Reduced-size kernel models for nonlinear hybrid system identification

van Luong Le, Gérard Bloch, Fabien Lauer

Reduced-Size Kernel Models for Nonlinear Hybrid System Identification

Van Luong Le, Gérard Bloch and Fabien Lauer

***Abstract*—The paper focuses on the identification of nonlinear hybrid dynamical systems, i.e., systems switching between multiple nonlinear dynamical behaviors. Thus the aim is to learn an ensemble of submodels from a single set of input-output data in a regression setting with no prior knowledge on the grouping of the data points into similar behaviors. To be able to approximate arbitrary nonlinearities, kernel submodels are considered. However, in order to maintain efficiency when applying the method to large data sets, a preprocessing step is required in order to fix the submodel sizes and limit the number of optimization variables. This paper proposes four approaches, respectively inspired by the fixed-size least-squares support vector machines, the feature vector selection method, the kernel principal component regression and a modification of the latter, in order to deal with this issue and build sparse kernel submodels. These are compared in numerical experiments, which show that the proposed approach achieves the simultaneous classification of data points and approximation of the nonlinear behaviors in an efficient and accurate manner.**

***Index Terms*—Hybrid dynamical systems, kernel methods, system identification, sparse models, switched regression**

I. Introduction

Hybrid dynamical systems have been extensively studied by the control community over the recent years as a potential class of dynamical models to approximate the behavior of complex cyber-physical systems. Despite this significant amount of work, the major issue of obtaining a model of the system from experimental data remains open. Formally, this problem, known as hybrid system identification [1], takes the form of a nonconvex optimization problem involving a large number of integer variables that depends on the number of data. Consequently, most proposed methods do not apply to large data sets.

More specifically, hybrid (dynamical) systems are a class of discrete-time AutoRegressive with eXternal input (ARX) systems of the form (in the single-input single-output case)

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

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_a outputs y_{i-k} and n_c inputs u_{i-n_k-k}, where n_k is the pure delay. The *discrete state* (or mode) $\lambda_i \in \{1, \dots, n\}$ determines 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.

Linear hybrid systems, for which all subsystems are linear, are usually categorized in two main classes: Switched linear ARX (SARX) systems, where the switches between subsystems are arbitrary and independent of the regression vector $\boldsymbol{x}_i$, and PieceWise Affine (PWA) systems, where the discrete state λ_i entirely depends on $\boldsymbol{x}_i$ thus partitioning the regression space into different operating modes. Nonlinear hybrid systems follow a similar nomenclature including Switched Nonlinear ARX (SNARX) and PieceWise Smooth (PWS) systems.

In this paper, we aim at finding a nonlinear one-step-ahead predictor $f = \{f_j\}_{j=1}^n$ in the hybrid form (1) from input-output data $\{(x_i, y_i)\}_{i=1}^N$. Though the method will be applicable to PWS systems, we focus on the identification of SNARX systems and on the approximation of the response surfaces of the subsystems which lead to the classification of the data points into modes. On the basis of this classification, any nonlinear estimator can then be used to recover better submodels independently. We further assume that the number of submodels n and their regressors are known. Note that these assumptions do not alleviate the major difficulty of the problem stemming from the fact that it naturally includes two intertwined subproblems: classification of the data points into their corresponding modes and regression of a submodel for each mode. In case the number of submodels n is unknown, this parameter acts on the trade-off between the fit to the data and the overall model complexity.

Related work. Most of the approaches proposed to solve the hybrid system identification problem, of which a good overview can be found in [1], consider only hybrid systems switching between linear dynamics. Even though, these methods face a nonconvex problem and either implement a local optimization approach, resulting in algorithms that are rather sensitive to their initialization, or rely on combinatorial optimization, becoming prohibitively time consuming even for moderate-size data sets. Another line of research is followed by the algebraic approach [2], [3], which 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. Partly building on ideas from this approach, a continuous optimization framework was recently proposed in [4]. In addition to being robust to noise and outliers, this last approach also significantly alleviates the complexity bottleneck when compared to previous methods.

To the best of our knowledge, the first approach to deal with *nonlinear* hybrid system identification without prior

V.L. Le and G. Bloch are with the Centre de Recherche en Automatique de Nancy (CRAN), Nancy–University, CNRS, France Luong.Le-Van@ensem.inpl-nancy.fr, gerard.bloch@esstin.uhp-nancy.fr

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

knowledge of the nonlinearities was proposed in [5] as an extension to the support vector regression-based method [6] which is limited to small data sets. Further extending these works in the framework of [4] resulted in the first algorithm for nonlinear hybrid system identification for large data sets as described in [7]. Note that the crucial issue in this approach in order to deal with large data sets is to fix the submodel size and thus to limit the number of optimization variables. The only other reference directly dealing with a similar problem is [8], where a sparse optimization based method is proposed to iteratively estimate the submodels one by one. However, it relies on the assumption that the difference between the outputs of the nonlinear subsystems is larger than a bound on the noise for all input, which is unrealistic as soon as the subsystems are defined by intersecting functions f_j.

Building reduced-size kernel models has been previously studied for the particular case of Least-Squares Support Vector Machines (LS-SVM) in [9], where the number of Support Vectors (SVs) can be fixed to a predefined number. Another approach for LS-SVM has been considered in [10], [11], [12], where a minimal set of training vectors is selected such that it induces a basis for the subspace containing the data mapped in feature space. Sparse kernel models can also be built with the Kernel Principal Component Regression (KPCR) approach proposed in [13] and based on kernel principal component analysis [14]. All these approaches can be used to build reduced-size kernel hybrid models, since they are only based on the input data and do not use the target output, which is undetermined in this context due to the unknown switches of the hybrid system.

