Nonlinear Hybrid System Identification with Kernel Models

Fabien Lauer, Gérard Bloch, René Vidal

Nonlinear Hybrid System Identification with Kernel Models

Fabien Lauer, Gérard Bloch and René Vidal

***Abstract*— This paper focuses on the identification of nonlinear hybrid systems involving unknown nonlinear dynamics. The proposed method extends the framework of [1] by introducing nonparametric models based on kernel functions in order to estimate arbitrary nonlinearities without prior knowledge. In comparison to the previous work of [2], which also dealt with unknown nonlinearities, the new algorithm assumes the form of an unconstrained nonlinear continuous optimization problem, which can be efficiently solved for moderate numbers of parameters in the model, as is typically the case for linear hybrid systems. However, to maintain the efficiency of the method on large data sets with nonlinear kernel models, a preprocessing step is required in order to fix the model size and limit the number of optimization variables. A support vector selection procedure, based on a maximum entropy criterion, is proposed to perform this step. The efficiency of the resulting algorithm is demonstrated on large-scale experiments involving the identification of nonlinear switched dynamical systems.**

I. INTRODUCTION

By using tools from machine learning, such as kernel functions, this paper proposes an algorithm for the identification of nonlinear hybrid systems involving arbitrary and unknown nonlinearities.

Consider a class of discrete-time ARX hybrid systems of the form

$$y_i = f_{\lambda_i}(\boldsymbol{x}_i) + e_i, \tag{1}$$

where $\boldsymbol{x}_i = [y_{i-1} \ \dots \ y_{i-n_a}, \ u_{i-n_k} \ \dots \ u_{i-n_k-n_c+1}]^T$ is the *continuous state* (or regression vector) of dimension p containing the lagged n_c inputs u_{i-k} and n_a outputs y_{i-k}, $\lambda_i \in \{1, \dots, n\}$ is the *discrete state* (or mode) determining which one of the n subsystems $\{f_j\}_{j=1}^n$ is active at time step i, and e_i is an additive noise term.

This class of hybrid models can be classified with respect to the nature of the submodels $\{f_j\}$ and that of the evolution of the discrete state λ_i. According to the nomenclature defined in [2], Switched ARX (SARX) and Switched Nonlinear ARX (SNARX) models assume that the system switches arbitrarily. On the other hand, PieceWise ARX (PWARX) models consider a dependency between the discrete state and the regression vector. They can thus be defined by piecewise affine maps of the type $f(\boldsymbol{x}) = f_j(\boldsymbol{x})$, if $\boldsymbol{x} \in S_j, \ j = 1, \dots, n$, where $\{f_j\}$ are affine functions and $\{S_j\}$ are polyhedral domains defining a partition of the regression space. Similarly, PWNARX models can be defined by piecewise smooth maps, where $\{f_j\}$ are smooth nonlinear functions instead of affine functions. Piecewise models with arbitrary domains $\{S_j\}$ are considered in [2] and referred to as *Nonlinearly* PWARX or PWNARX (NPWARX or NPWNARX) models.

This paper concentrates on the problem of finding a nonlinear hybrid model $f = \{f_j\}_{j=1}^n$ of the form (1) from input–output data $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$. We assume that the number of models n and their regressors are known and focus on the identification of SNARX models. However, the proposed estimators are able to deal with all the piecewise forms described above without any modification. These estimators provide SNARX models, which can be used to estimate the discrete state. With the labels of the points thus obtained, determining the partition of the regression space simply amounts to a pattern recognition (or supervised classification) problem, for which efficient algorithms are available [3].

Related work. The hybrid system identification problem intrinsically implies to simultaneously classify the samples into their respective modes and estimate the model parameters for each mode. As such, it assumes a straightforward discrete optimization formulation, which is highly non-convex and scales exponentially with the number of data.

Many of the recent approaches proposed to solve this problem typically implement a local optimization method and are thus rather sensitive to their initialization. The clustering-based approaches, either using k-means [4] or Expectation Maximization (EM) [5], the Bayesian approach [6], [7] and the Support Vector Regression (SVR) approach [8], [9] fall into this category. On the other hand, the mixed integer programming (MIP) approach [10] and the bounded-error approach [11] are based on combinatorial optimization and become prohibitively time consuming for moderate-size data sets. Beside these methods, the algebraic approach [12], [13], [14], [15] circumvents the aforementioned computational issues by proposing a closed form solution to an approximation of the identification problem for SARX systems. However, this approach can be sensitive to noise. Another approach based on continuous optimization has been recently proposed in [1]. In addition to being robust to noise and outliers, this approach also significantly alleviates the complexity bottleneck of other methods.

