Identification of Piecewise Affine Systems via Mixed-Integer Programming

Jacob Roll, Alberto Bemporad, Lennart Ljung

Division of Automatic Control
Department of Electrical Engineering
Linköpings universitet, SE-581 83 Linköping, Sweden
WWW: http://www.control.isy.liu.se
E-mail: roll,ljung@isy.liu.se, bemporad@dii.unisi.it

January 9, 2003

Report no.: LiTH-ISY-R-2480
Submitted to Automatica

Identification of Piecewise Affine Systems via Mixed-Integer Programming

Jacob Roll*, Alberto Bemporad†, Lennart Ljung*

January 9, 2003

Abstract

This paper addresses the problem of identification of hybrid dynamical systems, by focusing the attention on hinging hyperplanes (HHARX) and Wiener piecewise affine (W-PWARX) autoregressive exogenous models. In particular, we provide algorithms based on mixed-integer linear or quadratic programming which are guaranteed to converge to a global optimum. For the special case where switches occur only seldom in the estimation data, we also suggest a way of trading off between optimality and complexity by using a change detection approach.

1 Introduction

Hybrid systems are systems with both continuous and discrete dynamics, the former typically associated with physical principles, the latter with logic devices. Most literature on hybrid systems has dealt with modeling [1,2], stability analysis [3,4], control [2,5,6], verification [7–9], and fault detection [10,11]. The different tools rely on a model of the hybrid system. Getting such a model from data is an identification problem, which does not seem to have received enough attention in the hybrid systems community, except for the recent contribution [12]. On the other hand, in other fields there has been extensive research on identification of general nonlinear black-box models [13]. A few of these techniques lead to piecewise affine (PWA) models of nonlinear dynamical systems [14–28], and thanks to the equivalence between PWA systems [1, 29, 30] and several classes of hybrid systems, they can be used to obtain hybrid models.

As will be pointed out, if the *guardlines* (i.e., the partition of the PWA mapping) are known, the problem of identifying PWA systems can easily be solved using standard techniques. However, when the guardlines are unknown the problem becomes much more difficult. There are two alternatives to tackle such a problem: (1) Define a priori a grid of cells over which the system dynamics is linear, or (2) Estimate the grid along with the linear models. The former approach is used, e.g., in [14], and gives a simple estimation process for the linear submodels, but suffers from the curse of dimensionality in the sense that the number of a priori given cells will have to be very large for reasonable flexibility even in the case of moderately many regressors. The second approach

*Division of Automatic Control, Linköping University, Sweden `roll,ljung@isy.liu.se`
†Dip. Ingegneria dell'Informazione, Università di Siena, Italy, `bemporad@dii.unisi.it`

allows for efficient use of fewer cells, but leads to potentially (very) many local minima, which may make it difficult to apply local search routines. Depending on how the partition is determined, one can distinguish between four different types of approaches:

- All parameters, both the parameters determining the partition and the parameters of the submodels, are identified simultaneously [15–18].
- All parameters are identified simultaneously for a model class with a very simple partition, and new submodels/regions are added when needed [17–22].
- The partition and submodels are identified iteratively or in several steps, each step considering either the partition or the models [12, 23–25].
- The partition is determined using only information about the distribution of the regression vectors [26, 27].

Most of these approaches [15–22, 27] assume that the system dynamics is continuous, while, e.g., [12] allows for discontinuities. For a more detailed description of the different approaches, see [31].

In this paper, we focus on the approach where both the partition and the submodels are identified simultaneously, and point to reformulations of the identification problem for two subclasses of PWA models that lead to mixed-integer linear or quadratic programming problems that can be solved for the global optimum (in contrast to the previously mentioned contributions, which can only guarantee suboptimal solutions). These classes are the *hinging hyperplane ARX (HHARX) models* and *piecewise affine Wiener models (W-PWARX)*. Although the worst-case complexity is high, these algorithms may be useful in cases where relatively few data are available (e.g., where it is very costly to obtain data), and where it is of importance to get a model which is as good as possible. As we will see, however, for one of the two model classes, namely Wiener models, the worst-case complexity will not be exponential, but polynomial.

We also discuss some ideas on how complexity can be drastically reduced for the case of slowly varying PWA systems.

This paper extends results previously presented in [31–34].

2 PWARX Models

To begin with, let us consider systems on the form

$$y_t = g(\phi_t) + e_t \tag{1}$$

where $\phi_t \in \mathbb{R}^n$ is our regression vector, e_t is white noise, and g is a PWA function of the form

$$g(\phi) = d_j'\phi + c_j \quad \text{if} \quad \bar{H}_j\phi \le \bar{D}_j \tag{2}$$

where $d_j \in \mathbb{R}^n$, $c_j \in \mathbb{R}$, $\bar{H}_j \in \mathbb{R}^{M_j \times n}$, $\bar{D}_j \in \mathbb{R}^{M_j}$, "$\le$" denotes componentwise inequality, and the sets $\mathcal{C}_j \triangleq \{\phi : \ \bar{H}_j\phi \le \bar{D}_j\}$, $j = 1, \ldots, s$ are a polyhedral partition of the ϕ-space. The subscripts in, e.g., $\bar{H}_j$ refer to the different parts of the partition, while superscripts, e.g., $\bar{H}_j^i$ will be used to denote the ith row of $\bar{H}_j$. To allow for a more compact notation, we let $\varphi_t = \left[\begin{smallmatrix} 1 \\ \phi_t \end{smallmatrix}\right]$, $\theta_j = \left[\begin{smallmatrix} c_j \\ d_j \end{smallmatrix}\right]$, and $H_j = [-\bar{D}_j \ \bar{H}_j]$. In this way (2) can be written as

$$g(\varphi) = \varphi' \theta_j \quad \text{if} \quad H_j \varphi \leq 0 \tag{3}$$

φ_t could, e.g., consist of old inputs and outputs, i.e.,

$$\varphi_t = [1 \ y_{t-1} \ \ldots \ y_{t-n_a} \ u_{t-1} \ \ldots \ u_{t-n_b}]' \tag{4}$$

In this case we call the systems PWARX (PieceWise affine AutoRegressive eXogenous) systems. We do not assume that g is necessarily continuous over the boundaries, commonly referred to as *guardlines*. Without this assumption, definition (2) is not well posed in general, as the function can be multiply defined over common boundaries of the sets $\mathcal{C}_j$. Although one can avoid this issue by replacing some of the "$\leq$" inequalities into "$<$" in the definition of the regions $\mathcal{C}_j$, this issue is not of practical interest from a numerical point of view.

2.1 Identification of PWARX Models

Now suppose that we are given y_t and φ_t, $t = 1, \ldots, N$, and want to find the PWARX model that best matches the given data. The identification of model (3) can be carried out by solving the optimization problem

$$\min_{\theta_j, H_j} \frac{1}{2N} \sum_{t=1}^{N} \left(\sum_{j=1}^{s} \| y_t - \varphi_t' \theta_j \| \mathcal{J}_j(\varphi_t) \right) \tag{5a}$$

$$\text{subj. to } \mathcal{J}_j(\varphi_t) = \begin{cases} 1 & \text{if } H_j \varphi_t \leq 0 \\ 0 & \text{otherwise} \end{cases} \tag{5b}$$

$$+ \text{ linear bounds over } \theta_j, H_j \tag{5c}$$

where θ_j, H_j, $j = 1, \ldots, s$ are the unknowns. In (5), we will focus on the 1-norm ($|\cdot|$) and the squared Euclidean norm ($\|\cdot\|_2^2$), as they allow to express (5) as a *mixed-integer linear or quadratic program* (MILP/MIQP), respectively, for which efficient solvers exist [35–38]. The problem can be also recast as an MILP by using infinity norm over time (i.e. $\max_{t=1,\ldots,N}$ instead of $\sum_{t=1}^{N}$), although this would be highly sensitive to possible outliers in the estimation data. We distinguish between two main cases:

A. Known Guardlines H_j (i.e., the partition of the φ-space) are known, θ_j have to be estimated. If using 2-norm in (5), we can see that this is an ordinary least-squares problem which can be solved efficiently.

