Available online at www.sciencedirect.com
ScienceDirect
ELSEVIER
Nonlinear Analysis: Hybrid Systems xx (xxxx) xxx–xxx
Nonlinear Analysis: Hybrid Systems
www.elsevier.com/locate/nahs

Model predictive control of nonlinear hybrid systems with discrete inputs employing a hybrid fuzzy model

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

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

Received 18 May 2006; accepted 6 June 2006

Abstract

The paper deals with model predictive control (MPC) of nonlinear hybrid systems with discrete inputs based on reachability analysis. In order to implement a MPC algorithm, a model of the process that we are dealing with is needed. In the paper, a hybrid fuzzy modelling approach is proposed. The hybrid system hierarchy is explained and the Takagi–Sugeno fuzzy formulation for hybrid fuzzy modelling purposes is tackled. An efficient method of identification of the hybrid fuzzy model is also discussed.

An algorithm that is – due to its MPC nature – suitable for controlling a wide spectrum of systems (provided that they have discrete inputs only) is presented.

The benefits of the algorithm employing a hybrid fuzzy model are 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–nonlinear–stiff systems such as the batch reactor.

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

Keywords: Fuzzy systems; Hybrid systems; Model predictive control

1. Introduction

Dynamic systems, which involve continuous and discrete states, are called *hybrid systems*. Most industrial processes contain 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 the topic was not treated systematically [24]. Lately however, hybrid systems have received much attention from the computer science and control community.

In recent years, some hybrid fuzzy identification methods have been proposed. A hierarchical identification of the resulting fuzzy switched system is introduced in [17]. Furthermore, two structure-selecting methods for nonlinear models with mixed discrete and continuous inputs are presented in [10].

Many control methods employ system models described as *mixed logical dynamical* (MLD) [5]. A lot of interest has also been devoted to *piecewise affine* (PWA) systems [21], which have been proven to be equivalent to many

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

1751-570X/$ - see front matter © 2007 Published by Elsevier Ltd
doi:10.1016/j.nahs.2006.06.005

classes of hybrid systems [12]. In addition, MLD systems can be converted into PWA form as well [4]. The optimal control problem for discrete-time PWA systems can be converted to a mixed-integer optimization problem and solved on-line [6,16]. On the other hand, in [14,3,7] the optimal control problem for PWA systems is treated by solving a number of multiparametric programs off-line. A solution in the form of a PWA state feedback law is obtained, which can be efficiently implemented on-line.

The methods consider mainly systems with continuous inputs, despite the fact that the *(multiparametric) mixed-integer linear/quadratic programming* (mp-MIQP/MILP) based solutions can be applied to systems with discrete inputs as well. However, the computational complexity increases drastically with the number of discrete states, and thus the methods can become computationally too demanding. An algorithm for efficient model predictive control of hybrid systems with discrete inputs only is proposed in [18].

Most of previous work related to model predictive control of hybrid systems is based on linear models. In this paper, we propose the use of a hybrid fuzzy model for model predictive control of nonlinear hybrid systems with discrete inputs based on reachability analysis [20].

The outline of the paper is as follows. In Section 2, modelling of a hybrid fuzzy model is discussed. It is followed by Section 3, where identification of the model is dealt with. In Section 4, an algorithm for model predictive control of systems with discrete inputs based on reachability analysis is treated. In the following sections, a batch reactor process is introduced. Modelling and identification of the process are tackled and the control algorithm is verified by means of a simulation experiment. Finally, a comparison between MPC employing a hybrid linear model and a hybrid fuzzy model is carried out.

2. Modelling of a hybrid fuzzy model

In order to implement the control algorithm a model of the treated process is needed. The complex hybrid and nonlinear nature of many processes met in practice causes problems both in modelling and identification; therefore obtaining a model that is suitable for MPC is often a difficult task. Classical modelling and especially identification methods that originate from linear-system theory are inadequate for treating such systems. Hence, the need for special methods and formulations in dealing with hybrid systems is more than evident.

Dynamic systems are usually modelled by feeding back delayed input and output signals. In the discrete-time domain a common nonlinear model structure is a NARX (Nonlinear AutoRegressive with eXogenous inputs) model [19], 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.

2.1. Hybrid system hierarchy

As already mentioned, many processes met in practice demonstrate a hybrid nature, which means that the continuous dynamics are interlaced with the discrete dynamics. A special class of such systems is called switched systems, where the continuous states remain continuous even when the discrete states are changed, i.e. no jumps of the continuous state vector are allowed. In this paper we are dealing with a 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).

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

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

Here $\mathbf{x} \in \mathbb{R}^n$ is the continuous state vector and $\mathbf{u} \in \mathbb{R}^n$ is the input vector. $q \in \mathbb{Q}$ (where $\mathbb{Q} = \{1, \ldots, s\}$) is the discrete state, which defines the switching region. 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 $\mathbf{f}_q$.

Please cite this article in press as: G. Karer, et al., Model predictive control of nonlinear hybrid systems with discrete inputs employing a hybrid fuzzy model, Nonlinear Analysis: Hybrid Systems (2007), doi:10.1016/j.nahs.2006.06.005

Eq. (3) introduces a generalization of the strict Witsenhausen hybrid system formulation [25] in the sense that the discrete state in the next time-step $q(k+1)$ depends on the input vector $\mathbf{u}(k)$ as well as on the continuous state vector $\mathbf{x}(k)$ and the current discrete state $q(k)$.

The continuous part of the system is generally nonlinear; therefore it can be modelled by a Takagi–Sugeno fuzzy subsystem as shown in Section 2.2.

Once both the discrete model and the continuous model are obtained, there are two obvious ways of merging them into a whole.

- The discrete part of the model selects which of the continuous models $\mathbf{f}_q$ to use at time-step k. The selection is based on the current discrete state $q(k)$. Hence, we are dealing with a multimodel approach.
- The discrete part of the model is included in the fuzzy formulation of the continuous part by using a corresponding number of additional crisp membership functions.

2.2. Takagi–Sugeno fuzzy formulation

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 [9,11].

The system dynamics can be formulated as a Takagi–Sugeno fuzzy model. In this instance, the rule base of the fuzzy system is given in Eq. (4).

$$\mathbf{R}^j\text{: }\textbf{if } x_1 \text{ is } A_1^j \text{ and } \ldots \text{ and } x_N \text{ is } A_N^j \textbf{ then } y = f_j(x_1,\ldots,x_N), \quad j = 1,\ldots,K. \tag{4}$$

The **if**-parts (antecedents) of the rules describe fuzzy regions in the space of input variables. Here $x_1,\ldots,x_N$ are input variables of the fuzzy system and A_i^j are fuzzy sets characterized by their membership functions. The number of relevant fuzzy rules in the fuzzy model is K.

Generally speaking, the number of fuzzy rules in the fuzzy model K depends on the number of membership functions for each antecedent variable $x_1,\ldots,x_N$. 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 for each antecedent variable $x_1,\ldots,x_N$.

The **then**-parts (consequences) are functions of inputs. Here y is an output variable. There is one function of inputs f_j defined for each rule $\mathbf{R}^j$; $j = 1,\ldots,K$, in the fuzzy model. In general, f_j can be a nonlinear function. However, usually a linear function f_j is used as shown in Eq. (5).

$$\mathbf{R}^j\text{: } y = f_j(x_1,\ldots,x_N) = a_{1j}\,x_1 + \cdots + a_{Nj}\,x_N + r_j, \quad j = 1,\ldots,K. \tag{5}$$

In this case $a_{1j},\ldots,a_{Nj}$ and r_j denote consequent parameters.