To the best of our knowledge, the first approach to deal with *nonlinear* hybrid system identification without prior knowledge of the nonlinearities has been proposed in [2] as an extension to the SVR-based method [8]. However, this approach optimizes over a number of variables that grows with the number of data points, and is thus limited to small data sets. In comparison to the method of [2], the algorithm

F. Lauer is with the LORIA, Université Henri Poincaré Nancy 1, France fabien.lauer@loria.fr

G. Bloch is with the Centre de Recherche en Automatique de Nancy (CRAN UMR 7039), Nancy–University, CNRS, France gerard.bloch@esstin.uhp-nancy.fr

R. Vidal is with the Center for Imaging Science, Department of Biomedical Engineering, Johns Hopkins University, USA rvidal@jhu.edu

proposed here builds on the work of [1] and is able to deal with large data sets.

Paper contribution. In this paper, we extend the *continuous optimization* framework originally proposed in [1] for linear hybrid system identification to nonlinear hybrid systems. By continuous optimization we refer to the optimization of a continuous cost function over a continuous domain, which excludes for instance integer programs. In particular, two formulations of the algorithm are considered. The first one is based on a non-differentiable cost function involving min operations. The second one, inspired by the algebraic approach [12], offers a differentiable approximation using products of error terms. Nonlinear model based on kernel functions are introduced in these algorithms to be able to estimate unknown nonlinearities. As the efficiency of the method of [1] heavily relies on the number of optimization variables, we propose *fixed-size* kernel submodels. As a consequence, the resulting unconstrained optimization program, though non-convex, can be solved by standard global optimization algorithms even for large data sets.

Paper organization. Section §II presents the proposed continuous optimization framework for hybrid system identification with the two estimators based on the minimum of errors (§II-A) and product of errors (§II-B) terms, respectively. Kernel methods for nonlinear hybrid system identification are introduced in §III. The paper ends with numerical experiments in §IV and conclusions in §V.

Notations. For the sake of clarity, the notation $\underset{\theta}{\text{minimize}}\ J$ is used to refer to an optimization problem minimizing a cost function J over some variable θ, whereas $\min_{j=1,\ldots,n} L_j$ refers to the function returning the minimum of some finite set of values $\{L_1, \ldots, L_n\}$.

II. HYBRID SYSTEM IDENTIFICATION FRAMEWORK

This section presents the framework of the proposed algorithms and recalls the form of the estimators for hybrid systems derived in [1], the minimum-of-errors estimator and the product-of-errors estimator, in §II-A and in §II-B, respectively. In order to remain efficient on large data sets, these estimators are devised so as to lead to continuous optimization programs with a number of variables which does not depend on the number of data.

A. Minimum-of-Errors Estimator

The Minimum-of-Errors (ME) estimator assumes that sample $\boldsymbol{x}_i$ must be assigned to the submodel that best estimates the target output y_i with respect to a given loss function l, i.e.,

$$\hat{\lambda}_i = \arg \min_{j=1,\ldots,n} l(y_i - f_j(\boldsymbol{x}_i)),\ i = 1, \ldots, N. \quad (2)$$

The error minimization framework estimates the model f as the one that minimizes, on a given data set, the error

$$J = \frac{1}{N} \sum_{i=1}^{N} l(y_i - f(\boldsymbol{x}_i)). \quad (3)$$

Explicitly including (2) in this framework leads to the *Minimum-of-Errors* (ME) estimator as obtained by solving

$$\underset{\{f_j\}}{\text{minimize}}\ \ J^{ME}, \quad (4)$$

$$\text{where}\ \ J^{ME} = \frac{1}{N} \sum_{i=1}^{N} \left(\min_{j=1,\ldots,n} l(y_i - f_j(\boldsymbol{x}_i)) \right). \quad (5)$$

Note that the minimum of continuous functions of some variables is a continuous functions of these variables (discontinuities only occur in the derivatives). It follows that for submodels $\{f_j\}$ given by continuous functions of their parameters and any loss function $l(e)$ continuous in its argument, the minimum over j of the loss functions $l(y_i - f_j(\boldsymbol{x}_i))$ is a continuous function of the parameters to estimate. As a consequence, the cost function in (5) is a continuous function of the variables parametrizing the submodels $\{f_j\}$. Thus (4) is an unconstrained continuous optimization problem only involving real variables, which are the parameters of the submodels $\{f_j\}$. After solving for these parameters, the mode estimates are simply recovered by using (2) (or $\hat{\lambda}_i = \arg\min_{j=1,\ldots,n} |y_i - f_j(\boldsymbol{x}_i)|$, if the loss function l cannot yield the decision).

