A continuous optimization framework for hybrid system identification

Fabien Lauer, Gérard Bloch, René Vidal

A continuous optimization framework for hybrid system identification

Fabien Lauer [a] Gérard Bloch [b] René Vidal [c]

[a] *LORIA, Université Henri Poincaré Nancy 1, France*

[b] *Centre de Recherche en Automatique de Nancy (CRAN UMR 7039), Nancy–University, CNRS, France*

[c] *Center for Imaging Science, Department of Biomedical Engineering, Johns Hopkins University, USA*

Abstract

We propose a new framework for hybrid system identification, which relies on continuous optimization. This framework is based on the minimization of a cost function that can be chosen as either the minimum or the product of loss functions. The former is inspired by traditional estimation methods, while the latter is inspired by recent algebraic and support vector regression approaches to hybrid system identification. In both cases, the identification problem is recast as a continuous optimization program involving only the real parameters of the model as variables, thus avoiding the use of discrete optimization. This program can be solved efficiently by using standard optimization methods even for very large data sets. In addition, the proposed framework easily incorporates robustness to different kinds of outliers through the choice of the loss function.

Key words: hybrid system; identification; robustness to outliers; large-scale.

1 Introduction

Consider a class of discrete-time ARX hybrid systems of the form

$$y_i = f_{\lambda_i}(\boldsymbol{x}_i) + v_i, \tag{1}$$

where $\boldsymbol{x}_i = [y_{i-1} \ \dots \ y_{i-n_a}, \ u_{i-n_k} \ \dots \ u_{i-n_k-n_c+1}]^T$ is the *continuous state* (or regression vector) of dimension p containing the lagged n_c inputs, u_{i-k}, and n_a outputs, y_{i-k}, $\lambda_i \in \{1, \dots, n\}$ is the *discrete state* (or mode) determining which one of the n subsystems, $\{f_j\}_{j=1}^n$, is active at time step i, and v_i is an additive noise term. Two classes of hybrid models can be distinguished on the basis of the evolution of the discrete state λ_i. In particular, Switched ARX (SARX) models assume that the system switches arbitrarily, while PieceWise ARX (PWARX) models consider a dependency between the discrete state and the regression vector. The latter are usually defined by piecewise affine maps of the type $f(\boldsymbol{x}) = f_j(\boldsymbol{x})$, if $\boldsymbol{x} \in S_j, \ j = 1, \dots, n$, where $\{f_j\}$ are affine functions and $\{S_j\}$ are polyhedral domains defining a partition of the regression space $\mathbb{R}^p$.

This paper concentrates on the problem of finding a hybrid model $f = \{f_j\}_{j=1}^n$ of the form (1) from input–output data $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$. We assume that the number of models n and their orders are known and focus on the identification of SARX models. However, the proposed estimators are able to deal with PWARX models without any modification. They provide an estimate of the parameters of an SARX model. Then, determining the partition of the regression space amounts to a pattern recognition problem [2], where the labeling of the points is given by the estimated discrete state.

Related work. One of the main challenges in hybrid system identification is that one needs to simultaneously classify the samples into their respective modes and estimate the model parameters for each mode. A general and straightforward approach to this problem is to optimize over both the model parameters and a set of binary variables, that control the assignment of the samples to the modes. This, however, amounts to a nontrivial and non-continuous optimization problem.

Two main classes of methods have been proposed to solve directly this problem. The first class includes the

⋆ A preliminary version of this paper was presented at the 15th IFAC symposium on system identification, Saint-Malo, France, July 6-8, 2009.

⋆⋆This work was partially supported by ANR project ArHyCo, Programme "Systèmes Embarqués et Grandes Infrastructures" - ARPEGE, ANR-2008 SEGI 004 01-30011459, and by the grant NSF CNS 0931805.

clustering-based approach, using either k-means [3] or Expectation Maximization (EM) [11], and the Bayesian approach [5]. These methods alternate between solving the classification problem for fixed model parameters and solving the estimation problem for a fixed classification. Hence they are prone to lead to local minima and are sensitive to initialization. On the other hand, methods in the second class optimize over both continuous and discrete variables simultaneously. This involves solving a combinatorial optimization problem, which can be prohibitively time consuming. The mixed integer programming (MIP) approach [12] and the bounded-error approach [1] fall into this category.

