Int. J. Appl. Math. Comput. Sci., 2009, Vol. 19, No. 1, 127–141
DOI: 10.2478/v10006-009-0012-8

EFFECTIVE DUAL–MODE FUZZY DMC ALGORITHMS WITH ON–LINE QUADRATIC OPTIMIZATION AND GUARANTEED STABILITY

PIOTR M. MARUSAK, PIOTR TATJEWSKI

Institute of Control and Computation Engineering
Warsaw University of Technology, Nowowiejska 15/19, 00–665 Warsaw, Poland
e-mail: {Marusak,Tatjewski}@ia.pw.edu.pl

Dual-mode fuzzy dynamic matrix control (fuzzy DMC-FDMC) algorithms with guaranteed nominal stability for constrained nonlinear plants are presented. The algorithms join the advantages of fuzzy Takagi-Sugeno modeling and the predictive dual-mode approach in a computationally efficient version. Thus, they can bring an improvement in control quality compared with predictive controllers based on linear models and, at the same time, control performance similar to that obtained using more demanding algorithms with nonlinear optimization. Numerical effectiveness is obtained by using a successive linearization approach resulting in a quadratic programming problem solved on-line at each sampling instant. It is a computationally robust and fast optimization problem, which is important for on-line applications. Stability is achieved by appropriate introduction of dual-mode type stabilization mechanisms, which are simple and easy to implement. The effectiveness of the proposed approach is tested on a control system of a nonlinear plant—a distillation column with basic feedback controllers.

Keywords: nonlinear systems, fuzzy systems, model predictive control, stability, constrained control, dual-mode control.

1. Introduction

Predictive control algorithms for nonlinear plants are widely investigated nowadays. A general and theoretically optimal approach consists in solving a constrained nonlinear programming problem at each sampling instant. Unfortunately, this optimization problem is in general nonconvex. Therefore, the solution itself, even if found by an applied numerical procedure, could only be a local extremum. However, the stability results presented in most theoretical papers are based on the assumption that a global optimum is found (Mayne *et al*., 2000).

On the other hand, the amount of computations needed to solve the constrained nonlinear programming problem on-line usually makes the approach inapplicable in practical implementations. Moreover, the time needed to obtain the result cannot be anticipated and guaranteed. These are the reasons why, despite problems with theoretical analysis, in particular with the formulation of stability conditions, "…linearization is the only method which has found any wider use in industry beyond demonstration projects. For industry there has to be a clear justification for solving non-linear programs on-line in a dynamic setting and there are no examples to bear that out in a convincing manner." (Morari and Lee, 1999). That is why algorithms using linearization, though sub-optimal, were developed in order to formulate the problem of predictive control calculation as a convex linear-quadratic optimization problem (Garcia, 1984; Gattu and Zafiriou, 1992; Lee and Ricker, 1994; Li and Biegler, 1989; Marusak and Tatjewski, 2000; Mutha *et al*., 1997; 1998).

In this paper, stable fuzzy DMC (FDMC) algorithms using Takagi-Sugeno (TS) fuzzy models and a successive linearization approach are proposed. They are based on a DMC model predictive control (MPC) algorithm. The DMC algorithm is a standard one in industrial applications (Blevins *et al*., 2003; Rossiter, 2003; Tatjewski, 2007). Rossiter (2003) claims that it is the most popular algorithm in industrial applications, and Blevins *et al.* (2003) remark that "Most MPC implementations to date use step response models proven in DMC applications." It is so despite the fact that step response models can be applied only to open-loop stable plants. However, it should be emphasized that MPC is usually applied in a multilayer control structure, in the advanced control (constraint control) layer being above the basic control (direct control) layer which is responsible for the stabilization of the control plant, see, e.g., (Tatjewski, 2007). Thus, MPC is, in most cases, applied to stable or pre-stabilized control plants (as it is done in an example discussed later). That is why DMC type algorithms are widely used in practical applications.

The idea of the discussed FDMC algorithms is to obtain at each iteration (at each sampling instant) an approximate, linearized process model and then to use the same numerical calculations as in the standard DMC algorithm (with a linear process model). The advantage of the use of the fuzzy nonlinear model is that the approximate process model is obtained in a simple and natural way using fuzzy reasoning.

In the first part of the paper, the idea of Takagi-Sugeno fuzzy models is shortly reminded. Then, a general formulation of FDMC algorithms based on fuzzy reasoning and a solution of an approximate quadratic optimization problem at each sampling instant is described. Later, the explicit unconstrained FDMC algorithm (a control law resulting from an analytical solution to the optimization problem) is presented. The next section details original results concerning modifications of FDMC algorithms performed in such a way that nominal stability can be proved. Finally, an application of the presented approach to a control system of a distillation column is described. The paper is concluded by a short summary. For sake of simplicity, but without loss of generality, all formulae in the paper are derived for the single-input single-output case.

