# FEEDFORWARD CONTROL OF A CLASS OF HYBRID SYSTEMS USING AN INVERSE MODEL

G. Karer[1], G. Mušič[1], I. Škrjanc[1], B. Zupančič[1]
[1]University of Ljubljana, Faculty of Electrical Engineering

Corresponding author: G. Karer, University of Ljubljana, Faculty of Electrical Engineering
Tržaška 25, 1000 Ljubljana, Slovenia, gorazd.karer@fe.uni-lj.si

**Abstract.**

In this paper we describe the design of a control algorithm for MISO systems, which can be modelled as *hybrid fuzzy systems*. The control scheme we are discussing splits the control algorithm in two parts: the feedforward part and the feedback part. In the paper, we deal with the feedforward part of the control algorithm, which is based on an inverse of a hybrid fuzzy model.

We discuss the formulation of a hybrid fuzzy model. This is followed by a derivation of the inverse model and its implementation in the control algorithm. Next, a batch-reactor process is introduced. The modelling of the batch reactor is tackled and the results of the simulation experiments using the proposed control algorithm are presented. The experiments involved controlling the temperature of a batch reactor by using two on/off input valves and a continuous mixing valve.

The main advantage of the proposed approach is that the feedforward part of the control algorithm can bring the system close to the desired adjusted feasible trajectory, which avoids the need for a very complex feedback part of the algorithm. Therefore, the control algorithm presents a low computational burden, particularly comparing to the standard model predictive control algorithms. These usually require a considerable computational effort, which often thwarts their implementation on real industrial systems.

## 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 [17]. In recent years, however, hybrid systems have received a great deal of attention from the computer science and control community.

The principle of *model predictive control* (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) [3]. A lot of interest has also been devoted to *piecewise affine* (PWA) formulation [15], which has been proven to be equivalent to many classes of hybrid systems [7]. 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 [12].

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 [18]. 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 [19] is introduced in [11]. Furthermore, two structure-selecting methods for nonlinear models with mixed discrete and continuous inputs are presented in [5]. In [13] 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 [8]. 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 the feedforward part of a control algorithm suitable for controlling MISO systems, which can be modelled by a hybrid fuzzy model.

The outline of the paper is as follows. Section 2 introduces the basic control scheme and the idea of the feedforward part of the control algorithm. We also discuss the inclusion of a feedback part in the control algorithm, which is still research in progress. In section 3 the structure modelling of a hybrid fuzzy model is discussed. This is followed by section 4, which deals with derivation of the inverse model and its implementation in the control algorithm. In the following section, a batch-reactor process is introduced. The modelling of the batch reactor is tackled and the results of the simulation experiments using the proposed control algorithm are presented. Finally, we give some concluding remarks.

## 2 The control scheme

The basic idea of the control scheme discussed in the paper is to split the control algorithm in two parts: the feedforward part and the feedback part (see fig. 1).

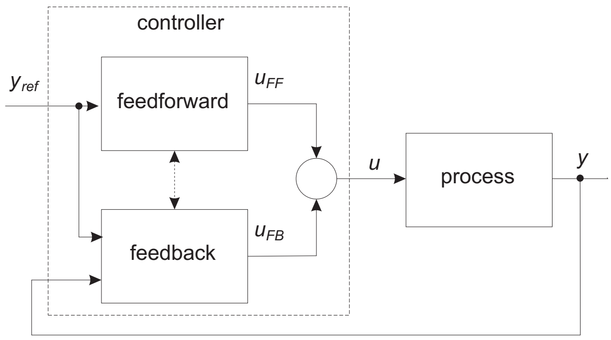


**Figure 1:** The control scheme.

In the paper, we focus on the feedforward part of the control algorithm. The feedforward part of the algorithm uses a hybrid fuzzy model of the MISO system we are to control in order to calculate the appropriate inputs. By feeding the reference signal into the inverse hybrid fuzzy model the algorithm obtains the appropriate input signal.

The second part of the control algorithm is a feedback controller. However, we do not deal with the feedback part of the algorithm in this paper.

