Mario Vašak, Nedjeljko Perić

Combining Identification and Constrained Optimal Control of Piecewise Affine Systems

Original scientific paper

In this paper we focus on identification and time optimal control of nonlinear processes modeled as piecewise affine systems. We combine the piecewise ARX process model identification based on clustering and the constrained time optimal controller design for discrete-time piecewise affine systems. The two procedures are improved and bound into a systematic procedure for design of high-performance nonlinear control systems: from the identification data to the closed-form time optimal controller. We successfully experimentally verify the procedure on the electronic throttle control system case study.

Key words: piecewise affine model, model identification, clustering, constrained time optimal control, electronic throttle

1 INTRODUCTION

Constrained optimal control methods offer a way of systematic design of high-quality control systems for demanding applications. The control system respects the given constraints on its variables and achieves performance that is optimal with respect to the design objective. In most cases the optimal control designs require a mathematical model of the controlled process. While the constrained optimal control with linear process models nowadays becomes classic, its extension to general nonlinear models is still unresolved.

As one of the first promising lunges in the area of nonlinear model based optimal control, in this decade evolved discrete-time linear hybrid systems based optimal control [1]. Discrete-time linear hybrid systems are basically systems that have both continuous- and discrete-valued variables. They consist of several linear differential equations and of logic rules resolving which equation is currently valid for model evolution. As such, they are used to model a wide class of dynamical systems. Especially useful for optimal control is the Discrete-Time Piecewise Affine (DTPWA) class of hybrid systems. DTPWA systems comprise several affine equations for state-updates, each valid in a separate polyhedral set of the state-input space. They are under very mild assumptions equivalent to many other classes of hybrid systems [2] and are very useful since they can approximate any nonlinear system.

In this paper we focus on explicit state-feedback reference-tracking controller design. The term »explicit« means that such design yields the optimal controller in a closed-form, i.e. as an explicit function of the state measurements, output references and past control inputs. We use Constrained Time Optimal Control (CTOC) problem formulation [3] for DTPWA systems where the objective to be minimized is the time needed for the system to confine to certain invariant state-space target set. A set of system states is called invariant if for any state in the set one can find a control action that keeps the state trajectory in the set. The optimal control law by CTOC for DTPWA systems is in most cases non-unique, but it can always be selected in the PWA, i.e. in the lookup-table form [3].

About 90 % of the optimal control system design time is spent on obtaining the representative mathematical model of the process [4]. Two ways of obtaining the DTPWA process model are: (i) analytical derivation from the first-principles (physical laws of conservation) and (ii) direct model identification.

The analytical way is suitable if the model can be directly from the first-principles derived in the form of several linear differential equations and logic rules that resolve which equation is valid. If this is not the case, difficulties occur in parameterizing the nonlinear model itself and linearizing it. A very hard issue is the choice of linearization points for the PWA model design and additionally,

in case of continuous-time models, discretization could generate unwanted effects in the resulting DTPWA model (see e.g. the case of friction non-linearity in [5]). Such model design induces a lot of unwanted degrees of freedom that, in most cases, deteriorate the model approximation abilities.

An attractive alternative to analytical PWA model design is the model identification in PieceWise ARX (PWARX) form [6]. Basically, PWARX model maps regression vectors, formed of the past and present process inputs and outputs, to the predicted outputs using one of several ARX models. Identification of the PWARX model is a hard task since one has to simultaneously estimate the coefficients of each ARX submodel and the polyhedral regions in the regression vectors space where those submodels are valid. Several procedures were proposed in the last few years for this task [7, 6]. When the a priori knowledge on the process is limited and the measured outputs are corrupted with noise, one of the most appropriate methods for PWARX process model identification is the clustering-based procedure introduced in [8]. A local ARX model is identified around each regression vector and the obtained model parameters are clustered by similarity in the pre-fixed number of clusters (affine models). The final parameters in each cluster are obtained by utilizing the weighted mean. The separating hyperplanes between the clusters in the regressor space are finally computed using some of linear classification techniques which yields the model polyhedral partition.

This paper binds the clustering-based PWARX model identification technique [8] and CTOC for DTPWA systems [3], with improvements presented in [9], to give a unified approach to explicit optimal controller design for piecewise affine systems. Two important modifications are introduced in the standard clustering-based PWARX model identification [8]: (i) systematic pre-processing of the identification data and (ii) linear classification of clusters through linear classification of vertices of clusters. DTPWA models that straightforwardly follow from identified PWARX model usually have a complex model partition and are thus problematic in the time-optimal controller computation phase. We introduce important computational modifications in the controller design phase that keep the resulting controller time-optimal, while ensuring its implementability even for complex models. As a case-study, this unified approach is used to design a time-optimal controller for the electronic throttle used in cars with internal combustion engines. Electronic throttle is highly non-linear and as such requires quality control to operate according to the demands of the automotive industry [5].

The paper is divided into five sections. After this introductory section we outline the clustering-based identification procedure with the proposed modifications in Section 2. Then we proceed with the exposition of controller computation in Section 3. The unified identification-control approach using DTPWA process model is applied to the electronic throttle control system design in Section 4. The paper conclusions are given in Section 5.

2 CLUSTERING-BASED IDENTIFICATION OF PIECEWISE ARX MODELS

PWARX models map the regression vector to the next process output using a PWA map. The regression vector $\varphi \in \mathbb{R}^n$ is formed of the observed process data – past and present inputs and outputs:

$$\varphi_k = \begin{bmatrix} y_k & y_{k-1} \cdots y_{k-n_a+1} & u_k^T & u_{k-1}^T \ldots u_{k-n_b+1}^T \end{bmatrix}^T \quad (1)$$

where k denotes the discrete-time instant, $y \in \mathbb{R}$ is the process output and $u \in \mathbb{R}^m$ the process input, n_a and n_b are the numbers of past outputs and inputs used in the regression vector, $n = n_a + mn_b$. Note that we restrict our attention to MISO models. However, the procedures outlined in the sequel can be straightforwardly extended to identify MIMO models [6, 10]. Piecewise affine function f on the regression vector φ is defined as

$$f(\varphi) = \begin{cases} \Theta_1^T \begin{bmatrix} \varphi \\ 1 \end{bmatrix} & \text{if } \varphi \in \Phi_1, \\ \vdots & \\ \Theta_s^T \begin{bmatrix} \varphi \\ 1 \end{bmatrix} & \text{if } \varphi \in \Phi_s, \end{cases} \quad (2)$$

where s is the number of affine terms in f, $\Theta_i \in \mathbb{R}^{m+1}$ is the parameter vector and Φ_i is the polytopic region where i-th affine function is valid. Polytopes $\Phi_i \in \mathbb{R}^n$ form a partition of the polytopic regressor set Φ:

$$\bigcup_{i=1}^{s} \Phi_i = \Phi \subset \mathbb{R}^n, \ \text{int}(\Phi_i) \cap (\Phi_j) = 0 \ \text{for } i \neq j, \quad (3)$$

where Φ is a polytope usually defined by the physical constraints on inputs and outputs. Note that we allow f to be multiple-valued on $\partial \Phi_i \cap \partial \Phi_j$ for $i \neq j$. PWARX model is of the form

$$y_{k+1} = f(\varphi_k) + e_k, \quad (4)$$

where e is the model prediction error.

The goal of the identification procedure is to estimate f based on a given data-set containing N pairs (φ_k, y_{k+1}), with $1 \leq k \leq N$, such that the model prediction error is as small as possible. The unknowns in f are: s, n_a, n_b, Θ_i, Φ_i. Most often the numbers of past inputs and outputs used in the regression vector are fixed in advance [8]. Note that if additionally s is fixed, we have a fixed structure of f. The problem of estimating f may be decomposed in two subproblems: (i) estimation of s and Θ_i and assignment of each data-point to a submodel that most-likely generated it and (ii) partition of the set Φ according to that assignment. While the step (ii) is performed similarly for majority of the existing PWARX model identification techniques, the step (i) is substantially different for all. The most relevant techniques are [6, 7]: clustering--based procedure [8], bounded-error procedure [11], Bayesian procedure [12] and algebraic geometric procedure [13]. In the rest of the paper we consider only the clustering-based identification.

2.1 Clustering-based Data Classification and Parameter Identification

We assume that the number of outputs and inputs in the regression vector φ, n_a and n_b, as well as the number s of affine terms in f are fixed in advance. In practice usually only lower and upper bounds on n_a and n_b can be introduced, based on e.g. outputoutput and output-input correlation functions. It is a common practice in ARX model identification to estimate the models with varying n_a and n_b and choose the one with the best validation performance. In the case of clustering-based PWARX model identification this should be done with caution since PWARX models with over-estimated n_a and n_b may have poor approximation properties [7].

In this section for the clustering-based PWARX model identification procedure [8] we discuss how parameter estimates Θ_i, $i = 1, \ldots, s$, are obtained and how each data-point φ_k, $k = 1, \ldots, N$ is classified into one of s data-clusters $\mathcal{F}_i$. The cluster $\mathcal{F}_i$ collects all the data estimated to be processed using i-th ARX model. The identification is performed on the given data-set

$$\mathcal{S} = \{(\varphi_k, y_{k+1}) \mid 1 \leq k \leq N\}.$$

The procedure relies on the assumption that the model is locally linear around each data-point. Following this, a number c is fixed and around each φ_k $c - 1$ nearest points φ_j are determined and corresponding $(\varphi_j, y_j + 1)$ are gathered with (φ_k, y_{k+1}) in a Local Data-set (LD) $\mathcal{C}_k$ such that:

$$\left\| \varphi_k - \varphi_j \right\|_2 \leq \left\| \varphi_k - \varphi_{j'} \right\|_2, \quad \forall_j, \quad \forall (\varphi_{j'}, y_{j'+1}) \notin \mathcal{C}_k.$$

The indices of data-points in $\mathcal{C}_k$ are denoted with $j_{k,1}, \ldots, j_{k,c}$. Parameter c is the tuning knob of the algorithm and must be properly chosen for good identification results. A short discussion on its choice is provided later in this subsection.

