Proceedings of the American Control Conference
Arlington, VA June 25-27, 2001

Identification of Piecewise Affine and Hybrid Systems

Giancarlo Ferrari-Trecate[†], Marco Muselli[‡], Diego Liberati[§], Manfred Morari[†]

[†] Institut für Automatik, ETH - Swiss Federal Institute of Technology, ETHZ - ETL, CH-8092 Zurich, Switzerland,
Tel. +41-1-6327812, Fax +41-1-6321211, E-mail: {ferrari,morari}@aut.ee.ethz.ch

[‡] Istituto per i Circuiti Elettronici, Consiglio Nazionale delle Ricerche, Via De Marini, 6 I-16149 Genova, Italy,
Tel: +39-010-6475213, Fax : +39-010-6475200, E-mail: muselli@ice.ge.cnr.it

[§] Centro di studio sulle Tecnologie dell'Informatica e dell'Automazione, Consiglio Nazionale delle Ricerche,
Dipartimento di Elettronica e Informazione, Politecnico di Milano, Piazza Leonardo da Vinci 32 I-20133 Milano, Italy
Fax : +39-02-23993412, E-mail: liberati@elet.polimi.it

Abstract

In this paper we focus on the identification of discrete-time hybrid systems in the Piece-Wise Affine (PWA) form. This problem can be formulated as the reconstruction of a possibly discontinuous PWA map with a multi-dimensional domain. In order to achieve our goal, we propose an algorithm that exploits the combined use of clustering, linear identification, and classification techniques. This allows to identify both the affine submodels and the polyhedral partition of the domain on which each submodel is valid.

Keywords: Nonlinear identification, hybrid systems, clustering, linear regression, classification.

1 Introduction

In recent years a large stream of research focused on hybrid systems because of their capability to describe the interaction between dynamic and logical components. Among other formalisms for hybrid systems, Bemporad and Morari [2] developed a unified model for a wide range of discrete-time hybrid systems, the *Mixed Logic Dynamical (MLD)* form [2]. For instance, linear hybrid dynamical systems, hybrid automata, Piece-Wise Affine (PWA) systems and some classes of discrete-event systems can be represented in the MLD form. The MLD representation is also suitable to solve many analysis and synthesis problems (like model predictive control [2], state estimation [7], verification problems [3], observability and controllability tests [1]) by expressing them as mixed-integer programs.

In this paper we focus on the problem of identifying hybrid systems in the MLD form from experimental data. We exploit the equivalence between MLD and Piece-Wise Affine (PWA) systems [1] and we study the identification problem for the latter class. In Section 2 we introduce the class of input-output models that reflects the structure of a PWA system. Not surprisingly those models are obtained by generalizing classical AutoRegressive eXogenous (ARX) models to Piecewise ARX (PWARX) models. PWARX models are obtained by partitioning the space of the regressors in a finite number of polyhedra and by considering an affine submodel on each one of those regions. Therefore the identification problem can be viewed as the problem of reconstructing a PWA map with a multi-dimensional domain from a finite set of input-output measurements.

In the past years, the Neural Networks community developed algorithms to solve regression problems with PWA maps. Among others, one may cite Breiman's hinging Hyperplanes [6] and multilayer neural networks with PWA activation functions [10]. However all such algorithms focus on the estimation of a *continuous* PWA function. The key feature of PWARX models is that the output map can be discontinuous along the boundary of the regions. This is due to the fact that many logic conditions can be represented through discontinuities in the state-update and output maps of a PWA system. Therefore, an identification algorithm for hybrid systems should be capable of identifying *discontinuos* PWA maps. To the knowledge of the authors such problem received very little attention so far. In a very recent work [9] a regression problem with monodimensional PWA maps was considered whereas multilayer neural networks with logic gates are proposed in [13].

The main difficulty in identifying discontinuos PWA maps is that the estimation of the linear submodels cannot be separated from the problem of finding the domain of each sub-model. In order to achieve our goal, we exploit a combined use of *clustering*, *classification* and *linear identification* techniques. The various steps of the main algorithm are illustrated in Section 3. Moreover, in Section 4 we discuss the algorithm highlighting future research directions and possible modifications in order to estimate also the number of submodels and the model orders from the data set.

