2016 European Control Conference
June 29 - July 1, 2016. Aalborg, Denmark

Identification of Hybrid and Linear Parameter Varying Models via Recursive Piecewise Affine Regression and Discrimination

Valentina Breschi, Alberto Bemporad and Dario Piga

***Abstract*— Piecewise affine (PWA) regression is a supervised learning method which aims at estimating, from a set of training data, a PWA map approximating the relationship between a set of explanatory variables (commonly called regressors) and continuous-valued outputs. In this paper, we describe a recursive and numerically efficient PWA regression algorithm, and discuss its application to the identification of multi-input multi-output PWA dynamical models in autoregressive form and to the identification of linear parameter-varying models.**

I. Introduction

One of the most challenging problems in regression learning is choosing a proper structure to describe the underlying relation between input (or regressor) vectors $x \in \mathbb{R}^{n_x}$ and continuous-valued target outputs $y \in \mathbb{R}^{n_y}$. On the one hand, too simple models (e.g., linear functions of the independent variable x) might not be flexible enough to describe the input/output relationship. On the other hand, complex model structures tend to be computationally difficult and to overfit the training data.

Motivated by the need of simple and flexible model structures, algorithms for computing *PieceWise Affine* (PWA) approximations of nonlinear regression functions have been developed, and extended to the identification of *PieceWise Affine autoRegressive with eXogenous inputs* (PWARX) systems (see [14] for a literature overview). Among the developed algorithms, we mention the bounded-error methods [7], [13], the clustering-based procedures [2], [9], [11], the mixed integer quadratic programming approach [18], the Bayesian method [10], and the sparse optimization method [12]. Some discussions on the identifiably of PWARX systems are provided in [15], [16]. All these algorithms are executed in a batch fashion, and thus they are not suitable for an on-line implementation, apart from the method in [2].

This paper reviews an alternative, numerically efficient PWA regression algorithm recently developed by the authors and detailed in [6], showing how the presented method can be applied for the identification of discrete-time multi-input multi-output PWARX and *Linear Parameter-Varying* (LPV) models. In this case, a PWA approximation of the scheduling variable dependent LPV model coefficients is estimated from the training data, along with a polyhedral partition of the scheduling variable domain. The main idea of the algorithm is to sequentially process the observed regressor/output pairs and assign each pair to the affine submodel that is most compatible to it, updating the corresponding parameters at the same time through recursive least-squares. The second stage starts once all the observations have been classified. At this stage, the regressor domain is partitioned into polyhedral regions through a piecewise linear separator, which can be efficiently computed either off-line through a regularized piecewise-smooth Newton method or recursively through an averaged stochastic gradient descent algorithm.

The paper is organized as follows. The general PWA regression problem is formulated in Section II. The specific problems of identification of PWARX models and LPV models are formulated in Section II-B and Section II-C, respectively. Section III summarizes the general PWA regression method, and its application to the identification of PWARX and LPV models is discussed in Section IV through two numerical examples.

A. Notation

Let $I \subset \{1, 2, \dots, \}$ be a finite set of integers and denote by $|I|$ the cardinality of I. Given a vector $a \in \mathbb{R}^n$, let a_i denote the i-th entry of a, a_I the subvector obtained by collecting the entries a_i for all $i \in I$, $\|a\|_2$ the Euclidean norm of a, a_+ a vector whose i-th element is $\max\{a_i, 0\}$. Given two vectors $a, b \in \mathbb{R}^n$, $\max(a, b)$ is the vector whose i-th component is $\max\{a_i, b_i\}$. Given a matrix $A \in \mathbb{R}^{n \times m}$, A' denotes the transpose of A, A_i the i-th row of A . Given two matrices A and B, $A \otimes B$ denotes the Kronecker product between A and B. Let I_n be the identity matrix of size n and $\mathbf{1}_n$ be the n-dimensional column vector of ones.

II. Problem statement

A. Piecewise affine regression

Consider the vector-valued PWA function $f : \mathcal{X} \to \mathbb{R}^{n_y}$

$$f(x) = \begin{cases} A_1 \begin{bmatrix} 1 \\ x \end{bmatrix} & \text{if } x \in \mathcal{X}_1, \\ \vdots & \\ A_s \begin{bmatrix} 1 \\ x \end{bmatrix} & \text{if } x \in \mathcal{X}_s, \end{cases} \tag{1}$$

where $x \in \mathbb{R}^{n_x}$, $\mathcal{X} \subseteq \mathbb{R}^{n_x}$, $s \in \mathbb{N}$ denotes the number of modes (i.e., the number of affine functions defining f), $A_i \in \mathbb{R}^{n_y \times (n_x+1)}$ are parameter matrices, and the sets $\mathcal{X}_i$, $i = 1, \dots, s$ are polyhedra defined by the linear inequalities

$$\mathcal{X}_i \doteq \{x \in \mathbb{R}^{n_x} : \ \mathcal{H}_i x \leq \mathcal{D}_i\}, \tag{2}$$

with $\mathcal{H}_i$ and $\mathcal{D}_i$ being real-valued matrices, that form a complete polyhedral partition of the space $\mathcal{X}$. The PWA regression problem can be summarized as follows.