LD may be either such that it gathers points generated by a single ARX model and is then called pure LD or it can gather points generated by several ARX models and is then called mixed LD. The identification relies on the assumption that the number of mixed LDs is small compared to the number of pure LDs.

The next step of the procedure is to identify parameters of a local ARX model for each LD using standard linear regression techniques like Least--Squares (LS):

$$\Theta_k^{\mathrm{LS}} = (\Psi_k^T \Psi)^{-1} \Psi_k^T y_{\mathcal{C}_k},$$

$$\Psi_k = \begin{bmatrix} \varphi_{jk,1} & \varphi_{jk,2} & \ldots & \varphi_{jk,c} \\ 1 & 1 & & 1 \end{bmatrix}^T, \quad (5)$$

$$y_{\mathcal{C}_k} = \begin{bmatrix} y_{jk,1} + 1 & \cdots & y_{jk,c} + 1 \end{bmatrix}.$$

Empirical parameter covariance matrix of $\mathcal{C}_k$ is [14]

$$V_k = \frac{SSR_k}{c-n+1} (\Psi_k^T \Psi_k)^{-1},$$

$$SSR_k = y_{Ck}^T (I - (\Psi_k^T \Psi_k)^{-1} \Psi_k^T) y_{\mathcal{C}_k}, \quad (6)$$

where SSR stands for »sum-squared residual«. The mean regression vector m_k in $\mathcal{C}_k$ together with its covariance (scatter) matrix Q_k are also computed:

$$m_k = \frac{1}{c} \sum_{\varphi \in \mathcal{C}_k} \varphi, \quad Q_k = \sum_{\varphi \in \mathcal{C}_k} (\varphi - m_k)(\varphi - m_k)^T. \quad (7)$$

We assume that the data sampling is fair, i.e. that the data-points are not in majority placed on the facets of Φ_i. Under proper choice of c, matrices V_k and Q_k are inverse proportional to confidence that $\mathcal{C}_k$ is a pure LD which is of utmost importance for the identification algorithm. If $\mathcal{C}_k$ is pure and generated by the i-th submodel, the parameter vector Θ_k^{LS} is the Gaussian random variable with mean close to some Θ_i and the covariance determined dominantly by the characteristics of output measurement noise. On the other hand, if LD is mixed, Θ_k^{LS} contains less confident or even useless information about the model which in principle leads to bigger entries in the parameter covariance matrix V_k. As far as Q_k is in question, dense points

in $\mathcal{C}_k$ are more likely to belong to the same region of the PWARX model. However, note that if Q_k becomes smaller indicating higher points density, the covariance V_k rises. In the limit, when the whole group $\mathcal{C}_k$ degenerates into a single point in the regressor space, entries in V_k tend to infinity.

For the low number of mixed LDs it is beneficial to have c as small as possible, e.g. $c = n + 1$. However, if the output measurement is noise-corrupted, low number of data-points in $\mathcal{C}_k$ may result in poor parameter estimates Θ_k^{LS} that are not accompanied with higher values in V_k and thus not properly »marked« for further identification procedure. The parameter c should be tuned to optimally weigh the two opposing phenomena. Also, it should be noted that c should increase with the number of data-points N since for higher N and fixed c the sets $\mathcal{C}_k$ become denser.

For the purpose of data classification based on clustering, to each LD $\mathcal{C}_k$ a representative, so-called feature vector, is assigned:

$$\xi_k = \begin{bmatrix} \Theta_k^{\mathrm{LS}} \\ m_k \end{bmatrix} \in \mathbb{R}^{2n+1}, \tag{8}$$

$$R_k = \begin{bmatrix} V_k & \mathbf{0}_{(n+1)\times n} \\ \mathbf{0}_{n\times(n+1)} & Q_k \end{bmatrix}. \tag{9}$$

Note that the connection between the point φ_k, LD $\mathcal{C}_k$ and the feature vector ξ_k, $k = 1, \ldots, N$, is bijective. This means that once ξ_k are classified in clusters the classification can be directly applied also to φ_k in order to obtain $\mathcal{F}_i$.

It remains thus to show how feature vectors are classified. For that purpose in [8] authors propose a variation of the K-means clustering algorithm [15] in which feature vectors ξ_k are clustered into s disjoint clusters $\mathcal{D}_i$. The clustering is performed by attaining to minimize the cost

$$J\{\mathcal{D}_i\}_{i=1}^s, \{\mu_i\}_{i=1}^s = \sum_{i=1}^s \sum_{\xi_k \in \mathcal{D}_i} \|\xi_k - \mu_i\|_{R_k^{-1}}^2, \tag{10}$$

where μ_i represents the center of the cluster $\mathcal{D}_i$ and $\|\xi\|_A^2 = \xi^T A \xi$. Since the distance of ξ_k from the cluster center is penalized with R_k^{-1}, less confident feature vectors (stemming from mixed LDs) have smaller influence to the result of clustering. Note that J is non-convex – one of its variables is the classification of ξ_k into clusters $\{\mathcal{D}_i\}$ which is discrete. For a fixed classification of ξ_k in $\{\mathcal{D}_i\}$, finding the centers $\{\mu_i\}$ to minimize J is a sequence

of s unconstrained convex optimization problems and each μ_i is obtained by solving the linear equations system

$$\left(\sum_{k:\xi_k \in \mathcal{D}_i} R_k^{-1}\right)\mu_i = \sum_{k:\xi_k \in \mathcal{D}_i} R_k^{-1}\xi_k. \tag{11}$$

On the other hand, when the centers $\{\mu_i\}$ are fixed, the best possible classification is selected such that each point ξ_k is classified to the $\mathcal{D}_i$ with

$$i = \arg\min_{i'} \|\xi_k - \mu_{i'}\|_{R_k^{-1}}^2. \tag{12}$$

The following algorithm is proposed in [8] to locally minimize J.

Algorithm 1 [8] Clustering of feature vectors.

    INPUT ξ_k, R_k
    OUTPUT $\mathcal{D}_i, \mu_i$

1. Initialize $\{\mathcal{D}_i^{(0)}\}$ by randomly grouping ξ_k in them such that each $\mathcal{D}_i^{(0)}$ contains approximately N/s points

2. LET $j \leftarrow 1$

3. Compute $\{\mu_i^{(j)}\}$ from the fixed $\{\mathcal{D}_i^{(j-1)}\}$ using (11)

4. IF $j > 1$ AND $J(\{\mathcal{D}_i^{(j-1)}\}, \{\mu_i^{(j)}\}) =$
   $= J(\{\mathcal{D}_i^{(j-1)}\}, \{\mu_i^{(j)}\})$
   (a) LET $\{\mathcal{D}_i\} \leftarrow \{\mathcal{D}_i^{(j-1)}\}, \{\mu_i\} \leftarrow \{\mu_i^{(j)}\}$
   (b) RETURN

5. END

6. Compute the classification $\{\mathcal{D}_i^{(j)}\}$ from the fixed $\{\mu_i^{(j)}\}$ according to (12)

7. IF $J(\{\mathcal{D}_i^{(j)}\}, \{\mu_i^{(j)}\}) = J(\{\mathcal{D}_i^{(j-1)}\}, \{\mu_i^{(j)}\})$
   (a) LET $\{\mathcal{D}_i\} \leftarrow \{\mathcal{D}_i^{(j)}\}, \{\mu_i\} \leftarrow \{\mu_i^{(j)}\}$
   (b) RETURN

8. END

9. $j \leftarrow j + 1$ and go to step 3.

Theorem 1 [8] *Consider Algorithm 1. The following facts hold:*

1. *The sequence of the values of J through iterations j is non-increasing.*

2. *The algorithm terminates in a finite number of iterations j.*

Remark 1 The clustering in Algorithm 1 depends also on the random initialization of the classification of ξ_k in $\{\mathcal{D}_i^{(0)}\}$. Therefore it is advisable to repeat Algorithm 1 several times in order to obtain the solution close to the global optimum of J.

□

Now we are in position to mention also a way how the number of clusters s could be estimated along the clustering-based procedure. Note that prior to clustering in Algorithm 1 the number s is not needed. The authors in [8] reference to procedures of neural gas growing given in [16]. The idea is to start with a small number s and add a new center-cluster pair whenever Algorithm 1 converges. The new center is introduced into the cluster with the highest variance of points.

Note that feature vectors ξ_k incorporate also the information about the localization of the LD $\mathcal{C}_k$. This feature is useful when some affine submodel is valid over a non-convex region such that in the final PWARX model several convex regions share the same parameters Θ_i. However, when the submodels are actually different in different regions, discrimination over m_k in ξ_k could spoil the parameter clustering.

Once the clusters of feature vectors $\mathcal{D}_i$ are obtained, where $\mathcal{D}_i$ collects N_i feature vectors, the clusters $\mathcal{F}_i$ of regression vectors φ_k and $\mathcal{F}_i^y$ of pairs (φ_k, y_{k+1}) are simultaneously built ($|\mathcal{F}_i| = |\mathcal{F}_i^y| = N_i$). The final parameter estimates Θ_i in each cluster $\mathcal{D}_i$ are obtained by unconstrained minimization of the following cost function (weighted LS):

$$J_\Theta(\Theta_i) = \sum_{(\varphi_k, y_{k+1}) \in \mathcal{F}_i^y} w_k \left\| y_{k+1} - \Theta_i^T \left[\varphi_k^T \ 1 \right]^T \right\|_2^2 \quad (13)$$

with w_k being the peak of the Gaussian function associated with the covariance matrix R_k:

$$w_k = \frac{1}{\sqrt{(2\pi)^{2n+1} \det(R_k)}}. \quad (14)$$

2.1.1 Identification data pre-processing

We assume to deal with a continuous-time nonlinear process which can after time-discretization be arbitrarily well modeled with an appropriately parameterized discrete-time PWARX model. Moreover, the sampling times used for digital control of those processes are such that their output values change negligibly between the samples. Note that this assumption holds for majority of the digital control systems since usually their sampling time is chosen to be less than a tenth of the nonlinear process step response rise time from any operating point [17].