2. Takagi-Sugeno fuzzy modeling

In this section, the idea of Takagi-Sugeno fuzzy models (Takagi and Sugeno, 1985) is shortly reminded. Takagi-Sugeno fuzzy models consist of a set of rules. The algorithms described in the paper exploit models which consist of the following rules:

$$\begin{aligned}&\text{Rule } i: && (1)\\ &\text{if} \quad \underbrace{\begin{array}{l} y_k \text{ is } B_1^i \text{ and } \dots \text{ and } y_{k-n_P+1} \text{ is } B_{n_P}^i \text{ and} \\ u_k \text{ is } C_1^i \text{ and } \dots \text{ and } u_{k-m_P+1} \text{ is } C_{m_P}^i \end{array}}_{\text{antecedent}} \\ &\text{then} \quad \underbrace{\begin{array}{l} y_{k+1}^i = b_1^i \cdot y_k + \dots + b_{n_B}^i \cdot y_{k-n_B+1} \\ \qquad +c_1^i \cdot u_k + \dots + c_{m_C}^i \cdot u_{k-m_C+1}, \end{array}}_{\text{consequent}}\end{aligned}$$

where $b_1^i, \dots, b_{n_B}^i, c_1^i, \dots, c_{m_C}^i$ are the coefficients of the i-th local (linear) model, y_k stands for the value of the output variable of the control plant model at the k-th sampling instant, u_k stands for the value of the manipulated variable at the k-th sampling instant, $B_1^i, \dots, B_{n_P}^i, C_1^i, \dots, C_{m_P}^i$ denote fuzzy sets, $i = 1, \dots, l$, and l is the number of rules.

The consequents in Takagi-Sugeno models are called local models. They are usually linear (it is possible to use nonlinear models). The Takagi-Sugeno models can be therefore treated as a kind of a nonlinear generalization of linear models. This is the reason why they are also called quasi-linear models. Since the consequents are functions, using Takagi-Sugeno models one can describe relatively complicated dynamics employing relatively few rules. In the paper, input-output type models are used.

In order to derive the output of the model, one should perform fuzzy reasoning. It consists in:

1. Calculation, for each rule, of values of weighting coefficients w_i (called levels of activation or firing strengths), $i = 1, \dots, l$.

2. Determination of the values of the consequent functions y_{k+1}^i, $i = 1, \dots, l$.

3. Calculation of a weighted sum of the consequent values according to the formula

$$y_{k+1} = \frac{\sum_{i=1}^{l} w_i \cdot y_{k+1}^i}{\sum_{i=1}^{l} w_i}, \tag{2}$$

where

$$w_i = \prod_{j=1}^{n_P} \mu^{B_j^i}(y_{k-j+1}) \cdot \prod_{j=1}^{m_P} \mu^{C_j^i}(u_{k-j+1}), \tag{3}$$

$\mu^{B_j^i}(y_{k-j+1})$ and $\mu^{C_j^i}(u_{k-j+1})$ are the values of the membership functions obtained for a current operating point. Further in the text, the normalized weights $\widetilde{w}_i = w_i / \sum_{i=1}^{l} w_i$ are used.

A thorough description of the idea of fuzzy logic and fuzzy modelling can be found in the abundant literature, e.g., in (Driankov *et al.*, 1993; Piegat, 2001; Yager and Filev, 1994).

Remark 1. Antecedents used in the fuzzy control plant model (1) are the most common ones. However, not only conjunction but also the disjunction operation can be used in them. Moreover, even more complex antecedents can be used (Chen *et al.*, 1998). It is also possible to use different operators during fuzzy reasoning. Stable fuzzy algorithms proposed in the paper can use a TS model with practically any type of antecedents. The only demand is that the weights w_i should be normalized. It is, thus, also possible to use models in which the weights are given as analytical formulae for each rule and in the model there is no explicit antecedent part (Cao *et al.*, 1997).

3. Nonlinear fuzzy DMC algorithms

In this section, numerical fuzzy DMC (FDMC) algorithms will be presented. They are based on two approaches: the DMC predictive control technique and the Takagi-Sugeno fuzzy modeling, inheriting advantages of both approaches (Marusak and Tatjewski, 2000). More detailed discussions of these algorithms and their properties are presented in (Marusak, 2002). The usage of these algorithms usually gives satisfactory results even for highly nonlinear plants with large time delays.