An equivalent estimator to the one presented above can be derived in the maximum likelihood framework, as shown in [1].

B. Product-of-Errors Estimator

For a smooth loss function l, the *Product-of-Errors* (PE) estimator is obtained by solving the smooth optimization program

$$\underset{\{f_j\}}{\text{minimize}}\ \ J^{PE}, \quad (6)$$

$$\text{where}\ \ J^{PE} = \frac{1}{N} \sum_{i=1}^{N} \prod_{j=1}^{n} l(y_i - f_j(\boldsymbol{x}_i)). \quad (7)$$

The cost function (7) of the PE estimator can be seen as a smooth approximation to the ME cost function (5). In particular, for noiseless data, they share the same global minimum $J^{ME} = J^{PE} = 0$. Note that for linear submodels f_j, solving the optimization problem of the PE estimator in the noiseless case gives an exact solution to the identification problem, as shown in [12].

III. ESTIMATION OF NONLINEAR HYBRID MODELS

In this section, we extend the framework to the identification of hybrid systems involving unknown nonlinear dynamics.

A. Kernel Models for Hybrid Systems

Following the Linear Programming Support Vector Regression (LP-SVR) approach [16], nonlinear submodels are expressed in the kernel expansion form

$$f_j(\boldsymbol{x}) = \sum_{k=1}^{N} \alpha_{kj} k_j(\boldsymbol{x}_k, \boldsymbol{x}) + b_j, \quad (8)$$

where $\boldsymbol{\alpha}_j = [\alpha_{1j}, \ldots, \alpha_{Nj}]^T$ and b_j are the parameters of the submodel f_j and $k_j(\cdot,\cdot)$ is a kernel function satisfying Mercer's condition. Typical kernel functions are the linear ($k(\boldsymbol{x}_k, \boldsymbol{x}) = \boldsymbol{x}_k^T \boldsymbol{x}$), Gaussian Radial Basis Function (RBF) ($k(\boldsymbol{x}_k, \boldsymbol{x}) = \exp(-\|\boldsymbol{x}_k - \boldsymbol{x}\|_2^2/2\sigma^2)$ and polynomial ($k(\boldsymbol{x}_k, \boldsymbol{x}) = (\boldsymbol{x}_k^T \boldsymbol{x} + 1)^d$) kernels. A kernel function implicitly computes inner products, $k(\boldsymbol{x}_k, \boldsymbol{x}) = \langle \boldsymbol{\Phi}(\boldsymbol{x}_k), \boldsymbol{\Phi}(\boldsymbol{x}) \rangle$, between points in a higher-dimensional feature space $\mathcal{F}$ obtained by an hidden nonlinear mapping $\Phi : \boldsymbol{x} \mapsto \boldsymbol{\Phi}(\boldsymbol{x})$. The higher the dimension of $\mathcal{F}$ is, the higher the approximation capacity of the model is, up to the universal approximation capacity obtained for an infinite feature space, as with Gaussian RBF kernels. With (8), different kernel functions k_j can be used for the different submodels f_j. It is thus possible to take prior knowledge into account such as the number of modes governed by linear dynamics or information on the type of a particular nonlinearity, if available. Note, however, that this is not a requirement for the proposed method.

As in Support Vector Machines (SVMs) [17], we refer to the vectors $\boldsymbol{x}_k$ for which the associated $\{\alpha_{kj}\}_{j=1,\ldots,n}$ parameters are nonzero as the *Support Vectors* (SVs), since these are the only data points kept in the final model. SVM methods are typically known to yield sparse models in terms of these SVs, which allows faster computations of the output.

1) Regularization: In order to avoid overfitting, the control of the complexity (or flexibility) of the model is a crucial issue when estimating nonlinear kernel models. This control can be achieved by minimizing a regularized cost as in

$$\underset{f}{\text{minimize}}\ \mathcal{R}(\boldsymbol{\alpha}) + C\mathcal{J}(f, \mathcal{D}), \tag{9}$$

where $\mathcal{R}(\boldsymbol{\alpha})$ is a regularization term acting on the model parameters $\boldsymbol{\alpha} = [\boldsymbol{\alpha}_1^T, \ldots, \boldsymbol{\alpha}_n^T]^T$ and $\mathcal{J}(f, \mathcal{D})$ is the data term measuring the error of the model f on the dataset $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1,\ldots,N}$.

In the following, we consider regularization of the model f through the regularization of the submodels f_j and define an overall regularizer as

$$\mathcal{R}(\boldsymbol{\alpha}) = \frac{1}{n} \sum_{j=1}^{n} R(\boldsymbol{\alpha}_j), \tag{10}$$

where $R(\boldsymbol{\alpha}_j)$ is the regularizer for the submodel f_j.