2 Problem statement

A PWA system is defined by the state-space equations

$$\begin{array}{rcl} z(t+1) & = & A_i z(t) + B_i u(t) + d_i \\ y(t) & = & C_i z(t) + g_i \\ (z,u) \in \mathcal{Z} & & \end{array}, \text{ for } \left[\begin{smallmatrix} z(t) \\ u(t) \end{smallmatrix}\right] \in \mathcal{Z}_i \tag{1}$$

$z \in \mathbb{R}^{n_c} \times \{0,1\}^{n_\ell}$ are the continuous and binary states, $u \in \mathbb{R}^{m_c} \times \{0,1\}^{m_\ell}$ are the inputs, $y \in \mathbb{R}^{p_c} \times \{0,1\}^{p_\ell}$ are the outputs. Moreover, the state+input set $\mathcal{Z} \subset \mathbb{R}^{n_c} \times \{0,1\}^{n_\ell} \times \mathbb{R}^{m_c} \times \{0,1\}^{m_\ell}$ is a bounded polyhedron, $\{\mathcal{Z}_i\}_{i=1}^{s}$ is a polyhedral partition[1] of $\mathcal{Z}$ and d_i, g_i are constant vectors of suitable dimension. In particular, we restrict our attention to single-output PWA systems, i.e. $p_c + p_\ell = 1$.

Since input/output data are used for the identification of a PWA system, we need an *input/output* description of (1). We consider Piecewise ARX (PWARX) models that are defined relying on the s submodels

$$y(k) = \begin{cases} a_{1,1}y(k-1) + a_{1,2}y(k-2) + \ldots + \\ \quad + a_{1,n_a}y(k-n_a) + b_{1,1}u(k-1) + b_{1,2}u(k-2) + \\ \quad + \ldots + b_{1,n_b}u(k-n_b) + f_1 + \epsilon_k \\ \vdots \\ a_{s,1}y(k-1) + a_{s,2}y(k-2) + \ldots + \\ \quad + a_{s,n_a}y(k-n_a) + b_{s,1}u(k-1) + b_{s,2}u(k-2) + \\ \quad + \ldots + b_{s,n_b}u(k-n_b) + f_s + \epsilon_k \end{cases} \tag{2}$$

where u and y are the inputs and the output respectively, f_i are displacements and ϵ_k are noise samples. We consider a simple noise model by assuming that ϵ_k are Gaussian independent identically distributed random variables with zero mean and variance σ^2. The vector of the regressors is denoted by

$$x(k) \triangleq [y(k-1) \quad \ldots \quad y(k-n_a) \quad u(k-1) \quad \ldots \quad u(k-n_b)]'$$

and we assume that the regressors lie in a bounded polyhedron $\mathcal{X}$ hereafter referred to as *regressor set*. Obviously, $\mathcal{X} \subset \mathbb{R}^n$ where $n = n_a + n_b$. In order to specify a PWARX model completely, a polyhedral partition $\{\mathcal{X}_i\}_{i=1}^{s}$ of $\mathcal{X}$ is given and the switching law between the models is specified by the rule: if $x(k) \in \mathcal{X}_i$, the i-th dynamic of (2) is active. When an input/output pair $(x(k), y(k))$ is such that $x(k) \in \mathcal{X}_i$ we say that the pair belongs to the i-th submodel.

An advantage of model (2) is that it is easy to map this representation into the standard state-space form (1). In fact it is enough to choose the state vector as $z(k) = x(k)$, to set $\mathcal{Z} = \mathcal{X}$ and $\mathcal{Z}_i = \mathcal{X}_i$, $i = 1, \ldots, s$ and to exploit classical realization theory (for instance controllability/observability canonical forms) in order to derive the matrices A_i, B_i, C_i, d_i and g_i from the parameters of the i-th submodel in (2).

Throughout this paper we assume that N input/output points $(y(k), u(k))$, $k = 0, \ldots, N$ have been collected. These are the data available for the identification of the PWARX model.