Originally, regression vectors are formed of successively sampled outputs and inputs. Since outputs do not change much within their neighboring samples captured in the regression vector (1), those

samples are of similar values. This implies that the regressor space is filled with the data-points placed only narrowly around the intersection of hyperplanes $y_k = y_{k-1}$, $y_{k-1} = y_{k-2}$, $\ldots y_{k-n_a+2} = y_{k-n_a+1}$. Thus, we expect all the borders in the regressor space between different ARX models to be compressed in that tiny space. Grouping of regression vectors in different groups $\mathcal{C}_k$ based on their such arrangement often results in many mixed LDs [18]. This can be mainly assigned to the fact that the model partitioning usually depends on output and its derivatives. Since the output derivatives are proportional to the difference between neighboring output samples, a general idea is to make the output difference $y_k - y_{k-1}$ more influential in grouping. This can be done by introducing a full-rank linear transformation on the regression vectors which changes the distances between them and thus affects grouping in LDs. In [18] we derived this transformation for a particular problem based on the a-priori process knowledge with many hard-to-tune parameters in it. We automated that transformation in [10] as follows. We compute an ellipse $\mathcal{E}$ with the smallest volume that contains all the regression vectors, see Figure 1(a). A single semi-definite program can be used to find its description [19]

$$(\varphi - \varphi_0)^T E(\varphi - \varphi_0) \leq 1, \quad (15)$$

where $\varphi_0 \in \mathbb{R}^n$ is the ellipse center and a symmetric matrix $E \in \mathbb{R}^{n \times n}$ is of full rank if all of the regression vectors do not lie on a single hyperplane which is practically always the case. If E is of full rank, we can also find a full-rank matrix $L \in \mathbb{R}^{n \times n}$ as its Cholesky factorization:

$$E = L^T L. \quad (16)$$

Now we can rewrite (15) in the following way:

$$(\varphi - \varphi_0)^T L^T L(\varphi - \varphi_0) \leq 1, \quad (17)$$

and if we note
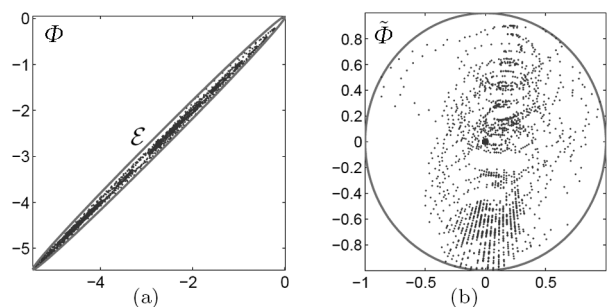
$$\tilde{\varphi} = L(\varphi - \varphi_0) \quad (18)$$

Fig. 1 Regression vectors in Φ encircled with the ellipse $\mathcal{E}$ (a) and the transformed ones in $\tilde{\Phi}$ in a unit ball (b)

we see that according to (17) the points $\tilde{\varphi} \in \tilde{\Phi}$ are all placed in a unit ball, see Figure 1(b). We now identify the model in $\tilde{\Phi}$, where the grouping is much more sensitive to output derivatives and thus results in more pure LDs. Once the model is found over $\tilde{\Phi}$, it can trivially be transformed into the corresponding model over Φ.

2.2 Model Partition Estimation

Once φ_k are classified in $\mathcal{F}_i$, the regions Φ_i can be obtained using linear classification techniques. Their aim is to find the separating hyperplanes in the regressor space

$$M_{ij}^T \varphi = m_{ij}, \qquad (19)$$

$M_{ij} \in \mathbb{R}^n$, $m_{ij} \in \mathbb{R}$, between each two clusters $\mathcal{F}_i$ and $\mathcal{F}_j$, $i < j$, such that φ may be in Φ_i only if $M_{ij}^T \varphi \leq m_{ij}$, and may be in Φ_j only if $M_{ij}^T \varphi \geq m_{ij}$. After all the separating hyperplanes are characterized, regions Φ_i are computed as

$$\Phi_i = \Phi \ \cap \ \bigcap_{j_1=1}^{i-1} \left(\left\{ \varphi | -M_{j_1 i}^T \varphi \leq -m_{j_1 i} \right\} \right) \cap$$

$$\bigcap_{j_2=i+1}^{s} \left(\left\{ \varphi | -M_{ij_2}^T \varphi \leq -m_{ij_2} \right\} \right)$$

$$:= \left\{ \varphi \ \ | \ H_{\Phi_i} \varphi \leq K_{\Phi_i} \right\}. \qquad (20)$$

Basically, the linear classification techniques can be divided into two groups. Ones are that process each pair of clusters (i, j) separately to find the separating hyperplane between them, like Robust Linear Programming (RLP) [20] or Support Vector Machines (SVM) [21]. The others simultaneously determine all the separating hyperplanes, like Multicategory RLP (M-RLP) [22] and Multicategory SVM (M-SVM) [23]. The basic disadvantage of RLP and SVM techniques compared to their M-RLP and M-SVM counterparts is that the resulting regions Φ_i may not form the partition of Φ since there may exist parts of Φ that do not belong to any Φ_i.

The most widely used technique in PWARX model identification is M-RLP, which is formulated as the following Linear Program (LP):

$$\min_{w_i,\gamma_i,z_{ij}} \ \sum_{i=1}^{s} \sum_{j=1}^{s} \frac{1}{N_i} \mathbf{1}_{N_i}^T z_{ij}$$

$$\text{s.t.} \begin{cases} F_i w_i - \gamma_i \mathbf{1}_{N_i} + z_{ij} \geq F_i w_j - \gamma_j \mathbf{1}_{N_i} + \mathbf{1}N_i, \\ \qquad i,j = 1,\cdots,s, i \neq j \\ z_{ij} \geq \mathbf{0}_{N_i,}, \ i,j = 1,\cdots,s, i \neq j, \qquad (21) \end{cases}$$

where $z_{ij} \in \mathbb{R}^{N_i}$ are introduced slack variables, $F_i \in \mathbb{R}^{N_i \times m}$ are row-wise formed of φ_k^T, $\varphi_k \in \mathcal{F}_i$. The optimal values of $w_i \in \mathbb{R}^n$ and $\gamma_i \in \mathbb{R}$ we denote with w_i^* i and γ_i^*, and they are used to construct the separating hyperplanes between the clusters:

$$M_{ij} = w_j^* - w_i^*, \quad m_{ij} = \gamma_j^* - \gamma_i^*, \quad i < j. \quad (22)$$

2.2.1 Modification in model partition estimation

The number of constraints and variables in the M-RLP problem (21) may be immense. It involves $(s-1)N + sn + s$ optimization variables and $2N(s-1)$ constraints such that the size of the matrix A of constraints $Az \leq b$ in the corresponding LP is $\mathcal{O}(N^2 s^2)$. The arising problem of matrix A storage in the computer memory can be partially alleviated by exploiting the sparse structure of A. In nonlinear model identification it is a known fact that the number of data-samples should always be large as long as the data are fairly distributed across the model domain [14]. This is problematic to achieve in PWARX identification if M-RLP is used for data classification. Namely, complex nonlinear processes often demand higher s for valuable identification results. This however imposes also N to be large such that each affine model may be satisfactorily identified.

If s is chosen suitably, it is expected that the clustering-based parameter identification and data classification produce nicely shaped clusters, i.e. almost piecewise linearly separable ones. In that case we propose a way-around to decrease the size of M-RLP: to identify the separating hyperplanes such that in each matrix F_i only extreme points of the cluster are left. The procedure is illustrated in Figure 2 with the toy-example of three piecewise linearly inseparable clusters. The main gain is that the number of data points may now grow very large, since the number of vertices of clusters in the same time statistically grows much slower [24].
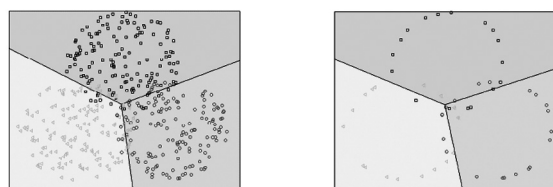
(a) Separating hyperplanes obtained using all points in M-RLP.

(b) Separating hyperplanes obtained using only vertices of clusters in M-RLP.

