernel functions and the construction of new kernels or [15] for examples of application in pattern recognition. In the hybrid system identification framework, this can be useful for instance when the type of nonlinearity of a particular mode is known beforehand.

In kernel regression via linear programming (LP), the ℓ_1-norm of the parameters $\boldsymbol{\alpha}$ of the kernel expansion is minimized together with the ℓ_1-norm of the errors $y_i - f(\boldsymbol{x}_i)$ by

$$\min_{(\boldsymbol{\alpha},b)} \|\boldsymbol{\alpha}\|_1 + C\sum_{i=1}^{N} |f(\boldsymbol{x}_i) - y_i| \ , \tag{3}$$

where a hyperparameter C is introduced to tune the trade-off between the minimization of the model complexity (measured by $\|\boldsymbol{\alpha}\|_1$) and the error on the data (measured by $\sum_{i=1}^{N} |f(\boldsymbol{x}_i) - y_i|$). Minimizing the complexity of the model allows to control its generalization capacity. In practice, this amounts to penalize non-smooth functions and implements the general smoothness assumption that two samples close in input space tend to give the same output.

Instead of the ℓ_1-norm of the errors, the ε-insensitive loss function, defined by [8] as

$$l(e) = |e|_\varepsilon = \begin{cases} 0 & \text{if } |e| \leq \varepsilon \ , \\ |e| - \varepsilon & \text{otherwise} \ , \end{cases} \tag{4}$$

can also be used to yield Linear Programming Support Vector Regression (LP-SVR). This loss function builds a tube of insensitivity in which the errors are meaningless. Errors larger than the tube width [1] ε are penalized linearly.

A possible formulation of the LP-SVR problem involves $4N + 1$ design variables [16]. In the remaining of the paper, we will follow the approach of [17] that involves only $3N + 1$ variables. Introducing two sets of optimization variables, in two positive slack vectors $\boldsymbol{a}$ and $\boldsymbol{\xi}$, this problem can be implemented as a linear program solvable by standard optimization softwares such as the MATLAB *linprog* function. In this scheme, the LP-SVR problem may be written as

$$\begin{aligned} \min_{(\boldsymbol{\alpha},b,\boldsymbol{\xi}\geq\mathbf{0},\boldsymbol{a}\geq\mathbf{0})} \quad & \mathbf{1}^T\boldsymbol{a} + C\mathbf{1}^T\boldsymbol{\xi} \\ \text{s.t.} \quad -\boldsymbol{\xi} - \varepsilon\mathbf{1} \leq \ & \boldsymbol{K}\boldsymbol{\alpha} + b\mathbf{1} - \boldsymbol{y} \leq \varepsilon\mathbf{1} + \boldsymbol{\xi} \\ -\boldsymbol{a} \leq \ & \boldsymbol{\alpha} \leq \boldsymbol{a} \ . \end{aligned} \tag{5}$$

The last set of constraints ensures that $\mathbf{1}^T\boldsymbol{a}$, which is minimized, bounds $\|\boldsymbol{\alpha}\|_1$. In practice, sparsity is obtained as a certain number of parameters α_i will tend to zero. The input vectors $\boldsymbol{x}_i$ for which the corresponding α_i are non-zero are called *support vectors*.

[1] Actually, ε does not stand for the tube width but for half of the tube section with respect to y.

2.2 Hybrid System Identification with Kernels

The bounded error approach, developed by [4] for the identification PWARX models, aims at finding a model with a predefined accuracy, i.e. that allows the error on all the training points $(\boldsymbol{x}_i, y_i)$ to be bounded by

$$|y_i - f_{\lambda_i}(\boldsymbol{x}_i)| = |e_i| \leq \delta,\ i = 1, \ldots, N\ . \tag{6}$$

The following presents a new method based on kernel regression to achieve this goal. As a direct benefit, nonlinear submodels f_j are easily handled by the choice of the kernel functions, thus providing a method for the estimation of both piecewise and switched nonlinear ARX models.