The output of the Takagi–Sugeno fuzzy model is given by the following equation.

$$\begin{aligned} y &= \sum_{j=1}^{K} \beta_j(x_1,\ldots,x_N)\cdot f_j(x_1,\ldots,x_N) \\ &= \sum_{j=1}^{K} \beta_j(x_1,\ldots,x_N)\cdot(a_{1j}\,x_1 + \cdots + a_{Nj}\,x_N + r_j), \quad j = 1,\ldots,K. \end{aligned} \tag{6}$$

Here $\beta_j(x_1,\ldots,x_N)$ represents the normalized degree of fulfillment. It is obtained by using a T-norm [22]. In our case it is a simple algebraic product given in Eq. (7).

$$\beta_j(x_1,\ldots,x_N) = \frac{\mu_{A_1^j}(x_1)\cdot\cdots\cdot\mu_{A_N^j}(x_N)}{\sum_{i=1}^{K}\mu_{A_1^i}(x_1)\cdot\cdots\cdot\mu_{A_N^i}(x_N)}. \tag{7}$$

Please cite this article in press as: G. Karer, et al., Model predictive control of nonlinear hybrid systems with discrete inputs employing a hybrid fuzzy model, Nonlinear Analysis: Hybrid Systems (2007), doi:10.1016/j.nahs.2006.06.005

Here $\mu_{A_1^j}(x_1)\dots\mu_{A_N^j}(x_N)$ denote the membership values [1,2,22]. The normalized degrees of fulfillment for the whole set of rules are written in the vector form in Eq. (8).

$$\boldsymbol{\beta}(x_1,\dots,x_N) = [\beta_1\beta_2,\dots,\beta_K]. \tag{8}$$

We assume the normalized degrees of fulfillment, which are generally time dependent, comply with Eq. (9) for every time-step k.

$$\sum_{j=1}^{K}\beta_j(k) = \boldsymbol{\beta}^{\mathrm{T}}(k)\boldsymbol{I} = 1. \tag{9}$$

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

From Eqs. (6) and (8) the output of the fuzzy system can be written as in Eq. (10).

$$y = \boldsymbol{\beta}^{\mathrm{T}}\mathbf{a}_1x_1 + \cdots + \boldsymbol{\beta}^{\mathrm{T}}\mathbf{a}_Nx_N + \boldsymbol{\beta}^{\mathrm{T}}\mathbf{r} = \sum_{i=1}^{N}\boldsymbol{\beta}^{\mathrm{T}}\mathbf{a}_ix_i + \boldsymbol{\beta}^{\mathrm{T}}\mathbf{r}. \tag{10}$$

Here $\mathbf{a}_i$ and $\mathbf{r}$ denote fuzzyfied parameters of the model as written in the following equations.

$$\begin{aligned}\mathbf{a}_i^{\mathrm{T}} &= [a_{i1}\dots a_{iK}]\\ \mathbf{r}^{\mathrm{T}} &= [r_1\dots r_K].\end{aligned} \tag{11}$$

So far we have been dealing with a general formulation of a Takagi–Sugeno fuzzy system. The fuzzy model represents a universal mapping between its inputs $(x_1,\dots,x_N)$ and its output y. In the field of MPC, however, we are interested in obtaining a predicted output in the next time-step $\hat{y}_p(k+1)$.

According to Eqs. (1), (4) and (10), we can use the fuzzy model as a predictor by treating the fuzzy model output at time-step $ky(k)$ as the predicted output in the next time-step $\hat{y}_p(k+1)$. In this case, the fuzzy model inputs at time-step $k(x_1(k),\dots,x_N(k))$ denote a suitable regressor, which depends on the order of the system n and the number of relevant inputs m.

For instance, let us assume we are dealing with a first-order SISO system ($n=1, m=1$). In this case the NARX formulation (1) can be written as follows.

$$\hat{y}_p(k+1) = F(y(k),u(k)). \tag{12}$$

The regression vector in this case consists of the current output $y(k)$ and the current input $u(k)$ and the output equation of the fuzzy model can be derived from (10) as follows.

$$\hat{y}_p(k+1) = \boldsymbol{\beta}^{\mathrm{T}}\mathbf{a}_1y(k) + \boldsymbol{\beta}^{\mathrm{T}}\mathbf{a}_2u(k) + \boldsymbol{\beta}^{\mathrm{T}}\mathbf{r}. \tag{13}$$

In general, fuzzy models can have multiple inputs and outputs (also known as multivariable models).

In the case where the system has several outputs, the functions of inputs f_j can be regarded as vector functions. However, when modelling and identifying we can concern ourselves only with single-output fuzzy models and accordingly presume f_j 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.

On the other hand, if the system has several inputs ($m > 1$), the regression vector is simply extended so as to include all the relevant 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 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 where 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 adequate to employ several (simpler) first-order models running in parallel in place of a single nth-order model for MPC purposes. If such first-order models are not feasible, it is still suitable to employ several lower-than-nth-order models instead. To put it another way, it is generally reasonable to make use of all the available data measured in a single time-step. However, due to unmeasurable system states it is sometimes not possible to undertake such approach.

Please cite this article in press as: G. Karer, et al., Model predictive control of nonlinear hybrid systems with discrete inputs employing a hybrid fuzzy model, Nonlinear Analysis: Hybrid Systems (2007), doi:10.1016/j.nahs.2006.06.005

3. Identification of the hybrid fuzzy model

The affine Takagi–Sugeno fuzzy system with a common consequence structure (described in Section 2.2) can be expressed as a global linear model. The input-dependent parameters given in Eq. (14) can be derived from Eq. (11).

$$\begin{aligned}\tilde{\mathbf{a}}_i(x_1,\ldots,x_N) &= \boldsymbol{\beta}^{\mathrm{T}}(x_1,\ldots,x_N)\mathbf{a}_1\\ \tilde{\mathbf{r}}(x_1,\ldots,x_N) &= \boldsymbol{\beta}^{\mathrm{T}}(x_1,\ldots,x_N)\mathbf{r}.\end{aligned} \tag{14}$$

In this case the fuzzy model output (10) can be described as in the following equation.

$$y = \sum_{i=1}^{N} \tilde{\mathbf{a}}_i(x_1,\ldots,x_N)x_i + \tilde{\mathbf{r}}(x_1,\ldots,x_N). \tag{15}$$

Since the inputs of the fuzzy system $(x_1,\ldots,x_N)$ are generally speaking time dependent, the procedure can be regarded as instantaneous linearization of the fuzzy model [1,2].

According to Eq. (9), it is possible to transform Eq. (10) into

$$\boldsymbol{\beta}^{\mathrm{T}}(k)\boldsymbol{I}y = \boldsymbol{\beta}^{\mathrm{T}}(k)\mathbf{a}_1x_1 + \cdots + \boldsymbol{\beta}^{\mathrm{T}}(k)\mathbf{a}_Nx_N + \boldsymbol{\beta}^{\mathrm{T}}(k)\mathbf{r} \tag{16}$$

which can be written in the following form.

$$\sum_{j=1}^{K}\beta_j(k)y = \sum_{j=1}^{K}(\beta_j(k)\mathbf{a}_{1j}x_1 + \cdots + \beta_j(k)\mathbf{a}_{Nj}x_N + \beta_j(k)r_j). \tag{17}$$

Eq. (17) can be separated into K equations (18).

$$\begin{aligned}&\mathbf{R}^1\colon \beta_1(k)y = \beta_1(k)a_{11}x_1 + \cdots + \beta_1(k)a_{N1}x_N + \beta_1(k)r_1\\ &\vdots\\ &\mathbf{R}^K\colon \beta_K(k)y = \beta_K(k)a_{1K}x_1 + \cdots + \beta_K(k)a_{NK}x_N + \beta_K(k)r_K.\end{aligned} \tag{18}$$

To identify a fuzzy system means to obtain the fuzzy model parameters $\mathbf{a}_1,\ldots,\mathbf{a}_N$ and $\mathbf{r}$. For each rule $\mathbf{R}^j;\ j = 1,\ldots,K$, the following form of regressor is used. The regressor is generally time dependent, so it can be formulated as in Eq. (19) for every time-step k.

$$\psi_j^{\mathrm{T}}(k) = [\beta_j(k)x_1(k)\ldots\beta_j(k)x_N(k)\ \beta_j(k)1]. \tag{19}$$

