

Shrinkage Strategies for Structure Selection and Identification of Piecewise Affine Models

Valentina Breschi and Manas Mejari

Abstract—We propose two *optimization-based* heuristics for structure selection and identification of *Piecewise Affine* (PWA) models with exogenous inputs. The first method determines the number of affine sub-models assuming known model order of the sub-models, while the second approach estimates the model order for a given number of affine sub-models. Both approaches rely on the use of *regularization-based* shrinking strategies, that are exploited within a *coordinate-descent* algorithm. This allows us to estimate the structure of the PWA models along with its model parameters. Starting from an over-parameterized model, the key idea is to alternate between an identification step and structure refinement, based on the sparse estimates of the model parameters. The performance of the presented strategies is assessed over two benchmark examples.

I. INTRODUCTION

Many real-world systems and phenomena are characterized by different operating conditions, among which they can commute smoothly and/or abruptly. In such scenarios, linear models are usually insufficient to accurately describe the behavior of the underlying system and one must resort to more complex model structures to attain satisfactory approximations. However, many learning frameworks lead to models that are rather hard to analyze and use, especially for control purposes.

A good trade-off between complexity, approximation power and intelligibility can be attained by considering *piecewise affine* (PWA) maps. Indeed, it is well known that PWA functions have universal approximation properties [13], [7], while featuring a rather simple structure, since they are characterized by a finite collection of affine sub-models defined over a polyhedral partition of the regressor space. Thus, PWA models can be analyzed and exploited for control design with state-of-the-art techniques (see *e.g.*, [5]).

Although appealing because of the aforementioned properties, PWA models are challenging to learn from data. Indeed, to identify the local models and estimate the associated regions of the regressor space one should know the operating condition of the real systems at each time step, which is usually not available a priori. The learning problem thus entails not only the identification of sub-models parameters and the computation of a polyhedral partition of the regressor space, but it also requires one to solve an unsupervised clustering problem, since each sample has to be assigned to one of the local models.

This work has been accepted to the 59-th IEEE Conference on Decision and Control.

V. Breschi is with Dipartimento di Elettronica e Informazione, Politecnico di Milano, Milano, Italy. valentina.breschi@polimi.it

M. Mejari is with IDSIA Dalle Molle Institute for Artificial Intelligence, SUPSI-USI, Manno, Switzerland. manas.mejari@supsi.ch

Several heuristics have been proposed over the years to deal with this demanding learning task, among which we mention the bounded-error method proposed in [4], the clustering-based procedures in [2], [8], [9], [15], the mixed-integer programming based approaches in [14], [17], the Bayesian method in [11] and the sparse optimization techniques proposed in [1], [16]. However, most of these methods rely on the prior knowledge on the number of operating conditions of the true system and of its order, which might not be available in practice. In [4], the number of sub-models is chosen by solving a *minimum* number of *feasible subsystems* (MIN PFS) problem, which is, in turn, formulated by relying on a bound for the estimation error. This parameter is unlikely to be available in practice, thus becoming a tuning knob for the algorithm. A refinement strategy is also proposed therein, where the number of sub-models is further reduced by merging local models that share the same parameters and discarding the ones that are associated to “few” data-points. By exploiting sparse optimization, the methods presented in [1], [16] both allows for the selection of the number of modes. Specifically, the approach proposed in [16] considers an over-parameterized model and, then, exploits L_1 -regularization to enforce the local parameter estimates to be similar. Instead, in [1], L_1 -regularization is used to iteratively search for the sparsest vector of fitting errors to cluster the data-point, estimate local models and decide on their number. To the best of our knowledge, only in [1], it is provided a possible solution for the selection of the sub-models structure, which relies on the introduction of an additional L_1 -regularization term on the parameters.

As an alternative, in this paper we present two *regularization-based* strategies that allow the partial selection of the PWA model structure along with parameter estimation. These techniques exploit the flexibility of the general purpose approach for jump model fitting presented in [3], which can easily incorporate regularization-based shrinking strategies. In particular, we propose a heuristic to estimate the number of local models by using L_∞ regularization on the sub-models parameters, while accounting for the cardinality of the associated clusters to refine the optimization-driven choice. In the second approach, the local model structure is estimated via the *elastic net*. We remark that the proposed strategy is inspired by the one briefly discussed in [1], but it inherits the benefits of elastic net over simple lasso regularization. The overall identification procedure with partial structure selection requires the alternation between a learning stage and a step in which the model is refined.

Once the model structure is determined, the final model is re-estimated by neglecting the shrinkage regularization terms. Although the proposed strategies are quite general and can be straightforwardly extended to other classes of PWA models, in this work we focus on *piecewise affine autoregressive models with exogenous inputs* (PWARX).

The paper is organized as follows. The problem of learning a PWARX model with partially unknown structure is introduced in Section II and the core algorithm used to address this learning task is summarized in Section III. The proposed regularization-based strategies are then discussed in Section IV, and their effectiveness is shown in Section V on a numerical example and an experimental case study. Conclusions and directions for future work are finally discussed in Section VI.

II. PROBLEM FORMULATION

Consider the following *dynamical* data-generating system, whose instantaneous response $y_t \in \mathcal{Y} \subseteq \mathbb{R}$ to an input $u_t \in \mathcal{U} \subseteq \mathbb{R}$ is given by

$$y_t = f_o(x_t) + e_t^o, \quad (1a)$$

where $e_t^o \in \mathbb{R}$ is a zero-mean additive white noise independent of the *regressor* $x_t \in \mathcal{X} \subseteq \mathbb{R}^{n_x}$. Let the regressor be given by the following collection of past input/output samples

$$x_t = [y_{t-1} \ \cdots \ y_{t-n_a} \ u_{t-1} \ \cdots \ u_{t-n_b}]', \quad (1b)$$

with $n_a, n_b \in \mathbb{R}^+$ denoting the *possibly unknown* order of the system. The *unknown* function $f_o : \mathcal{X} \rightarrow \mathbb{R}$ characterizing the underlying regressor/output relationship is assumed to be *nonlinear* and *possibly discontinuous*, but not necessarily piecewise affine.