Paper contribution. This paper extends the works of [7] and proposes efficient identification methods for *nonlinear* hybrid systems. In particular, four different approaches are considered to build sparse kernel submodels, which are the key to efficiency in this context. More specifically, these are inspired by the fixed-size LS-SVM [9], the Feature Vector Selection (FVS) [10], [11], the KPCR [13] and a modification of the latter, Reduced KPCR (RKPCR) by Incomplete Cholesky Decomposition [15], respectively. These approaches allow the number of optimization variables to remain small even when applied to large data sets, and thus to use global optimization solvers to estimate the model.

Paper organization. Section II introduces the nonlinear hybrid system identification framework. Then the reduced-size kernel models for large-scale problems are presented in §III with the four proposed methods: Entropy maximization in §III-A, FVS in §III-B, KPCR in §III-C and RKPCR in §III-D. Numerical experiments are given in §IV and conclusions in §V.

II. Nonlinear Hybrid System Identification Framework

This section reviews the *Product-of-Errors* (PE) identification framework proposed in [7], where kernel submodels are used to approximate arbitrary nonlinearities in hybrid systems.

A. Kernel Models for Nonlinear Hybrid Systems

Following the nonlinear black-box modeling approach of [7], each submodel of a nonlinear hybrid model is expressed as a kernel expansion built from the set of training input data $S = \{\boldsymbol{x}_i\}_{i=1}^N$, i.e., of the form

$$f_j(\boldsymbol{x}) = \sum_{k=1}^{N} \alpha_{kj} k_j(\boldsymbol{x}_k, \boldsymbol{x}) + b_j, \tag{2}$$

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. The remainder of the paper will focus on Gaussian RBF kernels.

A kernel function implicitly computes inner products, $k_j(\boldsymbol{x}_k, \boldsymbol{x}) = \langle \boldsymbol{\phi}_j(\boldsymbol{x}_k), \boldsymbol{\phi}_j(\boldsymbol{x}) \rangle$, between points in a higher-dimensional *feature space* $\mathcal{F}$ obtained by an hidden nonlinear mapping

$$\phi_j : \boldsymbol{x} \mapsto \boldsymbol{\phi}_j(\boldsymbol{x}), \tag{3}$$

of the points $\boldsymbol{x}$ in the original input space.

Note that different kernel functions k_j can be used in (2) for the different submodels f_j. Thus it is possible to take prior knowledge into account such as 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.

B. Nonlinear Product-of-Errors Estimator

The PE estimator of linear hybrid systems proposed in [4] relies on an optimization problem involving a product of error terms, also considered in the algebraic approach [3]. Introducing submodels in the kernel form (2) in this framework leads to the nonlinear PE estimator for nonlinear hybrid systems expressed as the solution to

$$\min_{\{\boldsymbol{\alpha}_j\},\{b_j\}} \ \frac{1}{n} \sum_{j=1}^{n} R(\boldsymbol{\alpha}_j) + \frac{C}{N} \sum_{i=1}^{N} \prod_{j=1}^{n} \ell \left(y_i - \sum_{k=1}^{N} \alpha_{kj} k_j(\boldsymbol{x}_k, \boldsymbol{x}_i) - b_j \right), \tag{4}$$

where ℓ is a smooth loss function and $R(\boldsymbol{\alpha}_j)$ is the regularizer acting on the parameters $\boldsymbol{\alpha}_j$ of the submodel f_j. For instance, the model complexity can be measured by the L_1-norm of the parameter vector, i.e., $R(\boldsymbol{\alpha}_j) = \|\boldsymbol{\alpha}_j\|_1$. This regularizer penalizes 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., $R(\boldsymbol{\alpha}_j) = \|\boldsymbol{\alpha}_j\|_2^2$ is also possible, but may result in less sparse models.

III. Reduced-Size Kernel Models For Large-Scale Problems

As in Support Vector Machines (SVMs) [16], we refer to the vectors $\boldsymbol{x}_i$ for which the associated $\{\alpha_{ij}\}_{j=1,\ldots,n}$ parameters

are nonzero as the *Support Vectors* (SVs), since these are the only data points kept in the final model. For submodels in kernel form (2), the optimization program (4) involves a large number of variables associated to the number of *potential* SVs. Since the kernel submodels consider all the data points $\boldsymbol{x}_i$, $i = 1, \ldots, N$, as potential SVs, the number of variables α_{ij} and b_j is $n(N+1)$. Thus solving this problem for large N with a global optimization solver may become prohibitively time consuming.

In this section, we propose four methods to reduce the number of parameters α_{kj} in (2) *before* starting the optimization. Let

$$S_j = \{\boldsymbol{x}_{kj}\}_{k=1}^{M_j} \tag{5}$$

denote the set of M_j SVs retained for the jth reduced-size submodel

$$\tilde{f}_j(\boldsymbol{x}) = \sum_{k=1}^{M_j} \beta_{kj} k_j(\boldsymbol{x}_{kj}, \boldsymbol{x}) + b_j. \tag{6}$$

The $(M_j + 1)$ parameters of submodel $\tilde{f}_j$ are now given by $\boldsymbol{\beta}_j = [\beta_{1j}, \ldots, \beta_{M_j j}]^T$ and b_j.

With these notations, the complete identification procedure is as follows.

1) Find the structure of each submodel $\tilde{f}_j(\boldsymbol{x})$ as in (6) by applying one of the methods presented below.
2) Train the hybrid model by solving

$$\min_{\{\boldsymbol{\beta}_j\},\{b_j\}} \frac{1}{n} \sum_{j=1}^{n} \frac{\boldsymbol{\beta}_j^T \boldsymbol{\beta}_j}{M_j} + \frac{C}{N} \sum_{i=1}^{N} \prod_{j=1}^{n} \ell\left(y_i - \tilde{f}_j(\boldsymbol{x}_i)\right). \tag{7}$$

3) Estimate the mode $\hat{\lambda}_i$ for each data point by

$$\hat{\lambda}_i = \arg \min_{j=1,\ldots,n} \ell(y_i - \tilde{f}_j(\boldsymbol{x}_i)),\ i = 1, \ldots, N, \tag{8}$$

and classify the data into n subsets accordingly.
4) Re-estimate the submodels with a nonlinear estimator applied independently to each data subset.