The main advantage of the proposed approach is that the feedforward part of the control algorithm can bring the system close[1] to the desired adjusted feasible trajectory. Therefore, in order to obtain a suitable control performance, a simple design of the feedback part of the control algorithm should be sufficient. For instance, one could use a model predictive controller employing a model, which is linearized at the operating point of the system. Since the feedback part of the algorithm takes into account the output signals of the system, the combined control algorithm could easily compensate for the inaccurate modelling, noise and eventual disturbances on the real system.

## 3 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.

$$\hat{y}_p(k+1) = F(y(k), y(k-1), \ldots, y(k-n+1), u(k), u(k-1), \ldots, u(k-m+1)) \tag{1}$$

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

[1] Depending on the accuracy of the hybrid fuzzy model.

In this paper, a special class of systems is addressed, i.e., nonlinear hybrid systems with discrete inputs. Therefore, in eq. (1) $u$ stands for the discrete input.

### 3.1 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).

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

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

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 [19] 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 3.2.

### 3.2 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 [4, 6].

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).

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

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))[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).

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

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.

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

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 $\boldsymbol{\beta}(k) = [\beta_1(k)\ \beta_2(k)\ ...\ \beta_K(k)]$. 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 [16]. In our case it is a simple algebraic product, given in eq. (8).

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

Here, $\mu_{A_1^j}(y(k))\ ...\ \mu_{A_n^j}(y(k-n+1))$ denote the membership values [1, 2, 16].

In eq. (6), $\mathbf{\Theta}(k)$ 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), $\mathbf{\Theta}(k)$ is actually a function of the discrete state of the hybrid fuzzy system in the current time-step $q(k)$.

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

The matrices $\mathbf{\Theta}_d$ 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 $\mathbf{\Theta}_d$ to be time-invariant.

Each matrix $\mathbf{\Theta}_d$ 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 $j = 1,...,K$. $\mathbf{\Theta}_d$ is constructed as shown in eq. (10).

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

[2] 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)).

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).

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

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 $y(k-1)$, ..., $y(k-n+1)$ 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 $y(k-1)$, ..., $y(k-n+1)$ 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 $n$th-order model for MPC purposes. If such first-order models are not feasible, it is still suitable to employ several lower-than-$n$th-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.

### 3.3 Identification of the hybrid fuzzy model

The parameters of a hybrid fuzzy model can be identified using a method described in [8].

## 4 Feedforward control using an inverse model

The basic idea of the control approach is to derive a hybrid fuzzy model of the MISO system we are to control and use the inverse of the model as a sort of feedforward controller. That means that the input of the controller is fed with the reference signal, whereas the output returns the calculated inputs of the system in the relevant time-step. Of course, it is very important to take into account the eventual constraints of the system.

### 4.1 Inverse model

We assume that the hybrid fuzzy model of the system is known in advance. The model of the system is formulated in a compact form in eq. (6). Furthermore, we assume the system has a single output and the model should be fuzzyfied with regard to the output ($\boldsymbol{\beta}(k) = \boldsymbol{\beta}(y(k))$). In addition, we assume the system has a single continuous input. The operating mode is defined by the discrete inputs[3]. Starting with eq. (6), the hybrid fuzzy model of the system can be rewritten in the following equation (12).

$$\hat{y}_p(k+1) = \boldsymbol{\beta}(y(k))\, \boldsymbol{\Theta}^T(q(k))\, \boldsymbol{\psi}(k) \tag{12}$$

With some mathematic operations using eqs. (12), (9), (10), and (11) it is possible to derive an inverse model by expressing the input in the following eqs. (13), (14), (15), (16) and (17).

$$u(k) = \frac{\hat{y}_p(k+1) - S_{A,inv} - S_{B,inv} - S_{R,inv}}{S_{U_inv}} \tag{13}$$

$$S_{A,inv} = \sum_{i=1}^{n} \sum_{j=1}^{K} \beta_j(y(k))\, a_{ijd}(q(k))\, y(k-i+1) \tag{14}$$

$$S_{B,inv} = \sum_{i=2}^{m} \sum_{j=1}^{K} \beta_j(y(k))\, b_{ijd}(q(k))\, u(k-i+1) \tag{15}$$

$$S_{R,inv} = \sum_{j=1}^{K} \beta_j(y(k))\, r_{jd}(q(k)) \tag{16}$$

[3] In case the operating mode should depend on the output of the model instead of the input, it is possible to take this into account by employing crisp sets in addition to fuzzy sets $A_i^j$ (see eq. (4)).