Following the SVR approach, submodels in kernel expansion form

$$f_j(\boldsymbol{x}) = \sum_{i=1}^{N} \alpha_{ij} k_j(\boldsymbol{x}, \boldsymbol{x}_i) + b_j = \boldsymbol{K}_j(\boldsymbol{x}, \boldsymbol{X}^T)\boldsymbol{\alpha}_j - b_j, \tag{7}$$

are trained by minimizing the ℓ_1-norm of the parameters $\boldsymbol{\alpha}_j$. As indicated by the subscript j, various kernel functions k_j can be associated to the different models f_j. This leads to vectors $\boldsymbol{K}_j(\boldsymbol{x}, \boldsymbol{X}^T)$ and kernel matrices $\boldsymbol{K}_j = \boldsymbol{K}_j(\boldsymbol{X}^T, \boldsymbol{X}^T)$, as defined in the notations at end of the introduction. It is thus possible to take prior information into account such as the number of modes governed by linear dynamics or knowledge on the type of a particular nonlinearity. In this setting, the problem of training n models under the bounded error constraint may be written as

$$\begin{aligned} \min_{\boldsymbol{\alpha}_j, b_j, \boldsymbol{a}_j \geq \boldsymbol{0}} \ & \sum_{j=1}^{n} \boldsymbol{1}^T \boldsymbol{a}_j \\ & -\delta\boldsymbol{1} \leq \boldsymbol{y} - \boldsymbol{K}_j\boldsymbol{\alpha}_j - b_j\boldsymbol{1} \leq \delta\boldsymbol{1},\ \forall \boldsymbol{x}_i \in S_j,\ j = 1, \ldots, n, \\ & -\boldsymbol{a}_j \leq \boldsymbol{\alpha}_j \leq \boldsymbol{a}_j,\ j = 1, \ldots, n\ . \end{aligned} \tag{8}$$

where $\boldsymbol{y} = [y_1\ y_2\ \ldots\ y_N]^T$ and the absolute error $|e_{ij}| = |f_j(\boldsymbol{x}_i) - y_i|$ is constrained to be less than δ only for the model j corresponding to the discrete state λ_i of the point $\boldsymbol{x}_i$. However, without further information on the classification of the data into modes, S_j are unknown and the problem is intractable. To circumvent this issue, consider the equivalent problem using the ε-insensitive loss function (4) for $\varepsilon = \delta$ implemented with slack variables ξ_{ij}, $i = 1, \ldots, N$, $j = 1, \ldots, n$, stacked in the n vectors $\boldsymbol{\xi}_j \in \mathbb{R}^N$ as in (5):

$$\begin{aligned} \min_{\boldsymbol{\alpha}_j, b_j, \boldsymbol{a}_j \geq \boldsymbol{0}, \boldsymbol{\xi}_j \geq \boldsymbol{0}} & \sum_{j=1}^{n} \boldsymbol{1}^T \boldsymbol{a}_j \\ & -\boldsymbol{\xi}_j - \delta\boldsymbol{1} \leq \boldsymbol{y} - \boldsymbol{K}_j\boldsymbol{\alpha}_j - b_j\boldsymbol{1} \leq \delta\boldsymbol{1} + \boldsymbol{\xi}_j,\ j = 1, \ldots, n, \\ & -\boldsymbol{a}_j \leq \boldsymbol{\alpha}_j \leq \boldsymbol{a}_j,\ j = 1, \ldots, n, \\ & \prod_{j=1}^{n} \xi_{ij} = 0,\ i = 1, \ldots, N\ . \end{aligned} \tag{9}$$

The last equalities stand for the fact that all points must be estimated with accuracy δ by at least one submodel f_j. In other words, for a given sample $(\boldsymbol{x}_i, y_i)$, there is at least one j for which $\xi_{ij} = 0$. As nonlinear equalities are not easy to deal with from an optimization point of view, these are approximated by

$$\begin{aligned} \min_{\boldsymbol{\alpha}_j, b_j, \boldsymbol{a}_j \geq \mathbf{0}, \boldsymbol{\xi}_j \geq \mathbf{0}} & \sum_{j=1}^{n} \mathbf{1}^T \boldsymbol{a}_j + C \sum_{i=1}^{N} \prod_{j=1}^{n} \xi_{ij} \\ -\boldsymbol{\xi}_j - \delta \mathbf{1} \leq \boldsymbol{y} - \boldsymbol{K}_j \boldsymbol{\alpha}_j - b_j \mathbf{1} \leq \delta \mathbf{1} + \boldsymbol{\xi}_j, & \ j = 1, \ldots, n, \\ -\boldsymbol{a}_j \leq \boldsymbol{\alpha}_j \leq \boldsymbol{a}_j, & \ j = 1, \ldots, n \ . \end{aligned} \tag{10}$$

