International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:5, No:5, 2011

Identification of a PWA model of a batch reactor for model predictive control

Gorazd Karer Igor Škrjanc Borut Zupančič

Abstract—The complex hybrid and nonlinear nature of many processes that are met in practice causes problems with both structure modelling and parameter identification; therefore, obtaining a model that is suitable for MPC is often a difficult task. The basic idea of this paper is to present an identification method for a piecewise affine (PWA) model based on a fuzzy clustering algorithm. First we introduce the PWA model. Next, we tackle the identification method. We treat the fuzzy clustering algorithm, deal with the projections of the fuzzy clusters into the input space of the PWA model and explain the estimation of the parameters of the PWA model by means of a modified least-squares method. Furthermore, we verify the usability of the proposed identification approach on a hybrid nonlinear batch reactor example. The result suggest that the batch reactor can be efficiently identified and thus formulated as a PWA model, which can eventually be used for model predictive control purposes.

Keywords—Batch reactor, fuzzy clustering, hybrid systems, identification, nonlinear systems, PWA systems.

I. Introduction

Dynamic systems that involve continuous and discrete states are called *hybrid systems*. Most industrial processes contain both continuous and discrete components, for instance, discrete valves, on/off switches, logical overrides, etc. The continuous dynamics are often inseparably interlaced with the discrete dynamics; therefore, a special approach to modelling and control is required. At first this topic was not treated systematically [20]. In recent years, however, hybrid systems have received a great deal of attention from the computer science and control community.

Model predictive control (MPC) presents one of the advanced approaches that is widely used in industrial practice. At first, MPC was only employed in the petrochemical industry, but it has been constantly gaining a reputation of a generally usable approach for a wide spectrum of control problems. Lately, MPC has not been limited only to slow processes, where there is plenty of time for calculations between successive time-steps, but it has also been gaining ground in the field of fast processes. That said, when dealing with control problems involving complex dynamics, computational complexity still remains the main issue. MPC is based on forecasting the future behavior of a system at each sampling instant using the process model. The complex hybrid and nonlinear nature of many processes that are met in practice causes problems with both structure modelling and parameter identification; therefore, obtaining a model that is suitable for MPC is often a difficult task. Hence, the need for special methods and formulations when dealing with hybrid systems is very clear.

G. Karer, I. Škrjanc and B. Zupančič are with the Faculty of Electrical Engineering, University of Ljubljana, Tržaška 25, SI–1000 Ljubljana, Slovenia, e-mail: gorazd.karer@fe.uni-lj.si.

Manuscript received February 28, 2011.

MPC methods for hybrid systems employ several model formulations. Often the system is described as *mixed logical dynamical* (MLD) [4]. A lot of interest has also been devoted to *piecewise affine* (PWA) formulation [17], which has been proven to be equivalent to many classes of hybrid systems [6]. What is more, MLD models can be transformed to the PWA form. The optimal control problem for discrete-time PWA systems can be converted to a mixed-integer optimization problem and solved online [10]. On the other hand, in [9] the authors tackle the optimal control problem for PWA systems by solving a number of multi-parametric programs offline. In such manner, it is possible to obtain a solution in the form of a PWA state feedback law that can be efficiently implemented online.

The aforementioned methods mainly consider systems with continuous inputs, despite the fact that solutions based on *(multiparametric) mixed integer linear/quadratic programming* (mp-MIQP/MILP) can be applied to systems with discrete inputs as well. However, the computational complexity increases drastically with the number of discrete states, and so these methods can become computationally too demanding. An algorithm for the efficient MPC of hybrid systems with discrete inputs only is proposed in [13].

Most of the previous work related to the MPC of hybrid systems is based on (piecewise) linear and equivalent models. However, such approaches can prove unsuccessful when dealing with distinctive nonlinearities. Since a PWA formulation can only represent piecewise affine systems, further segmentation is required in order to suitably approximate the nonlinearity. The new segments introduce new discrete auxiliary variables in the MILP/MIQP optimization program, which causes a higher complexity, often resulting in programs that are computationally too demanding.

A nonlinear modelling approach for MPC purposes is presented in [16]. The authors introduce an analytical predictive-control-law for fuzzy systems. The modelling and identification methodology is usable for plain nonlinear systems, but not for the structurally more complex class of hybrid systems. A hierarchical identification of a fuzzy switched system [21] is introduced in [11]. Furthermore, two structure-selecting methods for nonlinear models with mixed discrete and continuous inputs are presented in [5]. In [14] a fuzzy control method is implemented in the low control-level for a class of hybrid systems based on hybrid automata. The authors in [15] base the demand prediction on a fuzzy clustering

algorithm, which results in appropriate call probabilities for uncertain future.

