Extraction of Local Linear Models from Takagi-Sugeno Fuzzy Model with Application to Model-based Predictive Control

S. Mollov, P. van der Veen, R. Babuška, J. Abonyi*,
J. A. Roubos, H.B. Verbruggen

Delft University of Technology, Faculty of Information Technology and Systems
Control Laboratory, Mekelweg 4, P.O. Box 5031, 2600 GA Delft, The Netherlands
Phone: +31 15 2783371, Fax: +31 15 2786679, email: s.mollov@its.tudelft.nl
* On leave from: Department of Chemical Engineering Cybernetics
University of Veszprém, Hungary, P.O.Box 158. H-8201

Abstract: MBPC is a nice technique to control multivariable systems while dealing with constraints and certain objective. Linear MBPC (LMBPC) is currently a settled theory and can be applied straightforward for linear processes. In this paper we deal with nonlinear systems, for which linear models that can be extracted. This way a time varying linear representaion is obtained which is used in LMBPC. Different schemes to obtain such local linear models are assessed in the light of the achieved performance of the predictive controller. Takagi-Sugeno (TS) fuzzy models are chosen, because the model structure as local linear models can be derived from the linear rule consequences in a direct way.
Keywords: TS fuzzy models, Model-based predictive control, nonlinear systems, multivariable (MIMO) systems.

1 Introduction

Model-based predictive control (MBPC) is a powerful method to control multivariable systems. It has become a major research topic during the last few decades and, unlike many other advanced techniques, it has also been successfully applied in industry [10]. The main reason for this success is the ability of MBPC to control multivariable systems under various constraints in an optimal way.

A problem for the real-time control of nonlinear systems is that a nonlinear (and usually non-convex) optimization problem must be solved at each sampling period. This hampers the application for fast systems where iterative optimization techniques cannot be properly used, due to short sampling times.

To avoid non-convex optimization, a single local linear model or a set of local linear models is used, instead of a single nonlinear plant model. The nonlinear plant model is of fuzzy TS type. This type models (TS) are not only universal approximators but also can accommodate different pieces of knowledge (e.g., measurements, information based on first principles, qualitative knowledge) which must be integrated, in order to obtain a good process model [14, 4]. If little prior knowledge is available, fuzzy models can be extracted from the process measurements, using various techniques such as fuzzy clustering, neuro-fuzzy learning, orthogonal least squares, etc. Moreover, the local linear models can be derived from the linear rule consequences in a straightforward way.

At a certain point of the predicted trajectory a local linear model is computed by freezing the parameters of the fuzzy model. The resulting optimization problem has an analytic solution when no constraints are present; the constrained convex optimization problem can effectively be solved by quadratic programming (QP), which is modified such that to accommodate more than one model.

2 Fuzzy Model Structure

Takagi-Sugeno (TS) fuzzy models are suitable to model a large class of nonlinear systems [2, 13, 3]. Consider a MIMO system with n_i inputs: $\mathbf{u} \in U \subset R^{n_i}$, and n_o outputs: $\mathbf{y} \in Y \subset R^{n_o}$. This system will be approximated by a collection of coupled MISO discrete-time fuzzy models. Denote q^{-1} the backward shift operator: $q^{-1}y(k) \stackrel{\text{def}}{=} y(k-1)$, where y is a signal sampled at discrete time instants k. Given two integers, $i \leq j$, define an ordered sequence of delayed samples of the signal y as:

$$\{y(k)\}_i^j \stackrel{\text{def}}{=} [y(k-i),\ y(k-i-1),\ \ldots,\ y(k-j)] .$$

The MISO models are of the input–output NARX type:

$$y_l(k+1) = \mathcal{F}_l\left(\xi_l(k)\right), \quad l = 1, 2, \ldots, n_o, \tag{1}$$

where the regression vector $\xi_l(k)$ is given by:

$$\begin{aligned} \xi_l(k) \ = \ & \left[\{y_1(k)\}_0^{n_{yl1}}, \ldots, \{y_{n_o}(k)\}_0^{n_{yln_o}}, \right. \\ & \left. \{u_1(k+1)\}_{n_{dl1}}^{n_{ul1}}, \ldots, \{u_{n_i}(k+1)\}_{n_{dln_i}}^{n_{uln_i}}\right] . \end{aligned}$$