Assumption 1 *The data are generated from the PWARX model (2) specified by the orders $\bar{n}_a$, $\bar{n}_b$ the number of submodels $\bar{s}$, the parameter vectors*

$$\bar{\theta}_i = [\bar{a}_{i,1} \quad \bar{a}_{i,2} \quad \ldots \quad \bar{a}_{i,\bar{n}_a}\bar{b}_{i,1} \quad \bar{b}_{i,2} \quad \ldots \quad \bar{b}_{i,\bar{n}_b} \quad \bar{f}_i]' \tag{3}$$

and the sets $\bar{\mathcal{X}}$, $\bar{\mathcal{X}}_i$, $i = 1, \ldots, \bar{s}$,

Remark 1 If the data are generated according to Assumption 1 and $\bar{n}_a$, $\bar{n}_b$, $\bar{s}$, $\bar{\mathcal{X}}$ and $\bar{\mathcal{X}}_i$, $i = 1, \ldots, \bar{s}$ are known, the identification problem amounts to identify the $\bar{s}$ ARX submodels in (2). In fact, since the sets $\bar{\mathcal{X}}_i$ are known, we can *classify* the vectors $x(k)$ i.e. partition them in $\bar{s}$ sets $\mathcal{F}_i$ according to the rule $(x(k), y(k)) \in \mathcal{F}_i$ if $x(k) \in \bar{\mathcal{X}}_i$. Then the i-th submodel can be identified by using the datapoints in the set $\mathcal{F}_i$.

The identification problem becomes non-trivial if we do not know all the quantities mentioned in Remark 1. In particular the difficulty of the identification problem depends on which quantities are assumed to be known.

Assumption 2 *The number of submodels s is given.*

The number of models depends on the number of operative conditions in which the data are collected. For instance one can collect data knowing in advance that the systems may only switch between a normal and a faulty operating condition, i.e. $s = 2$. In this paper we first exploit Assumption 2 and in Section 4 we discuss how to estimate s from the data.

Assumption 3 *The orders n_a and n_b are fixed. If Assumption 1 holds, $\bar{n}_a$ and $\bar{n}_b$ are known and $n_a = \bar{n}_a$, $n_b = \bar{n}_b$.*

Assumption 3 is useful to discuss the property of the identified PWARX model, especially when Assumption 1 holds as well. For instance, consistency properties of ARX models are often proved in this scenario (i.e. the true system belongs to the class of models considered [11]). However, in practice $\bar{n}_a$ and $\bar{n}_b$ are seldom known and a large stream of research focused on the estimation of the "best" orders [11, 14] of ARX models. Note that if $\bar{n}_a$ and $\bar{n}_b$ are unknown, also the dimension $\bar{n} = \bar{n}_a + \bar{n}_b$ of $\bar{\mathcal{X}}$ is unknown. This means that neither $\bar{\mathcal{X}}$ nor the partition $\bar{\mathcal{X}}_i$, $i = 1, \ldots, s$ can be in general assumed as given. Assumption 3 will be used in order to focus on the peculiarities of the identification of PWARX systems by avoiding the difficulties arising from the order estimation. Nevertheless, in order to study the identification problem in a fair scenario, the regions $\bar{\mathcal{X}}_i$ are assumed unknown.

[1] Each set $\mathcal{Z}_i$ is a (not necessarily closed) convex polyhedron s.t. $\mathcal{Z}_i \bigcap \mathcal{Z}_j = \emptyset$, $\forall i \neq j$, $\bigcup_{i=1}^{s} \mathcal{Z}_i = \mathcal{Z}$.

Concerning the set $\mathcal{X}$, once n_a and n_b are chosen, it is often possible to specify a *candidate regressor set* $\mathcal{X}$. In fact, the shape of $\mathcal{X}$ should reflect physical constraints on the inputs and the output of the system. In practice, it is common to specify constraints in terms of box-bounds on each input/output sample or on each input/output increment. For example, typical constraints on the output are

$$|y(t)| \leq y_{max}, \text{ or } |y(t+1) - y(t)| \leq \Delta y_{max} \qquad (4)$$

Then it is easy to derive from (4) the explicit representation of the bounded polyhedron $\mathcal{X} \in \mathbb{R}^n$ where the regressors lie.

3 The main algorithm

Based on the previous discussion, the first identification problem we consider reads as

Problem 1 *Assume that the data $(y(k), u(k))$, $k = 0, \dots, N$ are generated according to Assumption 1 and that Assumptions 2 and 3 hold true. Estimate the partition $\mathcal{X}_i$, $i = 1, \dots, s$ and the parameter vectors*

$$\theta_i' = [a_{i,1} \;\; a_{i,2} \;\; \dots \;\; a_{i,n_a} b_{i,1} \;\; b_{i,2} \;\; \dots \;\; b_{i,n_b} f_i] \qquad (5)$$

characterizing the PWARX model (2) on the basis of the collected data.