The regression matrix $\boldsymbol{\Psi}_j$ for the rule $\mathbf{R}^j$ in Eq. (20) is obtained by using the whole group of input data of the fuzzy system. According to Eq. (19), k runs from 1 to P, where P is the number of data pairs. However, only data pairs from time-steps k that comply with the condition in Eq. (21) are actually used for constructing the regression matrix $\boldsymbol{\Psi}_j$. Here δ denotes a small positive number. Since the model parameters are obtained by matrix inversion as described later, compliance with Eq. (21) is necessary for obtaining suitably conditioned matrices.

$$\boldsymbol{\Psi}_j = \begin{bmatrix}\psi_j^{\mathrm{T}}(1)\\ \vdots\\ \psi_j^{\mathrm{T}}(P)\end{bmatrix} \tag{20}$$

$$\beta_j(k) \geq \delta. \tag{21}$$

The output variable of the fuzzy system y, which corresponds to the rule $\mathbf{R}^j$, is written in Eq. (22).

$$y^j(k) = \beta^j(k)y(k). \tag{22}$$

It is included in the output data vector, which corresponds to the rule $\mathbf{R}^j$ as written in Eq. (23). Again, only data pairs from time-steps k that comply with the condition in Eq. (21) are actually used for constructing the regression

Please cite this article in press as: G. Karer, et al., Model predictive control of nonlinear hybrid systems with discrete inputs employing a hybrid fuzzy model, Nonlinear Analysis: Hybrid Systems (2007), doi:10.1016/j.nahs.2006.06.005

matrix $\mathbf{\Psi}_j$.

$$\boldsymbol{Y}^j = \begin{bmatrix} \beta^j y(1) \\ \vdots \\ \beta_j y(P) \end{bmatrix}. \tag{23}$$

The output of the fuzzy model $y^j(k)$ in time-step (k) corresponding to the rule $\mathbf{R}^j$ is written in Eq. (24).

$$y^j(k) = \psi_j^{\mathrm{T}}(k)\mathbf{\Theta}_j. \tag{24}$$

Here vector $\mathbf{\Theta}_j$ contains the parameters of the fuzzy model for the rule $\mathbf{R}^j$.

$$\mathbf{R}^j\colon \mathbf{\Theta}_j^{\mathrm{T}} = [a_{1j} \ldots a_{Nj}\ r_j]. \tag{25}$$

According to Eqs. (20), (23) and (24), the fuzzy model parameters for the rule $\mathbf{R}^j$ can be obtained using the least squares method as written in Eq. (26).

$$\mathbf{R}^j\colon \mathbf{\Theta}_j = (\mathbf{\Psi}_j^{\mathrm{T}}\mathbf{\Psi}_j)^{-1}\mathbf{\Psi}_j^{\mathrm{T}}\boldsymbol{Y}_j. \tag{26}$$

By calculating the fuzzy model parameters for the whole group of rules $\mathbf{R}^j;\ j = 1, \ldots, K$, the fuzzy model parameters described in Eq. (18) can be obtained.

The parameters of the fuzzy model are estimated on the basis of measured input–output data using the least squares optimization method. The approach is based on decomposition of the data matrix $\mathbf{\Psi}$ into K submatrices $\mathbf{\Psi}_1, \ldots, \mathbf{\Psi}_K$. Hence, the parameters for each rule $\mathbf{R}^j;\ j = 1, \ldots, K$, are calculated separately. Due to better conditioning of the submatrices $\mathbf{\Psi}_1, \ldots, \mathbf{\Psi}_K$ comparing to the conditioning of the whole data matrix $\mathbf{\Psi}$, this approach leads to a better estimate of the fuzzy parameters, or to put it in another way, the variances of the estimated parameters are smaller compared to the classical approach given in literature [1,2,22,23].

The described instantaneous linearization generates the parameters of the global linear model (see Eq. (14)), which depends on the antecedents of the fuzzy system $x_1, \ldots, x_N$. 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 on-line.

4. Model predictive control of systems with discrete inputs based on reachability analysis

Model predictive control is an approach where a model of the system is used to predict the future evolution of the system [15,8]. The most appropriate input vector is established and applied every time-step. Its determination is an optimization problem that is solved within a finite horizon H, i.e. for a pre-specified number of time-steps ahead. Each time-step k a sequence of optimal input vectors (27) is acquired, which minimizes the selected cost function while considering the eventual constraints of 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 etc.

$$U_k^{k+H-1} = \left\{\underline{u}(k), \underline{u}(k+1), \ldots, \underline{u}(k+H-1)\right\}. \tag{27}$$

4.1. Tree of evolution

Since the proposed control algorithm is limited to systems with discrete inputs only, the evolution of the system over time horizons for a maximum of H steps can be illustrated by a tree of evolution as shown in Fig. 1 for $H = 4$ and 3 input vectors. 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 $\underline{x_i} = \underline{x}(k|k)$, the reachable states are computed and inserted in the tree as nodes V_i, where i indexes the nodes as they were successively computed. A cost value J_i is associated with each new node and based on the cost value the most perspective node is selected. After labelling the node

Please cite this article in press as: G. Karer, et al., Model predictive control of nonlinear hybrid systems with discrete inputs employing a hybrid fuzzy model, Nonlinear Analysis: Hybrid Systems (2007), doi:10.1016/j.nahs.2006.06.005

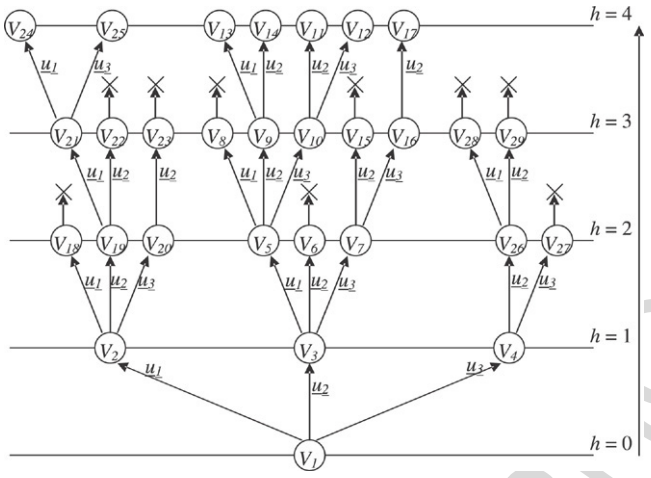

Fig. 1. Example of an explored tree of evolution.

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 ($J_i \geqslant J_{\text{opt}}$).
- The maximum step horizon has been reached ($h = H$).

If the first condition occurs, the node is labelled ~~imperspective~~ 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 ($J_{\text{opt}} = J_i$), 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 etc., until all the nodes are explored and the optimal input vector $u_{\text{opt}}(k)$ can be derived from the current optimal sequence. The optimal input vector is applied to the system $u_{\text{opt}}(k)$ and the whole procedure is repeated at the next time-step $k + 1$.

4.2. Complexity of the control problem

Complexity of the control problem with discrete inputs is primarily subject to the number of binary inputs of the system (m_b) and the maximum horizon (H) needed. Since there are no continuous inputs, there are 2^{m_b} possible combinations of input vector values. The obtainable sequences of input vectors are given in (28). Hence, the complexity of the control problem can be described as in (29). The complexity grows exponentially with the number of binary inputs of the system m_b and the maximum prediction horizon H.

$$U_k^{k+H-1} \in \{0, 1\}^{m_b \cdot H} \tag{28}$$

$$C_b = 2^{m_b \cdot H}. \tag{29}$$

Due to physical and technological constraints it is usually possible to select only a limited number (M) of combinations of input vector values, where $M < 2^{m_b}$. The obtainable sequences of input vectors are given in (30). The 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 (31).

$$U_k^{k+H-1} \in \left\{\underline{u_1}, \underline{u_2}, \ldots, \underline{u_M}\right\}^H \tag{30}$$

$$\frac{C_M}{C_b} = \left(\frac{M}{2^{m_b}}\right)^H. \tag{31}$$

Please cite this article in press as: G. Karer, et al., Model predictive control of nonlinear hybrid systems with discrete inputs employing a hybrid fuzzy model, Nonlinear Analysis: Hybrid Systems (2007), doi:10.1016/j.nahs.2006.06.005

4.3. Reachability analysis

Generally, it is not always possible to apply every input vector from the selection mentioned above. 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 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 ~~imperspective~~[1] 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.