Note that the reduced-size submodels (6) are based on an intrinsically sparse representation of the data, hence the choice of the smooth L_2-norm regularization over the low-dimensional parameter vectors $\boldsymbol{\beta}_j$ in (7).

The final optimization program (7) involves only $\sum_{j=1}^{n}(M_j + 1)$ variables instead of $n(N+1)$ as in (4). This allows the complexity of the algorithm (7) to scale only linearly with respect to the number of training data N (through the summation term), as experimentally verified in [4].

In this procedure, the first step may be interpreted as selecting a subset of the columns of the kernel matrix. In particular, the nature of the hybrid system identification problem and of the global optimization program (4) calls for feature selection methods that can apply *without* knowledge of the target values y_i (which cannot be assigned to a submodel ahead of Step 3) and *before* optimizing the parameters (which are too numerous otherwise). These constraints motivated the choice of the four methods described in the following subsections and the fact that, for feature selection, we do not consider regularization-based methods and other approaches requiring to solve an instance of the optimization problem (4) with the full model.

After the classification of the data in Step 3, the submodels can be re-estimated in Step 4 by considering n independent problems with n distinct data sets, to which any classical nonlinear estimation method can be applied. The sparsity and accuracy of the final model thus depends on the properties of this particular method. For instance, the experiments of Section IV will use SVMs.

In the following, $\boldsymbol{K}_j$ will denote the kernel matrix of mode j with components $(\boldsymbol{K}_j)_{ik} = k_j(\boldsymbol{x}_i, \boldsymbol{x}_k)$ and $\boldsymbol{K}_{S_j}$ its submatrix built from the rows and columns corresponding to the SVs in S_j, i.e.,

$$\boldsymbol{K}_{S_j} = \begin{bmatrix} k_j(\boldsymbol{x}_{1j}, \boldsymbol{x}_{1j}) & \ldots & k_j(\boldsymbol{x}_{1j}, \boldsymbol{x}_{M_j j}) \\ \vdots & \ddots & \\ k_j(\boldsymbol{x}_{M_j j}, \boldsymbol{x}_{1j}) & \ldots & k_j(\boldsymbol{x}_{M_j j}, \boldsymbol{x}_{M_j j}) \end{bmatrix}. \tag{9}$$

Also note that in the four proposed procedures, 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 to capture only the general distribution of the data in the feature space $\mathcal{F}$ in order to ensure sufficient support for the model. However, for piecewise models, where a particular submodel is only active in a given region of input space, the procedures also select SVs outside of this region. In this case, how to obtain sparser representations should be investigated.

A. Entropy Maximization

The fixed-size Least Squares SVM (LS-SVM) [9] 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_j (3) 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 (6).

As in fixed-size LS-SVM [9], the selection algorithm maximizes the quadratic Rényi entropy H, which quantifies the diversity, uncertainty or randomness of a system. For a particular mode j, we approximate H by

$$H_j \approx -\log \frac{1}{M_j^2} \mathbf{1}^T \boldsymbol{K}_{S_j} \mathbf{1}, \tag{10}$$

where $\boldsymbol{K}_{S_j}$ is given by (9), and the procedure to select the SVs is as follows.

1) Randomly select a subset S_j with M_j SVs from the training set S, and initialize $\overline{S} = S \setminus S_j$.
2) Randomly select an SV in S_j, $\boldsymbol{x}^\star$, and one of the remaining training samples, $\boldsymbol{x}^\dagger \in \overline{S}$.
3) If the criterion (10) increases via replacing $\boldsymbol{x}^\star$ by $\boldsymbol{x}^\dagger$, retain $\boldsymbol{x}^\dagger$ as an SV in S_j and replace $\boldsymbol{x}^\dagger$ by $\boldsymbol{x}^\star$ in $\overline{S}$.
4) Repeat from 2 until the increase of the criterion is too small or a maximum number of iterations is reached.

In this procedure, the numbers of SVs $\{M_j\}_{j=1,\ldots,n}$ are hyperparameters that must be fixed *a priori*. Following [7], for Gaussian RBF kernels with bandwidth parameter σ_j, the numbers M_j can be set according to the heuristic

$$M_j = \left\lfloor \frac{1}{\sigma_j} \max_{k=1,\ldots,p} \left(\max_{i=1,\ldots,N} x_{ik} - \min_{i=1,\ldots,N} x_{ik} \right) \right\rfloor, \quad (11)$$

where $\lfloor \cdot \rfloor$ denotes the integer part of its argument and x_{ik} is the kth component of $\boldsymbol{x}_i$.

This heuristic is not optimal in the sense of minimizing the generalization error, but it ensures sufficient support for the model over the whole input space. The numbers M_j in (11) strongly depend on the bandwidths σ_j, since more SVs are needed to cover the whole input space with a smaller bandwidth. In practice, the values of σ_j can influence the quality of the model as they control the smoothness of the submodels. Proper tuning of these values may require multiple trials or prior knowledge on the relative smoothness of the subsystems in the model. However, suboptimal numbers M_j are sufficient to obtain rough mode estimates $\hat{\lambda}_i$ and a data classification to re-estimate the submodels in Step 4. If these refined submodels are learned by SVM techniques for instance, then the final number of SVs is automatically determined.

B. Feature Vector Selection

Following the Feature Vector Selection (FVS) method of [10], [11], the selection of support vectors aims at finding a suitable set of basis vectors in the feature space $\mathcal{F}$ that spans the data subspace.