In standard LP-SVR, the model complexity is measured by the L_1-norm of the parameter vector, i.e.,

$$R(\boldsymbol{\alpha}_j) = \|\boldsymbol{\alpha}_j\|_1, \tag{11}$$

In practice, minimizing $\|\boldsymbol{\alpha}_j\|_1$ amounts to penalizing non-smooth functions and ensures sparsity as a certain number of parameters α_{ij} will tend towards zero. Regularization over the L_2-norm of the parameter vectors, i.e.,

$$R(\boldsymbol{\alpha}_j) = \|\boldsymbol{\alpha}_j\|_2^2 = \boldsymbol{\alpha}_j^T \boldsymbol{\alpha}_j, \tag{12}$$

is also possible, but may result in less sparse models.

2) Nonlinear ME estimator: By using submodels in kernel form (8) in the ME estimator (5), the algorithm for nonlinear hybrid system identification becomes

$$\underset{\{\boldsymbol{\alpha}_j\},\{b_j\}}{\text{minimize}}\ \frac{1}{n} \sum_{j=1}^{n} R(\boldsymbol{\alpha}_j) + \frac{C}{N} \sum_{i=1}^{N} \min_{j=1,\ldots,n} l\left(y_i - \sum_{k=1}^{N} \alpha_{kj} k_j(\boldsymbol{x}_k, \boldsymbol{x}_i) - b_j \right). \tag{13}$$

3) Nonlinear PE estimator: Similarly, one can define the nonlinear PE estimator as the solution to

$$\underset{\{\boldsymbol{\alpha}_j\},\{b_j\}}{\text{minimize}}\ \frac{1}{n} \sum_{j=1}^{n} R(\boldsymbol{\alpha}_j) + \frac{C}{N} \sum_{i=1}^{N} \prod_{j=1}^{n} l\left(y_i - \sum_{k=1}^{N} \alpha_{kj} k_j(\boldsymbol{x}_k, \boldsymbol{x}_i) - b_j \right). \tag{14}$$

B. Fixed-Size Kernel Models for Large-Scale Problems

For submodels in kernel form (8), the optimization programs (13) and (14) involve a large number of variables associated to the number of potential SVs. Since the kernel submodels consider all the data points $\boldsymbol{x}_k$, $k = 1, \ldots, N$, as potential SVs, the number of variables α_{kj} is $n \times N$. Thus solving this problem for large N with a global optimizer may become prohibitively time consuming. Here the key to reducing the number of parameters α_{kj} is to select the support vectors $\boldsymbol{x}_k$ *before* starting the optimization.

1) Selection of Support Vectors (SVs): The fixed-size Least Squares SVM (LS-SVM) [18] is a particular implementation of SVMs, in which the SVs are selected before minimizing a regularized least squares criterion. This method is based on the maximization of an entropy criterion to ensure a sufficient coverage of the feature space by the SVs. Then the selected SVs are used to build an approximation of the nonlinear mapping Φ hidden in the kernel function, which is in turn used to recast the problem into a linear form in the approximated feature space. However, in our experiments, this method was rather sensitive to the numbers of selected SVs. Therefore, we will apply a similar but more straightforward method for Gaussian RBF kernels, where we do not build an approximation of the nonlinear mapping, but instead use the SVs as RBF centers directly. This leads to reduced submodels

$$f_j(\boldsymbol{x}) = \sum_{k=1}^{M_j} \alpha_{i_{kj}j} k_j(\boldsymbol{x}_{i_{kj}}, \boldsymbol{x}) + b_j, \tag{15}$$

where M_j is the number of SVs $\boldsymbol{x}_{i_{kj}}$ and $\{i_{kj}\}_{k=1,\ldots,M_j}$ is the list of indexes of the SVs retained for the jth submodel. Note that the parameter vector of submodel f_j is now given by $\boldsymbol{\alpha}_j = [\alpha_{i_{1j}j}, \ldots, \alpha_{i_{M_j j}j}]^T$ and of dimension M_j.