4.4. Cost function

Cost function selection has a great influence on the behavior of the system, on the size of fully explored tree of evolution and hence on the computational complexity of the control problem. Generally, the cost function can be described as in (32).

$$J_i = J(X_k^{k+h-1}, U_k^{k+h-2}, k, h) \quad \text{for } h = 1, 2, \ldots, H;$$
$$\text{where } X_k^{k+h} = \left\{\underline{x}(k), \underline{x}(k+1), \ldots, \underline{x}(k+h)\right\}; \text{ where } \underline{x} \text{ is the vector of system states.} \tag{32}$$

As said in Section 4.1, it is desired to detect and eliminate the ~~imperspective~~ 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 prediction step h as in (33).

$$J(X_k^{k+h-1}, U_k^{k+h-2}, k, h-1) \leqslant J(X_k^{k+h}, U_k^{k+h-1}, k, h); \quad \text{for } h = 2, 3, \ldots, H. \tag{33}$$

In (34)–(36), we have proposed a universally usable form of cost function that can easily be applied to most of the systems met in practice.

$$J(X_k^{k+h}, U_k^{k+H-1}, k, h) = J(X_k^{k+h-1}, U_k^{k+H-2}, k, h-1) + f(x(k+h|k), u(k+h-1|k, k, h)) \quad \text{for } h = 1, 2, \ldots, H \tag{34}$$

$$f(x(k+h|k), u(k+h-1|k, k, h)) \geqslant 0 \quad \text{for } h = 1, 2, \ldots, H \tag{35}$$

$$J(X_k^k, \{\}, k, 0) = 0 \quad \text{for } h = 0. \tag{36}$$

The function f (35) is an arbitrary non-negative function, which estimates the quality of control. Its value is added to the sum of 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 (35), 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 the condition (33).

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. (37)): the sampling time T_S and the maximum prediction horizon H.

$$T_{\text{pred}} = H \cdot T_S. \tag{37}$$

[1] ~~Imperspective~~ states are states that are certainly not leading to the optimal solution of the control problem.

Please cite this article in press as: G. Karer, et al., Model predictive control of nonlinear hybrid systems with discrete inputs employing a hybrid fuzzy model, Nonlinear Analysis: Hybrid Systems (2007), doi:10.1016/j.nahs.2006.06.005

Since the sampling time T_S is determined by factors that in most cases cannot be changed,[2] the only way to reach a longer time of prediction T_{pred} seems to be by increasing the maximum prediction horizon H. However, as said in Section 4.2, on 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 (38). Here $H_Z = H/Z$ is the new maximum prediction horizon required to reach the desired maximum time of prediction T_{pred}.

$$T_{\text{pred}} = T_S \cdot Z \cdot H_Z. \tag{38}$$

The complexity of the control problem (provided the T_{pred} is constant) can thus be reduced by the factor shown in Eq. (39), or rather, a longer time of prediction T_{pred} can be reached, whereas the complexity increases linearly instead of exponentially.

$$\frac{C_Z}{C_M} = \frac{(M \cdot Z)^{H_Z}}{M^H} = \left(\frac{Z}{M^{Z-1}}\right)^{H_Z}. \tag{39}$$

By holding the inputs through Z time-steps several objectives can be attained.

- 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 control for a class of systems, where changing the input vector every sample time is not needed or wanted, e.g. for stiff systems. The parameter Z should be determined by means of simulation.

Due to a longer reachable maximum time of prediction T_{pred}, we expect that raising Z should cause the simulation results to improve and eventually reach the optimum. On the other hand, further raising of the parameter Z is expected to cause deterioration of the results, which happens because the inputs do not change frequently enough to ensure a satisfactory control [13].

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 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 core (temperature T) is heated or cooled through the reactor water jacket (temperature T_j). 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 temperature of the fresh input water depends on the position of the on/off valves k_H and k_C. There are two possible operating modes of the on/off valves. In the case $k_C = 1$ and $k_H = 0$, the input water is cool ($T_{\text{in}} = T_C = 12\ ^\circ\text{C}$), whereas if $k_C = 0$ and $k_H = 1$, the input water is hot ($T_{\text{in}} = T_H = 75\ ^\circ\text{C}$).

The ratio of fresh input water to reflux water is controlled by the position of the mixing valve k_M. There are five 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 system with three discrete inputs (k_M, k_H and k_C) and two measurable outputs (T and T_j).

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 core. Therefore, the batch reactor is considered a stiff system.

[2] For example, the dynamics of the system, the accuracy of prediction needed and the eventual sampling time of sensors used etc.

Please cite this article in press as: G. Karer, et al., Model predictive control of nonlinear hybrid systems with discrete inputs employing a hybrid fuzzy model, Nonlinear Analysis: Hybrid Systems (2007), doi:10.1016/j.nahs.2006.06.005

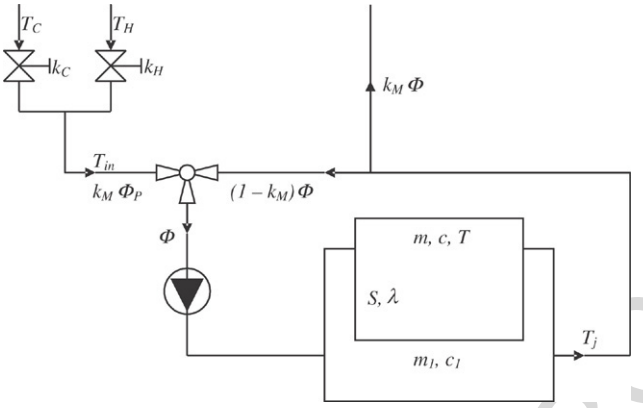

Fig. 2. Scheme of the batch reactor.

5.1. Modelling and identification of the batch reactor

A model of the batch reactor is obtained in two steps.

- The system is divided into simpler subsystems.
- Each discrete-time submodel is identified.

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

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

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

In order to carry out the identification, a suitable input–output data set for the batch reactor process has to be obtained. Therefore, we have generated a pseudo-random input signal and applied it to the batch reactor process. A sampling time of $T_S = 10$ s has been used. We have recorded the measurable outputs, i.e. the reactor core temperature T and the reactor water jacket temperature T_j. The outputs are shown in Fig. 3. A close-up is shown in Fig. 4.

5.2. Temperature in the reactor core

The temperature in the reactor core is influenced only by the heat conduction between the reactor core and the reactor water jacket, which depends on the temperature difference between the reactor water jacket T_j and the reactor core T. Therefore, a first-order submodel can be presumed as shown in Eq. (40). Here the regressor consists of the temperature in the water jacket $T_j(k)$ and the temperature in the core $T(k)$ in the current time-step.

$$\hat{T}(k+1) = F_1(T_j(k), T(k)). \tag{40}$$

Since we have surmised the heat conduction to be proportional to the temperature difference between the reactor core and the reactor water jacket, we can presume a linear model as in Eq. (41).

$$\hat{T}(k+1) = [T_j(k)\ T(k)]\boldsymbol{\Theta}_1. \tag{41}$$

After conducting a least squares optimization, we obtain the system parameters $\boldsymbol{\Theta}_1$.

$$\boldsymbol{\Theta}_1 = \begin{bmatrix} 0.0033 \\ 0.9967 \end{bmatrix}. \tag{42}$$

Please cite this article in press as: G. Karer, et al., Model predictive control of nonlinear hybrid systems with discrete inputs employing a hybrid fuzzy model, Nonlinear Analysis: Hybrid Systems (2007), doi:10.1016/j.nahs.2006.06.005