We aim at finding a PWA approximation f of the unknown function f_o , based on a set of inputs $U = \{u_t\}_{t=1}^T$ and output samples $Y = \{y_t\}_{t=1}^T$, generated by (1a)-(1b). We remark that, due to the universal approximation properties of PWA maps [13], [7], PWA function f can approximate f_o with an arbitrary accuracy.

The approximating PWA function $f : \mathcal{X} \rightarrow \mathbb{R}$ is defined as

$$f(x_t) = \begin{cases} \tilde{x}_t' \theta_{y,1}, & \text{if } x_t \in \mathcal{X}_1, \\ \vdots \\ \tilde{x}_t' \theta_{y,K}, & \text{if } x_t \in \mathcal{X}_K, \end{cases} \quad (2)$$

with $\tilde{x}_t = [x_t' \ 1]'$. This *piecewise affine autoregressive* model with *exogenous inputs* (PWARX) is a collection of $K \in \mathbb{N}$ local affine models, characterized by the parameters $\Theta_y = (\theta_{y,1}, \dots, \theta_{y,K})$ and defined over a *complete* polyhedral partition¹ $\{\mathcal{X}_k\}_{k=1}^K$ of the regressor space $\mathcal{X}$. We stress that all the local models share the same structure by definition.

¹ $\{\mathcal{X}_k\}_{k=1}^K$ is a complete polyhedral partition of $\mathcal{X}$ if $\bigcup_{k=1}^K \mathcal{X}_k = \mathcal{X}$ and $\mathcal{X}_i \cap \mathcal{X}_j = \emptyset$, for $i \neq j$, $i, j = 1, \dots, K$, with $\mathring{\mathcal{X}}_i$ denoting the interior of the i -th polyhedron $\mathcal{X}_i$.

We characterize the partition through a piecewise affine separator function $\phi : \mathbb{R}^{n_x} \rightarrow \mathbb{R}$, defined as

$$\phi(x) = \max_{k=1, \dots, K} \{\theta'_{x,k} \tilde{x}\}, \quad (3)$$

where $\theta'_{x,k} \in \mathbb{R}^{n_x+1}$. Thus, a polyhedron $\mathcal{X}_i$ can be described by the following linear inequalities

$$\mathcal{X}_i = \{x \in \mathbb{R}^{n_x} : (\theta_{x,i} - \theta_{x,j})' \tilde{x} \geq 0, j = 1, \dots, K, j \neq i\}. \quad (4)$$

Therefore, learning a PWARX model from data amounts at: (i) choosing the number of affine sub-models K ; (ii) estimating the parameters Θ_y of the local models, (iii) finding the parameters $\Theta_x = (\theta_{x,1}, \dots, \theta_{x,K})$ characterizing the PWA separator in (3), and, eventually, (iv) selecting the order of the sub-models, namely n_a and n_b , when it is unknown.

In this work, we split model structure selection into two separate tasks by assuming that *either* the number of local models K *or* the sub-model order is fixed a priori. In the first task, we estimate (Θ_y, Θ_x, K) for fixed n_a, n_b , while in the second one we fix K and retrieve $(\Theta_y, \Theta_x, n_a, n_b)$ from data.

III. LEARNING PWARX MODEL

It is well known that learning PWA approximating map is quite challenging, since this problem is NP-hard [12] even for an a-priori fixed model structure. Indeed, it requires the concurrent solution of local linear regressions, a classification and a clustering problem. To cope with this, we recall the *modular* optimization problem presented in [3, Section 2.2.1.] for PWA regression and we summarize the steps of the *coordinate-descent* approach proposed therein to solve the problem at hand.

Let $X = \{x_t\}_{t=1}^T$ be the sequence of regressors constructed from the available input/output dataset $\{U, Y\}$. Let $s_t \in \{1, \dots, K\}$ be the *latent* variable, that indicates which local model is associated with the t -th regressor/output pair $\{\tilde{x}_t, y_t\}$, such that the sequence $\mathcal{S} = \{s_t\}_{t=1}^T$ comprises the label associated to each data point.

Within the optimization-based framework of [3], learning a generic *jump* model entails the solution of the following optimization problem:

$$\begin{aligned} \min_{\Theta_y, \Theta_x, \mathcal{S}} \quad & \sum_{t=1}^T [\ell_y(y_t, x_t, \theta_{y,s_t}) + \rho \ell_x(x_t, \theta_{x,s_t})] + \\ & + \sum_{k=1}^K [r_y(\theta_{y,k}) + \rho r_x(\theta_{x,k})], \end{aligned} \quad (5a)$$

In particular, for the PWARX model class (2)-(3), the *loss function* $\ell_y : \mathcal{X} \times \mathcal{Y} \times \mathbb{R}^{n_x+1} \rightarrow \mathbb{R} \cup \{+\infty\}$ is chosen as the squared *local fitting error*, i.e.,

$$\ell_y(y_t, x_t, \theta_{y,s_t}) = (y_t - \tilde{x}_t' \theta_{y,s_t})^2, \quad (5b)$$

and the loss function $\ell_x : \mathcal{X} \times \mathbb{R}^{n_x+1} \rightarrow \mathbb{R} \cup \{+\infty\}$ is

$$\ell_x(x_t, \theta_{x,s_t}) = \sum_{\substack{j=1 \\ j \neq s_t}}^K \max \{0, (\theta_{x,j} - \theta_{x,s_t})' \tilde{x}_t + 1\}^2, \quad (5c)$$

so to account for the violations of the inequalities in (4). The *regularizers* $r_y : \mathbb{R}^{n_x+1} \rightarrow \mathbb{R} \cup \{+\infty\}$ and $r_x : \mathbb{R}^{n_x+1} \rightarrow \mathbb{R} \cup \{+\infty\}$ act on the parameters Θ_y and Θ_x , characterizing the local models and the PWA separator respectively. The positive hyper-parameter ρ balances the effect of each term on the overall cost. From (5a), it is clear that the discrete sequence $\mathcal{S}$ couples the regularized linear regressions and the multi-category discrimination problems, that have to be solved to find Θ_y and Θ_x , respectively. Nonetheless, once $\mathcal{S}$ is fixed, the parameters Θ_y and Θ_x can be found independently from one another. On the other hand, for a given Θ_y and Θ_x , the discrete sequence $\mathcal{S}$ can be computed using *dynamic programming* (DP) [6] technique in a computationally efficient manner.

