Identification of Hybrid Systems via Mixed-Integer Programming

Alberto Bemporad

Automatic Control Laboratory
ETH - ETL I 26
8092 Zurich, Switzerland

Dip. Ingegneria dell'Informazione
Università di Siena
Via Roma 56, 53100 Siena, Italy

bemporad@aut.ee.ethz.ch
http://control.ethz.ch/~bemporad

Tech. Report AUT00-28

September 15, 2000

Abstract

This paper addresses the problem of identification of hybrid dynamical systems, by focusing the attention on piecewise affine (PWARX) and hinging hyperplanes (HHARX) autoregressive exogenous models. In particular, for the latter class we provide an algorithm based on mixed-integer linear or quadratic programming which, contrarily to former approaches for estimation of hinging hyperplane models, converges to a global optimum. We also discuss issues of state-space realization of HHARX models into several existing discrete-time hybrid state-space forms. The goal is to be able to use on models estimated from data the hybrid tools recently developed by several authors for stability analysis, control synthesis, verification/safety analysis, and fault detection.

Keywords Hybrid systems, identification, piecewise linear models, hinging hyperplanes, mixed-integer programming

1 Introduction

Hybrid models describe systems composed by both continuous and discrete components, the former typically associated with physical first principles, the latter with logic devices, such as switches, digital circuitry, software code, etc. A typical instance are real-time systems, where physical processes are controlled by embedded controllers. Their study has recently seen a rapid development, thanks to the interaction between the computer science and the control engineering communities.

Most of the literature of hybrid systems has dealt with the issues of modeling [1, 2, 3], and tools have been proposed for stability analysis [4, 5], control [3, 6, 7], verification [8, 9, 10, 11], and fault detection [12, 13]. Such tools, relay on a model of the hybrid system, which sometimes is either not available because some parameters are unknown, or is too complex to be handled. Getting a simple hybrid model from data or from a more complex model is an identification problem, which does not seem to have received enough attention in the hybrid systems community.

On the other hand, other communities dedicated extensive research to develop tools for identification of piecewise linear (or piecewise affine) models of nonlinear dynamical systems (see e.g. the survey [14]), such as neural networks with piecewise affine percerptrons [15], self-exciting threshold autoregressive (SETAR) models for time-series analysis [16], and hinging hyperplanes models [17, 18]. In parallel, in the circuit and systems community such a problem has been also addressed for the representation and the analysis of nonlinear electronic circuits, mostly based on Chua's canonical piecewise linear representation [19, 20, 21, 22, 23].

Because of the equivalence between hybrid and piecewise affine systems [24, 25], such tools for PWA identification can be used to obtain hybrid models, or at least for translating nonlinear/unknown parts of the system into a hybrid form, for which the tools mentioned above can be applied.

In this paper, we introduce the class of PWARX (Piecewise-affine AutoRegressive eXogenous) systems for system identification, and show that the the problem is simple if the *guardlines* (i.e., the polyhedral partition of the PWA mapping) are known (this is the case for instance of the canonical piecewise linear representation [22]), while it is much more complex if also the guardlines have to be determined. In order to attach the problem algorithmically, we focus our attention on HHARX (Hinging-Hyperplane AutoRegressive eXogenous) systems as hybrid linear discrete-time models [26], by showing how mixed-integer linear programming can be used to guarantee a *global* minimal identification error. At the best of our knowledge, the algorithms proposed in the literature for determining HH models from data [17, 18] only provide *local* minima.

2 PWARX Models

Autoregressive models rely on the assumption that $y(t)$ only depends on a certain number of past inputs and outputs $\varphi(t) = [y(t-1)\ \dots\ y(t-n_a)\ u(t-1)\ \dots\ u(t-n_b)]' \in \mathbb{R}^n$, $n \triangleq n_a + n_b$, where n_a, n_b are parameters decided by the user (criteria exist for choosing n_a, n_b, like Akaike's criterion, false nearest neighbors, etc.). The vector $\varphi(t)$ is denoted as regression vector, and for simplicity of notation, we have assumed that the system is SISO. In nonlinear system black-box identification, the aim is at determining a function $f(\varphi)$ such that $y(t) = f(\varphi(t))$ well fits a given estimation data set $\{u(t), y(t)\}_{t=0}^{T-1}$, e.g., $\sum_{t=0}^{T-1} \|y(t) - f(\varphi(t))\|$ is minimized (if f is partially known, for instance because of some physical insight, we refer to as *graybox* identification). In order to render the optimization problem finite dimensional (and therefore computationally tractable), we restrict our attention to a family of functions $\mathcal{G}$, parameterize such a family by a vector of real valued parameters θ, and look for the best θ^* whose corresponding function $g^* \in \mathcal{G}$ is the best approximation of f in $\mathcal{G}$ according to the selected fitting criterion.

In this paper we consider the family $\mathcal{G}$ of piecewise affine functions

$$g(\varphi) = d^j \varphi + c^j \quad \text{if} \quad H_j \varphi \leq D_j \tag{1}$$

where $d^j \in \mathbb{R}^n$, $c^j \in \mathbb{R}$, and the sets $\mathcal{C}_j \triangleq \{\varphi :\ H_j\varphi \leq D_j\}$, $j = 1, \dots, s$ are a polyhedral partition of the φ-space. We do not assume that g is necessarily continuous over the boundaries, also referred to as *guardlines* in the hybrid systems literature. Without the hypothesis of continuity over the boundaries, definition (1) is not well posed in general, as the function can be twice (or more times) defined over common boundaries of the sets $\mathcal{C}_j$. Although one can avoid this issue by dealing with open polyhedra and boundaries separately (as in [27]), or by replacing some of the "$\leq$" inequalities into "$<$" in the definition of the regions, this issue is not of practical interest for the numerical approach of this paper.

We define PWARX (Piecewise-affine AutoRegressive eXogenous) the model

$$y(t) = g(\varphi(t)) + e(t) \tag{2}$$

where g is a piecewise affine (not necessarily continuous) function of the form (1), and $e(t)$ is an additive term accounting for the fact that $y(t)$ is not exactly a function of past inputs and data. When g is uniquely defined over the whole φ-space, the trajectories of the PWARX system (2) are well defined for all t.

There are several reasons for focusing on PWARX systems for hybrid systems identification. From a theoretical point of view, piecewise affine systems are equivalent to linear hybrid systems, as shown by Sontag [24] and later in [25], where the authors also provide a constructive proof. From a computational point of view, the advantage in using PWA models is twofold. First, PWA mappings are suitable for efficient identification algorithms. Although this is certainly true also for sigmoidal neural networks and other nonlinear mappings, the second computational advantage comes on the use of the identified models for control and formal verification purposes: most of such hybrid tools are only available for PWA or linear hybrid models.

2.1 Identification of PWARX Models

Model (2) can be more explicitly rewritten in the form

$$y(t) = \sum_{k=0}^{n_a} a_k^j y(t-k) + \sum_{k=1}^{n_b} b_k^j u(t-k) + c^j + e(t) \quad \text{if} \quad H_j \varphi(t) \leq D_j \tag{3}$$

where $j = 1, \ldots, s$, and let $\theta_j = [a_1^j \ \ldots \ a_{n_a}^j \ b_1^j \ \ldots \ b_{n_b}^j \ c^j]' \in \mathbb{R}^{n+1}$.