Solving this problem with a sufficiently large constant C leads to functions f_j solutions of the former problem (8). Moreover, the discrete state λ_i, in which the system was for each data point $\boldsymbol{x}_i$ is readily available from the variables ξ_{ij} vanishing to zero as $\hat{\lambda}_i = j$, for $\xi_{ij} = 0$. The cases where the bounded error constraint is not satisfied, i.e. no ξ_{ij} is zero, can be further discriminated by letting $\hat{\lambda}_i = \arg\min_j(\xi_{ij})$. On the other hand, for cases where more than one ξ_{ij} is zero, the absolute error is considered and $\hat{\lambda}_i = \arg\min_j |e_{ij}|$, with $e_{ij} = y_i - f_j(\boldsymbol{x}_i)$.

In the case of PWARX or PWNARX models where the modes are linearly separable in the continuous state space, undetermined points can be reclassified after the training of separating hyperplanes (the boundaries between the sets S_j) based on the determined cases only. The linear classification issue is not discussed here due to size constraints and the reader is referred to [8] and [9] for an introduction to state-of-the-art methods, whereas multi-class pattern recognition is considered for instance by [18]. In the next Section, extensions of the PWARX and PWNARX models to nonlinearly piecewise models, where the domains S_j are no more constrained to be polyhedral, will be discussed.

An advantage of the proposed approach is the possibility to deal easily with a noise level that also switches with the model. In order to do so, multiple loss functions with different parameters δ_j for each mode can be used and implemented in the following final problem

$$\begin{aligned} \min_{\boldsymbol{\alpha}_j, b_j, \boldsymbol{a}_j \geq \mathbf{0}, \boldsymbol{\xi}_j \geq \mathbf{0}} & \sum_{j=1}^{n} \mathbf{1}^T \boldsymbol{a}_j + C \sum_{i=1}^{N} \prod_{j=1}^{n} \xi_{ij} \\ -\boldsymbol{\xi}_j - \delta_j \mathbf{1} \leq \boldsymbol{y} - \boldsymbol{K}_j \boldsymbol{\alpha}_j - b_j \mathbf{1} \leq \delta_j \mathbf{1} + \boldsymbol{\xi}_j, & \ j = 1, \ldots, n, \\ -\boldsymbol{a}_j \leq \boldsymbol{\alpha}_j \leq \boldsymbol{a}_j, & \ j = 1, \ldots, n \ . \end{aligned} \tag{11}$$

Another possible formulation with $n \times N$ less variables and constraints involves the minimization of the squares of the parameters α_{ij} as

$$\begin{aligned} \min_{\boldsymbol{\alpha}_j, b_j, \boldsymbol{\xi}_j \geq \mathbf{0}} & \sum_{j=1}^{n} \boldsymbol{\alpha}_j^T \boldsymbol{\alpha}_j + C \sum_{i=1}^{N} \prod_{j=1}^{n} \xi_{ij} \\ -\boldsymbol{\xi}_j - \delta_j \mathbf{1} \leq \boldsymbol{y} - \boldsymbol{K}_j \boldsymbol{\alpha}_j - b_j \mathbf{1} \leq \delta_j \mathbf{1} + \boldsymbol{\xi}_j, & \ j = 1, \ldots, n \ . \end{aligned} \tag{12}$$

The solution of this problem is not sparse as the one of (11) but can usually be computed in less time.

Remark 1. In the case of a linear kernel $k_j(\boldsymbol{x}, \boldsymbol{x}_i) = \boldsymbol{x}^T \boldsymbol{x}_i$, the parameters of the linear model $f_j(\boldsymbol{x}) = \boldsymbol{w}_j^T \boldsymbol{x} + b_j$ can be explicitly recovered by $\boldsymbol{w}_j = \boldsymbol{X}^T \boldsymbol{\alpha}_j$.

