

Provided for non-commercial research and education use.
Not for reproduction, distribution or commercial use.



This article was published in an Elsevier journal. The attached copy is furnished to the author for non-commercial research and education use, including for instruction at the author's institution, sharing with colleagues and providing to institution administration.

Other uses, including reproduction and distribution, or selling or licensing copies, or posting to personal, institutional or third party websites are prohibited.

In most cases authors are permitted to post their version of the article (e.g. in Word or Tex form) to their personal website or institutional repository. Authors requiring further information regarding Elsevier's archiving and manuscript policies are encouraged to visit:

<http://www.elsevier.com/copyright>



Hybrid fuzzy model-based predictive control of temperature in a batch reactor

Gorazd Karer*, Gašper Mušič, Igor Škrjanc, Borut Zupančič

Faculty of Electrical Engineering, University of Ljubljana, Tržaška 25, 1000 Ljubljana, Slovenia

Received 4 August 2006; received in revised form 16 November 2006; accepted 14 January 2007

Available online 17 January 2007

Abstract

Processes in industry, such as batch reactors, often demonstrate a hybrid and non-linear nature. Model predictive control (MPC) is one of the approaches that can be successfully employed in such cases. However, due to the complexity of these processes, obtaining a suitable model is often a difficult task. In this paper a hybrid fuzzy modelling approach with a compact formulation is introduced. The hybrid system hierarchy is explained and the Takagi–Sugeno fuzzy formulation for the hybrid fuzzy modelling purposes is presented. An efficient method for identifying the hybrid fuzzy model is also proposed.

A MPC algorithm suitable for systems with discrete inputs is treated. The benefits of the MPC algorithm employing the proposed hybrid fuzzy model are verified on a batch-reactor simulation example: a comparison between MPC employing a hybrid linear model and a hybrid fuzzy model was made. We established that the latter approach clearly outperforms the approach where a linear model is used.

© 2007 Elsevier Ltd. All rights reserved.

Keywords: Process control; Model predictive control; Fuzzy systems; Hybrid 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, and so forth. 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 (van der Schaft & Schumacher, 1999). 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 non-linear 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. Classic modelling and especially iden-

tification methods that originate from linear system theory are inadequate for treating such systems. 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) (Bemporad & Morari, 1999). A lot of interest has also been devoted to piecewise affine (PWA) formulation (Sontag, 1981), which has been proven to be equivalent to many classes of hybrid systems (Heemels, Schutter, & Bemporad, 2001). 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 (Mayne & Raković, 2003). On the other hand, in Kerrigan and Mayne (2002) the authors tackle the optimal control problem for PWA systems by solving a number of multiparametric 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

* Corresponding author.

E-mail address: gorazd.karer@fe.uni-lj.si (G. Karer).

Nomenclature*Modelling and identification*

a_{1jd}, \dots, a_{njd}	consequent fuzzyfied parameters for \mathbf{R}^{jd}
b_{1jd}, \dots, b_{mjd}	consequent fuzzyfied parameters for \mathbf{R}^{jd}
$q(k)$	discrete state
r_{jd}	consequent fuzzyfied parameters for \mathbf{R}^{jd}
\mathbf{R}^{jd}	rule jd
$\mathbf{u}(k)$	input vector
$\mathbf{x}(k)$	continuous state vector
$\mathbf{y}(k)$	output vector
$\hat{y}_p(k+1)$	predicted output of the system
\mathbf{Y}_{jd}	output data vector
$\beta(k)$	normalized degrees of fulfillment
$\mu_{A_1^j}, \dots, \mu_{A_n^j}$	membership values
$\Theta(k)$	matrix of consequent fuzzyfied parameters
$\psi(k)$	regressor
Ψ_{jd}	regression matrix for \mathbf{R}^{jd}
<i>MPC algorithm</i>	
h	current level of prediction
H	maximum prediction horizon
H_Z	maximum prediction horizon in case $Z > 1$
J	cost function
M	number of possible input vector values
Q_k^{k+h}	sequence of discrete states
T_{pred}	maximum time of prediction
T_S	sampling time
$\underline{u}_1, \dots, \underline{u}_M$	input vectors used in the MPC algorithm
U_k^{k+h-1}	sequence of input vectors
X_k^{k+h}	sequence of continuous state vectors
Z	number of time-steps through which the inputs are held

Batch reactor

c	specific heat capacity of the ingredients in the reactor's core
c_w	specific heat capacity of the ingredients in the reactor's water jacket
k_C	position of the on/off valve for cool water
k_H	position of the on/off valve for hot water
k_M	position of the mixing valve
m	mass of the ingredients in the reactor's core
m_w	mass of the ingredients in the reactor's water jacket
S	contact area
T	temperature of the reactor's core
T_C	temperature of the cool water
T_H	temperature of the hot water
T_{in}	temperature of the fresh input water
T_w	temperature of the reactor's water jacket
T_0	temperature of the surroundings
ϕ	flow of the water in the reactor's water jacket
λ	thermal conductivity

Θ	vector of parameters for subsystem T
Θ_1, Θ_2	matrix of consequent fuzzyfied parameters for subsystem T_w
$\Theta_{1,\text{lin}}, \Theta_{2,\text{lin}}$	vector of parameters for subsystem T_w -hybrid linear model

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 Potočnik, Mušič, and Zupančič (2005).

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 non-linearities. Since a PWA formulation can only represent piecewise affine systems, further segmentation is required in order to suitably approximate the non-linearity. 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 non-linear modelling approach for MPC purposes is presented in Škrjanc and Matko (2001). The authors introduce an analytical predictive-control-law for fuzzy systems. The modelling and identification methodology is usable for plain non-linear systems, but not for the structurally more complex class of hybrid systems. A hierarchical identification of a fuzzy switched system (Witsenhausen, 1966) is introduced in Palm and Driankov (1998). Furthermore, two structure-selecting methods for non-linear models with mixed discrete and continuous inputs are presented in Girimonte and Babuška (2004). In Qin and Jia (2002) a fuzzy control method is implemented in the low control-level for a class of hybrid systems based on hybrid automata.

The basic idea of this paper is to present an efficient approach for obtaining a hybrid fuzzy model by means of identifying the unknown system. Some concepts from the previous work are extended to non-linear hybrid systems. In the paper a formulation for a hybrid fuzzy model that is based on a hierarchical structure and can be written in a compact form is proposed. In addition, an efficient parameter-estimation method is introduced. We suggest that such a hybrid fuzzy model is suitable for implementation in the MPC of non-linear hybrid systems with discrete inputs based on a reachability analysis. By using a more accurate model for the MPC, a better control performance can be achieved, which is a significant advantage of the presented approach.

The outline of the paper is as follows. In Section 2 the structure modelling of a hybrid fuzzy model is discussed. This is followed by Section 3, which deals with the parameter estimation of the model. In Section 4 an algorithm for the MPC of systems with discrete inputs based on a reachability analysis is treated. In the following sections, a batch-reactor process is introduced. The modelling and parameter estimation of the

process are tackled and the framework is verified by means of a simulation experiment. Finally, a comparison between MPC employing a hybrid linear model and a hybrid fuzzy model is made.

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 non-linear model structure is the Non-linear AutoRegressive with eXogenous inputs (NARX) model (Sjoberg et al., 1995), which gives the mapping between the past input–output data and the predicted output.

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

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

In this paper, a special class of systems is addressed, that is, non-linear hybrid systems with discrete inputs. Therefore, in Eq. (1) u stands for the discrete input.

