2016 IEEE 55th Conference on Decision and Control (CDC)
ARIA Resort & Casino
December 12-14, 2016, Las Vegas, USA

Learning hybrid models with logical and continuous dynamics via multiclass linear separation

Valentina Breschi, Dario Piga and Alberto Bemporad

Abstract— **Hybrid dynamical models are a powerful tool for describing the behaviour of many industrial processes and physical phenomena in which logical (discrete) and analog (continuous) dynamics exist and interact. Black-box identification of hybrid models from input/output observations and no information on the operating mode of the system is a challenging problem, as both the logical and the continuous dynamics must be retrieved.**

In this work, we consider the identification of *discrete hybrid automata* (DHA), which represent a mathematical abstraction of hybrid models whose logical dynamics are described by a *finite state machine* (FSM) and the continuous dynamics are represented through affine discrete-time dynamical models. We propose a two-stage estimation algorithm based on the joint use of clustering, multi-model recursive least-squares and linear multicategory discrimination techniques, which allows us to estimate both the affine models describing the continuous dynamics and the FSM governing the logical dynamics of the system.

I. Introduction

Hybrid models represent a powerful mathematical abstraction to describe the behaviour of the real systems that are characterized by the interaction between logical and continuous dynamics. Four-stroke engines, switching electronic circuits and mechanical systems with collisions are only some of the processes that admit a natural hybrid representation.
More generally, systems working under changing operating conditions can be satisfactorily modelled using the hybrid formalism. An extensive review on theory and applications of hybrid models can be found in the special issue [4].

The problem of identifying hybrid systems from data has been studied by both computer scientists and control theorists, but a well established unified framework for the identification of hybrid systems is still missing. An exhaustive review of the main results in this field can be found in [4]. However, most of the papers reviewed therein address the identification of the number of possible operating regimes and the conditions leading to transitions of the logical states, while the identification of the continuous dynamics is not considered. In [10] the combined use of a prefix tree and linear regression techniques is employed to estimate both the discrete and continuous dynamics of timed automata. However, as timed automata are considered, the approach accounts only for transitions triggered by time.
Several contributions on the identification of dynamical *piecewise affine* (PWA) systems with real-valued states are also available in the literature, among which we mention the clustering based procedures [3], [6], [9]. However, the identification of PWA systems do not involve the modelling of the logical dynamics leading to a change in the operating condition of the process.

In this paper, the identification of *discrete hybrid automata* (DHA) from only a set of input/output data generated by the system is addressed. DHA models, introduced in [11], result from the connection of a FSM (which governs the logic component of the system) with a switched affine discrete-time dynamical model (which describe the continuous states of the system). As a model class, DHA are quite general and include, among others, MLD models, PWA dynamical models and *max-min-plus-scaling* systems [8].

Under the hypothesis that the transitions between different operating condition are due to the crossing of thresholds by the signal $\varphi \in \Phi$, which is either equal to the output $y \in \mathcal{Y}$ or the regressor $x \in \mathcal{X}$, we propose a two-stage algorithm for the identification of the addressed DHA. In the first stage, the input/output observations are sequentially processed, simultaneously assigning each of them to the most compatible mode and updating the parameters of the corresponding dynamical sub-model. To accomplish this task, an approach similar to the one presented in [3] is used and extended to handle DHA with modes sharing the same continuous dynamics. This first stage leads to the generation of a set of clusters, each associated to one of the discrete states of the system. Based on the estimated clusters, the conditions governing the evolution of the discrete dynamics are then retrieved, by computing a collection of polyhedral partitions of space Φ through the multi-class linear separation (offline) method presented by the authors in [3].
The proposed algorithm extends the approach presented in [2], [3], as it enables to retrieve a model for the discrete dynamics of the hybrid system.

The paper is organized as follows. Discrete hybrid automata are formally introduced in Section II and the addressed identification problem is formulated in Section III. The proposed two-stage identification approach is then described in Section IV. Two case studies, one of which involving experimental data, are presented in Section V to show the effectiveness of the developed identification method.

A. Notation

Let $\mathbb{R}^n$ be the set of real vectors of dimension n. Let $I \subset \{1, 2, \ldots, \}$ be a finite set of integers and denote by

The authors are with the IMT School for Advanced Studies Lucca, Piazza San Francesco 19, 55100 Lucca, Italy. E-mail: valentina.breschi@imtlucca.it; dario.piga@imtlucca.it; alberto.bemporad@imtlucca.it.

$|I|$ the cardinality of I. Denote with $\{0,1\}^n$ the set of n-dimensional vectors whose elements take Boolean values. Given a vector $a \in \mathbb{R}^n$, let a^i denote its i-th component and $\|a\|_2$ be the Euclidean norm of a. Given a matrix $A \in \mathbb{R}^{n \times m}$, A' denotes the transpose of A, $tr(A)$ its trace. I_n indicates the identity matrix of size n, $\mathbf{1}_n$ and $\mathbf{0}_n$ are the n-dimensional column vector of ones and zeros, respectively.