The identification of model (3) can be carried out by solving the optimization problem

$$\min \frac{1}{2N} \sum_{k=1}^{N} \left(\sum_{j=1}^{s} \|y_k - \varphi_k' \theta_j\| \mathcal{J}_j(\varphi_k) \right) \tag{4a}$$

$$\text{subj. to } \mathcal{J}_j(\varphi_k) = \begin{cases} 1 & \text{if } H_j \varphi_k \leq D_j \\ 0 & \text{otherwise} \end{cases} \tag{4b}$$

$$+ \text{ linear bounds over the components of } \theta_j,\ H_j,\ D_j \tag{4c}$$

where θ_j, H_j, D_j $j = 1, \ldots, s$ are the unknowns, $y_k = y(k-1)$, $\varphi_k = \varphi(k-1)$ are the estimation data, and $N = T$ is the number of data. In (4), we will focus on the following two norms: $\|\cdot\|_2^2$ (squared Euclidean norm) and $\|\cdot\|_\infty$ (infinity norm over space), as they allow to express (4) as a (mixed-integer) quadratic and (mixed-integer) linear program, respectively. The infinite norm over space and time is not recommended, as the estimate would be highly sensitive to possible outliers in the estimation data. Note that every norm in (4a) provides a consistent minimum with respect to a probability distribution associated to the noise $e(t)$, and in particular 1-norm is optimal with respect to exponentially distributed noise ($\rho(x) = \rho e^{-x}$), 2-norm is optimal with respect to Gaussian noise. We distinguish below among two main cases:

- Case A: H_j, D_j are known, θ_j have to been estimated;
- Case B: H_j and/or D_j are unknown, θ_j is unknown.

2.2 Known Guardlines

In (4), only θ_j have to be determined. This is for instance the case of canonical piecewise linear functions, where the partition of the φ-space is assigned, and only the linear gains θ_j in each polyhedral cell have to be determined.

Given H_j, D_j, $j = 1, \dots, s$, the identification algorithm is based on the following optimization problem

$$\Theta^* \triangleq \arg\min \frac{1}{2N} \sum_{k=1}^{N} \left(\sum_{j=1}^{s} \|y_k - \varphi_k' \theta_j\| \mathcal{J}_j(\varphi_k) \right) \tag{5a}$$

$$\text{subj. to } \theta_j^- \leq \theta_j \leq \theta_j^+ \tag{5b}$$

where $\Theta \triangleq [\theta_1 \ \dots \ \theta_s] \in \mathbb{R}^{(n+1)\times s}$ are the unknowns, and $\mathcal{J}_j(\varphi_k)$ are numbers computed directly from the estimation data.

Let $I^j = \{k \leq N : \ H_j \varphi_k \leq D_j\}$ the set of time instants the regression vector φ_k lies in region $\#j$ (i.e., the indicator function $\mathcal{J}_j(\varphi_k) = 1$ iff $k \in I^j$). Then the cost function in (5a) can be rewritten as

$$\frac{1}{2N}\Theta' \begin{bmatrix} \sum_{k\in I^1} \varphi_k \varphi_k' & & \\ & \ddots & \\ & & \sum_{k\in I^q} \varphi_k \varphi_k' \end{bmatrix} \Theta +$$

$$-\frac{1}{N} \left[\ \sum_{k\in I^1} y_k' \varphi_k \ \ \cdots \ \ \sum_{k\in I^q} y_k' \varphi_k \ \right] \Theta + \frac{1}{2N} \sum_{k=1}^{N} y_k' y_k$$

The problem has clearly a block diagonal structure (both the cost function and the constraints over Θ), and hence the problem can be separated by identifying each subsystem j independently

$$\theta_j^* = \arg\min \sum_{k\in I^j} \|y_k - \varphi_k' \theta_j\| \tag{6a}$$

$$\text{subj. to } \ \theta_j^- \leq \theta_j \leq \theta_j^+ \tag{6b}$$

which can be solved by a standard QP solver (Euclidean norms) or LP solver (infinity norm). In particular, if the box constraints (6b) are not included (for instance because there is no a priori information about Θ), and the squared Euclidean norm is used, (6) can be solved by the following recursive least squares (RLS) form [28]

$$\theta_j^*(t+1) = \theta_j^*(t) + \frac{P^j(t)\varphi(t+1)}{1+\varphi'(t+1)P^j(t)\varphi(t+1)} [y(t+1) - \varphi'(t+1)\theta_j(t)]\delta^j(t) \tag{7a}$$

$$P^j(t+1) = P^j(t) - \frac{P^j(t)\varphi(t+1)\varphi'(t+1)P^j(t)}{1+\varphi'(t+1)P^j(t)\varphi(t+1)}\delta^j(t) \tag{7b}$$

$$\delta^j(t) = 1 \quad \text{if} \quad t \in I^j, \ 0 \quad \text{otherwise} \tag{7c}$$

Note that the estimate θ_j^* is only updated when $t \in I^j$ (i.e., the regression vector $\varphi(t)$ enters region j), otherwise it remains constant.

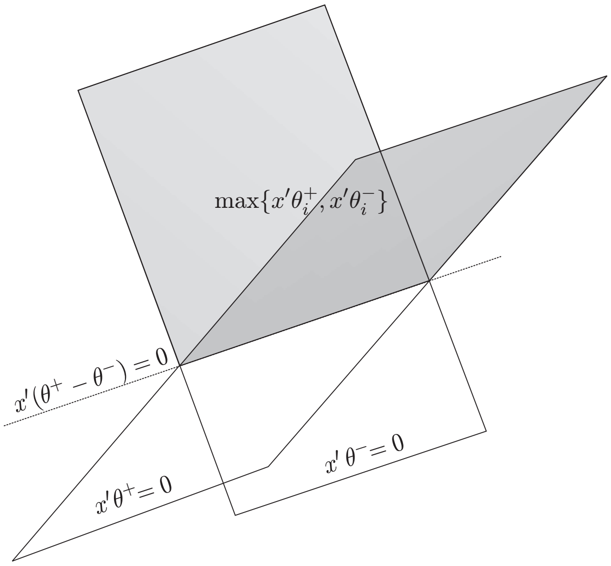


Figure 1: Hinging hyperplanes and hinge function

2.3 Unknown Guardlines

The optimization problem (4) can be recast as a mixed-integer linear program (MILP) (infinity norm) or mixed-integer quadratic program (MIQP) (Euclidean norm). In the following section, we focus on a subset of PWA functions, the Hinging Hyperplanes, and detail the mixed-integer program associated to the identification problem. In general, the complexity of the mixed-integer program needed to solve (4) is related to the number of samples T and regions s, and the number of parameters H_j, D_j, θ_j that are unknown. Note that in general, the guardlines $H_j^i\varphi \leq D_j^i$, (where, given a matrix M, M^i denotes the i-th row of M) cannot be determined exactly from the given estimation data set, as the pairs y_k, φ_k are a discrete set of points which, in principle, can be divided by a continuum of possible guardlines.

3 Hinging Hyperplane Models

Hinging hyperplanes (HH) where introduced by Breiman [17] in the context of regression, classification, and function approximation problems. Typically, in nonlinear system identification the family $\mathcal{G}$ of models where we look for an optimal approximation is generated by parameterizing a truncated set of basis functions $g_i(\varphi)$, $i = 1, \ldots, M$, namely

$$g(\varphi) = \sum_{i=1}^{M} \alpha_i g_i(\varphi)$$

where g_i have universal approximation properties ($g(\varphi) \to f(\varphi)$ as $M \to \infty$). In general, $g_i(\varphi)$ are obtained by parameterizing a single "mother" function $\kappa(z)$ [14]

$$g_i(\varphi) = \kappa(\beta_i(\varphi - \gamma_i))$$

and once M has been also fixed, g^* is selected by solving a finite optimization problem over α_i, β_i, γ_i, $i = 1, \ldots, M$.