$$S_{U,inv} = \sum_{j=1}^{K} \beta_j(y(k))\, b_{1jd}(q(k)) \tag{17}$$

### 4.2 The feedforward part of the control algorithm

**The basic control law.** In order to obtain the desired feedforward controller, we derive the control law from eq. (13). The predicted output of the model in the next time-step is exchanged with the desired output, i.e. the reference signal: $\hat{y}_p(k+1) \rightarrow y_{ref}(k)$. Accordingly, the feedforward design is implemented by substituting the model outputs in the previous time-steps with the reference signal: $y(k-i+1) \rightarrow y_{ref}(k-i)$. The normalized degrees of fulfillment are also calculated from the reference signal instead of the actual model output: $\beta_j(y(k)) \rightarrow \beta_j(y_{ref}(k-1))$.

With the aforementioned substitution it is now possible to derive the control law as expressed in the following eqs. (18), (19), (20) and (21).

$$u_{FF}(k) = \frac{y_{ref}(k) - S_A - S_B - S_R}{S_U} \tag{18}$$

$$S_A = \sum_{i=1}^{n} \sum_{j=1}^{K} \beta_j(y_{ref}(k-1))\, a_{ijd}(q(k))\, y_{ref}(k-i) \tag{19}$$

$$S_B = \sum_{i=2}^{m} \sum_{j=1}^{K} \beta_j(y_{ref}(k-1))\, b_{ijd}(q(k))\, u_{FF}(k-i+1) \tag{20}$$

$$S_R = \sum_{j=1}^{K} \beta_j(y_{ref}(k-1))\, r_{jd}(q(k)) \tag{21}$$

$$S_U = \sum_{j=1}^{K} \beta_j(y_{ref}(k-1))\, b_{1jd}(q(k)) \tag{22}$$

Eq. (18) presents the core of the control algorithm where the appropriate input signal $u_{FF}(k)$ is calculated from the inverse model so that the desired output signal $y_{ref}$ is obtained. Note that due to causality reasons, the actual output will be delayed by one time-step with regard to the reference signal $y_{ref}$.

**Handling the input constraints and the operating-mode selection.** The basic control law as described in eq. (18) returns the calculated feedforward input $u_{FF}(k)$ without considering its constraints. However, in most industrial applications the inputs are inherently constrained. Therefore, instead of just using the inverse model to calculate the often unreasonable input value $u_{FF}(k)$, which should be fed into the system so as to cause the output to track the desired reference trajectory $y_{ref}$, the control algorithm has to take these constraints into account. In other words, in case the desired reference trajectory $y_{ref}$ is impossible to track, the algorithm has to adjust it in order to make it feasible. The adjustment is done in the following manner.

The algorithm first uses the hybrid fuzzy model in eq. (12) to calculate the range of the outputs $\hat{y}_p(k+1)$ of the hybrid fuzzy system considering the input constraints for each operating mode individually. Note that since the hybrid fuzzy system is monotonous in a sense that an extreme[4] input signal $u(k)$ results in an extreme predicted output $\hat{y}_p(k+1)$ in the domain of each individual operating mode, we can establish a reachability matrix $\hat{R}_p(k+1)$ as given in eq. (23) with only two calculations of the predicted output per operating mode.

$$\hat{R}_p(k+1) = \begin{bmatrix} \hat{y}_{p,min}(k+1)|_{q(k)=1} & \hat{y}_{p,max}(k+1)|_{q(k)=1} \\ \vdots & \vdots \\ \hat{y}_{p,min}(k+1)|_{q(k)=s} & \hat{y}_{p,max}(k+1)|_{q(k)=s} \end{bmatrix} \tag{23}$$

The reachability matrix $\hat{R}_p(k+1)$ is made up of the range intervals of the predicted output in the next time-step $\hat{y}_p(k+1)$ of the hybrid fuzzy model. Each row represents an individual operating mode $q(k)$. There are two possible scenarios.

[4] Extreme in a sense that $u(k)$ is at the endpoint of the constrained interval.