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, that is, 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 the following equations:

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

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

Here, $\mathbf{x} \in \mathbb{R}^n$ is the continuous state vector, which includes all relevant system outputs y (see Eq. (1)), that is, 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 $\mathbb{Q} = \{1, \dots, s\}$) 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 (Witsenhausen, 1966) 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 non-linear, therefore it can be modelled as a Takagi–Sugeno fuzzy model, as shown in Section 2.2.

2.2. Generalization of the Takagi–Sugeno formulation for a non-linear hybrid system

In order to approximate a non-linear system, a fuzzy formulation can be employed. Fuzzy models can be regarded as universal approximators, which can approximate continuous functions to an arbitrary precision (Castro, 1995; Girosi & Poggio, 1990).

The system dynamics can be formulated as a Takagi–Sugeno fuzzy model. In order to address non-linear 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 the following equation:

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

The **if**-parts (antecedents) of the rules describe hybrid fuzzy regions in the space of the input variables of the hybrid fuzzy model. Here, $q(k) \in \{1, \dots, s\}$ stands for the discrete state of the non-linear hybrid system, that is, 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 Ks . Generally speaking, K depends on the number of fuzzy membership functions for each antecedent variable $y(k), \dots, y(k-n+1), u(k), \dots, u(k-m+1)$. 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 $y(k), y(k-1), \dots, y(k-n+1), u(k), \dots, u(k-m+1)$. 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 non-linear 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 two discrete input variables and each variable can have four discrete values, the number of operating modes (due to discrete inputs) is eight. 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)).¹ There is one function of inputs f_{jd} defined for each rule \mathbf{R}^{jd} ; $j=1, \dots, K$ and $d=1, \dots, s$ in the hybrid fuzzy model. In general, f_{jd} can be a non-linear function. However, usually an affine function f_{jd} is used, as shown in the following equation:

$$f_{jd}(y(k), \dots, y(k-n+1), u(k), \dots, u(k-m+1)) = a_{1jd}y(k) + \dots + a_{njd}y(k-n+1) + b_{1jd}u(k) + \dots + b_{mjd}u(k-m+1) + r_{jd} \quad (5)$$

In this case f_{jd} determines the output, while $a_{1jd}, \dots, a_{njd}, b_{1jd}, \dots, b_{mjd}$ 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:

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

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

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

Here, \mathbf{I} is the unity vector.

The normalized degree of fulfillment $\beta_j(k)$ corresponding to a set of rules \mathbf{R}^{jd} for every $d=1, \dots, s$ is obtained by using a T-norm (Sugeno & Tanaka, 1991). In our case it is a simple algebraic product, given in the following equation:

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

Here, $\mu_{A_1^j}(y(k)) \cdot \dots \cdot \mu_{A_n^j}(y(k-n+1))$ denote the membership values (Babuška, 1996; Babuška & Verbruggen, 1996; Sugeno & Tanaka, 1991).

In Eq. (6), $\Theta(k)$ 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), $\Theta(k)$ is actually a function of the discrete state of the

hybrid fuzzy system in the current time-step $q(k)$.

$$\Theta(k) = \Theta(q(k)) = \begin{cases} \Theta_1 & \text{if } q(k) = 1 \\ \vdots & \vdots \\ \Theta_s & \text{if } q(k) = s \end{cases} \quad (9)$$

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

Each matrix Θ_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 $j=1, \dots, K$. Θ_d is constructed as shown in the following equation:

$$\Theta_d^T = \begin{bmatrix} a_{11d} & \dots & a_{n1d} & b_{11d} & \dots & b_{m1d} & r_{1d} \\ \vdots & & \vdots & \vdots & & \vdots & \vdots \\ a_{1Kd} & \dots & a_{nKd} & b_{1Kd} & \dots & b_{mKd} & r_{Kd} \end{bmatrix} \quad (10)$$

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

$$\psi^T(k) = [y(k) \ \dots \ y(k-n+1) \quad u(k) \ \dots \ u(k-m+1) \quad 1] \quad (11)$$

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 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. Identification of the hybrid fuzzy model

The hybrid fuzzy system with a common consequence structure (described in Section 2.2) can be expressed as a global linear

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)).

model. The input-dependent parameters given in Eq. (12) can be derived from Eq. (10).

$$\tilde{\Theta}(k) = \Theta(k)\beta(k)^T \quad (12)$$

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

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

To identify a hybrid fuzzy system means to obtain the hybrid fuzzy model parameters $a_{1jd}, \dots, a_{njd}, b_{1jd}, \dots, b_{mjd}$ and r_{jd} for each rule $\mathbf{R}^{jd}; j=1, \dots, K$ and $d=1, \dots, s$. To put it another way, all the matrices Θ_d have to be established (see Eq. (10)).

The regression matrix Ψ_{jd} for the rule \mathbf{R}^{jd} in Eq. (14) is obtained by using the whole set of input data for the hybrid fuzzy system. Here, index k runs from k_1 to $k_{P_{jd}}$, 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. (15) and (16) are actually used for constructing the regression matrix Ψ_{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. (16) is essential for obtaining suitably conditioned matrices:

$$\Psi_{jd} = \begin{bmatrix} \beta_j(k_1)\psi^T(k_1) \\ \vdots \\ \beta_j(k_{P_{jd}})\psi^T(k_{P_{jd}}) \end{bmatrix} \quad (14)$$

$$q(k) = d \quad (15)$$

$$\beta_j(k) \geq \delta \quad (16)$$

The output variable of the system y is included in the output data vector \mathbf{Y}_{jd} , which corresponds to the rule \mathbf{R}^{jd} , as written in Eq. (17). Again, only data from time-steps $(k+1)$ that comply with the conditions in Eqs. (15) and (16) are actually used for constructing the output data vector \mathbf{Y}_{jd} .

$$\mathbf{Y}_{jd} = \begin{bmatrix} \beta_j(k_1)y(k_1+1) \\ \vdots \\ \beta_j(k_1)y(k_{P_{jd}}+1) \end{bmatrix} \quad (17)$$

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

$$\beta^j(k_1)\hat{y}_p^{jd}(k+1) = \Theta_{jd}^T(\beta^j(k_1)\psi(k)) \quad (18)$$

Here, vector Θ_{jd} represents a column in the matrix Θ_d , which contains the parameters of the hybrid fuzzy model corresponding to the rule \mathbf{R}^{jd} as denoted in the following equation:

$$\Theta_{jd}^T = [a_{1jd} \dots a_{njd} \quad b_{1jd} \dots b_{mjd} \quad r_{jd}] \quad (19)$$

According to Eqs. (14), (17) and (18), the hybrid fuzzy model parameters for the rule \mathbf{R}^{jd} can be obtained using the least-squares identification method as written in the following

equations:

$$\Theta_{jd} = (\Psi_{jd}^T \Psi_{jd})^{-1} \Psi_{jd}^T \mathbf{Y}_{jd} \quad (20)$$

By calculating the hybrid fuzzy model parameters for the whole set of rules $\mathbf{R}^{jd}; j=1, \dots, K$ and $d=1, \dots, s$ 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 Ψ into Ks submatrices Ψ_{jd} . Hence, the parameters for each rule \mathbf{R}^{jd} ($j=1, \dots, K$ and $d=1, \dots, s$) are calculated separately. Due to better conditioning of the submatrices Ψ_{jd} , compared to the conditioning of the whole data matrix Ψ , 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 (Babuška, 1996; Babuška & Verbruggen, 1996; Sugeno & Tanaka, 1991; Takagi & Sugeno, 1985).