One hidden-layer feedforward sigmoidal neural networks corresponds to choosing as mother function the sigmoid $\kappa(z) = \frac{1}{1+e^{-z}}$. HH models are instead based the hinge function

$$g_i(x) = \pm \max\{\beta_i^+\varphi + \gamma_i^+, \beta_i^-\varphi + \gamma_i^-\}$$

corresponding to an "open book" in the (y, φ)-space, as depicted in Fig. 1. By letting $x = [\varphi'\ 1]'$, the HH model can be expressed as

$$g(x) = \sum_{i=1}^{M} \pm \max\{x'\theta_i^+, x'\theta_i^-\} \tag{8}$$

where each pair of hyperplanes "hinges" upon the set $\{x :\ x'(\theta^+ - \theta^-) = 0\}$, and has either the plus or the minus sign. As shown in [18], the original HH model (8) proposed by Breiman [17] is over-parameterized, and can be reduced to

$$g(x, \Theta) = x'\theta_0 + \sum_{i=1}^{M} \pm \max\{x'\theta_i, 0\} \tag{9}$$

where

$$\theta_0 \triangleq \sum_{i=1}^{M} \pm\theta_i^-,\ \ \theta_i \triangleq \theta_i^+ - \theta_i^-$$

and $\Theta \triangleq [\theta_0\ \ldots\ \theta^M] \in \mathbb{R}^{(n+1)\times M+1}$. Note that (9) corresponds to interpreting (8) as the sum of a linear function and basis functions generated by the mother $\kappa(z) = \max(z, 0)$.

Although the form (9) proposed in [18] has $(M-1)(n+1)$ less parameters than the original form proposed by Breiman [17], it has still redundancies. In fact, we can rewrite (9) as

$$\begin{aligned} g(x, \Theta) &= x'\theta_0 + \sum_{i=1}^{M} \pm x'\theta_i \mp x'\theta_i \pm \max\{x'\theta_i, 0\} \\ &= x'(\theta_0 + \sum_{i=1}^{M} \pm\theta_i) + \sum_{i=1}^{M} \pm \max\{-x'\theta_i, 0\} \end{aligned}$$

where the last equality follows from the relations $-z + \max\{z, 0\} = \max\{-z, 0\}$, and $z - \max\{z, 0\} = -\max\{-z, 0\}$, $\forall z \in \mathbb{R}$. Therefore, the sets of parameters $\Theta = [\theta_0,\ \theta_1,\ \ldots,\ \theta^M]$ and $\bar{\Theta} = [\theta_0 + \sum_{i=1}^{M} \pm\theta_i,\ \mp\theta_1,\ \ldots,\ \mp\theta_M]$ are equivalent, as one should expect by observing that in (8) the max function is commutative (i.e., the roles of θ^- and θ^+ can be switched). As each vector θ_i can be mirrored into $-\theta_i$, $i = 1, \ldots, M$, this redundancy leads to multiple solutions in the identification problem. In order to avoid this, we will pose the extra condition

$$w'\theta_i \geq 0,\ i = 1, \ldots, M \tag{10}$$

where w is any nonzero vector of $\mathbb{R}^n$, e.g., $w = \underline{1} \triangleq [1\ 1\ \ldots\ 1]'$ (or any random vector), although clearly the redundancy problem is not completely solved, as vectors θ on the hyperplane $w'\theta = 0$ can be still mirrored.

3.1 HHARX Models

By setting $g(\varphi)$ in (2) as in (9), we obtain the following HHARX (Hinging-Hyperplane AutoRegressive eXogenous) model

$$y(t) = x'(t)\theta_0 + \sum_{i=1}^{M} \pm \max\{x'(t)\theta_i, 0\} \tag{11}$$

$x(t) = [y(t-1)\ \ldots\ y(t-n_a)\ u(t-1)\ \ldots\ u(t-n_b)\ 1]' \in \mathbb{R}^{n+1}$, where "$\pm$" means that each max function is taken either with the plus or the minus sign. Note that, as a special case, since the max function is convex, the sum of only $+\max$ terms leads to a convex mapping between $y(t)$ and the regressors. The sign to be associated with each max function is a choice of the user when the model is identified from data, as described in the next section.

4 Identification Algorithms for HH Models

The first algorithm for estimating HH models was proposed by Breiman [17]. Later, in [18] the authors show that the original algorithm is a special case of Newton's method, and provide a modification which guarantees convergence to a *local* minimum. Other algorithms have been proposed based on tree hinging hyperplane models [29]. In this paper, we propose an alternative approach based on mixed-integer programming, which provides a *global* minimum, at the price of an increased computational effort with respect to the algorithms in [17, 18].

Consider the problem of estimating a HH function of the form (9) from the estimation data set $\{y_k, x_k\}_{k=1}^N$. We choose the optimal parameters Θ^* by solving the mathematical program

$$\Theta^* \triangleq \arg\min V(\Theta) \triangleq \sum_{k=1}^{N} |y_k - g(x_k, \Theta)| \tag{12a}$$

$$\text{subj. to} \quad \begin{cases} \theta^{j-} \leq \theta_j \leq \theta^{j+} \\ \underline{1}'\theta_i \geq 0, \ \ i = 1, \ldots, M \end{cases} \tag{12b}$$

where the inequalities in (12b) are componentwise and optionally included for reflecting partial a priori information about the range of the parameters θ_i. As mentioned above, the reason for using the absolute value in (12) is mainly numerical, because as we will show below the problem can be recast as a *mixed-integer linear program* (MILP) (see also [30]). Another reasonable possibility is to use the squared Euclidean norm $|y_k - g(x_k, \Theta|^2$, as the problem can be recast as a *mixed integer quadratic program* (MIQP). The problem can be also recast as an MILP by using infinity norm over time (i.e. $\max_{k=1,\ldots,N}$ instead of $\sum_{k=1}^N$, although this would be highly sensitive to possible outliers in the estimation data, as mentioned above.