Beside these methods, the algebraic approach [14,10,13] circumvents the aforementioned issues thanks to a continuous approximation of the general optimization problem. This approximation results in a closed form solution to the identification of SARX systems. However, the algebraic method is rather sensitive to noise compared to the other approaches. Inspired by the algebraic approach, the Support Vector Regression (SVR) approach [6,7] provides a convenient way of dealing with noisy data and small sample sizes by incorporating regularization into the optimization framework. However, it optimizes over a number of variables that grows with the number of data points, thus it is limited to small data sets. We conclude this short review of the related methods by noting that, other than the bounded-error approach [1], none of them explicitly deals with outliers in the data.

Paper contribution. We propose a *continuous optimization* framework for hybrid system identification that benefits from the advantages of the algebraic and SVR approaches, while also tackling their respective weaknesses. By continuous optimization we refer to the optimization of a continuous cost function over a non-discontinuous domain, which excludes for instance integer programs. In particular, two reformulations of the mixed integer program at the core of hybrid system identification are considered. The first one is based on a non-differentiable cost function involving min operations. The second one offers a differentiable approximation using products of error terms, as in the algebraic method. These reformulations give rise to the following contributions.

- The proposed framework can include any suitable loss function. Thus robustness to outliers and leverage points is easily incorporated through the choice of a robust loss function as defined in the analysis. In addition, the derivation of the method in the maximum likelihood framework allows the loss function to be chosen with respect to the noise density model as in standard estimation theory.
- This paper proposes a reformulation of the hybrid system identification problem as an unconstrained optimization program. Though non-convex, this nonlinear program involves a low number of variables, equal to the number of parameters in the hybrid model, which allows its complexity to scale only linearly with the number of data. Thus, the problem can be solved efficiently for any number of data by standard global optimization algorithms.

Paper organization. The paper starts in §2 by presenting the hybrid system identification problem and the main inherent issues. §3 then details the proposed approach in both the maximum likelihood (§3.1) and the error minimization (§3.2) frameworks, before describing the product-of-errors based approximation (§3.3) and analyzing the robustness to outliers (§3.4). Optimization issues are discussed in §3.5, while examples can be found in §4 and conclusions in §5.

2 General formulation of the problem

One of the main difficulties in hybrid system identification is that it involves optimization over both discrete and continuous variables. To see this, notice that one can write the problem as the mixed integer program

$$\begin{aligned} \underset{\{f_j\},\{\beta_{ij}\}}{\text{minimize}} \quad & \sum_{i=1}^{N}\sum_{j=1}^{n} \beta_{ij} l(y_i - f_j(\boldsymbol{x}_i)) \qquad (2) \\ \text{s.t.} \quad & \beta_{ij} \in \{0,1\},\ i = 1,\dots,N,\ j = 1,\dots,n, \\ & \sum_{j=1}^{n} \beta_{ij} = 1,\ i = 1,\dots,N, \end{aligned}$$

where β_{ij} is a binary variable and $l(y_i - f_j(\boldsymbol{x}_i))$ is a suitable loss function, e.g., the squared loss, $l(y_i - f_j(\boldsymbol{x}_i)) = (y_i - f_j(\boldsymbol{x}_i))^2$. The discrete variables β_{ij} encode the assignment of point i to submodel j, while continuous variables encode the parameters of the submodels f_j.

One way to solve this mixed program is to use alternating minimization: i) given the submodels $\{f_j\}$, compute the assignment of points to submodels according to $\beta_{ij} = 1$, if $j = \arg\min_{k=1,\dots,n} l(y_i - f_k(\boldsymbol{x}_i))$, and 0 otherwise; and ii) given the assignments β_{ij}, compute one submodel for each group of points. This approach is effective when the submodels are linear, i.e., $f_j(\boldsymbol{x}_i) = \boldsymbol{\theta}_j^T \boldsymbol{x}_i$, and a convex loss function is used, because the estimation of each submodel is a linear system identification problem. However, this approach is sensitive to initialization. Note that problem (2) can also be solved directly by using mixed integer programming techniques, as proposed in [12] for hinging hyperplane models. These latter optimization techniques can guarantee to find the global minimum, but, due to their high complexity, they can only be used in practice for small data sets.

3 Continuous optimization approach

This section presents the proposed estimators for hybrid systems. In particular, two closely related estimators are

derived in the maximum likelihood (§3.1) and the error minimization (§3.2) frameworks, respectively. In order to remain efficient on large data sets, these estimators are devised so as to lead to continuous optimization programs with a small number of variables, which does not depend on the number of data. The section ends with the description of a smooth approximation to the proposed estimators and an analysis of the robustness to outliers.