B. Unknown Guardlines Both H_j and θ_j are unknown. This is a much harder problem, since it is nonconvex and the objective function generally contains several local minima. However, the optimization problem (5) can be recast as an MILP or MIQP. In the following sections, we focus on two subsets of PWA functions, namely the Hinging Hyperplanes (HH) and Wiener processes with PWA static output mapping, and detail the mixed-integer program associated to the identification problem. In general, the complexity of the mixed-integer program needed to solve (5) is related to the number of samples N and regions s, and the number of parameters H_j, θ_j that are unknown. Note that in general, the guardlines $H_j^i \varphi \leq 0$, cannot be determined exactly from the given estimation data set, as the pairs y_t, φ_t are a discrete set of points which can be divided by a continuum of possible guardlines.

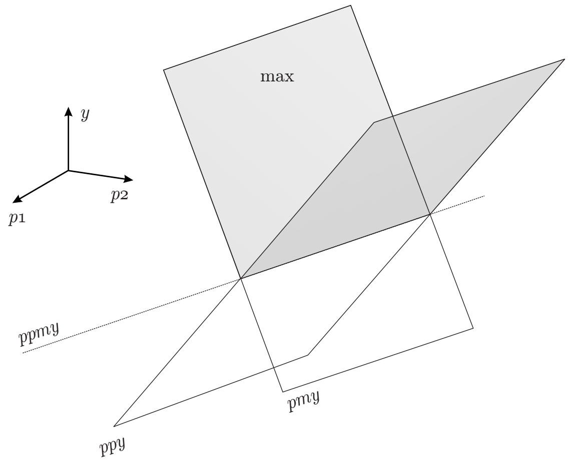


Figure 1: Hinging hyperplanes and hinge function

3 Hinging Hyperplane Models

Hinging hyperplane (HH) models were introduced by Breiman [19]. They are defined as a sum of hinge functions $g_i(\varphi) = \pm\max\{\varphi'\theta_i^+, \varphi'\theta_i^-\}$, which each consist of two half-hyperplanes, parametrized by θ_i^+ and θ_i^-, respectively (see Fig. 1). The $\pm$ sign is needed to represent both convex and nonconvex functions. However, since this will only have a minor effect on the computations in this paper, we will exclude it for notational simplicity, and only use positive max functions. Using an alternative parametrization we obtain the following HHARX (Hinging-Hyperplane AutoRegressive eXogenous) model

$$y_t = \varphi_t'\theta_0 + \sum_{i=1}^{M} \max\{\varphi_t'\theta_i, 0\} + e_t \tag{6}$$

Since $-z + \max\{z, 0\} = \max\{-z, 0\}$, $\forall z \in \mathbb{R}$, there are redundancies in (6) (i.e., the structure is not globally identifiable, so the same system can be described by several different sets of parameter values), which can be partially avoided by introducing the requirement

$$w'\theta_1 \geq \cdots \geq w'\theta_M \geq 0,\ i \in [1, M] \tag{7}$$

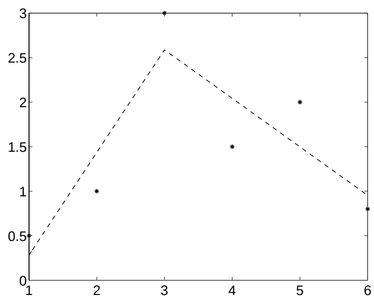
where w is any nonzero vector in $\mathbb{R}^n$, e.g., $w = \underline{1} \triangleq [1\ 1\ \ldots\ 1]'$ (or any random vector).

4 Identification Algorithms for HH Models

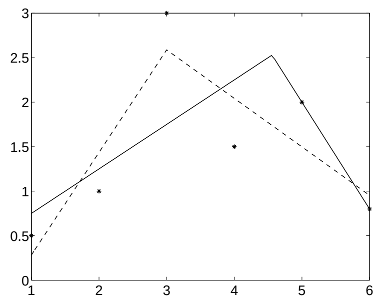
The first algorithm for estimating HH models was proposed by Breiman [19]. Later, in [18] it is shown that the original algorithm is a special case of Newton's method, and a modification is provided which guarantees convergence to a *local* minimum. Other algorithms have been proposed based on tree HH models [22]. 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.

For a noiseless system consisting of one single hinge, the method proposed in [19] was shown to converge to the global minimum. However, for noisy systems or systems with multiple hinges, local minima may lead to problems even in very simple examples, as the following example shows.

Example 1 *Consider the problem of fitting a hinge function to the six data samples given in Fig. 2(a), using a 2-norm criterion. Fig. 2(a) also shows the corresponding globally optimal function, with the optimal cost* 0.98*. However, Breiman's method will not converge to the optimal solution (regardless of the initial value), but will in most cases converge to the local minimum corresponding to the function plotted in Fig. 2(b), with the associated cost* 2.25*. The modified method provided in [18] will converge to the global optimum if starting sufficiently close to it, but will converge to the local optimum if the hinge is originally placed between* 4 *and* 5*.*



(a) Data samples ($*$) and globally optimal model (solid).



(b) Locally (solid) and globally (dashed) optimal model.

Figure 2: Identification of a single hinge function.

Consider the problem of estimating a HH function of the form (6) from the estimation data set $\{y_t, \varphi_t\}_{t=1}^N$. Let us introduce the notation

$$\Theta = \begin{pmatrix} \theta_0 & \dots & \theta_M \end{pmatrix}$$

and $g(\varphi_t, \Theta)$ for the parametrized system function according to (6). We choose the optimal parameters Θ^* by solving

$$\Theta^* \triangleq \arg\min V(\Theta) \triangleq \sum_{t=1}^{N} |y_t - g(\varphi_t, \Theta)| \tag{8a}$$

$$\text{subj. to} \quad \begin{cases} \theta^{j-} \leq \theta_j \leq \theta^{j+} \\ \underline{1}'\theta_i \geq 0, \ i \in [1, M] \end{cases} \tag{8b}$$

where the inequalities in (8b) are componentwise. As we will see, (8) can be reformulated as an MILP. Another possibility is to use the squared Euclidean norm $(y_t - g(\varphi_t, \Theta))^2$, which gives a problem that can be recast as an MIQP.