The main difficulty in solving Problem 1 is that the problem of estimating the regions $\bar{\mathcal{X}}_i$ cannot be decoupled from the identification of each submodel. The algorithm we propose to solve Problem 1 exploits a combined use of clustering, classification and linear regression techniques. Such a procedure will be illustrated by using the following toy example.

Example 1 *The data are generated by the PWARX system*

$$y(k) = \begin{cases} \begin{bmatrix} -1 & 0 \end{bmatrix} \begin{bmatrix} u(k-1) & 1 \end{bmatrix}' + \epsilon(k), \\ \quad \text{if } u(k-1) = x(k) \in \bar{\mathcal{X}}_1 = [-4, 0] \\ \\ \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} u(k-1) & 1 \end{bmatrix}' + \epsilon(k), \\ \quad \text{if } u(k-1) = x(k) \in \bar{\mathcal{X}}_2 = (0, 2) \\ \\ \begin{bmatrix} 3 & -2 \end{bmatrix} \begin{bmatrix} u(k-1) & 1 \end{bmatrix}' + \epsilon(k), \\ \quad \text{if } u(k-1) = x(k) \in \bar{\mathcal{X}}_3 = [2, 4] \end{cases} \qquad (6)$$

where $\bar{s} = 3$, $\bar{n}_a = 0$, $\bar{n}_b = 1$, $\mathcal{X} = [-4, 4]$, and the input samples $u(k) \in \mathbb{R}$ are generated randomly according to the uniform distribution on $\mathcal{X}$.

The system and a data set of 50 samples with noise variance $\sigma^2 = 0.01$ are depicted in Figure 1.

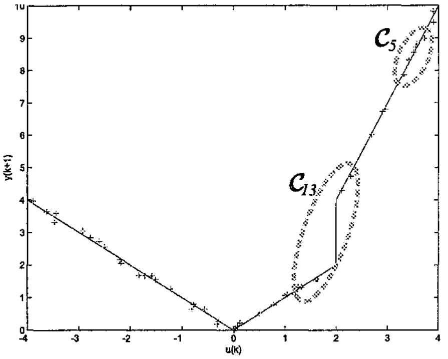

Figure 1: The PWARX system (6) (-) and the dataset (crosses)

The first step of our algorithm is to cluster the datapoints $(x(k), y(k))$ in a suitable way. In fact, a PWA map is *locally linear.* Thus small subsets of points $x(k)$ that are close each other are likely to belong to the same region $\bar{\mathcal{X}}_i$ [12]. For a fixed integer c and for each datapoint $(x(j), y(j))$, $j = 1, \dots, N$, we build a cluster $\mathcal{C}_j$ collecting $(x(j), y(j))$ and the $c-1$ distinct datapoints $(\tilde{x}, \tilde{y})$ that satisfy

$$\forall (\tilde{x}, \tilde{y}) \in \mathcal{C}_j, \;\; \|x(j) - \tilde{x}\|^2 \leq \|x(j) - \hat{x}\|^2, \;\; \forall (\hat{x}, \hat{y}) \in \mathcal{S} \backslash \mathcal{C}_j. \qquad (7)$$

In other words the cluster $\mathcal{C}_j$ collects the point $x(j)$ and its $c-1$ neighboring samples along with the corresponding output samples. Note that each cluster $\mathcal{C}_j$ can be labeled with the point $x(j)$ so having a bijective map between x-points and clusters. The parameter c has to be fixed by the user and this is a knob of our algorithm that can be adjusted. Some clusters will collect only data belonging to a single submodel (for instance the cluster $\mathcal{C}_5$ in Figure 1). Those clusters will be referred to as *pure* clusters. Clusters collecting data generated by different submodels will be called *mixed* clusters (see the cluster $\mathcal{C}_{13}$ in Figure 1).