*This work was partially supported by the European Commission under project H2020-SPIRE-636834 "DISIRE - Distributed In-Situ Sensors Integrated into Raw Material and Energy Feedstock" (http://spire2030.eu/disire/).

The authors are with the IMT School of Advanced Studies Lucca, 55100 Lucca, Italy. E-mails: {valentina.breschi, alberto.bemporad, dario.piga} @imtlucca.it

Problem 1: **PWA regression**
Consider the data-generating system:

$$y(k) = f_{\rm o}(x(k)) + e_{\rm o}(k), \tag{3}$$

where $e_{\rm o}(k) \in \mathbb{R}^{n_y}$ is a zero-mean additive random noise independent of $x(k)$ and $f_{\rm o} : \mathcal{X} \to \mathbb{R}^{n_y}$ is an unknown function, possibly discontinuous and not necessarily PWA. The PWA regression problem aims at finding a PWA approximation f of the unknown regression function $f_{\rm o}$ based on a set of N observations $\{x(k), y(k)\}_{k=1}^{N}$. ■

Estimating a PWA approximation f of the true function $f_{\rm o}$ requires (i) choosing the number of modes s, (ii) computing the parameter matrices $\{A_i\}_{i=1}^{s}$ that characterize the PWA submodels defining f, and (iii) finding the polyhedral partition $\{\mathcal{X}_i\}_{i=1}^{s}$ of the regressor space $\mathcal{X}$ where those PWA submodels are defined. In this work, we assume that the number of modes s is fixed by the user and can be chosen through cross-calibration based procedures.

The following subsections show how the problems of identifying hybrid dynamical models in PWARX form and Linear Parameter-Varying models can be cast as the PWA regression problem 1.

B. Identification of PWARX models

Model (1) represents a *Multi-Input Multi-Output* (MIMO) PWARX discrete-time dynamical system if the regressor vector $x(k)$ is a collection of past inputs and outputs, i.e.,

$$\begin{aligned} x(k) = [&y'(k-1) \;\; y'(k-2) \;\cdots\; y'(k-n_a) \\ &u'(k-1) \;\; u'(k-2) \;\cdots\; u'(k-n_b)]', \end{aligned} \tag{4}$$

where $u(k) \in \mathbb{R}^{n_u}$ and $y(k) \in \mathbb{R}^{n_y}$ denote the observed input and output signals at time k, respectively.

C. Identification of LPV models

Linear parameter-varying systems can be seen as an extension of *Linear Time-Invariant* (LTI) systems, as the dynamic relation between input and output signals is linear but it can change over time according to a measurable time-varying signal $p(t) \in \mathbb{R}^{n_p}$, the so-called *scheduling vector*.

The identification problem of MIMO LPV systems can be formulated as the PWA regression Problem 1 by adopting the following MIMO LPV-ARX form

$$y(k) = a_0(p(k)) + \sum_{j=1}^{n_a} a_j(p(k))y(k-j) + \sum_{j=1}^{n_b} a_{j+n_a}(p(k))u(k-j)$$

where $a_j(p(k))$, $j = 0, \ldots, n_a + n_b$, are PWA functions of $p(k)$ defined as:

$$a_j(p(k)) = \begin{cases} A_1^j(p(k)) & \text{if } p(k) \in \mathcal{P}_1, \\ \vdots & \\ A_s^j(p(k)) & \text{if } p(k) \in \mathcal{P}_s, \end{cases} \tag{5}$$

and $p(k) \in \mathcal{P} \subseteq \mathbb{R}^{n_p}$ is the value of the scheduling vector at time k. By imposing that each entry of matrix $A_i^j(p(k))$ depends affinely on the scheduling vector $p(k)$, the LPV identification problem reduces to reconstructing the PWA mapping of the p-dependent coefficient functions $\{a_j(p(k))\}_{j=0}^{n_a+n_b}$ over the polyhedral partition $\{\mathcal{P}_i\}_{i=1}^{s}$ of the scheduling vector set $\mathcal{P}$. In this case, the N-length sequence of observations $\{x(k), y(k)\}_{k=1}^{N}$ requires $x(k) = [y'(k-1) \; \ldots y'(k-n_a) \; u'(k-1) \; \ldots u'(k-n_b)]' \otimes [1 \; p'(k)]'$.

Note that, since the polyhedral partition $\{\mathcal{P}_i\}_{i=1}^{s}$ is not fixed a priori, the underlying dependencies of the functions $a_j(p(k))$ on the scheduling vector $p(k)$ are directly reconstructed from data. This represents one of the main advantages w.r.t. to widely used parametric LPV identification approaches (see, e.g., [3], [17]), which in turn require to parameterize $a_j(p(k))$ as a linear combination of some basis functions specified a priori (e.g., polynomial functions).

III. PWA REGRESSION ALGORITHM

In this section, we summarize the two-stage algorithm to solve the general PWA regression problem (namely, Problem 1) proposed in [6]. Its application to the identification of PWARX and LPV models is discussed in Section IV through two simulation examples.