II. Description of discrete hybrid automata

In order to frame the considered identification problem, each element of a DHA is formally defined in this section, according to the definitions introduced in [11].

A. Switched Affine Systems

A Switched Affine System (SAS), $\mathcal{S}$, is a collection of affine dynamical systems. Let $k \in \mathbb{N}$ be the discrete time index, $u(k) \in \mathbb{R}^{n_u}$ be the exogenous continuous input and $y(k) \in \mathcal{Y} \subseteq \mathbb{R}^{n_y}$ be the continuous output at time k. For fixed model orders n_a and n_b, $\mathcal{S}$ can be represented in the autoregressive form:

$$y(k) = \theta'_{i(k)} \tilde{x}(k), \tag{1}$$

where $\theta_{i(k)}$, with $i(k) = 1, \dots, s$, are the parameter matrices describing the affine dynamical submodels and $\tilde{x}(k) \in \mathbb{R}^{n_{\tilde{x}}}$ is the extended regressor vector defined as $\tilde{x}(k) = \left[\, x(k) \;\, 1 \,\right]'$, with $x(k) \in \mathcal{X} \subseteq \mathbb{R}^{n_x}$ being the collection of past input and output measurements, i.e.,

$$x(k) = [y'(k-1) \dots y'(k-n_a) \; u'(k-1) \dots u'(k-n_b)]' .$$

B. Event Generator

An Event Generator (EG) $\mathcal{E}$ generates a Boolean vector $\delta_e(k) \in \mathcal{D} \subseteq \{0,1\}^{n_e}$ according to the satisfaction of affine constraints, i.e.,

$$\delta_e(k) = f_H(\varphi(k)), \tag{2}$$

where $f_H : \Phi \to \mathcal{D} \subseteq \{0,1\}^{n_e}$ is a vector of descriptive functions of a linear hyperplane. Specifically we suppose that, for a given matrix $A \in \mathbb{R}^{n_e \times n_\varphi}$ and vector $B \in \mathbb{R}^{n_e}$, $[\delta_e^i(k) = 1] \leftrightarrow [A_i \varphi(k) \leq B^i]$, where $\varphi(k) \in \Phi \subseteq \mathbb{R}^{n_\varphi}$ is either equal to the output $y(k)$ or the regressor $x(k)$.

C. Finite State Machine

A Finite State Machine (FSM) $\mathcal{F}$ is a discrete dynamic process that evolves according to a logic state update function

$$x_b(k+1) = f_B(x_b(k), u_b(k), \delta_e(k)),$$

where $x_b \in \mathcal{X}_b \subseteq \{0,1\}^{n_{x_b}}$ is the discrete (or logic) state, $u_b \in \mathcal{U}_b \subseteq \{0,1\}^{n_{u_b}}$ is an exogenous Boolean input, $\delta_e(k)$ is the endogenous input coming from the event generator $\mathcal{E}$, and $f_B : \mathcal{X}_b \times \mathcal{U}_b \times \mathcal{D} \to \mathcal{X}_b$ is a deterministic logic function[1].

We suppose that no exogenous Boolean input $u_b(k)$ is present, i.e.,

$$x_b(k+1) = f_B(x_b(k), \delta_e(k)). \tag{3}$$

[1]Synchronous FSM will be considered, so that the transitions may happen only at the sampling instants.

D. Mode Selector

The Mode Selector (MS) $\mathcal{M}$ is a Boolean function $f_M : \mathcal{X}_b \times \mathcal{D} \to \{1, \dots, s\}$:

$$i(k) = f_M(x_b(k), \delta_e(k)) \tag{4}$$

where $i(k)$ is the selected dynamic submodel of the SAS, called *active mode*.
In this work, we assume that the mode selector $\mathcal{M}$ is only a function of the current logic state $x_b(k)$. This means that one and only one mode $i(k)$ is associated to the logic state $x_b(k)$, and the active mode $i(k)$ is driven by the logic state update function f_B in (3). Because of this assumption, with some abuse of notation, we will use interchangeably the terms *logic/discrete state* and *active mode* in the rest of the paper.

III. Problem statement

Based on the definitions introduced in Section II, the identification problem addressed in the paper is formally stated.