If we let $\boldsymbol{w}_j = \sum_{k=1}^{N} \alpha_{kj} \boldsymbol{\phi}_j(\boldsymbol{x}_k)$, then the kernel expansion (2) can be rewritten in terms of inner products in feature space to yield a linear form with respect to $\boldsymbol{w}_j$ as

$$f_j(\boldsymbol{x}) = \sum_{k=1}^{N} \alpha_{kj} \langle \boldsymbol{\phi}_j(\boldsymbol{x}_k), \boldsymbol{\phi}_j(\boldsymbol{x}) \rangle + b_j = \langle \boldsymbol{w}_j, \boldsymbol{\phi}_j(\boldsymbol{x}) \rangle + b_j. \quad (12)$$

The vector $\boldsymbol{w}_j$ is represented by means of a set of N vectors $\{\boldsymbol{\phi}_j(\boldsymbol{x}_k)\}_{k=1}^{N}$ and there are N parameters α_{kj} to be determined. In practice, the dimension of the subspace which contains the whole nonlinearly-mapped data set in feature space is significantly lower than N and equal to the numerical rank of the kernel matrix $\boldsymbol{K}_j$.

Thus, in order to reduce the number of parameters, one can express $\boldsymbol{w}_j$ from a reduced set of basis vectors $\{\boldsymbol{\phi}_j(\boldsymbol{x}_{kj})\}_{k=1}^{M_j}$ as

$$\boldsymbol{w}_j = \sum_{k=1}^{M_j} \beta_{kj} \phi_j(\boldsymbol{x}_{kj}), \quad (13)$$

where $\boldsymbol{x}_{kj} \in S_j$, with typically $M_j \ll N$. This leads to the jth reduced-size submodel in the form (6) as

$$\tilde{f}_j(\boldsymbol{x}) = \sum_{k=1}^{M_j} \beta_{kj} \langle \boldsymbol{\phi}_j(\boldsymbol{x}_{kj}), \boldsymbol{\phi}_j(\boldsymbol{x}) \rangle + b_j = \sum_{k=1}^{M_j} \beta_{kj} k_j(\boldsymbol{x}_{kj}, \boldsymbol{x}) + b_j. \quad (14)$$

In comparison to the previous method, M_j is not fixed *a priori*, but simply corresponds to the dimension of the smallest subspace containing the data in feature space.

The set S_j (5) induces a basis vector set in feature space by the mapping ϕ_j (3), which can then be used to approximate $\boldsymbol{\phi}_j(\boldsymbol{x})$ for any $\boldsymbol{x}$ in input space. The Feature Vector Selection proposed in [10] searches for the set S_j that minimizes the reconstruction error between this approximation and the true mapping of the points over the entire training data set S. As detailed in [10] this is equivalent to finding the set S_j which maximizes the following criterion

$$J(S_j) = \frac{1}{N} \sum_{i=1}^{N} \frac{\boldsymbol{k}_{S_j i}^T \boldsymbol{K}_{S_j}^{-1} \boldsymbol{k}_{S_j i}}{k_j(x_i, x_i)}, \quad (15)$$

where $\boldsymbol{k}_{S_j i} = \left[k_j(\boldsymbol{x}_{1j}, \boldsymbol{x}_i), ..., k_j(\boldsymbol{x}_{M_j j}, \boldsymbol{x}_i) \right]^T$.

Though the method proposed in [10] to maximize (15) can be improved for efficiency as in [12], it remains rather time consuming for large data sets. In order to maintain as low as possible the computational cost of the overall estimation procedure, in which the basis selection is only the first step, we instead propose the following randomized algorithm.

1) Initialize $S_j = \emptyset$, $\overline{S} = S$, $k = 1$ and define $J(\emptyset) = 0$.
2) Append to S_j a randomly selected vector $\boldsymbol{x}_{kj}$ from the set $\overline{S}$ of remaining training input data and compute $J(S_j)$.
3) If $J(S_j)$ increases, retain $\boldsymbol{x}_{kj}$ in S_j and update $\overline{S} = \overline{S} \setminus \{\boldsymbol{x}_{kj}\}$, otherwise remove $\boldsymbol{x}_{kj}$ from S_j.
4) Loop from Step 2 until $\boldsymbol{K}_{S_j}$ is no longer invertible $\left(\det(\boldsymbol{K}_{S_j}) < \epsilon\right)$.

Then the number of basis vectors M_j to use in (14) is given by $M_j = |S_j|$.

C. Kernel Principal Component Regression

Following the KPCR method in [13], the number of optimization variables in (4) can be reduced by using only several principal components of the kernel matrix which are sufficient to account for most of the structure in the data.

Formally, for a particular mode j, we are interested in finding the kernel principal components that can represent all data points associated to this mode. However, as the discrete state λ_i (determining to which mode belongs a data point) is unknown for the training data, we have to compute the kernel principal components from the whole data set S for each mode. Note nevertheless that these principal components can be different from one mode to another if the kernel functions k_j are different.

Let $\boldsymbol{\Phi}_j$ be the $(L \times N)$ matrix whose ith column is the vector $\boldsymbol{\phi}_j(\boldsymbol{x}_i)$ of the observation $\boldsymbol{x}_i$ mapped into the L-dimensional feature space $\mathcal{F}$. We assume that the mapped data is centered in feature space, i.e., $\sum_{i=1}^{N} \boldsymbol{\phi}_j(\boldsymbol{x}_i) = 0$. If not, the kernel matrix $\boldsymbol{K}_j$ must be substituted by

$$\hat{\boldsymbol{K}}_j = \left(\boldsymbol{I}_N - \frac{1}{N} \boldsymbol{1}_{N \times N} \right) \boldsymbol{K}_j \left(\boldsymbol{I}_N - \frac{1}{N} \boldsymbol{1}_{N \times N} \right), \quad (16)$$

as proposed in [14].