Fig. 2 Illustration of the proposed simplification of M-RLP

3 CONSTRAINED TIME OPTIMAL CONTROL OF PIECEWISE AFFINE PROCESSES

Discrete-time Piecewise Affine (DTPWA) process model is given by

$$x_{k+1} = A_i x_k + B_i u_k + a_i, \qquad (23a)$$

$$y_k = C_i x_k + c_i, \qquad (23b)$$

$$\text{if } \begin{bmatrix} x_k \\ u_k \end{bmatrix} \in \mathcal{D}_i, 1 \leq i \leq s, \qquad (23c)$$

$$\mathcal{D}_i = \left\{ \begin{bmatrix} x \\ u \end{bmatrix} \Big| H_i x + L_i u \leq K_i \right\}, \qquad (23d)$$

where $x \in \mathbb{R}^n$ is the model state, $u \in \mathbb{R}^m$ is the input and $y \in \mathbb{R}^p$ the output, A_i, B_i, a_i, C_i, c_i, H_i, L_i, K_i are properly dimensioned matrices. The set

$$\mathcal{D} = \bigcup_{i=1}^{s} \mathcal{D}_i$$

is called the model domain and $\mathcal{D}_i$ form its polyhedral partition. The polyhedra intersection $\mathcal{D}_i \cap \mathcal{D}_j$ may be lower-dimensional and thus the state-update and output equations may be multiplevalued on those intersections. The affine state-update in (23a) is shortly referred to as the i-th PWA model dynamics and switching between different dynamics in the neighboring sampling instants is called the dynamics switching. For the purpose of the reference tracking controller design we introduce the change of the control input

$$\bar{u}_k = u_k - u_{k-1}$$

and extend the state x_k with the past control input u_{k-1} and reference r_k, i.e.

$$\bar{x}_k = \begin{bmatrix} x_k^T & u_{k-1}^T & r_k^T \end{bmatrix}^T$$

where $\bar{x} \in \mathbb{R}^{n+m+p}$ and $r_{k+1} = r_k$, i.e. reference is assumed constant during model predictions. The input to the extended system now becomes $\bar{u}_k$, while the output is the tracking error $\bar{y} = r - y$. The extended model is

$$\bar{x}_{k+1} = \bar{A}_i \bar{x}_k + \bar{B}_i \bar{u}_k + \bar{a}_i, \qquad (24a)$$

$$\bar{y}_k = \bar{C}_i \bar{x}_k + \bar{c}_i, \qquad (24b)$$

$$\text{if } \begin{bmatrix} \bar{x}_k \\ \bar{u}_k \end{bmatrix} \in \bar{\mathcal{D}}_i, 1 \leq i \leq s, \qquad (24c)$$

$$\bar{\mathcal{D}}_i = \left\{ \begin{bmatrix} \bar{x} \\ \bar{u} \end{bmatrix} \Big| \bar{H}_i \bar{x} + \bar{L}_i \bar{u} \leq \bar{K}_i \right\}, \qquad (24d)$$

where $\bar{A}_i, \bar{B}_i, \bar{a}_i, \bar{C}_i, \bar{c}_i, \bar{H}_i, \bar{L}_i, \bar{K}_i$ straightforwardly follow from the non-extended model. Model (24) is shortly denoted with

$$\bar{x}_{k+1} = \bar{f}_{\text{PWA}}(\bar{x}_k, \bar{u}_k), \qquad (25a)$$

$$\bar{y}_k = \bar{g}_{\text{PWA}}(\bar{x}_k). \qquad (25b)$$

During operation the control system should respect the constraints on the states and inputs given with the polyhedral set $\bar{\mathcal{C}}^{xu}$:

$$\bar{\mathcal{C}}^{xu} = \left\{ \begin{bmatrix} \bar{x} \\ \bar{u} \end{bmatrix} \Big| C^x \bar{x} + C^u \bar{u} \leq C^0 \right\}, \qquad (26)$$

Without loss of generality, we assume that those constraints are included in the description of regions $\bar{\mathcal{D}}_i$.

Computation of the time optimal controller for a DTPWA system [3] consists of two distinct steps: (i) invariant set computation with the corresponding control law and (ii) computation of the time optimal control law outside the invariant set.

3.1 Invariant set computation

Controlled invariant set

$$\mathcal{X}^{\text{I}} = \left\{ \bar{x} \,|\, \exists \bar{u} : \bar{f}_{\text{PWA}}(\bar{x}, \bar{u}) \in \mathcal{X}^{\text{I}} \right\} \qquad (27)$$

is computed as the maximum controlled invariant set [25] contained in the so-called tracking origin $\mathcal{T}^0$ – a small polytopic subset of the set of feasible augmented states where $\|\bar{y}\|_l \leq \varepsilon$:

$$\mathcal{T}^0 = \left\{ \bar{x} \in \text{proj}_{\bar{x}} \bar{\mathcal{C}}^{xu} \,\Big|\, \|\bar{g}_{\text{PWA}}(\bar{x})\|_l \leq \varepsilon \right\}, \qquad (28)$$

where ε should reflect the desired control system tracking accuracy. To keep the consecutive computations in the polytopic set class, the norm in (28) should be linear, i.e. $l \in \{1, \infty\}$. The invariant set $\mathcal{X}^{\text{I}}$ is computed in an iterative manner. At the iteration step q (starting with $q = 0$) the set

$$\mathcal{H}_i^{0,q} \subset \bar{\mathcal{X}} \times \bar{\mathcal{U}}, \ i = 1, \ldots, s, \qquad ,$$

is computed for which $\bar{x}$ can enter $\mathcal{T}^q$ in one time instant using dynamics i while respecting all constraints defined by the polyhedron $\bar{\mathcal{D}}_i$

$$\mathcal{H}_i^{0,q} = \left\{ \begin{bmatrix} \bar{x}^T & \bar{u}^T \end{bmatrix}^T \in \bar{\mathcal{D}}_i \,\Big|\, \bar{A}_i \bar{x} + \bar{B}_i \bar{u} + \bar{f}_i \in \mathcal{T}^q \right\}. \qquad (29)$$

The target set for the next iteration step is then computed as

$$\mathcal{T}^{q+1} = \mathcal{T}^q \cap \left(\bigcup_{i=1}^{s} \mathcal{R}_i^{0,q} \right), \qquad (30)$$

where $\mathcal{R}_i^{0,q} = \underset{\bar{x}}{\text{proj}}\, \mathcal{H}_i^{0,q}$.

Remark 2. A special attention deserves the case when $\bar{u} \in \mathbb{R}$. Assume that $\mathcal{T}^q$ is a polytope. Let $\mathcal{H}_i \subset \bar{\mathcal{D}}$ be a full-dimensional polyhedron defined by (29) and let $\mathcal{R}_i$ be its projection on $\bar{x}$-space. Then for a given $\bar{x} \in \mathcal{R}_i$ there exists a unique one-dimensional interval $\mathcal{I}_i(\bar{x}) := [a(\bar{x}), b(\bar{x})] \subset \mathbb{R}$, with $a(\bar{x}) \leq b(\bar{x})$, such that for any $\bar{u} \in \mathcal{I}_i$ the state $\bar{x}$ moves in $\mathcal{T}^q$ at the next time-step, by using the i-th dynamics, while respecting the constraints. For a given $\bar{x}$ the interval $\mathcal{I}_i$ can be easily computed from the matrix description of $\mathcal{H}_i$, see Figure 3 for illustration. If $\mathcal{T}^q$ is given as a non-convex union of polytopes, then, in general, $\mathcal{H}_i$ also has the form of a non-convex union of polyhedra. In such a case feasible control actions $\bar{u}$ might comprise several nonconnected one-dimensional intervals.
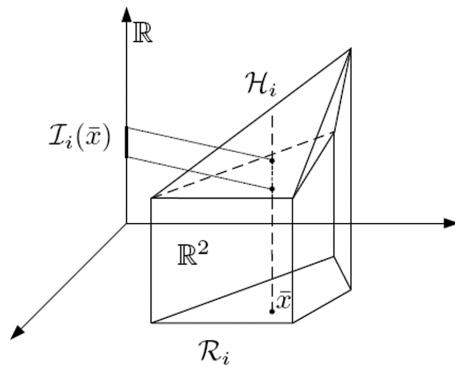
$\square$

Fig. 3 Determining the interval $\mathcal{I}_i(\bar{x})$

If $\mathcal{T}^{q+1} = \mathcal{T}^q$ the algorithm terminates and the controlled invariant set is found, with $\mathcal{X}^{\mathrm{I}} = \mathcal{T}^q$. Otherwise the whole procedure is repeated for the iteration step $q + 1$.

Suppose that the algorithm terminates at iteration q_0, i.e. $\mathcal{T}^{q_0} = \mathcal{T}^{q_0+1} = \mathcal{X}^{\mathrm{I}}$. For simplicity, in the rest of the paper the part of the set $\mathcal{H}_i^{0,q_0}$ over $\mathcal{T}^{q_0}$ is denoted with $\mathcal{H}_i^0$. Note that this set, as well as its projection $\mathcal{R}_i^0$, may be non-convex for $q_0 \geq 1$, and therefore they are in general represented as unions of polyhedra

$$\mathcal{H}_i^0 = \bigcup_{j=1}^{h_i} \mathcal{H}_{i,j}^0, \qquad (31)$$

$$\mathcal{R}_i^0 = \bigcup_{j=1}^{r_i} \mathcal{R}_{i,j}^0. \qquad (32)$$

The invariant set $\mathcal{X}^{\mathrm{I}}$ is thus

$$\mathcal{X}^{\mathrm{I}} = \bigcup_{i=1}^{s} \mathcal{R}_i^0 = \bigcup_{i=1}^{s} \bigcup_{j=1}^{r_i} \mathcal{R}_{i,j}^0. \qquad (33)$$

Remark 3. The control law extracted from $\mathcal{H}_i^0$ in general guarantees only the invariance property of the set $\mathcal{X}^{\mathrm{I}}$, i.e. the tracking error $\bar{y}$ will be bounded in a small set for constant references. The described procedure does not guarantee that $\bar{y}$ converges to zero. Therefore, in general, limit cycles and/or non-zero equilibrium points might occur. In the special case of a single-output system with the same output matrix for all dynamics ($\bar{C}_i \equiv \bar{C}, \forall_i$), one can achieve asymptotic convergence of $\bar{y}$ to zero by replacing (29) with

$$\mathcal{H}_i^{0,q} = \left\{ \begin{bmatrix} \bar{x} \\ \bar{u} \end{bmatrix} \in \bar{\mathcal{D}}_i \;\middle|\; \begin{array}{c} \bar{A}_i \bar{x} + \bar{B}_i \bar{u} + \bar{f}_i \in \mathcal{T}^q \\ \left| \bar{C}(\bar{A}_i \bar{x} + \bar{B}_i \bar{u} + \bar{f}_i) \right| \leq \delta \left| \bar{C}\, \bar{x} \right| \end{array} \right\}, \qquad (34)$$

where $0 < \delta < 1$ is an upper bound on the asymptotic convergence rate.

