IDENTIFICATION OF A HYBRID FUZZY MODEL OF A BATCH REACTOR FOR SIMULATION AND CONTROL

Gorazd Karer[1], Igor Škrjanc[1], Borut Zupančič[1]

[1]University of Ljubljana, Faculty of Electrical Engineering
1000 Ljubljana, Tržaška 25, Slovenia

gorazd.karer@fe.uni-lj.si(Gorazd Karer)

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. In this paper we focus on using the hybrid fuzzy model formulation. The framework is suitable for modelling nonlinear hybrid systems and can be implemented in model predictive control design. The basic idea of this paper is to present an identification method for a hybrid fuzzy model based on a fuzzy clustering algorithm. In the paper, we first introduce the hybrid fuzzy model. We present the hierarchical structure and the generalization of the Takagi-Sugeno formulation for the nonlinear hybrid system and give the output of the hybrid fuzzy model in a compact form. 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 hybrid fuzzy model and explain the estimation of the parameters of the hybrid fuzzy 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 hybrid fuzzy model, which can eventually be used for model predictive control purposes.

Keywords: Identification, Hybrid fuzzy model, Batch reactor, Hybrid systems, Nonlinear systems.

Presenting Author's Biography

Gorazd Karer received his B.Sc. and Ph.D. degrees in 2004 and 2009, respectively, from the Faculty of Electrical Engineering, University of Ljubljana. He is currently a researcher and assistant at the same institution. His research interests are in hybrid and nonlinear systems and model predictive control.

1 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 [1]. 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 to 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.

MPC methods for hybrid systems employ several model formulations. Often the system is described as *mixed logical dynamical* (MLD) [2]. A lot of interest has also been devoted to *piecewise affine* (PWA) formulation [3], which has been proven to be equivalent to many classes of hybrid systems [4]. 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 [5]. On the other hand, in [6] 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 [7].

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 [8]. 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 [9] is introduced in [10]. Furthermore, two structure-selecting methods for nonlinear models with mixed discrete and continuous inputs are presented in [11]. In [12] a fuzzy control method is implemented in the low control-level for a class of hybrid systems based on hybrid automata.

In this paper we focus on using the hybrid fuzzy model formulation presented in [13]. The framework is suitable for modelling nonlinear hybrid systems and can be implemented in model predictive control design. The basic idea of this paper is to present an identification method for a hybrid fuzzy model based on a fuzzy clustering algorithm.

The outline of the paper is as follows. Section 2 introduces the hybrid fuzzy model. Next, in section 3 the identification method is explained. We verify the usability of the proposed identification approach on a nonlinear hybrid batch reactor example in section 4. Finally, we give some concluding remarks.

2 Modelling of a hybrid fuzzy model

Dynamic systems are usually modelled by feeding back delayed input and output signals. In the discrete-time domain a common nonlinear model structure is the NARX (Nonlinear AutoRegressive with eXogenous inputs) model [14], which gives the mapping between the past input-output data and the predicted output.

$$\begin{aligned}\hat{y}_p(k+1) &= F(y(k), y(k-1), ..., \\ &y(k-n+1), u(k), u(k-1), ..., u(k-m+1))\end{aligned} \quad (1)$$

Here, $y(k), y(k-1), ..., y(k-n+1)$ and $u(k), u(k-1), ..., u(k-m+1)$ denote the delayed process output and input signals, respectively. Hence, the model of the system is represented by the (nonlinear) function F.

2.1 Hybrid system hierarchy

As already mentioned, many processes met in practice demonstrate a hybrid nature, which means that the continuous dynamics are interlaced with the discrete dynamics. A special class of such systems is called switched systems, where the continuous states remain continuous even when the discrete states are changed, i.e. no jumps of the continuous state vector are allowed. In this paper we deal with hybrid systems represented by a hierarchy of discrete and continuous subsystems where the discrete part is atop the hierarchy. A discrete-time formulation is described in eqs. (2) and (3).

$$\mathbf{x}(k+1) = \mathbf{f}_q(\mathbf{x}(k), \mathbf{u}(k)) \qquad (2)$$

$$q(k) = g(\mathbf{x}(k), q(k-1), \mathbf{u}(k)) \qquad (3)$$

Here, $\mathbf{x} \in \mathbb{R}^n$ is the continuous state vector, which includes all relevant system outputs y (see eq. (1)), i.e. measurable continuous states (delayed and non-delayed) that influence the state vector in the next time-step. $\mathbf{u} \in \mathbb{R}^m$ denotes the input vector. $q \in \mathbb{Q}$ (where $\mathbb{Q} = \{1, ..., s\}$) is the discrete state, which defines the switching region. Discrete states are also referred to as operating modes. There are s operating modes of the hybrid system. The hybrid states are hence described at any time-step k by the set of states $(\mathbf{x}(k), q(k))$ in the domain $\mathbb{R}^n \times \mathbb{Q}$.

