Piecewise smooth system identification in reproducing kernel Hilbert space

Fabien Lauer, Gérard Bloch

Piecewise Smooth System Identification in Reproducing Kernel Hilbert Space

Fabien Lauer and Gérard Bloch

Abstract— **The paper extends the recent approach of Ohlsson and Ljung for piecewise affine system identification to the nonlinear case while taking a clustering point of view. In this approach, the problem is cast as the minimization of a convex cost function implementing a trade-off between the fit to the data and a sparsity prior on the number of pieces. Here, we consider the nonlinear case of piecewise smooth system identification without prior knowledge on the type of nonlinearities involved. This is tackled by simultaneously learning a collection of local models from a reproducing kernel Hilbert space via the minimization of a convex functional, for which we prove a representer theorem that provides the explicit form of the solution. An example of application to piecewise smooth system identification shows that both the mode and the nonlinear local models can be accurately estimated.**

I. Introduction

Hybrid dynamical systems switch between multiple subsystems, either arbitrarily (e.g., due to unobserved external inputs) or according to a partition of the space of the observed variables. This switching behavior prevents their direct identification via classical procedures even for the most simple case of static linear subsystems. The main difficulty comes from the combinatorial nature of the problem, where one has to simultaneously assign the data points to the different subsystems, i.e., determine the modes, and to estimate a model for each one of these subsystems.

Related work. Formally, hybrid system identification has been considered in the literature either as a switching regression or a piecewise affine (PWA) regression problem with an ARX set of regressors (see [1] for details). Consequently, most approaches, e.g., [2], [3], [4], [5], [6], [7], focus on regression in a supervised learning framework, i.e., a context where both the input and the output of the function to be estimated are available in the data. Such approaches often treat the classification of the data points into consistent groups corresponding to the subsystems as a by-product of the estimation.

On the contrary, this paper considers an unsupervised learning framework by focusing on the classification problem inherent in hybrid system identification: determining the active mode for each data point. The rationale is that once this classification is obtained, then classical estimation techniques provide the solution to the regression problem. This point of view was originally considered in the seminal work of [8] on PWA regression, where local models were first estimated independently at each data point and then clustered into a small number of modes. More recently, the sum-of-norms approach of [9] allowed for the simultaneous estimation of all local models with a sparsity prior on the number of different models, which directly yields the classification. Similar approaches were also proposed in [10], [11], [12] for the segmentation of ARX systems, i.e., the special (and much easier) case where the data points are ordered in time and the modes are defined as intervals of the time axis. Other works with a focus on the classification subproblem include the Bayesian approach of [13], the method of [14] based on Dempster-Shafer theory, the adaptation of the k-means clustering algorithm to switching regression discussed in [15] and the geometric approach of [16].

Recent nonlinear extensions of the methods typically involve a "kernelization" step where linear models are replaced by linear combinations of kernel functions (see [17] for an introduction to kernel functions). In particular, [18] extends the continuous optimization approach of [6], and [19], [20] extend the sparse optimization approach of [5]. But these focus on arbitrarily switched regression and are not suitable for piecewise smooth (PWS) systems, as will be emphasized in the example of Sect. IV-A. In addition, [18] and [19] assume a restricted function class for the models based on a finite combination of basis functions fixed a priori. Note that [21] extends the sum-of-norms approach of [10] without such restrictions on the model, but only for the segmentation of nonlinear ARX systems from data ordered in time.

Contribution. We extend the sum-of-norms approach of [9], originally proposed for PWA systems, to PWS system identification, with a focus on the clustering point of view. This yields the first approach based on convex optimization that is effective for piecewise smooth regression with unknown nonlinearities. More precisely, we consider a convex problem formulation, in which the cost functional is a trade-off between a data fitting term and a regularization term controlling the complexity of the global model via two aspects: the number of pieces or submodels and the complexity of each of the submodels. Then, the framework is derived for local models that belong to a reproducing kernel Hilbert space, which provides sufficient flexibility to learn PWS systems with arbitrary (but smooth) nonlinearities. The explicit form of the solution to this learning problem is obtained thanks to a new representer theorem. To complete the method, we show how to cluster the resulting functions in such spaces and obtain the classification of the data.

Paper organization. The paper first presents the PWS system identification problem in Sect. II with the general ap-

F. Lauer is with LORIA, Université de Lorraine, CNRS, Inria, Nancy, France fabien.lauer@loria.fr

G. Bloch is with CRAN, Université de Lorraine, CNRS, Nancy, France gerard.bloch@univ-lorraine.fr

proach in Sect. II-A and straightforward instances in Sect. II-B. Then, Sect. III sets up the learning problem in reproducing kernel Hilbert space and provides its solution in Sect. III-A, while details on the clustering of functions in such spaces are given in Sect. III-B. Finally, numerical examples are presented in Sect. IV and conclusions in Sect. V.

II. Piecewise smooth system identification

Consider the class of systems in input–output ARX form, i.e., with regressors $\boldsymbol{x}_i = [y_{i-1}, \ldots, y_{i-n_y}, u_i, \ldots, u_{i-n_u}]^T \in \mathcal{X} \subset \mathbb{R}^p$, that are PWS. These systems take the form

$$\begin{cases} q_i = g(\boldsymbol{x}_i), \\ y_i = h_{q_i}(\boldsymbol{x}_i) + v_i, \end{cases} \tag{1}$$

where the discrete state (or mode) q_i is determined by a partition of the regression space $\mathcal{X}$, represented in the above by the function $g : \mathcal{X} \to \{1, \ldots, n\}$, and the output $y_i \in \mathbb{R}$ is computed within each region of this partition by one of the smooth (i.e., of class C^∞) functions $\{h_j\}_{j=1}^n$ implementing the dynamics of the subsystems, plus a noise term v_i.