4.1 Optimization Problem

MILP Formulation. To recast (8) as an MILP, we introduce the 0-1 variables δ_{it}:

$$[\delta_{it} = 0] \leftrightarrow [\varphi_t'\theta_i \leq 0],\ i \in [1, M],\ t \in [1, N] \tag{9}$$

and the new continuous variables z_{it}

$$z_{it} = \max\{\varphi_t'\theta_i, 0\} = \varphi_t'\theta_i\delta_{it} \tag{10}$$

The relations (9) and (10) can be transformed into mixed-integer linear inequalities, by using a slight modification of standard techniques described in [6] (see also [31]). By assuming that the bounds over θ_i are all finite, Eq. (9) and (10) are equivalent[1] to the inequalities

$$\begin{aligned} & z_{it} \geq 0 \\ & z_{it} \leq M_{it}^{\theta}\delta_{it} \\ & \varphi_t'\theta_i \leq z_{it} \\ & (1 - \delta_{it})m_{it}^{\theta} + z_{it} \leq \varphi_t'\theta_i \end{aligned} \tag{11}$$

where M_{it}^{θ} and m_{it}^{θ} are upper and lower bounds on $\varphi_t'\theta_i$, respectively, derived from the bounds on θ_i.

Finally, by introducing auxiliary slack variables $\epsilon_t \geq |y_t - g(\varphi_t, \Theta)|$, $t = 1, \ldots, N$, the following holds:

Proposition 1 *The optimum of problem* (8) *is equivalent to the optimum of the following MILP*

$$\begin{aligned} & \min_{\epsilon_t, \theta_i, z_{it}, \delta_{it}} \sum_{t=1}^{N} \epsilon_t \\ & \text{subj. to } \epsilon_t \geq y_t - \varphi_t'\theta_0 - \sum_{i=1}^{M} z_{it} \\ & \qquad \epsilon_t \geq \varphi_t'\theta_0 + \sum_{i=1}^{M} z_{it} - y_t \\ & \qquad (11), (7) \end{aligned} \tag{12}$$

Example 2 *Consider the following HHARX model*

$$\begin{aligned} y_t = {} & 0.8y_{t-1} + 0.4u_{t-1} - 0.1+ \\ & + \max\{-0.3y_{t-1} + 0.6u_{t-1} + 0.3, 0\} \end{aligned} \tag{13}$$

The model is identified on the data reported in Fig. 4(a), by solving an MILP with 66 variables (of which 20 integers) and 168 constraints. The problem is solved by using Cplex 6.5 [38] (1014 LP solved in 0.68 s on a Sun Ultra 10

[1] For the equivalence to hold, the last inequality of (11) should be strict; otherwise δ_{it} will not be uniquely determined when $\varphi_t'\theta_i = 0$. However, because of the continuity of the hinge functions, it does not matter in this case if δ_{it} is 0 or 1, and therefore the non-strict inequality will be used to facilitate implementation.

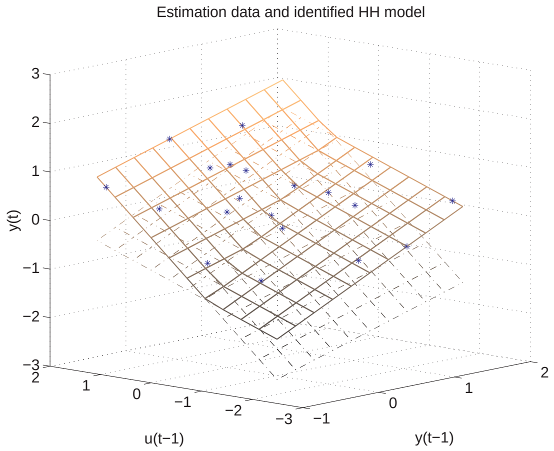
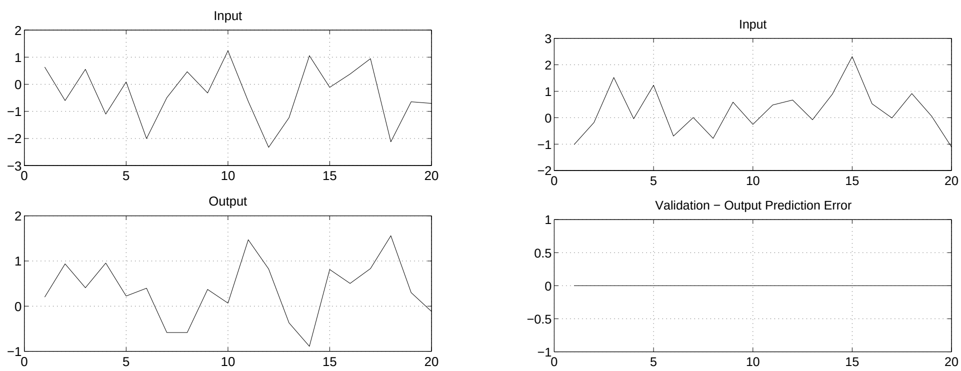


Figure 3: Identification of model (13) – noiseless case. Identified HH model.



(a) Estimation data

(b) Validation data

Figure 4: Identification of model (13) – noiseless case

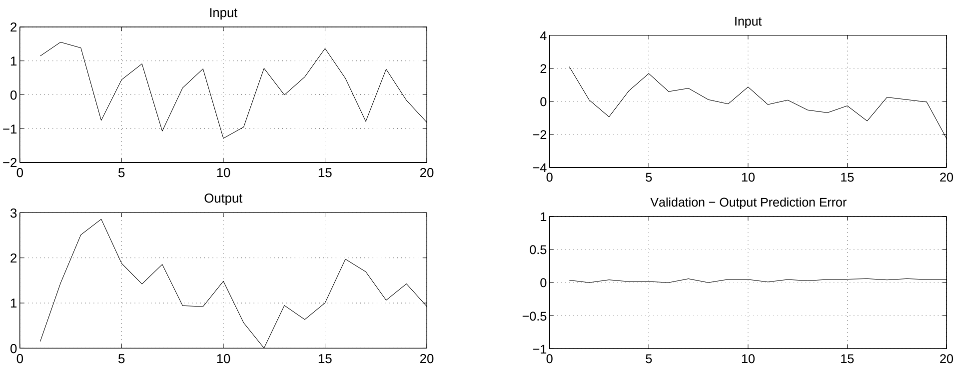
running Matlab 5.3), and, for comparison, using BARON [36] (73 LP solved in 3.00 s, same machine), which results in a zero output prediction error (Fig. 4(b)). The fitted HH model is shown in Fig. 3. After adding white Gaussian noise e_t with zero mean and variance 0.01 to the output y_t, the following model

$$\begin{aligned} y_t = 0.83y_{t-1} + 0.34u_{t-1} - 0.20+ \\ + \max\{-0.34y_{t-1} + 0.62u_{t-1} + 0.40, 0\} \end{aligned} \tag{14}$$

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. 5(a), and produces the validation data reported in Fig. 5(b). For comparison, we identified the linear ARX model

$$y_t = 0.82y_{t-1} + 0.72u_{t-1} \tag{15}$$

on the same estimation data, obtaining the validation data reported in Fig. 6 (higher order ARX models did not produce significant improvements). Clearly,



(a) Estimation data (b) Validation data

Figure 5: Identification of model (13) – noisy case

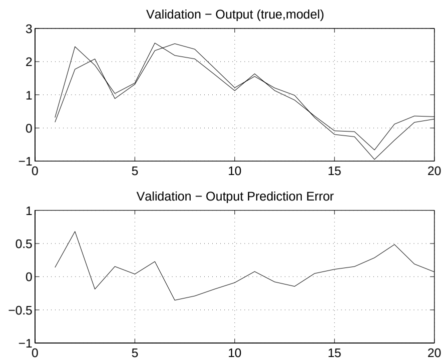


Figure 6: Identification of a linear ARX model – same estimation and validation data as in Fig. 5