Here n_y and n_u are the number of delayed outputs and inputs, respectively, and n_d is the number of pure (transport) delays from the input to the output. n_y is an $n_o \times n_o$ matrix, and n_u, n_d are $n_o \times n_i$ matrices. $\mathcal{F}_l$ are rule-based fuzzy models of the Takagi–Sugeno (TS) type [13]:

$$\begin{aligned} R_{li}: \quad & \textbf{If } \xi_{l1}(k) \text{ is } \Omega_{li1} \textbf{ and } \ldots \textbf{ and } \xi_{lp}(k) \text{ is } \Omega_{lip} \\ & \textbf{then } y_{li}(k+1) = \boldsymbol{\zeta}_{li}\mathbf{y}(k) + \boldsymbol{\eta}_{li}\mathbf{u}(k) + \theta_{li} \\ & i = 1, 2, \ldots, K_l . \end{aligned} \tag{2}$$

Here Ω_{li} are the antecedent fuzzy sets of the ith rule, $\boldsymbol{\zeta}$ and $\boldsymbol{\eta}$ are vectors of polynomials in q^{-1}, e.g., $\boldsymbol{\zeta} = \zeta_0 + \zeta_1 q^{-1} + \zeta_2 q^{-2} + \ldots$, and θ is the offset. K_l is the number of rules in the lth model. The MIMO TS rules are estimated from input-output system data [14, 3]. Then the model output is computed as the degree of fulfillment (DOF) for each antecedent variable is calculated and the resulting for every rule DOFs are combined with the linear consequences.

3 Local Linear Models

Extraction of local linear models in a state space form a TS fuzzy model is investigated in order to extend the operation range of linear model based predictive controllers. The point where the local linear model is obtained by freezing the parameters of the fuzzy model is different in described methods. Here three methods are presented, namely a single-model one and two multi-models and exploited in combination with linear model-based predictive controllers (Appendix A).

3.1 Single-model method

In the single-model method (SM) [11], at each sample time k, given the operating point $\mathbf{u}(k-1)$ and $\mathbf{y}(k)$ ($\mathbf{u}(k)$ is not available), the local state-space model is calculated as follows:

Calculate the degrees of fulfillment $\omega_i\left(\mathbf{x}(k)\right)$ of the antecedents, using product as the fuzzy logic **and** operator. The rule inference gives:

$$\mathbf{y}_l(k+1) = \frac{\sum_{i=1}^{K_l} \omega_{li}(\mathbf{x}_l(k)) \cdot y_{li}(k+1)}{\sum_{i=1}^{K_l} \omega_{li}(\mathbf{x}_l(k))} , \tag{3}$$

$$\mathbf{y}_{li}(k+1) = (\boldsymbol{\zeta}_{li}\mathbf{y}(k) + \boldsymbol{\eta}_{li}\mathbf{u}(k) + \theta_{li}) . \tag{4}$$

Define $\boldsymbol{\zeta}_l^*$, $\boldsymbol{\eta}_l^*$ and $\boldsymbol{\theta}_l^*$ as:

$$\boldsymbol{\zeta}_l^* \ = \ \frac{\sum_{i=1}^{K_l} \omega_{li}(\mathbf{x}_l(k)) \cdot \boldsymbol{\zeta}_{li}}{\sum_{i=1}^{K_l} \omega_{li}(\mathbf{x}_l(k))} ,$$

$$\boldsymbol{\eta}_l^* = \frac{\sum_{i=1}^{K_l} \omega_{li}(\mathbf{x}_l(k)) \cdot \boldsymbol{\eta}_{li}}{\sum_{i=1}^{K_l} \omega_{li}(\mathbf{x}_l(k))},$$
$$\boldsymbol{\theta}_l^* = \frac{\sum_{i=1}^{K_l} \omega_{li}(\mathbf{x}_l(k)) \cdot \boldsymbol{\theta}_{li}}{\sum_{i=1}^{K_l} \omega_{li}(\mathbf{x}_l(k))}.$$