Problem 1: **Discrete Hybrid Automaton identification** Consider an unknown data-generating DHA:

$$\mathcal{H} = \{\mathcal{S}, \mathcal{E}, \mathcal{F}, \mathcal{M}\}, \tag{5}$$

where $\mathcal{S}$ is the correspondent SAS, $\mathcal{E}$ is the Event Generator, $\mathcal{F}$ is the associated FSM and $\mathcal{M}$ is the Mode Selector.
Given a set of N input/output pairs $\{u(k), y(k)\}_{k=1}^N$ generated by the "true" underlying system (5), the identification problem addressed in this work aims at estimating a DHA model describing the input/output relationship between the data, under the hypothesis that the measured outputs $y(k)$ may be noisy and the sequence of active modes $i(k)$ is not directly observable. ■

The DHA identification problem requires one to:

- **(i)** choose the number of local affine submodels s (or, equivalently, the number of different values of the logic state x_b defining the FSM $\mathcal{F}$). In this work, we assume that s is fixed by the user. It can be chosen through cross-validation, with a possible upper-bound dictated by the maximum tolerable complexity of the estimated DHA model.
- **(ii)** estimate the parameter matrices $\{\theta_i\}_{i=1}^s$ associated to each affine dynamical system of $\mathcal{S}$ (see eq. (1));
- **(iii)** identify, from the input/output observations $\{u(k), y(k)\}_{k=1}^N$, the transitions between the active modes (or equivalently, between the discrete states x_b) and derive the conditions leading to such transitions.

IV. DHA identification algorithm

A two-step algorithm is used to tackle the DHA identification problem. Specifically, the proposed method consists of the following stages:

S1. The simultaneous *clustering* of the data $\{x(k), y(k)\}_{k=1}^N$ and *estimation* of the parameter matrices $\theta_1, \dots, \theta_s$. This is performed recursively, by processing the training samples $\{x(k), y(k)\}_{k=1}^N$ sequentially. After this stage, each sample pair

$\{x(k), y(k)\}$ is associated to a cluster $\mathcal{C}_i$ ($i = 1, \dots, s$) that represents the set of data points associated to the i-th logic state of the system. Therefore, both the parameter matrices θ_i and the sequence of *active modes* $\mathcal{I} = \{i(k)\}_{k=1}^{N}$ are estimated.

S2. The definition of s different partitions $\{\Phi_j\}_{j=1}^{s}$ of the space Φ where the signal φ is defined. Each partition Φ_j implicitly represent the function $f_H(\varphi(k))$ in eq. (2) describing the event generator $\mathcal{E}$ and the logic state update rule $x_b(k+1) = f_B(x_b(k), \delta_e(k))$, given that the active mode $i(k)$ at the current time k is $i(k) = j$. More specifically, after estimating the sequence of *active modes* $\mathcal{I} = \{i(k)\}_{k=1}^{N}$ in stage S1, s different clusters $\{\mathcal{C}_j^{i^+}\}_{i^+=1}^{s}$ are constructed for each $j = 1, \dots, s$. Each cluster $\mathcal{C}_j^{i^+}$ contains the samples $\{\varphi(k)\}_{k=1}^{N-1}$ such that $i(k) = j$ and $i(k+1) = i^+$. The j-th partition Φ_j of the space Φ is thus obtained by separating the clusters $\{\mathcal{C}_j^{i^+}\}_{i^+=1}^{s}$ through a multi-class linear separation method.

As already introduced, $\varphi \in \Phi$ is supposed to be either equal to the regressor or the output. Among the two possible choices, the signal driving the transitions φ can be selected through cross-validation procedures or on the basis of the prior knowledge on the system.

Before providing technical details, it is worth remarking that the two-stage algorithm described above is based on a proper extension of the method introduced earlier by the authors in [3] for the identification of PieceWise Affine (PWA) functions, which represent a subclass of DHA systems. The main improvement with respect to [3] is that we now take into account the time evolution of the logic states, through the construction of s different partitions $\{\Phi_j\}_{j=1}^{s}$ of the space Φ.

A. Iterative clustering and parameter estimation

Stage S1 is carried out as described in Algorithm 1 where, to recursively update the values of the parameters θ_i and to generate the clusters $\mathcal{C}_i$, an approach similar to the one proposed in [3] is used. Even if Algorithm 1 is suited for both offline and online learning, only its execution in a batch (i.e., offline) mode will be considered.

The main idea of Algorithm 1 is to compute, at each time instant k, the estimation error $e_i(k)$ ($i \in \{1, \dots, s\}$) provided by all the s local affine submodels, and select the local model that "best fits" the current output observation $y(k)$ (Steps 3.1 and 3.2). The sequence of *active modes* $\mathcal{I}$ is then updated as in 3.4 and, at step 3.5, the parameter matrix $\theta_{i(k)}$ associated to the selected model is updated using the recursive least-squares algorithm in [1] based on inverse QR decomposition.