1. In case the desired reference is in an interval $\hat{y}_p(k+1) \in [\hat{y}_{p,min}(k+1)|_{q(k)=d}, \hat{y}_{p,max}(k+1)|_{q(k)=d}]$ represented by an individual row of the reachability matrix $\hat{R}_p(k+1)$ denoting the range of the hybrid fuzzy system in the domain of the corresponding operating mode $q(k) = d$, we can conclude that the desired reference $\hat{y}_p(k+1)$ is feasible.
   The corresponding operating mode $q(k) = d$ (and hence the discrete input signal) is selected. The continuous input $u_{FF}(k)$ is calculated using the basic control law in eq. (18) as described above.
   If the desired reference is in more than one interval represented by individual rows of the reachability matrix $\hat{R}_p(k+1)$, the operating mode $q(k)$ is selected on a higher level. The continuous input $u_{FF}(k)$ is calculated using the basic control law in eq. (18) for each operating mode, which returns a feasible solution, individually. Next, the algorithm selects the most suitable continuous input $u_{FF}(k)$ and its corresponding operating mode $q(k)$ from the previously calculated set of solutions according to some pre-specified high-level rules or cost functions.
2. In case the desired reference is not in any of the intervals $\hat{y}_p(k+1) \notin [\hat{y}_{p,min}(k+1)|_{q(k)=d}, \hat{y}_{p,max}(k+1)|_{q(k)=d}]$ represented by individual rows of the reachability matrix $\hat{R}_p(k+1)$ denoting the range of the hybrid fuzzy system in the domain of the corresponding operating mode $q(k) = d = 1, \cdots, s$, we can conclude that the desired reference $\hat{y}_p(k+1)$ is infeasible.
   Therefore, the desired reference signal has to be adjusted by moving it into the range of the hybrid fuzzy model. The algorithm selects the closest feasible solution in the reachability matrix $\hat{R}_p(k+1)$ to the desired reference and treats the new value as the adjusted feasible reference signal. The corresponding operating mode $q(k) = d$ (and hence the discrete input signal) is selected. The continuous input $u_{FF}(k)$ is calculated using the basic control law in eq. (18) as described above, the only diference being that the adjusted feasible reference signal is used in eqs. (18)–(22) from this time-step on.

# 5 Control of a batch reactor

The control approach was tested on a simulation example of a real batch reactor 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.

### 5.1 The batch reactor

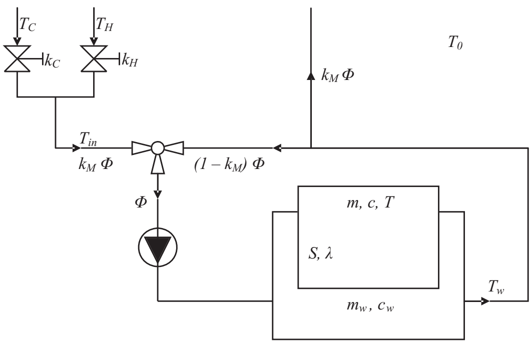


**Figure 2:** Scheme of the batch reactor

A scheme of the batch reactor is shown in Fig. 2. The reactor's core (temperature $T$) is heated or cooled through the reactor's water jacket (temperature $T_w$). The heating medium in the water jacket is a mixture of fresh input water, which enters the reactor through on/off valves, and reflux water. The water is pumped into the water jacket with a constant flow $\phi$. The dynamics of the system depend on the physical properties of the batch reactor, i.e., the mass $m$ and the specific heat capacity $c$ of the ingredients in the reactor's core and in the reactor's water jacket (here, the index $w$ denotes the water jacket). $\lambda$ is the thermal conductivity, $S$ is the contact area and $T_0$ is the temperature of the surroundings.

The temperature of the fresh input water $T_{in}$ depends on two inputs: the position of the on/off valves $k_H$ and $k_C$. However, there are two possible operating modes of the on/off valves. In case $k_C = 1$ and $k_H = 0$, the input water is cool ($T_{in} = T_C = 12\,^0C$), whereas if $k_C = 0$ and $k_H = 1$, the input water is hot ($T_{in} = T_H = 75\,^0C$).