Given a data set $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$ generated by (1), the PWS system identification problem is to estimate n, g and $\{h_j\}_{j=1}^n$. However, since different triplets $\left(n, g, \{h_j\}_{j=1}^n\right)$ can generate the exact same data, this problem is intractable unless we introduce additional knowledge or desired properties for the model. In the following, we assume that we have access to a fairly good estimate $\hat{n}$ of the number of modes.

A. General approach

The most difficult subtask in PWS system identification is to compute the estimates $\hat{q}_i$ of the mode q_i at all data points, from which both the partitioning function g and the models h_j in (1) can be easily estimated. This point of view leads to the overall procedure suggested by [8] and depicted in Algorithm 1. In this method, Step 1 estimates the mode via two sub-steps, in which local models associated to data points are first learned and then clustered. After that, the partition of the regression space corresponding to g can be obtained in Step 2 by standard *supervised* classification tools, such as support vector machines [22], applied to the data $\boldsymbol{x}_i$ labeled by $\hat{q}_i$. In Step 3, standard (i.e., non-hybrid) regression or system identification methods applied independently within each mode yield estimates of the models h_j. Therefore, we focus on Step 1 and the estimation of the mode q_i in the following.

Step 1.a) For the purpose of the identification, we consider the following alternative model of (1):

$$y_i = f_i(\boldsymbol{x}_i) + e_i,$$

where a local model f_i is assigned to each data point to predict y_i with an error e_i. These local models are to be estimated within a function class $\mathcal{F} \subset \mathbb{R}^{\mathcal{X}}$ (where $\mathbb{R}^{\mathcal{X}}$ is the set of functions from $\mathcal{X}$ into $\mathbb{R}$), which can encode the prior knowledge on the structure of the models $\{h_j\}_{j=1}^n$, as discussed in Sect. II-B, or be sufficiently large to contain satisfactory approximations of arbitrary functions, as in Sect. III.

Algorithm 1 Overall procedure

focus of this paper

1) Estimate the modes $\{\hat{q}_i\}_{i=1}^N$:
   - 1.a) learn the local models $\{f_i\}_{i=1}^N$;
   - 1.b) cluster the f_i's into $\hat{n}$ groups to estimate the labels $\hat{q}_i$.

classical problems with known methods

2) Estimate $\hat{g}$ from $\{(\boldsymbol{x}_i, \hat{q}_i)\}_{i=1}^N$.
3) Estimate $\hat{h}_j$ from $\{(\boldsymbol{x}_i, y_i) : \hat{q}_i = j\}$, $j = 1, \ldots, \hat{n}$.

This learning phase relies on the following observation. Since the optimal clustering of the f_i's corresponds to a partition of $\mathcal{X}$ induced by g, we are interested in finding a set of f_i's such that for two points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ close to each other, f_i and f_j should be the same function. This is obtained by minimizing the variations over the set $\{f_i\}_{i=1}^N$, i.e., the sum of distances between local models of neighboring points. Formally, we consider the following learning problem:

$$\min_{\{f_i \in \mathcal{F}\}_{i=1}^N} \sum_{i=1}^N \ell(y_i - f_i(\boldsymbol{x}_i)) + \lambda \sum_{i=1}^N \sum_{j=1}^N w_{ij} d_{\mathcal{F}}(f_i, f_j), \tag{2}$$

where the error is measured by a loss function $\ell : \mathbb{R} \to \mathbb{R}^+$, such as the squared loss, $\ell(e) = e^2$, or the absolute loss, $\ell(e) = |e|$, $d_{\mathcal{F}}$ denotes a suitable distance measure in $\mathcal{F}$ and the w_{ij}'s are precomputed weights. The role of these weights is to encode the assumptions on the piecewise nature of the model, as in [9]: $w_{ij} > 0$ for neighboring points and $w_{ij} = 0$ for points that are presumably not in the same region. A typical choice is to compute the weights as $w_{ij} = 1/\|\boldsymbol{x}_j - \boldsymbol{x}_i\|_2$ if $\boldsymbol{x}_j$ is one of the N_n neighbors of $\boldsymbol{x}_i$ and 0 otherwise. Thus, the interaction between local models decreases when the distance between their base points increases.

Remark 1: In a sparse optimization framework as the one developed in [9], the second term in the cost functional of (2) can be seen as a convex surrogate for the number of different local models. In this context, this number corresponds to the so-called ℓ_0-pseudo-norm of the vector $\boldsymbol{d}$ of all weighted distances $w_{ij} d_{\mathcal{F}}(f_i, f_j)$, denoted $\|\boldsymbol{d}\|_0$ and whose direct minimization is intractable. However, a large body of work, notably in the field of compressed sensing [23], [24], shows that the ℓ_1-norm $\|\boldsymbol{d}\|_1$, appearing in (2), can be used as a surrogate for $\|\boldsymbol{d}\|_0$ in minimization problems.

In addition, as suggested in [9], reweighting procedures, similar to the ones used in compressed sensing [25], [26], can be applied to enhance the sparsity of the solution, i.e., to decrease the number of different functions f_i. For instance, the reweighting of [25] leads to the initialization $w_{ij}^0 = w_{ij}$, with w_{ij} computed as before, and $w_{ij}^k = w_{ij}/(d_{\mathcal{F}}(f_i, f_j) + \epsilon)$ at iteration k for a small $\epsilon > 0$. The selective ℓ_1-minimization scheme of [26] can also be applied with the same initialization by setting the weight of the maximal weighted distance to 0 at each iteration.