The local behavior of the model described in eq. (2) depends on the discrete state $q(k)$, which defines the current function $\mathbf{f}_q$.

Eq. (3) introduces a modification of the strict Witsenhausen hybrid system formulation [9] in the sense that the discrete state $q(k)$ depends on the input vector $\mathbf{u}(k)$ as well as on the continuous state vector $\mathbf{x}(k)$ and the previous discrete state $q(k-1)$.

The continuous part of the system is generally nonlinear, therefore it can be modelled as a Takagi-Sugeno fuzzy model, as shown in subsection 2.2.

2.2 Generalization of the Takagi-Sugeno formulation for a nonlinear hybrid system

In order to approximate a nonlinear system, a fuzzy formulation can be employed. Fuzzy models can be regarded as universal approximators, which can approximate continuous functions to an arbitrary precision [15, 16].

The system dynamics can be formulated as a Takagi-Sugeno fuzzy model. In order to address nonlinear hybrid systems, we have generalized the model formulation by incorporating the discrete part of the system dynamics given in eq. (3) in the rule base. In this instance, the rule base of the hybrid fuzzy system is represented in eq. (4).

$$\begin{aligned}
&\mathbf{R}^{jd}: \\
&\quad \textbf{if } q(k) \text{ is } Q_d \text{ and } y(k) \text{ is } A_1^j \text{ and ... and} \\
&\qquad y(k-n+1) \text{ is } A_n^j \\
&\quad \textbf{then } \hat{y}_p(k+1) = f_{jd}(y(k), ..., y(k-n+1), \\
&\qquad u(k), ..., u(k-m+1))
\end{aligned} \qquad (4)$$

for $j = 1, ..., K$ and $d = 1, ..., s$

The **if**-parts (antecedents) of the rules describe hybrid fuzzy regions in the space of the input variables of the hybrid fuzzy model. Here, $q(k) \in \{1, ..., s\}$ stands for the discrete state of the nonlinear hybrid system, i.e., its operating mode. Q_d and A_i^j represent (fuzzy) sets characterized by their crisp and fuzzy membership functions, respectively.

The number of relevant rules in the hybrid fuzzy model is $K \cdot s$. Generally speaking, K depends on the number of fuzzy membership functions for each antecedent variable $y(k), ..., y(k-n+1), u(k), ..., u(k-m+1)$. The membership functions have to cover the whole operating area of the system. What is more, the rules have to distinguish all possible combinations of the membership functions in the antecedent variable space. Hence, K is a product of the number of membership functions corresponding to each antecedent variable $y(k), y(k-1), ..., y(k-n+1), u(k), ..., u(k-m+1)$. Note that there are K fuzzy sets A_i^j as the appurtenant membership functions are the same for every rule $\mathbf{R}^{jd}$, regardless of d. This means that the fuzzy partitioning of the state-space is the same, regardless of the current discrete state (operating mode) of the system. In other words, the normalized degrees of fulfillment are calculated only from the continuous states of the system.

On the other hand, s denotes the number of operating modes of the nonlinear hybrid system, which is also the number of crisp membership functions characterizing the sets Q_d. The number of operating modes depends on the partitioning of the state-space and the number of discrete inputs. For instance, in case we have 2 discrete input variables and each variable can have 4 discrete values, the number of operating modes (due to discrete inputs) is 8. However, if there are some infeasible (unwanted or unneeded) input combinations, the number of operating modes of a hybrid fuzzy system is appropriately reduced.

The **then**-parts (consequences) are functions of the inputs of the hybrid fuzzy model. Here, $\hat{y}_p(k+1)$ is an output variable representing the predicted output of the process in the next time step (see eq. (1)). When applying the Takagi-Sugeno formulation MPC purposes, $\hat{y}_p(k+1)$ can also be regarded as the predicted state of the system $\hat{x}(k+1)$ (see eq. (2)). There is one function of inputs f_{jd} defined for each rule $\mathbf{R}^{jd}$; $j = 1, ..., K$ and $d = 1, ..., s$ in the hybrid fuzzy model. In general, f_{jd} can be a nonlinear function. However, usually an affine function f_{jd} is used, as shown in eq. (5).

$$\begin{aligned}
&f_{jd}(y(k), ..., y(k-n+1), u(k), ..., u(k-m+1)) = \\
&\quad = a_{1jd}\, y(k) + ... + a_{njd}\, y(k-n+1) + \\
&\qquad + b_{1jd}\, u(k) + ... + b_{mjd}\, u(k-m+1) + r_{jd}
\end{aligned} \qquad (5)$$

In this case f_{jd} determines the output, while $a_{1jd}, ..., a_{njd}, b_{1jd}, ..., b_{mjd}$ and r_{jd} denote consequent parameters, all corresponding to the rule $\mathbf{R}^{jd}$.