Remark 2. The hyperparameters of the method are the kernel types, the number n of modes, the bounds δ_j, the regularization parameter C and the number of lagged inputs and outputs (dynamic order). They can be tuned on a subset of the data put aside for validation. When too few data are available, cross-validation techniques can be used. Moreover, the algorithms (11) and (12) can be extended to automatically tune the bounds δ_j to the noise level by using a trick similar to the one introduced in ν-SVR [16, 17]. This is studied in [19] for linear submodels and directly applicable to the problems above. Besides, the proposed method is well adapted when some basic prior knowledge on the system is available such as the number n of modes. However, due to the universal approximation capability of kernel models, the tuning of n is less crucial than when using linear or affine submodels. For piecewise maps, a good fit can be obtained with an underestimated n.

Remark 3. Problems (11) and (12) are linearly constrained nonlinear programs. They involve the minimization of a criterion composed of a linear (11) or quadratic (12) term and a product of nonnegative variables subject to linear constraints. These problems are not convex and have multiple minima. This can be seen from their symmetric structure, leading to multiple solutions for simple permutations of models. All these solutions are acceptable and yield the same objective function value corresponding to a global optimum. However, care must be taken when using different kernels for different models, in which case permuting models is no more without effect and may lead to local minima.

A possible initialization of the optimization can be obtained by solving the feasibility problems corresponding to the constraints of (11) and (12), which are simple linear programs.

2.3 Nonlinear Boundaries Between Modes

A direct extension of the PWARX and PWNARX models, in which the discrete state is determined by a set of separating hyperplanes in the continuous state space, is obtained by introducing nonlinear boundaries or arbitrary regions (also pointed out in the conclusion of [4]). In these "nonlinearly piecewise" models (denoted by NPWARX and NPWNARX, see Table 1 in the introduction), the discrete state is still a function of the continuous state, but the separating surfaces are no more restricted to hyperplanes. This can lead to a decrease of the number of submodels if the true system corresponds to this description. Indeed, in this case, the linear separability assumption may require to build multiple identical submodels for different regions of the continuous state space that are however governed by the same dynamics. Moreover, regrouping the data available in several regions of the continuous state space into one submodel may help to get better estimates for regions with few samples.

Nonlinear classification methods have to be used in this case and are readily available in number (KPCA, KFD, SVM...) [14] thanks to the kernel trick used above. In particular, SVMs are similarly very easily extended to nonlinear classification by an appropriate choice of kernel function. Moreover, the final classifier is given as a sparse kernel expansion allowing for relatively fast estimation of the mode for a new sample. For the binary case (only 2 modes), the nonlinear separating surface $\mathcal{S}$ is given by

$$h(\boldsymbol{x}) = \sum_{i=1}^{N} \beta_i k_c(\boldsymbol{x}, \boldsymbol{x}_i) + b_c = 0 \ , \tag{13}$$

where $k_c(.,.)$ is a kernel function and the β_i, b_c are the trainable parameters of the classifier. Simply taking the sign of the function h yields the class of a pattern $\boldsymbol{x}$, i.e. $+1$ if $h(\boldsymbol{x}) \geq 0$ and -1 otherwise.

The method proposed in Sect. 2 can deal with nonlinearly piecewise maps (themselves either affine or nonlinear) without any modification and provide the labeling of the data, required to train a classifier, through $\hat{\lambda}_i$. Indeed, any method (including the bounded-error, Bayesian or algebraic approaches) that estimates the discrete state without dependency on the continuous state, and thus without any assumption regarding the linear separability of the data in the continuous state space, can deal with nonlinearly piecewise maps. In practice, the procedure is as follows.

1. Train a hybrid model on the input-output data $(\boldsymbol{x}_i, y_i)$, $i = 1, \ldots, N$, by solving (11) or (12),
2. Estimate the discrete states, e.g. by $\hat{\lambda}_i = \arg\min_j |e_{ij}|$, $i = 1, \ldots, N$,
3. Train a classifier on the labeled data $(\boldsymbol{x}_i, \hat{\lambda}_i)$, $i = 1, \ldots, N$.