3.1 Maximum likelihood framework

Let the random variables $(\boldsymbol{x}, y)$ be described by a joint probability density function (pdf) $p(y, \boldsymbol{x})$. Assume that the conditional pdf $p(y|\boldsymbol{x})$ takes the functional form $p(y|\boldsymbol{x}, f)$, dependent on the model f. Then, the maximum likelihood approach consists in finding the model f that most likely generated the data. For N i.i.d. samples, this is equivalent to maximizing the log-likelihood, $\sum_{i=1}^{N} \ln p(y_i|\boldsymbol{x}_i, f)$, with respect to f.

In the context of hybrid systems, the Maximum Likelihood (ML) estimator assigns each data sample $(\boldsymbol{x}_i, y_i)$ to the most likely submodel f_j, i.e., the one with maximal likelihood of the sample w.r.t. f_j. This leads to

$$\hat{\lambda}_i = \arg \max_{j=1,\dots,n} p(y_i|\boldsymbol{x}_i, f_j). \tag{3}$$

This allows the likelihood of a sample with respect to f to be written as $p(y_i|\boldsymbol{x}_i, f) = p(y_i|\boldsymbol{x}_i, f_{\hat{\lambda}_i}) \propto \max_{j=1,\dots,n} p(y_i|\boldsymbol{x}_i, f_j)$. The log-likelihood is thus given by

$$\sum_{i=1}^{N} \ln p(y_i|\boldsymbol{x}_i, f) \propto \sum_{i=1}^{N} \ln \max_{j=1,\dots,n} p(y_i|\boldsymbol{x}_i, f_j), \tag{4}$$

which is equivalently maximized by solving

$$\underset{\{f_j\}}{\text{maximize}} \sum_{i=1}^{N} \max_{j=1,\dots,n} \ln p(y_i|\boldsymbol{x}_i, f_j). \tag{5}$$

Finally, the ML estimator is given as the solution to

$$\underset{\{f_j\}}{\text{minimize}} \; J^{ML} = \sum_{i=1}^{N} \left(\min_{j=1,\dots,n} - \ln p(y_i|\boldsymbol{x}_i, f_j) \right). \tag{6}$$

Note that the minimum of a finite set of continuous functions of some variables is a continuous function of these variables (discontinuities only occur in the derivatives). Therefore, if the submodels $\{f_j\}$ are given by continuous functions of their parameters and if the likelihood function $p(y_i|\boldsymbol{x}_i, f_j)$ is continuous in f_j, then the minimum over j of the negative log-likelihood functions $-\ln p(y_i|\boldsymbol{x}_i, f_j)$ in (6) is a continuous function of the parameters to be estimated. As a consequence, the cost function in (6) is a continuous function of the variables parametrizing the submodels $\{f_j\}$ and (6) is a continuous optimization problem.

3.2 Minimum-of-errors estimator

In the following, we derive the *Minimum-of-Errors* (ME) estimator in the framework of loss function minimization and show its relationship to the ML estimator.

The formulation in (2) relies on the assumption that sample $\boldsymbol{x}_i$ must be assigned to the submodel that best estimates the target output y_i. In order to minimize the overall cost, we thus have to set the estimated mode as

$$\hat{\lambda}_i = \arg \min_{j=1,\dots,n} l(y_i - f_j(\boldsymbol{x}_i)), \; i = 1, \dots, N. \tag{7}$$

Explicitly including this result in (2) leads to the ME estimator obtained by solving

$$\underset{\{f_j\}}{\text{minimize}} \; J^{ME} = \sum_{i=1}^{N} \left(\min_{j=1,\dots,n} l(y_i - f_j(\boldsymbol{x}_i)) \right). \tag{8}$$

The relationship between ML and ME estimators appears when choosing the loss function l in the ME estimator (8) according to $l(y_i - f_j(\boldsymbol{x}_i)) = -\ln p(y_i|\boldsymbol{x}_i, f_j)$. This provides the choice of the loss function in the case of a known noise distribution.

Using a similar continuity argument for the cost in (8) as for that in (6), we see that, for all loss functions $l(e)$ that are continuous in their argument, (8) is a continuous optimization problem with respect to the parameters of the submodels $\{f_j\}$. Thus instead of solving (2), which is a non-continuous problem due to the discrete variables $\{\beta_{ij}\}$, we can equivalently solve the continuous problem (8), which only involves real variables. After solving for the submodels $\{f_j\}$, the mode estimates are simply recovered by using (7) (or $\hat{\lambda}_i = \arg\min_{j=1,\dots,n} |y_i - f_j(\boldsymbol{x}_i)|$, if the loss function l cannot yield the decision).