The output of the hybrid fuzzy model in a compact form is given by the following equation.

$$\hat{y}_p(k+1) = \boldsymbol{\beta}(k)\ \boldsymbol{\Theta}^T(k)\ \boldsymbol{\psi}(k) \tag{6}$$

Here, $\boldsymbol{\beta}(k)$ represents the normalized degrees of fulfillment for the whole set of fuzzy rules ($j = 1, ..., K$) in the current time-step k, written in the vector form $\boldsymbol{\beta}(k) = [\beta_1(k)\ \beta_2(k)\ \dots\ \beta_K(k)]$. We assume the normalized degrees of fulfillment, which are generally time-dependent, comply with eq. (7) for every time-step k.

$$\boldsymbol{\beta}(k)\boldsymbol{I} = \sum_{j=1}^{K} \beta_j(k) = 1 \tag{7}$$

Here, $\boldsymbol{I}$ is the unity vector.

The normalized degree of fulfillment $\beta_j(k)$ corresponding to a set of rules $\mathbf{R}^{jd}$ for every $d = 1, ..., s$ is obtained by using a T-norm [17]. In our case it is a simple algebraic product, given in eq. (8).

$$\beta_j(k) = \frac{\mu_{A_1^j}(y(k)) \cdot \dots \cdot \mu_{A_n^j}(y(k-n+1))}{\sum_{i=1}^{K} \mu_{A_1^i}(y(k)) \cdot \dots \cdot \mu_{A_n^i}(y(k-n+1))} \tag{8}$$

Here, $\mu_{A_1^j}(y(k))\ \dots\ \mu_{A_n^j}(y(k-n+1))$ denote the membership values [18, 19, 17].

In eq. (6), $\boldsymbol{\Theta}(k)$ denotes a matrix with $n+m+1$ rows and K columns, which contains the consequent fuzzyfied parameters of the hybrid fuzzy model in the current time-step k. As noted in eq. (9), $\boldsymbol{\Theta}(k)$ is actually a function of the discrete state of the hybrid fuzzy system in the current time-step $q(k)$.

$$\boldsymbol{\Theta}(k) = \boldsymbol{\Theta}(q(k)) = \left\{ \begin{array}{ccc} \boldsymbol{\Theta}_1 & \text{if} & q(k) = 1 \\ \vdots & & \vdots \\ \boldsymbol{\Theta}_s & \text{if} & q(k) = s \end{array} \right\} \tag{9}$$

The matrices $\boldsymbol{\Theta}_d$ contain the consequent fuzzyfied parameters of the hybrid fuzzy model for each operating mode ($q = d \in \{1, ..., s\}$), individually. We assume the set of matrices $\boldsymbol{\Theta}_d$ to be time-invariant.

Each matrix $\boldsymbol{\Theta}_d$ contains all the consequent fuzzyfied parameters of the hybrid fuzzy model for the set of hybrid fuzzy rules $\{\mathbf{R}^{jd}\}$, where d is fixed and $j = 1, ..., K$. $\boldsymbol{\Theta}_d$ is constructed as shown in eq. (10).

$$\boldsymbol{\Theta}_d^T = \begin{bmatrix} a_{11d} & \cdots & a_{n1d} & b_{11d} & \cdots & b_{m1d} & r_{1d} \\ \vdots & & \vdots & \vdots & & \vdots & \vdots \\ a_{1Kd} & \cdots & a_{nKd} & b_{1Kd} & \cdots & b_{mKd} & r_{Kd} \end{bmatrix} \tag{10}$$

In eq. (6), $\boldsymbol{\psi}(k)$ denotes a regressor in time-step k. The regressor contains all the relevant model inputs that are needed in f_{jd}. $\boldsymbol{\psi}(k)$ is constructed as shown in eq. (11).

$$\boldsymbol{\psi}(k) = \begin{bmatrix} y(k) \\ \vdots \\ y(k-n+1) \\ u(k) \\ \vdots \\ u(k-m+1) \\ 1 \end{bmatrix} \tag{11}$$

In general, hybrid fuzzy models can have multiple inputs and outputs (also known as multivariable models). In the case that the system has several outputs, the functions of the inputs f_{jd} can be regarded as vector functions. In modelling, however, we can concern ourselves only with single-output hybrid fuzzy models and, accordingly, presume f_{jd} to be a scalar function. In the case of modelling a multiple-output process, several models in parallel can be used instead, without any loss of generality. Furthermore, if the system has several inputs, the regression vector is simply extended so as to include all the relevant model inputs.