The proposed PWA regression algorithm consists of two stages:

- S1. Simultaneous *clustering* of the regressor vectors $\{x(k)\}_{k=1}^{N}$ and *estimation* of the parameter matrices $\{A_i\}_{i=1}^{s}$ describing the PWA map in (1). This is performed recursively, by processing the training pairs $\{x(k), y(k)\}$ sequentially.
- S2. *Computation of a polyhedral partition* of the regressor space $\mathcal{X}$ through a multi-category linear separation method. This stage is performed after stage S1, and it can be executed either offline or online (recursively).

A. Iterative clustering and parameter estimation

Stage S1 is carried out as described in Algorithm 1. The algorithm is an extension to the case of multiple linear regressions and clustering of the approach proposed in [1] for solving recursive least squares problems using inverse QR decomposition.

Algorithm 1 requires an initial guess for the matrices A_i and the centroids c_i, $i = 1, \ldots, s$. Clearly, the final estimate of $\{A_i\}_{i=1}^{s}$ and of the clusters $\{\mathcal{C}_i\}_{i=1}^{s}$ depends on their initial values. Among the possible choices introduced in [6], in this paper the initial guess for the parameters is obtained executing Algorithm 1 without the first term in (6) and updating the cluster centroids and covariance matrices only once all the data have been classified. When working offline on a batch of data, estimation quality may be improved by repeating Algorithm 1 iteratively, by using its output as initial condition for its following execution.

After computing the estimation error $e_i(k)$ for all models i at Step 2.1, Step 2.2 picks up the "best" mode $i(k)$ to which the current sample $x(k)$ must be associated with (the reader is referred to [6] for details on the choice of the weight Λ_e and for a stochastic interpretation of the discrimination criterium (6) used to cluster the data).

Step 2.4 updates the parameter matrix $A_{i(k)}$ associated to the selected mode $i(k)$ using the recursive least-squares

Algorithm 1 Recursive clustering and parameter estimation

Input: Sequence of observations $\{x(k), y(k)\}_{k=1}^{N}$, desired number s of modes, noise covariance matrix Λ_e; initial condition for matrices A_i, cluster centroids c_i, and centroid covariance matrices R_i, $i = 1, \ldots, s$.

1. **let** $\mathcal{C}_i \leftarrow \emptyset$, $i = 1, \ldots, s$;
2. **for** $k = 1, \ldots, N$ **do**

2.1. **let** $e_i(k) \leftarrow y(k) - A_i \begin{bmatrix} 1 \\ x(k) \end{bmatrix}$, $i = 1, \ldots, s$;

2.2. **let**

$$i(k) \leftarrow \arg \min_{i=1,\ldots,s} (x(k) - c_i)' R_i^{-1} (x(k) - c_i) + e_i(k)' \Lambda_e^{-1} e_i(k); \quad (6)$$

2.3. **let** $\mathcal{C}_{i(k)} \leftarrow \mathcal{C}_{i(k)} \cup \{x(k)\}$;

2.4. **update** the parameter matrices $A_{i(k)}$ using the inverse QR factorization approach of [1];

2.5. **let** $\delta c_{i(k)} \leftarrow \dfrac{1}{|\mathcal{C}_{i(k)}|} (x(k) - c_{i(k)})$;

2.6. **update** the centroid $c_{i(k)}$ of the cluster $\mathcal{C}_{i(k)}$:

$$c_{i(k)} \leftarrow c_{i(k)} + \delta c_{i(k)}; \quad (7)$$

2.7. **update** the centroid covariance matrix $R_{i(k)}$ for cluster $\mathcal{C}_{i(k)}$

$$R_{i(k)} \leftarrow \frac{|\mathcal{C}_{i(k)}| - 2}{|\mathcal{C}_{i(k)}| - 1} R_{i(k)} + \delta c_{i(k)} \delta c_{i(k)}' + \frac{1}{|\mathcal{C}_{i(k)}| - 1} \left[x(k) - c_{i(k)}\right] \left[x(k) - c_{i(k)}\right]'; \quad (8)$$

3. **end for**;
4. **end**.

Output: Estimated matrices $\{A_i\}_{i=1}^{s}$, clusters $\{\mathcal{C}_i\}_{i=1}^{s}$.

algorithm of [1] based on inverse QR factorization. Note that only the parameters of the matrix $A_{i(k)}$ associated to the selected mode $i(k)$ are updated at time k, while the parameters associated to the other modes are not.

B. Partitioning the regressor space

When one is interested in getting also the partition $\{\mathcal{X}_i\}_{i=1}^{s}$, besides the affine models $\{A_i\}_{i=1}^{s}$, the clusters $\{\mathcal{C}_i\}_{i=1}^{s}$ must be separated via linear multicategory discrimination.

The linear multicategory discrimination problem amounts to computing a convex piecewise affine separator function $\phi : \mathbb{R}^{n_x} \to \mathbb{R}$ discriminating between the clusters $\mathcal{C}_1, \ldots, \mathcal{C}_s$. The piecewise affine separator ϕ is defined as the maximum of s affine functions $\{\phi_i(x)\}_{i=1}^{s}$, i.e.,

$$\phi(x) = \max_{i=1,\ldots,s} \phi_i(x). \quad (9)$$

The affine functions $\phi_i(x)$ are described by the parameters $\omega^i \in \mathbb{R}^{n_x}$ and $\gamma^i \in \mathbb{R}$, namely: $\phi_i(x) = \begin{bmatrix} x' & -1 \end{bmatrix} \begin{bmatrix} \omega^i \\ \gamma^i \end{bmatrix}$.