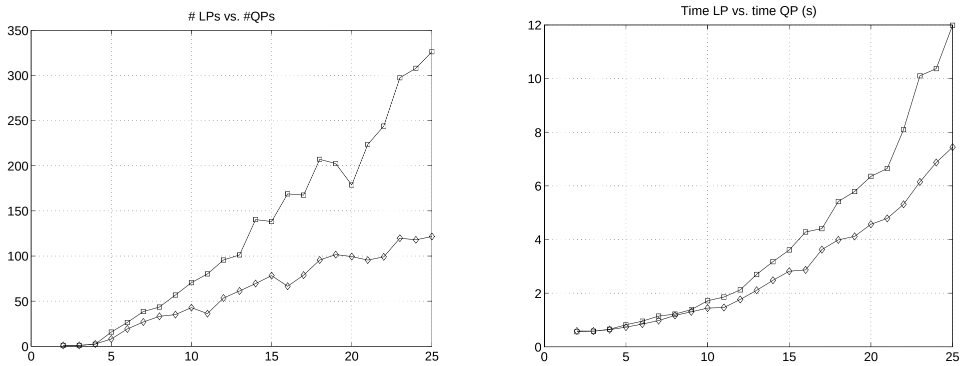
the error generated by driving the ARX model in open-loop with the validation input u_t is much larger, and would not make (15) *suitable for instance for formal verification tools, where a good performance of open-loop prediction is a critical requirement.*

MIQP Formulation. When the squared 2-norm is used in the objective function, the optimization problem can be recast as the MIQP

$$\begin{aligned} \min_{\theta_i,\delta_{it},z_{it}} \ & V(\Theta) \triangleq \sum_{t=1}^{N}(y_t - (\varphi_t'\theta_0 + \sum_{i=1}^{M} z_{it}))^2 \\ & \text{subj. to } (11), (7) \end{aligned} \tag{16}$$

Note that the problem is not strictly positive definite, for instance the cost function does not depend on θ_i, δ_{it} (which only appear in the constraints). For numerical reasons, a term σI, where σ is a small number, may be added to the Hessian associated to the MIQP (16).

Example 3 *Consider again the PWARX system* (13)*. In Fig. 7 we compare the performance in terms of LP/QPs and total computation time of the linear criterion* (12) *versus the quadratic criterion* (16)*. The reported numbers are computed on a Sun Ultra 60 (*2×360 *MHz) running Matlab 5.3 and the solver BARON [36], by averaging the number of LP/QPs and computation times, respectively, for ten estimation data sets generated by feeding random Gaussian inputs* u_t *and zero output noise to system* (13)*.*



(a) Average number of LPs and QPs

(b) Average computation time

Figure 7: Identification of model (13) – MILP (diamonds) vs. MIQP (squares). The horizontal axes show the number of estimation data samples.

4.2 Complexity

Despite the good solvers available [35, 36, 38], 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 approach is computationally affordable only for model with few data, or if data are clustered together. An example of the latter approach is given in Section 5, where a piecewise affine function is identified over a sliding window.

4.3 Discontinuous HHARX Models

In HHARX models, the output y_t is a continuous function of the regressor ϕ_t. On the other hand, hybrid systems often consist of PWA discontinuous mappings. In order to tackle discontinuities, we can modify the HH model (6) in the form

$$g(\varphi_t, \Theta) = \varphi_t'\theta_0 + \sum_{i=1}^{M}(\varphi_t'\theta_i + a_i)\delta_{it}(\varphi_t) \tag{17a}$$

$$[\delta_{it}(\varphi_t) = 0] \leftrightarrow [\varphi_t'\theta_i \leq 0],\ i \in [1, M],\ t \in [1, N] \tag{17b}$$

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

$$g(\varphi_t, \Theta) = \varphi_t'\theta_0 + \sum_{i=1}^{M}(\varphi_t'\theta_i)\delta_{it}(\varphi_t) \tag{18a}$$

$$[\delta_{it}(\varphi_t) = 0] \leftrightarrow [\varphi_t'\mu_i \le 0],\ i \in [1, M], t \in [1, N] \tag{18b}$$

where μ_i, $i = 1, \dots, M$ are additional free vectors of parameters, $\mu_i^- \le \mu_i \le \mu_i^+$, $\underline{1}'\mu_i \ge 0$. Similarly to (12), both the identification problems (17) and (18) can be again recast as an MILP. With respect to (12), the MILP has μ_i or a_i as additional optimization variables. Note that the problem in general does not have a unique solution, just as for general PWARX systems.

4.4 Robust HHARX Models

In formal verification methods, model uncertainty needs to be handled in order to provide safety guarantees. Typically, the model is associated with a bounded uncertainty. In the present context of HHARX models, we wish to find an uncertainty description of the form

$$g(\varphi_t, \Theta^-) \le y_t \le g(\varphi_t, \Theta^+),\ \forall t \ge 0 \tag{19}$$

for an inclusion-type of description, or the form

$$y_t = g(\varphi_t, \Theta^*) + n_t,\ n^- \le n_t \le n^+ \tag{20}$$

for an additive-disturbance-type of description. Clearly, since the model is identified from a finite estimation data set, fulfillment of (19) or (20) for other data than the estimation data 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 (12) or (16) with the additional linear constraints

$$y_t \ge g(\varphi_t, \Theta),\ \forall t \in [1, N] \tag{21}$$

for estimating Θ^-, and

$$y_t \le g(\varphi_t, \Theta),\ \forall t \in [1, N] \tag{22}$$

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

1. First, identify a model Θ^* by solving (12) or (16) and then compute

$$\begin{aligned} n^+ &\triangleq \max_{t=1,\dots,N} y_t - g(\varphi_t, \Theta^*) \\ n^- &\triangleq \min_{t=1,\dots,N} y_t - g(\varphi_t, \Theta^*) \end{aligned} \tag{23}$$

2. Modify the MILP (12) by replacing ϵ_t 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 Using Change Detection to Reduce Complexity

Many PWA systems of interest may only seldom switch between different modes. For such systems, it should be possible to use a change detection algorithm to roughly find the timepoints when switches occur, and use this information to reduce the complexity of (12) or (16) by forcing several samples, lying in the same interval between two switches, to belong to the same subsystem. Here we propose to use an MILP algorithm over a sliding window as a change detection algorithm. The formulation (12) is used, taking only data from time $t_0, \ldots, t_0 + L - 1$ into account, where L is the length of the window. Furthermore, only one switch is allowed in each window. Hence, the MILP solved takes the form

$$\begin{aligned} &\min_{\epsilon_t, \theta_i, z_{it}, \delta_t} \sum_{t=t_0}^{t_0+L-1} \epsilon_t \\ &\text{subj. to } \epsilon_t \geq y_t - \varphi_t' \theta_0 - z_{1t} + z_{2t} \\ &\qquad\quad\; \epsilon_t \geq \varphi_t' \theta_0 + z_{1t} - z_{2t} - y_t \\ &\qquad\quad\; \delta_{t_0} \leq \cdots \leq \delta_{t_0+L-1} \\ &\qquad\quad\; (11) \text{ with } \delta_{1t} = \delta_{2t} = \delta_t \end{aligned} \tag{24}$$

Note that we only need two hinges (one positive and one negative) and L discrete variables since only one switch is allowed, compared to the MN discrete variables needed in (12). (If the PWARX structure we would like to identify just contains positive hinges, we would only need one (positive) hinge in (24).) Furthermore, the inequalities $\delta_{t_0} \leq \cdots \leq \delta_{t_0+L-1}$ also help to reduce the complexity drastically.