A similar approach can be taken into consideration when dealing with higher-than-first-order processes ($n > 1$). The regression vector therefore comprises all the system outputs from past time-steps $y(k-1),\ \dots,\ y(k-n+1)$ needed for predicting $\hat{y}_p(k+1)$. However, in the case that it is possible to measure the relevant system states, which can substitute the system outputs from the past time-steps $y(k-1),\ \dots,\ y(k-n+1)$ in order to predict $\hat{y}_p(k+1)$, it is generally more appropriate to employ several (simpler) first-order models running in parallel in place of a single nth-order model for MPC purposes. If such first-order models are not feasible, it is still suitable to employ several lower-than-nth-order models instead. To put it another way, it is generally reasonable to make use of all the available data measured in a single time-step. However, due to unmeasurable system states it is sometimes not possible to carry out such an approach.

3 Identification of a hybrid fuzzy model

3.1 Fuzzy clustering

When identifying a hybrid fuzzy model we often have to face the fact that we do not know the dynamics of the system well enough to determine the suitable fuzzy sets A_i^j, which make up the premise of the hybrid fuzzy model. This means that we do not know the suitable membership functions, which is a prerequisite for estimating the parameters of the hybrid fuzzy 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 hybrid fuzzy model $\mathcal{D}_{IO}$ in order to separate the identification data into several *fuzzy clusters*. Every single piece of identification data, i.e., a point in the input-output space of the hybrid fuzzy 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 particular 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. (12).

$$J_{MR} = \sum_{i=1}^{N} \sum_{j=1}^{C} \mu_{ij}^{m} \|x_i - c_j\|^2 \tag{12}$$

In eq. (12), 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 hybrid fuzzy model $\mathcal{D}_{IO}$; 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. (13).

$$\|x\| = \sqrt{x^T \cdot x} \tag{13}$$

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. (14) and (15), respectively.

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

$$\text{so that for every } i \in \{1, \ldots, N\} \text{ holds } \sum_{j=1}^{C} \mu_{ij} = 1. \tag{14}$$

$$c_j = \frac{\sum_{i=1}^{N} \mu_{ij}^{m} \cdot x_i}{\sum_{i=1}^{N} \mu_{ij}^{m}} \tag{15}$$

In eqs. (14) and (15), 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. (14) 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.

3.2 Projections of the fuzzy clusters into the input space of the hybrid fuzzy model

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 hybrid fuzzy model $\mathcal{D}_{IO}$. The membership functions that are defined in such a manner can be directly used for parameter estimation of a hybrid fuzzy model. However, such a definition is not usable for predicting the bahaviour of the system in model predictive control strategies. Namely, it is not possible to determine the normalized degrees of fulfilment for a particular fuzzy cluster $\beta_j(k)$, where $j \in \{1, 2, \ldots, C\}$ – see eq. (8) – because the values of the membership functions depend on the distances to the centers of the particular fuzzy clusters in the *input-output* space of the hybrid fuzzy model $\mathcal{D}_{IO}$ – see eq. (14). When using the hybrid fuzzy 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 hybrid fuzzy model $\mathcal{D}_I$.

The membership functions are established by means of fuzzy clustering in the *input-output* space of the hybrid fuzzy model $\mathcal{D}_{IO}$, therefore, we have to find a way to appropriately map the information on the membership functions from the *input-output* space $\mathcal{D}_{IO}$ to the *input* space of the hybrid fuzzy model $\mathcal{D}_I$, in order to make the model usable for predicting the behavior of the system.

As shown in eq. (14), the values of the membership functions μ_{ij} depend on the distances to the centers of particular fuzzy clusters in the *input-output* space of the hybrid fuzzy model $\mathcal{D}_{IO}$. Hence, we are able to determine the appropriate values of the membership functions μ_{ij} by knowing these distances. The problem is thus reduced to projecting the information on the distances to the centers of the particular fuzzy clusters from the *input-output* space $\mathcal{D}_{IO}$ into the *input* space of the hybrid fuzzy model $\mathcal{D}_I$.

Let us assume that the norm is defined as in eq. (13). The sets of points in the *input-output* space of the hybrid fuzzy model $\mathcal{D}_{IO}$ that are equidistant from a chosen point – in our case from the center of a particular cluster c_j – can be thus represented geometrically by a hypersphere with the center in the chosen point in the *input-output* space of the hybrid fuzzy model $\mathcal{D}_{IO}$. The radius of the hypersphere defines the distance from the chosen point. The dimension of the hypersphere is defined by the dimension of the *input-output* space $\mathcal{D}_{IO}$, i.e., by the number of inputs and outputs of the hybrid fuzzy model.