The described instantaneous linearization generates the parameters of the global linear model (see Eq. (12)), which depends on the antecedents of the hybrid fuzzy system $q(k), y(k), \dots, y(k-n+1), u(k), \dots, u(k-m+1)$. 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. Model predictive control of systems with discrete inputs based on a reachability analysis

Model predictive control is an approach where a model of the system is used to predict the future evolution of the system (Camacho & Bordons, 1998; Maciejowski, 2002). The most appropriate input vector is established and applied for every time-step. Its determination is an optimization problem that is solved within a finite horizon H , that is, for a pre-specified number of time-steps ahead. For each time-step k a sequence of optimal input vectors (21) is acquired, which minimizes the selected cost function while considering the eventual constraints of the inputs, outputs and system states. However, only the first vector of the optimal sequence is actually applied during the current time-step. In the next time-step, a new optimal sequence is determined, and so forth.

$$U_k^{k+H-1} = \{u(k), u(k+1), \dots, u(k+H-1)\} \quad (21)$$

4.1. Tree of evolution

Since the proposed control algorithm is limited to systems with discrete inputs only, the possible evolution of the system over time-steps h up to a maximum prediction horizon H can be illustrated by a tree of evolution, as shown in Fig. 1 for $H=4$ and 3 input vectors. The nodes of the tree represent reachable states, and branches connect two nodes if a transition exists between the corresponding states.

For a given root-node V_1 , representing the initial state $x_i = x(k|k)$ and $q_i = q(k|k)$, the reachable states are computed and

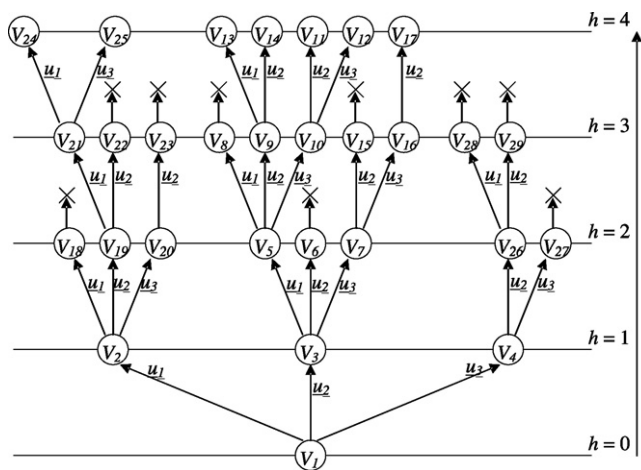


Fig. 1. Example of an explored tree of evolution. The optimal node is V_{14} , therefore the input $u_{opt} = u_2$ is selected and applied.

inserted in the tree as nodes V_i , where i indexes the nodes as they were successively computed. The notation $(k_1|k_2)$ denotes the time-step of the current node in the MPC algorithm (k_1) and the time-step, in which the algorithm started (k_2), that is, the actual time-step in the control-process.

A cost value J_i is associated with each new node, and based on the cost value the most promising node is selected. After labelling the node as explored, new reachable states emerging from the selected node are computed. The construction of the tree of evolution continues upwards first until one of the following conditions occurs:

- The value of the cost function (see Section 4.4) at the current node is greater than the current optimal one ($J_i \geq J_{opt}$).²
- The maximum step horizon has been reached ($h=H$).

If the first condition occurs, the node is labelled non-promising and thus eliminated from further exploration. On the other hand, if the node satisfies the second condition only, it becomes the new current optimal node ($J_{opt} = J_i$), whereas the sequence of input vectors leading to it becomes the current optimal one.

The exploration continues from the topmost step horizon, where unexplored nodes can be found, and so forth, until all the nodes are explored and the optimal input vector $u_{opt}(k)$ can be derived from the current optimal sequence. The optimal input vector is applied to the system $u_{opt}(k)$ and the whole procedure is repeated at the next time-step $k + 1$.

4.2. Complexity of the control problem

The complexity of the control problem with discrete inputs is primarily subject to the maximum prediction horizon (H) and the number of discrete inputs with the associated possible input val-

ues. Let us denote the first input u_1 and the associated number of discrete values m_1 (and so on). Let us assume there are l inputs. Since there are no continuous inputs, the number of combinations of input-vector values can be calculated as in Eq. (22). The worst-case complexity of the system is therefore proportional to the value in Eq. (23). The complexity grows exponentially with the number of combinations of input-vector values M_0 and the maximum prediction horizon H .

$$M_0 = m_1, \dots, m_l \quad (22)$$

$$C_0 = M_0^H \quad (23)$$

Due to physical and technological constraints it is usually possible to select only a limited number (M) of combinations of input-vector values, where $M < M_0$. The worst-case complexity of the same control problem and consequently the time needed to solve the optimization problem can thus be reduced by the factor shown in the following equation:

$$\frac{C_M}{C_0} = \left(\frac{M}{M_0} \right)^H \quad (24)$$

4.3. Reachability analysis

Generally, it is not always possible to apply every input vector from the selection mentioned above, for example branch u_1 from node V_4 (see Fig. 1). Some of the input vectors, for instance, may lead the system into undesirable states. In every time-step, such input vectors must be detected and omitted, which can be done by means of a reachability analysis. However, many of the reachable states do not lead to an optimal solution or, to put it another way, a better solution has already been found. Therefore, it is reasonable to detect and eliminate such non-promising³ states from further exploration as early as possible, in order to reduce the complexity of the control problem. The algorithm is a kind of *branch and bound* procedure, which involves the generation of a tree of evolution that was described in Section 4.1.

The inputs, outputs and system states can be constrained for a number of reasons (either physical constraints due to the nature of the system, or constraints in connection with quality or safety in control). The physical constraints must be included in the model and are considered when building the tree of evolution by means of reachability analysis. On the other hand, the control-related constraints are usually considered in the cost function.

4.4. Cost function

Cost-function selection has a great influence on the behavior of the system, on the size of the fully explored tree of evolution, and hence on the computational complexity of the control problem. Generally, the cost function can be described as in the

² Before beginning the exploration of the tree of evolution, the initial value of the current optimal node is set to infinity $J_{opt} = \infty$.