The basic idea of this paper is to present an identification method for a PWA model based on a fuzzy clustering algorithm. The outline of the paper is as follows. Section II introduces the PWA model. Next, in section III the identification method is explained. We verify the usability of the proposed identification approach on a nonlinear hybrid batch reactor example in section IV. Finally, we give some concluding remarks.

II. Modelling of a PWA model

In the literature that deals with hybrid systems (particularly predictive control of hybrid systems) a lot of attention has been devoted to *piecewise affine* (*PWA*) *models* [17]. In PWA models the input-state space is partitioned into several subspaces. In every subspace the dynamics of the modelled system are described by an affine system. In this regards, PWA models represent one of the simplest formulations, which in a way generalize the classic linear-system formulation in the state-space domain – see eq. (1) – and are capable of describing the hybrid and nonlinear dynamics to a theoretically arbitrary precision. In addition, the PWA models can be transformed into several other formulations under in reality not very limiting conditions [6], [3], [12].

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) + Du(k) \end{aligned} \tag{1}$$

In eq. (1), $x(k) \in \mathbb{R}^n$ denotes the state of the system, $u(k) \in \mathbb{R}^m$ stand for the input of the system and $y(k) \in \mathbb{R}^l$ for the output of the system. k denotes the current time-step. The matrices $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, $C \in \mathbb{R}^{l \times n}$ and $D \in \mathbb{R}^{l \times m}$ define the dynamics of the linear system.

By generalizing the discrete-time linear-system formulation in eq. (1), we can derive the PWA formulation as given in eqs. (2) and (3).

$$\begin{aligned} x(k+1) &= A_i x(k) + B_i u(k) + f_i \\ y(k) &= C_i x(k) + D_i u(k) + g_i \end{aligned} \tag{2}$$

$$\begin{bmatrix} x(k) \\ u(k) \end{bmatrix} \in \Omega_i \tag{3}$$

In eq. (2), $x(k) = [x_c(k)^T x_b(k)^T]$ denotes the state of the system. Here, $x_c(k)$ represents the continuous part of the state and $x_b(k)$ the binary part of the state. Similarly, the input of the system $u(k) = [u_c(k)^T u_b(k)^T]$ and the output of the system $y(k) = [y_c(k)^T y_b(k)^T]$ are divided into the continuous and the binary part.

The matrices A_i, B_i, C_i and D_i and the vectors f_i and g_i define the affine dynamic in a particular subspace of the system, which is denoted by the index $i \in \mathbb{Z} < \infty$. In case the vectors f_i and g_i are 0, we are dealing with a piecewise linear system.

The dimensions of the matrices and vectors are defined in eqs. (4), (5) and (6).

$$\begin{aligned} &x_c(k) \in \mathbb{R}^{n_c},\ x_b(k) \in \{0,1\}^{n_b} \\ &u_c(k) \in \mathbb{R}^{m_c},\ u_b(k) \in \{0,1\}^{m_b} \\ &y_c(k) \in \mathbb{R}^{l_c},\ y_b(k) \in \{0,1\}^{l_b} \end{aligned} \tag{4}$$

$$\begin{aligned} A_i &\in (\mathbb{R}^{n_c} \times \{0,1\}^{n_b}) \times (\mathbb{R}^{n_c} \times \{0,1\}^{n_b}) \\ B_i &\in (\mathbb{R}^{n_c} \times \{0,1\}^{n_b}) \times (\mathbb{R}^{m_c} \times \{0,1\}^{m_b}) \\ C_i &\in (\mathbb{R}^{l_c} \times \{0,1\}^{l_b}) \times (\mathbb{R}^{n_c} \times \{0,1\}^{n_b}) \\ D_i &\in (\mathbb{R}^{l_c} \times \{0,1\}^{l_b}) \times (\mathbb{R}^{m_c} \times \{0,1\}^{m_b}) \end{aligned} \tag{5}$$

$$\begin{aligned} f_i &\in \mathbb{R}^{n_c} \times \{0,1\}^{n_b} \\ g_i &\in \mathbb{R}^{l_c} \times \{0,1\}^{l_b} \end{aligned} \tag{6}$$

A particular subspace $\Omega_i \subset (\mathbb{R}^{n_c} \times \{0,1\}^{n_b}) \times (\mathbb{R}^{m_c} \times \{0,1\}^{m_b})$ in eq. (3) represents a convex polyhedron in the input-state space of the system. Each polyhedron is defined by a system of linear inequalities in the input-state space of the system as shown in eq. (7).

$$\Omega_i \equiv \left\{ \begin{bmatrix} x(\cdot) \\ u(\cdot) \end{bmatrix} ; \begin{array}{l} H_i x(\cdot) + J_i u(\cdot) \leq K_i, \\ \tilde{H}_i x(\cdot) + \tilde{J}_i u(\cdot) < \tilde{K}_i \end{array} \right\} \tag{7}$$

In eq. (7) H_i, J_i, K_i and $\tilde{H}_i$, $\tilde{J}_i$, $\tilde{K}_i$ denote the matrices that define the subspace Ω_i. the number of rows in the matrices equals the number of the inequalities, i.e. the number of hyperplanes delimiting the subspace Ω_i.