Define $\mathbf{x}, \mathbf{u}$ and $\mathbf{y}$ for state-space description as:

$$\mathbf{x}(k) = [\mathrm{x}_1(k), \ldots, \mathrm{x}_1(k-n_{y1}), \ldots, \mathrm{x}_{n_o}(k), \ldots, \mathrm{x}_{n_o}(k-n_{yn_o}), \mathrm{u}_1(k-1), \ldots, \mathbf{u}_{n_i}(k-n_d-n_u)]^T,$$
$$\mathbf{u}(k) = [\mathrm{u}_1(k), \mathrm{u}_2(k), \ldots, \mathrm{u}_{n_i}(k]^T,$$
$$\mathbf{y}(k) = [\mathrm{x}_1(k), \mathrm{x}_2(k), \ldots, \mathrm{x}_{n_o}(k)]^T.$$

In order to use QP for systems which depend not only on the current input, but also on the previous ones, it is necessary to construct a state-space representation, such that the state vector $\mathbf{x}(k)$ to accommodate not only the state variables, appearing in $\mathbf{y}(k)$, but also the previous inputs and the offset as last element. This results in a system with only current inputs, but leads to more complex $\mathbf{A}^*$-matrix. The latter contains also η^*s, corresponding to the previous inputs. If the maximal delay in the input i, $i = 1, \ldots, n_i$ is $u_{i,dmax}$, then the number of the additional columns is $\sum_{i=1}^{n_i} \max(u_{i,dmax} - 1, 0)$. In the last column of $\mathbf{A}^*$ are stored the offsets θ^*s. The columns with η^*s correspond to the previous inputs, stored in the state vector; these columns are not included in $\mathbf{B}^*$. The ones in $\mathbf{A}^*$ correspond to the delayed values of a certain variable. The local linear system matrices are derived as follows: $\mathbf{A}^*$ is a $\alpha \times \alpha$, where $\alpha = \alpha_1 + \sum_{i=1}^{n_i} \max(u_{i,dmax} - 1, 0) + 1$, $\alpha_1 = \sum_{j=1}^{n_o} \max(n_{yj}, 1)$, $\mathbf{B}^*$ is an $\alpha \times n_i$ and $\mathbf{C}$ is a $n_o \times \alpha$ matrix:

$$\mathbf{A}^* = \begin{bmatrix}
\zeta_{1,1}^* & \zeta_{1,2}^* & \cdots & \cdots & \cdots & \cdots & \zeta_{1,\alpha_1}^* & \eta_{1,i}^* & \cdots & \eta_{1,j}^* & \theta_{1,1}^* \\
1 & 0 & 0 & & \cdots & & 0 & 0 & \cdots & 0 & 0 \\
0 & 1 & \vdots & & \ddots & & 0 & 0 & \cdots & 0 & 0 \\
0 & \cdots & \ddots & & \ddots & & \vdots & \vdots & \ddots & \vdots & \vdots \\
\zeta_{2,1}^* & \zeta_{2,2}^* & \cdots & \cdots & \cdots & \cdots & \zeta_{2,\alpha_1}^* & \eta_{2,i}^* & \cdots & \eta_{2,j}^* & \theta_{2,1}^* \\
0 & \vdots & \ddots & & \ddots & & \vdots & \vdots & \ddots & \vdots & \vdots \\
\zeta_{n_o,1}^* & \zeta_{n_o,2}^* & \cdots & \cdots & \cdots & \cdots & \zeta_{n_o,\alpha_1}^* & \eta_{n_o,i}^* & \cdots & \eta_{n_o,j}^* & \theta_{n_o,\alpha_1}^* \\
0 & \cdots & 0 & 1 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 \\
\vdots & & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & \cdots & 0 & 0 & \cdots & 1 & 0 & 0 & \cdots & 0 & 0 \\
0 & \cdots & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 1
\end{bmatrix}, \tag{5}$$