³ Non-promising states are states that are definitely not leading to the optimal solution of the control problem.

following equation:

$$J_i = J(X_k^{k+h-1}, Q_k^{k+h-1}, U_k^{k+h-2}, k, h)$$

for $h = 1, 2, \dots, H$;

where $X_k^{k+h-1} = \{\underline{x}(k), \underline{x}(k+1), \dots, \underline{x}(k+h-1)\}$;

$Q_k^{k+h-1} = \{q(k), q(k+1), \dots, q(k+h-1)\}$;

where \underline{x} is the vector of continuous states and q

is the discrete state of the system (25)

As stated in Section 4.1, it is desirable to detect and eliminate the non-promising states from further exploration as early as possible. Since the detection is based on comparing J_i to J_{opt} , it must be ensured that no better solution than the current optimal one can be found by continuing the exploration from the eliminated node. Therefore, the cost value has to be *monotonically increasing* with the prediction-step h , as in the following equation:

$$J(X_k^{k+h-1}, Q_k^{k+h-1}, U_k^{k+h-2}, k, h-1) \leq J(X_k^{k+h}, Q_k^{k+h-1}, U_k^{k+h-1}, k, h); \quad \text{for } h = 2, 3, \dots, H \quad (26)$$

In (27)–(29) we have proposed a universally usable form of cost function that can easily be applied to most of the systems met in practice.

$$J(X_k^{k+h}, Q_k^{k+h-1}, U_k^{k+h-1}, k, h) = J(X_k^{k+h-1}, Q_k^{k+h-1}, U_k^{k+h-2}, k, h-1) + f(\underline{x}(k+h|k), q(k+h|k), \underline{u}(k+h-1|k), k, h); \quad \text{for } h = 1, 2, \dots, H \quad (27)$$

$$f(\underline{x}(k+h|k), q(k+h|k), \underline{u}(k+h-1|k), k, h) \geq 0; \quad \text{for } h = 1, 2, \dots, H \quad (28)$$

$$J(X_k^k, Q_k^k, \{\}, k, 0) = 0; \quad \text{for } h = 0 \quad (29)$$

The function f (28) is an arbitrary non-negative function that estimates the quality of control. Its value is added to the sum of the cost functions calculated along the path from the root-node to the current node in the tree of evolution. Function f basically penalizes the predicted deviations of system states from the reference trajectory (e.g., by calculating the sum of pondered squares of deviations), therefore, the cost-function value increases more rapidly along non-optimal paths in the tree of evolution. Since the function f does not have any special requirements apart from (28), it is relatively easy to determine a suitable cost function for an actual problem.

It is trivial to prove that the proposed cost function complies with condition (26).

4.5. Holding the inputs through a number of time-steps

The maximum time of prediction reached in the control algorithm T_{pred} depends on two parameters (see Eq. (30)): the

sampling time T_S and the maximum prediction horizon H .

$$T_{pred} = HT_S \quad (30)$$

Since the sampling time T_S is determined by factors that in most cases cannot be changed,⁴ the only way to reach a longer time of prediction T_{pred} seems to be by increasing the maximum prediction horizon H . However, as stated in Section 4.2, by doing that the complexity of the control problem increases drastically.

In many cases it is possible (or even recommendable) not to change the input-vector values each sampling time. For instance, when the sampling time is relatively short, actuators could get overloaded. For this reason we have proposed an approach where the same input-vector values are kept through several (Z) time-steps or, to put it another way, the input-vector values can change only every Z time-steps.

In this case the maximum reachable time of prediction T_{pred} is determined in (31). Here, $H_Z = H/Z$ is the new maximum prediction horizon required to reach the desired maximum time of prediction T_{pred} :

$$T_{pred} = T_S Z H_Z \quad (31)$$

The complexity of the control problem (provided T_{pred} is constant) can thus be reduced by the factor shown in Eq. (32), or rather, a longer time of prediction T_{pred} can be reached, whereas the complexity increases linearly instead of exponentially:

$$\frac{C_Z}{C_M} = \frac{(MZ)^{H_Z}}{M^H} = \left(\frac{Z}{M^{Z-1}} \right)^{H_Z} \quad (32)$$

By holding the inputs through Z time-steps several objectives can be achieved:

- Decrease the maximum prediction horizon H required to reach the desired maximum time of prediction T_{pred} and thus reduce the complexity of the control problem.
- Enable a longer maximum time of prediction T_{pred} (with the same equipment).
- Enable a shorter sampling time T_S while retaining the maximum time of prediction T_{pred} (on the same equipment).
- Relieve the potentially overloaded actuators.

We believe that holding the inputs through Z time-steps can improve the control for a class of systems, where changing the input vector every sample time is not needed or wanted, for example, for stiff systems. The optimal value of the parameter Z depends on the system. The higher the value of Z , the lower the maximum frequency of input changes and vice versa. Its determination is a trial and error procedure that can be done using simulation.

Due to a longer reachable maximum time of prediction T_{pred} , we expect that increasing Z should cause the simulation results to improve and eventually reach the optimum. On the other hand, further increasing of the parameter Z is expected to cause a deterioration of the results, which happens because the inputs do not

⁴ For example, the dynamics of the system, the accuracy of prediction needed and the eventual sampling time of the sensors used, and so forth.

change frequently enough to ensure a satisfactory control (Karer, Mušič, & Zupančič, 2005).

5. Batch reactor

The usability of the control algorithm in combination with the hybrid fuzzy model has been verified 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.

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 ϕ . The dynamics of the system depend on the physical properties of the batch reactor, namely, 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, index w denotes the water jacket). λ is the thermal conductivity; S the contact area; 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 ($T_{in}=T_C=12^\circ\text{C}$), whereas if $k_C=0$ and $k_H=1$, the input water is hot ($T_{in}=T_H=75^\circ\text{C}$).

The ratio of fresh input water to reflux water is controlled by the third input, that is, by the position of the mixing valve k_M . There are six possible ratios that can be set by the mixing valve. The fresh input water share can be either 0, 0.01, 0.02, 0.05, 0.1 or 1.

We are therefore dealing with a multivariable system with three discrete inputs (k_M , k_H and k_C) 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.

5.1. Modelling and identification of the batch reactor

The model of the batch reactor is obtained in two steps:

- The multivariable system is divided into simpler MISO sub-systems.
- Each discrete-time submodel is identified individually.

It is also feasible to construct a model with only one operating mode. However, since the model parameters are obtained by matrix inversion, this could result in numerical problems due to bad conditioning of the matrices used in identification. In addition, more parameters would have to be estimated (as a result of a higher dimension of Θ), which impairs the quality of identification results. Therefore, by using the proposed hybrid fuzzy model structure we have the advantage of avoiding the possible numerical problems in identification.

The heat-flow in the reactor system can be divided as follows:

- Heat conduction between the reactor's core and the reactor's water jacket.
- Heat conduction between the reactor's water jacket and the surroundings.
- Heat convection due to an influx of fresh input water into the reactor's water jacket.
- Heat convection due to outflow of the heating medium from the reactor's water jacket.

In this manner it is possible to divide the unknown system into simpler MISO subsystems and thus apply some prior knowledge of the system in the modelling. Hence, such an identification of the unknown system is regarded as gray-box identification.

In order to carry out the identification, suitable input–output data of the batch-reactor process has to be obtained. Therefore, we generated a pseudo-random input signal and applied it to the batch-reactor process. A sampling time of $T_S=10\text{ s}$ was used. The inputs at each time-step were randomly selected so that the input signal was a sort of step function. The frequency of input changes must be high enough to cover the whole frequency range of the system that we are interested in. This way, the identification data can characterize the complete dynamics of the system.

The time needed to acquire the data used for identification was longer than one batch cycle. However, this does not affect the dynamics of the system, because there are no endo- or exothermic reactions in the reactor. The reactor core contains granulated material that is stirred, heated and cooled in order to produce a homogenous compound.

We recorded the measurable outputs, that is, the reactor's core temperature T and the reactor's water-jacket temperature T_w . The outputs are shown in Fig. 3. A close-up is shown in Fig. 4.