It should be noted that in practice, due to numerical problems in defining the polyhedra using the inequalities above, $<$ is substituted with $\leq$. In this case, the polyhedra are defined as shown in eq. (8), where $\varepsilon \in \mathbb{R}^+$ stands for a small positive number chosen according to the numerical accuracy of the algorithm.

$$\Omega_i \equiv \left\{ \begin{bmatrix} x(\cdot) \\ u(\cdot) \end{bmatrix} ; \begin{array}{l} H_i x(\cdot) + J_i u(\cdot) \leq K_i, \\ \tilde{H}_i x(\cdot) + \tilde{J}_i u(\cdot) \leq \tilde{K}_i + \varepsilon \end{array} \right\} \tag{8}$$

III. Identification of a PWA model

A. Fuzzy clustering

When identifying PWA models we often have to face the fact that we do not know the dynamics of the system well enough to determine the suitable input-state space partitions, which make up the basis of the PWA model. This means that we do not know the suitable polyhedra, which is a prerequisite for estimating the parameters of the PWA model. In such a case we can make use of fuzzy clustering algorithms, such as *fuzzy c-means clustering*.

Fuzzy clustering is carried out over the *input-output* space of the PWA model in order to partition the identification data into several *fuzzy clusters*. Every single piece of identification data, i.e., a point in the input-output space of the PWA model, is a member of a particular fuzzy cluster with a certain membership degree, which is calculated according to the distance of the point from the centers of the fuzzy clusters, which are determined in every step of the algorithm.

The *fuzzy c-means clustering* is based on the minimization of a criterion given in eq. (9).

$$J_{MR} = \sum_{i=1}^{N} \sum_{j=1}^{C} \mu_{ij}^m \|x_i - c_j\|^2 \tag{9}$$

In eq. (9), m represents a predefined real number that satisfies the following inequality: $1 \leq m < \infty$. N stands for the number of pieces of identification data, i.e., the number of points in the *input-output* space of the PWA model; C denotes the number of clusters, μ_{ij} represents the value of the membership function of cluster j for the ith data point x_i. c_j denotes the center of cluster j; $\|\cdot\|$ is the norm, which defines the degree of dissimilarity between the center of the cluster c_j and the data point x_i.

Usually, the Euclidean norm is used, as given in eq. (10).

$$\|x\| = \sqrt{x^T \cdot x} \tag{10}$$

Fuzzy clustering is conducted iteratively: in every step of the algorithm the values of the membership functions μ_{ij} and the centers of the clusters c_j are calculated, as shown in eqs. (11) and (12), respectively.

$$\mu_{ij} = \frac{1}{\sum_{k=1}^{C} \left(\frac{\|x_i - c_j\|}{\|x_i - c_k\|} \right)^{\frac{2}{m-1}}},$$

$$\text{so that for every } \; i \in \{1, \ldots, N\} \; \text{ holds } \; \sum_{j=1}^{C} \mu_{ij} = 1. \tag{11}$$

$$c_j = \frac{\sum_{i=1}^{N} \mu_{ij}^m \cdot x_i}{\sum_{i=1}^{N} \mu_{ij}^m} \tag{12}$$

In eqs. (11) and (12), i denotes the index of a particular point in the identification data; j and k stand for the index of a particular fuzzy cluster and its center, respectively. The value of the parameter m defines the *fuzzyness* or. *crispness* of the distribution of the fuzzy membership functions in space.

In the extreme case that the parameter m is set to $m = 1$, the membership functions degenerate into crisp degrees of membership. The range of the membership functions is therefore limited to two values only: $\mu_{ij} \in \{0, 1\}$. From eq. (11) we can see that the value $\mu_{ij} = 1$ if the norm $\|x_i - c_j\|$ for the ith identification point and the cluster center j is the smallest comparing to the other centers of clusters. As for the other clusters, the value $\mu_{ik} = 0$, where $k \in \{1, 2, \ldots, C\} \backslash j$.

On the other hand, in the the extreme case that the parameter m is set to $m = \infty$, the membership functions degenerate into completely fuzzy degrees of membership. The values of the membership functions are equal across the whole space: $\mu_{ij} = \frac{1}{C}$ for every $j \in \{1, 2, \ldots, C\}$.

Usually, the parameter m is set to either $m = 1.25$ or $m = 2$.

The *fuzzy c-means clustering* algorithm can be described with the following steps.