Step 1.b) In our approach, the mode estimates $\hat{q}_i$ correspond to the labels obtained by clustering the set of functions, $\{f_i\}_{i=1}^N$, resulting from Step 1.a of Algorithm 1. This clustering can take two different forms. In the ideal case, the number $\hat{n}$ of different functions f_i is small and consistent with the expected number of modes. Then, the labels $\hat{q}_i$ belong to $\{1, \dots, \hat{n}\}$ and are simply set such that $\hat{q}_i = \hat{q}_j$ if and only if $f_i = f_j$.

In order to reduce the number of modes, or to improve the labeling in case the set of f_i's is noisy and too many different functions are obtained, a clustering algorithm such as k-means can be applied to the f_i's, i.e., to the data mapped into $\mathcal{F}$, with a fixed number of groups $\hat{n}$. Indeed, even with noisy f_i's, the different groups are expected to be well separated in $\mathcal{F}$ due to the variational regularization imposed on the f_i's in Step 1.a.

B. Straightforward instances

1) Piecewise affine regression: In PWA regression, we consider linear models, i.e., $\mathcal{F} = \{f : f(\boldsymbol{x}) = \boldsymbol{x}^T\boldsymbol{\theta}, \boldsymbol{\theta} \in \mathbb{R}^p\}$, and assume the last component of $\boldsymbol{x}_i$ to be 1 for affine models. In this case, the distance $d_{\mathcal{F}}$ can simply be computed as the Euclidean distance between the parameter vectors: $d_{\mathcal{F}}(f_i, f_j) = \|\boldsymbol{\theta}_i - \boldsymbol{\theta}_j\|_2$. For typical loss functions, Problem (2) can be solved in this setting by Second Order Cone Programming (SOCP) general purpose solvers, such as [27], with the parameter vectors $\{\boldsymbol{\theta}_i\}_{i=1}^N$ as variables. Then, the estimated parameter vectors can be easily clustered by k-means in $\mathbb{R}^p$ in case this yields too many modes.

2) Explicit nonlinearities: A nonlinear extension of the PWA case can be obtained in a straightforward manner by preprocessing the data with a nonlinear feature map $\phi : \boldsymbol{x} \mapsto \phi(\boldsymbol{x}) = [\phi_1(\boldsymbol{x}), \phi_2(\boldsymbol{x}), \dots]^T \in \mathbb{R}^d$. This corresponds to the learning problem (2) using the function class $\mathcal{F}_\phi = \{f : f(\boldsymbol{x}) = \boldsymbol{\theta}^T\phi(\boldsymbol{x}),\ \boldsymbol{\theta} \in \mathbb{R}^d\}$, in a parametric setting similar to the PWA case, i.e., with $d_{\mathcal{F}_\phi}$ computed as a norm of the parameter vector difference.

However, this formulation clearly suffers from major limitations due to the requirement of an explicit nonlinear map: the basis functions ϕ_j must be fixed or known, and in limited number to avoid the curse of dimensionality.

To circumvent these difficulties, the following takes a different path by assuming local models that are smooth functions of a Reproducing Kernel Hilbert Space (RKHS).

III. PWS regression in RKHS

We now briefly introduce the required background on kernel functions and associated function spaces.

Definition 1 (Real-valued positive definite function): A real-valued function K on $\mathcal{X}^2$ is called a positive definite function if it is symmetric and $\forall N \in \mathbb{N},\ \forall \{\boldsymbol{x}_i\}_{i=1}^N \in \mathcal{X}^N,\ \forall \{a_i\}_{i=1}^N \in \mathbb{R}^N,\ \sum_{i=1}^N \sum_{j=1}^N a_i a_j K(\boldsymbol{x}_i, \boldsymbol{x}_j) \geq 0$.

Definition 2 (Reproducing kernel Hilbert space): Let $(\mathcal{H}, \langle\cdot,\cdot\rangle_{\mathcal{H}})$ be a Hilbert space of real-valued functions on $\mathcal{X}$ with inner product $\langle\cdot,\cdot\rangle_{\mathcal{H}}$. A real-valued function K on $\mathcal{X}^2$ is a reproducing kernel of $\mathcal{H}$ if and only if

1) $\forall \boldsymbol{x} \in \mathcal{X},\ K(\boldsymbol{x}, \cdot) \in \mathcal{H}$;
2) $\forall \boldsymbol{x} \in \mathcal{X},\ \forall f \in \mathcal{H},\ \langle f, K(\boldsymbol{x}, \cdot)\rangle_{\mathcal{H}} = f(\boldsymbol{x})$ (reproducing property).

A Hilbert space of real-valued functions which possesses a reproducing kernel is called a reproducing kernel Hilbert space (RKHS).

Note that the reproducing property of K implies in particular that $\langle K(\boldsymbol{x}, \cdot), K(\boldsymbol{x}', \cdot)\rangle_{\mathcal{H}} = K(\boldsymbol{x}, \boldsymbol{x}')$. In the following, we shall refer to such functions satisfying Definition 1 as kernel functions. The Moore–Aronszajn theorem states that for any kernel function K, there is one and only one RKHS with K as reproducing kernel [28].