For $i = 1, \ldots, s$, let M_i be a $m_i \times n_x$ dimensional matrix (with m_i denoting the cardinality of cluster $\mathcal{C}_i$) obtained by stacking the regressors $x(k)'$ belonging to $\mathcal{C}_i$ in its rows. According to [8], in case of linearly separable clusters, the piecewise-affine separator ϕ satisfies the conditions:

$$\begin{bmatrix} M_i & -\mathbf{1}_{m_i} \end{bmatrix} \begin{bmatrix} \omega^i \\ \gamma^i \end{bmatrix} \geq \begin{bmatrix} M_i & -\mathbf{1}_{m_i} \end{bmatrix} \begin{bmatrix} \omega^j \\ \gamma^j \end{bmatrix} + \mathbf{1}_{m_i}, \quad i, j = 1, \ldots, s, \ i \neq j. \quad (10)$$

1) Off-line multicategory discrimination: Rather than solving a *robust linear programming* (RLP) problem as in [8], the parameters $\{\omega^i, \gamma^i\}_{i=1}^{s}$ are computed by solving the convex unconstrained optimization problem

$$\min_{\xi} \frac{\lambda}{2} \sum_{i=1}^{s} \left(\|\omega^i\|_2^2 + (\gamma^i)^2 \right) + \sum_{i=1}^{s} \sum_{\substack{j=1 \\ j \neq i}}^{s} \frac{1}{m_i} \left\| \left(\begin{bmatrix} M_i & -\mathbf{1}_{m_i} \end{bmatrix} \begin{bmatrix} \omega^j - \omega^i \\ \gamma^j - \gamma^i \end{bmatrix} + \mathbf{1}_{m_i} \right)_+ \right\|_2^2, \quad (11)$$

with $\xi = \begin{bmatrix} (\omega^1)' & \ldots & (\omega^s)' & \gamma^1 & \ldots & \gamma^s \end{bmatrix}'$. Problem (11) aims at minimizing the averaged squared 2-norm of the violation of the inequalities in (10). The regularization parameter $\lambda > 0$ is introduced so that the objective function in (11) is strongly convex. Problem (11) can be efficiently solved via the *Regularized Piecewise-Smooth Newton* (RPSN) method explained in [6] and originally proposed in [5].

2) Recursive multicategory discrimination: As an alternative to the off-line approach, or in addition to it for refining the partition ϕ on-line based on streaming data, a recursive approach to solve problem (11) based on on-line convex programming can be used.

Let us treat the data-points $x \in \mathbb{R}^{n_x}$ as random vectors and assume that an oracle function $i : \mathbb{R}^{n_x} :\to \{1, \ldots, s\}$ exists that, to any $x \in \mathbb{R}^{n_x}$, assigns the corresponding mode $i(x) \in \{1, \ldots, s\}$. Function i implicitly defines clusters in the data-point space $\mathbb{R}^{n_x}$. Let us also assume that the following values $\pi_i = \text{Prob}[i(x) = i]$ are known for all $i = 1, \ldots, s$. The multicategory discrimination problem (11) can be the generalized to the following convex regularized stochastic optimization problem

$$\xi^* = \min_{\xi} E_{x \in \mathbb{R}^{n_x}} \left[\ell(x, \xi) \right] + \frac{\lambda}{2} \|\xi\|_2^2 \quad (12)$$

$$\ell(x, \xi) = \sum_{\substack{j=1 \\ j \neq i(x)}}^{s} \frac{1}{\pi_{i(x)}} \left(x'(\omega^j - \omega^{i(x)}) - \gamma^j + \gamma^{i(x)} + 1 \right)_+^2$$

where $E_x[\cdot]$ is the expected value w.r.t. x. Problem (12) aims at violating the least, on average over x, the condition in (10) for $i = i(x)$. For details the reader is referred to [6]. The solution of (12), which provides the piecewise affine multicategory discrimination function ϕ, can be computed by online convex optimization, through the *Averaged Stochastic Gradient Descent* (ASGD) method in [6].

IV. Simulation examples

In this section, we consider two numerical examples to show the application of the reviewed PWA regression approach to the identification of PWARX and LPV systems. All computations are carried out on an i7 2.40-GHz Intel core processor with 4 GB of RAM running MATLAB R2014b.

In both examples, the output used in the training phase is corrupted by an additive zero-mean white noise $e_{\rm o}$ with Gaussian distribution. The effect of measurement noise on the output signal is quantified through the Signal-to-Noise Ratio (SNR), that is defined for the i-th output channel as

$$\mathrm{SNR}_i = 10 \log \frac{\sum_{k=1}^{N} \left(y_i(k) - e_{\mathrm{o},i}(k)\right)^2}{\sum_{k=1}^{N} e_{\mathrm{o},i}^2(k)}, \qquad (13)$$

with $e_{\mathrm{o},i}(k)$ denoting the i-th component of $e_{\rm o}(k)$.