1) Set the number of clusters C and the parameter m and establish the initial membership matrix $\Upsilon(0) = [\mu_{ij}]$.
2) In kth iteration determine the centers of the clusters c_j for $j = 1, \ldots, C$ according to $\Upsilon(k)$.
3) Calculate the new membership matrix $\Upsilon(k+1)$.
4) If $\|\Upsilon(k+1) - \Upsilon(k)\| < \varepsilon$ stop the algorithm, otherwise continue from step 2.

B. Estimation of the parameters by means of a modified least-squares method

A fuzzy system with a common consequence structure can be expressed as a global linear model. The input-dependent parameters are given in eq. (14), where $\boldsymbol{\beta}(k)$ represents the normalized degrees of fulfilment (for details see [7]).

$$\hat{y}_p(k+1) = \tilde{\boldsymbol{\Theta}}(k)^T \, \boldsymbol{\psi}(k) \tag{13}$$

$$\tilde{\boldsymbol{\Theta}}(k) = \boldsymbol{\Theta}(k) \, \boldsymbol{\beta}(k)^T \tag{14}$$

The regression matrix $\boldsymbol{\Psi}_{jd}$ for the rule $\mathbf{R}^{jd}$ in eq. (15) is obtained by using the whole set of input data for the hybrid fuzzy system. Here, index k runs from k_1 to k_{Pjd}, where P_{jd} denotes the number of input-output data pairs corresponding to the rule $\mathbf{R}^{jd}$.

However, only data from time-steps k that comply with the conditions in eqs. (16) and (17) are actually used for constructing the regression matrix $\boldsymbol{\Psi}_{jd}$. Here, δ denotes a small positive number. Since the model parameters are obtained by matrix inversion (described later in this section), compliance with eq. (17) is essential for obtaining suitably conditioned matrices.

$$\boldsymbol{\Psi}_{jd} = \begin{bmatrix} \beta_j(k_1) \; \psi^T(k_1) \\ \vdots \\ \beta_j(k_{Pjd}) \; \psi^T(k_{Pjd}) \end{bmatrix} \tag{15}$$

$$q(k) = d \tag{16}$$

$$\beta_j(k) \geq \delta \tag{17}$$

The output variable of the system y is included in the output data vector $\boldsymbol{Y}_{jd}$, which corresponds to the rule $\mathbf{R}^{jd}$, as written in eq. (18). Again, only data from time-steps $(k+1)$ that comply with the conditions in eqs. (16) and (17) are actually used for constructing the output data vector $\boldsymbol{Y}_{jd}$.

$$\boldsymbol{Y}_{jd} = \begin{bmatrix} \beta_j(k_1) \; y(k_1+1) \\ \vdots \\ \beta_j(k_1) \; y(k_{Pjd}+1) \end{bmatrix} \tag{18}$$

The output contribution $\hat{y}_p^{jd}(k+1)$ corresponding to the rule $\mathbf{R}^{jd}$ is written in eq. (19).

$$\beta^j(k_1) \; \hat{y}_p^{jd}(k+1) = \boldsymbol{\Theta}_{jd}^T \; (\beta^j(k_1) \; \psi(k)) \tag{19}$$

According to eqs. (15), (18) and (19), the parameters for the rule $\mathbf{R}^{jd}$ can be obtained using the least-squares identification method as written in eq. (20).

$$\boldsymbol{\Theta}_{jd} = (\boldsymbol{\Psi}_{jd}^T \boldsymbol{\Psi}_{jd})^{-1} \boldsymbol{\Psi}_{jd}^T \boldsymbol{Y}_{jd} \tag{20}$$

By calculating the parameters for the whole set of rules $\mathbf{R}^{jd}$; $j = 1, ..., K$ and $d = 1, ..., s$, the model is finally established.

The parameters are estimated on the basis of measured input-output data using the least-squares identification method. The approach is based on decomposition of the data matrix $\boldsymbol{\Psi}$

into $K \cdot s$ submatrices $\mathbf{\Psi}_{jd}$. Hence, the parameters for each rule $\mathbf{R}^{jd}$ ($j = 1, ..., K$ and $d = 1, ..., s$) are calculated separately. Due to better conditioning of the submatrices $\mathbf{\Psi}_{jd}$, compared to the conditioning of the whole data matrix $\mathbf{\Psi}$, this approach leads to a better estimate of the hybrid fuzzy parameters, or to put it in another way, the variances of the estimated parameters are smaller compared to the classic approach given in the literature [1], [2], [18], [19].

The described instantaneous linearization generates the parameters of the global linear model (see eq. (14)), which depends on the antecedents of the system $q(k),\ y(k), ..., y(k-n+1),\ u(k), ..., u(k-m+1)$. In the case of MPC, the global linear parameters can be used directly to predict the behavior of the system. In this case, the controller has to adapt to the dynamic changes online.