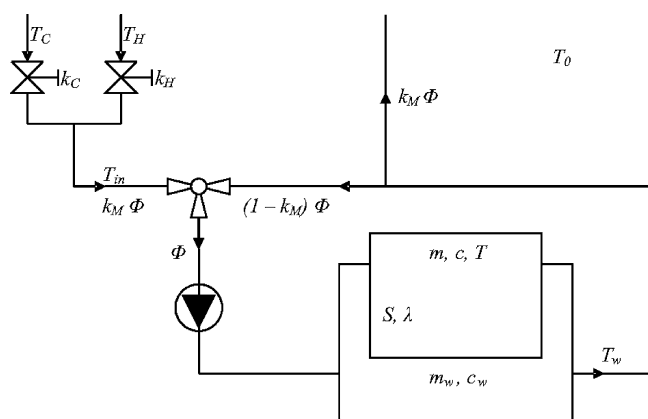


Fig. 2. Schematic representation of the batch reactor.

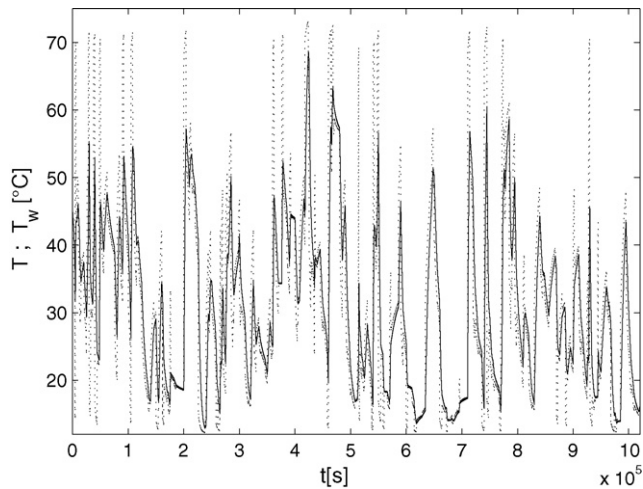


Fig. 3. Core temperature T (solid line) and water-jacket temperature T_w (dotted line).

5.2. Temperature in the reactor's core

The temperature in the reactor's core is influenced only by the heat conduction between the reactor's core and the reactor's water jacket, which depends on the temperature difference between the reactor's water jacket T_w and the reactor's core T . Therefore, a first-order MISO submodel can be presumed, as shown in Eq. (33). Here, the regressor consists of the temperature in the water jacket $T_w(k)$ and the temperature in the core $T(k)$ in the current time-step.

$$\hat{T}(k+1) = f(T_w(k), T(k)) \quad (33)$$

Since we have surmised that the heat conduction is proportional to the temperature difference between the reactor's core and the reactor's water jacket, we can presume a linear model, as in the following equation:

$$\hat{T}(k+1) = \Theta^T [T_w(k) T(k)]^T \quad (34)$$

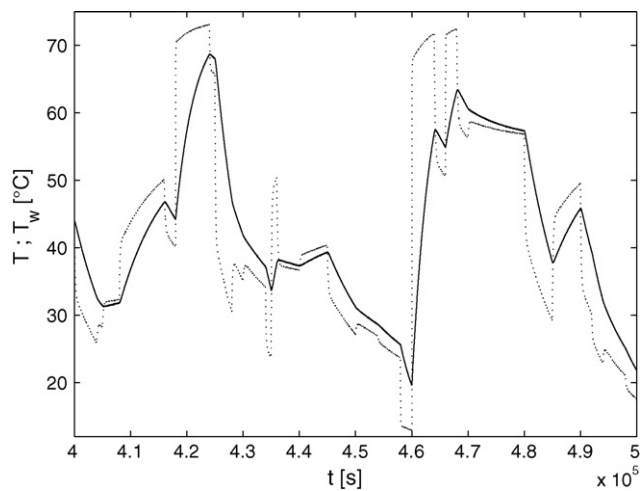


Fig. 4. Core temperature T (solid line) and water-jacket temperature T_w (dotted line).

After conducting a least-squares identification we obtain the system parameters Θ .

$$\Theta^T = [0.0033 \quad 0.9967] \quad (35)$$

5.3. Temperature in the reactor's water jacket

The temperature in the reactor water jacket T_w is influenced by all the heat-flow sources previously mentioned. Therefore, a MISO submodel can be presumed, as shown in Eq. (36). Here, the regressor consists of the values in the current time-step k of the temperature in the water jacket $T_w(k)$, the temperature in the core $T(k)$, the fresh input water inflow at the mixing valve $k_M(k)$, and the position of the cold-water and hot-water on/off valves $k_C(k)$ and $k_H(k)$, respectively.

$$\hat{T}_w(k+1) = F(T_w(k), T(k), k_M(k), k_C(k), k_H(k)) \quad (36)$$

The modelling and parameter estimation of the subsystem were carried out in a similar manner to that described in Sections 2.2 and 3.

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

By further subdividing the MISO model in Eq. (36) we have established the discrete part of the hybrid fuzzy model, which is given in the following equation:

$$q(k) = q(k_C(k), k_H(k)) = \begin{cases} 1 & \text{if } k_C(k) = 0 \wedge k_H(k) = 1 \\ 2 & \text{if } k_C(k) = 1 \wedge k_H(k) = 0 \end{cases} \quad (37)$$

Next, the membership functions have to be defined. The system is fuzzified with regard to the temperature in the reactor's water jacket $T_w(k)$. Simple triangular functions are used, as shown in Fig. 5.

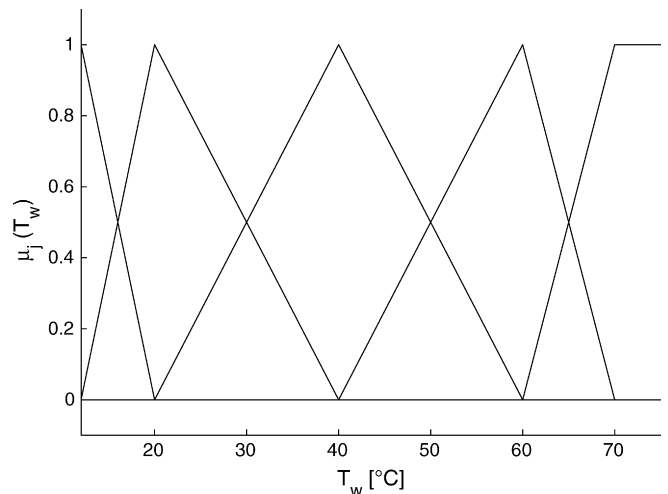


Fig. 5. Membership functions.

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

The rule base of the hybrid fuzzy model is hence given in Eq. (38). We presume that a local system corresponding to an individual rule \mathbf{R}^{jd} is affine.

$$\mathbf{R}^{jd} : \quad \text{if } q(k) \text{ is } Q_d \text{ and } T_w(k) \text{ is } A_1^j \\ \text{then } T_w(k+1) = a_{1jd}T_w(k) + a_{2jd}T(k) + b_{1jd}k_M(k) + r_{jd}, \\ \text{for } j = 1, \dots, 5 \text{ and } d = 1, 2 \quad (38)$$

The output of the hybrid fuzzy model corresponding to rule \mathbf{R}^{jd} can therefore be formulated as in the following equation:

$$\hat{T}_w^{jd}(k+1) = \Theta_{jd}^T [T_w(k) \quad T(k) \quad k_M(k) \quad 1]^T \quad (39)$$

After conducting a least-squares identification for each rule \mathbf{R}^{jd} respectively (see Section 3), we obtain the matrices with the system parameters below for both operating modes (Θ_1 for $q=1$ and Θ_2 for $q=2$).