The results obtained after the training phase are validated on a noiseless data sequence. Let $y_{\rm o}$ and $\hat{y}$ denote, respectively, the vectors staking the actual and the simulated outputs of the estimated model, let $\bar{y}_{\mathrm{o},i}$ be the sample mean of the i-th output, and $N_{\rm V}$ the length of the validation data sequence. The *Best Fit Rate* (BFR) indicator

$$\mathrm{BFR}_i = \max\left\{1 - \frac{||\hat{y}_i - y_{\mathrm{o},i}||_2}{||y_{\mathrm{o},i} - \bar{y}_{\mathrm{o},i}||_2}, 0\right\} \cdot 100\% \qquad (14)$$

defined for each output channel i, $i = 1, \ldots, n_y$, is used to assess the quality of the estimated models.

A. Identification of a multivariable PWARX system

Let the system generating the data be a MIMO PWARX system described by the difference equation

$$\begin{bmatrix} y_1(k) \\ y_2(k) \end{bmatrix} = \begin{bmatrix} -0.83 & 0.20 \\ 0.30 & -0.52 \end{bmatrix} \begin{bmatrix} y_1(k-1) \\ y_2(k-1) \end{bmatrix} + \begin{bmatrix} -0.34 & 0.45 \\ -0.30 & 0.24 \end{bmatrix} \begin{bmatrix} u_1(k-1) \\ u_2(k-1) \end{bmatrix} + \begin{bmatrix} 0.20 \\ 0.15 \end{bmatrix} + \max\left\{ \begin{bmatrix} 0.20 & -0.90 \\ 0.10 & -0.42 \end{bmatrix} \begin{bmatrix} y_1(k-1) \\ y_2(k-1) \end{bmatrix} + \begin{bmatrix} 0.42 & 0.20 \\ 0.50 & 0.64 \end{bmatrix} \begin{bmatrix} u_1(k-1) \\ u_2(k-1) \end{bmatrix} + \begin{bmatrix} 0.40 \\ 0.30 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \end{bmatrix} \right\} + e_{\rm o}(k),$$

which is characterized by $\bar{s} = 4$ operating modes, given by the possible combinations of sign of the components of the first vector argument of the "max" operator. The excitation input $u(k)$ is a white noise sequence with uniform distribution in the box $[-1 \;\; 0] \times [-0.4 \;\; 0.6]$ and length $N = 4000$, $e_{\rm o}(k) \in \mathbb{R}^2$ is a zero-mean white noise with covariance matrix $\Lambda_e = \left[\begin{smallmatrix} 0.02 & 0.02 \\ 0.02 & 0.02 \end{smallmatrix}\right]$. This corresponds to signal-to-noise ratios equal to $\mathrm{SNR}_1 = 8.7$ dB and $\mathrm{SNR}_2 = 6.9$ dB on the first and second output channels, respectively.

We run Algorithm 1 with $s = \bar{s} = 4$ for 15 times. The clusters generated by Algorithm 1 are then separated through the off-line multicategory discrimination method described in Section III-B.1, by solving (11) via the RPSN method described in [6], which is initialized with $\{\omega^i, \gamma^i\}_{i=1}^{s} = 0$. The regularization parameter λ is set to 10^{-5}.

The quality of the estimated PWA model is assessed w.r.t. a validation set of 500 samples. The true output $y_{\rm o}$ and the open-loop simulated output $\hat{y}$ generated by the estimated model are plotted in Fig. 1, along with the simulation error $y_{\rm o}(k) - \hat{y}(k)$. For the sake of visualization, only the samples from time 101 to 200 related to the first channel are shown in Fig. 1. The obtained BFR is 96.1% and 96.3% for the first and the second output channel, respectively. The estimated polyhedral partition of the regressor space is such that only 12 out of 500 data samples are misclassified.

As the accuracy of the final model estimate and the total CPU time is influenced by the number M of runs of Algorithm 1, the performance of the proposed learning approach has been tested with respect to both M and the dimension N of the training set. The obtained BFR as a function of iterations of Algorithm 1 is plotted in Fig. 2 for different values of N. Clearly, there is no improvement in BFR on the two output channels after about 8 runs.

TABLE I

PWARX : BFR VS LENGTH N OF THE TRAINING SET.

		$N = 4000$	$N = 20000$	$N = 100000$
BFR_1	(Off-line) RLP [8]	96.0 %	96.5 %	99.0 %
	(Off-line) RPSN	96.2 %	96.4 %	98.9 %
	(On-line) ASGD	86.7 %	95.0 %	96.7 %
BFR_2	(Off-line) RLP [8]	96.2 %	96.9 %	99.0 %
	(Off-line) RPSN	96.3 %	96.8 %	99.0 %
	(On-line) ASGD	87.4 %	95.2 %	96.4 %

TABLE II

PWARX: CPU TIME VS LENGTH N OF THE TRAINING SET.