$\square$

For the control law computation only the polyhedra that define $\mathcal{H}_i^0$, $i = 1, \dots, s$, are needed. On the other hand, for the further off-line construction of the timeoptimal control law outside the invariant set only the polyhedra that define $\mathcal{R}_i^0$, $i = 1, \dots, s$, are needed.

For simplicity of the maximum controlled invariant set computation and of the consecutive computations it is very important to allow the regions $\mathcal{R}_i^0$ to have a full-dimensional intersection.

3.2 Time-Optimal Control Law Outside the Invariant Set

The time-optimal control problem for all $\bar{x}_0 \notin \mathcal{X}^{\mathrm{I}}$ is posed as follows

$$J^*(\bar{x}_0) = \min_{U_k, k} k$$

$$\text{subj. to} \begin{cases} \bar{x}_{k'} = \bar{f}_{\mathrm{PWA}}(\bar{x}_{k'-1}, \bar{u}_{k'-1}) \\ \bar{x}_{k'} \in \underset{\bar{x}}{\text{proj}}\, \bar{C}^{xu} \\ k' = 1, \cdots, k \\ \bar{x}_k \in \mathcal{X}^{\mathrm{I}} \end{cases} \qquad (35)$$

where $U_k = \{\bar{u}_0, \dots, \bar{u}_{k-1}\}$. The optimal cost corresponds to the minimal number of time steps in which the state $\bar{x}_0$ can be moved in the invariant set while respecting all constraints. The z-th cost-to-go set, denoted with $\mathcal{X}^z$, $z \in \{1, 2, \dots\}$, is the

set of all $\bar{x}$ for which the invariant set $\mathcal{X}^{\mathrm{I}}$ is attainable within z time-steps. By definition, $\mathcal{X}^0 := \mathcal{X}^{\mathrm{I}}$.

In [3] the solution to (35) is constructed by solving multi-parametric programs in a dynamic programming procedure. We also employ the dynamic programming approach when solving (35), but use the multi-parametric programming only in the case when $\bar{u}$ is more than one-dimensional [9]. The procedure starts with $z = 1$ and, similarly to the idea in Subsection 3.1, computes the set $\mathcal{X}^z$ from $\mathcal{X}^{z-1}$ (recall that $\mathcal{X}^0 := \mathcal{X}^{\mathrm{I}}$, where $\mathcal{X}^{\mathrm{I}}$ is given by (33)) as follows

$$\mathcal{X}^z = \bigcup_{i=1}^{s} \mathcal{R}_i^z, \qquad (36)$$

where $\mathcal{R}_i^z$ is the projection of the set

$$\mathcal{H}_i^z = \left\{ \begin{bmatrix} \bar{x}^T & \bar{u}^T \end{bmatrix}^T \in \bar{\mathcal{D}}_i \,\middle|\, \bar{A}_i \bar{x} + \bar{B}_i \bar{u} + \bar{f}_i \in \mathcal{X}^{z-1} \right\} \quad (37)$$

on the $\bar{x}$-space.

Note that $\mathcal{H}_i^z$ and $\mathcal{R}_i^z$ are unions of a finite number of polyhedra. The algorithm implementation also utilizes the idea from [26, 9] to reduce the switchings between different DTPWA model dynamics while preserving the time-optimality. The switching reduction is important for three practical reasons: (i) the model is usually the most inaccurate on dynamics switchings, (ii) the computed controller is simpler, and (iii) the switching between dynamics is usually connected with a more active control input.

The off-line computation stops at iteration $z_m + 1$ if $\mathcal{X}^{z_m+1} \subseteq \mathcal{X}^{z_m}$. The maximal controllable set (for more details see [27]) is then

$$\mathcal{K}_{\mathrm{PWA}}^\infty = \mathcal{X}^{z_m}. \qquad (38)$$

The design procedure outlined in this section requires practically no tuning once the DTPWA model is fixed. The only parameter that must be given is the desired tracking accuracy ε in the invariant set.

The controller is on-line implemented in the receding horizon fashion [28], i.e. at each sampling instant the control input $\bar{u}_0$ from the time optimal control sequence U_k^* is applied to the process. Since the function $\bar{u}_0(\bar{x}_0)$ is PWA, on-line computation is reduced to lookup table evaluation [5].

4 HYBRID APPROACH TO ELECTRONIC THROTTLE CONTROL – A CASE-STUDY

Electronic throttle is a controllable valve used in cars with internal combustion engines to regulate the air inflow into the combustion process. The electronic throttle control (ETC) system (Figure 4(a)) comprises a controller typically implemented in a microcontroller, a bipolar chopper and an electronic throttle. The electronic throttle consists of a DC drive powered by the chopper, a gearbox, a valve plate, a dual return spring and a position sensor. All throttle components are assembled in a compact electronic throttle body (ETB), shown in Figure 4(b), which is mounted on the engine air tube. As depicted in Figure 4(a) the control signal is fed to the bipolar chopper, which supplies the DC drive with the appropriate armature voltage. The armature current produces the motor torque that is transmitted through the gearbox to the throttle plate. The valve plate movement stops as the motor torque is counterbalanced by the torque of the dual return spring, the gearbox friction torque and the load torque caused by the air inflow. The opening angle of the valve corresponds to the angle between the valve plate and the air tube cross section and it spans from 13° (closed valve – no air inflow) to 103° (totally open valve). At the extreme valve positions mechanical safety stops prevent further valve plate shaft movement.
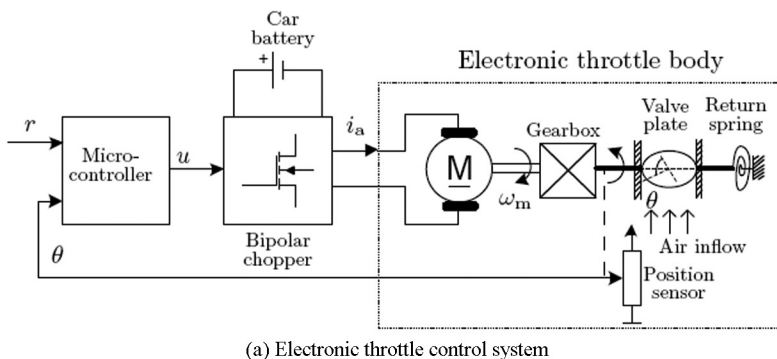
(a) Electronic throttle control system

(b) Electronic throttle body

Fig. 4 Electronic throttle

The opening angle is measured by the potentiometer sensor and this is the only feedback signal available in the standard ETC system.

Fast and accurate following of the reference opening angle in ETC has many benefits on the overall car performance, like faster torque control, reduction of fuel consumption and pollutants emission. However, this is hard to accomplish since the ETB exhibits two strong nonlinear effects: gearbox friction and the nonlinear return spring characteristics.

4.1 Clustering-based Identification of the PWARX Electronic Throttle Model

In this subsection we use the clustering-based procedure to identify the PWARX electronic throttle model (4) where the output y is the valve opening angle θ, while the control input u is the chopper control signal. The number of past outputs n_a is chosen to be 3 since three states may be observed on the throttle – the valve angle, its velocity and the presliding displacement connected with the dynamic friction torque [5]. The numbers n_b and s are selected based on validation of identification results obtained by varying them. Finally: $n_b = 1$, $s = 12$. Note that the number of submodels is more than halved compared to the first-principles model in [5] where $s = 30$. The identification data $\mathcal{S} = \{(x_k, y_{k+1})\}_{k=1}^{N}$ were collected in closed-loop using PID controller from [29] with $N = 4300$. The data is fairly distributed over the whole throttle operating range which was achieved using two periods of reference ramp signals (see Figure 5): in the first one the ramp was burdened with high level of noise to catch the process dynamic behavior and in the second one the noise level was significantly lowered in order to properly identify the static behavior.
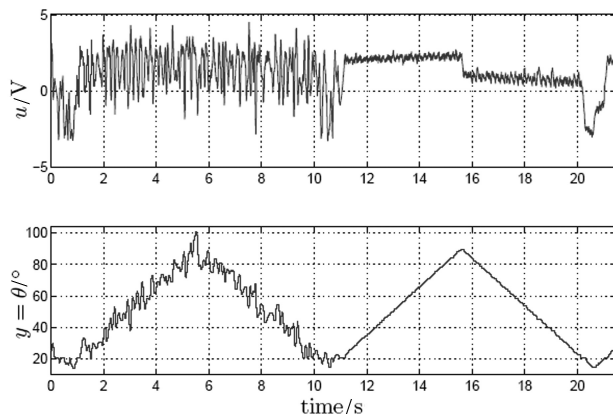

Fig. 5 Identification data for the electronic throttle PWARX model

Since significant influence of measurement quantization noise ($0.11°$) is present in output samples, the number c of data in the group should be relatively large and proportional to N to reliably identify the local parameters. In our identification setup, c is 70. Large c, however, gives rise to the number of mixed LDs. The friction nonlinearity present in the throttle is mostly the function of the plate angular velocity, i.e. of the output difference $y_k - y_{k-1}$ and this causes most of the LDs to be mixed if the original data are grouped in LDs. Namely, the identification data are densely sampled ($T = 5$ ms) and this makes the regression vectors with different sign of the difference close in the regressor space. According to the discussion in subsection 2.1.1, the regression vectors should be pre-processed by a linear transformation defined by the minimum-volume ellipse encircling all the regression vectors (15).