As in fixed-size LS-SVM, the selection algorithm maximizes the quadratic Rényi entropy H_R, which quantifies the diversity, uncertainty or randomness of a system. We approximate H_R by

$$H_R \approx -\log \frac{1}{M_j^2} \mathbf{1}^T \boldsymbol{K}_j^{M_j} \mathbf{1}, \tag{16}$$

where

$$\boldsymbol{K}_j^{M_j} = \begin{bmatrix} k_j(\boldsymbol{x}_1, \boldsymbol{x}_{i_{1j}}) & \dots & k_j(\boldsymbol{x}_1, \boldsymbol{x}_{i_{M_j j}}) \\ \vdots & \ddots & \\ k_j(\boldsymbol{x}_N, \boldsymbol{x}_{i_{1j}}) & & k_j(\boldsymbol{x}_N, \boldsymbol{x}_{i_{M_j j}}) \end{bmatrix},$$

is the kernel matrix for the jth mode. Following [18], the procedure to select the SVs for a particular mode j is as follows.

1) Randomly select M_j SVs from the training samples $\boldsymbol{x}_i,\ i = 1, \dots, N$.
2) Randomly select one of the M_j SVs, $\boldsymbol{x}^\star$, and one of the remaining training samples, $\boldsymbol{x}^\dagger$.
3) Replace $\boldsymbol{x}^\star$ by $\boldsymbol{x}^\dagger$ in the set of SVs.
4) If the criterion (16) increases, retain $\boldsymbol{x}^\dagger$ as a SV, otherwise replace $\boldsymbol{x}^\dagger$ by $\boldsymbol{x}^\star$ in the set of SVs.
5) Repeat from 2 until the increase of the criterion is too small or a maximum number of iterations is reached.

Note that in this procedure, a data point $\boldsymbol{x}_i$ originally generated by a particular mode can be considered as a SV for another mode. The main idea here is only to capture the general distribution of the data in feature space to ensure sufficient support of the model. However, for piecewise models, where a particular submodel is only active in a given region of input space, this procedure may be suboptimal as it also selects SVs outside of this region. In this case, how to obtain sparser representations should be investigated.

2) Complete estimation procedure: A *fixed-size* nonlinear hybrid model is estimated as follows.

1) Select n sets of SVs of indexes $\{i_{kj}\}_{k=1,\dots,M_j}$, with sizes M_1, …, M_n, by applying the procedure of Sect. III-B.1 to maximize the criterion (16).
2) Train the hybrid model by solving (9), e.g., for the PE estimator with L_2-regularization,

$$\underset{\{\boldsymbol{\alpha}_j\},\{b_j\}}{\text{minimize}} \ \frac{1}{n}\sum_{j=1}^{n}\frac{\boldsymbol{\alpha}_j^T\boldsymbol{\alpha}_j}{M_j} + \frac{C}{N}\sum_{i=1}^{N}\prod_{j=1}^{n} l(y_i - f_j(\boldsymbol{x}_i)), \tag{17}$$

where $f_j(\boldsymbol{x}_i)$ is computed by (15).

The final optimization program (17) involves only $\sum_{j=1}^{n}(M_j+1)$ variables instead of $n(N+1)$ as in (14).

In this procedure, the numbers of SVs $\{M_j\}_{j=1,\dots,n}$ are the hyperparameters that must be fixed *a priori* and may influence the quality of the model. In standard SVR or neural network problems, such hyperparameters may be tuned on the basis of an estimate of the generalization error, which is either obtained on out-of-sample validation data or by a cross-validation procedure. However, here, these estimates of the generalization error cannot be obtained without knowledge of the discrete state λ to choose with which submodel f_j the output should be computed. Therefore, instead of tuning the numbers M_j, we consider the following heuristics for Gaussian RBF kernels of width parameter σ_j:

$$M_j = \left\lfloor \frac{1}{\sigma_j} \max_{k=1,\dots,p} \left(\max_{i=1,\dots,N} x_{ik} - \min_{i=1,\dots,N} x_{ik} \right) \right\rfloor, \tag{18}$$

where $\lfloor\cdot\rfloor$ denotes the integer part of its argument and x_{ik} is the kth component of $\boldsymbol{x}_i$. This heuristics is not optimal, but ensures sufficient support of the model over the whole input space. Moreover, notice that we only need suboptimal numbers M_j that lead to rough mode estimates rather than a perfect fit of the data. Then, it is always possible to re-estimate the submodels separately on the basis of this data classification. If this re-estimation is performed by standard SVR, then the number of SVs is automatically determined. This will be illustrated in the experiments of Sect. IV-A.

IV. NUMERICAL EXPERIMENTS

This section starts by presenting an illustrative example involving the estimation of a function switching between two unknown nonlinear functions (Sect. IV-A). Large-scale experiments demonstrating the identification of a nonlinear switched system are described in Sect. IV-B.

As proposed in [1], all optimization programs are solved with the Multilevel Coordinate Search algorithm[1] [19]. Though the MCS algorithm can deal with unbounded variables, box constraints are used to limit the search space and restrain the variables to the interval $[-100, 100]$ (which is not very restrictive).