	$N = 4000$	$N = 20000$	$N = 100000$
(Off-line) RLP [8]	0.308 s	3.227 s	112.435 s
(Off-line) RPSN	0.016 s	0.086 s	0.365 s
(On-line) ASGD	0.013 s	0.023 s	0.067 s

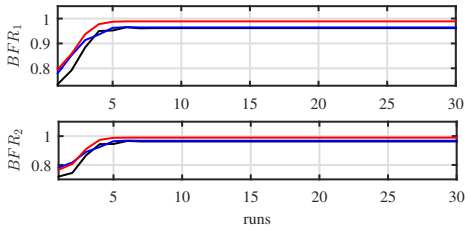


Fig. 2. PWARX: BFR on the first and on the second output channel vs number of runs of Algorithm 1 (N : 4000 (black), 20000 (blue) and 100000 (red)).

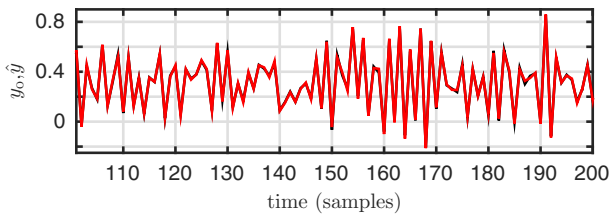
The total CPU time for estimating the PWARX model is 0.76 s, of which 0.016 s are taken to solve the linear multi-category discrimination problem (11). For the sake of comparison, both the RLP approach of [8] and the on-line method described in Section III-B.2[1] are also run to generate the partition. Results in Table I show that all of the three algorithms used to compute the partition of the regressor space lead to an accurate estimate of the system, with BFRs larger then 95 % (except when $N = 4000$ and the on-line multi-category discrimination method is used). Furthermore, the results show that the estimated models become more accurate as the number of training samples increases. The CPU times taken to compute the polyhedral partition are reported in Table II, which shows that, for a large training set (i.e., $N = 100000$), the RPSN and the ASGD method are about 300x and 1600x faster, respectively, than the robust linear programming method of [8].

B. Identification of an LPV system

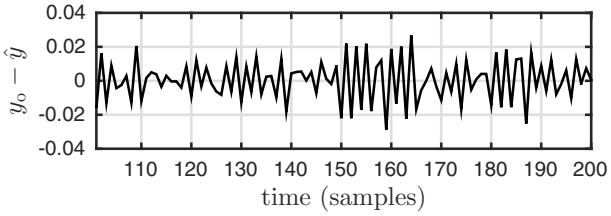
Let the data be collected from the MIMO LPV system

$$\begin{bmatrix} y_1(k) \\ y_2(k) \end{bmatrix} = \begin{bmatrix} \bar{a}_{1,1}(p(k)) & \bar{a}_{1,2}(p(k)) \\ \bar{a}_{2,1}(p(k)) & \bar{a}_{2,2}(p(k)) \end{bmatrix} \begin{bmatrix} y_1(k-1) \\ y_2(k-1) \end{bmatrix} + \begin{bmatrix} \bar{b}_{1,1}(p(k)) & \bar{b}_{1,2}(p(k)) \\ \bar{b}_{2,1}(p(k)) & \bar{b}_{2,2}(p(k)) \end{bmatrix} \begin{bmatrix} u_1(k-1) \\ u_2(k-1) \end{bmatrix} + e_{\rm o}(k),$$

[1] In executing the on-line approach the weights π_i and the initial guess of ϕ used by the stochastic gradient method are computed by executing the batch Algorithm in Section III-B.1 on the first 3000 training samples.



(a) First output channel (output signal): black = true, red = estimated



(b) First output channel (simulation error)

Fig. 1. PWARX: output signal and simulation error on the first output channel.

where

$$\bar{a}_{1,1}(p(k)) = \begin{cases} -0.3 & \text{if} \quad 0.4\,(p_1(k)+p_2(k)) \le -0.3, \\ 0.3 & \text{if} \quad 0.4\,(p_1(k)+p_2(k)) \ge 0.3, \\ 0.4\,(p_1(k)+p_2(k)) & \text{otherwise}, \end{cases}$$

$$\bar{a}_{1,2}(p(k)) = 0.5\,(|p_1(k)|+|p_2(k)|)\,, \quad \bar{a}_{2,1}(p(k)) = p_1(k)-p_2(k),$$

$$\bar{a}_{2,2}(p(k)) = \begin{cases} 0.5 & \text{if} \quad p_1(k) < 0, \\ 0 & \text{if} \quad p_1(k) = 0, \\ -0.5 & \text{if} \quad p_1(k) > 0, \end{cases}$$

$$\bar{b}_{1,1}(p(k)) = 3p_1(k) + p_2(k),$$

$$\bar{b}_{1,2}(p(k)) = \begin{cases} 0.5 & \text{if} \quad 2\left(p_1^2(k)+p_2^2(k)\right) \ge 0.5, \\ 2\left(p_1^2(k)+p_2^2(k)\right) & \text{otherwise}, \end{cases}$$

$$\bar{b}_{2,1}(p(k)) = 2\sin\{p_1(k)-p_2(k)\}\,, \quad \bar{b}_{2,2}(p(k)) = 0.$$

Both the input $u(k)$ and the scheduling vector $p(k)$ are white noise sequences (independent of each other) of length $N = 11000$ with uniform distribution in the boxes $[-0.5 \;\; 0.5] \times [-0.5 \;\; 0.5]$ and $[-1 \;\; 1] \times [-1 \;\; 1]$, respectively. The noise covariance matrix of $e_o(k) \in \mathbb{R}^2$ is $\Lambda_e = \left[\begin{smallmatrix} 0.25 & 0 \\ 0 & 0.25 \end{smallmatrix}\right]$. This corresponds to signal-to-noise ratios on the first and on the second output channel equal to $\mathrm{SNR}_1 = 4$ dB and $\mathrm{SNR}_2 = 7$ dB, respectively. The goal is to estimate, from the gathered data, a PWA approximation of the p-dependent nonlinear functions $\bar{a}_{i,j}$ and $\bar{b}_{i,j}$ defining the behaviour of the LPV data-generating system.