The mathematical definition of a hypersphere in the *input-output* space of the hybrid fuzzy model $\mathcal{D}_{IO}$ is given in eq. (16).

$$[x - c_j]^T \cdot [x - c_j] = r_j^2 = \|x - c_j\|^2, \quad \text{where } j \in \{1, 2, \ldots, C\}. \tag{16}$$

In eq. (16), vector x denotes a point in the *input-output* space of the hybrid fuzzy model $\mathcal{D}_{IO}$ lying on the hypersphere with the center in c_j. The radius of the hypersphere is calculated using the Euclidean norm $r_j = \|x - c_j\|$. The dimensions of the vectors x and c_j are defined by the dimension of the *input-output* space of the hybrid fuzzy model $\mathcal{D}_{IO}$. Index j, where $j \in \{1, 2, \ldots, C\}$, stands for the index of the particular fuzzy cluster we are dealing with.

As described in section 2.2, there is one function f_{jd} defined for every single rule $\mathbf{R}^{jd}$ in the hybrid fuzzy model. If we assume the function f_{jd} is affine, as shown in eq. (5), we can regard it as a representation of an affine submodel in the case that $\beta_j = 1$ and $\beta_k = 0$, for $k \neq j$, and $q = d$.

The sets of points in the *input-output* space of the hybrid fuzzy model $\mathcal{D}_{IO}$ that are defined by the affine function f_{jd} can be represented geometrically by a hyperplane in the *input-output* space of the hybrid fuzzy model $\mathcal{D}_{IO}$.

The mathematical definition of a hyperplane in the *input-output* space of the hybrid fuzzy model $\mathcal{D}_{IO}$ is given in eq. (17).

$$[x - s_j]^T \cdot n_j = 0, \quad \text{where } j \in \{1, 2, \ldots, C\}. \tag{17}$$

In eq. (17), vector x denotes a point in the *input-output* space of the hybrid fuzzy model $\mathcal{D}_{IO}$ lying on the hyperplane, which is defined by an arbitrary point s_j lying on the hyperplane and the normal vector n_j. For clarity reasons, it is possible to assume – without losing generality – that s_j is the point lying on the hyperplane (17), which is the closest to the corresponding center of the fuzzy cluster c_j. If the center of the fuzzy cluster c_j lies on the hyperplane (17), the point s_j coincides with the center of the fuzzy cluster c_j. Again, the dimensions of the vectors x, s_j and n_j are defined by the dimension of the *input-output* space of the hybrid fuzzy model $\mathcal{D}_{IO}$.

The vector x, which is defined in the *input-output* space of the hybrid fuzzy model $\mathcal{D}_{IO}$, can be divided into the components x_I, which are defined in the *input* space of the hybrid fuzzy model $\mathcal{D}_I$, and the component x_O, which is defined in the the *output* space of the hybrid fuzzy model $\mathcal{D}_O$, as shown in eq. (18).

$$x = \begin{bmatrix} x_I \\ x_O \end{bmatrix} \tag{18}$$

When using the hybrid fuzzy model for predicting the system behavior we only deal with x_I, and not with x_O. However, below we show that this is a manageable problem.

If we assume that x_O is a parameter, then the parameterized equations (16) and (17) define a contour in the *input-output* space of the hybrid fuzzy model $\mathcal{D}_{IO}$. The contour represents the intersection of a hypersphere (16) and a hyperplane (17). When treating the existence of the contour, we have to deal with three alternatives.

- If $0 < r_j < r_{j,min}$, then the contour does not exist.
- If $r_j = r_{j,min}$, then the contour degenerates into a point.
- If $r_{j,min} < r_j < \infty$, then the contour is a hypercircle.

In the aforementioned conditions, $r_{j,min}$ denotes the minimal value of the parameter r_j, which assures the existence of the intersecting hypercircle.

Obviously, we deal primarily with the third alternative, i.e., the case that the contour is a hypercircle. A hypercircle defined in such a manner represents the points in the *input-output* space of the hybrid fuzzy model $\mathcal{D}_{IO}$, for which the the normalized difference between vectors equals the radius of the hypersphere (16) $\|x - c_j\| = r_j$, and at the same time lie on the hyperplane (17). Hence, the eq. (19) holds.

$$0 \leq r_{j,min} \leq r_j < \infty \tag{19}$$

In case that the center of the hypersphere (16) c_j lies on the hyperplane (17), then $r_{j,min} = 0$.

The hypercircle defined in such a manner can be projected into the *input* space of the hybrid fuzzy model $\mathcal{D}_I$. Therefore, we can obtain a hyperellipse, which defines the points in the *input* space of the hybrid fuzzy model $\mathcal{D}_I$, which can be assigned the value r_j.