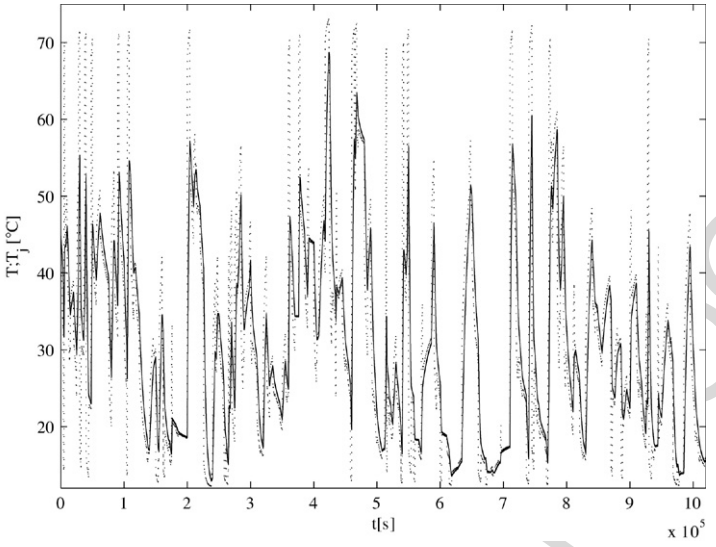

Fig. 3. Core temperature T (solid line) and water jacket temperature T_j (dotted line).

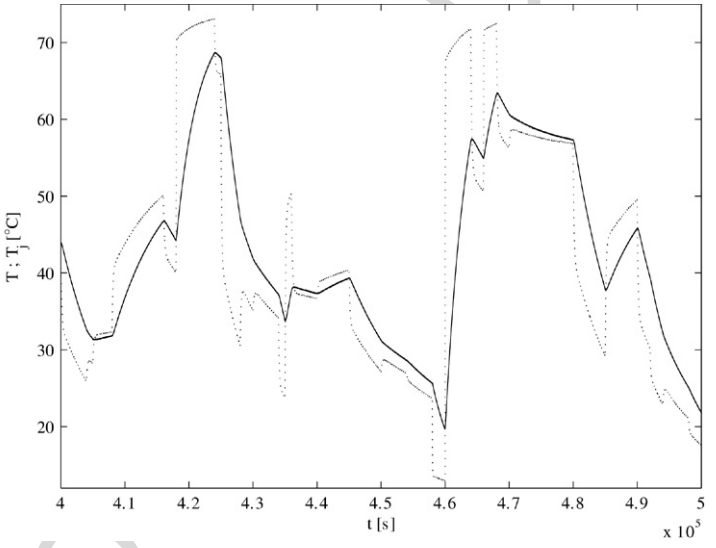

Fig. 4. Core temperature T (solid line) and water jacket temperature T_j (dotted line).

5.3. Temperature in the reactor water jacket

The temperature in the reactor water jacket T_j is influenced by all the heat-flow sources previously mentioned. Therefore, a submodel can be presumed as shown in Eq. (43). Here the regressor consists of the values in the current time-step k of the temperature in the water jacket $T_j(k)$, 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 $k_C(k)k_H(k)$ respectively.

$$\hat{T}_j(k+1) = F_2(T_j(k), T(k), k_M(k), k_C(k), k_H(k)). \tag{43}$$

As said in Section 2.1 there are two ways for merging the continuous part and the discrete part of the model. For the batch reactor, we have chosen the multimodel approach.

In this sense, it is possible to further subdivide the submodel in Eq. (43). Let us assume two operating modes.

Please cite this article in press as: G. Karer, et al., Model predictive control of nonlinear hybrid systems with discrete inputs employing a hybrid fuzzy model, Nonlinear Analysis: Hybrid Systems (2007), doi:10.1016/j.nahs.2006.06.005

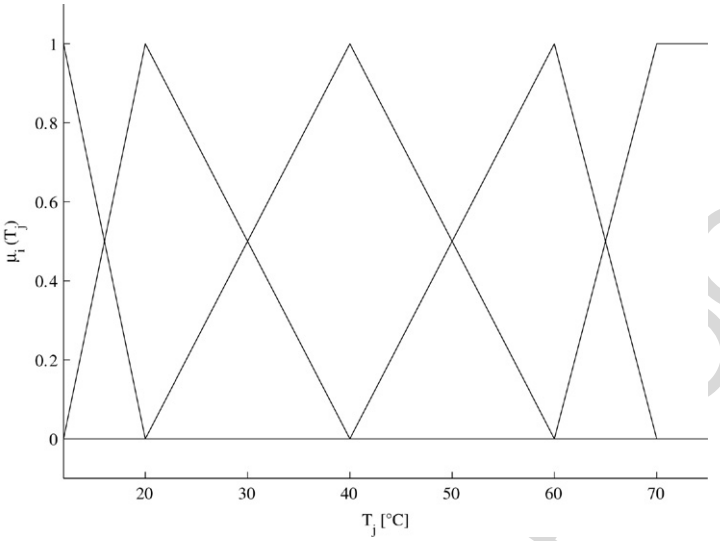

Fig. 5. Membership functions.

- The first operating mode is the case when the fresh input water is hot, i.e. $k_C(k) = 0$ and $k_H(k) = 1$. In this case the model formulation in Eq. (43) is reduced to Eq. (44).

$$\hat{T}_j(k+1) = F_H(T_j(k), T(k), k_M(k)). \tag{44}$$

- The second operating mode is the case when the fresh input water is cool, i.e. $k_C(k) = 1$ and $k_H(k) = 0$. In this case the model formulation in Eq. (43) is reduced to Eq. (45).

$$\hat{T}_j(k+1) = F_C(T_j(k), T(k), k_M(k)). \tag{45}$$

Initially, let us devote ourselves to the first operating mode. Modelling and identification of the subsystem have been carried out in a manner similar to that described in Sections 2.2 and 3.

The rule base of the fuzzy system is given in Eq. (46).

$$\mathbf{R}^i\text{: } \textbf{if } T_j(k) \text{ is } A_1^i \textbf{ then } T_j(k+1) = F_H^j(T_j(k), T(k), k_M(k)), \quad i = 1, \ldots, K. \tag{46}$$

The antecedent part is thus defined by the temperature in the water jacket $T_j(k)$. We presume that a local system corresponding to an individual rule $\mathbf{R}^i$ is affine as written in Eq. (47).

$$\mathbf{R}^i\text{: } T_j(k+1) = a_{1i} T_j(k) + a_{2i} T(k) + a_{3i} k_M(k) + r_i, \quad i = 1, \ldots, K. \tag{47}$$

Next, membership functions have to be defined. In our case, simple triangular functions are used, as shown in Fig. 5.

Such a form of the membership functions ensures that the normalized degrees of fulfillment $\beta_i(T_j)$ are equal to the membership values $\mu_i(T_j)$ over the whole operating range for each rule $\mathbf{R}^i$, respectively.

The maxima of the membership functions are determined in order to cover the whole operating range. In this case the maxima are at 12, 20, 40, 60 and 70 °C.