Let $\boldsymbol{\Lambda}_j = diag\{\tilde{\lambda}_1, \ldots, \tilde{\lambda}_L\}$, $\boldsymbol{V}_j = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_L]$ be the eigenvalue diagonal matrix and the corresponding normalized orthogonal eigenvector matrix of the covariance matrix $\frac{1}{N} \boldsymbol{\Phi}_j \boldsymbol{\Phi}_j^T$. As in the PCA method, a feature vector $\boldsymbol{\phi}_j(\boldsymbol{x})$ is

transformed to new coordinates by the eigenvectors $\boldsymbol{v}_k$, i.e., the projection of $\boldsymbol{\phi}_j(\boldsymbol{x})$ onto the eigenvectors $\boldsymbol{v}_k$,

$$\boldsymbol{\phi}_j^{new}(\boldsymbol{x}) = \boldsymbol{V}_j^T \boldsymbol{\phi}_j(\boldsymbol{x}). \tag{17}$$

The KPCR model can be written as

$$\tilde{f}_j(\boldsymbol{x}) = \boldsymbol{\beta}_j^T \boldsymbol{\phi}_j^{new}(\boldsymbol{x}) + b_j = \boldsymbol{\beta}_j^T \boldsymbol{V}_j^T \boldsymbol{\phi}_j(\boldsymbol{x}) + b_j, \tag{18}$$

where $\boldsymbol{\beta}_j$ is a coefficient vector in the feature space $\mathcal{F}$. Note that all $\boldsymbol{v}_k$ with $\tilde{\lambda}_k \neq 0$ lie in the span of N mappings $\{\boldsymbol{\phi}_j(\boldsymbol{x}_1), \ldots, \boldsymbol{\phi}_j(\boldsymbol{x}_N)\}$. To avoid an explicit mapping of the data, the eigenvectors $\boldsymbol{v}_k$ are computed thanks to an equivalent eigenvalue problem proposed by Schölkopf *et al.* in [14]:

$$\boldsymbol{K}_j \boldsymbol{\gamma}_k = \mu_k \boldsymbol{\gamma}_k, \tag{19}$$

$$\tilde{\lambda}_k = \frac{\mu_k}{N}, \tag{20}$$

$$\boldsymbol{v}_k = \boldsymbol{\Phi}_j \frac{\boldsymbol{\gamma}_k}{\sqrt{N\tilde{\lambda}_k}}, \quad k = 1, \ldots, N, \tag{21}$$

where μ_k and $\boldsymbol{\gamma}_k$ ($k = 1, \ldots, N$) are eigenvalues and eigenvectors of the matrix $\boldsymbol{K}_j$.

After arranging the eigenvectors $\boldsymbol{\gamma}_k$ with the corresponding eigenvalues μ_k in decreasing order, one only uses the first M_j nonlinear principal components of $\boldsymbol{\phi}_j^{new}(\boldsymbol{x})$ which is computed by (17) and (21). This reduces the size of the coefficient vector $\boldsymbol{\beta}_j$ to be estimated and leads to

$$\tilde{f}_j(\boldsymbol{x}) = \boldsymbol{\beta}_j^T \boldsymbol{A}_j \boldsymbol{k}_j(\cdot, \boldsymbol{x}) + b_j, \tag{22}$$

where $\boldsymbol{A}_j = \left[\frac{\boldsymbol{\gamma}_1}{\sqrt{\mu_1}}, \ldots, \frac{\boldsymbol{\gamma}_{M_j}}{\sqrt{\mu_{M_j}}}\right]^T \in \mathbb{R}^{M_j \times N}$ and $\boldsymbol{k}_j(\cdot, \boldsymbol{x}) = [k_j(\boldsymbol{x}_1, \boldsymbol{x}), ..., k_j(\boldsymbol{x}_N, \boldsymbol{x})]^T \in \mathbb{R}^N$.

The procedure to build the reduced-size kernel form for a particular mode j is as follows.

1) Compute the kernel matrix $\boldsymbol{K}_j$ from the training data set S.
2) Compute the M_j largest eigenvalues and corresponding eigenvectors of $\boldsymbol{K}_j$ and calculate $\mathbf{A}_j$.
3) Apply the form (22) in (7).

The number of nonlinear principal components M_j must be sufficient to describe the structure of the data. For a given $\rho \in [0, 1]$, the cumulative energy content can be used to estimate M_j as the smallest number m such that

$$\frac{\sum_{i=1}^{m} \mu_i}{\sum_{i=1}^{N} \mu_i} \geq \rho, \tag{23}$$

where μ_i, $i = 1, ..., N$, are the eigenvalues arranged in decreasing order and $\sum_{i=1}^{N} \mu_i = \text{Trace}(\boldsymbol{K}_j)$. Note that $\text{Trace}(\boldsymbol{K}_j) = N$ in case of a Gaussian RBF kernel matrix.

D. Reduced Kernel Principal Component Regression

In the method above, one obtains a reduced-size kernel submodel form (22) with only $M_j + 1$ parameters that need to be estimated. However, the resulting model needs to retain the N original data points instead of M_j as in the form (6). Indeed computing its output for a new input $\boldsymbol{x}$ involves the vector $\boldsymbol{k}_j(\cdot, \boldsymbol{x}) \in \mathbb{R}^N$. Moreover, the eigenvalue decomposition of a too large kernel matrix $\boldsymbol{K}_j$ can be prohibitive. To avoid these issues, the kernel matrix can be approximated by a low rank matrix $\tilde{\boldsymbol{K}}_j$ via the Nyström method [17].