The mathematical definition of a hyperellipse in the *input* space of the hybrid fuzzy model $\mathcal{D}_I$ is given in eq. (20).

$$[x_I - s_{j,I}]^T \; A_{r_j} \; [x_I - s_{j,I}] = r_j^2,$$
$$\text{where } j \in \{1, 2, \ldots, C\} \text{ and } 0 \leq r_{j,min} \leq r_j < \infty. \tag{20}$$

In eq. (20), vector x_I denotes a point in the *input* space of the hybrid fuzzy model $\mathcal{D}_I$; $s_{j,I}$ stands for the center of the hyperellipse in the *input* space of the hybrid fuzzy model $\mathcal{D}_I$, which is obtained by projecting the center of the hypersphere s_j from the *input-output* space of the hybrid fuzzy model $\mathcal{D}_{IO}$ into the *input* space of the hybrid fuzzy model $\mathcal{D}_I$; the square matrix A_{r_j} geometrically defines the orientation and lengths of the semiaxes of the hyperellipse.

The value of the parameter r_j defines different hyperellipses in the *input* space of the hybrid fuzzy model $\mathcal{D}_I$. It goes without saying that eq. (19) must hold.

In the described manner, we can thus derive C functions that are defined in the *input* space of the hybrid fuzzy model $\mathcal{D}_I$. The functions assign a value $r_j(x_I)$ to every point x_I (for every $j \in \{1, 2, \ldots, C\}$) as shown in eq. (21).

$$\begin{aligned} &r_j : \mathcal{D}_I \rightarrow [r_{j,min}, \infty), \\ &r_j : x_I \mapsto r_j(x_I), \\ &\text{where } j \in \{1, 2, \ldots, C\}. \end{aligned} \tag{21}$$

If we know the functions $r_j(x_I)$ for every $j \in \{1, 2, \ldots, C\}$, we can derive the membership functions that correspond to the results of the clustering algorithm and that are defined in the *input* space of the hybrid fuzzy model $\mathcal{D}_I$. The membership functions assign a value $\mu_j(x_I)$ to every point x_I (for every $j \in \{1, 2, \ldots, C\}$) as shown in eq. (22).

$$\begin{aligned} &\mu_j = \frac{1}{\sum_{k=1}^{C} \left(\frac{r_j}{r_k}\right)^{\frac{2}{m-1}}}, \\ &\text{where } j \in \{1, 2, \ldots, C\}. \end{aligned} \tag{22}$$

The membership functions defined in eq. (22) can be directly implemented in the hybrid fuzzy model for predicting the system behavior – see eq. (6) – i.e., for determining the membership values corresponding to every particular cluster $\beta_j(k)$, where $j \in \{1, 2, \ldots, C\}$ – see eq. (8).

3.3 Global linear model

The hybrid fuzzy system with a common consequence structure (described in subsection 2.2) can be expressed as a global linear model. The input-dependent parameters given in eq. (23) can be derived from eq. (10).

$$\tilde{\mathbf{\Theta}}(k) = \mathbf{\Theta}(k) \; \boldsymbol{\beta}(k)^T \tag{23}$$

In this case the hybrid fuzzy model output (6) can be described as in the following equation.

$$\hat{y}_p(k+1) = \tilde{\mathbf{\Theta}}(k)^T \; \boldsymbol{\psi}(k) \tag{24}$$

3.4 Preparation of the data for estimation of the parameters of the hybrid fuzzy model

The hybrid fuzzy model parameters $a_{1jd}, \; \ldots \; , a_{njd}$, $b_{1jd}, \; \ldots \; , b_{mjd}$ and r_{jd} have to be estimated for each rule $\mathbf{R}^{jd}$; $j = 1, ..., K$ and $d = 1, ..., s$. To put it another way, all the matrices $\mathbf{\Theta}_d$ have to be established (see eq. (10)).

The regression matrix $\mathbf{\Psi}_{jd}$ for the rule $\mathbf{R}^{jd}$ in eq. (25) 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. (26) and (27) are actually used for constructing the regression matrix $\mathbf{\Psi}_{jd}$. Here, δ denotes a small positive number. Since the model parameters are obtained by matrix inversion (described later in this section), compliance with eq. (27) is essential for obtaining suitably conditioned matrices.

$$\mathbf{\Psi}_{jd} = \begin{bmatrix} \beta_j(k_1) \; \psi^T(k_1) \\ \vdots \\ \beta_j(k_{Pjd}) \; \psi^T(k_{Pjd}) \end{bmatrix} \tag{25}$$

$$q(k) = d \tag{26}$$

$$\beta_j(k) \geq \delta \tag{27}$$

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. (28). Again, only data from time-steps $(k+1)$ that comply with the conditions in eqs. (26) and (27) are actually used for constructing the output data vector $\boldsymbol{Y}_{jd}$.