In each position t_0 of the window, the fit of the local HHARX model (i.e., the optimal value of the cost function in (24)) is compared to the fit of a linear model over the same window. The value of the relative improvement of the cost function,

$$k_{t_0} = 1 - \frac{V^*_{HHARX}}{V^*_{ARX}} \tag{25}$$

is assigned to the time point of the change, and as the window is moving, these values are summed up (for each time point). If the sum of the relative improvements for a certain time point exceeds a threshold K_0, chosen by the user, this time point will be considered as a possible switch time.

The advantage of using (12) instead of a standard change detection algorithm, e.g., Brandt's GLR method (see, e.g., [39]), is that the latter does not require linear separability between the classes; nor does it take the continuity of the PWA function into account.

After having obtained the estimated possible time points of the switches as described above, we solve (12) or (16), but using the same δ variable for all samples lying in the same time interval between two consecutive possible switches. This will force the samples to belong to the same submodel, and will reduce the complexity considerably. To summarize, the algorithm consists of two phases:

1. Use a sliding window with a local MILP algorithm to detect possible switches and divide the time series into segments.

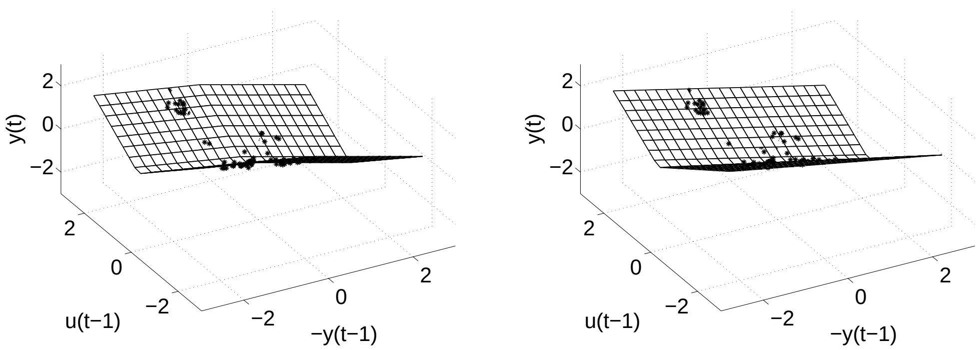
2. Use an MILP to simultaneously assign the different segments to different submodels and estimate the parameters of the submodel.

Once again, note that in the first step, the MILP solved just uses two hinges, independently of how many hinge functions the final global model contains.

Example 4 *The system*

$$\begin{aligned} y_t = & -0.3 + 1.2y_{t-1} - u_{t-1} \\ & + \max\{-1.2 + 2u_{t-1}, 0\} \\ & - \max\{-0.2y_{t-1}, 0\} + e_t \end{aligned} \tag{26}$$

where e_t is white Gaussian noise with variance 0.01, is identified using 100 data samples. The true system function and the data samples are shown in Figure 8(a). The proposed sliding window algorithm was used with $L = 15$ and $K_0 = 1$.



(a) System function (26) and estimation data.

(b) Identified model.

Figure 8: Identification of (26).

This resulted in the system shown in Figure 8(b). Table 1 shows the values of the objective function (8a) *for the true system and the identified model, for the estimation data and a set of validation data. As can be seen, the identified model shows a good performance. The computation time running CPLEX on a 333 MHz Pentium II laptop (128 MB RAM) was 144 s (42 s for the sliding windows and 102 s for the final large MILP). This should be compared to solving the MILP* (12) *directly, which did not return a solution within a maximum allotted time of 3 hours on the same computer.*

	True system	Identified model
Estimation data	7.8213	7.4833
Validation data	7.8668	8.6777

Table 1: Identification of (26) – values of the objective function (8a).

5.1 Complexity

The advantage of the described sliding window algorithm, compared to solving (12) or (16) directly, lies in the reduction of the computational complexity. In the sliding window phase, the complexity is linear in the number of data N when using a window of fixed size, as opposed to the exponential complexity of (12) and (16).

For the second phase, the complexity is closely related to the number of possible switches. Here, the thresholding procedure makes it possible to explicitly trade off between complexity of the algorithm and optimality: The higher the threshold value, the fewer possible switch times will be considered. If it is high enough, no switches will be allowed, which means that all samples will be forced to belong to the same submodel, and we will end up with a linear model. If, on the other hand, the threshold value is chosen to be zero, every time point will be considered as a possible switch time, and we will again get the globally optimal solution.

As previously mentioned, the described algorithm requires the system to switch only seldom between different modes. The general issue of designing input signals having the desired properties of sufficiently exciting the modes of the system and letting the system switch seldom is a subject for future research.

5.2 Approximating General Nonlinear Systems

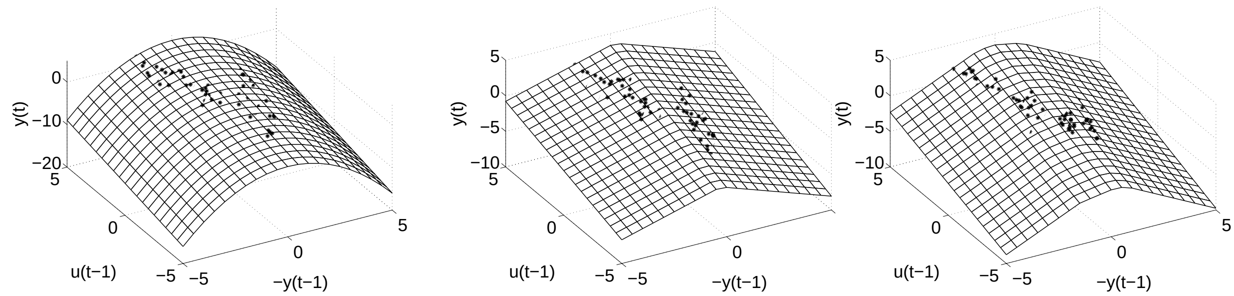
To give another example of the described sliding window algorithm, the problem of approximating a simple nonlinear system is considered. The capability of approximating arbitrary nonlinear systems is an interesting issue. Since HH functions have the universal approximation property (see, e.g., [40]), they can (under mild conditions) approximate any function arbitrarily well, given a large enough number of hinges. As a very simple illustration, a quadratic NARX (nonlinear ARX) system is approximated by a HHARX model in the following example.

Example 5 *Consider the system*

$$y_t = -0.5y_{t-1}^2 + 0.7u_{t-1} + e_t \tag{27}$$

where e_t is white Gaussian noise with variance 0.01, is identified using 100 data samples. The input is designed to make the output change sign only seldom. The true system function and the data samples are shown in Figure 9(a). Using the sliding window algorithm with one hinge, $L = 10$, and a threshold $K_0 = 1$, resulted in the system shown in Figure 9(b). We can see that the parabola is approximated by the hinge in a natural way. The computation time running CPLEX on a 333 MHz Pentium II laptop (128 MB RAM) was 19.7 s (17.7 s for the sliding windows and 2 s for the final large MILP). Solving the MILP (12) *directly required about 1300 s of computations.*

If we instead use three hinges to approximate the true system function, we get the result shown in Figure 9(c). The computation time was 152 s (20 s for the sliding windows and 132 s for the final large MILP).