As already pointed out in [3], due to the greedy nature of Algorithm 1, the estimates of the model parameters θ_i and the clusters $\mathcal{C}_i$ are influenced by the the initial choice of the parameters θ_i. A possible initialization for the matrices θ_i is to take $\theta_1, \dots, \theta_s$ all equal to the best linear model, i.e.

$$\theta_i \equiv \arg\min_{\theta} \sum_{k=1}^{N} \|y(k) - \theta' x(k)\|_2^2, \ \forall i = 1, \dots, s. \quad (7)$$

Algorithm 1 Recursive clustering and parameter estimation algorithm

Input: Sequence of observations $\{x(k), y(k)\}_{k=1}^{N}$, desired number s of logic states, initial condition for the parameter matrices θ_i, $i = 1, \dots, s$.

```
1. let $\mathcal{I} \leftarrow \mathbf{0}_N$;
2. let $\mathcal{C}_i \leftarrow \emptyset$, $i = 1, \dots, s$;
3. for $k = 1, \dots, N$ do
  3.1. let $e_i(k) \leftarrow y(k) - \theta_i' \tilde{x}(k)$, $i = 1, \dots, s$;
  3.2. let
         $i(k) \leftarrow \arg\min_{i=1,\dots,s} e_i'(k) e_i(k);$     (6)
  3.3. let $\mathcal{C}_{i(k)} \leftarrow \mathcal{C}_{i(k)} \cup \{x(k), y(k)\}$;
  3.4. let $\mathcal{I}(k) \leftarrow i(k)$;
  3.5. update $\theta_{i(k)}$ using the recursive least-squares Algorithm [1];
4. end for;
5. end.
```

Output: Estimated matrices $\theta_1, \dots, \theta_s$, clusters $\mathcal{C}_1, \dots, \mathcal{C}_s$ and sequence of active modes $\mathcal{I}$.

Moreover, as suggested in [3], the estimation quality can be improved by reiterating Algorithm 1 multiple times, using its output as an initial condition for the following iteration.

Dealing with logic states with equal continuous dynamics

When Algorithm 1 is used to identify a DHA system with two or more logic states sharing the same continuous dynamics, the discrete behaviour (estimated at stage S2) may not be accurately reconstructed since data points originated from different logic states with the same continuous dynamics may all be associated with the same cluster. Thus, the function $f_H(\varphi(k))$ characterizing the event generator $\mathcal{E}$ might not be correctly retrieved. As a heuristic to overcome this problem, Algorithm 1 is extended as explained in Algorithm 2.

At step 2, the number $c_\emptyset$ of "almost empty" clusters is computed, where a cluster is considered "almost empty" if the percentage of points belonging to it is smaller than a threshold ε. Algorithm 1 is then run again with the reduced number of clusters $\bar{s} = s - c_\emptyset$ (steps 4-5).
However, as the number of logic states s is supposed to be a priori decided by the user, the parameter matrices associated with the discarded $s - \bar{s}$ logic states have to be retrieved. In order to accomplish this task, given the clusters $\bar{\mathcal{C}}_i$, $i = 1, \dots, \bar{s}$ resulting from step 5, the trace Σ_i of the sample covariance matrices R_i associated with each cluster is computed in step 7 in order to evaluate their "dispersion".K-means [7] is then iteratively applied starting from the cluster with maximum dispersion (denoted as $\bar{\mathcal{C}}_{j^\star}$ in step 9.3), until the desired number s of cluster is retrieved. In particular, at step 9, the chosen cluster $\bar{\mathcal{C}}_{j^\star}$ is partitioned into l subclusters. The parameter l (computed at stage 9.2) is set as the value corresponding to clustering solution leading to the smallest

Algorithm 2 Splitting clustering algorithm

Input: Sequence of training samples $\{x(k), y(k)\}_{k=1}^{N}$, desired number s of logic states; parameter matrices θ_i and clusters $\mathcal{C}_i$ $(i = 1, \ldots, s)$ provided by Algorithm 1; threshold $\varepsilon \geq 0$.

```
1. let $c_\emptyset \leftarrow 0$;
2. for $i = 1, \ldots, s$ do
 2.1. if $\frac{|\mathcal{C}_i|}{N} \leq \varepsilon$ let $c_\emptyset \leftarrow c_\emptyset + 1$;
3. end for;
4. let $\bar{s} = s - c_\emptyset$;
5. compute new clusters $\bar{\mathcal{C}}_i$ and parameter matrices $\theta_i$ $i = 1, \ldots, \bar{s}$ through Algorithm 1;
6. compute the clusters' sample covariance matrices $R_i,\ i = 1, \ldots, \bar{s}$;
7. let $\Sigma_i \leftarrow \mathrm{tr}(R_i)$, $i = 1 : \ldots, \bar{s}$;
8. let $\mu \leftarrow \bar{s}$ and $\beta \leftarrow \bar{s}$;
9. while $\mu < s$ do
  9.1. let $j^\star \leftarrow \arg\max_{j\in\{1,\ldots,\beta\}} \Sigma j$;
  9.2. let $l \leftarrow \arg\min_{q\in\{1,\ldots,(s-\mu+1)\}} DB(q)$;
  9.3. divide $\bar{\mathcal{C}}_{j^\star}$ into $l$ clusters through K-means [7];
  9.4. associate the parameter vector $\theta_{j^\star}$ to the obtained clusters;
  9.5. let $\beta \leftarrow \beta - 1$ and remove $\bar{\mathcal{C}}_{j^\star}$ from the set of clusters that can be partitioned;
  9.6. update $\mu \leftarrow \mu + (l - 1)$;
  9.7. let $\mathcal{C}_i$, $i = 1, \ldots, \mu$ be the new collection of clusters and $\theta_i$ the associated parameter matrices;
10. end while;
```