3.3 Product-of-errors estimator

For a smooth loss function l, the *Product-of-Errors* (PE) estimator is obtained by solving the smooth optimization program

$$\underset{\{f_j\}}{\text{minimize}} \; J^{PE} = \sum_{i=1}^{N} \prod_{j=1}^{n} l(y_i - f_j(\boldsymbol{x}_i)). \tag{9}$$

The cost function of the PE estimator in (9) can be seen as a smooth approximation to the ME cost function in (8). In particular, for noiseless data, they share the same global minimum $J^{ME} = J^{PE} = 0$.

Remark 1: *Note that for the particular case of linear submodels f_j and squared loss function $l(e) = e^2$, the cost function of the PE estimator in* (9) *coincides with the cost minimized by the algebraic approach [10]. However, the PE estimator aims at directly minimizing this cost with respect to the model parameters, while the algebraic approach minimizes this cost with respect to the tensor product of the parameters. This allows the algebraic algorithm to obtain the solution more efficiently by solving a linear system, but introduces errors in the case of noisy data, because the tensor product of the parameters is an overparametrization of the space of parameters.*

Remark 2: *The PE estimator given by* (9) *is also equivalent to the (unregularized) SVR-based approach described in [6] when using the ε-insensitive loss function defined as $l(e) = \max(0, |e| - \varepsilon)$, for a threshold ε. In this case, the proposed method allows to reduce the number of optimization variables from $n(p + N)$ to np.*

3.4 Robustness to outliers and leverage points

Two types of outlying observations, i.e., points which deviate significantly from the rest of the data, are commonly distinguished. Arbitrary large values $|y_i|$ in the response variable are simply called outliers, while arbitrary large values in the regression vector $\boldsymbol{x}_i$ are called leverage points. It is worth noting that, for the estimation of ARX model parameters, due to the presence of lagged outputs in the regression vector, the same abnormal value can be both an outlier and a leverage point.

For an estimator to be robust to outliers, the effect of a single point on the estimation must be bounded. For instance, in classical regression problems, the influence function of the squared loss, i.e., its derivative with respect to y_i, is unbounded. On the other hand, the absolute loss, $l(e) = |e|$, has an influence function bounded by 1 and thus leads to more robust estimators.

In this subsection, the different cases are investigated for the ME and PE estimators in the context of hybrid systems. $l'(e) = dl(e)/de$ denotes the derivative of the loss function l with respect to its scalar argument e.

Minimum-of-errors estimator. For a given $\boldsymbol{x}_i$ with an arbitrary large $|y_i|$, we can consider the estimated mode as fixed (either given by $\lambda_i = \arg\max_j f_j(\boldsymbol{x}_i)$ or $\lambda_i = \arg\min_j f_j(\boldsymbol{x}_i)$, depending on the sign of y_i). Then we can write the influence of this point as

$$\frac{\partial J^{ME}}{\partial y_i} = \frac{\partial l(y_i - f_{\lambda_i}(\boldsymbol{x}_i))}{\partial y_i} = l'(y_i - f_{\lambda_i}(\boldsymbol{x}_i)). \quad (10)$$

This implies that the influence of an outlier is bounded if l has a bounded derivative $l'(e)$, as is the case with $l(e) = |e|$. On the other hand, the influence of leverage points

$$\frac{\partial J^{ME}}{\partial \boldsymbol{x}_i} = -l'(y_i - f_{\lambda_i}(\boldsymbol{x}_i))\frac{df_{\lambda_i}(\boldsymbol{x}_i)}{d\boldsymbol{x}_i}, \quad (11)$$

with $\lambda_i = \arg\min_{j=1,\ldots,n} l(y_i - f_j(\boldsymbol{x}_i))$, is unbounded even with bounded derivative $l'(e)$.