(a) System function (27) and estimation data.

(b) Identified model using one hinge.

(c) Identified model using three hinges.

Figure 9: Identification of (27).

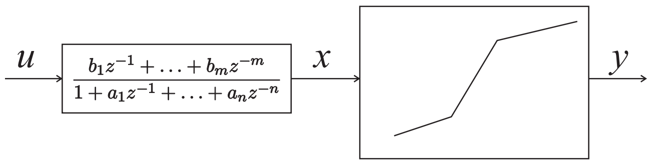


Figure 10: Wiener process with PWA static output mapping

6 Piecewise Affine Wiener Models

Let us now turn to the class of piecewise affine Wiener (W-PWARX) models, for which, as it will turn out, one can design an optimal identification algorithm whose worst-case complexity is polynomial in the number of data. The models considered will be in the form shown in Fig. 10, described by the relations

$$\begin{aligned} A(z)x_t &= B(z)u_t \\ y_t &= f(x_t) \end{aligned} \tag{28a}$$

where $A(z) = 1 + \sum_{l=1}^{n_a} a_l z^{-l}$, $B(z) = \sum_{l=1}^{n_b} b_l z^{-l}$, and z^{-1} is the delay operator, $z^{-1}x_t = x_{t-1}$. We assume that $f(x)$ is a piecewise affine, invertible function (without restrictions we can assume that f is strictly increasing), and parameterize its inverse as

$$x_t = y_t - \alpha_0 + \sum_{i=1}^{M} \pm \max\{\beta_i y_t - \alpha_i, 0\} \tag{28b}$$

Both signs $\pm$ are allowed in order to be able to represent nonconvex functions. We assume that the number M^+ of positive signs is known (without restrictions we can let these be the first terms of the sum). As $\max\{-z, 0\} = -z + \max\{z, 0\}$ for all $z \in \mathbb{R}$, without loss of generality we can also assume $\beta_i \geq 0$.

6.1 Identification of W-PWARX Models

As seen from Fig. 10, a Wiener model consists of a linear dynamic system followed by an output nonlinearity. In some cases, the two can be identified separately: first the inverse nonlinearity is estimated by supplying a quasi-static input, and then a linear dynamic model is identified by using standard linear techniques [41]. On the other hand, in some other cases the input signal cannot be designed arbitrarily, as input/output estimation data are simply supplied by other sources. Then one algorithm which estimates the whole Wiener process is desirable. Here, we describe an algorithm based on mixed-integer programming, which identifies W-PWARX models of the form (28). Such PWA form is particularly useful when the identified system models an unknown part of a larger hybrid model. We assume that we are given an estimation data set $\{y_t, u_t\}_{t=1}^N$.

Like in the HHARX case, the first thing to do is to get rid of the max functions. This is done by introducing the discrete variables $\delta_{it} \in \{0, 1\}$

$$[\delta_{it} = 1] \leftrightarrow [\beta_i y_t - \alpha_i \geq 0]\,,\; i \in [1, M],\; t \in [1, N] \tag{29}$$

Before continuing with the usual reformulation into an MIQP, let us consider some additional structure that can be used to reduce the complexity of the problem. Without loss of generality, we can assume that the M^+ first breakpoints in the PWA output nonlinearity are ordered, and similarly for the $M - M^+$ last breakpoints. Clearly, the logic constraint

$$[\delta_{it} = 1] \quad \rightarrow \quad [\delta_{jt} = 1] \tag{30}$$

should hold for all $i, j \leq M^+$ such that $j < i$, and for all $i, j > M^+$ such that $j < i$. Each constraint (30) is translated into

$$\delta_{it} - \delta_{jt} \leq 0, \tag{31}$$

and a minimal set of inequalities is obtained by collecting (31) only for pairs of consecutive indices i, j. Moreover, since the output data y_t can be ordered, we can also get additional relations on δ_{it} by using (29). In fact, if $\delta_{it_0} = 1$ and $y_{t_1} > y_{t_0}$, it must follow that $\delta_{it_1} = 1$. We can translate these relations into

$$\delta_{it_0} - \delta_{it_1} \leq 0,\; \forall t_1 \neq t_0 : y_{t_1} \geq y_{t_0} \tag{32}$$

Both (31) and (32) will help to reduce the search space considerably in the optimization.

One specific problem for this model structure is that we will get products between the coefficients a_h of the $A(z)$ polynomial and the coefficients inside the max functions, β_i and α_i. Furthermore, since a_h may very well be negative, the inequalities in the definition (29) of $\delta_i(t)$ may change directions if we multiply by a_h. This is not desirable, so to get rid of these problems, first define $a_h = a_h^+ - a_h^-$, where $a_h^+, a_h^- \geq \gamma$, and $\gamma > 0$ is any positive scalar. Then

$$\begin{aligned}
&a_h \max\{\beta_i y_{t-h} - \alpha_i, 0\} = \\
&= \max\{a_h^+ \beta_i y_{t-h} - a_h^+ \alpha_i, 0\} - \\
&\quad - \max\{a_h^- \beta_i y_{t-h} - a_h^- \alpha_i, 0\} \\
&= \max\{c_{ih}^+ y_{t-h} - d_{ih}^+, 0\} - \max\{c_{ih}^- y_{t-h} - d_{ih}^-, 0\}
\end{aligned}$$

where

$$\begin{aligned}
c_{ih}^\pm &\triangleq a_h^\pm \beta_i, \\
d_{ih}^\pm &\triangleq a_h^\pm \alpha_i,\;\; i \in [1, M],\;\; h \in [1, n_a]
\end{aligned}$$

Let also

$$
\begin{aligned}
c_{i0} &= c_{i0}^{+} = c_{i0}^{-} \triangleq \beta_i \\
d_{i0} &= d_{i0}^{+} = d_{i0}^{-} \triangleq \alpha_i \\
d_{0h} &\triangleq a_h \alpha_0 \\
d_{00} &\triangleq \alpha_0 \\
\bar{d}_0 &\triangleq \sum_{h=0}^{n_a} d_{0h} = \left(1 + \sum_{h=1}^{n_a} a_h\right) \alpha_0
\end{aligned}
$$

As $a_h^+, a_h^- > 0$, from (29) it now follows

$$
[\delta_{it} = 1] \leftrightarrow \left[c_{ih}^{\pm} y_t - d_{ih}^{\pm} \geq 0\right] \tag{33}
$$

Let us also introduce the auxiliary continuous variables

$$
\begin{aligned}
z_{it0} &\triangleq (c_{i0} y_t - d_{i0}) \delta_{it} \\
z_{ith} &\triangleq [(c_{ih}^{+} - c_{ih}^{-}) y_{t-h} - (d_{ih}^{+} - d_{ih}^{-})] \delta_{i(t-h)}, \;\; h \in [1, n_a]
\end{aligned} \tag{34}
$$

Using the same techniques as in [6], we can translate (33) and (34) to linear inequalities.