The quality of the models is evaluated on an independent test set by the Mean Squared Error, $\text{MSE} = 1/N_t \sum_{i=1}^{N_t}(y_i - f_{\lambda_i}(\boldsymbol{x}_i))^2$, where N_t is the number of test samples.

A. Illustrative Example

Consider the function arbitrarily switching between two nonlinear behaviors as

$$y(x) = \begin{cases} x^2, & \text{if } \lambda = 1 \\ \sin(3x) + 2, & \text{if } \lambda = 2. \end{cases} \tag{19}$$

A training set of $N = 2000$ points is generated by this function with additive Gaussian noise ($\sigma_v = 0.3$). Figure 1 shows the normalized data with zero mean and unit variance (black dots). The procedure proposed in section III-B.2 with the PE estimator is used to estimate two submodels, f_1 and f_2, which use RBF kernels with $\sigma_1 = 0.8$ and $\sigma_2 = 0.2$, respectively. The difference between σ_1 and σ_2 reflects the basic assumption that one of the two models should be smoother than the other. The SVs are first selected, with the numbers $M_1 = 4$ and $M_2 = 17$ set as in (18). Then, Eq. (17) is solved with $C = 100$. Finally, the resulting submodels (top plot) are used to cluster the data and standard SVR is applied to re-estimate the submodels separately (bottom plot). Table I shows that the heuristics (18) leads to almost optimal M_j.

[1]The software is freely available as Matlab code at `http://www.mat.univie.ac.at/~neum/software/mcs/`.

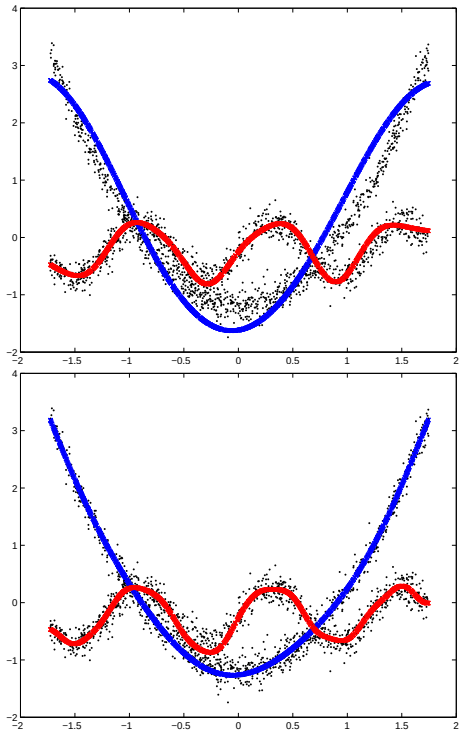

Fig. 1. Simultaneous estimation (top) and separate SVR re-estimation (bottom) of a switched nonlinear function from 2000 noisy samples (black dots).

TABLE I

TEST MSE ($\times 10^{-3}$) (TOP) AND CLASSIFICATION ERROR RATE (BOTTOM) OF THE REFINED HYBRID MODEL FOR DIFFERENT NUMBERS OF SVS M_1 AND M_2. THE NUMBERS FOR THE PROPOSED HEURISTICS (MARKED WITH ASTERISKS) ARE ALMOST OPTIMAL.

M_2 / M_1	16	17*	18	19
3	42.34	44.56	40.33	36.56
	15.33%	15.33%	15.13%	14.73%
4*	10.96	10.72*	**10.67**	10.83
	9.20%	**8.73%***	9.13%	9.13%
5	17.51	40.11	17.04	39.34
	11.07%	15.00%	11.20%	15.00%

B. Switched Nonlinear Dynamical System

Consider the dynamical system arbitrarily switching between two modes as

$$y_i = \begin{cases} 0.9y_{i-1} + 0.2y_{i-2}, & \text{if } \lambda_i = 1 \\ (0.8 - 0.5\exp(-y_{i-1}^2))y_{i-1} - \\ \quad (0.3 + 0.9\exp(-y_{i-1}^2))y_{i-2} + & \text{if } \lambda_i = 2, \\ \quad 0.4\sin(2\pi y_{i-1}) + 0.4\sin(2\pi y_{i-2}), \end{cases} \tag{20}$$

Training sets of various sizes N are generated by this system for the initial condition $y_0 = y_{-1} = 0.1$ and additive Gaussian noise ($\sigma_v = 0.1$), leading to trajectories with $\sigma_y \approx 0.8$. The test set of 2000 points is built from noiseless data starting at the initial condition $y_0 = 0.4,\ y_{-1} = -0.3$. To be able to evaluate the quality of the results, reference models are computed with full knowledge of the discrete state by two separate trainings of standard SVR (one for each mode).