We assume that $c > n$ so that we can identify an affine model by using the samples contained in each cluster. For this purpose every linear regression technique can be used and we adopt least squares estimation. The vector of coefficients $\theta^{LS,j}$ estimated from the data in $\mathcal{C}_j$ is then computed through the well-known formula

$$\begin{aligned} \theta^{LS,j} &= (\Phi_j' \Phi_j)^{-1} \Phi_j' y_{\mathcal{C}_j}, \qquad (8) \\ \Phi_j &= \begin{bmatrix} x_1 & x_2 & \dots & x_c \\ 1 & 1 & \dots & 1 \end{bmatrix}' \end{aligned}$$

where x_i are the vectors of regressors belonging to $\mathcal{C}_j$ and $y_{\mathcal{C}_j}$ is the vector of the output samples in $\mathcal{C}_j$. For the toy example, the vectors $\theta^{LS,j}$ corresponding to clusters with $c = 6$ elements are shown in Figure 2. It is apparent that the vectors $\theta^{LS,j}$ belong to the *dual space* $\mathbb{R}^{n+1}$.

If the data are corrupted by a small amount of noise and c is "small enough", a picture of $\theta^{LS,j}$, $j = 1, \dots, N$

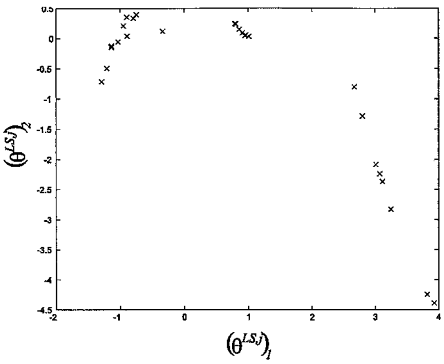

Figure 2: The vectors $\theta^{LS,j}$ in the dual space

should show s major clusters and some isolated points hereafter referred to as *outliers*. In fact we observe that if $\mathcal{C}_{j_1}$ and $\mathcal{C}_{j_2}$ are pure clusters collecting datapoints belonging to the same submodel, then θ^{LS,j_1} and θ^{LS,j_2} should be similar. In the limit case of noiseless data ($\sigma^2 = 0$) all such vectors coincide. The outliers correspond to parameter vectors computed from mixed clusters.

At this point the role of the parameter c should be clear: if the signal to noise ratio is high, even a low c would produce well defined clusters in the dual space. Moreover a low c means that the ratio between the number of mixed and non mixed clusters is low and then the number of isolated points in the dual space is low. For instance, from Figure 1 it is apparent that with $c = 2$ the number of mixed clusters is at most 4. Anyway, when noise is present, a low c means that the parameter vectors will be poorly estimated i.e. they will have a high variance. Then, it may happen that the clusters in the dual space become overlapping, thus preventing a good partitioning of the parameter vectors. In this case, the natural remedy is to increase c. On the other hand, if c is too large a high number of mixed clusters (and then outliers) will be generated. In the limit case $c = N$ all the clusters are mixed and all the $\theta^{LS,j}$ will be outliers. Then, in order to have well defined clusters, a good choice of the parameter c is always a trade-off between the two phenomena described above. A reasonable idea would be to tune c with cross-validation techniques and this will be the subject of further investigations. On the basis of these remarks, the next step amounts to clustering the parameter vectors in s disjoint subsets.

The clustering technique we adopt is a variation of the classical *K-means* algorithm. K-means (see [5, 8] for comprehensive tutorials) is a clustering algorithm where the number of clusters is given that seeks to partition the data points $\theta^{LS,j}$ into s disjoint subsets $\mathcal{D}_i$ in such a way to minimize the sum-of-squares clustering function given by

$$J = \sum_{i=1}^{s} \sum_{\theta^{LS,j} \in \mathcal{D}_i} \|\theta^{LS,j} - \mu^i\|^2. \tag{9}$$

In (9) μ^i is the mean of the data points in set $\mathcal{D}_i$ and is given by

$$\mu^i = \frac{1}{N_i} \sum_{\theta^{LS,j} \in \mathcal{D}_i} \|\theta^{LS,j}\|^2 \tag{10}$$

where N_i is the cardinality of $\mathcal{D}_i$. However, the K-means algorithm does not exploit all the information we have on the vectors $\theta^{LS,j}$.