$$\mathbf{B}^* = \begin{bmatrix}
\eta_{1,1}^* & \eta_{1,2}^* & \cdots & \eta_{1,n_i}^* \\
0 & \cdots & \cdots & 0 \\
\vdots & & & \vdots \\
0 & \cdots & \cdots & 0 \\
\eta_{2,1}^* & \eta_{2,2}^* & \cdots & \eta_{2,n_i}^* \\
\vdots & \ddots & \ddots & \vdots \\
\eta_{n_o,1}^* & \eta_{n_o,2}^* & \cdots & \eta_{n_o,n_i}^* \\
\vdots & \ddots & \ddots & \vdots
\end{bmatrix}, \tag{6}$$

$$\mathbf{C} = \begin{bmatrix}
1 & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\
\vdots & & \ddots & & \ddots & & \vdots \\
0 & \cdots & \cdots & 1 & 0 & \cdots & 0
\end{bmatrix}. \tag{7}$$

The ones in $\mathbf{C}$ are positioned such that $\mathrm{y}_l(k) = \mathrm{x}_l(k)$.

The obtained linear state-space model is utilized in the MBPC scheme. Initially, at moment k the control signal $\mathbf{u}(k-1)$ is used; $\mathbf{u}(k)$ is not known. However, after the optimization, $\mathbf{u}(k)$ is available. It could be used in the next iterations, which is necessary to achieve convergence.

3.2 Multi-models method 1

In the single-model method, the local linear model $M(k) = \{\mathbf{A}_k^*, \mathbf{B}_k^*, \mathbf{C}_k\}$, obtained at the current time instant k, is used for the entire prediction horizon [11]. The optimal control signal is calculated by means of quadratic programming. For many-steps-ahead control, the influence of the approximation errors may significantly deteriorate the performance. This can be remedied by computing the model along the simulated with the fuzzy model trajectory, i.e., a sequence of models $M(i)$ is obtained over $i = k, \dots, k + H_p$. There are two ways to obtain such a sequence. In [8, 12] it has been proposed the following scheme (MM-M) to obtain a set of local linear models:

1. Use the already obtained linear model $M(k)$ and compute the control signal $\mathbf{u}(k)$ over the entire prediction horizon.
2. Take $\mathbf{u}(k)$ and compute $\mathbf{y}_m(k+1)$.
3. Freeze the parameters of the TS model locally around $\big(\mathbf{y}_m(k+1), \mathbf{u}(k)\big)$ and obtain $M(k+1)$.
4. Use $M(k)$ and $M(k+1)$ to compute the new control sequence $\mathbf{u}$ for the whole prediction horizon.
5. Take $\mathbf{u}(k)$ and $\mathbf{u}(k+1)$ and compute $\mathbf{y}_m(k+2)$.
6. Freeze around $\big(\mathbf{y}_m(k+2), \mathbf{u}(k+1)\big)$ and obtain $M(k+2)$.
7. Use $M(k)$, $M(k+1)$ and $M(k+2)$ compute the new control sequence $\mathbf{u}$ for the whole prediction horizon.

Steps 5 through 7 are repeated for $i = k+1, \dots, k+H_p$. Then, based on $M(k)$, $M(k+1)$, ..., $M(k+H_p)$, the final control $\mathbf{u}$ is computed. At step 1, when only a model $M(k)$ is available, $\mathbf{u}$ is obtained as in the single-model case. Hereafter a set of linear models $\{M(k+i)\}_{i=1}^{H_p}$ is used (Appendix A). For $i < H_p$, $M(k+i+j) = M(k+i)$, $j = 1, \dots, H_p - i$.

3.3 Multi-models method 2

The above method is apparently slow due to the iterative procedure for computing the control signal – the fuzzy model is simulated H_p times and for each simulation only one new local linear model is obtained. Abonyi [1] proposed a similar scheme (MM-A), where the fuzzy model is simulated for the entire horizon and all local models are obtained:

1. Use the already obtained linear model $M(k)$ and compute the control signal $\mathbf{u}(k)$ for the whole prediction horizon.
2. Simulate the fuzzy model over the prediction horizon.
3. Freeze the TS model along each point in the predicted trajectory $\big(\mathbf{y}_m(k+i), \mathbf{u}(k+i)\big)$ and obtain $M(k+i)$ for $i = 1, \dots, H_p$.
4. Use $M(k+i)$, $i = 1 \dots H_p$ and compute the new control sequence $\mathbf{u}$ for the whole prediction horizon.