Now,

$$
\begin{aligned}
x_t &= y_t - d_{00} + \sum_{i=1}^{M} \pm z_{it0} \\
a_h x_{t-h} &= a_h y_{t-h} - d_{0h} + \sum_{i=1}^{M} \pm z_{ith}
\end{aligned} \tag{35}
$$

By (28) and (35),

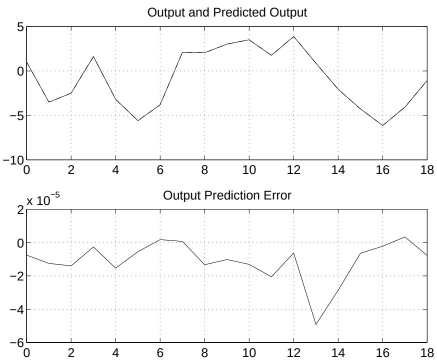
$$
\begin{aligned}
x_t = y_t - d_{00} + \sum_{i=1}^{M} \pm z_{it0} = \sum_{k=1}^{n_b} b_k u_{t-k} - \\
- \sum_{h=1}^{n_a} \left(a_h y_{t-h} - d_{0h} + \sum_{i=1}^{M} \pm z_{ith} \right)
\end{aligned}
$$

which provides the relation

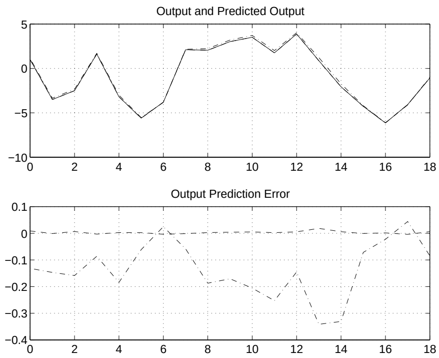
$$
y_t = -\sum_{h=1}^{n_a} a_h y_{t-h} + \sum_{k=1}^{n_b} b_k u_{t-k} + \bar{d}_0 - \sum_{i=1}^{M} \sum_{h=0}^{n_a} \pm z_{ith} \tag{36}
$$

In order to fit the estimation data to model (36), we solve the mixed-integer quadratic program (MIQP)

$$
\begin{aligned}
\min \frac{1}{N} \sum_{t=1+\max\{n_a, n_b\}}^{N} \Bigg| y_t + \sum_{h=1}^{n_a} a_h y_{t-h} - \\
- \sum_{k=1}^{n_b} b_k u_{t-k} - \bar{d}_0 + \sum_{i=1}^{M} \sum_{h=0}^{n_a} \pm z_{ith} \Bigg|^2 \\
\text{subj. to linear constr. from (31), (32), (33), and (34)}
\end{aligned} \tag{37}
$$



(a) System estimated with noiseless data.



(b) System estimated with output noise $|e_t| \leq 0.01$ (dashed), and $|e_t| \leq 0.1$ (dot-dashed)

Figure 11: Validation results

with respect to the variables a_h, b_k, c_{i0}, d_{i0}, $\bar{d}_0$, $c_{ih}^{\pm}$, $d_{ih}^{\pm}$, z_{ith}, and the binary variables δ_{it}. The solution to (37) provides the optimal parameters a_h^*, b_h^*, and $\alpha_0^* \triangleq \frac{\bar{d}_0^*}{1+\sum_{h=1}^{n_a} a_h^*}$, $\alpha_i^* \triangleq d_{i0}^*$, $\beta_i^* \triangleq c_{i0}^*$. Finally, we can obtain the estimation $f^*(x)$ by inverting (28b) (see [31] for details).

Example 6 *A Wiener model constituted by a first-order linear system and a nonlinearity with two breakpoints is identified, using $N = 20$ estimation data points. The system is first identified using noiseless data, and then using noisy measurements $\tilde{y}_t = y_t + e_t$, where e_t are independent and uniformly distributed on a symmetric interval around 0. The MIQP problem* (37) *is solved by running BARON [36] on a Sun Ultra 10. The resulting estimates are shown in Table 2. The estimated parameters are overall very close to the true values, the closer the*

Par.	True	$e_t = 0$	$\|e_t\| < 0.01$	$\|e_t\| < 0.1$
a_1	-0.5	-0.5000	-0.4990	-0.5360
b_1	2	2.0000	2.0024	2.0003
α_0	-2	-2.0000	-2.0001	-1.7748
α_1	0.5	0.5000	0.5095	0.5509
α_2	-1.5	-1.5000	-1.4924	-1.4999
β_1	0.5	0.5000	0.5016	0.5028
β_2	0.5	0.5000	0.4988	0.4876
CPU	-	45.44 s	51.33 s	90.34 s

Table 2: Estimation results

lower the intensity of the output noise, as it should be expected. The estimated model was also tested on a set of validation data, and we report in Fig. 11 the resulting one-step-ahead predicted output and output error. Note that such a good performance cannot be achieved by using standard linear identification techniques.

6.2 Complexity Analysis

By imposing the constraints expressed by (31) and (32), the degrees of freedom for the integer variables, and hence the complexity, are reduced considerably. In fact, instead of having to test 2^{MN} different cases in the *worst case*, only $\binom{M+N}{M} \cdot \binom{M}{M+}$ combinations would be tested. For example, for $N = 20$ and $M = 2$ this means that the number of possible combinations of integer variables decreases from approximately 10^{12} to 462. In general, for a fixed M the worst-case complexity grows as N^M. Note that this simplification is possible since the nonlinearity is one-dimensional, which allows an ordering of the breakpoints and of the output data.

7 State-Space Realizations

In the recent literature on hybrid systems, several formalisms for describing different system classes have emerged. In this section we show how HHARX and W-PWARX models can be expressed using some of these formalisms.

Similarly to the linear ARX case, assuming that the identified polynomials $A(z)$, $B(z)$ are coprime and $n_a \geq n_b$, a minimal state-space realization containing $n_a + n_b$ states can be obtained for HHARX models, by using different hybrid state-space discrete-time paradigms introduced recently in the hybrid systems literature. For W-PWARX, a minimal state-space realization containing n_a states can be obtained.

MLD Realization Mixed Logical Dynamical (MLD) systems [6] are a discrete-time formalism for systems containing both continuous and boolean/discrete variables. The key idea is to transform the Boolean variables into 0-1 integers, and to express the relations as mixed-integer linear inequalities, similarly to what was done in (11) and (31). The MLD model has the form

$$\xi_{t+1} = \Phi\xi_t + \mathcal{G}_1 u_t + \mathcal{G}_2 \delta_t + \mathcal{G}_3 z_t \tag{38a}$$

$$y_t = \mathcal{H}\xi_t + \mathcal{D}_1 u_t + \mathcal{D}_2 \delta_t + \mathcal{D}_3 z_t \tag{38b}$$

$$\mathcal{E}_2 \delta_t + \mathcal{E}_3 z_t \leq \mathcal{E}_1 u_t + \mathcal{E}_4 \xi_t + \mathcal{E}_5 \tag{38c}$$

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, $\delta \in \{0,1\}^{r_\ell}$, and $z \in \mathbb{R}^{r_c}$ are auxiliary variables.