Most of the computations for the low rank approximation $\tilde{\boldsymbol{K}}_j$ of a kernel matrix $\boldsymbol{K}_j$ involve only a subset of the training data. In the original Nyström method, the subset selection is random with the subset size fixed beforehand. Such a selection influences the accuracy of the solution and leads to a more complex implementation. Thus, the Nyström method based on an incomplete Cholesky decomposition is proposed in [15]. The incomplete Cholesky decomposition of matrix $\boldsymbol{K}_j$ provides automatically an $R_j \times N$-dimensonal matrix $\boldsymbol{C}_j = [\boldsymbol{L}_j \ \boldsymbol{N}_j]$ such that $\tilde{\boldsymbol{K}}_j = \boldsymbol{C}_j^T \boldsymbol{C}_j$ and a corresponding data subset S'_j of size R_j ($R_j < N$) such that $\boldsymbol{K}_{S'_j} = \boldsymbol{L}_j^T \boldsymbol{L}_j$. Then, the R_j eigenvalues of the $R_j \times R_j$ correlation matrix $\boldsymbol{Q}_j = \boldsymbol{C}_j \boldsymbol{C}_j^T$ are identical to the largest ones of $\tilde{\boldsymbol{K}}_j$. According to this method, the model (22) is rewritten with matrix $\boldsymbol{A}_j$ replaced by one of dimension $R_j \times R_j$ as

$$\boldsymbol{A}_{jR_j} = \boldsymbol{E}_j^T \boldsymbol{L}_j^{-T}, \tag{24}$$

where $\boldsymbol{E}_j$ is the eigenvector matrix of $\boldsymbol{Q}_j$ and its columns are arranged in decreasing order of the related eigenvalues.

As before, only the first $M_j \leq R_j$ first eigenvector columns of $\boldsymbol{E_j}$ are selected according to the criterion (23) to form a reduced model as in (22):

$$\tilde{f}_j(\boldsymbol{x}) = \boldsymbol{\beta}_j^T \boldsymbol{A}_{jM_j} \tilde{\boldsymbol{k}}_j(\cdot, \boldsymbol{x}) + b_j, \tag{25}$$

where $\boldsymbol{\beta}_j$ is the M_j-dimensional parameter vector, $\boldsymbol{A}_{jM_j} = \boldsymbol{E}_{jM_j}^T \boldsymbol{L}_j^{-T}$ is an $M_j \times R_j$ matrix and the reduced vector $\tilde{\boldsymbol{k}}_j(\cdot, \boldsymbol{x}) = [k(\boldsymbol{x}_{1j}, \boldsymbol{x}), ..., k_j(\boldsymbol{x}_{R_jj}, \boldsymbol{x})]^T$ is calculated for an $\boldsymbol{x}$ with $\boldsymbol{x}_{ij}$, $i = 1, \ldots, R_j$, in the selected subset S'_j.

The procedure to build the reduced-size kernel form for a particular mode j is as follows.

1) Compute the kernel matrix $\boldsymbol{K}_j$ from the training data set S.
2) Obtain the matrix $\boldsymbol{C}_j$ and the subset S'_j by an incomplete Cholesky decomposition of $\boldsymbol{K}_j$.
3) Compute the M_j largest eigenvalues and corresponding eigenvectors of $\boldsymbol{Q}_j$ and calculate $\mathbf{A}_{jM_j}$.
4) Apply the form (25) in (7).

IV. Numerical Experiments

This section presents numerical results on two examples. The first one involves the estimation of a function switching between two unknown nonlinear functions in Sect. IV-A, while the second one considers the identification of a switched nonlinear dynamical system in Section IV-B.

As proposed in [4], all optimization programs are solved with the Multilevel Coordinate Search (MCS) algorithm [18]. 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). All experiments are performed using only Matlab code on a standard desktop computer.

This section compares the four proposed methods for building reduced-size kernel submodels: Entropy maximization (Sect. III-A), FVS (Sect. III-B), KPCR (Sect. III-C) and

RKPCR (Sect. III-D). In the following Tables of results, the size of the SV sets for mode 1 (M_1) and mode 2 (M_2) is given by (11) for the Entropy maximization while being automatically determined for the other methods. For the KPCR and RKPCR methods, ρ in (23) is set to 0.9. The quality of the models is evaluated on an independent and noise-free test set of $N_t = 2000$ data points by the following performance indexes: the normalized criterion FIT $= 100\left(1 - \frac{\|\hat{\boldsymbol{y}}-\boldsymbol{y}\|_2}{\|\boldsymbol{y}-\bar{y}\mathbf{1}\|_2}\right)$, where $\boldsymbol{y}$ contains the target outputs, $\bar{y}$ their mean and $\hat{\boldsymbol{y}}$ the predicted outputs using either the estimated discrete state $\hat{\lambda}_i$ (8) (FITa) or the true λ_i (FITb) and the classification error rate on the test set (Test Classif. err.). The classification error rate on the training set (Train. Classif. err.) is also given in order to analyze the ability of the methods to separate between the modes. The computing times of the methods are reported by distinguishing between the time required by the SV selection in Step 1 of the complete procedure (Selection Time) and the time required by the MCS solver for Step 2 (Optimization Time). The re-estimation tables correspond to the refinement of the submodels in Step 4 by standard SVM for regression [16] applied independently to each group of data according to the classification given by $\hat{\lambda}_i$ (8). This step uses the same kernel hyperparameters and regularization trade-off $C = 100$ as all the compared methods. The loss function $\ell(e) = e^2$ is used in (7). Note that all numbers in the Tables below account for averages and standard deviations over 100 trials with different random noise sequences.