Steps 3 and 4 are repeated until $\mathbf{u}$ converges. In order to provide convergence it is necessary the input and the output to be taken at the same moment, i.e., the method needs initial value for the control signal. It reflects the convergence rate, which turns out to be connected to the computational load. SM method can be used to provide such an initial value.

4 Application to four cascaded tanks system

4.1 System description

Figure 1 presents the laboratory setup of a four cascaded tanks system. The fuzzy model is obtained by means of fuzzy modeling [2] using real data from the system. The structure of the model is built using insight in the physical structure of the system. Vessels 1 and 2 cannot influence vessels 3 and 4 because the water cannot flow back. Also vessel 1 cannot influence vessel 2, and vessel 3 cannot influence vessel 4. There is one sample time delay in the inputs.

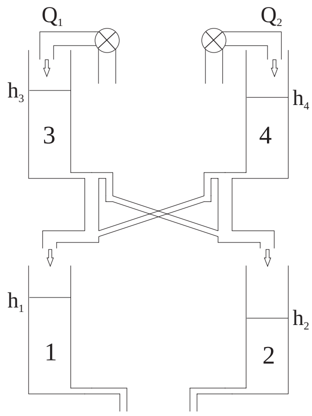

Figure 1: Four cascaded tanks system

The objective is to keep the levels in the two lower tanks as close as possible to some prescribed trajectory with minimum energy consumption. The methods performance is evaluated by sum square mean error criteria and the control actions are assessed by their energy.

4.2 Simulation results

Model-based predictive control with the models obtained by the different methods is applied to control the process. A prediction horizon of 20 and a control horizon of 1 are used and the results are presented in Fig. 2. To avoid coupling, the reference is smoothed by a first-order Butterworth filter. Because of the small quantities, the criteria values are shown in Tab. 1, related to each other as the single-model values for the different criteria are taken for 100%. Both controllers based on sets of local linear models show better performance than the controller using only single local model. The achieved improvement is most significant for MM-A, where the sum squared mean error is about 15% lower than the same value for SM. Also the MM-A method is faster than MM-M. The reason is that in MM-A the entire control sequence is available right after the first iteration and the next iterations can only improve it, while in MM-M the entire control sequence is obtained after H_p iterations. The extra computational load imposed in MM-A in order to obtain the $H_p - 1$ more local linear models than in MM-M is neglectful. Thus in MM-A after a couple of iterations an acceptable control signal is obtained. Acceptable means for stable models the control is expected to converge to a value that provides satisfactory system performance.

Table 1: Performance criteria

Method	SM	MM-M	MM-A
Output_1	0.02149 (100%)	97.3%	90.3%
Output_2	0.01826 (100%)	83.6%	76.3%
Input_1	0.3674 (100%)	99%	99%
Input_2	0.2198 (100%)	100%	98%

5 Conclusions

A local linear state space model can be extracted from nonlinear TS fuzzy model at the current working point. Model based predictive control using such local linear models shows good results for the considered process. It can cope with delayed inputs as well, which is accomplished by storing the previous inputs in the state vector. When long prediction horizon is necessary, the controllers based on a set of local linear models perform better than the one with a single linear model. MM-A is considered to be the best of the three not only because it gives slightly better results than MM-M but also due to its computational complexity - lower than the MM-M complexity.