Let K be a kernel function as in Definition 1 and $(\mathcal{H}, \langle\cdot,\cdot\rangle_{\mathcal{H}})$ the associated RKHS. Then, the class of functions $\mathcal{H}$ can be written as

$$\mathcal{H} = \left\{ f \in \mathbb{R}^{\mathcal{X}} : f = \sum_{i=1}^{m} \alpha_i K(\boldsymbol{x}_i, \cdot), \right. \\ \left. m \in \mathbb{N}, \alpha_i \in \mathbb{R}, \boldsymbol{x}_i \in \mathcal{X}, \|f\|_{\mathcal{H}} < +\infty \right\},$$

where $\|f\|_{\mathcal{H}} = \sqrt{\langle f, f\rangle_{\mathcal{H}}}$ is the norm in $\mathcal{H}$ induced by the inner product defined for two functions, $f = \sum_{i=1}^m \alpha_i K(\boldsymbol{x}_i, \cdot)$ and $g = \sum_{i=1}^{m'} \beta_i K(\boldsymbol{x}'_i, \cdot)$, as

$$\langle f, g\rangle_{\mathcal{H}} = \sum_{i=1}^{m} \sum_{j=1}^{m'} \alpha_i \beta_j K(\boldsymbol{x}_i, \boldsymbol{x}'_j).$$

A typical kernel function is the Gaussian kernel, $K(\boldsymbol{x}, \boldsymbol{x}') = \exp(-\|\boldsymbol{x} - \boldsymbol{x}'\|_2^2/2\sigma^2)$, for which $\mathcal{H}$ consists of all infinitely differentiable (i.e., smooth) functions of $\mathcal{X} \to \mathbb{R}$. With such a kernel, $\mathcal{H}$ enjoys the so-called universal approximation capacity, i.e., any continuous function can be arbitrarily well approximated by a function in $\mathcal{H}$.

A. Learning problem and its solution

We now focus on Step 1.a of Algorithm 1, in which we consider local models f_i as functions of an RKHS $\mathcal{H}$ and set $\mathcal{F} = \mathcal{H}$ in the learning problem (2). In this case, in order to avoid overfitting the noise, the complexity of the f_i's should be controlled at the local level and not only at the global level of their number. This is related to the smoothness assumption on the functions h_j in (1) and the fact that $\mathcal{H}$ is typically a very flexible function class, possibly including an f that can perfectly fit noisy data. Thus, in addition to the variational regularization aiming at the minimization of the number of different local models, we penalize the complexity of the local models.

More precisely, we consider the standard measure of complexity for functions f_i in an RKHS, as employed for instance in support vector machines [22], [29], i.e., the RKHS squared norm, $\|f_i\|_{\mathcal{H}}^2$. This norm also naturally serves to define the distance between functions of the RKHS as $d_{\mathcal{H}}(f_i, f_j) = \|f_i - f_j\|_{\mathcal{H}}$. Thus, the learning problem

becomes

$$\min_{\{f_i\in\mathcal{H}\}_{i=1}^N} \sum_{i=1}^N \ell(y_i - f_i(\boldsymbol{x}_i)) + \gamma \sum_{i=1}^N \|f_i\|_{\mathcal{H}}^2 + \lambda \sum_{i=1}^N \sum_{j=1}^N w_{ij}\|f_i - f_j\|_{\mathcal{H}}, \quad (3)$$

where $\gamma > 0$ is the parameter that controls the complexity of the functions f_i, while λ controls the complexity of the global model in terms of the number of different local models f_i.

A fundamental difference between (3) and the versions of (2) using parametrized models as discussed in section II-B is that the variables $\{f_i\}_{i=1}^N$ are functions of $\mathcal{H}$ and not vectors of $\mathbb{R}^p$. However, a finite-dimensional formulation of (3) is obtained thanks to the following theorem, which extends the representer theorem originally proposed in [30] and generalized in [31]. Indeed, the formulations in [30], [31] do not include multiple functions to be learned, nor allow for variational terms involving the norm of the difference between these functions.

Theorem 1 (Representer theorem): Any solution $\{f_i^*\}_{i=1}^N$ to (3) is a collection of functions that all lie in the span of the kernel functions taken at the data points:

$$\forall i \in \{1,\dots,N\}, \quad f_i^* \in \text{Span}\left(\{K(\boldsymbol{x},\cdot)\}_{\boldsymbol{x}\in X}\right),$$

where $X = \{\boldsymbol{x}_i\}_{i=1}^N$ contains all regression vectors.

Proof: See the Appendix. ∎

By applying Theorem 1, we can replace the f_i's in (3) by the linear combinations of kernel functions

$$f_i = \sum_{k=1}^N \alpha_{ik} K(\boldsymbol{x}_k, \cdot),$$

with weights $\alpha_{ik} \in \mathbb{R}$ to be estimated and function values computed via the reproducing property of K (see Definition 2) as

$$f_i(\boldsymbol{x}_i) = \langle f_i, K(\boldsymbol{x}_i,\cdot)\rangle_{\mathcal{H}} = \sum_{k=1}^N \alpha_{ik} \langle K(\boldsymbol{x}_k,\cdot), K(\boldsymbol{x}_i,\cdot)\rangle_{\mathcal{H}} = \sum_{k=1}^N \alpha_{ik} K(\boldsymbol{x}_k, \boldsymbol{x}_i).$$

This yields the finite-dimensional and convex optimization problem

$$\min_{\{\boldsymbol{\alpha}_i\in\mathbb{R}^N\}_{i=1}^N} \sum_{i=1}^N \ell\left(y_i - \boldsymbol{\alpha}_i^T \boldsymbol{k}_i\right) + \gamma \sum_{i=1}^N \boldsymbol{\alpha}_i^T \boldsymbol{K} \boldsymbol{\alpha}_i + \lambda \sum_{i=1}^N \sum_{j=1}^N w_{ij} \sqrt{(\boldsymbol{\alpha}_i - \boldsymbol{\alpha}_j)^T \boldsymbol{K} (\boldsymbol{\alpha}_i - \boldsymbol{\alpha}_j)}, \quad (4)$$

where $\boldsymbol{\alpha}_i = [\alpha_{i1},\dots,\alpha_{iN}]^T$, $\boldsymbol{K}$ is the Gram matrix of the kernel K, i.e., $\forall (k,i) \in \{1,\dots,N\}^2$, $(\boldsymbol{K})_{ki} = K(\boldsymbol{x}_k, \boldsymbol{x}_i)$, and $\boldsymbol{k}_i$ is its ith column.