A. Illustrative Example

Consider the function arbitrarily switching between two nonlinear behaviors as

$$y(x) = \begin{cases} x^2, & \text{if } \lambda = 1, \\ \sin(3x) + 2, & \text{if } \lambda = 2. \end{cases} \tag{26}$$

A training set of $N = 2000$ points is generated by (26) with additive zero-mean Gaussian noise (standard deviation $\sigma_e = 0.3$) for uniformly distributed random $x_i \in [-3, 3]$ and uniformly distributed random $\lambda_i \in \{1, 2\}$. The data are shown in Figure 1 as black dots. The difficulty of this toy example lies in the crossing of the submodels, which results in strongly mixed data at particular locations (e.g., for $-1 < x < -0.2$ in Fig. 1). In particular, these crossings potentially generate undesired switches between the submodels and violate the assumption required by the method in [8].

In this experiment, the training data are normalized to zero mean and unit variance. The optimization program (7) is solved with two reduced-size submodels of the form (6) using Gaussian RBF kernels of width $\sigma_1 = 0.8$ and $\sigma_2 = 0.2$, respectively. Representative examples of the resulting submodels are shown in Figure 1. Table I shows the results. For a comparison, the FIT of the reference model obtained by applying the re-estimation step from the true classification is 93.50 ± 2.91.

The classification error rates on the training set as low as 10% show that the algorithm is able to correctly separate between the two modes. Remaining classification errors are mostly due to indistinguishable points at the intersection of the two nonlinear functions. Thus they do not incur significant errors in the re-estimation step, which, according to the FITa, leads to accurately refined models, especially for the RKPCR method.

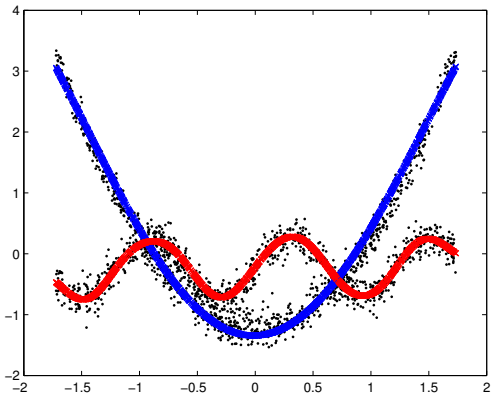

Fig. 1. Estimation of a switched nonlinear function from 2000 noisy data points (black dots). The red and blue curves show the estimated reduced-size submodels based on the KPCR method.

B. Switched Nonlinear Dynamical System

The next example considers the identification of a dynamical system arbitrarily switching between two modes as

$$y_i = \begin{cases} 0.9y_{i-1} + 0.2y_{i-2}, & \text{if } \lambda_i = 1, \\ (0.8 - 0.5\exp(-y_{i-1}^2))y_{i-1} - \\ \quad (0.3 + 0.9\exp(-y_{i-1}^2))y_{i-2} + & \text{if } \lambda_i = 2. \\ \quad 0.4\sin(2\pi y_{i-1}) + 0.4\sin(2\pi y_{i-2}), \end{cases} \tag{27}$$

A training set of $N = 2000$ points is generated by (27) with a uniformly distributed random sequence of $\lambda_i \in \{1, 2\}$ and an additive zero-mean Gaussian noise (standard deviation $\sigma_e = 0.1$) from the initial condition $y_0 = y_{-1} = 0.1$, whereas the noise-free test set uses $y_0 = 0.4$, $y_{-1} = -0.3$. Note that the noise is added to y_i during the data generation process, resulting in colored noise.

For the identification, the submodel f_1 uses a linear kernel with an arbitrary number of SVs $M_1 = 5$ for the entropy maximization method (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 ($\sigma = 0.3$). Corresponding results are reported in Table II. For a comparison, the FIT of the reference model with known mode is 92.79 ± 2.67. In these experiments, the PCA-based methods (KPCR and RKPCR) yield better FITs and fewer classification errors for a low computing time.

V. Conclusions

This paper focused on the switched regression problem at the core of hybrid system identification in the particular case of systems switching between unknown nonlinear dynamics. The proposed approach relies on the ability to express each submodel in a sparse kernel form, which allows a global

TABLE I
COMPARISON OF THE FOUR PROPOSED METHODS TO BUILD AND ESTIMATE REDUCED-SIZE KERNEL HYBRID MODELS.

Method	Entropy max.	FVS	KPCR	RKPCR
Estimation				
M_1 / M_2	4 / 17	7.0 ± 0.7 / 14.9 ± 1.8	3 ± 0 / 10 ± 0	3 ± 0 / 10 ± 0
FITa(%)	87.06 ± 2.03	87.95 ± 3.83	88.33 ± 3.89	86.20 ± 2.17
FITb(%)	81.51 ± 21.04	78.25 ± 26.48	88.21 ± 3.97	82.35 ± 21.43
Test Classif. err. (%)	5.22 ± 9.94	7.71 ± 10.63	2.14 ± 1.08	4.32 ± 5.35
Train. Classif. err. (%)	8.47 ± 9.65	10.80 ± 9.92	5.25 ± 0.58	7.30 ± 4.80
Selection Time (s)	0.94 ± 0.01	7.27 ± 1.13	8.80 ± 0.80	0.06 ± 0.04
Optimization Time (s)	3.1 ± 0.7	3.3 ± 1.1	1.9 ± 0.5	1.3 ± 0.4
Re-estimation				
FITa(%)	91.51 ± 3.50	92.36 ± 2.44	89.81 ± 4.26	92.75 ± 2.70
FITb(%)	85.77 ± 22.33	82.00 ± 27.4	89.69 ± 4.340	88.92 ± 22.20
Test Classif. err. (%)	4.75 ± 10.26	6.65 ± 10.70	2.13 ± 1.10	3.05 ± 5.45

TABLE II
ESTIMATION OF AN ARBITRARILY SWITCHED NONLINEAR ARX SYSTEM.