The proposed controller with a local linear model or set of models results in a convex optimization problem,

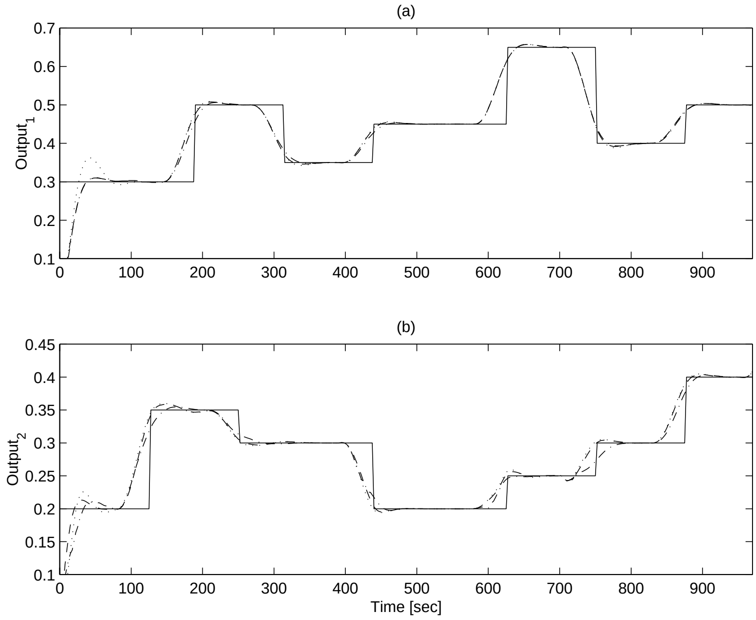

Figure 2: System performance: (a) Output$_1$, (b) Output$_2$. Solid line represents the reference, dotted line – SM, dashed – MM-M, dashdot – MM-A.

which is a main advantage in comparison with the use of relational models [7, 9, 5], where nonlinear optimization methods are necessary.

Future research is aimed toward ensuring stability of the controller/model and establishing necessary and sufficient conditions for the convergence of control signal.

Acknowledgement. This research is supported by the FAMIMO Esprit project LTR 219 11. More information is available at http://iridia.ulb.ac.be/FAMIMO.

Appendix

A Model Based Predictive Control

In linear MBPC [6], a linear model is used to predict the output $\hat{\mathbf{y}}$ as a function of the control signal $\Delta\mathbf{u}(k, \ldots, k + H_p)$, with H_p the prediction horizon. The objective function

$$J = \sum_{i=1}^{H_p} \|(\mathbf{r}_{k+i} - \hat{\mathbf{y}}_{k+i})\|^2_{\mathbf{P}_i} + \sum_{i=1}^{Hc} \|(\Delta\mathbf{u}_{k+i-1})\|^2_{\mathbf{Q}_i} \ , \tag{8}$$

is minimized for a given reference trajectory. The signal $\Delta\mathbf{u}$ may change over the control horizon H_c ($H_c \leq H_p$) and remains constant between H_c and H_p. In order to accommodate the set linear models, the quadratic programming (QP) problem is modified as follows:

$$\min_{\Delta\mathbf{u}} \left\{ \frac{1}{2} \Delta\mathbf{u}^T \mathbf{H} \Delta\mathbf{u} + \mathbf{c}^T \Delta\mathbf{u} \right\} \ , \tag{9}$$

with

$$\begin{cases} \mathbf{H} = 2(\mathbf{R}_u^T \mathbf{P} \mathbf{R}_u + \mathbf{Q}) \\ \mathbf{c} = 2[\mathbf{R}_u^T \mathbf{P}^T (\mathbf{R}_x \mathbf{A}_k \mathbf{x}(k) - \mathbf{r})]^T \ , \end{cases} \tag{10}$$