Under assumption that the regression vectors are not all placed on a single hyperplane, which is practically always the case, the symmetric matrix $E \in \mathbb{R}^{n \times n}$ is of full rank. This means that its square-root matrix L exists ($E = L^T L$) and the linear transformation on regression vectors is defined with

$$\tilde{\varphi}_k = L(\varphi_k - \varphi_0). \qquad (39)$$

Note that this transformation does not require any tuning, unlike our earlier solutions for clustering-based throttle identification given in [18]. The whole identification algorithm is then actually performed on the data $\tilde{\mathcal{S}} = \{(\tilde{\varphi}_k, y_{k+1})\}_{k=1}^{N}$ which finally yields the PWARX model

$$y_{k+1} = f'(\tilde{\varphi}_k) + e_k \qquad (40)$$

with

$$f'(\tilde{\varphi}) = \begin{cases} \tilde{\Theta}_1^T \begin{bmatrix} \tilde{\varphi} \\ 1 \end{bmatrix} & \text{if } \tilde{\varphi} \in \tilde{\Phi}_1, \\ \vdots \\ \tilde{\Theta}_s^T \begin{bmatrix} \tilde{\varphi} \\ 1 \end{bmatrix} & \text{if } \tilde{\varphi} \in \tilde{\Phi}_s, \end{cases} \qquad (41)$$

After the clustering, multicategory linear classification using M-RLP is performed on all the data points which makes the linear program (21) extremely large and impossible to handle on a standard desktop computer. The classification is thus performed over vertices of clusters as described in subsection 2.2.1 which results in a tractable LP. Once f' is estimated, it is straightforward to obtain f from it using (39):

$$\Theta_i = \begin{bmatrix} L^T & \mathbf{0}_n \\ -\varphi_0^T L^T & 1 \end{bmatrix} \tilde{\Theta}_i, \qquad (42)$$

$$\tilde{\Phi}_i = \left\{ \tilde{\varphi} \middle| H_{\tilde{\Phi}_i} \, \tilde{\varphi} \le K_{\tilde{\Phi}_i} \right\} \rightarrow \Phi_i = \left\{ \varphi \middle| H_{\Phi_i} \varphi \le K_{\Phi_i} \right\}$$

$$H_{\Phi_i} = H_{\tilde{\Phi}_i} L,$$

$$K_{\Phi_i} = K_{\tilde{\Phi}_i} + H_{\tilde{\Phi}_i} L \varphi_0. \qquad (43)$$

In Figure 6 we show the model behavior on the validation data in the off-line validation for a noisy control input. »Off-line« here means that the regression vector of the model is initialized at time 0 from the process and later on they evolve independently – they are just supplied by the same input. The model exhibits offsets at some time instants which is mainly assigned to the stochastic behavior of friction and the high process dynamic gain. However, more important is that the response shape is precisely followed since in the control application the model is used for predictions from the current measured state and is thus practically re-initialized at each sampling instant. The lower subfigure in Figure 6 reveals that all the affine submodels are activated during the process operation. In Figure 7 we show a detail of the off-line model validation for low-noised control input (quasi-steadystate model validation).

4.2 Off-line Synthesis Based on PWARX Electronic Throttle Model

4.2.1 DTPWA model

We define the reference tracking problem for the electronic throttle using model $\bar{x}_{k+1} = \bar{f}_{PWA}(\bar{x}_k, \bar{u}_k)$ with the state $\bar{x}$:

$$\bar{x}_k = \left[y_k \ y_{k-1} \ y_{k-2} \ u_{k-1} \ r_k \right]^T. \qquad (44)$$

Matrices of the model for $i = 1, \ldots, s$ are:

$$\bar{A}_i = \begin{bmatrix} \Theta_{i,1} & \Theta_{i,2} & \Theta_{i,3} & \Theta_{i,4} & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad \bar{B}_i = \begin{bmatrix} \Theta_{i,4} \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix},$$

$$\bar{a}_i = \left[\Theta_{i,5} \ 0 \ 0 \ 0 \ 0 \right]^T, \qquad \bar{C}_i = \left[-1 \ 0 \ 0 \ 0 \ 1 \right],$$

$$\bar{c}_i = [0], \qquad\qquad \bar{H}_i = \left[H_{\Phi_i} \ \mathbf{0} \right],$$

$$\bar{L}_i = \left[H_{\Phi_i} \cdot \left[0 \ 0 \ 0 \ 1 \ 0 \right]^T \right], \quad \bar{K}_i = \left[K_{\Phi_i} \right].$$
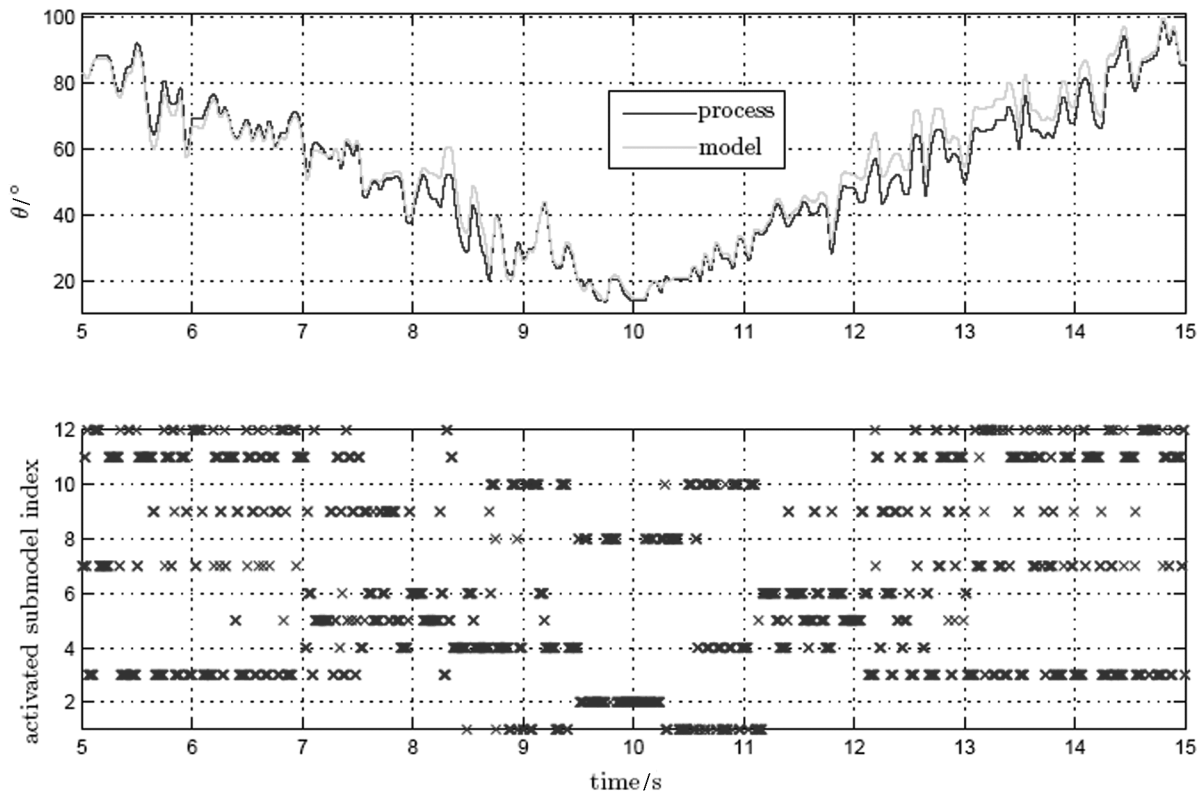

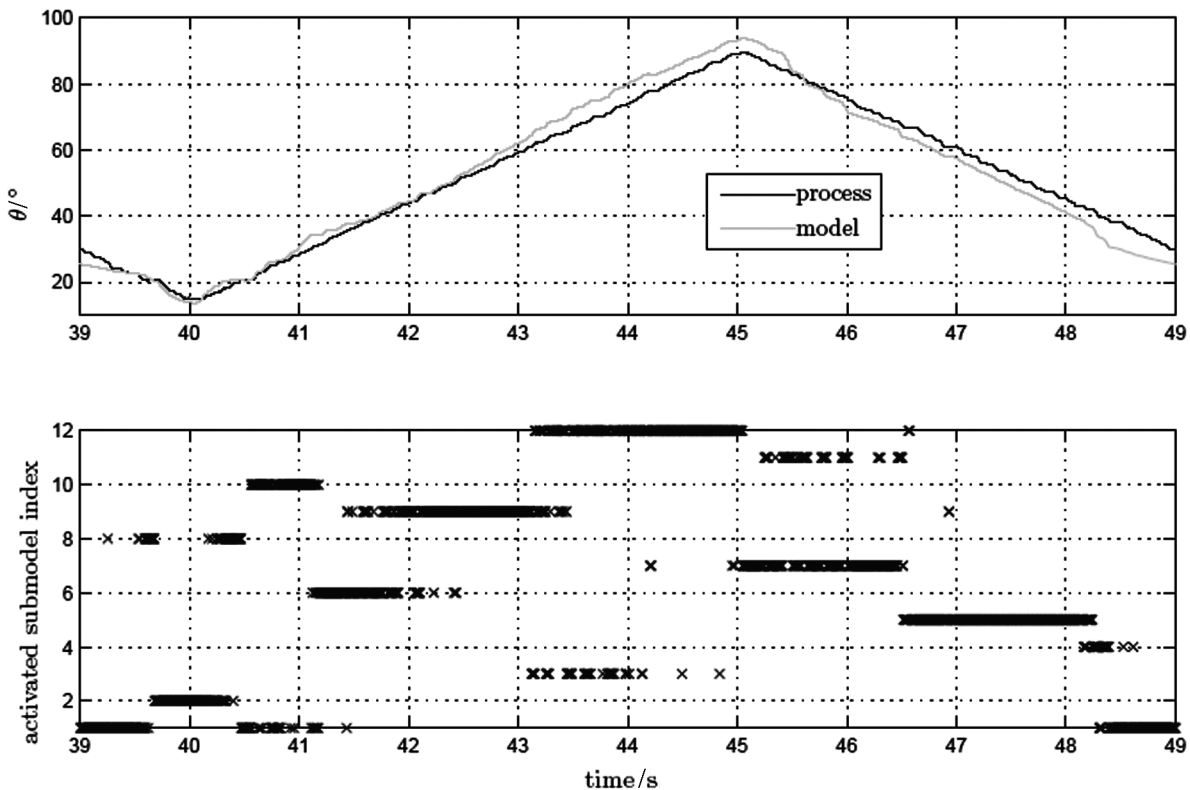*Fig. 6 Detail of the off-line model validation for a noisy control input*