For the squared or absolute loss functions, Problem (4) can be rewritten in a SOCP form, suitable for general purpose solvers, by computing the Cholesky factorization of the Gram matrix, $\boldsymbol{K} = \boldsymbol{R}^T\boldsymbol{R}$.

Remark 2: In the case where the index i provides the ordering of the data in time, replacing the variational term in (3) with $\sum_{i=1}^{N-1} \|f_{i+1} - f_i\|_{\mathcal{H}}$ yields a method similar in spirit to [21] for segmenting ARX systems over time.

B. Clustering functions in RKHS

We now turn to Step 1.b of Algorithm 1. After solving (4), we have a set of N functions $f_i \in \mathcal{H}$, with the expectation that only a few different functions (corresponding to the true number of modes) are obtained. However, the ℓ_1 relaxation discussed in Remark 1 might not yield a truly sparse distance vector $\boldsymbol{d}$, in which case the functions f_i are tightly clustered around a few mean functions. Then, the recovery of the data classification amounts to a well separated clustering problem in the function space $\mathcal{H}$, that can be tackled as follows.

Consider the classical k-means algorithm which clusters feature vectors, $\boldsymbol{\varphi}_i \in \mathbb{R}^p$, by minimizing the sum of squared Euclidean distances,

$$\sum_{k=1}^{n_G} \sum_{\boldsymbol{\varphi}_i \in G_k} d(\boldsymbol{\varphi}_i, \overline{\boldsymbol{\varphi}}_k)^2 = \sum_{k=1}^{n_G} \sum_{\boldsymbol{\varphi}_i \in G_k} \|\boldsymbol{\varphi}_i - \overline{\boldsymbol{\varphi}}_k\|_2^2,$$

with respect to the means $\overline{\boldsymbol{\varphi}}_k$ of n_G groups $\{G_k\}_{k=1}^{n_G}$ in $\mathbb{R}^p$. In order to cluster functions of the RKHS, the distances must be computed in $\mathcal{H}$ with $d_{\mathcal{H}}(f_i, \overline{f}_k)^2 = \|f_i - \overline{f}_k\|_{\mathcal{H}}^2 = \langle f_i - \overline{f}_k, f_i - \overline{f}_k\rangle_{\mathcal{H}}$. However, since all f_i belong to the span of $\{K(\boldsymbol{x}_i,\cdot)\}_{i=1}^N$, the mean functions also belong to this subspace of $\mathcal{H}$ and can be expressed as $\overline{f}_k = \sum_{j=1}^N \overline{\alpha}_{kj} K(\boldsymbol{x}_j,\cdot)$. Using the factorization $\boldsymbol{K} = \boldsymbol{R}^T\boldsymbol{R}$, this simplifies the computations as

$$d_{\mathcal{H}}(f_i, \overline{f}_k)^2 = (\boldsymbol{\alpha}_i - \overline{\boldsymbol{\alpha}}_k)^T \boldsymbol{K} (\boldsymbol{\alpha}_i - \overline{\boldsymbol{\alpha}}_k) = \|\boldsymbol{R}(\boldsymbol{\alpha}_i - \overline{\boldsymbol{\alpha}}_k)\|_2^2.$$

Thus, k-means can be applied in a straightforward manner with the Euclidean distance and feature vectors $\boldsymbol{\varphi}_i = \boldsymbol{R}\boldsymbol{\alpha}_i \in \mathbb{R}^N$ in order to cluster the f_i's and produce the final classification of the data points.

IV. Examples

A. Illustrative example

Figure 1 presents an example of PWS regression with data generated by $y_i = \sin(x_i \mod 2) + v_i$ with x_i uniformly distributed in $[0,4]$ and a Gaussian noise $v_i \sim \mathcal{N}(0, 0.05^2)$. Here, the proposed approach yields the correct classification of the $N = 100$ data points into 2 groups; and Figure 2 shows that, within each group, the functions f_i solution to (3) are close to each other and many are identical (only 7 different functions are obtained). Note that, in Step 3 of Algorithm 1 (not shown here) one could easily estimate more accurate local models than the mean functions $\overline{f}_1$, $\overline{f}_2$ from the correctly classified data. For a comparison, Fig. 1 shows the results of the method of [20], which also estimates the local models in an RKHS via convex optimization. However, this method is dedicated to switching regression and does not deal with the fact that, in PWS regression, a single nonlinear model can easily approximate many points of

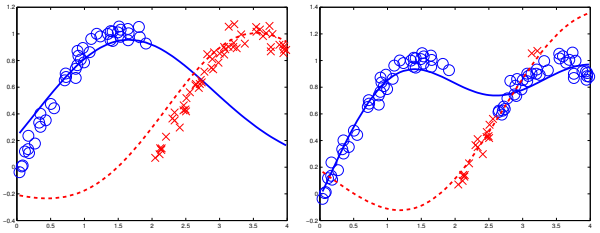

Fig. 1. Example of PWS regression. Left: data classified (as $\circ$ or $\times$) by the proposed method and the mean functions $\overline{f}_1$ (–) and $\overline{f}_2$ (- -). Right: classification and models obtained by the switching nonlinear regression method of [20].