IV. Batch reactor

The presented identification method for systems that can be formulated as PWA models has been tested on a simulation example of a real batch reactor [8] that is situated in a pharmaceutical company and is used in the production of medicines. The goal is to control the temperature of the ingredients stirred in the reactor core so that they synthesize into the final product. In order to achieve this, the temperature has to follow the reference trajectory given in the recipe as accurately as possible. In addition, the temperature in the reactor's water jacket should be constrained between a minimum and maximum value. A scheme of the batch reactor is shown in fig. 1.

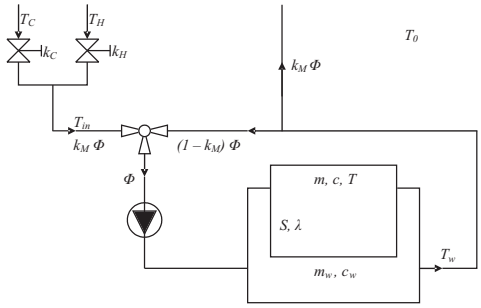

Figure 1. Scheme of the batch reactor

The control demands can be achieved using a model predictive control strategy. However, in order to implement such an approach, a suitable model of the system is needed. Therefore, we develop a PWA model of the batch reactor using the proposed identification approach.

A. Modelling and identification

In order to identify the PWA model the batch reactor we need appropriate input-output signals that enable the estimation of the dynamics of the system. The input signals have been generated using a pseudorandom generator, whereas the output signals are represented by the recorded responses of the system (for detailed information see [7]).

The model of the batch reactor is derived in several steps.

- First, we split the multivariable system into two simpler subsystems with multiple inputs and a single output (MISO).
- Taking into account the influence of the outputs on both subsystems we establish the structure of the submodels for each subsystem.
- We identify each subsystem using the method described in section III.

According to heat flows that occur in the batch reactor we can split the system into two subsystems, which primarily deal with:

- the temperature in the core of the batch reactor T;
- the temperature in the water jacket of the batch reactor T_w.

In this manner we take advantage of the prior knowledge of the structure of the system: we conduct a sort of a *grey-box identification*, which presents a compromise between a *black-box identification* and pure *theoretical modelling*.

The temperature in the core of the batch reactor T depends only on the heat conduction between the core and the water-jacket of the batch reactor.

We are therefore dealing with a MISO model as shown in eq. (21). The regressor consists of the temperature in the water jacket $T_w(k)$ and in the core $T(k)$ of the batch reactor in the actual time-step k.

$$\hat{T}(k+1) = f(T_w(k), T(k)) \tag{21}$$

We assume that the heat flow is proportional to difference between the temperature in the water jacket $T_w(k)$ and in the core $T(k)$ of the batch reactor. Hence, we can derive a linear 1st-order model as shown in eq. (22).

$$\hat{T}(k+1) = \boldsymbol{\theta}^T \left[T_w(k)\ \ T(k)\right]^T \tag{22}$$

After conducting a least-squares estimation we obtain the following parameters.

$$\boldsymbol{\theta} = [0.0033\ \ 0.9967]^T \tag{23}$$

The temperature in the water jacket of the batch reactor T_w depends on the heat flow between the water jacket and the core and between the water jacket and the surroundings. In addition, we have to take into account the heat flow due to inflow and outflow of the water in the jacket of the batch reactor.

We are therefore dealing with a MISO model as shown in eq. (24). The regressor consists of the temperature in the water jacket $T_w(k)$ and in the core $T(k)$ of the batch reactor and the input signals, i.e., the position of the mixing valve $k_M(k)$ and the hot- $k_H(k)$ and cold-water valves $k_C(k)$ in the actual time-step k.

$$\hat{T}_w(k+1) = F(T_w(k), T(k), k_M(k), k_C(k), k_H(k)) \tag{24}$$

Since the dynamics concerning the temperature the water jacket of the batch reactor $T_w(k)$ involve both hybrid and nonlinear properties, the submodel will be formulated as a PWA model.

A general modelling and identification procedure is introduced in sections II in III. By following the aforementioned procedures we obtain the following PWA model of the batch reactor.

The first divison of the input space of the PWA model $\mathcal{D}_{[T_w,T,k_M,k_C,k_H]^T}$ is according to the operating mode of the batch reactor, i.e. according to the position of the on/off valves.

- First mode: the fresh input water is hot ($T_{in} = T_H$, i.e. $k_C(k) = 0$ and $k_H(k) = 1$).
- Second model: the fresh input water is cool ($T_{in} = T_C$, i.e. $k_C(k) = 1$ and $k_H(k) = 0$).

Next, the remaining input subspaces for each operating mode of the PWA model $\mathcal{D}_{[T_w,T,k_M]^T}$ are further partitioned so that the final model takes into account the nonlinear dynamics of the system by using several affine models, one for each partition. Therefore, we obtain suitable convex polyhedra that cover the whole operating range of the system. Every particular convex polyhedron defines a partition in which the dynamics of the system is approximated by an affine model.