Additionally, in a refinement step, the training points could be re-assigned to the different modes by the classifier and the submodels f_j retrained one by one on the relevant data only.

An illustrative example of this procedure is given in Sect. 4.2, where the method is applied to estimate a Nonlinearly PieceWise Affine (NPWA) map.

3 Interpretation and Links with other Approaches

The proposed method can be interpreted as a bridge between the bounded error approach [4], that can deal easily with noise, and the algebraic procedure [5], that can deal with arbitrarily switched systems, while providing nonlinear extensions to these. More precisely, it amounts to a bounded error relaxation of the hybrid decoupling constraint used in the algebraic procedure as follows.

The hybrid decoupling constraint of the algebraic procedure can be expressed as a function of the submodel errors, $e_{ij} = y_i - f_j(\boldsymbol{x}_i)$, by

$$\prod_{j=1}^{n} e_{ij} = 0, \ i = 1, \ldots, N. \tag{14}$$

These constraints account for the fact that there must be at least one of the submodels f_j that can estimate the ith point with zero error. In the case of noisy data, these constraints cannot be satisfied for all the N points. Considering the bounded error approach, "decoupling" can be however enforced. The bounded error constraints (6) act similarly to (14), though being less restrictive on the estimation error (with a threshold δ). Combining these two approaches results in constraints of the form

$$\prod_{j=1}^{n} [|e_{ij}| \geq \delta] = 0, \ i = 1, \ldots, N \ , \tag{15}$$

where $[\cdot] = 1$, if the bracketed expression is true, and 0 otherwise. Using these constraints, the absolute value of the error $|e_i| = \min_j |e_{ij}|$ (assuming that $\hat{\lambda}_i = \arg\min_j |e_{ij}|$) of the hybrid model is bounded by the threshold δ. Approximating $[|e_{ij}| \geq \delta]$ for all j by an ε-insensitive loss function and minimizing their product leads to the algorithm (11).

4 Numerical Examples

The following presents three examples of application. The first one shows the simultaneous estimation of two functions, one linear and one nonlinear, from datasets overlapping in the input space, while the second one is concerned with the estimation of a nonlinearly piecewise affine (NPWA) map. The last example shows the identification of a SNARX model of a hybrid system arbitrarily switching between linear and nonlinear dynamics. For examples 1 and 3, the discrete state is arbitrarily switched and the type of nonlinearity is unknown. In all the examples, the problems are formulated as the optimization program (11) and solved by the MATLAB function *fmincon*.

4.1 Switching Function with Unknown Nonlinearity

In this one-dimensional example, the data are generated by two models: a linear submodel $y_1(x) = ax + b + e = 2x - 1 + e$ and a polynomial submodel $y_2(x) = 0.5x^2 + e$, where e is a zero-mean Gaussian noise of standard deviation 0.5. The discrete state λ, determining which submodel is active, is independent of the variable x. Two data points, $(x, y_1(x))$ and $(x, y_2(x))$, are generated for 30 values of x in the interval $]-5, 1[$. Beside these 60 data, the only prior knowledge is that one submodel is linear and the other is nonlinear. The aim of this example is to show that the proposed method can discriminate between the two submodels and correctly approximate each one without further knowledge on the type of nonlinearity. Figure 1 shows the results obtained for $\delta_1 = \delta_2 = 0.5$, $C = 100$, a linear kernel k_1 and a Gaussian RBF kernel k_2 with $\sigma = 2$. The estimated parameters for the linear submodel are $\hat{a} = 1.999$ and $\hat{b} = -0.863$. The overall Mean Square Error (MSE) is $\text{MSE} = \frac{1}{N} \sum_{i=1}^{N} (y_i - f_{\hat{\lambda}_i}(x_i))^2 = 0.205$, which is rather good compared to the noise variance $\sigma_b^2 = 0.25$.