Based on the above considerations, we exploit the *coordinate-descent* approach proposed in [3] by alternatively solving problem (5a) with respect to Θ_y and Θ_x , for a fixed sequence $\mathcal{S}$, and optimizing with respect to $\mathcal{S}$ for fixed Θ_y and Θ_x . The three step procedure tailored for the solution of problem (5a) is summarized in Algorithm 1 where, with a slight abuse of notation, we denote

$$\ell_y(Y, X, \Theta_y, \mathcal{S}) = \sum_{t=1}^T \ell_y(y_t, x_t, \theta_{y,s_t}), \quad (6a)$$

$$\ell_x(X, \Theta_x, \mathcal{S}) = \sum_{t=1}^T \ell_x(x_t, \theta_{x,s_t}), \quad (6b)$$

$$r_y(\Theta_y) = \sum_{k=1}^K r_y(\theta_{y,k}), \quad r_x(\Theta_x) = \sum_{k=1}^K r_x(\theta_{x,k}). \quad (6c)$$

Given the structure of the local models, the problem ought to be solved at step 1.1 can be split into K distinct regularized linear regression problems, one for each local models. We remark that each sub-model is updated by using only the data points that are associated to it, according to the estimated mode sequence $\mathcal{S}^{i-1}$. The estimation of the PWA separator at step 1.2 entails the solution of a multi-category discrimination problem, which can be efficiently computed with the Newton-like method proposed in [8]. The mode sequence is finally updated at step 1.3 via DP [6], as detailed in [3], by neglecting the regularization terms as they are independent of $\mathcal{S}$. Note that, the tunable parameter $\rho \in \mathbb{R}^+$ is not considered when optimizing with respect to Θ_y and Θ_x (steps 1.1-1.2), but it is exploited when updating the mode sequence to trade-off between the fitting error and the cost of choosing a certain polyhedral region.

Remark 1 (Inference hints): Once a PWARX model is retrieved by running Algorithm 1, for each new regressor x_t the active local model can be computed as

$$\hat{s}_t = \max_{k=1, \dots, K} \{(\theta_{x,k}^*)' \tilde{x}_t\}, \quad (7a)$$

and, accordingly, the corresponding output can be obtained as follows:

$$\hat{y}_t = \tilde{x}_t' \theta_{y, \hat{s}_t}^*. \quad (7b)$$

Algorithm 1 [Coordinated descent for learning PWARX [3]]

Input: Data $\{Y, X\}$; initial mode sequence $\mathcal{S}^0$; $\rho \in \mathbb{R}^+$.

1. **for** $i = 1, \dots$ **do**

▷ Estimate local functions

1.1. $\Theta_y^i \leftarrow \arg \min_{\Theta_y} \ell_y(Y, X, \Theta_y, \mathcal{S}^{i-1}) + r_y(\Theta_y)$

▷ Estimate linear separators

1.2. $\Theta_x^i \leftarrow \arg \min_{\Theta_x} \ell_x(X, \Theta_x, \mathcal{S}^{i-1}) + r_x(\Theta_x)$

▷ Estimate mode sequence

1.3. $\mathcal{S}^i \leftarrow \arg \min_{\mathcal{S}} \ell_y(Y, X, \Theta_y^i, \mathcal{S}) + \rho \ell_x(X, \Theta_x^i, \mathcal{S})$

2. **until** $\mathcal{S}^i = \mathcal{S}^{i-1}$

Output: Local models $\Theta_y^* = \Theta_y^i$; PWA separator $\Theta_x^* = \Theta_x^i$; sequence $\mathcal{S}^* = \mathcal{S}^i$.

IV. REGULARIZATION-BASED STRATEGIES FOR MODEL STRUCTURE SELECTION

Suppose that either the number of sub-models characterizing the PWA map in (2) or their local structure is *unknown* and assume that bounds on the values of these unknowns are given by the maximum tolerable complexity of the estimated model.

To determine the unknown PWARX model structure (*i.e.*, either the number of sub-models K or their model order n_a, n_b), the regularization terms in the objective function of (5a) can be shaped so to use well-known *regularization-based* shrinkage techniques to choose the structure of the estimated PWARX model in (2), directly from data. This allows to avoid grid searches, which usually require an exhaustive exploration of the unknown parameter space.

In this work, starting from an over-parameterized model, we aim at devising iterative structure selection strategies that rely on runs of Algorithm 1 with tailored regularization terms. In particular, we focus on the selection of the *group regularization* $r_y(\Theta_y)$, which acts on the parameters of the local models.

Two different choices of $r_y(\Theta_y)$ are made to tackle the problems of estimating the number of sub-models K and the model order n_a, n_b respectively. For both problems, we fix the regularizer $r_x(\Theta_x)$ acting on the parameters Θ_x of the PWA separator as follows

$$r_x(\Theta_x) = \lambda \sum_{k=1}^K \|\theta_{x,k}\|_2^2, \quad (8)$$

with $\lambda \in \mathbb{R}^+$ being a hyper-parameter to be tuned. This choice makes the multi-category discrimination optimization at step 1.2 of Algorithm 1 strictly convex, thus reducing the complexity of the resulting model. In the following, we introduce two different strategies based on the choice of $r_y(\Theta_y)$, respectively tailored to handle the selection of K , for fixed local model structure, and the choice of n_a and n_b , for a given number of local models. In the latter scenario, we assume that $n_a, n_b \geq 1$, namely that the response of

the system at a given instant always depends at least on the previous input/output pair. In both cases, we assume that the dataset is relatively *balanced*, namely we expect that the local models are associated to a reasonable number of samples for the overall model to be identifiable with sufficient accuracy.