The partitioning of is carried out using the fuzzy c-means clustering algorithm – see section III – over the input-output space of the PWA model. In such manner we assign a particular membership value for each cluster to every identification data point. The membership value is based on the distance of the identification point from the center of a particular cluster. The goal is to obtain suitable clusters that can be represented by affine models and limited by convex polyhedra in the input space of the PWA model.

Before conducting the clustering, the identification data is normalized, i.e. scaled so that they are limited to the interval $[0, 1]$. The clustering algorithm is carried out in the normalized input-output space of the PWA model given in eq. (25).

$$\mathcal{D}_{[\breve{T}_w(\tau),\breve{T}(\tau),\breve{k}_M(\tau)]^T} \times \mathcal{D}_{\breve{T}_w(\tau+1)} \equiv [0, 1]^4 \tag{25}$$

The notation $\breve{}$ stands for the normalized space.

In our case the parameter m was set to $m = 1,25$.

The clustering algorithm partitions the identification data according to the input variables $T_w(\tau)$, $T(\tau)$ and $k_M(\tau)$ and the output variable $T_w(\tau + 1)$ into fuzzy clusters. The index τ denotes the time-step of a particular identification point.

The clustering algorithm also defines the centers of the clusters in the normalized space (25) $\breve{c}_j$, where $j \in \{1, 2, \ldots, C\}$, and every identification point is assigned C values of membership functions. In our case, we define $C = 5$ fuzzy clusters for every operating mode.

The centers of the clusters (and the corresponding membership functions) that are returned by the clustering algorithm, are defined in the *input-output* space of the PWA model (25). The membership functions that are defined in such a manner can be directly used for parameter estimation of a PWA model. However, such a definition is not usable for predicting the bahaviour of the system in MPC strategies. When using the PWA model for prediction, we are not able to determine these distances, because we are primarily dealing with a vector in the *input* space of the PWA model.

Therefore, we have to project the centers of the fuzzy clusters into the input space of the PWA model $\mathcal{D}_{[\breve{T}_w,\breve{T},\breve{k}_M)]^T}$.

The centers of the fuzzy clusters in the normalized input space of the PWA model for $C = 5$ are given in eq. (26). The results are given for both operating modes:

- the centers from $\breve{c}_1$ to $\breve{c}_5$ for $k_C = 0$ and $k_H = 1$;
- the centers from $\breve{c}_6$ to $\breve{c}_{10}$ for $k_C = 1$ in $k_H = 0$.

$$\begin{aligned}
\breve{c}_1 &= [0,9495 \;\; 0,5507 \;\; 1,0000] \\
\breve{c}_2 &= [0,4880 \;\; 0,4765 \;\; 0,0305] \\
\breve{c}_3 &= [0,6715 \;\; 0,6935 \;\; 0,0500] \\
\breve{c}_4 &= [0,1148 \;\; 0,0980 \;\; 0,0053] \\
\breve{c}_5 &= [0,3180 \;\; 0,2655 \;\; 0,0218] \\
\breve{c}_6 &= [0,3718 \;\; 0,5257 \;\; 0,0328] \\
\breve{c}_7 &= [0,2180 \;\; 0,2939 \;\; 0,0279] \\
\breve{c}_8 &= [0,0818 \;\; 0,1010 \;\; 0,0412] \\
\breve{c}_9 &= [0,0147 \;\; 0,1273 \;\; 1,0000] \\
\breve{c}_{10} &= [0,0562 \;\; 0,4671 \;\; 1,0000]
\end{aligned} \tag{26}$$

According to the results of the clustering algorithm it is possible to partition the input space of the PWA model $\mathcal{D}_{[T_w,T,k_M]^T}$ into convex polyhedra defined by the separating hyperplanes[1]. For every pair of partitions Ω_i and Ω_j we can define a separating hyperplane as shown in eq. (27).

$$(\breve{x} - \breve{s}_{ij}) \cdot \breve{n}_{ij} = 0 \tag{27}$$

In eq. (27) $\breve{x}$ denotes a general point in the normalized input-output space on the separating hyperplane and $\breve{n}_{ij}$ stands for the normal vector of the hyperplane. Without losing the generality we can assume that $\breve{s}_{ij}$ denotes the point on the separating hyperplane that lies on the straight line between the centers of the i-h and j-th clusters $\breve{c}_i$ and $\breve{c}_j$.