Proposition 2 *HHARX models* (6) *admit an MLD state-space realization with* $n_a + n_b$ *states.*

Proof. The realization can be obtained by defining auxiliary variables δ_{it}, z_{it} similarly to what was done in (9), (10), and (11), and setting

$$\xi_t = [y_{t-1} \ \ldots \ y_{t-n_a} \ u_{t-1} \ \ldots \ u_{t-n_b}]'$$

□

Proposition 3 *W-PWARX models* (28) *admit an MLD state-space realization with* n_a *states.*

Proof. Let

$$\begin{aligned} \xi_t &= \Phi\xi_{t-1} + \mathcal{G}_1 u_t \\ x_t &= C\xi_t \\ y_t &= C\xi_t - \bar{\alpha}_0 + \sum_{i=1}^{M} \pm \max\{\bar{\beta}_i C\xi_t - \bar{\alpha}_i, 0\} \end{aligned} \tag{39}$$

be a minimal state-space realization of (28a). Define M auxiliary binary variables

$$[\delta_{it} = 1] \;\leftrightarrow\; \left[\bar{\beta}_i C\xi_t - \bar{\alpha}_i \geq 0\right]$$

and M continuous $z_{it} = (\bar{\beta}_i C\xi_t - \bar{\alpha}_i)\delta_{it}$. With translations into mixed-integer inequalities as in (11) or [6], the MLD form can be immediately obtained. □

PWA Realization Analogously to what was defined in (2), a PWA state-space system is defined as

$$\begin{aligned} \xi_{t+1} &= A_j\xi_t + B_j u_t + f_j \\ y_t &= C_j\xi_t + D_j u_t + g_j \end{aligned}, \text{ for } \begin{bmatrix} \xi_t \\ u_t \end{bmatrix} \in \mathcal{C}_j \tag{40}$$

where $\xi \in \mathbb{R}^n$, and $\{\mathcal{C}_j\}_{j=0}^{s-1}$ is a polyhedral partition of the combined state-input-space. The following propositions can be obtained as corollaries of the equivalence between MLD and PWA systems [30], and allows to construct a PWA state-space realization of (28) via the above MLD realization.

Proposition 4 *HHARX models* (6) *admit a PWA state-space realization* (40) *with $n_a + n_b$ states and at most 2^M regions.*

Proposition 5 *W-PWARX models* (28) *admit a PWA state-space realization* (40) *with n_a states and at most 2^M regions.*

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

$$\begin{aligned} \xi_{t+1} &= \mathcal{M}_\xi(\xi_t, u_t, d_t) \\ y_t &= \mathcal{M}_y((\xi_t, u_t, d_t) \\ c &\geq \mathcal{M}_c(\xi_t, u_t, d_t) \end{aligned} \tag{41}$$

where $\mathcal{M}_\xi$, $\mathcal{M}_y$, $\mathcal{M}_c$ are expressions defined by the composition of max and min functions, sum, and multiplication by a scalar, in terms of the state ξ_t, the input u_t, and the auxiliary variables d_t, which are all real-valued. When $\mathcal{M}_c$ is empty, we refer to these systems as unconstrained MMPS systems.

Proposition 6 *HHARX models* (6) *admit an MMPS state-space realization* (41) *with $n_a + n_b$ states.*

Proof. As for the MLD realization, define $\xi_t = [y_{t-1} \;\ldots\; y_{t-n_a} \; u_{t-1} \;\ldots\; u_{t-n_b}]'$. Then $\mathcal{M}_y(\xi_t, u_t)$ is directly obtained by (6), and the state update is obtained by shifting the elements of ξ and using $\mathcal{M}_y(\xi_t, u_t)$.

Proposition 7 *W-PWARX models* (28) *admit an unconstrained MMPS state-space realization* (41) *with n_a states.*

Proof. Eq. (39) is already on MMPS form. □

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

8 Conclusions

In this paper we have addressed the problem of identification of hybrid dynamical systems, by focusing our attention on piecewise affine (PWARX), hinging hyperplanes (HHARX), and Wiener piecewise affine (W-PWARX) autoregressive exogenous models. In particular, for the two latter classes we have provided algorithms that always converge to the global optimum, based on mixed-integer linear or quadratic programming. As a possible step in the direction towards faster suboptimal algorithms based on the mixed-integer approach, we have also proposed a suboptimal sliding window algorithm for slowly changing HHARX models.

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.

References

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

[2] 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.

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

[4] 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.

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

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

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

[8] 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.

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

[10] 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.

[11] A. Bemporad, D. Mignone, and M. Morari. Moving horizon estimation for hybrid systems and fault detection. In *Proc. American Contr. Conf.*, pages 2471–2475, Chicago, IL, 1999.

[12] G. Ferrari-Trecate, M. Muselli, D. Liberati, and M. Morari. A clustering technique for the identification of piecewise affine systems. In A. Sangiovanni-Vincentelli and M. D. Di Benedetto, editors, *Hybrid Systems: Computation and Control*, Lecture Notes in Computer Science. Springer Verlag, 2001.

[13] 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.

[14] 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.

[15] R. Batruni. A multilayer neural network with piecewise-linear structure and back-propagation learning. *IEEE Trans. Neural Networks*, 2(3):395–403, May 1991.

[16] E. F. Gad, A. F. Atiya, S. Shaheen, and A. El-Dessouki. A new algorithm for learning in piecewise-linear neural networks. *Neural Networks*, 13(4-5):485–505, 2000.

[17] P. Julián, M. Jordán, 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.

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

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

[20] E. A. Heredia and G. R. Arce. Piecewise linear system modeling based on a continuous threshold decomposition. *IEEE Transactions on Signal Processing*, 44(6):1440–1453, June 1996.

[21] D. R. Hush and B. Horne. Efficient algorithms for function approximation with piecewise linear sigmoidal networks. *IEEE Trans. Neural Networks*, 9(6):1129–1141, November 1998.

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

[23] A. Skeppstedt, L. Ljung, and M. Millnert. Construction of composite models from observed data. *Int. J. Control*, 55(1):141–152, 1992.

[24] R. Murray-Smith and T. A. Johansen, editors. *Multiple Model Approaches to Modelling and Control*. Taylor & Francis, 1997.

[25] 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.

[26] J.-E. Strömberg, F. Gustafsson, and L. Ljung. Trees as black-box model structures for dynamical systems. In *Proc. European Control Conf.*, pages 1175–1180, Grenoble, France, July 1991.

[27] C.-H. Choi and J. Y. Choi. Constructive neural networks with piecewise interpolation capabilities for function approximations. *IEEE Trans. Neural Networks*, 5(6):936–944, November 1994.

[28] 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.

[29] 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.

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

[31] J. Roll. Robust verification and identification of piecewise affine systems. Licentiate Thesis, Dept. of Electrical Engineering, Linköping Univ., 2001. `http://www.control.isy.liu.se/publications/`.

[32] A. Bemporad, J. Roll, and L. Ljung. Identification of hybrid systems via mixed-integer programming. In *Proc. 40th IEEE Conf. on Decision and Control*, pages 786–792, December 2001.

[33] A. Bemporad. Identification of hybrid systems: Global convergence via mixed-integer programming. Technical Report AUT00-28, ETH, Zürich, September 2000.

[34] A. Bemporad, J. Roll, and L. Ljung. Identification of hybrid systems via mixed-integer programming. Technical Report AUT00-29, ETH, Zürich, October 2000.

[35] Dash Associates. *XPRESS-MP User Guide*, 1999. `http://www.dashopt.com`.

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

[37] A. Bemporad and D. Mignone. *MIQP.M: A Matlab function for solving mixed integer quadratic programs*. ETH Zürich, 2000. Code available at `http://control.ethz.ch/~hybrid/miqp`.

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

[39] F. Gustafsson. *Adaptive Filtering and Change Detection*. John Wiley & Sons, 2000.

[40] J.-N. Lin and R. Unbehauen. Canonical piecewise-linear approximations. *IEEE Trans. Circuits Syst. I*, 39(8):697–699, August 1992.

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