$$\boldsymbol{Y}_{jd} = \begin{bmatrix} \beta_j(k_1) \; y(k_1+1) \\ \vdots \\ \beta_j(k_1) \; y(k_{Pjd}+1) \end{bmatrix} \tag{28}$$

The output contribution $\hat{y}_p^{jd}(k+1)$ corresponding to the rule $\mathbf{R}^{jd}$ is written in eq. (29).

$$\beta^j(k_1) \; \hat{y}_p^{jd}(k+1) = \mathbf{\Theta}_{jd}^T \; (\beta^j(k_1) \; \psi(k)) \tag{29}$$

Here, vector $\mathbf{\Theta}_{jd}$ represents a column in the matrix $\mathbf{\Theta}_d$, which contains the parameters of the hybrid fuzzy model corresponding to the rule $\mathbf{R}^{jd}$ as denoted in eq. (30).

$$\mathbf{\Theta}_{jd}^T = [a_{1jd} \; ... \; a_{njd} \;\; b_{1jd} \; ... \; b_{mjd} \;\; r_{jd}] \tag{30}$$

3.5 Estimation of the parameters of the hybrid fuzzy model by means of a modified least-squares method

According to eqs. (25), (28) and (29), the hybrid fuzzy model parameters for the rule $\mathbf{R}^{jd}$ can be obtained using the least-squares identification method as written in eq. (31).

$$\mathbf{\Theta}_{jd} = (\mathbf{\Psi}_{jd}^T \mathbf{\Psi}_{jd})^{-1} \mathbf{\Psi}_{jd}^T \mathbf{Y}_{jd} \tag{31}$$

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

The parameters of the hybrid fuzzy model 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 $\mathbf{\Psi}$ 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 [18, 19, 17, 20].

The described instantaneous linearization generates the parameters of the global linear model (see eq. (23)), which depends on the antecedents of the hybrid fuzzy 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.

4 Batch reactor

The presented identification method for systems that can be formulated as hybrid fuzzy models has been tested on a simulation example of a real batch reactor [13] 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.

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 hybrid fuzzy model of the batch reactor using the proposed identification approach.

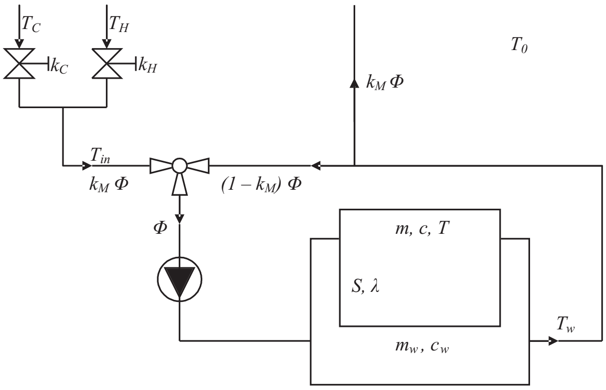

Fig. 1 Scheme of the batch reactor

4.1 Modelling and identification

In order to identify the hybrid fuzzy 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 pseudo-random generator, whereas the output signals are represented by the recorded responses of the system (for detailed information see [21]).

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 3.

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. (32). 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.

$$\hat{T}(k+1) = f(T_w(k), T(k)) \tag{32}$$

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. (33).

$$\hat{T}(k+1) = \boldsymbol{\theta}^T \left[T_w(k) \;\; T(k)\right]^T \tag{33}$$

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

$$\boldsymbol{\theta} = \left[0.0033 \;\; 0.9967\right]^T \tag{34}$$

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 betwen 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. (35). 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.

$$\hat{T}_w(k+1) = F(T_w(k), T(k), k_M(k), k_C(k), k_H(k)) \tag{35}$$

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 hybrid fuzzy model.

A general modelling and identification procedure is introduced in sections 2 in 3. By following the aforementioned procedures we obtain the following hybrid fuzzy model of the batch reactor. We define $C = 5$ clusters for each operating mode.