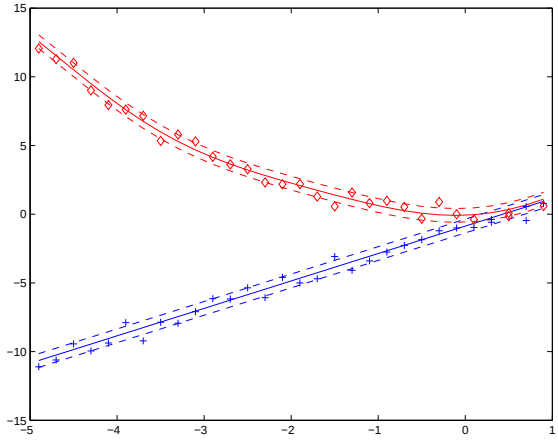

Fig. 1. One-dimensional example of the simultaneous estimation of two functions on noisy data. Points associated to the linear and RBF models are respectively represented by crosses (+) and diamonds (◇).

4.2 Nonlinearly Piecewise Affine Map Estimation

In this illustrative example, the problem is to estimate a Nonlinearly PieceWise Affine (NPWA) map defined as $y = x + 0.5 + e$, for $x \in]-\infty, -1] \bigcup [1, \infty[$, and $y = -0.5x - 1 + e$, for $x \in]-1, 1[$, where e is a zero-mean Gaussian noise of standard deviation 0.1. This problem could be solved by considering a PWA map with 3 modes linearly separable in the x variable, or, as proposed in this example, by considering only 2 modes with a nonlinear boundary between mode 1 and mode 2. Thus, two models with linear kernels are trained on $N = 60$ data points for $\delta_1 = \delta_2 = 0.1$ and $C = 100$. The resulting models, shown on Fig. 2, are $y = 1.01x + 0.53$ and $y = -0.48x - 1.02$. All the points are associated to the correct model except for one point, $(x_i, y_i) = (-0.9, -0.31)$, close to the mode boundary. The training of a SVM classifier, with a polynomial kernel $k_c = (xx_i + 1)^3$, on the data labeled by $\hat{\lambda}_i,\ i = 1, \ldots, N$, yields a nonlinear boundary $\mathcal{S}$ between the modes given by 2 support vectors $x_1 = -3$ and $x_{60} = 2.9$. As the data x_i are in $\mathbb{R}$, this nonlinear separating surface is a set of points defined as $\mathcal{S} = \{x\ :\ h(x) = -0.24(-3x + 1)^3 - 0.22(2.9x + 1)^3 + 10.3 = 0\}$, as shown on the right hand side of Fig. 2. This classifier yields no classification error with respect to the target labels $\hat{\lambda}_i$.

4.3 Simulated Hybrid System Identification

Consider the hybrid system switching between mode 1: $y_t = -0.905y_{t-1} + 0.9u_{t-1} + e_t$, and mode 2: $y_t = -0.4y_{t-1}^2 + 0.5u_{t-1} + e_t$, where e_t is a zero-mean Gaussian noise of standard deviation $\sigma_b = 0.1$. An output trajectory of $N = 100$ points of this system is generated with a random initial condition

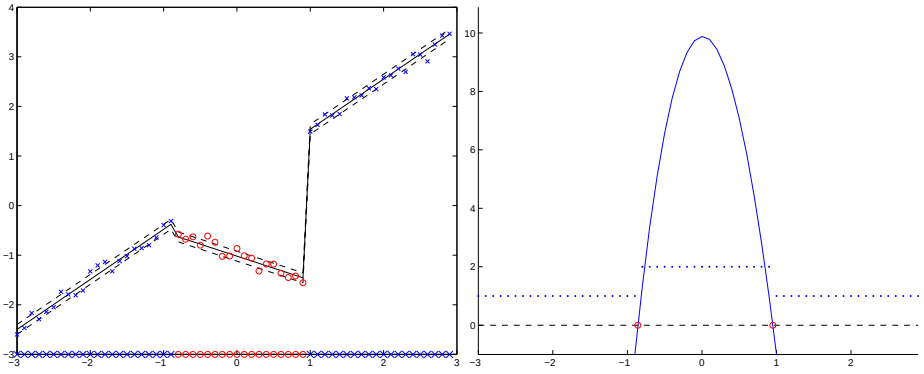

Fig. 2. Nonlinear boundary between 2 modes. *Left*: Hybrid model (—) with its insensitivity tube (- -) approximating the data points represented by '×' for $\hat{\lambda}_i = 1$ and "○' for $\hat{\lambda}_i = 2$. The estimated mode $\hat{\lambda}_i$ also appears on the x-axis as '×' for mode 1 and '○' for mode 2 to highlight the partition of the input space $\mathcal{X}$. *Right*: Class labels $\hat{\lambda}_i$ (.) for each data point x_i used to learn the nonlinear boundary $\mathcal{S}$ (○), defined as the zeros of $h(x)$ (—).