Fig. 7 Detail of the off-line model validation in quasi-steady-state

4.2.2 Constraints

All the constraints we want to impose on the throttle variables must be expressed by the state $\overline{x}$ and the input $\overline{u}$ to form the set $\overline{C}^{xu}$. This is straightforward for the constraints on the angle θ, on the control input u and on the control input change $\overline{u}$. However, to express the constraint on the motor angular velocity ω_m and the armature current i_a we have three choices. One is to estimate both using, e.g. continuous nonlinear model and Kalman filtering technique while collecting identification data and then to find the best linear fit between $\overline{x}_k$ and $\hat{\omega}_k$ and $\overline{x}_k$ and $\hat{i}_{a,k}$:

$$\omega_m = h_1^T x,$$
$$i_a = h_2^T x.$$

The second way is to approximately derive h_1 and h_2 based on the known physical process parameters K_a, K_{ch}, K_l, K_v [5]:

$$\omega_{m,k} \approx \frac{\pi}{180 K_l} \frac{y_k - y_{k-1}}{T}$$

$$i_{a,k} \approx K_a \left[K_{ch}(\overline{u}_k + u_{k-1}) - K_v \omega_{m,k} \right] \approx$$

$$\approx K_a \left[K_{ch}(\overline{u}_k + u_{k-1}) - K_v \frac{\pi}{180 K_l} \frac{y_k - y_{k-1}}{T} \right].$$

The only way to avoid the necessity for an auxiliary process model or for some physical process parameters is to collect the data of ω_m and i_a using additional sensors in the identification phase and then to use the first approach of their best fitting with some affine function of x. We use here the second approach with known physical parameters to derive constraints on ω_m and i_a. The maximum allowed value for $|\omega_m|$ is $\omega_{m,\lim} = 180$ rad/s. The maximum allowed value for $|i_a|$ is $i_{a,\lim} = 2$ A.

4.2.3 Off-line controller design

We compute the maximum controlled invariant set inside the set

$$\mathcal{T}^0 = \{\overline{x} | |\overline{y}| \le 0.05\} \tag{45}$$

which ensures the desired tracking accuracy of the ETC system [5]. The invariant set $\mathcal{X}^I \equiv \mathcal{X}^0$ is computed within 5 iterations and consists of 91 polytopes.

Two cost-to-go sets around the invariant set are computed which are enough to test the obtained time-optimal control system for small reference steps. Computations were performed with the aid of Multi-Parametric Toolbox [19]. The controller consists of about 2800 polytopes in the state-input space.
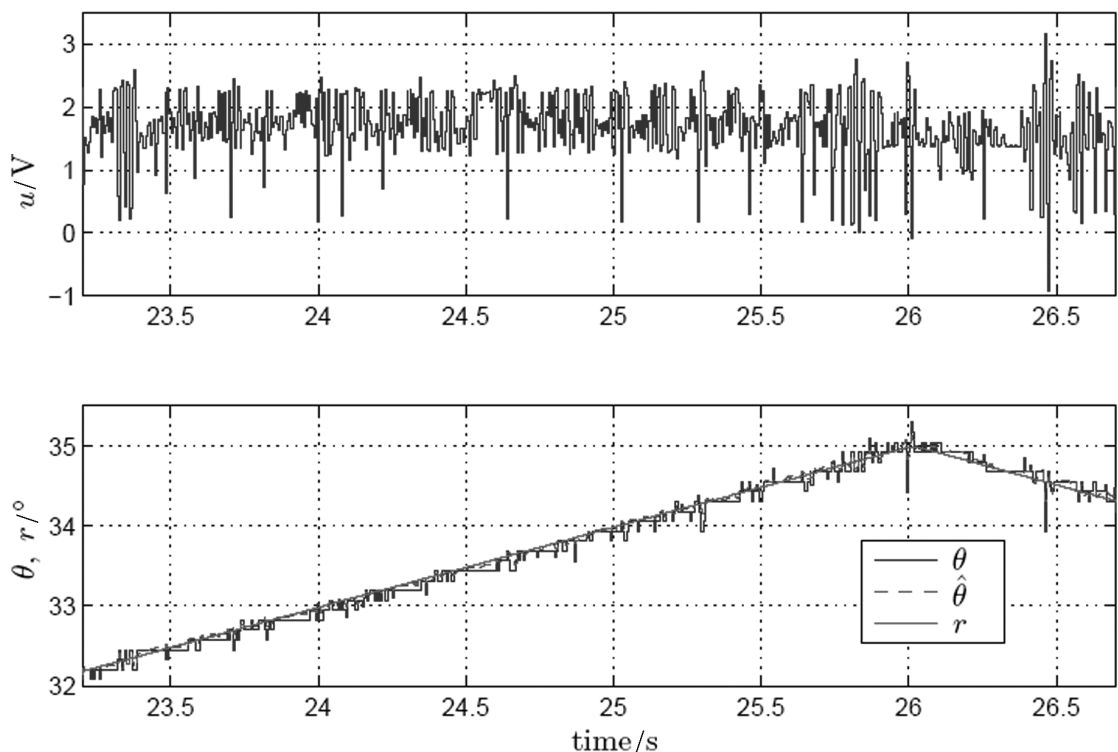
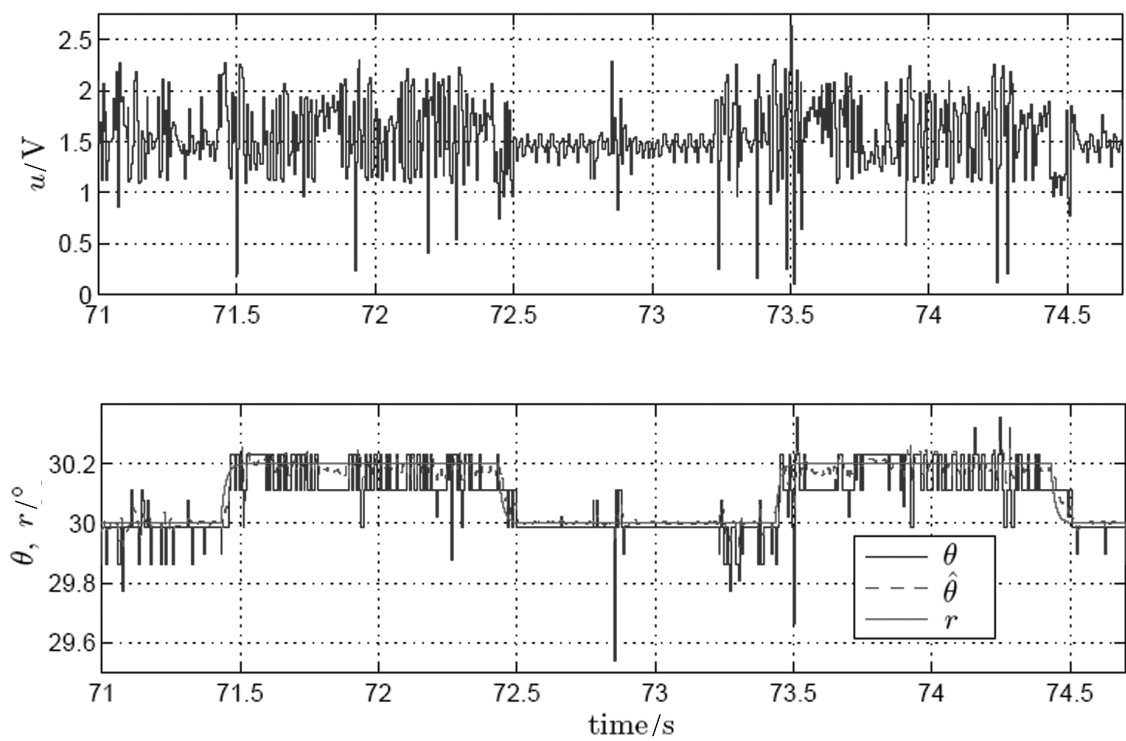Fig. 8 Response of the control system based on the identified model to the ramp reference

Fig. 9 Response of the control system based on the identified model to the 0.2° square reference
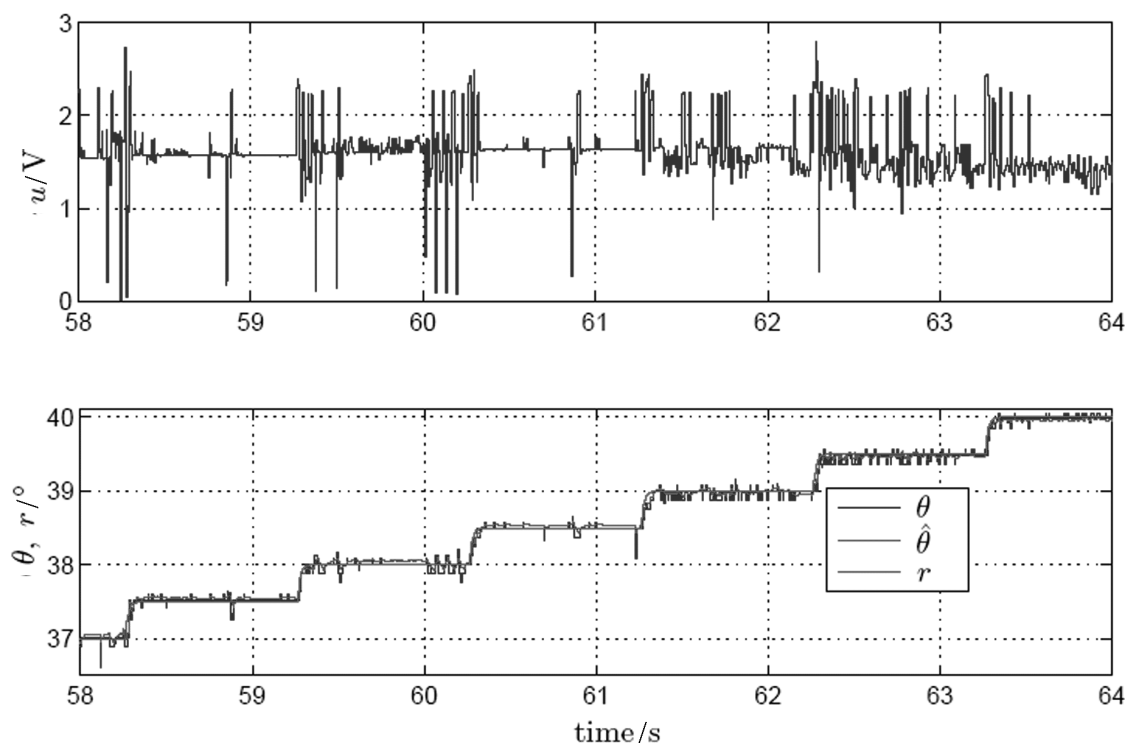
Fig. 10 Response of the control system based on the identified model to the 0.5° stairs reference