1) Choice of the number of modes: The number s of polyhedral regions defining the partition of the scheduling vector space $\mathcal{P} = [-1 \;\; 1] \times [-1 \;\; 1]$ is chosen through cross validation. Specifically, the 11000-length training data set is split into two disjoint sets. The first 10000 samples are used to estimate a PWA approximation of $\bar{a}_{i,j}$ and $\bar{b}_{i,j}$, along with the polyhedral partition of the scheduling vector space $\mathcal{P}$, for different values of s in the range 5–30. For each value of s, the identification Algorithm 1 is run 10 times. The second part of training data (i.e., the remaining 1000 samples) is used to assess the quality of the identified LPV models. Among the identified LPV models, the one providing the largest aggregated $\mathrm{BFR_T} = \mathrm{BFR}_1 + \mathrm{BFR}_2$ is selected, which corresponds to $s = 10$ polyhedral regions. The computed polyhedral partition, obtained by solving problem (11) through the RPSN method explained in [6], is plotted in Fig. 3 (the Hybrid Toolbox for MATLAB [4] has been used to plot the polytopes in Fig. 3).

2) Model quality assessment: The quality of the estimated LPV model is then assessed w.r.t. a validation dataset, consisting of a new sequence of 2000 noiseless samples used neither to estimate the LPV model nor to select the number of modes s. For the sake of comparison, the nonlinear

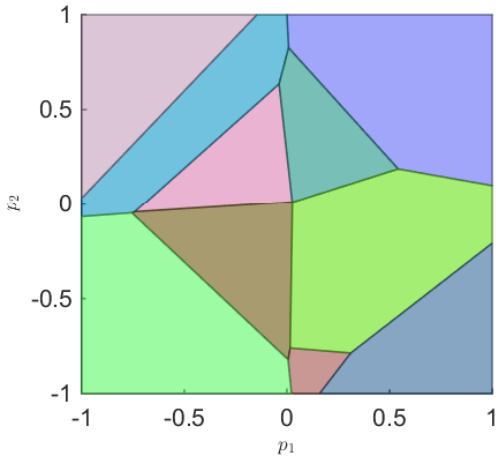


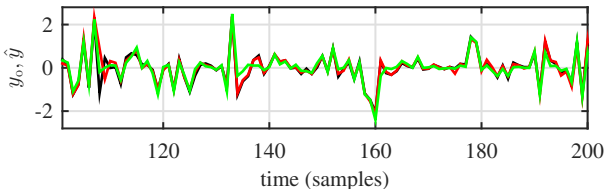
Fig. 3. LPV: polyhedral partition of the scheduling vector space $\mathcal{P}$.

coefficient functions $\bar{a}_{i,j}(p(k))$ and $\bar{b}_{i,j}(p(k))$ are also estimated through the parametric LPV identification approach proposed in [3], by parameterizing the nonlinear functions $\bar{a}_{i,j}(p(k))$ and $\bar{b}_{i,j}(p(k))$ as fourth-order polynomials in the two-dimensional scheduling vector $p(k)$.

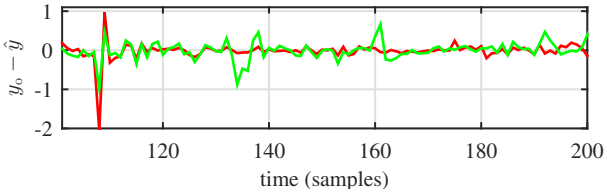
The true outputs y_o and the simulated output sequences $\hat{y}$ of the estimated LPV models are plotted in Fig. 4, along with the simulation error $y_o(k) - \hat{y}(k)$. For the sake of visualization, only the samples from time 101 to 200 related to the second channel are reported. The BFR on the two output channels is reported in Table III. The obtained results show that the proposed LPV identification approach based on the PWA approximation of the coefficient functions $\bar{a}_{i,j}(p(k))$ and $\bar{b}_{i,j}(p(k))$) outperforms the parametric LPV identification approach in [3].

We also remark that the "online" computational time required to evaluate the output of the LPV model, given the current value of the scheduling vector $\bar{p}$ and the past input/output observations is about 120 μs, 40 μs of which are required to evaluate which region the current scheduling vector belongs to. This relatively "low" online computational time is mainly due to the PWA structure of the coefficient functions describing the LPV model, and it allows to use the estimated LPV model in applications requiring a "fast" online determination of the operating mode, such as in gain scheduling or in LPV model predictive control.