y_0, a random input sequence u_t uniformly distributed in the interval $[0, 1]$ and a mode switch from mode 1 to mode 2 at time $t = 41$. The signal-to-noise ratio of this trajectory is 12 dB corresponding to a variance of the noise free trajectory of 0.20 and a noise variance of 0.012. These data are then used to train a SNARX model with $C = 1000$, $\delta_1 = \delta_2 = 0.1$, a linear kernel k_1 and a RBF kernel k_2 with $\sigma = 1$. Thus, the only prior knowledge is that one submodel is linear and the other is nonlinear. The trajectory of the resulting model $\hat{y}_t = f_{\hat{\lambda}_t}(\hat{y}_{t-1}, u_{t-1})$ is shown on Figure 3. The estimated parameters of the linear mode 1 are -0.929 and 0.960, to be compared to -0.905 and 0.9. The discrete state is estimated by $\hat{\lambda}_t = \arg\min_j(\xi_{tj})$. As shown at the bottom of Fig. 3, 22 classification errors occur on the whole trajectory. The effect of these errors is limited and their origin can be explained. Most of them occur on ambiguous points for which $f_1(y_{t-1}, u_{t-1}) = f_2(y_{t-1}, u_{t-1}) \pm (\delta_1 + \delta_2)$. Here, a switched system is identified, but note that in case of a piecewise system, these ambiguities could be removed by classifying the points with respect to a separating boundary in the continuous state space. The overall simulation error is RMSEsim $= \sqrt{1/N \sum_{t=1}^{N} (y_t - \hat{y}_t)^2} = 0.154$, which is slightly more than the noise standard deviation $\sigma_b = 0.1$. Only 8 support vectors with nonzero α_{ij} are selected from the 100 training samples to build the kernel expansion f_2.

5 Conclusion

In this paper, a new system identification method has been proposed to deal with nonlinear hybrid systems. In particular, this method is applicable to sys-

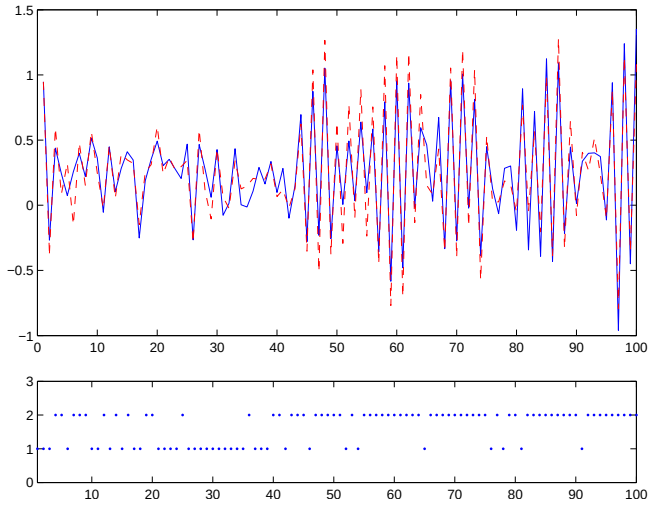

Fig. 3. Simulated hybrid system identification. *Top*: Trajectory of the system (blue plain line) and of the model (red dash line) in simulation mode (only the initial condition and the input is given to the model). *Bottom*: Estimated discrete state $\hat{\lambda}_t$.

tems switching between unknown nonlinear dynamics and nonlinearly piecewise systems with arbitrary nonlinear boundaries between the modes. It also bridges the gap between the bounded error approach and the algebraic procedure by making use of the ε-insensitive loss function proposed in the machine learning community for Support Vector Regression. Since no assumption on the discrete sequence that generates the data is required, arbitrarily switched systems can be treated as well as piecewise systems.