Output: Estimated parameter matrices $\theta_1, \ldots, \theta_s$, clusters $\mathcal{C}_1, \ldots, \mathcal{C}_s$ and updated sequence of active modes $\mathcal{I}$.

value of the Davies-Bouldin index [5] defined as

$$DB(q) = \frac{1}{q} \sum_{i=1}^{q} \max_{j \neq i} \left(\frac{\bar{d}_i + \bar{d}_j}{d_{ij}} \right). \tag{8}$$

where q corresponds to the number of created subclusters, $\bar{d}_i$ is the within-cluster distance of cluster i and d_{ij} is the distance between the centroids of the i-th and the j-th cluster. Once the $j^\star$-th cluster $\bar{\mathcal{C}}_{j^\star}$ is partitioned, the parameter vector $\theta_{j^\star}$ is associated with all the computed subclusters, at step 9.4. Moreover, in step 9.5, $\bar{\mathcal{C}}_{j^\star}$ is removed from the set of clusters that can further be divided and the number of clusters still to be retrieved is updated (step 9.6). At the end of Algorithm 2, s clusters are obtained, along with the corresponding parameter matrices $\{\theta_i\}_{i=1}^{s}$ and the resulting sequence of active modes $\mathcal{I}$.

B. Identification of the dynamics of logic states

Stage S1 provides the identified parameter matrices θ_i, $i = 1, \ldots, s$, along with the sequence of *active modes* $\mathcal{I} = \{i(k)\}_{k=1}^{N}$ obtained through recursive clustering of the training samples $\{x(k), y(k)\}_{k=1}^{N}$. Although, Stage S1 does not yield to a direct estimation of the law driving the discrete state, this information can be retrieved from the estimated sequence of *active modes* $\mathcal{I}$. Indeed, given the sample $\varphi(k)$, the sequence $\mathcal{I}$ provides an estimation of both the current mode $i(k)$ and the one-step ahead logic state $i(k+1)$. It is worth remarking that the signal $\varphi(k)$ is supposed to be either equal to the system output $y(k)$ or to the regressor $x(k)$.

Algorithm 3 presents a procedure to reconstruct the logic state dynamics. Specifically, s different polyhedral partitions Φ_j of the space Φ are defined. The partitions Φ_j are computed independently for each mode j, $j = 1, \ldots, s$.
At step 2.1, s different clusters $\{\mathcal{C}_j^{i^+}\}_{i^+=1}^{s}$ are constructed for each $j = 1, \ldots, s$. Each cluster $\mathcal{C}_j^{i^+}$ contains the samples $\{\varphi(k)\}_{k=1}^{N-1}$ such that $i(k) = j$ and $i(k+1) = i^+$. The j-th partition Φ_j of the space Φ is thus obtained by separating the clusters $\{\mathcal{C}_j^{i^+}\}_{i^+=1}^{s}$ through the multi-class linear separation method in [3] (step 2.3). It is worth pointing out that each of computed partitions Φ_j $(j = 1, \ldots, s)$ is defined by a set of s polyhedra. Such a set of polyhedra is described by a piecewise affine separator obtained as the maximum of s affine functions of $\varphi(k)$. The reader is referred to [3] for further details on the employed (offline) multi-class linear separation method.

Algorithm 3 Logic state dynamics identification algorithm

Input: Samples of $\{\varphi(k)\}_{k=1}^{N}$; identified sequence of logic states $\mathcal{I} = \{i(k)\}_{k=1}^{N}$.

```
1. let $\mathcal{C}_j^{i^+} \leftarrow \emptyset$,   $j, i^+ = 1, \ldots, s$;
2. for $j = 1, \ldots, s$ do
 2.1. for $i^+ = 1, \ldots, s$ do
   2.1.1. for $k = 1, \ldots, N - 1$ do
     2.1.1.1. if $i(k) = j$ and $i(k+1) = i^+$
     2.1.1.2. let $\mathcal{C}_j^{i^+} \leftarrow \mathcal{C}_j^{i^+} \cup \{\varphi(k)\}$;
     2.1.1.3. end if;
   2.1.2. end for;
 2.2. end for;
 2.3. for all $j = 1, \ldots, s$, compute the polyhedral partition $\Phi_j$ of the space $\Phi$ by separating the clusters $\mathcal{C}_j^{i^+}$ (with $i^+ = 1, \ldots, s$) through the multi-class linear separation method in [3].
3. end for;
```

Output: Polyhedral partitions Φ_j, $j = 1, \ldots, s$.