Product-of-errors estimator. The sensitivity of the cost function in (9) with respect to outliers is

$$\frac{\partial J^{PE}}{\partial y_i} = \sum_{j=1}^{n} l'(y_i - f_j(\boldsymbol{x}_i)) \prod_{k\in\{1,\ldots,n\}\setminus j} l(y_i - f_k(\boldsymbol{x}_i)), \quad (12)$$

and its sensitivity with respect to leverage points is

$$\frac{\partial J^{PE}}{\partial \boldsymbol{x}_i} = -\sum_{j=1}^{n} l'(y_i - f_j(\boldsymbol{x}_i))\frac{df_j(\boldsymbol{x}_i)}{d\boldsymbol{x}_i} \prod_{k\in\{1,\ldots,n\}\setminus j} l(y_i - f_k(\boldsymbol{x}_i)). \quad (13)$$

In these cases, the terms $l(y_i - f_k(\boldsymbol{x}_i))$ are unbounded, hence both sensitivity functions are unbounded.

To summarize, the only way to guarantee that the influence of outliers and leverage points on both the ME and PE estimators is bounded and obtain robust estimators is to use a loss function l leading to $l'(e) = 0$, for large values of the error $|e|$.

Such loss functions are considered in the framework of redescending M-estimators, where $l'(e)$ is known as the Ψ function and decreases smoothly towards zero when $|e|$ increases. As an example, the Hampel's loss function, defined as

$$l(e) = \begin{cases} \delta^2/\pi\,(1 - \cos(\pi e/\delta)), & \text{if } |e| \le \delta, \\ 2\delta^2/\pi, & \text{otherwise,} \end{cases} \quad (14)$$

which satisfies that $l'(e) = 0$ for $|e| > \delta$, will be considered in §4.1. Further experiments with other loss functions and the PE estimator can be found in [9].

3.5 Optimization of linear hybrid models

All the results developed in the previous sections apply similarly to linear and nonlinear submodels f_j. In the following, we focus on linear hybrid models, in which the submodels $f_j(\boldsymbol{x})$ are given in the linear form

$$f_j(\boldsymbol{x}) = \boldsymbol{\theta}_j^T \boldsymbol{x}, \; j = 1, \ldots, n, \quad (15)$$

where the parameter vectors $\boldsymbol{\theta}_j$ to be estimated are of dimension $p = n_a + n_c$. Note that affine submodels can be equivalently considered by appending a 1 to the regression vector and considering $p = n_a + n_c + 1$.

For all continuous loss functions l, the cost functions in (8) and (9) are both continuous functions of the parameters $\boldsymbol{\theta}_j$. Moreover, the number of variables involved in these problems is small and fixed to the number of model parameters, $n \times p$, for any number of data N. These two combined features allow a solver for continuous problems to find a satisfying solution in reasonable time, despite the NP-hard nature of the problem. Note however that the cost functions in (8) and (9) require to compute a sum over N terms, hence the linear complexity of the algorithm with respect to N.

In the following, we propose to solve (8) and (9) with the Multilevel Coordinate Search (MCS) algorithm [1] [4], that is guaranteed to converge if the objective is continuous in the neighborhood of the global minimizer. This optimizer uses only function values (when required, derivatives are estimated from these) and alternates between global and local search. The local search, done via sequential quadratic programming, speeds up the convergence once the global part has found a point in the bassin of attraction of the global minimizer.

4 Examples

We now present some illustrative examples and start with the identification of a switched linear system (Sect. 4.1), including a comparison with the algebraic procedure and the study of the robustness to outliers of the estimators. Large-scale experiments (Sect. 4.2) are then presented to analyze the complexity of the method with respect to the number of data and the number of parameters in the model.

Assuming no prior knowledge on the parameters, box constraints that limit the search space in the MCS algorithm are set for all variables θ_{jk} to the quite large interval $-100 \leq \theta_{jk} \leq 100$. Beside this, the MCS parameters are all of time-limiting nature (maximum number of iterations or function evaluations), for which the default values led to satisfying results. For all the problems, N samples are generated by

$$y_i = \boldsymbol{\theta}_{\lambda_i}^T \boldsymbol{x}_i + v_i, \quad i = 1, \ldots, N, \tag{16}$$

where the $\boldsymbol{\theta}_j \in \mathbb{R}^p$ are the true parameters to be recovered and $v_i \sim \mathcal{N}(0, \sigma_v^2)$ is a Gaussian noise. The methods are compared on the basis of the Normalized Mean Squared Error *on the parameters*, NMSE $= \sum_{j=1}^n \|\boldsymbol{\theta}_j - \hat{\boldsymbol{\theta}}_j\|_2^2 / \|\boldsymbol{\theta}_j\|_2^2$, where the $\hat{\boldsymbol{\theta}}_j$ are the estimated parameters. In the Tables, all numbers of the form $A \pm B$ correspond to averages (A) and standard deviations (B) over 100 trials. The Tables also show the number of failures (# fail) of the algorithms, i.e., the number of trials for which the parameter estimates are irrelevant. It is worth noting that the NMSE is computed without taking these trials into account. All experiments are performed on a standard desktop computer with Matlab.