The output of the fuzzy model corresponding to rule $\mathbf{R}^i$ can therefore be formulated as in Eq. (48).

$$\hat{T}_j^i(k+1) = \begin{bmatrix} T_j(k) & T(k) & k_M(k) & 1 \end{bmatrix} \boldsymbol{\Theta}_2^i. \tag{48}$$

After conducting a least squares optimization for each rule $\mathbf{R}^i$, we obtain the system parameters $\boldsymbol{\Theta}_2$.

$$\boldsymbol{\Theta}_2 = \begin{bmatrix} 0.9453 & 0.9431 & 0.9429 & 0.9396 & 0.7910 \\ 0.0376 & 0.0458 & 0.0395 & 0.0339 & 0.0225 \\ 19.6748 & 16.7605 & 10.5969 & 3.9536 & 1.6856 \\ 0.3021 & 0.2160 & 0.5273 & 1.2701 & 12.0404 \end{bmatrix}. \tag{49}$$

Please cite this article in press as: G. Karer, et al., Model predictive control of nonlinear hybrid systems with discrete inputs employing a hybrid fuzzy model, Nonlinear Analysis: Hybrid Systems (2007), doi:10.1016/j.nahs.2006.06.005

A similar procedure is carried out for the second operating mode. In this case the system parameters are given in $\boldsymbol{\Theta}_3$.

$$\boldsymbol{\Theta}_3 = \begin{bmatrix} 0.9803 & 0.9740 & 0.9322 & 0.9076 & 0.8945 \\ 0.0025 & 0.0153 & 0.0466 & 0.0466 & 0.0111 \\ -0.0704 & -0.6956 & -7.8013 & -12.2555 & -18.7457 \\ 0.2707 & 0.2033 & 0.5650 & 1.9179 & 5.6129 \end{bmatrix}. \tag{50}$$

To sum up, the model of the temperature in the reactor water jacket is given in Eq. (51).

$$\hat{T}_j(k+1) = \begin{Bmatrix} [T_j(k) & T(k) & k_M(k) & 1]\boldsymbol{\Theta}_2\boldsymbol{\beta}^{\mathrm{T}} \text{ if } k_C = 0 \bigwedge k_H = 1 \\ [T_j(k) & T(k) & k_M(k) & 1]\boldsymbol{\Theta}_3\boldsymbol{\beta}^{\mathrm{T}} \text{ if } k_C = 1 \bigwedge k_H = 0 \end{Bmatrix} \tag{51}$$

6. Control

Once we have obtained the model of the process that we wish to control, we can move on to establishing the input matrix, which contains every allowed combination of input vector values (see Eq. (52)). Here each column represents an input vector. The rows of 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 $k_H \in \{1, 0\}$.

$$M = \begin{bmatrix} 0 & 0.01 & 0.02 & 0.05 & 0.1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}. \tag{52}$$

In the last step we establish a suitable cost function (see Section 4.4. According to the prepositions in Eqs. (34)–(36)), a simple cost function, which takes into consideration the square of deviation of the core temperature T from the reference temperature T_{ref}, is given in Eq. (53). In such a manner, control performance can be quantitatively estimated along the tree of evolution.

$$J(X_k^{k+h}, U_k^{k+H-1}, k, h) = J(X_k^{k+h-1}, U_k^{k+H-2}, k, h-1) + (T(k+h) - T_{\mathrm{ref}}(k+h))^2. \tag{53}$$

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

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, control was carried out by the on/off valves, whereas the mixing valve was rather 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.
- 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 control by including a sort of penalty for the movement of the valves in the cost function in Eq. (53). The modified cost function is established in Eq. (54).

$$\begin{aligned} J(X_k^{k+h}, U_k^{k+H-1}, k, h) = {} & J(X_k^{k+h-1}, U_k^{k+H-2}, k, h-1) + w_1 \cdot (T(k+h) - T_{\mathrm{ref}}(k+h))^2 \\ & + w_2 \cdot (k_C(k+h) \cdot k_H(k+h-1)) \\ & + w_3 \cdot |k_M(k+h) - k_M(k+h-1)| \cdot k_H(k+h-1). \end{aligned} \tag{54}$$

The summands in the cost function were pondered as follows.

Please cite this article in press as: G. Karer, et al., Model predictive control of nonlinear hybrid systems with discrete inputs employing a hybrid fuzzy model, Nonlinear Analysis: Hybrid Systems (2007), doi:10.1016/j.nahs.2006.06.005

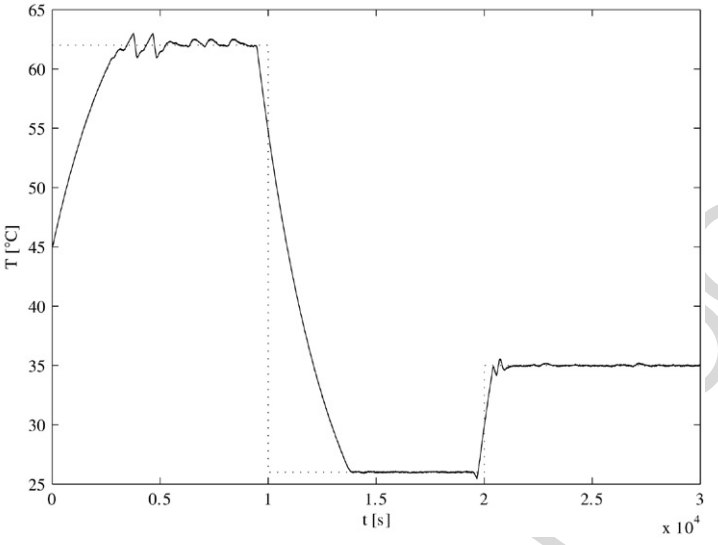

Fig. 6. Core temperature T (solid line) and reference temperature T_{ref} (dotted line).

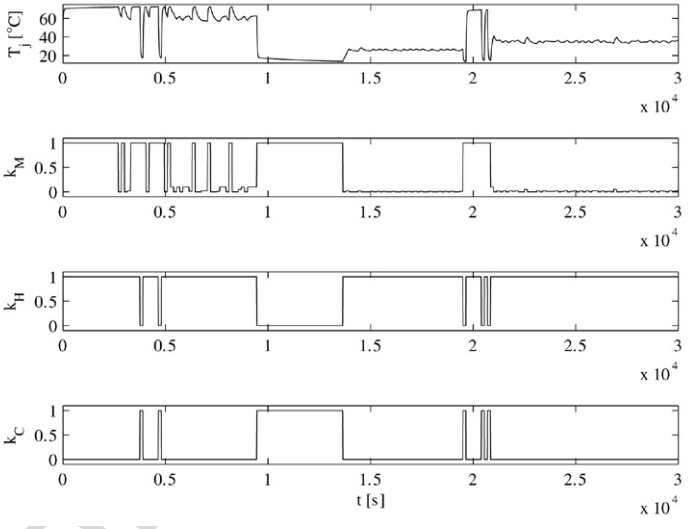

Fig. 7. Other system states.

- The square of deviation of the core temperature T from the reference trajectory T_{ref} was pondered according to the choice of the parameter Z, in order to enable the control performance comparison among simulations employing different Z. $w_1 = \frac{1}{Z}$.
- The weight w_2 ponders 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, the weight for w_3 was lowered, 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 right, all three problematic aspects of control have been satisfactorily solved.

Please cite this article in press as: G. Karer, et al., Model predictive control of nonlinear hybrid systems with discrete inputs employing a hybrid fuzzy model, Nonlinear Analysis: Hybrid Systems (2007), doi:10.1016/j.nahs.2006.06.005

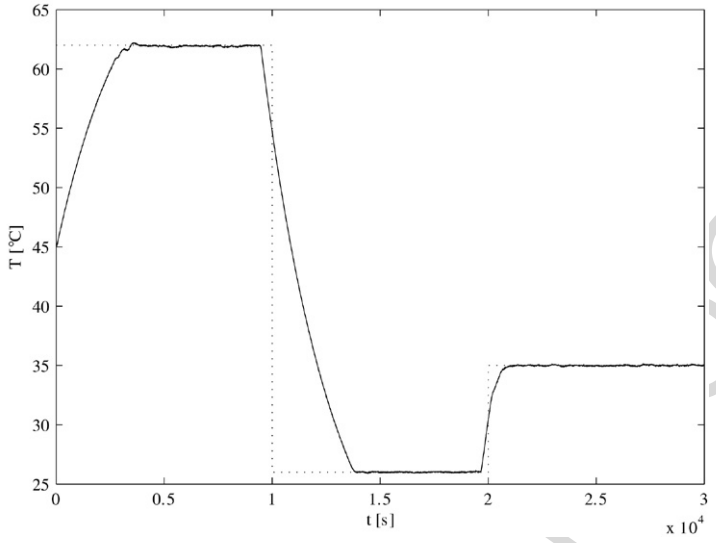

Fig. 8. Core temperature T (solid line) and reference temperature T_{ref} (dotted line).