A. Number of local model selection via L_∞ regularization

In this subsection, we outline the strategy to estimate number of local models K for a fixed model orders n_a, n_b . Let us assume that at most K_{max} local models in (2) are sufficient to accurately describe f_o in (1a) and K_{max} is known. As a first step towards the definition of a strategy to select the number of sub-models, we introduce a regularization term $r_y(\Theta_y)$ that shrinks the entire parameter vectors $\theta_{y,k}$ towards zero, when the corresponding sub-models are redundant. To this end, we choose $r_y(\Theta_y)$ to be a convex combination of *group Tikhonov* regularization and a term penalizing the mixed $L_{1-\infty}$ -norm of the parameter vectors, *i.e.*,

$$r_y(\Theta_y) = \mu \sum_{k=1}^K \|\theta_{y,k}\|_2^2 + \nu \sum_{k=1}^K \|\theta_{y,k}\|_\infty, \quad (9)$$

where $\mu, \nu \in \mathbb{R}$ are hyper-parameters to be tuned. The term penalizing the sum of the infinity norms (*i.e.*, the mixed $L_{1-\infty}$ -norm), ensures that, at the solution, the vector $\theta_{y,k}$ is either full or have all elements nearly identical to zero. Thus, only the component of $\theta_{y,k}$ with the highest absolute value affects the cost. The additional L_2 regularization term is introduced for a better numerical conditioning of the problem when the parameter vector is not shrunk towards zero.

Since the shrinkage strategy might lead to parameter vectors with rather small elements, that are still not identically zero, we label a local model as *redundant* if the associated parameter vector $\theta_{y,k}$ satisfies the following inequality:

$$\|\theta_{y,k}\|_\infty \leq \delta, \quad (10)$$

where $\delta \in \mathbb{R}^+$ is a relatively small hyper-parameter to be tuned.

Accordingly, starting from an over-parameterized model with $K = K_{max}$, the procedure for the automatic selection of the model structure is summarized in Algorithm 2. At the beginning of each iteration, Algorithm 1 is run with the regularization term in (9). Therefore, the estimation of the local models for a fixed mode sequence $\mathcal{S}$ (step 1.1 of Algorithm 1) entails the solution of the following problem:

$$\min_{\Theta_y} \sum_{t=1}^T \sum_{k=1}^K (y_t - \tilde{x}_t' \theta_{y,s_t})^2 + \mu \sum_{k=1}^K \|\theta_{y,k}\|_2^2 + \nu \sum_{k=1}^K \|\theta_{y,k}\|_\infty. \quad (11)$$

Once the new PWARX model is retrieved (*i.e.*, once algorithm 1 has stopped), redundant sub-models are detected by checking the condition in (10) for the estimated parameters Θ_y^* and the number of local models is updated accordingly (see steps 1.2 and 1.4.1). Specifically, at step 1.2, the set of indexes $\mathcal{R}^j$ corresponding to the redundant sub-models is updated based on the criterion (10) and at 1.4.1, the number of local models K^j is updated by subtracting the

Algorithm 2 [Selection of the number of sub-models K]

Input: Dataset $\{Y, X\}$ of length T ; initial mode sequence $\mathcal{S}^0$; $\rho, \mu, \nu, \lambda, \delta \in \mathbb{R}^+$; maximum number of local models $K^0 = K_{max}$.

1. **for** $j = 1, \dots$ **do**
 - 1.1. **run** Algorithm 1 for $K = K^{j-1}$ and $r_y(\Theta_y)$ as in (9);
 - 1.2. $\mathcal{R}^j \leftarrow \{k \in \{1, \dots, K^{j-1}\} : \|\theta_{y,k}^*\|_\infty \leq \delta\}$;
 - 1.3. **if** $\mathcal{R}^j = \emptyset$ **do**
 - 1.3.1. $\mathcal{C}_\emptyset^j \leftarrow \{k \in \{1, \dots, K^{j-1}\} : \#\mathcal{C}_k \leq 0.01 \cdot T\}$;
 - 1.3.2. $K^j \leftarrow K^{j-1} - K_\emptyset^j$, with $K_\emptyset^j = \#\mathcal{C}_\emptyset^j$;
 - 1.4. **else**
 - 1.4.1. $K^j \leftarrow K^{j-1} - K_R^j$, with $K_R^j = \#\mathcal{R}^j$;
 - 1.5. **end if**;
 2. **until** $K^j = K^{j-1}$
-

Output: Number of sub-models $K^* = K^j$.

number of redundant sub-models (*i.e.* the cardinality $\#\mathcal{R}^j$ of set $\mathcal{R}^j$) from the number of sub-models estimated K^{j-1} at the previous iteration. If no redundant model is detected ($\#\mathcal{R}^j = 0$), at steps 1.3.1-1.3.2 we further check the clusters $\{\mathcal{C}_k\}_{k=1}^K$ induced by $\mathcal{S}$, *i.e.*,

$$\mathcal{C}_k = \{x_t \in X : s_t^* = k\}, \quad k = 1, \dots, K \quad (12)$$

and we exploit the assumption that the dataset is balanced to remove as many local models as the number of *almost empty* clusters, namely sets with cardinality $\#\mathcal{C}_k$ smaller or equal than 1% of the overall dataset. This procedure is iterated until K does not vary over two consecutive iterations.

Once Algorithm 2 is terminated a new instance of Algorithm 1 is run with $r_y(\Theta_y)$ defined as in (9) with $\nu = 0$ to identify the final PWARX model with number of local models K^* .

B. Model order selection via the Elastic Net

In this section, we present the algorithm to estimate the model orders n_a and n_b of the local affine sub-models, assuming the number of modes K and upper bounds on the parameters n_a and n_b are known.