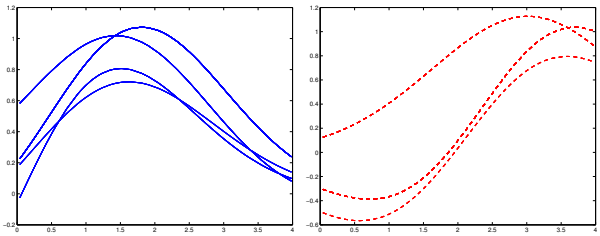

Fig. 2. The $N = 100$ functions f_i obtained by solving (3), classified in two groups (left and right plots) by the method of Sect. III-B. As desired, in the solution of (3) many of the one hundred f_i's are the same and they are easily separated in two groups.

multiple groups. The plot in the right-hand side of Fig. 1 illustrates this issue: the model of the first group (plain line), though very smooth, also fits half of the data of the second group, thus yielding many classification errors and leaving insufficient data for an accurate estimation of the second model.

B. Piecewise smooth system identification example

Consider the PWS dynamical system:

$$y_i = \begin{cases} \dfrac{y_{i-1}y_{i-2}(y_{i-1}+2.5)}{1+y_{i-1}^2+y_{i-2}^2} + u_{i-1} + v_i, & \text{if } y_{i-1}+y_{i-2} \geq 0, \\ \dfrac{2y_{i-1}\text{sinc}(y_{i-1}+u_{i-1})}{1+y_{i-2}^2 u_{i-1}^2} + v_i, & \text{otherwise,} \end{cases} \tag{5}$$

where $\text{sinc}(x) = \sin(\pi x)/(\pi x)$. A trajectory of 300 points is generated by (5) with a uniformly distributed input $u_i \in [-1, 1]$ and a Gaussian noise $v_i \sim \mathcal{N}(0, 0.1^2)$. The first $N = 200$ points are used as the training set and the last 100 form the test set. For the identification, the system (5) is assumed completely unknown except for the set of regressors, $\boldsymbol{x}_i = [y_{i-1}, y_{i-2}, u_{i-1}]^T$, and the number of modes, $n = 2$. The proposed method is applied with a Gaussian kernel ($\sigma = 0.5$), $\lambda = 1$ and $\gamma = 0.5$ to classify the training data and compute the labels $\hat{q}_i$. Then, the nonlinear submodels, $\hat{h}_1(\boldsymbol{x})$ and $\hat{h}_2(\boldsymbol{x})$, are estimated by support vector regression [32] applied to each subset $\{(\boldsymbol{x}_i, y_i) : \hat{q}_i = k\}$, $k = 1, 2$. The switching boundary is estimated by a linear support vector classifier [22] trained on $\{(\boldsymbol{x}_i, \hat{q}_i)\}_{i=1}^N$ to output the mode, $\hat{g}(\boldsymbol{x}) \in \{1, 2\}$. Note that this choice of final regression method and of classifier is purely arbitrary and that many other options are available for these tasks.

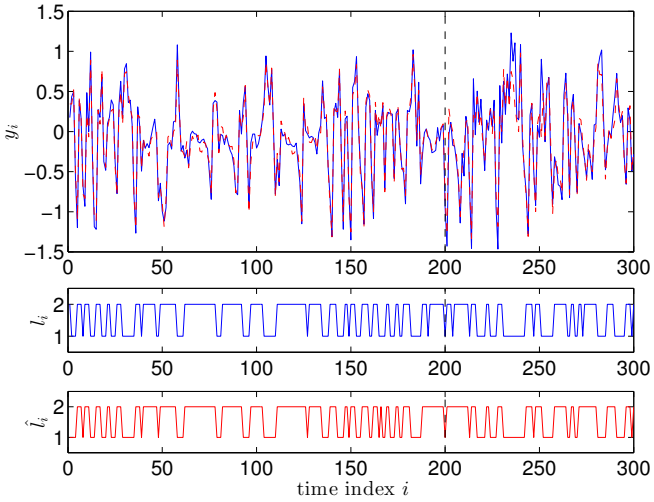

Fig. 3. Top: noisy data (plain line) and predictions (dashed line) for an output trajectory of system (5). Middle: corresponding true mode sequence. Bottom: estimated mode sequence. The vertical dashed line delimits the training set (left) from the test set (right).

Figure 3 shows the output trajectory $\{y_i\}$ and the predicted one $\{\hat{y}_i\}$. On the training set, the predictions are computed by $\hat{y}_i = \hat{h}_{\hat{q}_i}(\boldsymbol{x}_i)$ with a mean squared error, MSE $= 1/N \sum_{i=1}^N (y_i - \hat{y}_i)^2$, equal to 0.007. The bottom plots show that the mode is correctly estimated by the labels $\hat{q}_i$ for most training data points with an error rate of 2%. On the test set, the mode is given by the classifier $\hat{g}$ and $\hat{y}_i = \hat{h}_{\hat{g}(\boldsymbol{x}_i)}(\boldsymbol{x}_i)$, leading to a classification error rate of 4% and MSE = 0.0379. This, and the fact that estimating the submodels h_1, h_2 and the classifier g from the true mode q_i instead of $\hat{q}_i$ yields similar MSE and error rate on the test set, shows the effectiveness of the proposed approach.