4.1 Optimization Problem

4.1.1 MILP Formulation

The optimization problem (12) can be recast as an MILP by introducing additional integer 0-1 and continuous variables. To this end, we associate one 0-1 variable δ_{ik} to the sign of $x_k'\theta_i$

$$[\delta_{ik} = 0] \leftrightarrow [x_k'\theta_i \leq 0], \ \ i = 1, \ldots, M, \ \ k = 1, \ldots, N \tag{13}$$

and introduce new continuous variables z_{ik}

$$z_{ik} = \max\{x_k'\theta_i, 0\} = x_k'\theta_i\delta_{ik} \tag{14}$$

The relations (13) and (14) can be transformed into mixed-integer linear inequalities, by using standard techniques [31, 32, 33, 7]. By assuming that the bounds over θ_i are all finite, Eq. (13) is equivalent to

$$x_k'\theta_i \leq M_{ik}^\theta \delta_{ik} \tag{15a}$$

$$x_k'\theta_i \geq \varepsilon + (m_{ik}^\theta - \varepsilon)(1 - \delta_{ik}) \tag{15b}$$

where ε is a small positive scalar (e.g., the machine precision),

$$\begin{aligned} M_{ik}^\theta &\triangleq \|x_k\|_\infty M_i^\theta \left(\geq \max_{\theta_i^- \leq \theta_i \leq \theta_i^+} x_k'\theta_i \right) \\ m_{ik}^\theta &\triangleq -M_{ik} \left(\leq \min_{\theta_i^- \leq \theta_i \leq \theta_i^+} x_k'\theta_i \right) \end{aligned}$$

and

$$M_i^\theta \triangleq \max_{\theta_i^- \le \theta_i \le \theta_i^+} x_k'\theta_i$$

Similarly, (14) is equivalently rewritten as

$$-M_{ik}^\theta \delta_{ik} + z_{ik} \le 0 \tag{16a}$$
$$m_{ik}^\theta \delta_{ik} - z_{ik} \le 0 \tag{16b}$$
$$m_{ik}^\theta (1-\delta_{ik}) + z_{ik} \le x_k'\theta_i \tag{16c}$$
$$-M_{ik}^\theta (1-\delta_{ik}) - z_{ik} \le -x_k'\theta_i \tag{16d}$$

Finally, by introducing auxiliary slack variables $\epsilon_k \ge |y_k - g(x_k, \Theta)|$, $k = 1, \ldots, N$, (12) we can prove the following

Proposition 1 *The optimum of problem (12) is equivalent to the optimum of the following MILP*

$$\min_{\substack{\epsilon_1,\ldots,\epsilon_N \\ \theta_0,\ldots,\theta_M \\ \delta_{11},\ldots,\delta_{MN} \\ z_{11},\ldots,\delta_{MN}}} \sum_{k=1}^N \epsilon_k \tag{17a}$$

$$\textit{subj. to } \epsilon_k \ge y_k - \left(x_k'\theta_0 + \sum_{i=1}^N \pm z_{ik} \right) \tag{17b}$$

$$\epsilon_k \ge x_k'\theta_0 + \sum_{i=1}^N \pm z_{ik} - y_k \tag{17c}$$

$$(15),\ (16),\ (10) \tag{17d}$$

Proof. Clearly, given the data $\{y_k, x_k\}$ and for any Θ, there exist only one set of variables z_{ik}, δ_{ik} satisfying constraints (15), (16), and for any $\{\epsilon_k\}$ compatible with the constraints (17b)–(17c) it results $\sum_{i=1}^N \epsilon_k \ge V(\Theta) \ge V(\Theta^*)$, where Θ^* is the solution to (12). In particular, for the optimizer $\epsilon_i^\#$, $\Theta^\#$ of (17) it results $\sum_{i=1}^N \epsilon_k^\# \ge V(\Theta^\#) \ge V(\Theta^*)$. On the other hand, the values $\epsilon_k^* \triangleq |y_k - g(x_k, \Theta^*)|$, $\delta_{ik}^* \triangleq 0$ if $x_k'\theta \le 0$, or 1 otherwise, and $z_{ik}^* \triangleq \delta_{ik} x_k'\theta_i$, $i = 1, \ldots, M$, $k = 1, \ldots, N$, are admissible for problem (17), which proves that the minimum of (17) $V(\Theta^\#) = \sum_{i=1}^N \epsilon_k^\# \le \sum_{i=1}^N \epsilon_k^* \le V(\Theta^*)$. Therefore, $V(\theta^\#) = V(\theta^*)$. □

Example 1

Consider the following PWARX system

$$y(t) = \begin{cases} 0.8y(t-1) + 0.4u(t-1) - 0.1 & \text{if} \quad -0.3y(t-1) + 0.6u(t-1) \le -0.3 \\ 0.5y(t-1) + u(t-1) + 0.2 & \text{if} \quad -0.3y(t-1) + 0.6u(t-1) > -0.3 \end{cases} \tag{18}$$

or, equivalently, the HHARX model $y(t) = 0.8y(t-1) + 0.4u(t-1) - 0.1 + \max\{-0.3y(t-1) + 0.6u(t-1) + 0.3, 0\}$. The model is identified on the data reported in Fig. 3(a), by solving an MILP with 66 variables (of which 20 integers) and 168 constraints. The problem is solved by using Cplex 6.5 [34] (1014 LP solved in 0.68 s on a Sun Ultra 10 running Matlab 5.3), and, for

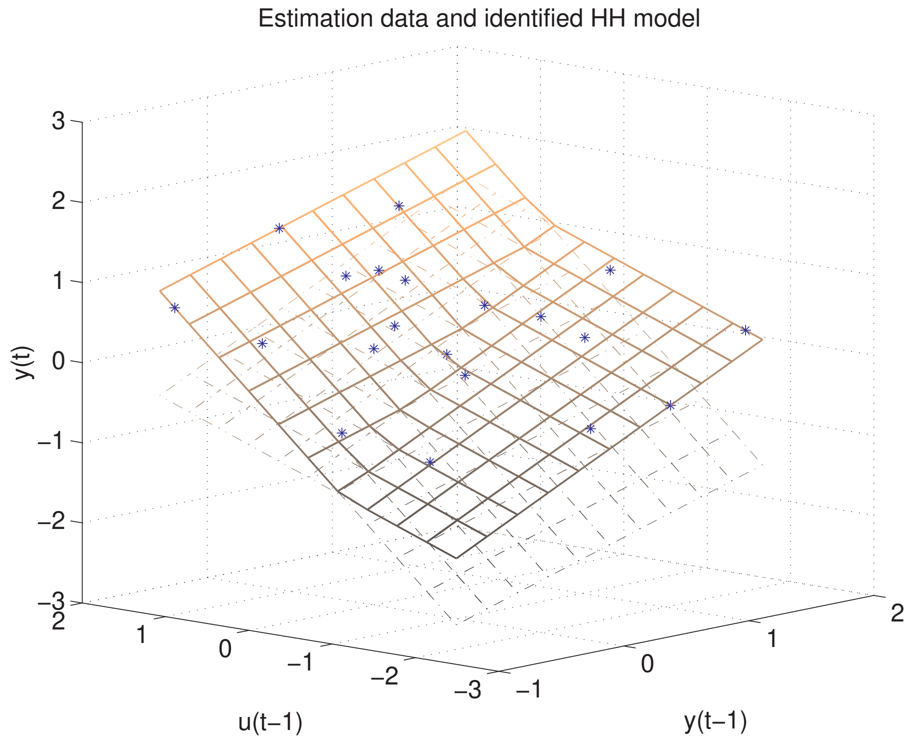


Figure 2: Identification of model (18) – noiseless case. Identified HH model

comparison, is solved again using BARON [35] (73 LP solved in 3.00 s, same machine), which results in a zero output prediction error (Fig. 3(b)). The fitted HH model is reported in Fig. 2. By adding a random Gaussian noise with zero mean and variance 0.1 on the measured output $y(t)$, the following model

$$y(t) = \begin{cases} 0.8315y(t-1) + 0.3431u(t-1) - 0.2014 & \text{if} \quad -0.3391y(t-1) + 0.6205u(t-1) \le -0.3977 \\ 0.4924y(t-1) + 0.9636u(t-1) + 0.1963 & \text{if} \quad -0.3391y(t-1) + 0.6205u(t-1) > -0.3977 \end{cases} \tag{19}$$

is identified in 1.39 s (3873 LP solved) using Cplex (7.86 s, 284 LP using BARON) on the estimation set reported in Fig. 4(a), and produces the validation data reported in Fig. 4(b). For comparison, we identified the linear ARX model