Consider the vectors $\theta^{LS,j}$ identified on a pure cluster. Then, a classical result on least squares [11] states that $\theta^{LS,j}$ is a Gaussian random vector whose mean and covariance are given by

$$\mathrm{E}\left[\theta^{LS,j}\right] = \bar{\theta}_i, \quad \mathrm{Cov}\left[\theta^{LS,j}\right] = \sigma^2(\Phi_j'\Phi_j)^{-1}. \tag{11}$$

Since the variance σ^2 is unknown a priori, a common way to estimate the covariance matrix is

$$\begin{aligned} V_j &= \frac{SSR_j}{c-n+1}(\Phi_j'\Phi_j)^{-1}, & (12)\\ SSR_j &= y_{\mathcal{C}_j}'\left(I - \Phi_j(\Phi_j'\Phi_j)^{-1}\Phi_j'\right) y_{\mathcal{C}_j} & (13) \end{aligned}$$

where SSR_j is the sum of the squared residuals.

In the K-means algorithm, the centers μ^i of the clusters should represent the vectors $\bar{\theta}_i$. Then, it is natural to weight the deviation of the points from the centers with the matrices V_j^{-1}. This can be done by simply replacing the cost functional (9) with

$$\tilde{J} = \sum_{j=1}^{s} \sum_{\theta^{LS,j} \in \mathcal{D}_i} \|\theta^{LS,j} - \mu^j\|^2_{V_j^{-1}}. \tag{14}$$

Note that this modification changes only the norm used in K-means and does not spoil the efficiency of the algorithm.

Concerning the parameter vectors identified from data in mixed clusters, it is sensible that they have a high variance. Indeed we fit a single model to data belonging to (at least) two different models. It turns out that the cost functional (14) puts little emphasis on such parameter vectors. Then it is expected that the centers μ^j will mainly depend on the $\theta^{LS,j}$ based on pure clusters. The output of this second clustering algorithm are the s disjoint sets $\mathcal{D}_i$, of θ-points. The result concerning the toy problem is plotted in Figure 3.

By using the bijective maps between coefficient vectors and clusters $\mathcal{C}_j$ and between clusters $\mathcal{C}_j$ and points $(x(j), y(j))$, we can map back the clusters $\mathcal{D}_i$ from the dual space to the data-space thus classifying the original datapoints. More precisely this can be done in the following way.

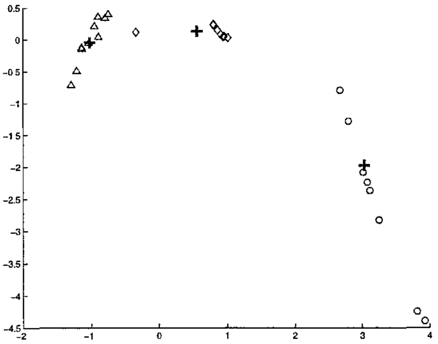

Figure 3: Clustering of the vectors $\theta^{LS,j}$ with the modified K-means algorithm. Triangles: first cluster, diamonds: second cluster, circles: third cluster. The crosses are the centers of each cluster.

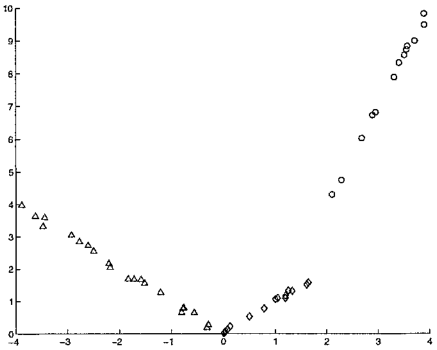

Figure 4: Clustering of the datapoints with Algorithm 1. Triangles: first cluster, diamonds: second cluster, circles: third cluster.

Algorithm 1

```
Let $\mathcal{D}_i$, $i = 1,\dots,s$ be specified by the
modified $K$-means algorithm and set $\mathcal{F}_i = \emptyset$.
for $i = 1:s$
 1. for all $\theta^{LS,j} \in \mathcal{D}_i$
  1.1. find the point $(x(j), y(j))$ by using
       the following maps: $\theta^{LS,j} \mapsto \mathcal{C}_j \mapsto$
       $(x(j), y(j))$.
  1.2. Add the pair $(x(j), y(j))$ to $\mathcal{F}_i$.
 2. end for
end for $i$
```

For the toy problem, the sets $\mathcal{F}_i$, $i = 1, 2, 3$ computed by Algorithm 1 are drawn in Figure 4.