$$\Theta_1 = \begin{bmatrix} 0.9453 & 0.9431 & 0.9429 & 0.9396 & 0.7910 \\ 0.0376 & 0.0458 & 0.0395 & 0.0339 & 0.0225 \\ 19.6748 & 16.7605 & 10.5969 & 3.9536 & 1.6856 \\ 0.3021 & 0.2160 & 0.5273 & 1.2701 & 12.0404 \end{bmatrix} \quad (40)$$

$$\Theta_2 = \begin{bmatrix} 0.9803 & 0.9740 & 0.9322 & 0.9076 & 0.8945 \\ 0.0025 & 0.0153 & 0.0466 & 0.0466 & 0.0111 \\ -0.0704 & -0.6956 & -7.8013 & -12.2555 & -18.7457 \\ 0.2707 & 0.2033 & 0.5650 & 1.9179 & 5.6129 \end{bmatrix} \quad (41)$$

To sum up, the output of the model of the temperature in the reactor's water jacket is written in a compact form in the following equation:

$$\hat{T}_w(k+1) = \beta(k)\Theta^T(k) [T_w(k) \quad T(k) \quad k_M(k) \quad 1]^T \quad (42)$$

6. Control

Once we have obtained the model of the process we wish to control, we can move on to establishing the input matrix P , which contains every allowed combination of input-vector values (see Eq. (43)). Here, each column represents an input vector. The rows of the respective input vectors have the following meaning:

- The first row denotes the mixing-valve input $k_M \in \{0, 0.01, 0.02, 0.05, 0.1, 1\}$.
- The second row denotes the cool-water on/off valve input $k_C \in \{0, 1\}$.
- The third row denotes the hot-water on/off valve input $k_H \in \{1, 0\}$.

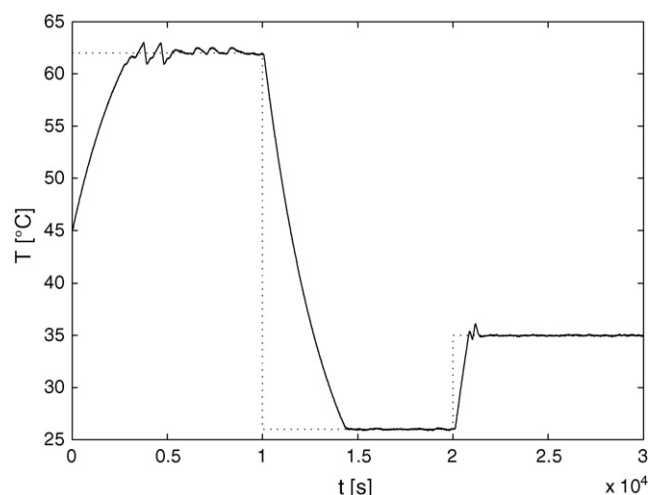


Fig. 6. Core temperature T (solid line) and reference temperature T_{ref} (dotted line).

$$P = \begin{bmatrix} 0 & 0.01 & 0.02 & 0.05 & 0.1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 \end{bmatrix} \quad (43)$$

In the last step we establish a suitable cost function (see Section 4.4). According to the prepositions in Eqs. (27)–(29), a simple cost function that takes into consideration the square of deviation of the core temperature T from the reference temperature T_{ref} , is given in Eq. (44). In such a manner the control performance can be quantitatively estimated along the tree of evolution.

$$J(X_k^{k+h}, Q_k^{k+h}, U_k^{k+h-1}, k, h) \\ = J(X_k^{k+h-1}, Q_k^{k+h-1}, U_k^{k+h-2}, k, h-1) \\ + (T(k+h) - T_{ref}(k+h))^2 \quad (44)$$

The results of the experiment are shown in Figs. 6 and 7. The maximum prediction horizon was $H_Z = 4$ and the number of time-steps, through which the inputs are held, was $Z = 15$.

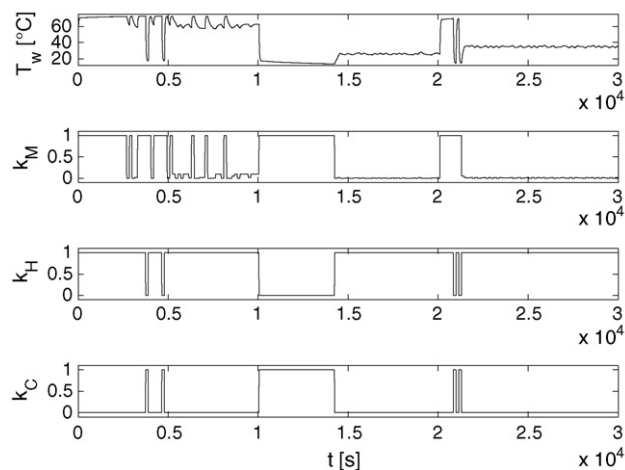


Fig. 7. Other system states.

We can ascertain that the reference temperature T_{ref} was well followed by the actual core temperature T . However, there are still three obvious aspects where improvement is needed.

- The valves were moving too much.
- In several time-frames the control was carried by the on/off valves, whereas the mixing valve was relatively open. However, such behavior had been expected, for the on/off valves can be regarded as another mixing valve, which has a greater influence on the core temperature T than the actual mixing valve.
- The total consumption of fresh input water was too high. Again, this is due to the fact that the mixing valve was fully open for too long during the experiment.

We tried to improve the problematic aspects of the control by including a sort of penalty for the movement of the valves in the cost function in Eq. (44). The modified cost function is established in the following equation:

$$\begin{aligned}
 J(X_k^{k+h}, Q_k^{k+h}, U_k^{k+h-1}, k, h) \\
 = J(X_k^{k+h-1}, Q_k^{k+h-1}, U_k^{k+h-2}, k, h-1) \\
 + w_1(T(k+h) - T_{\text{ref}}(k+h))^2 \\
 + w_2(k_C(k+h)k_H(k+h-1)) \\
 + w_3|k_M(k+h) - k_M(k+h-1)|k_H(k+h-1) \quad (45)
 \end{aligned}$$

The summands in the cost function were weighted as follows:

- The square of the deviation of the core temperature T from the reference trajectory T_{ref} was weighted according to the choice of the parameter Z , in order to enable the control performance comparison among simulations employing different Z . $w_1 = 1/Z$.
- w_2 weights the event of changing fresh input water from hot to cool. This prevents the changes of the on/off valves when there is no negative step in the reference temperature signal T_{ref} . The weight was set as high as possible, but low enough to allow the on/off valves to change when a reference step occurs. $w_2 = 15$.
- Next, w_3 was decreased, so that the control would not be taken over exclusively by the on/off valves. $w_3 = 0.03$.

The results of the experiment are shown in Figs. 8 and 9.

We can ascertain that the reference temperature T_{ref} was again well followed by the actual core temperature T . What is more, by setting the cost function correctly, all three problematic aspects of the control were satisfactorily solved.