Future work will focus on the tuning of the hyperparameters and optimization issues as well as experiments with real life applications. Among other perspectives, the simultaneous estimation of the submodels and the boundaries between the modes for piecewise systems could be investigated.

References

1. Ferrari-Trecate, G., Muselli, M., Liberati, D., Morari, M.: A clustering technique for the identification of piecewise affine systems. Automatica **39**(2) (2003) 205–217
2. Roll, J., Bemporad, A., Ljung, L.: Identification of piecewise affine systems via mixed-integer programming. Automatica **40**(1) (2004) 37–50
3. Juloski, A.L., Weiland, S., Heemels, W.: A Bayesian approach to identification of hybrid systems. IEEE Trans. on Automatic Control **50**(10) (2005) 1520–1533
4. Bemporad, A., Garulli, A., Paoletti, S., Vicino, A.: A bounded-error approach to piecewise affine system identification. IEEE Trans. on Automatic Control **50**(10) (2005) 1567–1580

5. Vidal, R., Soatto, S., Ma, Y., Sastry, S.: An algebraic geometric approach to the identification of a class of linear hybrid systems. In: Proc. of the 42nd IEEE Conf. on Decision and Control, Maui, Hawaï, USA. (2003) 167–172
6. Ma, Y., Vidal, R.: Identification of deterministic switched ARX systems via identification of algebraic varieties. In Morari, M., Thiele, L., eds.: Proc. of the 8th Int. Conf. on Hybrid Systems: Computation and Control. Volume 3414 of LNCS. (2005) 449–465
7. Juloski, A., Heemels, W., Ferrari-Trecate, G., Vidal, R., Paoletti, S., Niessen, J.H.G.: Comparison of four procedures for the identification of hybrid systems. In: Proc. of the Int. Conf. on Hybrid Systems: Computation and Control, Zurich, Switzerland. Volume 3414 of LNCS., Springer (2005) 354–369
8. Vapnik, V.N.: The nature of statistical learning theory. Springer-Verlag, New York, NY, USA (1995)
9. Mangasarian, O.: Generalized support vector machines. In Smola, A., Bartlett, P., Schölkopf, B., Schuurmans, D., eds.: Advances in Large Margin Classifiers. MIT Press, Cambridge, MA, USA (2000) 135–146
10. Drezet, P., Harrison, R.: Support vector machines for system identification. Proc. of the UKACC Int. Conf. on Control, Swansea, UK **1** (1998) 688–692
11. Mattera, D., Haykin, S.: Support vector machines for dynamic reconstruction of a chaotic system. In Schölkopf, B., Burges, C.J., Smola, A.J., eds.: Advances in kernel methods: support vector learning. MIT Press, Cambridge, MA, USA (1999) 211–241
12. Zhang, L., Xi, Y.: Nonlinear system identification based on an improved support vector regression estimator. In: Proc. of the Int. Symp. on Neural Networks, Dalian, China. Volume 3173 of LNCS., Springer (2004) 586–591
13. Smola, A.J., Schölkopf, B.: A tutorial on support vector regression. Statistics and Computing **14**(3) (2004) 199–222
14. Shawe-Taylor, J., Cristianini, N.: Kernel Methods for Pattern Analysis. Cambridge University Press (2004)
15. Lauer, F., Bloch, G.: Incorporating prior knowledge in support vector machines for classification: a review. Neurocomputing (2007)
16. Smola, A.J., Schölkopf, B., Rätsch, G.: Linear programs for automatic accuracy control in regression. In: Proc. of the 9th Int. Conf. on Artificial Neural Networks, Edinburgh, UK. Volume 2. (1999) 575–580
17. Mangasarian, O.L., Musicant, D.R.: Large scale kernel regression via linear programming. Machine Learning **46**(1-3) (2002) 255–269
18. Crammer, K., Singer, Y.: On the algorithmic implementation of multiclass kernel-based vector machines. Journal of Machine Learning Research **2** (2001) 265–292
19. Lauer, F., Bloch, G.: A new hybrid system identification algorithm with automatic tuning. In: Proc. of the 17th IFAC World Congress, Seoul, Korea. (2008)