where $\mathbf{A}_k$ is the extended matrix at time instant k

$$\mathbf{A}_k = \begin{bmatrix} A^* & B^* \\ 0 & I \end{bmatrix}$$

and the constraints on $\mathbf{u}$, $\Delta\mathbf{u}$, and $\mathbf{y}$

$$\mathbf{\Lambda}\Delta\mathbf{u} \leq \boldsymbol{\omega} \ , \tag{11}$$

$$\mathbf{\Lambda} = \begin{bmatrix} \mathbf{I}_{\Delta u} \\ -\mathbf{I}_{\Delta u} \\ \mathbf{I}^{H_p m} \\ -\mathbf{I}^{H_p m} \\ \mathbf{R}_u \\ -\mathbf{R}_u \\ \mathbf{R}_{u1} \\ \mathbf{dR}_u \\ -\mathbf{R}_{u1} \\ -\mathbf{dR}_u \end{bmatrix} \quad \boldsymbol{\omega} = \begin{bmatrix} \mathbf{u}^{max} - \mathbf{I}_u u_{k-1} \\ -\mathbf{u}^{min} - \mathbf{I}_u u_{k-1} \\ \Delta\mathbf{u}^{max} \\ -\Delta\mathbf{u}^{min} \\ \mathbf{y}^{max} - \mathbf{R}_x \mathbf{A}_k x_k \\ -\mathbf{y}^{min} - \mathbf{R}_x \mathbf{A}_k x_k \\ \mathbf{dy}^{max} - \mathbf{R}_{x1} \mathbf{A}_k x_k + y_k \\ \mathbf{dy}^{max1} - \mathbf{dR}_x \mathbf{A}_k x_k \\ -\mathbf{dy}^{min} + \mathbf{R}_{x1} \mathbf{A}_k x_k - y_k \\ -\mathbf{dy}^{min1} + \mathbf{dR}_x \mathbf{A}_k x_k \end{bmatrix}, \tag{12}$$

where $\mathbf{I}^{H_p m}$ is a $(H_p m \times H_p m)$ unity matrix. The matrices $\mathbf{I}_u, \mathbf{I}_{\Delta u}, \mathbf{R}_x, \mathbf{R}_u, \mathbf{dR}_x$ and $\mathbf{dR}_u$ are defined by:

$$\begin{bmatrix} u_k \\ \tilde{u}_{k+1} \\ \vdots \\ \tilde{u}_{k+H_c-1} \end{bmatrix} = \underbrace{\begin{bmatrix} I \\ I \\ \vdots \\ I \end{bmatrix}}_{\mathbf{I}_u} u_{k-1} + \underbrace{\begin{bmatrix} I0\ldots0 \\ II\ldots0 \\ \vdots \ddots \vdots \\ II\ldots I \end{bmatrix}}_{\mathbf{I}_{\Delta u}} \begin{bmatrix} \Delta u_k \\ \Delta\tilde{u}_{k+1} \\ \vdots \\ \Delta\tilde{u}_{k+H_c-1} \end{bmatrix}, \tag{13}$$

where u_k is the supplied control and $\tilde{u}_{k+i}$, $i = 1, \ldots, H_c - 1$ is the optimized control sequence (predicted).

$$\mathbf{R}_x = \begin{bmatrix} \mathbf{C}_k \\ \mathbf{C}_{k+1}\mathbf{A}_k \\ \vdots \\ \mathbf{C}_{k+H_p}\mathbf{A}_{k+H_p-1}\ldots\mathbf{A}_k \end{bmatrix}, \tag{14}$$

$$\mathbf{R}_u = \begin{bmatrix} \mathbf{C}_{k+1}\mathbf{B}_k & \ldots & 0 \\ \mathbf{C}_{k+2}\mathbf{A}_{k+1}\mathbf{B}_k & \ldots & 0 \\ \vdots & \ddots & \vdots \\ \mathbf{C}_{k+H_p}\mathbf{A}_{k+H_p-1}\ldots\mathbf{A}_{k+1}\mathbf{B}_k & \ldots & \mathbf{C}_{k+H_p}\mathbf{A}_{k+H_p-H_c}\ldots\mathbf{A}_{k+1}\mathbf{B}_{k+H_c} \end{bmatrix}, \tag{15}$$

$$\mathbf{dR}_x = \begin{bmatrix} \mathbf{C}_{k+1}\mathbf{A}_k - \mathbf{C}_k \\ \mathbf{C}_{k+2}\mathbf{A}_{k+1}\mathbf{A}_k - \mathbf{C}_{k+1}\mathbf{A}_k \\ \vdots \\ \mathbf{C}_{k+H_p}\mathbf{A}_{k+H_p-1}\ldots\mathbf{A}_k - \\ \mathbf{C}_{k+H_p-1}\mathbf{A}_{k+H_p-2}\ldots\mathbf{A}_k \end{bmatrix}, \tag{16}$$