6.1. Comparison between MPC employing a hybrid fuzzy model and a hybrid linear model

In order to compare MPC employing a hybrid fuzzy model to MPC employing a hybrid linear model, we have to attain the linear model for the temperature in the reactor's water jacket

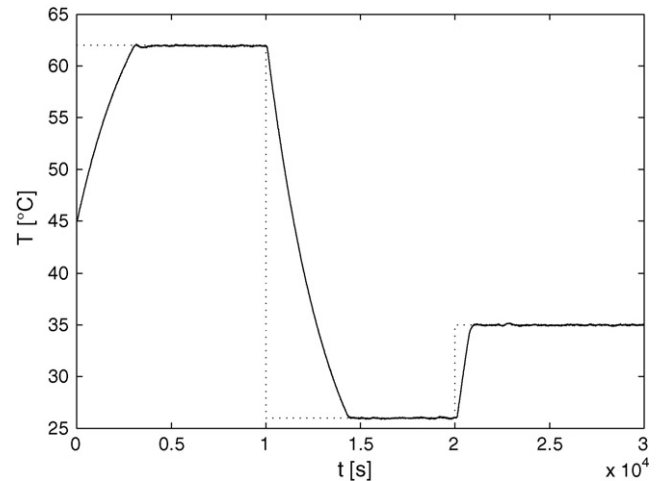


Fig. 8. Core temperature T (solid line) and reference temperature T_{ref} (dotted line).

(see Eq. (36)). The linear model can easily be derived from the fuzzy model by linearizing it close to the center of the fuzzyfied operating range. In other words, we have used a fixed degree of fulfillment vector $\beta = [0 \ 0 \ 1 \ 0 \ 0]$. Therefore, only one of the fuzzy regions was taken into account when identifying the linear model. In this sense, the model parameters are given in Eqs. (46) and (47), whereas the output is obtained according to Eq. (48).

$$\Theta_{1,\text{lin}} = [0.9429 \ 0.0395 \ 10.5969 \ 0.5273]^T \quad (46)$$

$$\Theta_{2,\text{lin}} = [0.9322 \ 0.0466 \ -7.8013 \ 0.5650]^T \quad (47)$$

$$\begin{aligned}
 \hat{T}_w(k+1) \\
 = \begin{cases} \Theta_{1,\text{lin}}^T [T_w(k) \ T(k) \ k_M(k) \ 1]^T & \text{if } k_C = 0 \wedge k_H = 1 \\ \Theta_{2,\text{lin}}^T [T_w(k) \ T(k) \ k_M(k) \ 1]^T & \text{if } k_C = 1 \wedge k_H = 0 \end{cases} \quad (48)
 \end{aligned}$$

The results of the experiment with the linear model employed in MPC are shown in Figs. 10 and 11. Again, the cost function in Eq. (45) was used.

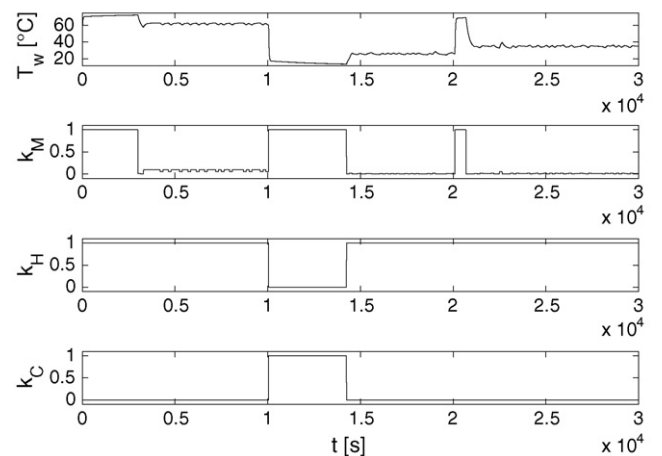


Fig. 9. Other system states.

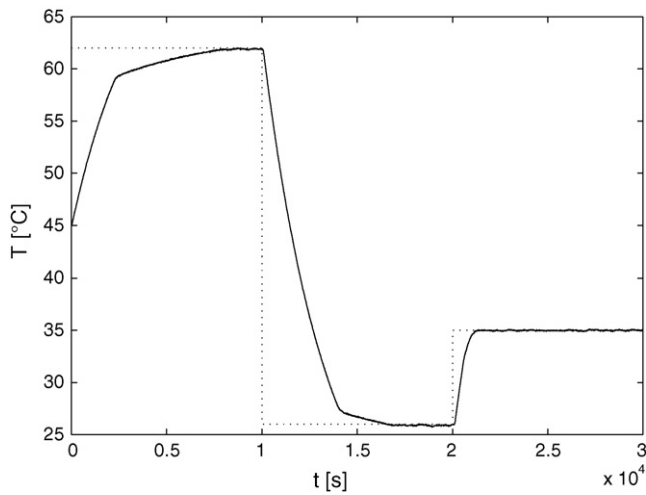


Fig. 10. Core temperature T (solid line) and reference temperature T_{ref} (dotted line).

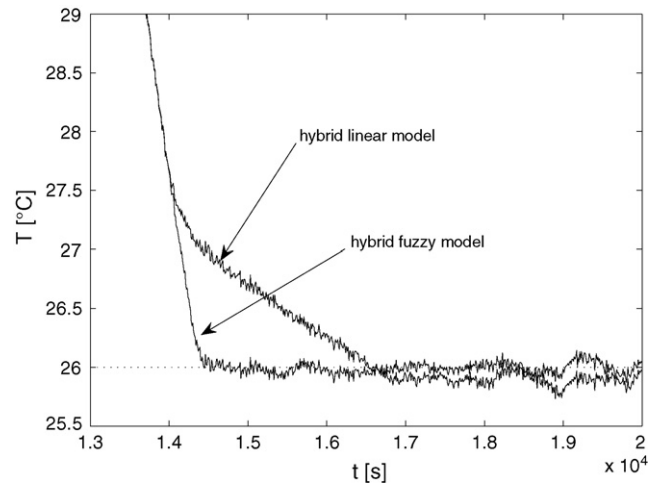


Fig. 13. Core temperature T : comparison between hybrid fuzzy model (lower trajectory) and hybrid linear model employed in MPC. $T_{ref} = 26^\circ\text{C}$.

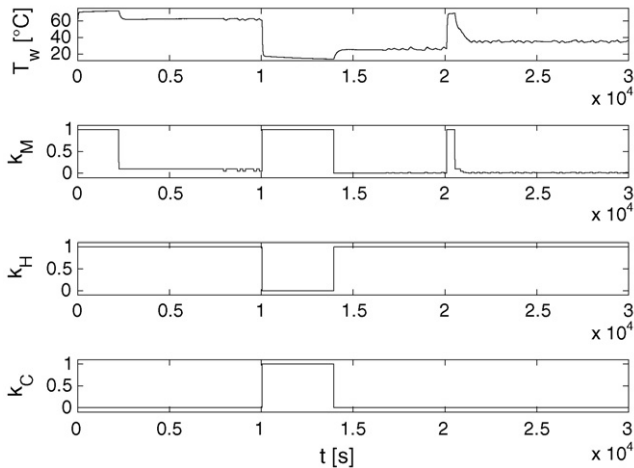


Fig. 11. Other system states.

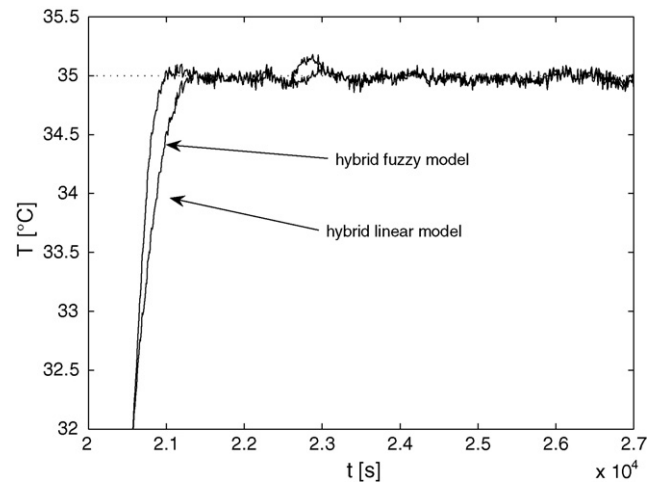


Fig. 14. Core temperature T : comparison between hybrid fuzzy model (higher trajectory) and hybrid linear model employed in MPC. $T_{ref} = 35^\circ\text{C}$.