V. Case studies

In order to show the effectiveness of the proposed identification technique, two examples are considered in this section: a simulation example and a simple experimental case study. All computations are carried out on a 2.8-GHz Intel Core i7 with 16 GB of RAM running MATLAB R2015a.

In both the examples it is supposed to be known a priori that $\{\varphi(k)\}_{k=1}^{N}$ is set equal to $\{y(k)\}_{k=1}^{N}$. The trained models are validated through the comparison of the simulated results with an actual output sequence not used for training. The initial logic state of the system is supposed to be available in validation.

The quality of the estimated models is assessed on a validation dataset not used for training, through the Best Fit Rate (BFR) indicator defined as:

$$\mathrm{BFR} = \max\left\{1 - \frac{||\hat{y} - y||_2}{||y - \bar{y}||_2}, 0\right\} \cdot 100\% \tag{9}$$

where y and $\hat{y}$ denote the vector stacking the actual and (open-loop) simulated outputs, respectively, and $\bar{y}$ is the sample mean of y. The time evolution of the logic state is simulated, in open-loop, through the polyhedral partitions computed by Algorithm 3.

A. Example 1: Identification of a 4 modes DHA

As a first example, we consider the identification of the DHA depicted in Fig. 1, where the SAS is constituted by 4 affine subsystems, two of which share the same dynamical behaviour. Each subsystem is described by the following first-order difference equation with no exogenous input signal:

$$y(k) = \theta_{i(k),1} y(k-1) + \theta_{i(k),2}. \tag{10}$$

The output is supposed to be corrupted by a noise $e_o(k)$, which is taken as a realization of an additive zero-mean white noise stochastic process with Gaussian distribution and variance $\Lambda_e = 1$. The true values of the parameters $\theta_{j,1}$ and $\theta_{j,2}$ (with $j = 1, \dots, 4$) are reported in Table I. The system is estimated based on a set of $N = 5000$ output observations. Note that, in this example, the regressor $x(k)$ is given by $y(k-1)$, and the space $\Phi = \mathcal{Y}$ coincides with $\mathbb{R}$. In order to assess the effect of the noise $e_o(k)$, the Signal-to-Noise Ratio (SNR), defined as:

$$\mathrm{SNR} = 10 \log \frac{\sum_{k=1}^{N} (y(k) - e_o(k))^2}{\sum_{k=1}^{N} e_o^2(k)}, \tag{11}$$

is computed, and it is equal to 34.9 dB.
Both Algorithm 1 and 2 are executed. As expected, only the second one leads to an accurate estimate of the system as mode S_1 and S_3 share the same continuous dynamics (see the true parameters θ_1 and θ_3 in Table I). Algorithm 2 is run 10 times, until convergence of the estimated model parameters is achieved. The threshold ε is calibrated through cross-validation, using a set of 500 noisy samples not employed in the training phase, and it is set equal to 0.05. This means that the clusters that are graded as "almost empty" are all the ones constituted by less than $250 = N\varepsilon$ samples. This value allows us to recognise that at least two modes share the same continuous dynamics.

Based on the results obtained from Algorithm 2, $s = 4$ partitions Φ_j ($j = 1, \dots, 4$) of the space Φ are computed through Algorithm 3 in order to estimate the discrete state dynamics. The CPU time required to compute each partition Φ_j ($j = 1, \dots, s$) is, in average, 0.018 s, while the entire identification procedure takes 1.215 s to train the model.

The oriented graph describing the estimated DHA model is depicted in Fig. 2 and the identified parameters $\{\theta_i\}_{i=1}^4$ defining each affine dynamical model are reported in Table I. The estimated partitions are shown in Fig. 3 and, from their

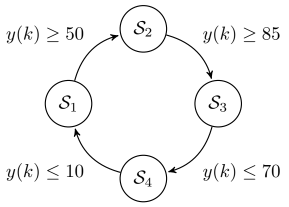

$$\begin{tikzcd} & \mathcal{S}_2 \arrow[dr, bend left, "y(k)\geq 85"] & \\ \mathcal{S}_1 \arrow[ur, bend left, "y(k)\geq 50"] & & \mathcal{S}_3 \arrow[dl, bend left, "y(k)\leq 70"] \\ & \mathcal{S}_4 \arrow[ul, bend left, "y(k)\leq 10"] & \end{tikzcd}$$

Fig. 1. Example 1: Oriented graph schematizing the data generating system. If the conditions on the edges are not satisfied, the system remains in the same logical state.