$\breve{s}_{ij}$ and $\breve{n}_{ij}$ can be derived from the centers of the clusters as showni in eqs. (28) and (29).

$$\breve{s}_{ij} = \frac{\breve{c}_i + \breve{c}_j}{2} \tag{28}$$

$$\breve{n}_{ij} = \breve{c}_j - \breve{c}_i \tag{29}$$

In this manner, the input space of the PWA model is divided into subspaces Ω_i as shown in eq. (30). For details see [7].

$$\begin{aligned}
&[T_w(k), T(k), k_M(k), k_C(k), k_H(k)]^T \in \Omega_i; \\
&\text{if} \quad \begin{matrix} H_i \; T_w(k) \; + \; J_i \; [T(k), k_M(k), k_C(k), k_H(k)]^T \leq K_i \\ \tilde{H}_i \; T_w(k) \; + \; \tilde{J}_i \; [T(k), k_M(k), k_C(k), k_H(k)]^T < \tilde{K}_i \end{matrix} \\
&\text{for } i = 1, \; \ldots, \; 10
\end{aligned} \tag{30}$$

For every partition Ω_i of the input space of the PWA model, we can derive a local affine model as in eq. (31).

$$\begin{aligned}
&T_w(k+1) = A_i T_w(k) + B_i \begin{bmatrix} T(k) \\ k_M(k) \end{bmatrix} + f_i \\
&\text{for } [T_w, T, k_M, k_C, k_H]^T \in \Omega_i
\end{aligned} \tag{31}$$

The membership-function values used in parameter estimation described in section III are obtained from the results of the fuzzy clustering algorithm. They are based on the distance of

[1]According to the fact that in this case we are dealing with partitioning of a 3-dimensional input space $\mathcal{D}_{[T_w,T,k_M]^T}$ the resulting hyperplanes are defined in 3 dimensions.

the identification points from the centers of particular clusters. The parameters of the PWA model are given in eq. (32).

$$\begin{aligned}
&\Omega_1: \ A_1 = 0,6312; \ B_1 = [0,0400 \ 1,7115]; \ f_1 = 22,4874\\
&\Omega_2: \ A_2 = 0,9257; \ B_2 = [0,0513 \ 12,5240]; \ f_2 = 0,7210\\
&\Omega_3: \ A_3 = 0,9361; \ B_3 = [0,0384 \ 6,9758]; \ f_3 = 1,0316\\
&\Omega_4: \ A_4 = 0,9404; \ B_4 = [0,0490 \ 19,3915]; \ f_4 = 0,1955\\
&\Omega_5: \ A_5 = 0,9277; \ B_5 = [0,0515 \ 16,8357]; \ f_5 = 0,5233\\
&\Omega_6: \ A_6 = 0,9059; \ B_6 = [0,0630 \ -10,8609]; \ f_6 = 0,8869\\
&\Omega_7: \ A_7 = 0,9337; \ B_7 = [0,0468 \ -4,5854]; \ f_7 = 0,4228\\
&\Omega_8: \ A_8 = 0,9462; \ B_8 = [0,0359 \ -1,5090]; \ f_8 = 0,3163\\
&\Omega_9: \ A_9 = 0,6279; \ B_9 = [0,0404 \ -15,3616]; \ f_9 = 19,3905\\
&\Omega_{10}: \ A_{10} = 0,6312; \ B_{10} = [0,0400 \ -14,9129]; \ f_{10} = 18,9084
\end{aligned} \tag{32}$$

B. Validation

We have validated the obtained PWA model by comparing its responses to the responses of the original batch reactor model. The input signals have been generated using a pseudo-random generator (for detailed information see [7]). We have recorded both measurable outputs, i.e., the temperature in the core T and the temperature in the water jacket of the batch reactor T_w.

Figure 2 shows a closeup of the trajectory of temperature in the water jacket of the batch reactor T_w obtained by a simulation using the PWA model of the batch reactor. The dotted line represents the original response of the batch reactor to the input signals.

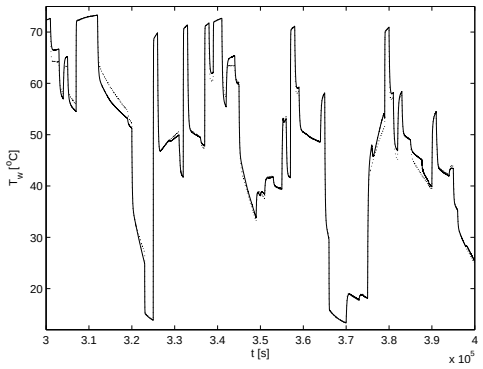

Figure 2. The response of the PWA model to the validation input signals (solid line) and the original response of the batch reactor (dotted line).

For validation purposes, we can calculate the following parameters that reflect the identification quality.

- The average squared discrepancy of the temperature in the water jacket of the batch reactor T_w: $\bar{J}_{T_w} = 1.2735$.
- The average squared discrepancy of the temperature in the core of the batch reactor T: $\bar{J}_T = 0.6780$.