The ratio of fresh input water to reflux water is controlled by the third input, i.e., by the position of the mixing valve $k_M$.

We are therefore dealing with a multivariable system with two discrete inputs ($k_C$, $k_H$), a continuous input ($k_M$) and two measurable outputs ($T$ and $T_w$). Due to the nature of the system, the time constant of the temperature in the water jacket is obviously much shorter than the time constant of the temperature in the reactor's core. Therefore, the batch reactor is considered as a stiff system.

### 5.2 Hybrid fuzzy model of the batch reactor

The modelling procedure is explained in detail in [8].

The temperature in the reactor's core $T$ is influenced only by the heat conduction between the reactor's core and the reactor's water jacket. Furthermore, we have surmised that the heat conduction is proportional to the temperature difference between the reactor's core $T$ and the reactor's water jacket $T_\mathrm{w}$. Therefore, a first-order linear MISO submodel can be presumed, as shown in (24). The system parameters are given below.

$$\hat{T}(k+1) = \mathbf{\Theta}_c^T \left[T_\mathrm{w}(k) \;\; T(k)\right]^T \tag{24}$$

$$\mathbf{\Theta}_c^T = [0.0033 \;\; 0.9967] \tag{25}$$

The temperature in the reactor water jacket $T_\mathrm{w}$ is influenced by the temperature in the core $T$, the fresh input water inflow at the mixing valve $k_M$, and the position of the cold-water and hot-water on/off valves $k_C$ and $k_H$.

Let us assume two operating modes of the subsystem ($s = 2$).

- The first operating mode ($q = 1$) is the case when the fresh input water is hot, i.e., $k_C(k) = 0$ and $k_H(k) = 1$.
- The second operating mode ($q = 2$) is the case when the fresh input water is cool, i.e., $k_C(k) = 1$ and $k_H(k) = 0$.

$$q(k) = q(k_C(k), k_H(k)) = \begin{cases} 1 & \text{if} \quad k_C(k) = 0 \wedge k_H(k) = 1 \\ 2 & \text{if} \quad k_C(k) = 1 \wedge k_H(k) = 0 \end{cases} \tag{26}$$

Next, the membership functions have to be defined. The system is fuzzyfied with regard to the temperature in the reactor's water jacket $T_\mathrm{w}(k)$. Simple triangular functions are used, as shown in Figure 3.

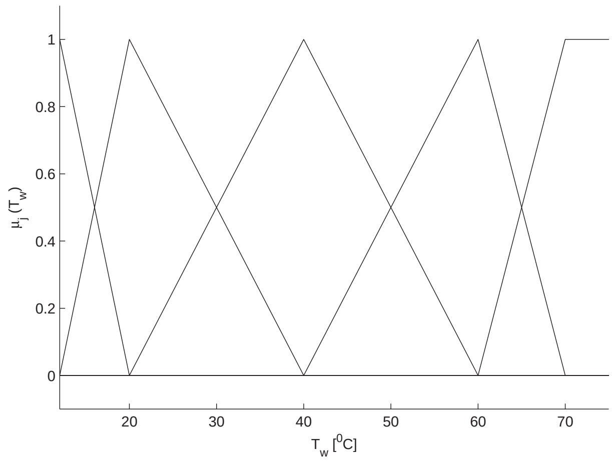


**Figure 3:** Membership functions.

Such a form of the membership functions ensures that the normalized degrees of fulfillment $\beta_j(T_\mathrm{w})$ are equal to the membership values $\mu_j(T_\mathrm{w})$ across the whole operating range for each rule $\mathbf{R}^{jd}$, respectively. The normalized degrees of fulfillment $\beta_j(T_\mathrm{w})$ make up a normalized vector of fulfillment $\boldsymbol{\beta}(T_\mathrm{w}(k)) = \boldsymbol{\beta}(k)$. In this case there are five membership functions ($K = 5$), with maximums at 12, 20, 40, 60 and $70^0$, so that the whole operating range is covered.