To this end, the regularization term $r_y(\Theta_y)$ needs to be selected in order *not* to shrink the entire parameter vector to zero (as done in the previous subsection), but only its *redundant* components. Accordingly, we exploit *elastic net* regularization by setting $r_y(\Theta_y)$ as:

$$r_y(\Theta_y) = \mu \sum_{k=1}^K \|\theta_{y,k}\|_2^2 + \nu \sum_{k=1}^K \|\theta_{y,k}\|_1, \quad (13)$$

with $\mu, \nu \in \mathbb{R}^+$ being hyper-parameters to be tuned to balance the effect of the two regularization terms. This choice allows us to still enforce parameter selection, thanks to the shrinkage property of the *group lasso* term which shrinks to zero all the parameters associated to unnecessary terms in the regressor, while the presence of the quadratic penalty leads to a more stable and efficient regularization path [18].

Algorithm 3 [Selection of the sub-models structure]

Input: Dataset $\{Y, U\}$ of length T ; initial mode sequence $\mathcal{S}^0$; $\rho, \mu, \nu, \lambda, \delta \in \mathbb{R}^+$; maximum model complexity $n_a^0 = n_{a,max}$ and $n_b^0 = n_{b,max}$.

1. **for** $j = 1, \dots, \mathbf{do}$

1.1. **construct** $X = \{x_t\}_{t=1}^T$ for $n_a = n_a^{j-1}$ and $n_b = n_b^{j-1}$;

1.2. **run** Algorithm 1 for $r_y(\Theta_y)$ in (13);

1.3. $n_a^j \leftarrow \max_k \#\{i \in \{1, \dots, n_a^{j-1}\} : \|\theta_{y,k}^*\|_i \geq \delta\}$;

1.4. $n_b^j \leftarrow \max_k \#\{i \in \{1, \dots, n_b^{j-1}\} : \|\theta_{y,k}^*\|_{i+n_a^{j-1}} \geq \delta\}$;

2. **until** $n_a^j = n_a^{j-1}$ **and** $n_b^j = n_b^{j-1}$

Output: Order of the sub-models $n_a^* = n_a^j$ and $n_b^* = n_b^j$.

As the regularization strategy might not lead to parameters that are exactly identical to zero, we remove all terms in the regressor but the one associated to the affine term, for which

$$\|[\theta_{y,k}]_i\| \leq \delta, \quad i = 1, \dots, n_a + n_b \quad (14)$$

where $[\theta_{y,k}]_i$ denotes the i -th component of $\theta_{y,k}$ and $\delta \in \mathbb{R}^+$ is a hyper-parameter to be tuned, accounting for the fact that the regularization strategy might not lead to parameters that are exactly identical to zero. With this we imply that the affine term should never be neglected.

Note that, by testing the condition in (14) independently for all local models, we do not consider that they share the same structure and thus we cannot enforce this property on the PWA map in (2). Nonetheless, a possible heuristic to enforce this property on the PWA map (2) is to select n_a and n_b as follows:

$$\hat{n}_a = \max_{k=1, \dots, K} \#\{i \in \{1, \dots, n_a\} : \|\theta_{y,k}\|_i \geq \delta\}, \quad (15a)$$

$$\hat{n}_b = \max_{k=1, \dots, K} \#\{i \in \{1, \dots, n_b\} : \|\theta_{y,k}\|_{n_a+i} \geq \delta\}, \quad (15b)$$

which evaluates condition in (14) for all local models and fixes the model order of all local models to that of the sub-model with the *least* shrinkage. We stress that this approach might be *conservative*, since model structure selection relies on the characteristics of at most 2 sub-models.

Algorithm 3 summarizes the proposed strategy for estimating the model orders of the local sub-models. Similar to the Algorithm 2, this strategy relies on recurrent iterations of Algorithm 1 run with $r_y(\Theta_y)$ defined as in (13). We start from an over-parameterized model with parameters $n_{a,max}$ and $n_{b,max}$ and run Algorithm 3 till the model order is invariant within two consecutive iterations. The model order is then updated at steps 1.3 and 1.4, according to the conditions given in (15). Note that, since the model order is iteratively refined, at the beginning of every new run the dataset has to be updated along with the model structure, by redefining $\{x_t\}_{t=1}^T$ according to the definition in (1b) (see step 1.1 of Algorithm 3).

Once the model structure has been retrieved, Algorithm 1 is run once more for n_a^* and n_b^* to obtain the PWARX model, imposing $\nu = 0$ in (13).

Remark 2: For a proper choice of the hyper-parameters μ, ν , the number of iterations required in Algorithms 2 and 3 might be ideally reduced to 1. Nevertheless, the iterative nature of Algorithms 2 and 3 is expected to result in a good selection of the model structure, even when the hyper-parameters are not tuned to optimality. In this way, the burden due to cross-validation/grid search for tuning the hyper-parameters can be reduced. ■

V. CASE STUDIES

In this section, the performance of the proposed model selection strategies is assessed on two benchmark examples, taken from [4]. The first academic example involves the identification of the PWARX system considered in [4], while the second case study involves modeling the dynamics of a placement process in a pick-and-place machine using experimental data [11]. In both case studies, the model structure selection and identification procedures are performed starting from over-parametrized models with $K_{max} = 10$ or $n_{a,max} = n_{b,max} = 10$, respectively. The tuning parameters μ and ν in the regularization terms (9) and (13) are selected so that $\mu \in (0, 1)$ and

$$\nu = 1 - \mu, \quad (16)$$

respectively. In this way, we reduce the number of tunable parameters, by considering a convex combination of the L_2 and lasso regularizations.

All computations are carried out on a i7 2.8 GHz Intel core processor running MATLAB 2019a. The local regression problem in Algorithm 1 are solved by using the CVX package [10], while the PWA separator is estimated via the computationally efficient batch approach proposed in [8], by exploiting the auxiliary parameters used in the example reported therein. To improve the quality of the solution and reduce its dependence on the initial conditions, Algorithm 1 is always executed for $N = 20$ different initial mode sequences $\mathcal{S}^0$ and the PWARX model is selected as the one leading to the minimum cost.