Since the original data are now classified, it is possible to identify the final s ARX submodels on the basis of the s clusters $\mathcal{F}_i$ in the data space. Again we exploit least squares to accomplish this task. One may guess that this step is superfluous, since the centers μ^j already yield an estimate of the coefficient vectors. However the estimate provided by μ^j may be poor, especially if the parameter c (the cardinality of the $\mathcal{C}$-clusters) is close to $n + 1$. Moreover the use of least squares allows checking the goodness of each submodel by using standard criteria like confidence intervals.

For the toy problem, by using the final clusters $\mathcal{F}_i$, $i = 1, 2, 3$ represented in Figure 4, we obtained the following estimates

$$\begin{aligned}\theta_1' &= \begin{bmatrix} -1.0228 & -0.0446 \end{bmatrix}\\ \theta_2' &= \begin{bmatrix} 0.9466 & 0.0445 \end{bmatrix}\\ \theta_3' &= \begin{bmatrix} 3.0247 & -2.0976 \end{bmatrix}\end{aligned}$$

that provide a good approximation of the PWARX system (6).

So far we have obtained an estimate of each affine submodel of the PWARX representation. The final step is to look for the shape of the polyhedral regions $\mathcal{X}_i$. Since the data have been classified, the problem of estimating the sets $\mathcal{X}_i$ amounts to a pattern recognition problem [5]. Note that there is a hyperplane that separates the set $\mathcal{X}_i$ from the set $\mathcal{X}_j$, $\forall j \neq i$ because all the sets $\mathcal{X}_i$ are polyhedral and convex. We can estimate such hyperplanes by applying a linear pattern recognition algorithm that separates the x-points in $\mathcal{F}_i$ from the x-points in $\mathcal{F}_j$, $\forall j \neq i$. The estimated hyperplane separating $\mathcal{F}_i$ from $\mathcal{F}_j$ is denoted with $M_{ij}x = m_{ij}$ where M_{ij} and m_{ij} are matrices of suitable dimensions. Moreover, we stipulate that the points in $\mathcal{X}_i$ belong to the half-space $M_{ij}x \leq m_{ij}$.

Due to errors in clustering, it may be not possible to find all the separating hyperplanes. Therefore, the classification algorithm should look for the hyperplanes that minimize the number of misclassified samples. For the classification we used linear Support Vector Machines [15] because they are appealing from a computational point of view (they can be solved through Linear or Quadratic Programming) and they isolate, as a byproduct, the misclassified samples.

In order to obtain a description of the set $\mathcal{X}_i$ in terms of linear inequalities, it is then enough to consider the bounded polyhedron

$$[M_{i1}' \ \dots \ M_{is}' \ M']'x \leq [m_{i1}' \ \dots \ m_{is}' \ m']'. \quad (15)$$

where $Mx \leq m$ are the linear inequalities describing $\mathcal{X}$. In (15) there may be redundant constraints that can be eliminated by using standard linear programming techniques. For the toy example, the following estimated sets were obtained

$$\begin{aligned}\mathcal{X}_1 &= [-4, -0.143], & (16)\\ \mathcal{X}_2 &= [-0.143, 1.873], & (17)\\ \mathcal{X}_3 &= [1.873, 4]. & (18)\end{aligned}$$

that provide a good approximation of $\bar{\mathcal{X}}_1$, $\bar{\mathcal{X}}_2$ and $\bar{\mathcal{X}}_3$ in Example 1

4 Discussion and generalizations

The proposed algorithm is made of six steps: build small clusters of the original data; identify a parameter vector based on each cluster; partition the parameter vectors in s clusters; classify the original data; estimate the s submodels; estimate the partition $\mathcal{X}_i$, $i = 1, \ldots, s$ by using a linear classification algorithm.

For the clusterization in the dual space, we propose a modified K-means algorithm. However K-means algorithm is not guaranteed to converge to the optimal clusters and may be trapped in local minima. Therefore it would be safer to repeat the clusterization in the dual space many times by randomly initializing the centers. A different way to cope with this problem is to resort to soft competitive clusterization algorithms that are less sensitive to initializations [8]. In order to improve the performance of the modified K-means, it is also possible to exploit the probabilistic information on the regressor vectors in order to detect the outliers in the dual space, eliminate them from the set of the θ-points and eliminate the corresponding data points from the clusters $\mathcal{F}_i$. In fact, the clusterization of the outliers may have a high degree of uncertainty and classification errors may spoil the accuracy of the final linear identification procedure. Such a procedure is currently under investigation.