[1] http://www.mat.univie.ac.at/~neum/software/mcs/.

4.1 Switched linear system identification

Consider the example taken from [13]. The aim is to recover, from $N = 1000$ samples, the parameters $\boldsymbol{\theta}_1 = [0.9, 1]^T$ and $\boldsymbol{\theta}_2 = [1, -1]^T$ of a dynamical system, arbitrarily switching between $n = 2$ modes, with continuous state $\boldsymbol{x}_i = [y_{i-1}, u_{i-1}]^T$ and input $u_i \sim \mathcal{N}(0, 1)$. The standard deviation of the generated trajectories is $\sigma_y \approx 2$.

Table 1
Average NMSE ($\times 10^{-3}$) and number of failures over 100 trials.

σ_v		ME	PE	Algebraic
0.00	NMSE	0.00	0.00	0.00
	# fail	0	0	0
0.02	NMSE	0.00 ± 0.00	0.00 ± 0.00	0.05 ± 0.28
	# fail	44	0	0
0.10	NMSE	0.03 ± 0.03	0.08 ± 0.06	2.99 ± 12.32
	# fail	20	0	4
0.20	NMSE	0.09 ± 0.10	0.27 ± 0.23	4.86 ± 10.38
	# fail	28	0	10
0.30	NMSE	0.27 ± 0.22	0.77 ± 0.65	10.31 ± 13.46
	# fail	27	0	10

Comparison with the algebraic approach. As described in [14], the algebraic method is originally implemented as a linear problem solved w.r.t. the tensor product of the parameters. It can also be implemented in the proposed PE framework (9), with squared loss, as the direct minimization over the model parameters (see Remark 1). These two algorithms are compared in Table 1 for different noise levels σ_v. The ME estimator (8) with squared loss is also included in the comparison. The results highlight the gain in solving the problem directly for the model parameters $\boldsymbol{\theta}_j$ versus the optimization over the product of the parameters. However, the gain in NMSE obtained by solving directly for the $\boldsymbol{\theta}_j$ comes at the cost of solving a nonlinear optimization program instead of a linear problem. This leads to a computing time about 30 times larger with the PE estimator (0.3 sec.) than with the algebraic method (0.01 sec.). Note however, that for $N = 1000$ points these times remain below one second, whereas directly solving (2) is already prohibitively time consuming.

In practice, the algebraic method estimates the normals $\boldsymbol{b}_j$ to the hyperplanes on which the data lie as $[\boldsymbol{x}_i, -y_i]\boldsymbol{b}_j = 0$, and assigns a coefficient to the output y_i. Thus, the original model parameters are recovered by dividing all the coefficients in $\boldsymbol{b}_j$ by the last component of $\boldsymbol{b}_j$, which leads to failure when this number is close

Table 2
Average NMSE ($\times 10^{-3}$) over 100 trials for the proposed estimators using either the squared loss or the Hampel's (*robust*) loss function when applied to data sets with additive Gaussian noise ($\sigma_v = 0.2$) and an increasing percentage of outliers.

% of outliers		0 %	10 %	20 %	30 %	40 %	50 %
Robust ME	NMSE	0.14 ± 0.12	0.14 ± 0.14	0.27 ± 0.22	0.22 ± 0.21	0.43 ± 0.44	0.71 ± 0.64
	# fail	10	11	15	19	21	27
ME	NMSE	0.09 ± 0.10	40.3 ± 43.8	118.8 ± 47.7	280.6 ± 72.3	379.6 ± 103.6	496.2 ± 149.3
	# fail	28	15	13	22	11	12
Robust PE	NMSE	0.15 ± 0.12	0.15 ± 0.11	0.22 ± 0.21	0.26 ± 0.24	0.43 ± 0.39	0.84 ± 0.89
	# fail	0	0	0	0	0	6
PE	NMSE	0.27 ± 0.23	$118.0 \pm 0.83.4$	251.7 ± 139.4	215.5 ± 96.3	425.2 ± 177.7	501.3 ± 210.0
	# fail	0	0	0	0	4	14

to zero. For the ME estimator, the failures are due to the convergence of the optimizer to bad local minima. However, failures aside, the ME estimator leads to an NMSE that is similar to the one obtained by separate least squares estimations for each mode with knowledge of the true mode λ_i. This indicates that good local, if not global, minima are found. Note that in these experiments, the PE estimator does not suffer from the convergence to bad local minima, but leads to slightly less accurate estimates.