A. Numerical example: identification of a PWARX system

Consider the following PWARX data-generating system with $K = 3$ operating conditions

$$y_t = \begin{cases} [-0.4 \ 1 \ 1.5] \tilde{x}_t + e_t^o, & \text{if } [4 \ -1 \ 10] \tilde{x}_t < 0, \\ [0.5 \ -1 \ -0.5] \tilde{x}_t + e_t^o, & \text{if } [4 \ -1 \ 10] \tilde{x}_t \geq 0 \ \& \\ & [5 \ 1 \ -6] \tilde{x}_t \leq 0 \\ [-0.3 \ 0.5 \ -1.7] \tilde{x}_t + e_t^o, & \text{if } [5 \ 1 \ -6] \tilde{x}_t > 0, \end{cases} \quad (17)$$

where

$$\tilde{x}_t = [y_{t-1} \ u_{t-1} \ 1]'$$

The system is excited with a sequence of random *i.i.d.* inputs uniformly distributed in the interval $[-4, 4]$ and the corresponding outputs are corrupted by a zero-mean additive noise sequence e_o , uniformly distributed in the interval $[-0.8, 0.8]$. To assess the performance of the proposed structure selection

strategies over different realizations of the training set, we perform a Monte-Carlo simulation with 80 runs. At each Monte-Carlo iteration, a new dataset of length $T = 2000$ is generated with different realizations of the inputs and the measurement noise sequences. The effect of the noise e^o on the measurement output yields an average *signal-to-noise ratio* (SNR),

$$\text{SNR} = 10 \log \frac{\sum_{t=1}^T (y_t - e_t^o)^2}{\sum_{t=1}^T (e_t^o)^2} = 16 \text{ dB}. \quad (18)$$

The hyper-parameters are set to $\lambda = 10^{-3}$, $\rho = 1$ and $\mu = 0.1$ and ν is selected according to (16).

1) *Unknown K* : Let us assume that the model order of the system is known, namely that $n_a = n_b = 1$, but that the number of local models K is unknown. We iterate Algorithm 2 to estimate the appropriate number of modes, with δ in (10) set to 0.01. Throughout the Monte Carlo runs only twice the true number of modes $K^* = 3$ is not retrieved, in which cases the number of sub-models is estimated to be either 4 or 5. The average values and standard deviations of the estimated sub-model parameters obtained over the remaining 78 runs (for which true number of modes $K^* = 3$ has been retrieved) are reported in Table I. These results clearly show that proposed approach allows us to accurately reconstruct the local models. In turn, this implies an accurate reconstruction of the polyhedral partition, as proven by the comparison between the true parameters characterizing the partition *vs* the mean and standard deviation of the ones obtained from the estimated PWA separator in Table II. Overall, the structure of the underlying PWARX system has been reconstructed accurately.

2) *Unknown n_a and n_b* : We now assume that the number of sub-models $K = 3$ is known, while the model orders n_a, n_b of the local sub-models are unknown. Algorithm 3 is iterated over the 80 Monte Carlo runs to estimate n_a and n_b along with the parameters of the PWARX model, with $\delta = 0.01$. As shown in the histogram reported in Figure 1, the model orders of the initial over-parameterized model (with $n_{a,max} = n_{b,max} = 10$) are shrunk and the final model is close in structure to the true one. From Figure 1, it can be seen that the proposed approach detected the true model order $n_a^* = 1$ and $n_b^* = 1$ for about 60% and 40% of the Monte-Carlo runs, respectively. Note that, for about 58% of the runs, a higher order of the input delays $n_b = 2$ is estimated. However, for these cases, as shown in Table III, the values of the corresponding model parameters are close to zero, and are negligible with respect to the others. Therefore, they barely influence the performance of the identified model. We stress that the results shown in Table III are obtained after a final iteration of Algorithm 1 with ν in (13) set to zero, and with the model orders selected by running Algorithm 3. The average and standard deviation of the parameters characterizing the regressor space partition are reported in Table IV, which are also accurately estimated. Therefore, despite the slight error in the reconstruction of the model orders in some of the Monte-Carlo runs, the overall

model structure as well as the estimated parameters match closely with that of the true system.

B. Modeling the dynamics of a pick-and-place process

As a second case study, we test the proposed algorithms to model an electronic component placement process in a pick-and-place machine. The set-up consists of a mounting head carrying an electronic component, that is pushed towards a printed circuit board and released, as soon as the two come in contact [11]. The process is characterized by two main operating modes: one associated to the behavior of the pick-and-place machine when it operates in an unconstrained environment and the other when the mounting head comes in contact with the circuit board.

A set of experimental data collected over an interval of 15 s with a sampling frequency of 400 Hz is available to model this process (see [11] for further details). This dataset comprises samples of the voltage applied to the motor driving the mounting head of the machine and the vertical position of the mounting head, which represent the input u and output y of the process, respectively. The overall data record is then split into two disjoint sets comprising the $T = 4800$ samples associated to the first 12 s of the experiment and the remaining $\tilde{T} = 1200$ samples, that are respectively used to train and validate a PWARX model for the process of interest.

The regularization parameters used in training are respectively selected as $\rho = 2.15 \cdot 10^{-4}$, and $\mu = 10^{-5}$ (see [3]), with ν chosen according to (16), while $\lambda = 10^{-8}$. In validation, the performance attained by simulating the learned PWARX model in open-loop and it is quantitatively assessed via the *best fit rate* (BFR) index, *i.e.*,

$$\text{BFR} = 100 \cdot \max \left\{ 0, 1 - \sqrt{\frac{\sum_{t=1}^{\tilde{T}} (y_t - \hat{y}_t)^2}{\sum_{t=1}^{\tilde{T}} (y_t - \bar{y})^2}} \right\} \% \quad (19)$$

with $\bar{y}$ and $\hat{y}_t$ denoting the average output over the validation set and the reconstructed one, respectively.