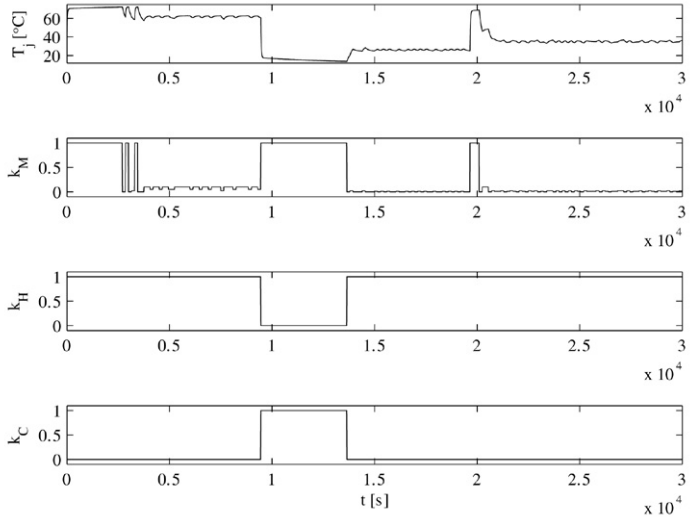

Fig. 9. Other system states.

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 water jacket (see Eq. (43)). 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 $\boldsymbol{\beta} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \end{bmatrix}$. Therefore, only one of the fuzzy regions has been taken into account when identifying the linear model. In this sense, the model parameters are given in Eqs. (55) and (56), whereas the output is obtained according to Eq. (57).

$$\boldsymbol{\Theta}_{2,\text{lin}} = \begin{bmatrix} 0.9429 \\ 0.0395 \\ 10.5969 \\ 0.5273 \end{bmatrix} \tag{55}$$

Please cite this article in press as: G. Karer, et al., Model predictive control of nonlinear hybrid systems with discrete inputs employing a hybrid fuzzy model, Nonlinear Analysis: Hybrid Systems (2007), doi:10.1016/j.nahs.2006.06.005

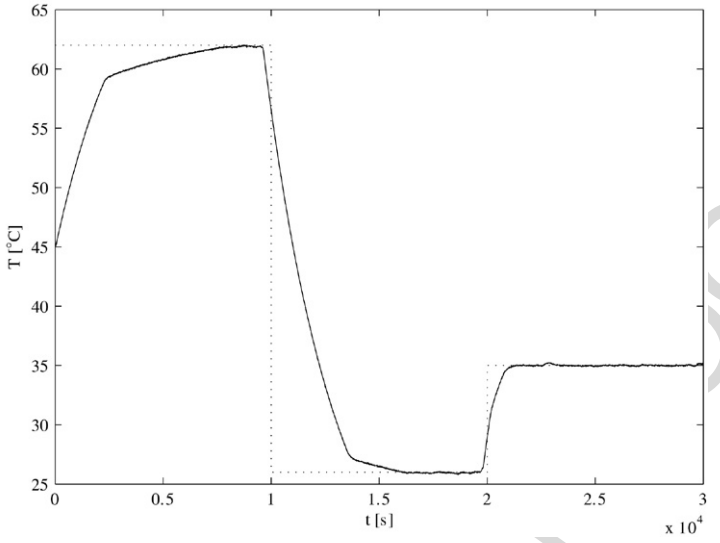

Fig. 10. Core temperature T (solid line) and reference temperature T_{ref} (dotted line).

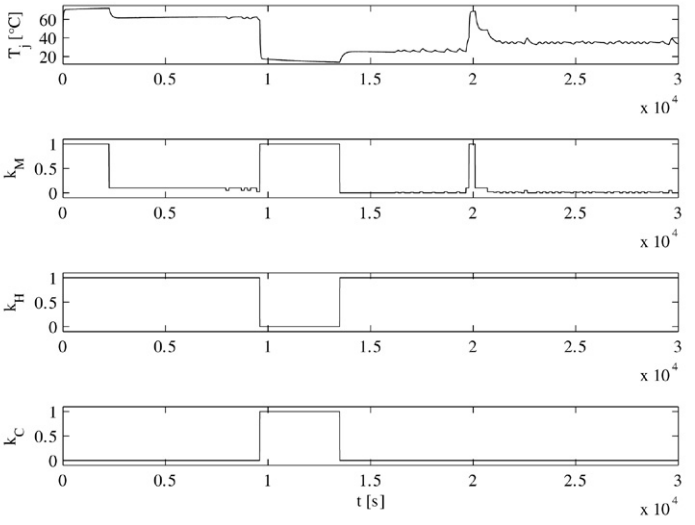

Fig. 11. Other system states.

$$\boldsymbol{\Theta}_{3,\text{lin}} = \begin{bmatrix} 0.9322 \\ 0.0466 \\ -7.8013 \\ 0.5650 \end{bmatrix} \tag{56}$$

$$\hat{T}_j(k+1) = \begin{cases} [T_j(k) \quad T(k) \quad k_M(k) \quad 1]\boldsymbol{\Theta}_{2,\text{lin}} \text{ if } k_C = 0 \bigwedge k_H = 1 \\ [T_j(k) \quad T(k) \quad k_M(k) \quad 1]\boldsymbol{\Theta}_{3,\text{lin}} \text{ if } k_C = 1 \bigwedge k_H = 0 \end{cases}. \tag{57}$$

The results of the experiment with the linear model employed in MPC are shown in Figs. 10 and 11.

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

Please cite this article in press as: G. Karer, et al., Model predictive control of nonlinear hybrid systems with discrete inputs employing a hybrid fuzzy model, Nonlinear Analysis: Hybrid Systems (2007), doi:10.1016/j.nahs.2006.06.005

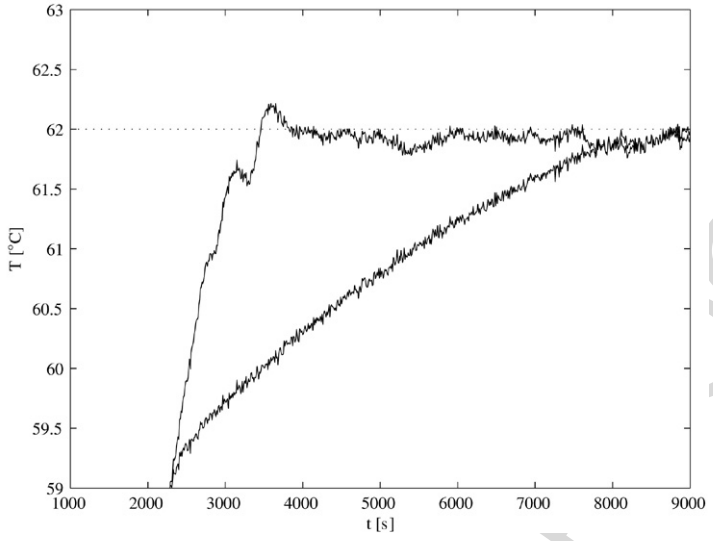

Fig. 12. Core temperature T: Comparison between hybrid fuzzy model (better) and hybrid linear model employed in MPC. $T_{\text{ref}} = 62\ ^{\circ}\text{C}$.

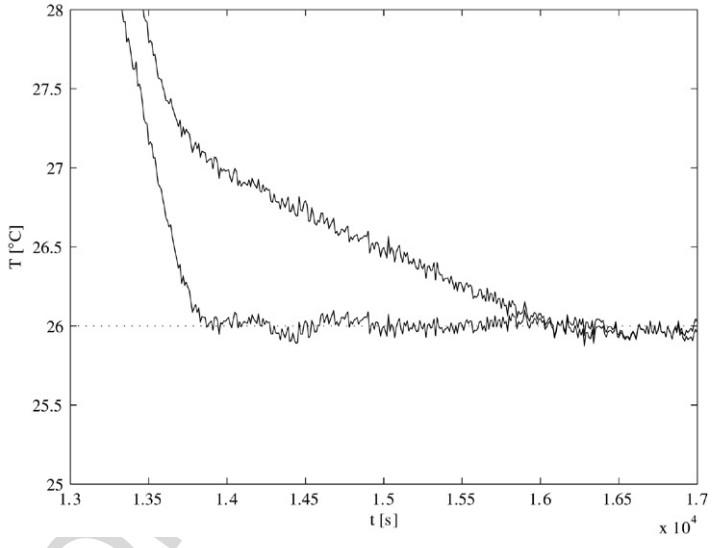

Fig. 13. Core temperature T: Comparison between hybrid fuzzy model (better) and hybrid linear model employed in MPC. $T_{\text{ref}} = 26\ ^{\circ}\text{C}$.