Robustness to outliers. We now illustrate the robustness to outliers of the ME (8) and PE (9) estimators obtained by using the Hampel's loss function (14) instead of the squared loss. The data are corrupted with additive Gaussian noise ($\sigma_v = 0.2$) and an increasing percentage of outliers by forcing y_i, at random time steps i, to take uniformly distributed random values in the interval $[-10, 10]$. Note that this also introduces outliers in the regressors $\boldsymbol{x}_i$, which are built from lagged outputs. According to the previous results shown in Table 1, the algebraic method cannot handle these highly corrupted data. Table 2 shows the benefit of using the robust versions of the proposed estimators over their standard counterpart. The resulting NMSE remain in the order of 10^{-4} to 10^{-3} with up to 50% of outliers for both the Robust ME and the Robust PE estimators, while being comparable to the NMSE obtained with the non-robust squared loss in the outlier-free case. However, the number of failures of the Robust ME estimator slightly increases with the number of outliers. Again, the Robust PE estimator is not susceptible to these failures even in the presence of outliers, except for the extreme case of 50 % of outliers. In these experiments, the parameter of the Hampel's loss function was set to $\delta = 2$. The estimation of the optimal value for δ is left to future work.

4.2 Large-scale experiments

Large data sets. The performance of the method on large data sets is evaluated on a set of 100 randomly generated problems with $n = 2$ and $p = 3$. The true parameters $\{\boldsymbol{\theta}_j\}$ are randomly drawn from a uniform distribution in the interval $[-2, 2]^p$, while very loose constraints, $\boldsymbol{\theta}_j \in [-100, 100]^p$, are applied for the estimation. The N data are generated by (16) with uniformly distributed random regression vectors $\boldsymbol{x}_i \in [-1, 1]^p$, a random switching sequence $\{\lambda_i\}$ and additive Gaussian noise ($\sigma_v = 0.2$). Due to the noise level in the data, the algebraic method is not included in the comparison, which focuses on the ME (8) and PE (9) estimators with squared loss. Table 3 shows the resulting average NMSE and computing times for an increasing number of data N. These times show that the complexity of the proposed algorithms scales linearly with the number of data. As a result, the method can be applied to very large data sets, which cannot be handled by previous approaches such as the ones described in [1,3,6,12]. Note that the programs, including the MCS optimization algorithm, are fully implemented in non-compiled Matlab code and that the variability over the 100 runs comes from the random sampling of the true parameters generating the data, not from the optimizer. However, in 5 runs out of 400 (over all experiments), the optimizer led to local minima for the ME estimator, which failed to yield a correct model. Note that such failures do not occur when using the PE estimator. Overall, the ME estimator leads to models with less error than the PE estimator, but requires twice as much time to compute and can occasionally lead to bad local minima.

Larger model structures. The method has been shown to be very effective on large data sets. However, the computing time heavily relies on the number of model parameters $n \times p$, against which the method is now tested. As before, a set of 100 random experiments is performed for $N = 10\,000$ samples and varying numbers of modes n and parameters per mode p. To be able to compare the computing times of the proposed PE estimator with those of the algebraic method, experiments without noise ($\sigma_v = 0$), for which both methods perfectly estimate the parameters, are considered. The results appear in Figure 1 for the PE estimator (9) with

Table 3
Average NMSE and computing time over 100 randomly generated problems with N samples.

N		NMSE ($\times 10^{-6}$)	#fail	Time (sec.)
10 000	ME	$\mathbf{1.3 \pm 0.9}$	0	2.7 ± 1.0
	PE	3.0 ± 2.3	0	1.3 ± 0.2
50 000	ME	$\mathbf{0.4 \pm 0.5}$	0	10.0 ± 4.0
	PE	1.1 ± 2.8	0	4.2 ± 0.6
100 000	ME	$\mathbf{0.3 \pm 0.2}$	1	23.4 ± 10.7
	PE	1.0 ± 1.8	0	10.4 ± 1.7
500 000	ME	$\mathbf{0.3 \pm 0.8}$	4	158.5 ± 49.2
	PE	0.7 ± 1.6	0	53.9 ± 6.1