For the hybrid models estimated by the PE estimator, the submodel f_1 uses a linear kernel with an arbitrary number of SVs $M_1 = 5$ (this is a fictive number, as the two linear parameters can be recovered from linear combinations of the SVs), while f_2 uses a Gaussian RBF kernel ($\sigma = 0.3$) with M_2 set as in (18). Both the SVR re-estimation and the reference models (described below) use the same kernel hyperparameters and the same regularization trade-off $C = 100$ as the PE estimator. Standard SVR is applied for re-estimation and the reference models with the ε-insensitive loss function parameter ε, which acts as a threshold on the minimal error taken into account, set to 0.1. Note that in the re-estimation procedure, all these hyperparameters could be tuned by cross-validation on the basis of the data classification previously obtained.

Table II shows the number of SVs for f_2, the test MSE, the classification error rate and the computing time of the hybrid model obtained by the PE estimator and the re-estimated SVR models. In this Table, all numbers of the form $A \pm B$ correspond to averages and standard deviations over 100 trials. The Table also shows the values for the reference models. In order to estimate the number of undecidable data, the classification error of the reference model is also computed from the mode estimates (2). Note that the number of SVs of the SVR models highly depends on the threshold ε used in the ε-insensitive loss function. In addition, any other nonlinear estimator can be used instead for the nonlinear mode, while linear system identification methods can be applied to the linear mode.

A number of remarks can be stated from these results. First, the PE estimator can accurately estimate the mode, leading to a classification error of 13 % on average if we discard the undecidable points. Moreover, this classification provides the ground for the re-estimation procedure, which yields submodels with better test errors than standard SVR using knowledge of the discrete state. This can be explained by the fact that the data is classified w.r.t. the minimum submodel error. Thus, some data points of one mode corrupted by a large amount of noise may be assigned to the other mode, for which the noise level converts into a small value. Finally, the computing time of the PE estimator is also quite reasonable: the model can for instance be estimated from thousands of data in seconds and from 50 000 data in less than 3 minutes. Note that we cannot observe a linear dependency between the computing time and the number of data N as in [1] for linear submodels, since the number of SVs M_2, on which depends the computing time of the PE estimator, also changes with N (due to (18)). However, the difference between the PE and SVR computing times decreases with N. This shows that for large N, though relying on non-convex global optimization, the PE estimator can be faster than a convex optimization based method using specifically tailored and compiled code (LibSVM [20]).

TABLE II

IDENTIFICATION OF A HYBRID SYSTEM WITH UNKNOWN NONLINEARITIES BY THE PE ESTIMATOR WITH AND WITHOUT SVR RE-ESTIMATION. THE REFERENCE MODEL IS OBTAINED WITH KNOWLEDGE OF THE MODE. THE COMPUTING TIME OF PE+SVR ONLY ACCOUNTS FOR THE SVR STEP.

N	Method	M_2	Test MSE ($\times 10^{-3}$)	Classif. err. (%)	Time (sec.)
2 000	PE	18 ± 2	210.43 ± 48.69	25.55 ± 2.83	2.9 ± 1.2
	PE + SVR	242 ± 26	49.99 ± 23.26	19.00 ± 3.00	0.6 ± 0.0
	Reference	373 ± 16	102.15 ± 34.38	12.51 ± 0.96	0.6 ± 0.0
10 000	PE	22 ± 2	199.03 ± 57.41	25.24 ± 2.41	17.8 ± 6.9
	PE + SVR	1034 ± 110	41.18 ± 16.35	18.54 ± 2.31	11.0 ± 0.5
	Reference	1677 ± 32	105.55 ± 40.37	12.65 ± 0.44	8.8 ± 0.3
20 000	PE	24 ± 3	208.25 ± 52.03	24.97 ± 1.85	42.7 ± 18.2
	PE + SVR	1924 ± 175	37.83 ± 11.64	17.97 ± 1.86	41.9 ± 2.4
	Reference	3272 ± 45	104.24 ± 32.98	12.67 ± 0.38	29.7 ± 0.7
50 000	PE	27 ± 2	210.05 ± 58.02	24.85 ± 2.06	154.1 ± 38.9
	PE + SVR	4689 ± 416	39.44 ± 11.48	17.97 ± 1.79	254.7 ± 17.7
	Reference	8060 ± 83	103.24 ± 34.85	12.71 ± 0.29	172.8 ± 2.6
100 000	PE	29 ± 3	211.40 ± 50.63	24.85 ± 1.65	464.2 ± 121.1
	PE + SVR	9238 ± 860	40.76 ± 9.70	17.99 ± 1.59	1133.8 ± 81.0
	Reference	16024 ± 120	109.96 ± 41.30	12.75 ± 0.46	809.2 ± 13.1