The rule base of the hybrid fuzzy model is given in (27). We presume that a local system corresponding to an individual rule $\mathbf{R}^{jd}$ is affine.

$$\begin{aligned} &\mathbf{R}^{jd}: \\ &\quad \textbf{if } q(k) \text{ is } Q_d \text{ and } T_\mathrm{w}(k) \text{ is } A_1^j \\ &\quad \textbf{then } T_\mathrm{w}(k+1) = a_{1jd}T_\mathrm{w}(k) + a_{2jd}T(k) + b_{1jd}k_M(k) + r_{jd} \\ &\text{for } j = 1, \ldots, 5 \text{ and } d = 1, 2 \end{aligned} \tag{27}$$

The output of the model of the temperature in the reactor's water jacket is written in compact form in (28), (29) and (30). For a detailed description of the identification procedure see [8].

$$\hat{T}_{\mathrm{w}}(k+1) = \boldsymbol{\beta}(k)\, \boldsymbol{\Theta}_{\mathrm{w}}^{T}(k)\; [T_{\mathrm{w}}(k)\;\; T(k)\;\; k_M(k)\;\; 1]^T \tag{28}$$

$$\boldsymbol{\Theta}_{\mathrm{w}}(k) = \begin{cases} \boldsymbol{\Theta}_{\mathrm{w1}} & \text{if} \quad q(k) = 1 \\ \boldsymbol{\Theta}_{\mathrm{w2}} & \text{if} \quad q(k) = 2 \end{cases} \tag{29}$$

$$\boldsymbol{\Theta}_{\mathrm{w1}} = \begin{bmatrix} 0.9453 & 0.9431 & 0.9429 & 0.9396 & 0.7910 \\ 0.0376 & 0.0458 & 0.0395 & 0.0339 & 0.0225 \\ 19.6748 & 16.7605 & 10.5969 & 3.9536 & 1.6856 \\ 0.3021 & 0.2160 & 0.5273 & 1.2701 & 12.0404 \end{bmatrix} \tag{30}$$

$$\boldsymbol{\Theta}_{\mathrm{w2}} = \begin{bmatrix} 0.9803 & 0.9740 & 0.9322 & 0.9076 & 0.8945 \\ 0.0025 & 0.0153 & 0.0466 & 0.0466 & 0.0111 \\ -0.0704 & -0.6956 & -7.8013 & -12.2555 & -18.7457 \\ 0.2707 & 0.2033 & 0.5650 & 1.9179 & 5.6129 \end{bmatrix} \tag{31}$$

### 5.3 Results

We tested the feedforward control algorithm first on the derived hybrid fuzzy model of the batch reactor. The reference trajectory is shown in the following figures: the temperature in the core of the reactor should first rise to $62\,^0C$, then fall to $26\,^0C$, and finally settle on $35\,^0C$. We also imposed a control constraint: the temperature in the reactor's water jacket should be constrained between $T_{w,min} = 20\,^0C$ and $T_{w,max} = 70\,^0C$.

The results of the experiment using the feedforward part of the control algorithm for hybrid fuzzy systems on the hybrid fuzzy model of the batch reactor are shown in the following figures. Note that the simulation model representing the process and the model used in the algorithm are exactly the same.

Fig. 4 depicts the temperature in the core of the reactor $T$ and the reference trajectory as stated above. The adjusted feasible reference trajectory calculated by the feedforward control algorithm using the hybrid fuzzy model is not shown in the figure, because it is perfectly followed by the output signal, except for the inherent delay due to causality reasons.

Fig. 5 shows the temperature in the water jacket of the reactor $T_w$ and the reference trajectory as stated above. Again, the adjusted feasible reference trajectory calculated by the feedforward control algorithm using the hybrid fuzzy model is not shown in the figure, because it is perfectly followed by the output signal, except for the inherent delay due to causality reasons.

The input signals are shown in fig. 6.

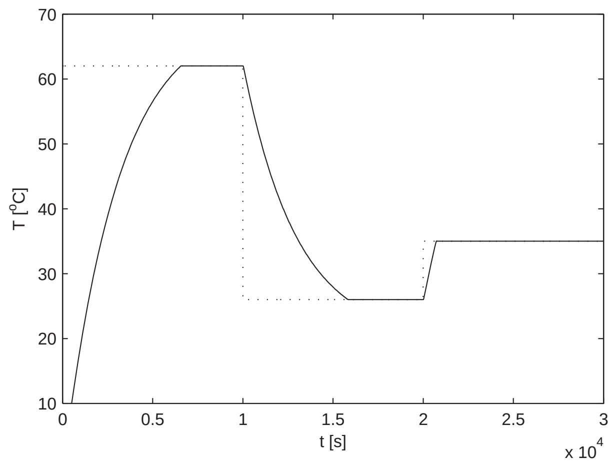


**Figure 4:** Feedforward control of the hybrid fuzzy model: Core temperature (solid line) and reference temperature (dotted line).