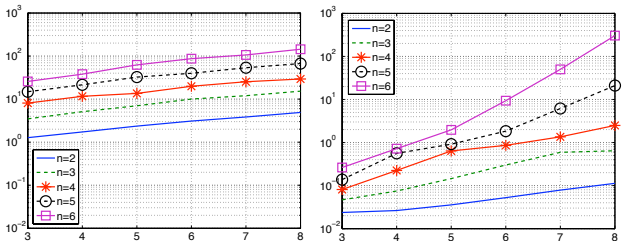

Fig. 1. Average computing time in seconds for the PE estimator (left) and the algebraic method (right) over 100 trials versus the number of parameters per mode p for different number of modes n.

squared loss and the algebraic method. These curves show that the PE estimator is clearly slower than the algebraic method for small model structures. However, the computing time of the algebraic method increases quickly with the number of parameters and exceeds the one of the PE estimator for $n = 6$ and $p = 8$. In addition, similar experiments with noisy data lead to similar time curves for the PE estimator. Thus the cost in computing time of the proposed method is still reasonable considering the gain in accuracy obtained by the PE estimator over the algebraic method in the noisy case (as shown by Table 1).

5 Conclusion

Two classes of estimators for hybrid systems have been proposed: the ME estimator, based on the minimum of the submodel errors, and the PE estimator, based on the product of the submodel errors. The ME estimator benefits from a straightforward maximum likelihood interpretation and is an exact reformulation of the mixed integer problem (2) into a continuous optimization problem. The PE estimator has been devised as a smooth approximation to the ME estimator, which appears as a convenient substitute in terms of avoiding local minima while maintaining a similar level of accuracy. In terms of efficiency, the proposed estimators have been tailored to tackle large-scale problems and have been shown to yield fast and accurate results in experiments with numerous data. The paper also focused on providing an algorithm which remains robust to high levels of noise compared to previous approaches such as [14]. In addition, the analysis shows that robustness to outliers either in the output y_i or the regression vector $\boldsymbol{x}_i$ can be obtained by a proper choice of the loss function. Future work will aim at finding better approximations to the ME estimator than the PE estimator and extend the approach to the estimation of unknown nonlinearities in the context of hybrid systems, as initiated in [8].

References

[1] A. Bemporad, A. Garulli, S. Paoletti, and A. Vicino. A bounded-error approach to piecewise affine system identification. *IEEE Trans. on Automatic Control*, 50(10):1567–1580, 2005.

[2] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley-Interscience, 2000.

[3] G. Ferrari-Trecate, M. Muselli, D. Liberati, and M. Morari. A clustering technique for the identification of piecewise affine systems. *Automatica*, 39(2):205–217, 2003.

[4] W. Huyer and A. Neumaier. Global optimization by multilevel coordinate search. *Journal of Global Optimization*, 14(4):331–355, 1999.

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

[6] F. Lauer and G. Bloch. A new hybrid system identification algorithm with automatic tuning. In *Proc. of the 17th IFAC world congress, Seoul, Korea*, pages 10207–10212, 2008.

[7] F. Lauer and G. Bloch. Switched and piecewise nonlinear hybrid system identification. In *Proc. of the 11th int. conf. on hybrid systems: computation and control, St. Louis, MO, USA*, volume 4981 of *LNCS*, pages 330–343, 2008.

[8] F. Lauer, G. Bloch, and R. Vidal. Nonlinear hybrid system identification with kernel models. In: *Proc. of the 49th IEEE int. conf. on decision and control, Atlanta, GA, USA*, 2010.

[9] F. Lauer, R. Vidal, and G. Bloch. A product-of-errors framework for linear hybrid system identification. In *Proc. of the 15th IFAC symp. on system identification, Saint-Malo, France*, 2009.

[10] Y. Ma and R. Vidal. Identification of deterministic switched ARX systems via identification of algebraic varieties. In *Proc. of the 8th int. conf. on hybrid systems: computation and control, Zurich, Switzerland*, volume 3414 of *LNCS*, pages 449–465, 2005.

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

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

[13] R. Vidal. Recursive identification of switched ARX systems. *Automatica*, 44(9):2274–2287, 2008.

[14] R. Vidal, S. Soatto, Y. Ma, and S. Sastry. An algebraic geometric approach to the identification of a class of linear hybrid systems. In *Proc. of the 42nd IEEE conf. on decision and control, Maui, Hawaï, USA*, pages 167–172, 2003.