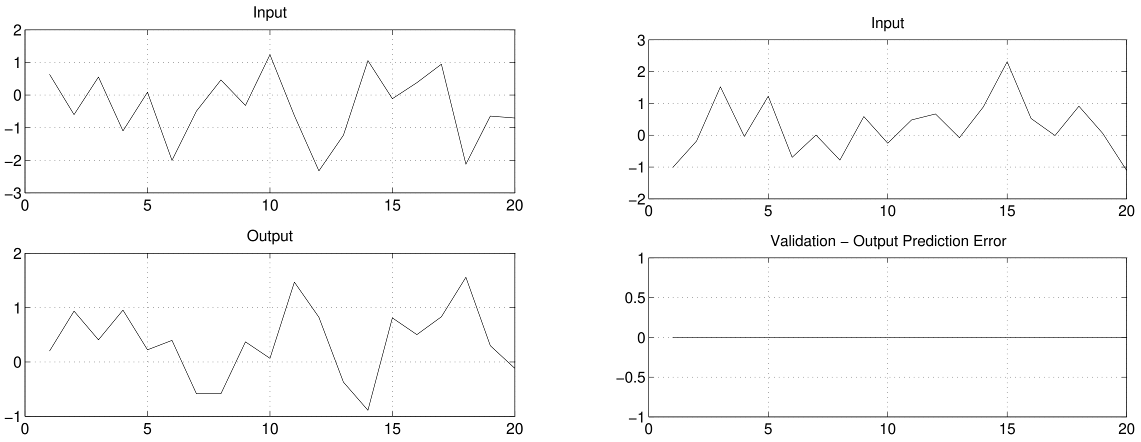
$$y(t) = 0.8250y(t-1) + 0.7217u(t) \tag{20}$$

on the same estimation data, obtaining the validation data reported in Fig. 5 (higher order ARX models did not produce significant improvements). Note that the output error on the validation data is computed by feeding the ARX model with the outputs obtained by the PWARX model (18). Clearly, the error generated by driving the ARX model in open-loop with the validation input $u(t)$ only is much larger, and would not make (20) suitable for instance for formal verification tools, where a good performance of open-loop prediction is a critical requirement.

4.1.2 MIQP Formulation

When the squared 2-norm is used in the objective function,

$$V(\Theta) \triangleq \sum_{k=1}^{N} (y_k - g(x_k, \Theta))^2 \tag{21}$$



(a) Estimation data

(b) Validation data

Figure 3: Identification of model (18) – noiseless case

the optimization problem can be recast as the MIQP

$$\min_{\substack{\theta_0,\ldots,\theta_M\\ \delta_{11},\ldots,\delta_{MN}\\ z_{11},\ldots,\delta_{MN}}} \sum_{k=1}^{N}\left(y_k-(x_k'\theta_0+\sum_{i=1}^{M}\pm z_{ik})\right)^2 \tag{22a}$$

$$\text{subj. to } (15),\ (16),\ (10) \tag{22b}$$

Note that the problem is not strictly positive definite, for instance the cost function does not depend on θ_i, δ_{ik} (which only appear in the constraints). For numerical reasons, the Hessian associated to the MIQP (22) is augmented with the term σI, where σ is a small fixed number.

4.1.3 Complexity

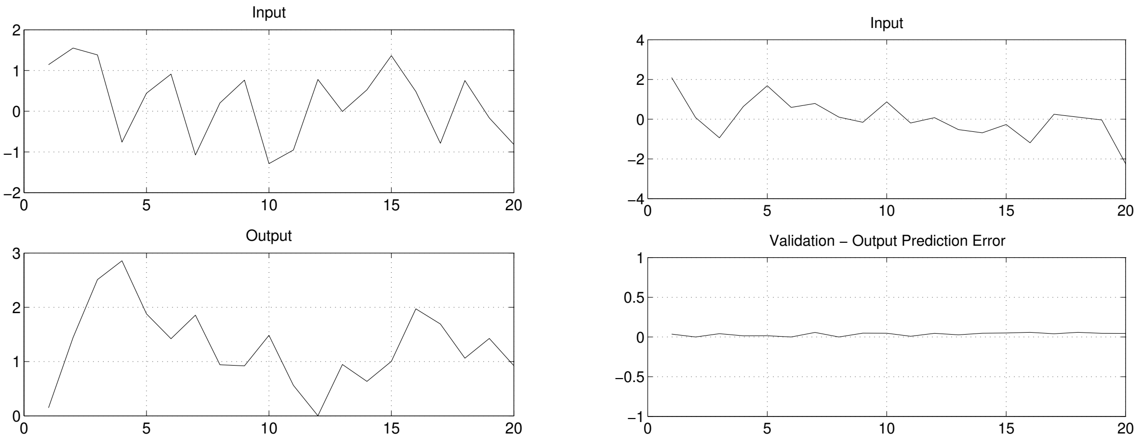
Despite the good solvers available, the complexity of the MILP or MIQP problems is well known to be $\mathcal{NP}$-hard, and in particular it is exponential in the number MN of binary variables. Therefore, the mixed-integer approach is computationally affordable only for model with a few data, or if data are clustered together (e.g., 100 data are averaged into 10 data). Otherwise, local search methods [17, 18] are preferable, although they do not guarantee the best fit (i.e., a global minimum to the optimization problem).

Example 2

Consider again the PWARX system (18). In Figure 6 we compare the performance in terms of LP/QPs and total computation time of the linear criterion (17) versus the quadratic criterion (22). The reported numbers are computed on a Sun Ultra 60 (2×360 MHz) running Matlab 5.3 and the solver BARON [35], by averaging the result of ten estimation data sets generated by feeding random Gaussian inputs $u(t)$ and zero output noise to system (18).

4.2 Discontinuous HHARX Models

In HHARX models, the output $y(t)$ is a continuous function of the regressor $\varphi(t)$. On the other hand, hybrid systems often consist of piecewise affine discontinuous mappings. In order to tackle



(a) Estimation data (b) Validation data

Figure 4: Identification of model (18) – noisy case, $y(t)$ is perturbed by a random Gaussian noise with zero mean and variance 0.1

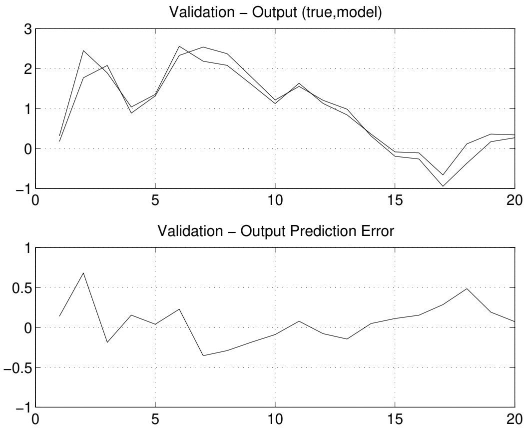


Figure 5: Identification of a linear ARX model – same estimation and validation data as in Fig. 4

discontinuities, we can modify the HH model (9) in the form

$$g(x, \Theta) = x'\theta_0 + \sum_{i=1}^{M} \pm(x'\theta_i + a_i)\delta_i(x) \tag{23a}$$