# 6 Conclusion and future work

In an ideal scenario, the hybrid fuzzy model used in the control algorithm would be exactly the same as the system being controlled. It is clear that in this case the control law in eq. (18) guarantees that the output follows the

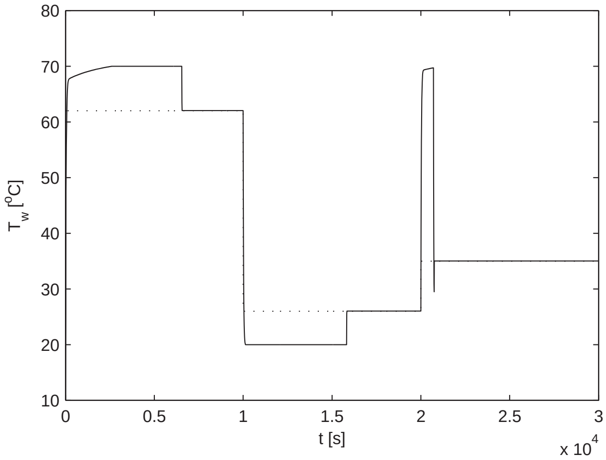


**Figure 5:** Feedforward control of the hybrid fuzzy model: Temperature in the water jacket (solid line) and reference temperature (dotted line).

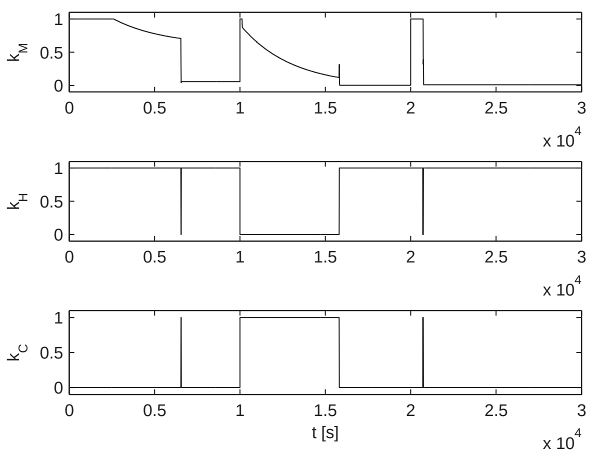


**Figure 6:** Input signals.

adjusted feasible reference trajectory ideally, which is also evident from the simulation results. The reference trajectory is adjusted according to feasible solutions, which are calculated considering the control constraints using the hybrid fuzzy model of the system. Obviously, the output is inherently delayed with regard to the adjusted feasible reference due to causality reasons (see section 4.2).

However, it goes without saying that in real-world applications the hybrid fuzzy model used in the control algorithm represents only an approximation of the system being controlled. Therefore, it is clear that the output can not exactly follow the adjusted feasible reference trajectory using only the feedforward part of the control algorithm. Hence, a feedback part should be included in the control algorithm as described in section 2.

The main advantage of the proposed approach is that such algorithm presents a low computational burden; both the feedforward and feedback part are computationally simple, particularly comparing to the standard model predictive control algorithms. These usually require a considerable computational effort, which often thwarts their implementation on real industrial systems.

The future work will therefore focus on developing and including the feedback part in the control algorithm and verifying its usefulness it on the studied batch reactor example.

# 7 References


[1] R. Babuska. *Fuzzy Modelling for Control*. KAP, 1998.

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

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

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

[5] D. Girimonte and R. Babuska. 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] F. Girosi and T. Poggio. Networks and the best approximation property. *Biological Cybernetics*, 63:169–176, 1990.

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

[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, December 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, September 1998.

[12] 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.

[13] 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.

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

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

[16] 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.

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

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

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