V. Conclusions

The paper proposed an approach based on convex optimization for the identification of piecewise smooth systems. The core of the method relies on learning a collection of functions from an RKHS by minimizing a trade-off between the fit to the data and the complexity of the model (number of pieces and complexity of each piece). The solution to this learning problem was obtained thanks to a representer theorem. This led to the first convex optimization-based algorithm that is effective for piecewise smooth regression with arbitrary nonlinearities, as the few previous approaches dealt with arbitrarily switched nonlinear regression and proved unsuited for the piecewise case.

Appendix

Proof: [of Theorem 1] Let $\mathcal{S} = \text{Span}\left(\{K(\boldsymbol{x}, \cdot)\}_{\boldsymbol{x} \in X}\right)$ denote the subspace of interest in $\mathcal{H}$ and $\mathcal{S}^c$ its orthogonal complement. Then, every function $f_i \in \mathcal{H}$ can be decomposed into a sum of two orthogonal components as

$$f_i = u_i + v_i, \quad u_i \in \mathcal{S}, \; v_i \in \mathcal{S}^c, \quad \mathcal{S} \perp \mathcal{S}^c. \tag{6}$$

Note that in this case, the function values, $f_i(\boldsymbol{x}_i) = u_i(\boldsymbol{x}_i) + v_i(\boldsymbol{x}_i)$, only depend on the components u_i, since

$$v_i \in \mathcal{S}^c \Rightarrow v_i \perp \mathcal{S} \Rightarrow \langle v_i, K(\boldsymbol{x}_i, \cdot)\rangle_{\mathcal{H}} = 0$$

and, by the reproducing property of K (see Definition 2),

$$v_i(\boldsymbol{x}_i) = \langle v_i, K(\boldsymbol{x}_i, \cdot)\rangle_{\mathcal{H}} = 0.$$

This implies $f_i(\boldsymbol{x}_i) = u_i(\boldsymbol{x}_i)$, and thus that $\ell(y_i - f_i(\boldsymbol{x}_i)) = \ell(y_i - u_i(\boldsymbol{x}_i))$, $i = 1, \ldots, N$, in the data term of (3).

Regarding the complexity-control term, note that for all $f_i \in \mathcal{H}$,

$$\begin{aligned}\|f_i\|_{\mathcal{H}}^2 &= \langle f_i, f_i\rangle_{\mathcal{H}} = \langle u_i, u_i\rangle_{\mathcal{H}} + \langle v_i, v_i\rangle_{\mathcal{H}} + 2\langle u_i, v_i\rangle_{\mathcal{H}} \\ &= \|u_i\|_{\mathcal{H}}^2 + \|v_i\|_{\mathcal{H}}^2,\end{aligned}$$

due to the orthogonality between u_i and v_i.

For the variational term, we have, $\forall (i,j) \in \{1, \ldots, N\}^2$,

$$\begin{aligned}\|f_i - f_j\|_{\mathcal{H}} &= \|u_i - u_j + v_i - v_j\|_{\mathcal{H}} \\ &= \sqrt{\|u_i - u_j\|_{\mathcal{H}}^2 + \|v_i - v_j\|_{\mathcal{H}}^2 + 2\langle u_i - u_j, v_i - v_j\rangle_{\mathcal{H}}}.\end{aligned}$$

Besides, $\langle u_i - u_j, v_i - v_j\rangle_{\mathcal{H}} = \langle u_i, v_i\rangle_{\mathcal{H}} - \langle u_i, v_j\rangle_{\mathcal{H}} - \langle u_j, v_i\rangle_{\mathcal{H}} + \langle u_j, v_j\rangle_{\mathcal{H}} = -\langle u_i, v_j\rangle_{\mathcal{H}} - \langle u_j, v_i\rangle_{\mathcal{H}}$. But since all u_i belong to $\mathcal{S}$ and all v_i are orthogonal to that subspace, we have $\forall (i,j) \in \{1, \ldots, N\}^2$, $\langle u_i, v_j\rangle_{\mathcal{H}} = 0$, leading to

$$\|f_i - f_j\|_{\mathcal{H}} = \sqrt{\|u_i - u_j\|_{\mathcal{H}}^2 + \|v_i - v_j\|_{\mathcal{H}}^2}$$

and

$$\|f_i - f_j\|_{\mathcal{H}} \geq \|u_i - u_j\|_{\mathcal{H}}.$$

Let $J\left(\{f_i\}_{i=1}^N\right)$ denote the cost functional of (3). Then, for any set of functions, $\{f_i\}_{i=1}^N \in \mathcal{H}^N$, decomposed as in (6), the partial results on the data, regularization and variational terms lead to

$$\begin{aligned}&J\left(\{f_i\}_{i=1}^N\right) - J\left(\{u_i\}_{i=1}^N\right) \\ &= \lambda \sum_{i=1}^N w_{ij}\left(\|f_i - f_j\|_{\mathcal{H}} - \|u_i - u_j\|_{\mathcal{H}}\right) + \gamma \sum_{i=1}^N \|v_i\|_{\mathcal{H}}^2 \\ &\geq 0.\end{aligned}$$

In addition, if $v_i \neq 0$ for some $i \in \{1, \ldots, N\}$, then $\|v_i\|_{\mathcal{H}} > 0$ and

$$J(\{f_i\}_{i=1}^N) > J(\{u_i\}_{i=1}^N).$$

Hence, any minimizer, $\{f_i^*\}_{i=1}^N$, of (3) admits a decomposition (6) with $v_i^* = 0$, $i = 1, \ldots, N$, which concludes the proof. ∎

REFERENCES

[1] S. Paoletti, A. L. 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] J. Roll, A. Bemporad, and L. Ljung, "Identification of piecewise affine systems via mixed-integer programming," *Automatica*, vol. 40, no. 1, pp. 37–50, 2004.