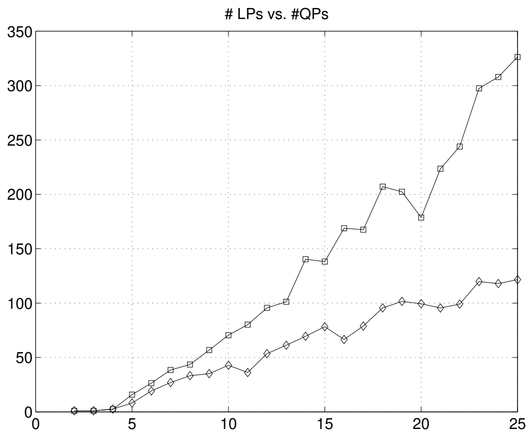
$$[\delta_i(x) = 0] \leftrightarrow [x'\theta_i \leq 0],\ i = 1, \ldots, M,\ k = 1, \ldots, N \tag{23b}$$

where a_i, $i = 1, \ldots, M$ are additional free parameters, $a_i^- \leq a_i \leq a_i^+$, or, more in general, in the form

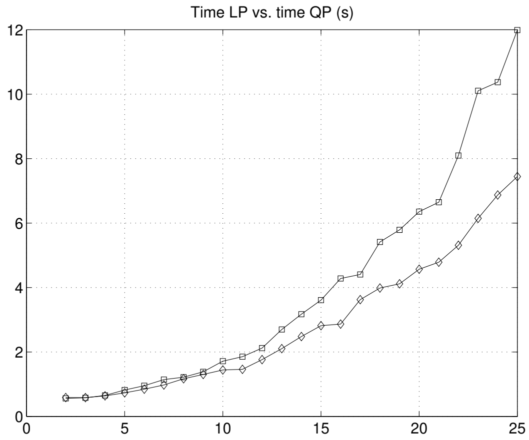
$$g(x, \Theta) = x'\theta_0 + \sum_{i=1}^{M} (x'\theta_i)\delta_i(x) \tag{24a}$$

$$[\delta_i(x) = 0] \leftrightarrow [x'\mu_i \leq 0],\ i = 1, \ldots, M,\ k = 1, \ldots, N \tag{24b}$$

where μ_i, $i = 1, \ldots, M$ are additional free vectors of parameters, $\mu_i^- \leq \mu_i \leq \mu_i^+$, $\underline{1}'\mu_i \geq 0$. Similarly to (17), by introducing new continuous variables z_{ik} both the identification prob-



(a) Average number of LPs and QPs



(b) Average computation time (Sun Ultra 60 (2×360 MHz) running Matlab 5.3 and the solver BARON)

Figure 6: Identification of model (18) – MILP vs. MIQP (results are averaged on ten estimation data sets generated by random Gaussian inputs $u(t)$ and zero output noise

lems (23) and (24) can be again recast as an MILP. With respect to (17), the MILP has μ_i or a_i, $i = 1, \dots, M$ as additional optimization variables. Note that the problem in general does not have a unique solution. For instance, once θ_i, a_i, z_{ik}, δ_{ik}, ϵ_k have been fixed, there exist in general infinitely many vectors μ_i satisfying the constraints (24b).

4.3 Robust HHARX Models

In formal verification methods, model uncertainty needs to be handled in order to provide a safety guarantees. Typically, the model is associated with a bounded uncertainty. In the context of symbolic solvers for timed automata, differential inclusions $a \leq \dot{x} \leq b$ are handled, for instance by the solver Hytech [36]. For piecewise affine and MLD systems, the verification algorithm proposed in [11] handles model uncertainty as additive input disturbances entering a nominal model.

In the present context of HHARX models, we wish to find an uncertainty description of the form

$$g(x(t), \Theta^-) \leq y(t) \leq g(x(t), \Theta^+), \ \forall t \geq 0 \tag{25}$$

for an inclusion-type of description, or the form

$$y(t) = g(x(t), \Theta^*) + n(t), \ n^- \leq n(t) \leq n^+ \tag{26}$$

for an additive-disturbance-type of description. Clearly, since the model is identified from a finite estimation data set, fulfillment of (25) or (26) for all t and for all initial conditions cannot be guaranteed, unless additional hypotheses on the model which generates the data are assumed.

Nevertheless, a pair of extreme models Θ^-, Θ^+ can be obtained by solving (17) or (22) with the additional linear constraints

$$y_k \geq g(x_k, \Theta), \ \forall k = 1, \dots, N \tag{27}$$

for estimating Θ^-, and

$$y_k \geq g(x_k, \Theta), \ \forall k = 1, \dots, N \tag{28}$$

for estimating Θ^+, respectively. An additive-disturbance description can be instead computed in two alternative ways:

1. First, identify a model Θ^* by solving (17) or (22) and then compute

$$\begin{aligned} n^+ &\triangleq \max\nolimits_{k=1,\ldots,N} y_k - g(x_k, \Theta^*) \\ n^- &\triangleq \min\nolimits_{k=1,\ldots,N} y_k - g(x_k, \Theta^*) \end{aligned} \tag{29}$$

2. Modify the MILP (17) by setting replacing ϵ_k with one variable ϵ only, and minimize ϵ. The corresponding optimum ϵ^* provides a nominal model such that the bound on the norm of the additive disturbance $n(t)$ is minimized.

5 State-Space Realizations of HHARX Models

In the linear ARX case, a minimal state-space realization containing n_a states (or less, if the identified polynomials $A(z)$, $B(z)$ are not coprime) can be obtained via canonical controllability or observability realizations, where the mapping from past y and u to the state ξ depends on the coefficients a_i, b_i of the polynomials. For HHARX models, in general, a state-space realization of the form

$$\begin{aligned} \xi(t+1) &= f(\xi(t), u(t)) \\ y(t) &= g(\xi(t), u(t)) \end{aligned} \tag{30}$$

requires n_a+n_b states. Below we describe how to obtain a state-space realization (30) of HHARX models, by using different state-space discrete-time paradigms introduced recently in the hybrid systems literature.

5.1 MLD Realization

Mixed Logical Dynamical (MLD) systems [7] are a discrete-time hybrid formalism which allows specifying the evolution of continuous variables through linear dynamic equations, of discrete variables through propositional logic statements and automata, and the mutual interaction between the two. The key idea of the approach consists of embedding the logic part in the state equations by transforming Boolean variables into 0-1 integers, and by expressing the relations as mixed-integer linear inequalities [31, 32, 33, 7]. MLD systems are therefore capable of modeling a broad class of systems, and examples of real-world applications that can be naturally modeled within the MLD framework are reported in [7, 37, 38].

The MLD model has the form

$$\xi(t+1) = \Phi\xi(t) + \mathcal{G}_1 u(t) + \mathcal{G}_2\delta(t) + \mathcal{G}_3 z(t) \tag{31a}$$

$$y(t) = \mathcal{H}\xi(t) + \mathcal{D}_1 u(t) + \mathcal{D}_2\delta(t) + \mathcal{D}_3 z(t) \tag{31b}$$

$$\mathcal{E}_2\delta(t) + \mathcal{E}_3 z(t) \leq \mathcal{E}_1 u(t) + \mathcal{E}_4\xi(t) + \mathcal{E}_5 \tag{31c}$$

where $\xi \in \mathbb{R}^{n_c} \times \{0,1\}^{n_\ell}$ is a vector of continuous and binary states, $u \in \mathbb{R}^{m_c} \times \{0,1\}^{m_\ell}$ are the inputs, $y \in \mathbb{R}^{p_c} \times \{0,1\}^{p_\ell}$ the outputs, $\delta \in \{0,1\}^{r_\ell}$, $z \in \mathbb{R}^{r_c}$ represent auxiliary binary and continuous variables respectively, which are introduced when transforming logic relations into mixed-integer linear inequalities, and Φ, $\mathcal{G}_1$, $\mathcal{G}_2$, $\mathcal{G}_3$, $\mathcal{H}$, $\mathcal{E}_1$,…,$\mathcal{E}_5$ are matrices of suitable dimensions.