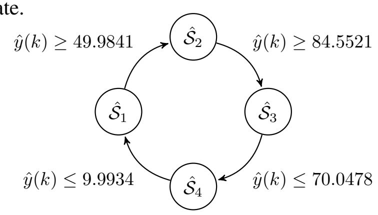

$$\begin{tikzcd} & \hat{\mathcal{S}}_2 \arrow[dr, bend left, "\hat{y}(k)\geq 84.5521"] & \\ \hat{\mathcal{S}}_1 \arrow[ur, bend left, "\hat{y}(k)\geq 49.9841"] & & \hat{\mathcal{S}}_3 \arrow[dl, bend left, "\hat{y}(k)\leq 70.0478"] \\ & \hat{\mathcal{S}}_4 \arrow[ul, bend left, "\hat{y}(k)\leq 9.9934"] & \end{tikzcd}$$

Fig. 2. Example 1: Oriented graph representing the identified model, outlining the estimated switching conditions.

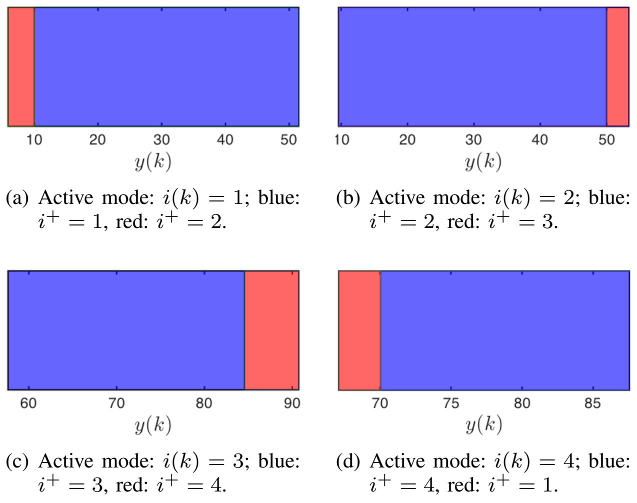
definition, the switching conditions presented in Fig. 2 are retrieved. Note that, in Fig. 3, each partition is characterized by only two polytopes. This is conformed to the true DHA system, where each logic state can be followed only by two other states (including itself).

Finally, the identified model is validated on a noiseless dataset of length $N_V = 200$. The obtained BFR is 99.51%, showing the effectiveness of the proposed identification method even in the challenging situation of two modes sharing the same continuous dynamics.

B. Example 2: Modelling of switching RC circuit

The proposed algorithm is also tested on an experimental case study consisting on the modelling of an RC circuit with switching load, whose schematic and switching pattern are represented in Fig. 4.

A 10 μF capacitor C and three 10 kΩ resistors R_1, R_2 and R_3 are used. The ON/OFF switches are implemented using MOSFETs. An Arduino UNO board is used for: (i)



(a) Active mode: $i(k) = 1$; blue: $i^+ = 1$, red: $i^+ = 2$.

(b) Active mode: $i(k) = 2$; blue: $i^+ = 2$, red: $i^+ = 3$.

(c) Active mode: $i(k) = 3$; blue: $i^+ = 3$, red: $i^+ = 4$.

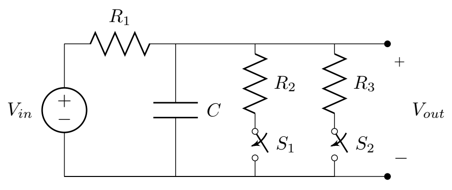
(d) Active mode: $i(k) = 4$; blue: $i^+ = 4$, red: $i^+ = 1$.

Fig. 3. Example 1: polyhedral partitions computed through Algorithm 3, with $i(k+1) = i^+$.

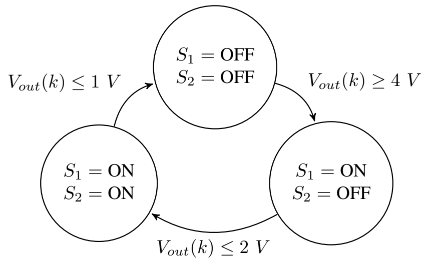
TABLE I

EXAMPLE 1: TRUE AND ESTIMATED PARAMETERS FOR EACH MODE OF THE SAS.

	S_1		S_2		S_3		S_4	
	$\theta_{1,1}$	$\theta_{1,2}$	$\theta_{2,1}$	$\theta_{2,2}$	$\theta_{3,1}$	$\theta_{3,2}$	$\theta_{4,1}$	$\theta_{4,2}$
true	0.9	6	0.8	20	0.9	6	0.7	0
estimated	0.9003	5.9628	0.7911	20.6345	0.9003	5.9628	0.6996	0.0451



(a) Schematic of the switching RC circuit.