V. CONCLUSION

A method for nonlinear hybrid system identification has been proposed, in which kernel functions have been introduced to estimate arbitrary and unknown nonlinearities. Large-scale experiments show that the resulting algorithm can accurately identify nonlinear hybrid systems from tens of thousands of noisy data in a reasonable time. Future work will focus on studying tuning procedures for the hyperparameters of the method, including the regularization constant C and the kernel parameter. Additionally, though the proposed method for the selection of support vectors in kernel submodels led to satisfactory results, better selection strategies will be investigated. In particular, we may expect some improvement by taking the target outputs y_i into account when selecting the support vectors. Future work will also consider piecewise systems, for which it may be preferable to select the SVs of a submodel only in the region where it is active.

REFERENCES

[1] F. Lauer, R. Vidal, and G. Bloch. A product-of-errors framework for linear hybrid system identification. In *Proc. of the 15th IFAC Symp. on System Identification (SYSID), Saint-Malo, France*, 2009.

[2] F. Lauer and G. Bloch. Switched and piecewise nonlinear hybrid system identification. In *Proc. of the 11th Int. Conf. on Hybrid Systems: Computation and Control (HSCC), St. Louis, MO, USA*, volume 4981 of *LNCS*, pages 330–343, 2008.

[3] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley-Interscience, 2000.

[4] G. Ferrari-Trecate, M. Muselli, D. Liberati, and M. Morari. A clustering technique for the identification of piecewise affine systems. *Automatica*, 39(2):205–217, 2003.

[5] H. Nakada, K. Takaba, and T. Katayama. Identification of piecewise affine systems based on statistical clustering technique. *Automatica*, 41(5):905–913, 2005.

[6] A. L. Juloski, S. Weiland, and W. Heemels. A Bayesian approach to identification of hybrid systems. *IEEE Trans. on Automatic Control*, 50(10):1520–1533, 2005.

[7] A. L. Juloski and S. Weiland. A Bayesian approach to the identification of piecewise linear output error models. In *Proc. of the 14th IFAC Symp. on System Identification (SYSID), Newcastle, Australia*, pages 374–379, 2006.

[8] F. Lauer and G. Bloch. A new hybrid system identification algorithm with automatic tuning. In *Proc. of the 17th IFAC World Congress, Seoul, Korea*, pages 10207–10212, 2008.

[9] F. Lauer. *From Support Vector Machines to Hybrid System Identification*. PhD thesis, Université Henri Poincaré Nancy 1, 2008.

[10] J. Roll, A. Bemporad, and L. Ljung. Identification of piecewise affine systems via mixed-integer programming. *Automatica*, 40(1):37–50, 2004.

[11] A. Bemporad, A. Garulli, S. Paoletti, and A. Vicino. A bounded-error approach to piecewise affine system identification. *IEEE Trans. on Automatic Control*, 50(10):1567–1580, 2005.

[12] R. Vidal, S. Soatto, Y. Ma, and S. Sastry. 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 (CDC), Maui, Hawaï, USA*, pages 167–172, 2003.

[13] Y. Ma and R. Vidal. Identification of deterministic switched ARX systems via identification of algebraic varieties. In *Proc. of the 8th Int. Conf. on Hybrid Systems: Computation and Control (HSCC), Zurich, Switzerland*, volume 3414 of *LNCS*, pages 449–465, 2005.

[14] L. Bako and R. Vidal. Identification of switched MIMO ARX models. In *Proc. of the 11th Int. Conf. on Hybrid Systems: Computation and Control (HSCC), St. Louis, MO, USA*, volume 4981 of *LNCS*, pages 43–57, 2008.

[15] R. Vidal. Recursive identification of switched ARX systems. *Automatica*, 44(9):2274–2287, 2008.

[16] O. L. Mangasarian and D. R. Musicant. Large scale kernel regression via linear programming. *Machine Learning*, 46(1-3):255–269, 2002.

[17] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, New York, NY, USA, 1995.

[18] J. A. K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, and J. Vandewalle. *Least Squares Support Vector Machines*. World Scientific, River Edge, NJ, USA, 2002.

[19] W. Huyer and A. Neumaier. Global optimization by multilevel coordinate search. *Journal of Global Optimization*, 14(4):331–355, 1999.

[20] C. Chang and C. Lin. LibSVM: a library for support vector machines, 2001. http://www.csie.ntu.edu.tw/∼cjlin/libsvm/.