[4] A. Bemporad, A. Garulli, S. Paoletti, and A. Vicino, "A bounded-error approach to piecewise affine system identification," *IEEE Transactions on Automatic Control*, vol. 50, no. 10, pp. 1567–1580, 2005.

[5] L. Bako, "Identification of switched linear systems via sparse optimization," *Automatica*, vol. 47, no. 4, pp. 668–677, 2011.

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

[7] F. Lauer, V. L. Le, and G. Bloch, "Learning smooth models of nonsmooth functions via convex optimization," in *Proc. of the IEEE Int. Workshop on Machine Learning for Signal Processing (MLSP), Santander, Spain*, 2012.

[8] G. Ferrari-Trecate, M. Muselli, D. Liberati, and M. Morari, "A clustering technique for the identification of piecewise affine systems," *Automatica*, vol. 39, no. 2, pp. 205–217, 2003.

[9] H. Ohlsson and L. Ljung, "Identification of switched linear regression models using sum-of-norms regularization," *Automatica*, vol. 49, no. 4, pp. 1045–1050, 2013.

[10] H. Ohlsson, L. Ljung, and S. Boyd, "Segmentation of ARX-models using sum-of-norms regularization," *Automatica*, vol. 46, no. 6, pp. 1107–1111, 2010.

[11] N. Ozay, M. Sznaier, C. Lagoa, and O. Camps, "A sparsification approach to set membership identification of switched affine systems," *IEEE Transactions on Automatic Control*, vol. 57, no. 3, pp. 634–648, 2012.

[12] D. Piga and R. Tóth, "An SDP approach for ℓ_0-minimization: Application to ARX model segmentation," *Automatica*, vol. 49, no. 12, pp. 3646–3653, 2013.

[13] A. L. Juloski, S. Weiland, and W. Heemels, "A Bayesian approach to identification of hybrid systems," *IEEE Transactions on Automatic Control*, vol. 50, no. 10, pp. 1520–1533, 2005.

[14] K. Boukharouba, L. Bako, and S. Lecoeuche, "Identification of piecewise affine systems based on Dempster-Shafer theory," in *Proc. of the 15th IFAC Symp. on System Identification, Saint-Malo, France*, 2009, pp. 1662–1667.

[15] F. Lauer, "Estimating the probability of success of a simple algorithm for switched linear regression," *Nonlinear Analysis: Hybrid Systems*, vol. 8, pp. 31–47, 2013, supplementary material available at http://www.loria.fr/~lauer/klinreg/.

[16] V. L. Le, F. Lauer, and G. Bloch, "Identification of linear hybrid systems: a geometric approach," in *Proc. of the American Control Conference (ACC), Washington, DC, USA*, 2013, pp. 830–835.

[17] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

[18] V. L. Le, G. Bloch, and F. Lauer, "Reduced-size kernel models for nonlinear hybrid system identification," *IEEE Transactions on Neural Networks*, vol. 22, no. 12, pp. 2398–2405, 2011.

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

[20] V. L. Le, F. Lauer, L. Bako, and G. Bloch, "Learning nonlinear hybrid systems: from sparse optimization to support vector regression," in *Proc. of the 16th ACM Int. Conf. on Hybrid Systems: Computation and Control (HSCC), Philadelphia, PA, USA*, 2013, pp. 33–42.

[21] T. Falck, H. Ohlsson, L. Ljung, J. A. K. Suykens, and B. De Moor, "Segmentation of times series from nonlinear dynamical systems," in *Proc. of the 18th IFAC World Congress, Milan, Italy*, 2011, pp. 13 209–13 214.

[22] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[23] D. L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[24] E. J. Candès, "Compressive sampling," in *Proc. of the Int. Congress of Mathematicians, Madrid, Spain*, 2006, pp. 1433–1452.

[25] E. J. Candès, M. B. Wakin, and S. P. Boyd, "Enhancing sparsity by reweighted ℓ_1 minimization," *Journal of Fourier Analysis and Applications*, vol. 14, no. 5-6, pp. 877–905, 2008.

[26] V. L. Le, F. Lauer, and G. Bloch, "Selective ℓ_1 minimization for sparse recovery," *IEEE Transactions on Automatic Control*, vol. 59, no. 11, 2014.

[27] E. Andersen and K. Andersen, "The MOSEK interior point optimizer for linear programming: an implementation of the homogeneous algorithm," *High Performance Optimization*, vol. 33, pp. 197–232, 2000.

[28] A. Berlinet and C. Thomas-Agnan, *Reproducing Kernel Hilbert Spaces in Probability and Statistics*. Kluwer Academic Publishers, Boston, 2004.

[29] I. Steinwart and A. Christmann, *Support Vector Machines*. Springer, 2008.

[30] G. Kimeldorf and G. Wahba, "Some results on Tchebycheffian spline functions," *Journal of Mathematical Analysis and Applications*, vol. 33, no. 1, pp. 82–95, 1971.

[31] B. Schölkopf, R. Herbrich, and A. Smola, "A generalized representer theorem," in *COLT/EuroCOLT*, 2001, pp. 416–426.

[32] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and Computing*, vol. 14, no. 3, pp. 199–222, 2004.