Proposition 2 *HHARX models (11) admit an MLD state-space realization with n_a+n_b states.*

Proof. The MLD state-space realization of (11) can be immediately obtained by defining auxiliary binary variables $\delta_i(t)$, $z_i(t)$ similarly to what was done in (15) (16), and setting $\xi(t) = [y(t-1)\ \ldots\ y(t-n_a)\ u(t-1)\ \ldots\ u(t-n_b)]' \in \mathbb{R}^{n_a+n_b}$. □

5.2 PWA Realization

Analogously to what defined in (1), a PWA state-space system is defined as

$$\begin{array}{rcl}\xi(t+1) & = & A_j\xi(t)+B_ju(t)+f_j \\ y(t) & = & C_j\xi(t)+g_j\end{array}, \text{ for } \xi(t)\in\mathcal{C}_j \triangleq \{\xi:\ H_j\xi\leq D_j\} \tag{32}$$

where $\xi\in\mathbb{R}^n$, and $\{\mathcal{C}_j\}_{j=0}^{s-1}$ is a polyhedral partition of the state-space (more in general, the partition is defined in the augmented state+input space and a direct feedthrough of the input on the output can be present, although this is not the case for the HHARX models considered in this paper).

Despite the fact that a PWA state-space realization can be directly obtained from the MLD realization, in view of the equivalence result of [25] mentioned earlier, a PWA realization can be obtained directly, according to the following

Proposition 3 *HHARX models (11) admit a PWA state-space realization (32) with* n_a+n_b *states and at most* 2^M *regions.*

Proof. As for the MLD realization, define $\xi(t)=[y(t-1)\ \dots\ y(t-n_a)\ u(t-1)\ \dots\ u(t-n_b)]'\in\mathbb{R}^{n_a+n_b}$. In (11), the max functions can be either 0 or $x'\theta_i$, depending on the sign of $x'\theta_i$. Since there are M max functions, we have 2^M combinations I_j, $j=1,\dots,2^M$,

$$y(t) = [\xi'(t)\ 1]\theta_0+\sum_{i\in I}[\xi'(t)\ 1]\theta_i \tag{33a}$$

$$\text{if}\ \left\{\begin{array}{ll}[\xi'(t)\ 1]\,\theta_i\geq 0, & \forall i\in I_j\\ [\xi'(t)\ 1]\,\theta_h\leq 0, & \forall h\in\{1,\dots,M\}\setminus I_j\end{array}\right. \tag{33b}$$

Clearly, (33a) defines an affine output gain C_j, g_j, and (33b) a polyhedral region $H_j\xi\leq D_j$. The matrices A_j, B_j, f_j can be easily defined. Note that $\{H_j,D_j\}$ is the polyhedral partition generated by the hinges in the ξ-space, and that in general some of the regions will be empty. A state-space realization of the discontinuous HH model introduced in Section 4.2 can be obtained similarly. □

Remark 1 While HH functions can be represented as PWA mappings, the converse is clearly false, as the guardlines of the partition induced by the hinging hyperplanes can be only unbounded, while in general this is not required by PWA mappings (consider for instance a "hive" as a partition).

5.3 MMPS Realization

Max-Min-Plus-Scaling (MMPS) state space models where introduced in [1] and have the form

$$\begin{array}{rcl}\xi(t+1) & = & \mathcal{M}_\xi(\xi(t),u(t),d(t))\\ y(t) & = & \mathcal{M}_y((\xi(t),u(t),d(t))\\ c & \geq & \mathcal{M}_c(\xi(t),u(t),d(t))\end{array} \tag{34}$$

where $\mathcal{M}_\xi$, $\mathcal{M}_y$, $\mathcal{M}_c$ are MMPS expressions (i.e., expressions defined by the composition of max and min functions, sum, and multiplication by a scalar) in terms of the state $x(t)$, the input $u(t)$, and the auxiliary variables $d(t)$, which are all real-valued.

Proposition 4 *HHARX models (11) admit a MMPS state-space realization (34) with* n_a+n_b *states.*

Proof. As for the MLD realization, define $\xi(t) = [y(t-1) \ \dots \ y(t-n_a) \ u(t-1) \ \dots \ u(t-n_b)]' \in \mathbb{R}^{n_a+n_b}$. Then,

$$\xi(t+1) = \begin{bmatrix} \mathcal{M}_y(\xi(t), u(t)) \\ \begin{bmatrix} I_{n_a-1} & 0 \\ 0 & I_{n_b-1} \end{bmatrix} \xi(t) + e_{n_a+1} u(t) \end{bmatrix} \tag{35a}$$

$$y(t) = \mathcal{M}_y(\xi(t), u(t)) \tag{35b}$$

where $\mathcal{M}_y$ is directly obtained by (11) ($\mathcal{M}_c$, $d(t)$ are empty), and $e_{n_a+1} = [0 \ \dots \ 1 \ 0 \ \dots \ 0]'$ is the (n_a+1)-th column of the identity matrix I_{n_a+n+b}.

5.4 LC and ELC Realization

Linear Complementary (LC) and Extended Linear Complementary (ELC) state-space realizations can be obtained either directly, or by exploiting the equivalences described in [1].

6 Conclusions

In this paper we have addressed the problem of identification of hybrid dynamical systems, by focusing our attention on piecewise affine autoregressive exogenous (PWARX) and hinging hyperplanes autoregressive exogenous (HHARX) models. In particular, for the latter class we have provided globally convergent algorithms based on mixed-integer linear or quadratic programming. Several problems remain open, such as the choice of persistently exciting input signals u for identification (i.e., that allow for the identification of all the affine dynamics), and criteria like Akaike's criterion for choosing the best order and number of hinging pairs in HHARX models. We have also discussed issues of state-space realization of HHARX models into several existing discrete-time hybrid state-space models (MLD, PWA, MMPS, LC, ELC).

The hybrid systems community has recently dedicated substantial attention to develop tools for modeling, stability analysis, control synthesis, verification, and fault detection, which strongly rely on the availability of a hybrid model of the process at hand. We hope that this paper attracts attention on the problem of estimating such a hybrid model from data.

Acknowledgments

We thank Lennart Ljung and Jacob Roll for their useful comments, and for the stimulating discussions.

References

[1] W.P.M.H. Heemels and B. De Schutter. On the equivalence of classes of hybrid systems: Mixed logical dynamical and complementary systems. Technical Report 00I/04, Eindhoven University of Technology, Dept. of Electrical Engineering, June 2000.

[2] G. Labinaz, M.M. Bayoumi, and K. Rudie. A survey of modeling and control of hybrid systems. *Annual Reviews of Control*, 21:79–92, 1997.

[3] M.S. Branicky, V.S. Borkar, and S.K. Mitter. A unified framework for hybrid control: model and optimal control theory. *IEEE Trans. Automatic Control*, 43(1):31–45, 1998.