3) Performance of multi-category discrimination algorithms: The CPU time required to estimate the LPV model through the proposed PWA regression approach is 759 s. This includes the cross-validation phase to compute the number of



(a) Second output channel (output signal): black = true, red = PWA regression, green =polynomial parametrization [3]



(b) Second output channel (simulation error): red = PWA regression, green =polynomial parametrization [3]

Fig. 4. LPV: output signal and simulation error on the second output channel.

TABLE III

LPV: BFR OBTAINED WITH PWA REGRESSION AND POLYNOMIAL PARAMETRIZATION [3]

	BFR_1	BFR_2
PWA regression	87 %	84 %
parametric LPV [3]	80 %	70 %

modes s. For $s = 10$, the CPU time required to compute the LPV model is 14 s, 0.4 s of which are spent to compute the polyhedral partition via problem (11) (RLP discrimination algorithm of [8] takes 4.2 seconds, i.e., almost 10x slower).

For a more exhaustive comparison between the RPSN approach and the RLP algorithm of [8], the CPU time required by the two algorithms to partition the scheduling parameter space is plotted, as a function of s, in Fig. 5. Fig. 5 also shows the CPU time required by the ASGD algorithm in [6] to compute the solution of problem (12). The weights π_i and the initial estimate used by the averaged stochastic gradient descent algorithm are computed by solving problem (11) on the first 1000 training samples. The remaining 9000 training samples are processed recursively. The regularization parameter λ in problems (11) and (12) is set to 10^{-5}. Results in Fig. 5 show that: (i) the CPU time required by all of the three discrimination algorithms to partition the scheduling vector space increases with the number of modes s (Fig. 5), as the number of parameters ξ defining the piecewise affine separator $\phi(x)$ in (9) increases linearly with s; (ii) the (offline) RPSN method and the (online) ASGD method used to solve problem (11) and (12), respectively, are faster (from 6x to 20x) than the robust linear programming based approach of [8].

V. CONCLUSIONS

In this paper we have reviewed the PWA regression algorithm introduced in [6], and discussed its application to the identification of PWARX and LPV systems. Through the examples, it has been shown that the presented approach is computationally effective for off-line and on-line learning of PWARX and LPV models. Future research includes the extension of the PWA regression algorithm presented in the paper to the identification of hybrid and LPV systems under different noise conditions and the generalization to piecewise-nonlinear models (such as piecewise polynomial).

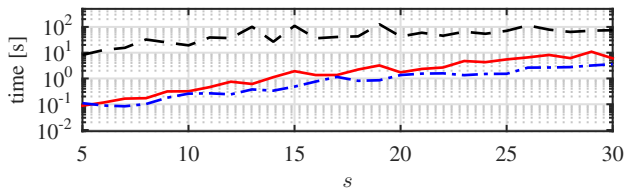


Fig. 5. LPV: CPU time vs number of modes (s). (black dashed: RLP [8]; red: RPSN; blue dash-dot: ASGN).

REFERENCES

[1] S.T. Alexander and A.L. Ghirnikar. A method for recursive least squares filtering based upon an inverse QR decomposition. *IEEE Trans. Signal Processing*, 41(1):20–30, 1993.

[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] B. A. Bamieh and L. Giarré. Identification of linear parameter-varying models. *International Journal of Robust Nonlinear Control*, 12(9):841–853, 2002.

[4] A. Bemporad. Hybrid Toolbox - User's Guide, 2004. url: http://cse.lab.imtlucca.it/ bemporad/hybrid/toolbox.

[5] A. Bemporad, D. Bernardini, and P. Patrinos. A convex feasibility approach to anytime model predictive control. Technical report, 2015. `http://arxiv.org/abs/1502.07974`.

[6] A. Bemporad, V. Breschi, and D. Piga. Piecewise affine regression via recursive multiple least squares and multicategory discrimination. Technical report, 2016. Submitted to Automatica. Available at: `http://www.dariopiga.com/TR/TR_PWAReg_BBP_2015.pdf`.

[7] 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, 2005.

[8] K.P. Bennett and O.L. Mangasarian. Multicategory discrimination via linear programming. *Opt. Methods and Software*, 3:27–39, 1994.

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

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

[12] H. Ohlsson and L. Ljung. Identification of switched linear regression models using sum-of-norms regularization. *Automatica*, 49(4):1045–1050, 2013.

[13] N. Ozay, C. Lagoa, and M. Sznaier. Set membership identification of switched linear systems with known number of subsystems. *Automatica*, 51:180–191, 2015.

[14] S. Paoletti, A. L. Juloski, G. Ferrari-Trecate, and R. Vidal. Identification of hybrid systems a tutorial. *European journal of control*, 13(2):242–260, 2007.

[15] M. Petreczky and L. Bako. On the notion of persistence of excitation for linear switched systems. In *50th Conference on Decision and Control*, pages 1840–1847, Orlando, FL, 2011.

[16] M. Petreczky, Laurent L.B̄ako, and S. A. Lecoeuche. Minimality and identifiability of SARX systems. In *16th IFAC Symposium on System Identification*, pages 541–546, Brussels, Belgium, 2012.

[17] D. Piga, P. Cox, R. Tóth, and V. Laurain. LPV system identification under noise corrupted scheduling and output signal observations. *Automatica*, 53:329–338, 2015.

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