V. Conclusion

The complex hybrid and nonlinear nature of many processes that are met in practice causes problems with both structure modelling and parameter identification; therefore, obtaining a model that is suitable for MPC is often a difficult task. The PWA model represents a widely used framework for modelling complex systems for control purposes in practice. However, it is often difficult to identify a complex nonlinear hybrid system and formulate it as a PWA model.

The identification method presented in this paper strives to overcome this obstacle by using a fuzzy clustering algorithm for identification purposes and project the resulting clusters defined in the *input-output* space of the PWA model into the *input* space of the PWA model. In this manner, we can obtain a PWA model suitable for model predictive control purposes.

We verified the identification approach on a hybrid nonlinear batch reactor example. The result suggest that the batch reactor can be efficiently identified and formulated as a PWA model, which can eventually be used in a model predictive control algorithm.

References

[1] R. Babuška. *Fuzzy Modelling for Control*. KAP, 1998.
[2] R. Babuška and H. B. Verbruggen. An overview of fuzzy modelling for control. *Control Engineering Practice*, 4(11):1593–1606, 1996.
[3] A. Bemporad. Efficient conversion of mixed logical dynamical systems into an equivalent piecewise affine form. *IEEE Trans. Automatic Control*, 49(5):832–838, 2004.
[4] A. Bemporad and M. Morari. Control of systems integrating logic, dynamics and constraints. *Automatica*, 35(3):407–427, 1999.
[5] D. Girimonte and R. Babuška. Structure for nonlinear models with mixed discrete and continuous inputs: a comparative study. In *Proc. of IEEE International Conf. on system, Man and Cybernetics*, pages 2392–2397, 2004.
[6] W. P. M. H. Heemels, B. De Schutter, and A. Bemporad. Equivalence of hybrid dynamical models. *Automatica*, 37(7):1085–1091, 2001.
[7] G. Karer. *Prediktivno vodenje hibridnih sistemov*. PhD thesis, Fakulteta za elektrotehniko, Univerza v Ljubljani, 2009.
[8] G. Karer, G. Mušič, I. Škrjanc, and B. Zupančič. Hybrid fuzzy model-based predictive control of temperature in a batch reactor. *Computers and Chemical Engineering*, 31:1552–1564, 2007.
[9] E. C. Kerrigan and D. Q. Mayne. Optimal control of constrained, piecewise affine systems with bounded disturbances. In *Proc. 41st IEEE Conference on Decision and Control*, Las Vegas, 2002.
[10] D. Q. Mayne and S. Raković. Model predictive control of constrained piecewise affine discrete-time systems. *International Journal of Robust and Nonlinear Control*, 13(3):261–279, 2003.
[11] R. Palm and D. Driankov. Fuzzy switched hybrid systems – modelling and identification. In *Proc. of the 1998 IEEE/ISCI/CIRA/SAS Joint Conf., Gaithersburg MD*, pages 130–135, 1998.
[12] B. Potočnik, G. Mušič, and B. Zupančič. A new technique for translating discrete hybrid automata into piecewise affine systems. *Math. comput. model. dyn. syst.*, 10(1):41–57, 2004.
[13] B. Potočnik, G. Mušič, and B. Zupančič. Model predictive control of discrete time hybrid systems with discrete inputs. *ISA Transactions*, 44(2):199–211, 2005.
[14] Y. Qin and L.-M. Jia. Fuzzy hybrid control and its application in complex combustion processes. In *2002 IEEE International Conference on Artificial Intelligence Systems (ICAIS'02)*, page 78. IEEE, 2002.
[15] D. Sáez, C. E. Cortés, and A. Núñez. Hybrid adaptive predictive control for the multi-vehicle dynamic pick-up and delivery problem based on genetic algorithms and fuzzy clustering. *Computers and Operations Research*, 35(11):3412 – 3438, 2008.
[16] I. Škrjanc and D. Matko. Fuzzy predictive functional control in the state space domain. *Journal of Intelligent and Robotic Systems*, 31:283–297, 2001.
[17] E. Sontag. Nonlinear regulation: The piecewise linear approach. *IEEE Transactions on Automatic control*, 26(2):346–358, 1981.
[18] M. Sugeno and K. Tanaka. Successive identification of a fuzzy model and its application to prediction of a complex system. *Fuzzy Sets and Systems*, 42:315–334, 1991.
[19] T. Takagi and M. Sugeno. Fuzzy identification of systems and its application to modelling and control. *IEEE Trans. System Man Cybernet.*, 15:116–132, 1985.
[20] A. Van der Schaft and H. Schumacher. An introduction to hybrid dynamical systems. *Lecture Notes in Control and Information Sciences*, 251:v–vii, 1999.
[21] H. S. Witsenhausen. A class of hybrid-state continuous time dynamic systems. *IEEE Trans. on Automatic Control*, 11(2):161–167, 1966.