4.3 Experimental Results

We experimentally verify the computed time optimal controller that demands approximately 1 MB of memory for its storage. In Figure 8 we show the control system response to the ramp reference, while Figure 9 and Figure 10 show the responses to the 0.2° and 0.5° reference steps, respectively.

Very fast transients may be observed. To the author's knowledge, this is the first time that the identified DTPWA process model is used for the synthesis and that the obtained controller is experimentally verified on the process. Obtained responses are comparable with the results obtained in [5]; in both cases the transients are approximately twice shorter than the ones obtained using PID control with model-based pre-compensations of nonlinearities [29].

5 CONCLUSION

In this paper we focus on identification and time optimal control of nonlinear processes modeled using piecewise affine class of hybrid systems. Piecewise affine models may approximate nonlinear systems arbitrarily well. We bind the piecewise ARX process model identification based on clustering and the constrained time optimal controller design for discretetime piecewise affine systems into a systematic procedure for the design of high-performance nonlinear control systems. The procedure starts from the collection of the identification data and results in a closedform time optimal controller. We experimentally verify the procedure on the electronic throttle control system case-study. Comparison of the obtained electronic throttle control system with the control system obtained by combining a PID control algorithm with nonlinearities compensations issues three major achievements: (i) the outlined design is systematic, (ii) constraints on control system variables are explicitly addressed and (iii) obtained responses are twice faster and yet without an overshoot.

ACKNOWLEDGEMENT

REFERENCES

[1] A. Bemporad, M. Morari, **Control of Systems Integrating Logic, Dynamics, and Constraints.** Automatica, 35(3):407–427, 1999.

[2] W. P. M. H. Heemels, B. De Schutter, A. Bemporad, **Equivalence of Hybrid Dynamical Models.** Automatica, 37(7):1085–1091, 2001.

[3] P. Grieder, M. Kvasnica, M. Baotić, M. Morari, **Stabilizing Low Complexity Feedback Control of Constrained Piecewise Affine Systems.** Automatica, 41(10):1683–1694, 2005.

[4] M. Morari, J. H. Lee, **Model Predictive Control: Past, Present and Future.** Computers & Chemical Engineering, 23:667–682, 1999.

[5] M. Vašak, M. Baotić, I. Petrović, N. Perić, **Hybrid Theory Based Time-Optimal Control of an Electronic Throttle.** IEEE Transactions on Industrial Electronics, 54(3):1483–1494, 2007.

[6] S. Paoletti, A. Lj. Juloski, G. Ferrari-Trecate, R. Vidal, **Identification of Hybrid Systems: A Tutorial.** European Journal of Control, 13(2–3):242–260, 2007.

[7] A. Lj. Juloski, W. P. M. H. Heemels, G. Ferrari-Trecate, R. Vidal, S. Paoletti, J. H. G. Niessen, **Comparison of Four Procedures for the Identification of Hybrid Systems.** In Hybrid Systems: Computation and Control, volume 3414/2005 of Lecture notes in computer science, pages 354–369. Springer Berlin/Heidelberg, March 2005.

[8] G. Ferrari-Trecate, M. Muselli, D. Liberati, M. Morari, **A Clustering Technique for the Identification of Piecewise Affine Systems.** Automatica, 39(2):205–217, 2003.

[9] M. Vašak, **Time Optimal Control of Piecewise Affine Systems.** PhD thesis, Faculty of Electrical Engineering and Computing, University of Zagreb, July 2007.

[10] M. Vašak, D. Klanjčić, N. Perić, **Piecewise Affine Identification of MIMO Processes**. In Proceedings of the Joint Conference on Control Applications, Symposium on Intelligent Control and Computer-Aided Control Systems Design, pages 1493–1498, Munich, Germany, 2006.

[11] A. Bemporad, A. Garulli, S. Paoletti, A. Vicino, **A Bounded-Error Approach to Piecewise Affine System Identification.** IEEE Transactions on Automatic Control, 50(10):1567–1580, 2005.

[12] A. Lj. Juloski, **Observer Design and Identification Methods for Hybrid Systems – Theory and Experiments.** PhD thesis, Technical University Eindhoven, October 2004.

[13] R. Vidal, S. Soatto, Y. Ma, S. Sastry. **An Algebraic Geometric Approach to the Identification of a Class of Linear Hybrid Systems.** In Proceedings of the 42nd IEEE Conference on Decision and Control, pages 167–172, Maui, Hawaii, USA, December 2003.

[14] L. Ljung, **System Identification – Theory For the User.** PTR Prentice Hall, New Jersey, 1999.

[15] R. O. Duda, P. E. Hart, **Pattern Classification and Scene Analysis.** Wiley, New York, 1973.

[16] B. Fritzke, **Some Competitive Learning Methods.** Technical report, Institute for Neural Computation, Ruhr-Universität Bochum, 1997.

[17] K. J. Åström, B. Wittenmark, **Computer Controlled Systems: Theory and Design.** Prentice Hall, New Jersey, 1989.

[18] M. Vašak, L. Mladenović, N. Perić, **Clustering-based Identification of a Piecewise Affine Electronic Throttle Model.** In Proceedings of the 31st Annual Conference of the IEEE Industrial Electronics Society, pages 177–182, Rayleigh, North Carolina, USA, November 2005.

[19] M. Kvasnica, P. Grieder, M. Baotic, M. Morari, **Multi-Parametric Toolbox (MPT).** 2003. http://control.ee.ethz.ch/~hybrid/mpt/.

[20] K. P. Bennet, O. L. Mangasarian, **Neural Network Training via Linear Programming.** In P. M. Pardalos, editor, Advances in Optimization and Parallel Computing, pages 56–67. North-Holland, Amsterdam, 1992.

[21] V. N. Vapnik, **The Nature of Statistical Learning Theory.** John Wiley & Sons, New York, 1996.

[22] K. P. Bennet, O. L. Mangasarian, **Multicategory Discrimination via Linear Programming.** Optimization Methods and Software, 3:27–39, 1994.

[23] E. J. Bredensteiner, K. P. Bennett, **Multicategory Classification by Support Vector Machines.** Computational Optimization and Applications, 12(1–3): 53–79, 1999.

[24] P. M. Pardalos, Y. Li, W. W. Hager, **Linear Programming Approaches to the Convex Hull Problem in $\mathbb{R}^m$.** Computers and Mathematics with Applications, 29(7):23–29, 1995.

[25] F. Blanchini, **Set Invarinace in Control – A Survey.** Automatica, 35(11):1747–1767, 1999.

[26] M. Vašak, M. Baotić, M. Morari, I. Petrović, N. Perić, **Constrained Optimal Control of an Electronic Throttle.** International Journal of Control, 79(5):465–478, 2006.

[27] S. Raković, P. Grieder, M. Kvasnica, D. Q. Mayne, M. Morari. **Computation of Invariant Sets for Piecewise Affine Discrete Time Systems Subject to Bounded Disturbances.** In Proceedings of the 43rd IEEE Conference on Decision and Control, pages 1418–1423, Atlantis, Bahamas, December 2004.

[28] D. Q. Mayne, J. B. Rawlings, C. V. Rao, P. O. M. Scokaert, **Constrained Model Predictive Control: Stability and Optimality.** Automatica, 36(6):789–814, 2000.

[29] J. Deur, D. Pavković, N. Perić, M. Jansz, D. Hrovat, **An Electronic Throttle Control Strategy Including Compensation of Friction and Limp-Home Effects.** IEEE Transactions on Industry Applications, 40(3): 821–834, 2004.

Povezivanje identifikacije i optimalnog upravljanja s ograničenjima za po dijelovima afine sustave.
U radu se razmatra identifikacija i upravljanje nelinearnih procesa modeliranih po dijelovima afinim modelom. Povezuju se postupak identifikacije po dijelovima ARX modela procesa temeljen na uskupljavanju i postupak sinteze eksplicitnog vremenski optimalnog regulatora uz prisutna ograničenja za vremenski diskretne po dijelovima afine sustave. Ovaj je pristup pogodan za sintezu nelinearnog sustava upravljanja visokih zahtjeva, te je u ovom radu eksperimentalno provjeren na primjeru sustava upravljanja elektroničkom zaklopkom automobila.

Ključne riječi: po dijelovima afini model, identifikacija modela, uskupljavanje, vremenski optimalno upravljanje uz prisutna ograničenja, elektronička zaklopka

AUTHORS' ADDRESS

Mario Vašak, Ph.D.E.E.
Prof. Nedjeljko Perić, Ph.D.E.E.
Department of Control and Computer Engineering,
Faculty of Electrical Engineering and Computing,
University of Zagreb,
Unska 3, HR-10000 Zagreb, Croatia