In the third part of the experiment, where the reference temperature is $T_{\text{ref}} = 35\ ^{\circ}\text{C}$, 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

Model predictive control of nonlinear hybrid systems with discrete inputs based on reachability analysis was dealt with.

In order to implement a MPC algorithm, a model of the process we are dealing with is needed. In the paper, a hybrid fuzzy modelling approach was introduced. First, the hybrid system hierarchy was explained and the Takagi–Sugeno fuzzy formulation for the hybrid fuzzy modelling purposes was tackled. An efficient method of identification of the hybrid fuzzy model was also discussed.

Please cite this article in press as: G. Karer, et al., Model predictive control of nonlinear hybrid systems with discrete inputs employing a hybrid fuzzy model, Nonlinear Analysis: Hybrid Systems (2007), doi:10.1016/j.nahs.2006.06.005

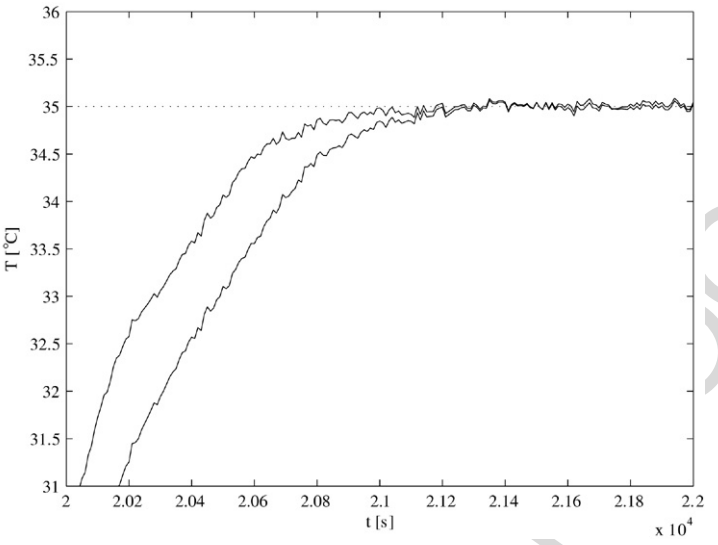

Fig. 14. Core temperature T: comparison between the hybrid fuzzy model (better) and hybrid linear model employed in MPC. $T_{\text{ref}} = 35$ °C.

A MPC algorithm suitable for systems with discrete inputs was treated. A universal cost function form that is adequate for a wide range of systems was presented. Complexity of the control method increases exponentially when raising the maximum prediction horizon H in order to reach a longer time of prediction T_{pred}. Hence, the approach for reducing the complexity by holding the inputs through a number (Z) of time-steps was proposed. In addition, the peculiarities concerning the choice of parameter Z were tackled.

The benefits of the algorithm employing a 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–nonlinear–stiff systems such as the batch reactor.

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

References

[1] R. Babuska, Fuzzy modelling and identification, Ph.D. Thesis, Delft, 1996.
[2] R. Babuska, H.B. Verbruggen, An overview of fuzzy modelling for control, Control Engineering Practice 4 (11) (1996) 1593–1606.
[3] M. Baotić, F.J. Christophersen, M. Morari, A new algorithm for constrained finite time optimal control of hybrid systems with a linear performance index, in: Proc. of the European Control Conference, Cambridge, September 2003.
[4] A. Bemporad, Efficient conversion of mixed logical dynamical systems into an equivalent piecewise affine form, IEEE Transactions on Automatic Control 49 (5) (2004) 832–838.
[5] A. Bemporad, F. Borrelli, M. Morari, On the optimal control law for linear discrete time hybrid systems, in: Hybrid Systems: Computation and Control: Lecture Notes in Computer Science, vol. 2289, 2002, pp. 105–119.
[6] A. Bemporad, M. Morari, Control of systems integrating logic, dynamics and constraints, Automatica 35 (3) (1999) 407–427.
[7] F. Borelli, M. Baotić, A. Bemporad, M. Morari, An efficient algorithm for computing the state feedback optimal control law for discrete time hybrid systems, in: Proc. of the American Control Conference, Denver, June 2003.
[8] E.F. Camacho, C. Bordons, Model Predictive Control, in: Advanced Textbooks in Control and Signal Processing, Springer-Verlag, London, 1998.
[9] J. Castro, Fuzzy logic controllers are universal approximators, IEEE Transactions on System Man Cybernetics (1995) 629–635.
[10] D. Girimonte, 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, 2004, pp. 2392–2397.
[11] F. Girosi, T. Poggio, Networks and the best approximation property, Biological Cybernetics 63 (1990) 169–176.
[12] W. Heemels, B.D. Schutter, A. Bemporad, Equivalence of hybrid dynamical models, Automatica 37 (7) (2001) 1085–1091.
[13] G. Karer, G. Mušič, B. Zupančič, 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, Limassol, June 2005, pp. 855–860.
[14] E.C. Kerrigan, 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.

Please cite this article in press as: G. Karer, et al., Model predictive control of nonlinear hybrid systems with discrete inputs employing a hybrid fuzzy model, Nonlinear Analysis: Hybrid Systems (2007), doi:10.1016/j.nahs.2006.06.005

[15] J.M. Maciejowski, Predictive Control: With Constraints, Prentice Hall, Harlow, 2002.
[16] D.Q. Mayne, S. Raković, Model predictive control of constrained piecewise affine discrete-time systems, International Journal of Robust and Nonlinear Control 13 (3) (2003) 261–279.
[17] R. Palm, D. Driankov, Fuzzy switched hybrid systems — modelling and identification, in: Proc. of the 1998 IEEE/ISCI/CIRA/SAS Joint Conf., Gaithersburg MD, September 1998, pp. 130–135.
[18] B. Potočnik, G. Mušič, B. Zupančič, Model predictive control of discrete time hybrid systems with discrete inputs, ISA Transactions 44 (2) (2005) 199–211.
[19] J. Sjoberg, A. Zhang, L. Ljung, A. Benveniste, B. Delyon, P.-Y. Glorennec, H. Hjalmarson, A. Juditsky, Nonlinear black-box modelling in system identification: A unified overview, Automatica 31 (12) (1995) 1691–1724.
[20] I. Škrjanc, D. Matko, Fuzzy predictive functional control in the state space domain, Journal of Intelligent and Robotic Systems 31 (2001) 283–297.
[21] E. Sontag, Nonlinear regulation: The piecewise linear approach, IEEE Transactions on Automatic Control 26 (2) (1981) 346–358.
[22] M. Sugeno, K. Tanaka, Successive identification of a fuzzy model and its application to prediction of a complex system, Fuzzy Sets and Systems 42 (1991) 315–334.
[23] T. Takagi, M. Sugeno, Fuzzy identification of systems and its application to modelling and control, IEEE Transactions on System, Man and Cybernetics 15 (1985) 116–132.
[24] A. van der Schaft, H. Schumacher, An introduction to hybrid dynamical systems, in: Lecture Notes in Control and Information Sciences, vol. 251, 1999, pp. v–vii.
[25] H.S. Witsenhausen, A class of hybrid-state continuous time dynamic systems, IEEE Transactions on Automatic Control 11 (2) (1966) 161–167.

Please cite this article in press as: G. Karer, et al., Model predictive control of nonlinear hybrid systems with discrete inputs employing a hybrid fuzzy model, Nonlinear Analysis: Hybrid Systems (2007), doi:10.1016/j.nahs.2006.06.005