1) *Unknown K* : Let n_a and n_b be fixed according to the order of the PWARX model estimated in [4], namely $n_a = n_b = 2$. We run Algorithm 2 with $\delta = 10^{-3}$ in (10), which terminates after 6 iterations and it leads to the selection of $K^* = 3$ modes. The proposed structure selection strategy thus introduces an additional mode with respect to the two main operating conditions of the underlying process. Nonetheless, we drastically reduce the complexity of the model with respect to the initial over-parameterized model, characterized by $K_{max} = 10$ modes. The comparison between the output reconstructed in validation with the learned PWARX model with 3 modes and the actual one is reported in Figure 2(a), along with the sequence of active local models. This result corresponds to a BFR of 81.2%.

2) *Unknown n_a and n_b* : We now assume the number of local models is fixed to $K = 2$ and we run Algorithm 3 to estimate the sub-model orders. We set the threshold for selection in (14) as $\delta = 10^{-3}$. Algorithm 3 terminates after 5 runs and the resulting PWARX model is of the second order,

TABLE I
SUB-MODELS PARAMETERS (UNKNOWN K): TRUE VS ESTIMATED (MEAN $\pm$ STANDARD DEVIATION) OVER 78 MONTE CARLO RUNS.

	modes					
	$k = 1$		$k = 2$		$k = 3$	
	true	mean $\pm$ std	true	mean $\pm$ std	true	mean $\pm$ std
$[\theta_{y,k}^*]_1$	-0.40	-0.40 ± 0.02	0.50	0.50 ± 0.01	-0.30	-0.30 ± 0.02
$[\theta_{y,k}^*]_2$	1.00	1.00 ± 0.01	-1.00	-1.00 ± 0.01	0.50	0.50 ± 0.01
$[\theta_{y,k}^*]_3$	1.50	1.48 ± 0.07	-0.50	-0.50 ± 0.02	-1.70	-1.70 ± 0.04

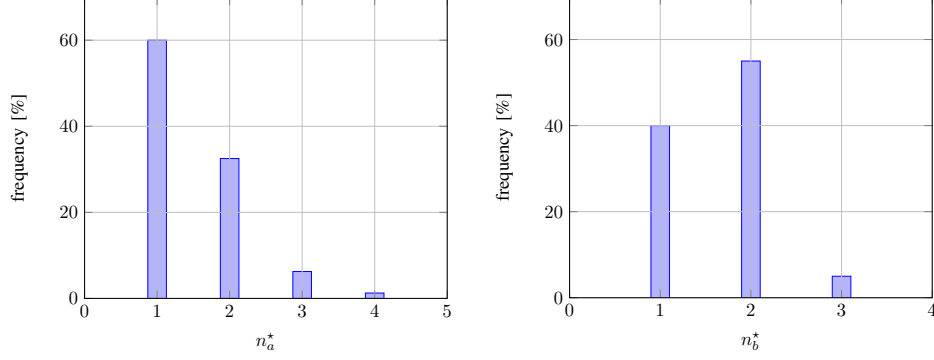


Fig. 1. Numerical example (fixed number of modes): frequency of the selected orders n_a^* and n_b^* .

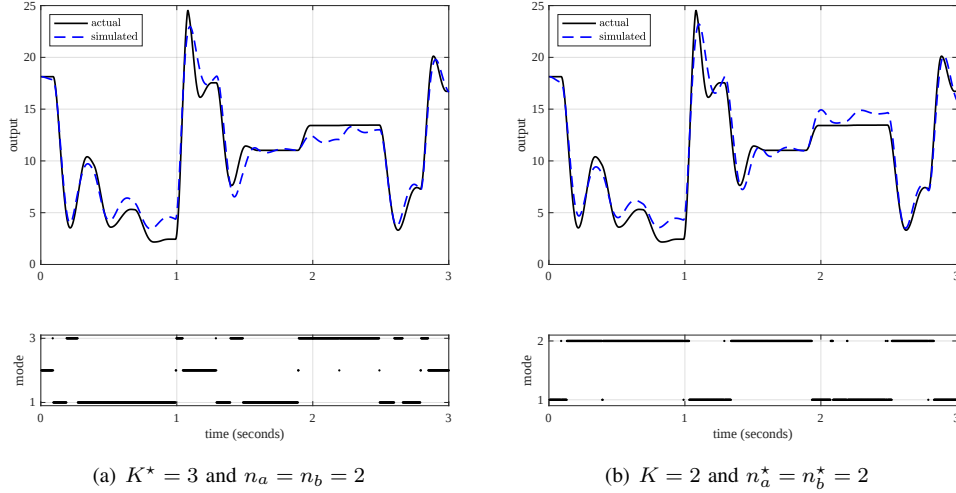


Fig. 2. Pick-and-place machine: simulated vs actual output (top) and reconstructed mode sequence for fixed local model structure (left) and fixed number of sub-models (right).

TABLE II
POLYHEDRAL PARTITION PARAMETERS (UNKNOWN K): TRUE VS ESTIMATED (MEAN $\pm$ STANDARD DEVIATION) OVER 78 MONTE CARLO RUNS.

first separator		second separator	
true	mean $\pm$ std	true	mean $\pm$ std
4.00	4.00 ± 0.06	5.00	5.01 ± 0.08
-1.00	-1.00 ± 0.00	1.00	1.00 ± 0.00
10.00	10.00 ± 0.13	-6.00	-6.01 ± 0.11

namely $n_a^* = n_b^* = 2$. Thus, the complexity of the initial over-parameterized model with $n_{a,max} = n_{b,max} = 10$ is shrunk, leading to a much simpler model, which is same as the one used in [4], [3]. Figure 2(b) shows the comparison

between the simulated and reconstructed output in validation, along with the obtained active mode sequence, which corresponds to a BFR of 79.8%. It can be observed that the estimated PWARX model with simpler model structure is still able to accurately describe the behavior of the placement process.