The output of the model in a compact form is given in eq. (36).

$$\hat{T}_w(k+1) = \boldsymbol{\beta}(k,q)\boldsymbol{\Theta}_w^T(q) \left[T_w(k) \; T(k) \; k_M(k) \; 1\right]^T \tag{36}$$

The discrete dynamics (operating mode) of the model are defined in eq. (37).

$$q(k) = q(k_H(k), k_C(k)) = \\ = \left\{ \begin{array}{lll} 1 & ; & k_C(k) = 0 \bigwedge k_H(k) = 1 \\ 2 & ; & k_C(k) = 1 \bigwedge k_H(k) = 0 \end{array} \right. \tag{37}$$

The parameters of the model are given in eqs. (38) and (39).

$$\boldsymbol{\Theta}_{w1} = \begin{bmatrix} 0.6312 & 0.9257 & 0.9361 & 0.9404 & 0.9277 \\ 0.0400 & 0.0513 & 0.0384 & 0.0490 & 0.0515 \\ 1.7115 & 12.5240 & 6.9758 & 19.3915 & 16.8357 \\ 22.4874 & 0.7210 & 1.0316 & 0.1955 & 0.5233 \end{bmatrix} \tag{38}$$

$$\boldsymbol{\Theta}_{w2} = \\ = \begin{bmatrix} 0.9059 & 0.9337 & 0.9462 & 0.6279 & 0.6312 \\ 0.0630 & 0.0468 & 0.0359 & 0.0404 & 0.0400 \\ -10.8609 & -4.5854 & -1.5090 & -15.3616 & -14.9129 \\ 0.8869 & 0.4228 & 0.3163 & 19.3905 & 18.9084 \end{bmatrix} \tag{39}$$

4.2 Validation

We have validated the obtained hybrid fuzzy model by comparing its responses to the responses of the original batch reactor model. The input signals have been generated using a pseudorandom generator (for detailed information see [21]). 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 hybrid fuzzy model of the batch reactor. The dotted line represents the original response of the batch reactor to the input signals.

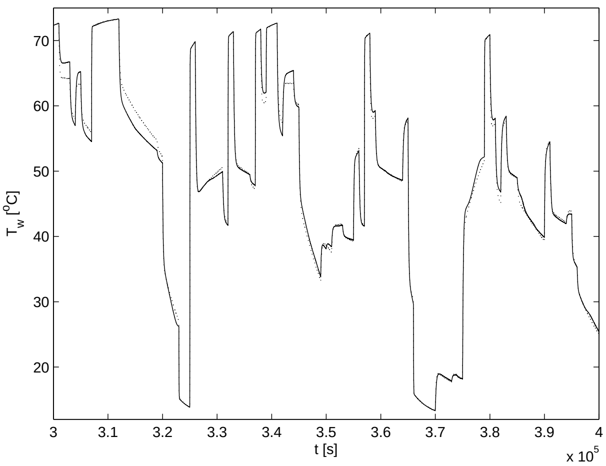

Fig. 2 The response of the hybrid fuzzy model to the validation input signals (solid line) and the original response of the batch reactor (dotted line).

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.

$$\bar{J}_{T_w} = 1.0474 \tag{40}$$

- The average squared discrepancy of the temperature in the core of the batch reactor T.

$$\bar{J}_T = 0.5574 \tag{41}$$

5 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 hybrid fuzzy model represents a convenient 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 hybrid fuzzy 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 hybrid fuzzy model $\mathcal{D}_{IO}$ into the *input* space of the hybrid fuzzy model $\mathcal{D}_I$. In this manner, we can obtain a hybrid fuzzy 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 hybrid fuzzy model, which can eventually be used in a model predictive control algorithm.

6 References

[1] A. Van der Schaft and H. Schumacher. An introduction to hybrid dynamical systems. *Lecture Notes in Control and Information Sciences*, 251:v–vii, 1999.

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

[3] E. Sontag. Nonlinear regulation: The piecewise linear approach. *IEEE Transactions on Automatic control*, 26(2):346–358, 1981.

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

[5] 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.

[6] 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.

[7] 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.

[8] I. Škrjanc and D. Matko. Fuzzy predictive functional control in the state space domain. *Journal of Intelligent and Robotic Systems*, 31:283–297, 2001.

[9] H. S. Witsenhausen. A class of hybrid-state continuous time dynamic systems. *IEEE Trans. on Automatic Control*, 11(2):161–167, 1966.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

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

[15] J. Castro. Fuzzy logic controllers are universal approximators. *IEEE Trans. System Man Cybernet*, 25:629–635, 1995.

[16] F. Girosi and T. Poggio. Networks and the best approximation property. *Biological Cybernetics*, 63:169–176, 1990.

[17] 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.

[18] R. Babuška. *Fuzzy Modelling for Control*. KAP, 1998.

[19] R. Babuška and H. B. Verbruggen. An overview of fuzzy modelling for control. *Control Engineering Practice*, 4(11):1593–1606, 1996.

[20] 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.

[21] G. Karer. *Prediktivno vodenje hibridnih sistemov*. PhD thesis, Fakulteta za elektrotehniko, Univerza v Ljubljani, 2009.