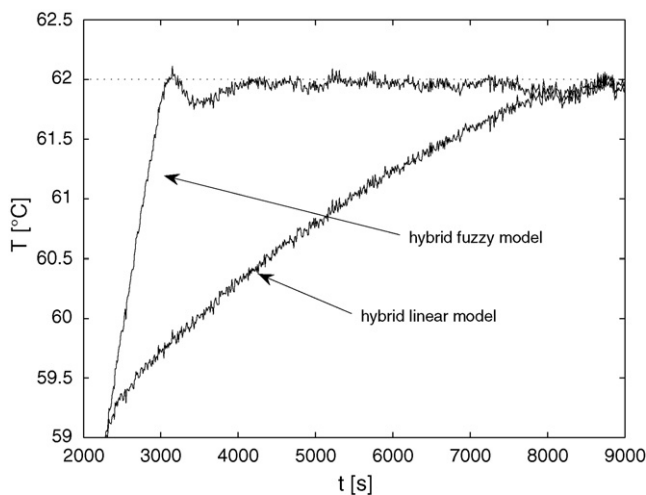


Fig. 12. Core temperature T : comparison between hybrid fuzzy model (higher trajectory) and hybrid linear model employed in MPC. $T_{ref} = 62^\circ\text{C}$.

A close-up of the relevant sections is presented in Fig. 12, where the reference temperature is $T_{ref} = 62^\circ\text{C}$, and in Fig. 13, where the reference temperature is $T_{ref} = 26^\circ\text{C}$. We can conclude that the MPC algorithm employing the hybrid fuzzy model clearly outperforms the case where the linear model is used.

In the third part of the experiment, where the reference temperature is $T_{ref} = 35^\circ\text{C}$, the hybrid linear model approximates the identified system more adequately. The control performance is therefore more comparable to the case in which the hybrid fuzzy model is employed. That said, still better performance is achieved in the latter case, as can be seen in Fig. 14.

7. Conclusion

A hybrid fuzzy modelling approach for the MPC of non-linear hybrid systems with discrete inputs based on a reachability analysis was treated.

In order to implement a MPC algorithm, a suitable model of the process we are dealing with is needed. In the paper, a hybrid fuzzy modelling approach with a compact formulation

was introduced. The hybrid system hierarchy was explained and the Takagi–Sugeno fuzzy formulation for the hybrid fuzzy modelling purposes was presented. An efficient method for identifying the hybrid fuzzy model was also discussed.

A MPC algorithm suitable for systems with discrete inputs was treated. The benefits of the MPC algorithm employing a proposed hybrid fuzzy model were verified on a batch-reactor example. The results suggest that by suitably determining the cost function, satisfactory control can be attained, even when dealing with complex hybrid–non-linear-stiff systems such as the batch reactor.

Finally, a comparison between MPC employing a hybrid linear model and a hybrid fuzzy model was made. It was established that the latter approach clearly outperforms the approach where a linear model is used.

References

- Babuška, R. (1996). *Fuzzy modelling and identification*, PhD Thesis, Delft.
- Babuška, R., & Verbruggen, H. B. (1996). An overview of fuzzy modelling for control. *Control Engineering Practice*, 4(11), 1593–1606.
- Bemporad, A., & Morari, M. (1999). Control of systems integrating logic, dynamics and constraints. *Automatica*, 35(3), 407–427.
- Camacho, E. F., & Bordons, C. (1998). Model predictive control. In *Advanced textbooks in control and signal processing*. London: Springer-Verlag.
- Castro, J. (1995). Fuzzy logic controllers are universal approximators. *IEEE Transactions on Systems, Man and Cybernetics*, 25, 629–635.
- Girimonte, D., & Babuška, R. (2004). Structure for nonlinear models with mixed discrete and continuous inputs: A comparative study. In *Proceedings of IEEE international conference on system, man and cybernetics* (pp. 2392–2397).
- Girosi, F., & Poggio, T. (1990). Networks and the best approximation property. *Biological Cybernetics*, 63, 169–176.
- Heemels, W., Schutter, B. D., & Bemporad, A. (2001). Equivalence of hybrid dynamical models. *Automatica*, 37(7), 1085–1091.
- Karer, G., Mušič, G., & Zupančič, B. (2005). Predictive control of temperature in a batch reactor with discrete inputs. In *Proceedings of the 2005 IEEE international symposium on intelligent control and 2005 Mediterranean conference on control and automation, IEEE* (pp. 855–860).
- Kerrigan, E. C., & Mayne, D. Q. (2002). Optimal control of constrained, piecewise affine systems with bounded disturbances. In *Proceedings of the 41st IEEE conference on decision and control*.
- Maciejowski, J. M. (2002). *Predictive control: With constraints*. Harlow: Prentice-Hall.
- Mayne, D. Q., & Raković, S. (2003). Model predictive control of constrained piecewise affine discrete-time systems. *International Journal of Robust and Nonlinear Control*, 13(3), 261–279.
- Palm, R., & Driankov, D. (1998). Fuzzy switched hybrid systems—Modelling and identification. In *Proceedings of the 1998 IEEE/ISCI/CIRA/SAS joint conference* (pp. 130–135).
- Potočnik, B., Mušič, G., & Zupančič, B. (2005). Model predictive control of discrete time hybrid systems with discrete inputs. *ISA Transactions*, 44(2), 199–211.
- Qin, Y., & Jia, L.-M. (2002). Fuzzy hybrid control and its application in complex combustion processes. In *Proceedings of the 2002 IEEE international conference on artificial intelligence systems (ICAIS'02), IEEE* (p. 78).
- Sjoberg, J., Zhang, A., Ljung, L., Benveniste, A., Delyon, B., Glorennec, P.-Y., et al. (1995). Nonlinear black-box modelling in system identification: A unified overview. *Automatica*, 31(12), 1691–1724.
- Škrjanc, I., & Matko, D. (2001). Fuzzy predictive functional control in the state space domain. *Journal of Intelligent and Robotic Systems*, 31, 283–297.
- Sontag, E. (1981). Nonlinear regulation: The piecewise linear approach. *IEEE Transactions on Automatic Control*, 26(2), 346–358.
- Sugeno, M., & Tanaka, K. (1991). Successive identification of a fuzzy model and its application to prediction of a complex system. *Fuzzy Sets and Systems*, 42, 315–334.
- Takagi, T., & Sugeno, M. (1985). Fuzzy identification of systems and its application to modelling and control. *IEEE Transactions on Systems, Man and Cybernetics*, 15, 116–132.
- van der Schaft, A., & Schumacher, H. (1999). An introduction to hybrid dynamical systems. *Lecture Notes in Control and Information Sciences*, 251, v–vii.
- Witsenhausen, H. S. (1966). A class of hybrid-state continuous time dynamic systems. *IEEE Transactions on Automatic Control*, 11(2), 161–167.