[4] M. Johannson and A. Rantzer. Computation of piece-wise quadratic Lyapunov functions for hybrid systems. *IEEE Trans. Automatic Control*, 43(4):555–559, 1998.

[5] M.S. Branicky. Multiple Lyapunov functions and other analysis tools for switched and hybrid systems. *IEEE Trans. Automatic Control*, 43(4):475–482, April 1998.

[6] J. Lygeros, C. Tomlin, and S. Sastry. Controllers for reachability specifications for hybrid systems. *Automatica*, 35(3):349–370, 1999.

[7] A. Bemporad and M. Morari. Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35(3):407–427, March 1999.

[8] R. Alur, T.A. Henzinger, and P.-H. Ho. Automatic symbolic verification of embedded systems. *IEEE Trans. on Software Engineering*, 22:181–201, 1996.

[9] A. Chutinan. *Hybrid System Verification Using Discrete Model Approximations.* PhD thesis, Department of Electrical and Computer Engineering, Carnegie Mellon University, May 1999.

[10] S. Kowalewski, O. Stursberg, M. Fritz, H. Graf, I. Hoffmann, J. Preußig, M. Remelhe, S. Simon, and H. Treseler. A case study in tool-aided analysis of discretely controlled continuos systems: the two tanks problem. In *Hybrid Systems V*, volume 1567 of *Lecture Notes in Computer Science*, pages 163–185. Springer-Verlag, 1999.

[11] A. Bemporad, F.D. Torrisi, and M. Morari. Optimization-based verification and stability characterization of piecewise affine and hybrid systems. In B. Krogh and N. Lynch, editors, *Hybrid Systems: Computation and Control*, volume 1790 of *Lecture Notes in Computer Science*, pages 45–58. Springer Verlag, 2000.

[12] J. Lunze. Diagnosis of quantized systems based on a timed discrete-event model. *IEEE Trans. on Systems, Man & Cybernetics, Part A*, 30(3):322–335, May 2000.

[13] A. Bemporad, D. Mignone, and M. Morari. Moving horizon estimation for hybrid systems and fault detection. In *Proc. American Contr. Conf.*, 1999.

[14] J. Sjöberg, Q. Zhang, L. Ljung, A. Benveniste, B. Delyon, P.Y. Glorennec, H. Hjalmarsson, and A. Juditsky. Nonlinear black-box modeling in system identification: a unified overview. *Automatica*, 31(12):1691–1724, 1995.

[15] C.A. Lehalle and R. Azencott. Piecewise affine neural networks and nonlinear control. In *Proc. 8th Int. Conf. on Artificial Neural Networks (ICANN 98)*, volume 2, pages 633–638, 1998.

[16] M.C. Medeiros, A. Veiga, and M.G.C. Resende. A combinatorial approach to piecewise linear time series analysis. Technical Report 99.5.1, AT&T Labs Research, Florham Park, NJ 07932 USA, 1999.

[17] L. Breiman. Hinging hyperplanes for regression, classification, and function approximation. *IEEE Transactions on Information Theory*, 39(3):999–1013, May 1993.

[18] P. Pucas and J. Sjöberg. On the hinge-finding algorithm for hinging hyperplanes. *IEEE Trans. Inform. Theory*, 44(3):1310–1319, May 1998.

[19] L.O. Chua and A. Deng. Canonical piecewise-linear representation. *IEEE Trans. Circuits Syst. I*, 35:511–525, 1988.

[20] C. Kahlert and L.O. Chua. The complete canonical piecewise-linear representation — part i: The geometry of the domain space. *IEEE Trans. Circuits Syst. I*, 39:222–236, 1992.

[21] P. Julián, A. Desages, and O. Agamennoni. High level canonical piecewise linear representation using a simplicial partition. *IEEE Trans. Circuits Syst. I*, 46:463–480, 1999.

[22] P. Julián. *A High Level Canonical Piecewise Linear Representation: Theory and Applications.* Phd thesis, Universidad National del Sur, Bah/'ıa Blanca, 1999.

[23] P. Julian, M. Jordan, and A. Desages. Canonical piecewise-linear approximation of smooth functions. *IEEE Trans. Circuits and Systems — I: Fundamental Theory and Applications*, 45(5):567–571, May 1998.

[24] E.D. Sontag. Interconnected automata and linear systems: A theoretical framework in discrete-time. In R. Alur, T.A. Henzinger, and E.D. Sontag, editors, *Hybrid Systems III - Verification and Control*, number 1066 in Lecture Notes in Computer Science, pages 436–448. Springer-Verlag, 1996.

[25] A. Bemporad, G. Ferrari-Trecate, and M. Morari. Observability and controllability of piecewise affine and hybrid systems. *IEEE Trans. Automatic Control*, to appear. `http://control.ethz.ch/`.

[26] Y. Chikkula, J.H. Lee, and B. Ogunnaike. Dynamic scheduled model predictive control using hinging hyperplane models. *AIChE Journal*, 44:2658–2674, 1998.

[27] E.D. Sontag. Nonlinear regulation: The piecewise linear approach. *IEEE Trans. Automatic Control*, 26(2):346–358, April 1981.

[28] L. Ljung. *System Identification : Theory for the User.* Prentice Hall, 2^{nd} edition, 1999.

[29] S. Ernst. Hinging hyperplane trees for approximation and identification. In *Proc. 37th IEEE Conf. on Decision and Control*, volume 2, pages 1266–1271, 1998.

[30] A. Bemporad, F. Borrelli, and M. Morari. Piecewise linear optimal controllers for hybrid systems. In *Proc. American Contr. Conf.*, 2000.

[31] T.M. Cavalier, P.M. Pardalos, and A.L. Soyster. Modeling and integer programming techniques applied to propositional calculus. *Computers Opns Res.*, 17(6):561–570, 1990.

[32] R. Raman and I.E. Grossmann. Relation between MILP modeling and logical inference for chemical process synthesis. *Computers Chem. Engng.*, 15(2):73–84, 1991.

[33] H.P. Williams. *Model Building in Mathematical Programming.* John Wiley & Sons, Third Edition, 1993.

[34] ILOG, Inc. *CPLEX 6.5 User Manual.* Gentilly Cedex, France, 1999.

[35] N. V. Sahinidis. BARON — Branch And Reduce Optimization Navigator. Technical report, University of Illinois at Urbana-Champaign, Dept. of Chemical Engineering, Urbana, IL, USA, 2000.

[36] T.A. Henzinger, P.-H. Ho, and H. Wong-Toi. HyTech: a model checker for hybrid systems. *Software Tools for Technology Transfer*, 1:110–122, 1997.

[37] 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.

[38] A. Bemporad, F.D. Torrisi, and M. Morari. Verification of mixed logical dynamical models — the batch evaporator process benchmark. Technical Report AUT00-04, Automatic Control Laboratory, ETH Zurich, Switzerland, February 2000. `http://control.ethz.ch/`.

[39] M.C. Medeiros, M.G.C. Resende, and A. Veiga. Piecewise linear time series estimation with GRASP. Technical report, AT&T Labs Research, Florham Park, NJ 07932 USA, 1999.

[40] M.A. Jordan and G.B. Sentoni. An integrating approach for modeling nonlinear dynamic systems with embedded logic. In *Proc. American Contr. Conf.*, pages 1092–1093, 1998.