(b) Oriented graph representing the discrete state dynamics of the system.

Fig. 4. Example 2: considered RC circuit and representation of its logic behaviour.

measuring the output voltage $y(k) = V_{out}(k)$; (ii) generating the input voltage $u(k) = V_{in}(k)$; (iii) driving the switches according to the logic behaviour represented in Fig. 4(b). A piecewise-constant signal is applied as input voltage $V_{in}(k)$ to generate the training and the validation datasets. Both of these sets have length 2000. The output voltage $V_{out}(k)$ is measured, at a sampling time of T_s = 100 ms, with an analog-to-digital (A/D) converter available on the Arduino board[2].

The number of logic states is set to $s = 3$. Each dynamical local model, identified through Algorithm 1, is described by a linear difference equation of order 1, relating $y(k)$ to the regressor $x(k) = [y(k-1)\ u(k-1)]'$. Algorithm 1 is iterated 10 times, and $s = 3$ partitions of the space $\Phi = \mathcal{Y}$ are computed through Algorithm 3. The overall computational time required to identify a DHA model of the circuit is 0.078 s, of which 0.033 s are taken to compute the 3 partitions.

In order to assess the quality of the estimated model, the (open-loop) simulated output $\hat{y}$ of the identified DHA is computed, and plotted in Fig. 5, along with the actual output y composing the validation set. A BFR of 98.64 % is achieved and, of the 2000-length sequence of logic states, only 8 are not correctly recognized. From Fig. 5, it can be noticed that the difference between the simulated and the actual output is negligible.

VI. CONCLUSIONS

This paper has proposed a numerically efficient two-stage algorithm for black-box identification of discrete hybrid automata (DHA). The approach consists of two stages: simultaneous clustering of the observations and estimation of the submodel associated to each mode of the DHA; computation of a set of polyhedral partitions of the space Φ used to describe the logical state dynamics. A simulation and an experimental example are reported to show the effectiveness of the proposed approach.

Future research activities include the extension of the presented approach to the identification of hybrid systems with stochastic switching (e.g., Markov processes with hidden states) and its use in challenging real-world applications.

[2]The A/D converter used in the experiment has an input rage of 0-5 V and a resolution of 10 bits.

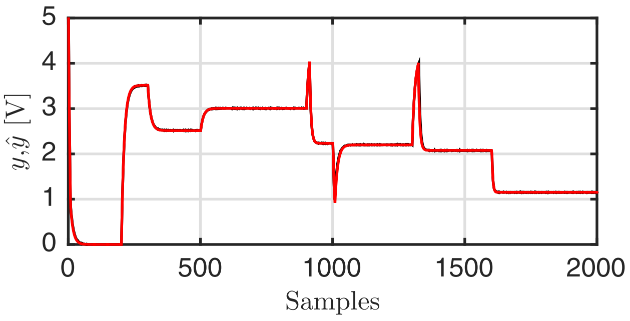


Fig. 5. Example 2: actual (black) and simulated (red) output.

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] V. Breschi, A. Bemporad, and D. Piga. Identification of hybrid and linear parameter varying models via recursive piecewise affine regression and discrimination. In *Proc. of the 15th European Control Conference*, pages 2632–2637, 2016.

[3] V. Breschi, D. Piga, and A. Bemporad. Piecewise affine regression via recursive multiple least squares and multicategory discrimination. *Automatica*, 73:155–162, 2016.

[4] M.P. Cabasino, P. Darondeau, M.P. Fanti, and C. Seatzu. *Model Identification and Synthesis of Discrete-Event Systems*. John Wiley and Sons, Inc., 2015.

[5] D.L. Davies and D.W. Bouldin. A cluster separation measure. *IEEE Trans. Pattern Anal. Mach. Intell.*, 1(2):224–227, February 1979.

[6] 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.

[7] J.A. Hartigan and M.A. Wong. Algorithm AS 136: A K-means clustering algorithm. *Applied Statistics*, pages 100–108, 1979.

[8] W.P.M.H. Heemels, B. De Schutter, and A. Bemporad. Equivalence of hybrid dynamical models. *Automatica*, 37(7):1085–1091, 2001.

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

[10] O. Niggemann, B. Stein, A. Vodencarevic, A. Maier, and H.K. Büning. Learning behavior models for hybrid timed systems. *AAAI*, 2:1083–1090, 2012.

[11] F.D. Torrisi and A. Bemporad. HYSDEL-a tool for generating computational hybrid models for analysis and synthesis problems. *IEEE Transactions on Control Systems Technology*, 12(2):235–249, 2004.