The proposed algorithm gives good results under the implicit assumption that the sampling in the x-space is "fair", i.e. that the input is persistently exciting and that the x points are not all concentrated around the boundary of the sets $\mathcal{X}_i$. In fact, in the latter case it may happen that all the clusters $\mathcal{C}_j$ become mixed even if a large number of samples belonging to each submodel has been collected. A thorough characterization of such conditions will be the subject of further research.

In the previous Sections we exploited Assumption 2. However, if the number s of submodels is unknown it can be estimated from the dataset. This can be done by replacing the modified K-means algorithm with a clustering algorithm where the number of clusters is not fixed a priori such as the Growing Neural Gas [8] or the MDL-based algorithm proposed in [4]. In such procedures the number of clusters is automatically detected. It is apparent that once that the regressor vectors have been classified, the remaining part of our procedure can be applied without modifications.

If the order n_a and n_b are unknown, we expect that their under/over estimation can be detected from a picture of the coefficients in the dual space (i.e. the clusters do not have a clear boundary). Under/over parametrization can be also detected by comparing the magnitude of the final parameter vectors with their standard deviation.

5 Conclusions

In this paper we propose an algorithm for the identification of hybrid systems in the piecewise affine form. Our procedure hinges on the combined use of clustering, linear identification and classification techniques. This allows us to identify both the affine submodels and the polyhedral partition of the domain on which each submodel is valid. Future investigations will focus on the performance analysis of the method. For instance it would be desirable to have bounds on the errors affecting the algorithm both in identifying the submodels and in detecting the regions. Also the order selection issue needs to be investigated further.

Acknowledgments

The research of G. Ferrari-Trecate has been supported by the Swiss National Science Foundation.

References

[1] A. Bemporad, G. Ferrari-Trecate, and M. Morari. Observability and controllability of piecewise affine and hybrid systems. *IEEE Trans. on Automatic Control*, 45(10):1864–1876, 2000.

[2] A. Bemporad and M. Morari. Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35(3):407–428, March 1999.

[3] A. Bemporad and M. Morari. Verification of hybrid systems via mathematical programming. In F.W. Vaandrager and J.H. van Schuppen, editors, *Hybrid Systems: Computation and Control*, volume 1569 of *Lecture Notes in Computer Science*, pages 31–45. Springer Verlag, 1999.

[4] H. Bischof, A. Leonardis, and A. Selb. MDL principle for robust vector quantisation. *Pattern Analysis & Applications*, 2:59–72, 1995.

[5] C.M. Bishop. *Neural networks for pattern recognition.* Clarenfon press, Oxford, 1995.

[6] L. Breiman. Hinging hyperplanes for regression, classification, and function approximation. *IEEE Trans. Inform. Theory*, 39(3):999–1013, 1993.

[7] G. Ferrari-Trecate, D. Mignone, and M. Morari. Moving Horizon Estimation for Piecewise Affine Systems. *Proceedings of the American Control Conference*, 2000.

[8] B. Fritzke. Some competitive learning methods. Technical Report NCRG/98/015, Institute for Neural Computation. Ruhr-Universit at Bochum, 1997.

[9] R.E. Groff, D.E. Koditschek, and P.P. Khargonekar. Piecewise linear homeomorphisms: The scalar case. *Proc. Int. Joint Conf. on Neural Networks*, 2000.

[10] S. Haykin. *Neural networks - a comprehensive foundation.* Macmillan, Englewood Cliffs, 1994.

[11] L. Ljung. *System Identification - Theory For the User.* PTR Prentice Hall, Upper Saddle River, N.J., 1999. 2nd ed.

[12] M. Muselli and D. Liberati. Training digital circuits with Hamming clustering. *IEEE Tr. Circuits and Systems - Part I*, 47(4):513–527, 2000.

[13] K. Nakayama, A. Hirano, and A. Kanbe. A structure trainable neural network with embedded gating units and its learning algorithm. *Proc. Int. Joint Conf. on Neural Networks*, 2000.

[14] T. Soderstrom and P. Stoica. *System Identification.* Hemel Hempstead:Prentice Hall inc., 1989.

[15] V. Vapnik. *Statistical Learning Theory.* John Wiley, NY, 1998.