$$\mathbf{dR}_u = \begin{bmatrix} \mathbf{C}_{k+2}\mathbf{A}_{k+1}\mathbf{B}_k - \mathbf{C}_{k+1}\mathbf{B}_k & \ldots & \mathbf{0} \\ \mathbf{C}_{k+3}\mathbf{A}_{k+2}\mathbf{A}_{k+1}\mathbf{B}_k - \mathbf{C}_{k+2}\mathbf{A}_{k+1}\mathbf{B}_k & \ldots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{C}_{k+H_p}\mathbf{A}_{k+H_p-1}\ldots\mathbf{A}_{k+1}\mathbf{B}_k - \mathbf{C}_{k+H_p-1}\mathbf{A}_{k+H_p-2}\ldots\mathbf{A}_{k+1}\mathbf{B}_k & \ldots & \mathbf{C}_{k+H_p}\mathbf{A}_{k+H_p-1}\ldots\mathbf{A}_{k+H_c+1}\mathbf{B}_{k+H_c} - \mathbf{C}_{k+H_p-1}\mathbf{A}_{k+H_p-2}\ldots\mathbf{A}_{k+H_c1}\mathbf{B}_{k+H_c} \end{bmatrix}. \tag{17}$$

The matrices $\mathbf{dR}_{x1}$ and $\mathbf{dR}_{u1}$ are defined as the hypothetical first rows in $\mathbf{dR}_x$ and $\mathbf{dR}_u$: $\mathbf{dR}_{x1} = \mathbf{C}_k$ and $\mathbf{dR}_{u1} = \mathbf{C}_{k+1}\mathbf{B}_k - \mathbf{C}_k$.

References

[1] J. Abonyi, L. Nagy, and F. Szeifert. Fuzzy model based predictive control by instantaneous linearization. *Fuzzy Sets and Systems*, 1999. To appear.

[2] R. Babuška. *Fuzzy Modeling for Control.* Kluwer Academic Publishers, Boston, 1998.

[3] R. Babuška, J.A. Roubos, and H.B. Verbruggen. Identification of MIMO systems by input-output TS fuzzy models. In *Proceesings FUZZ-IEEE'98*, Anchorage, Alaska, 1998.

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

[5] J. Valente de Oliveira and J.M. Lemos. Long-range predictive adaptive fuzzy relational control. *Fuzzy Sets and Systems*, 70:337–357, 1995.

[6] C.E. García, D.M. Prett, and M. Morari. Model predictive control: Theory and practice – a survey. *Automatica*, 25:335–348, 1989.

[7] D.A. Linkens and S. Kadiah. Long-range predictive control using fuzzy process models. *IChemE*, 74:77–88, 1996.

[8] S. Mollov, J.A. Roubos, R. Babuška, and H.B. Verbruggen. Mimo predictive control by multiple-step linearization of takagi-sugeno fuzzy models. In *AIRTC*, Arizona, US, 1998.

[9] B.E. Postlethwaite. Building a model-based fuzzy controller. *Fuzzy Sets and Systems*, 79:3–13, 1996.

[10] J. Richalet. Industrial applications of model based predictive control. *Automatica*, 29:1251–1274, 1993.

[11] J.A. Roubos, R. Babuška, and H.B. Verbruggen. Predictive control by local linearization of a Takagi-Sugeno fuzzy model. In *FUZZ-IEEE*, pages 37–42, Anchorage, Alaska, 1998.

[12] J.A. Roubos, S. Mollov, R. Babuska, and H.B. Verbruggen. Fuzzy model based predictive control by using takagi-sugeno fuzzy models. *Int. Journal of Approximate Reasoning*, 1999. To appear.

[13] T. Takagi and M. Sugeno. Fuzzy identification of systems and its application to modeling and control. *IEEE Transactions on Systems, Man and Cybernetics*, 15:116–132, 1985.

[14] R.R. Yager and D.P. Filev. *Essentials of Fuzzy Modeling and Control.* Johnn Wiley, New York, 1994.