VI. CONCLUSIONS

The paper has presented a preliminary study on the application of *regularization-based* strategies to select the model structure of a PWARX model. The proposed approaches combine the selective power of existing shrinkage techniques with heuristics, thus providing a tool for data-driven model structure selection. The results obtained in the considered

TABLE III

SUB-MODELS PARAMETERS (UNKNOWN ORDER): TRUE VS ESTIMATED (MEAN±STANDARD DEVIATION) OVER 80 MONTE CARLO RUNS.

		modes					
		$k = 1$		$k = 2$		$k = 3$	
		true	mean±std	true	mean±std	true	mean±std
$[\theta_{y,k}^*]_1$	1	-0.400	-0.406±0.022	0.500	0.503±0.016	-0.300	-0.299±0.014
$[\theta_{y,k}^*]_2$	2	0.000	0.001±0.007	0.000	0.001±0.006	0.000	0.001±0.009
$[\theta_{y,k}^*]_2$	2	0.000	0.000±0.004	0.000	0.000±0.002	0.000	0.001±0.005
$[\theta_{y,k}^*]_2$	2	0.000	0.000±0.000	0.000	0.000±0.000	0.000	0.000±0.003
$[\theta_{y,k}^*]_2$	2	1.000	1.000±0.009	-1.000	-1.000±0.007	0.500	0.501±0.006
$[\theta_{y,k}^*]_2$	2	0.000	0.002±0.011	0.000	-0.001±0.010	0.000	0.000±0.010
$[\theta_{y,k}^*]_2$	2	0.000	-0.001±0.004	0.000	0.000±0.003	0.000	0.000±0.002
$[\theta_{y,k}^*]_3$	3	1.500	1.475 ±0.079	-0.500	-0.497±0.018	-1.700	-1.701±0.059

TABLE IV

POLYHEDRAL PARTITION PARAMETERS (UNKNOWN ORDER): TRUE VS ESTIMATED (MEAN±STANDARD DEVIATION) OVER 80 MONTE CARLO RUNS.

first separator		second separator	
true	mean±std	true	mean±std
4.000	4.001±0.055	5.000	5.020±0.092
0.000	0.000±0.010	0.000	0.000±0.016
0.000	-0.001±0.006	0.000	0.000±0.006
0.000	0.000±0.000	0.000	0.000±0.000
-1.000	-1.000±0.000	1.000	1.000±0.000
0.000	-0.002±0.019	0.000	-0.004±0.028
0.000	0.000±0.004	0.000	-0.001±0.005
10.000	10.000±0.132	-6.00	-6.024±0.131

examples highlight the potential of these strategy.

Future research will be devoted to find less conservative criteria for the selection of the local model structure and to address a complete structure selection problem. The sensitivity of the presented algorithms w.r.t. a set of hyper-parameters will be analyzed and auto-tuning procedures for this parameters will be investigated. Thanks to the generality of the *coordinate-descent* approach on which our strategies rely, future work will also include their extension to other classes of switching models.

REFERENCES

- [1] L. Bako. Identification of switched linear systems via sparse optimization. *Automatica*, 47(4):668 – 677, 2011.
- [2] L. Bako, K. Boukharouba, E. Duviella, and S. Lecoeuche. A recursive identification algorithm for switched linear/affine models. *Nonlinear Analysis: Hybrid Systems*, 5(2):242 – 253, 2011.
- [3] A. Bemporad, V. Breschi, D. Piga, and S. Boyd. Fitting jump models. *Automatica*, 96:11 – 21, 2018.
- [4] A. Bemporad, A. Garulli, S. Paoletti, and A. Vicino. A bounded-error approach to piecewise affine system identification. *IEEE Transactions on Automatic Control*, 50(10):1567–1580, Oct 2005.
- [5] A. Bemporad and M. Morari. Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35(3):407 – 427, 1999.
- [6] D. P. Bertsekas. *Nonlinear programming*. Athena scientific Belmont, 1999.
- [7] L. Breiman. Hinging hyperplanes for regression, classification, and function approximation. *IEEE Transactions on Information Theory*, 39(3):999–1013, 1993.
- [8] V. Breschi, D. Piga, and A. Bemporad. Piecewise affine regression via recursive multiple least squares and multicategory discrimination. *Automatica*, 73:155–162, 2016.
- [9] 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.
- [10] M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming, version 2.1. <http://cvxr.com/cvx>, March 2014.
- [11] A. Juloski, W.P.M.H. Heemels, G. Ferrari-Trecate, R. Vidal, S. Paoletti, and J.H.G. Niessen. Comparison of four procedures for the identification of hybrid systems. In *HSCC*, volume 3414 of *Lecture Notes in Computer Science*, pages 354–369. Springer, 2005.
- [12] F. Lauer. On the complexity of piecewise affine system identification. *Automatica*, 62:148 – 153, 2015.
- [13] J. . Lin and R. Unbehauen. Canonical piecewise-linear approximations. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 39(8):697–699, 1992.
- [14] V. V. Naik, M. Mejari, D. Piga, and A. Bemporad. Regularized moving-horizon piecewise affine regression using mixed-integer quadratic programming. In *2017 25th Mediterranean Conference on Control and Automation (MED)*, pages 1349–1354, 2017.
- [15] H. Nakada, K. Takaba, and T. Katayama. Identification of piecewise affine systems based on statistical clustering technique. *IFAC Proceedings Volumes*, 37(12):179 – 184, 2004.
- [16] H. Ohlsson and L. Ljung. Identification of switched linear regression models using sum-of-norms regularization. *Automatica*, 49(4):1045 – 1050, 2013.
- [17] J. Roll, A. Bemporad, and L. Ljung. Identification of piecewise affine systems via mixed-integer programming. *Automatica*, 40(1):37 – 50, 2004.
- [18] H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.