Method	Entropy max.	FVS	KPCR	RKPCR
Estimation				
M_1 / M_2	5 / 30	2.0 ± 0 / 28.1 ± 3.8	2.0 ± 0 / 36.8 ± 2.4	2.0 ± 0 / 37.0 ± 2.7
FITa(%)	71.22 ± 2.75	69.57 ± 4.01	80.76 ± 3.44	81.67 ± 3.24
FITb(%)	53.86 ± 8.59	56.77 ± 8.50	73.17 ± 5.31	75.81 ± 4.75
Test Classif. err. (%)	20.16 ± 4.37	17.26 ± 4.84	8.85 ± 2.61	7.74 ± 2.31
Train. Classif. err. (%)	21.69 ± 3.60	19.34 ± 4.18	12.67 ± 2.28	11.94 ± 2.00
Selection Time (s)	1.07 ± 0.06	18.21 ± 4.15	1.85 ± 0.13	1.93 ± 0.15
Optimization Time (s)	6.5 ± 2.0	4.9 ± 2.0	6.60 ± 2.90	7.42 ± 3.07
Re-estimation				
FITa(%)	86.03 ± 2.36	85.17 ± 4.39	89.05 ± 4.63	90.03 ± 3.69
FITb(%)	77.19 ± 7.95	77.05 ± 8.95	83.88 ± 4.99	84.71 ± 4.08
Test Classif. err. (%)	12.29 ± 4.45	9.19 ± 5.16	4.40 ± 1.73	3.86 ± 1.18

optimization solver to efficiently estimate the parameters of the model. Four methods were proposed and compared for the selection of a subset of the training data on the basis of which such reduced-size models can be built. The entropy maximization approach requires to fix the model size arbitrarily or through the heuristic (11) for Gaussian RBF kernels. On the other hand, the other approaches can determine the model size either as a byproduct of the procedure or through a high-level parameter such as the ratio of cumulative energy content. Experiments showed that these latter methods can sufficiently reduce the model size to allow the overall problem to be solved.

Determining the number of submodels is an important issue for all hybrid system identification methods (linear and nonlinear). The paper focused on the estimation of the submodels under the assumption that this number is fixed *a priori*, as is the case with many other methods, and provided the first and most central building block for a complete nonlinear hybrid system identification procedure. Further investigation will focus on automatic procedures for the tuning of the number of submodels. In addition, one of the remaining open issues with the proposed method concerns colored noise which implies a bias in the estimation of dynamical systems. Future work will also aim at specializing the algorithm for the piecewise smooth regression setting, where the different modes and nonlinear behaviors are separated in the input space.

ACKNOWLEDGEMENTS

We are grateful to the reviewers for their comments and suggestions which substantially improved the paper.

REFERENCES

[1] S. Paoletti, A. Juloski, G. Ferrari-Trecate, and R. Vidal, "Identification of hybrid systems: a tutorial," *European Journal of Control*, vol. 13, no. 2-3, pp. 242–262, 2007.

[2] 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*, 2003, pp. 167–172.

[3] 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), Zürich, Switzerland*, ser. LNCS, vol. 3414, 2005, pp. 449–465.

[4] F. Lauer, G. Bloch, and R. Vidal, "A continuous optimization framework for hybrid system identification," *Automatica*, vol. 47, no. 3, pp. 608–613, 2011.

[5] 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*, ser. LNCS, vol. 4981, 2008, pp. 330–343.

[6] ——, "A new hybrid system identification algorithm with automatic tuning," in *Proc. of the 17th IFAC World Congress, Seoul, South Korea*, 2008, pp. 10 207–10 212.

[7] F. Lauer, G. Bloch, and R. Vidal, "Nonlinear hybrid system identification with kernel models," in *49th IEEE Int. Conf. on Decision and Control (CDC), Atlanta, GA, USA*, 2010, pp. 696–701.

[8] L. Bako, K. Boukharouba, and S. Lecoeuche, "An ℓ_0-ℓ_1 norm based optimization procedure for the identification of switched nonlinear systems," in *49th IEEE Int. Conf. on Decision and Control (CDC), Atlanta, GA, USA*, 2010, pp. 4467–4472.

[9] J. 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.

[10] G. Baudat and F. Anouar, "Feature vector selection and projection using kernels," *Neurocomputing*, vol. 55, no. 1-2, pp. 21–38, 2003.
[11] G. Cawley and N. Talbot, "Reduced rank kernel ridge regression," *Neural Processing Letters*, vol. 16, no. 3, pp. 293–302, 2002.
[12] ——, "Efficient formation of a basis in a kernel induced feature space," in *Proc. European Symposium on Artificial Neural Networks (ESANN), Bruges, Belgium*, 2002, pp. 1–6.
[13] R. Rosipal, M. Girolami, L. Trejo, and A. Cichocki, "Kernel PCA for feature extraction and de-noising in nonlinear regression," *Neural Computing & Applications*, vol. 10, no. 3, pp. 231–243, 2001.
[14] B. Schölkopf, A. Smola, and K. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural computation*, vol. 10, no. 5, pp. 1299–1319, 1998.
[15] A. Teixeira, A. Tomé, and E. Lang, "Unsupervised feature extraction via kernel subspace techniques," *Neurocomputing*, vol. 74, no. 5, pp. 820–830, 2011.
[16] A. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and Computing*, vol. 14, no. 3, pp. 199–222, 2004.
[17] C. Williams and M. Seeger, "Using the Nyström method to speed up kernel machines," in *Advances in Neural Information Processing Systems 13*. eds. T. K. Leen, T. G. Diettrich, V. Tresp. MIT Press, 2001.
[18] W. Huyer and A. Neumaier, "Global optimization by multilevel coordinate search," *Journal of Global Optimization*, vol. 14, no. 4, pp. 331–355, 1999.