3.1. Standard DMC algorithm. In the DMC algorithm based on a linear control plant model, the following performance index is minimized (Garcia and Morshedi, 1986):

$$\min_{\Delta \boldsymbol{u}} \sum_{i=1}^{p} \left(y_k^{sp} - y_{k+i|k}\right)^2 + \lambda \cdot \sum_{i=0}^{s} \left(\Delta u_{k+i|k}\right)^2, \quad (4)$$

subject to the constraints

$$\Delta \boldsymbol{u}^{\min} \le \Delta \boldsymbol{u} \le \Delta \boldsymbol{u}^{\max}, \quad (4a)$$
$$\boldsymbol{u}^{\min} \le \boldsymbol{u} \le \boldsymbol{u}^{\max}, \quad (4b)$$
$$\boldsymbol{y}^{\min} \le \boldsymbol{y} \le \boldsymbol{y}^{\max}, \quad (4c)$$

where y_k^{sp} is a set-point value, $y_{k+i|k}$ is an output value for the $(k+i)$-th sampling instant predicted at the k-th sampling instant, $\Delta u_{k+i|k}$ is a change in the manipulated variable for the $(k+i)$-th sampling instant calculated at the k-th sampling instant, p and s denote prediction and control horizons, respectively, $\lambda \ge 0$ is a penalty coefficient, $\Delta \boldsymbol{u} = \left[\Delta u_{k|k}, \ldots, \Delta u_{k+s-1|k}\right]^T$ is a vector of future manipulated variable changes,

$$\boldsymbol{u} = \left[u_{k-1} + \Delta u_{k|k}, \ldots, u_{k-1} + \sum_{i=0}^{s-1} \Delta u_{k+i|k}\right]^T$$

is a vector of future manipulated variable values, $\boldsymbol{y} = \left[y_{k+1|k}, \ldots, y_{k+p|k}\right]^T$ is a vector of predicted output values, $\Delta \boldsymbol{u}^{\min}$, $\Delta \boldsymbol{u}^{\max}$, $\boldsymbol{u}^{\min}$, $\boldsymbol{u}^{\max}$, $\boldsymbol{y}^{\min}$, $\boldsymbol{y}^{\max}$ are vectors of upper and lower bounds on changes and values of the control signals and on the output variable values, respectively.

The standard (nonfuzzy) DMC algorithm uses the control plant model in the form of its step response (Camacho and Bordons, 1999; Cutler and Ramaker 1979; Maciejowski, 2002; Tatjewski, 2007):

$$y_k^M = \sum_{i=1}^{p_d-1} a_i \cdot \Delta u_{k-i} + a_{p_d} \cdot u_{k-p_d}, \quad (5)$$

where y_k^M is the output of the plant model at the k-th sampling instant, Δu_k is a change in the manipulated variable at the k-th sampling instant, a_i $(i = 1, \ldots, p_d)$ are step response coefficients of the control plant, p_d is equal to the number of sampling instants after which the step response coefficients can be assumed as settled (the value of a_{p_d} is assumed equal or close to the static gain of the control plant), u_{k-p_d} is the value of the manipulated variable at the $(k-p_d)$-th sampling instant.

The predicted output values are then calculated using the following formula:

$$y_{k+i|k} = \sum_{j=1}^{i} a_j \cdot \Delta u_{k-j+i|k} + \sum_{j=i+1}^{p_d-1} a_j \cdot \Delta u_{k-j+i} + a_{p_d} \cdot u_{k-p_d+i} + d_k, \quad (6)$$

where $d_k = y_k - y_{k-1}^M$ is an unmeasured disturbance estimate assumed to be the same at each instant in the prediction horizon (a DMC type disturbance model). Using (5), Eqn. (6) can be transformed to the form

$$y_{k+i|k} = y_k + \sum_{j=i+1}^{p_d-1} a_j \cdot \Delta u_{k-j+i} + a_{p_d} \cdot \sum_{j=p_d}^{p_d+i-1} \Delta u_{k-j+i} - \sum_{j=1}^{p_d-1} a_j \cdot \Delta u_{k-j} + \sum_{j=1}^{i} a_j \cdot \Delta u_{k-j+i|k}, \quad (7)$$

where only the last component depends on future changes in the manipulated variable. Thus, the vector of predicted output values $\boldsymbol{y}$ can be decomposed into the following components:

$$\boldsymbol{y} = \boldsymbol{y}^{fr} + \boldsymbol{A} \cdot \Delta \boldsymbol{u}, \quad (8)$$

where $\boldsymbol{y}^{fr} = \left[y_{k+1|k}^{fr}, \ldots, y_{k+p|k}^{fr}\right]^T = \boldsymbol{y}_k + \boldsymbol{A}^p \cdot \Delta \boldsymbol{u}^p$ is called a free response of the plant, because it contains future output values calculated assuming that the control signal does not change on the prediction horizon (it describes the influence of the manipulated variable values applied to the control plant in previous iterations),

$$\boldsymbol{A}^p = \begin{bmatrix} a_2 - a_1 & a_3 - a_2 & \ldots & a_{p_d-1} - a_{p_d-2} & a_{p_d} - a_{p_d-1} \\ a_3 - a_1 & a_4 - a_2 & \ldots & a_{p_d} - a_{p_d-2} & a_{p_d} - a_{p_d-1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{p+1} - a_1 & a_{p+2} - a_2 & \ldots & a_{p_d} - a_{p_d-2} & a_{p_d} - a_{p_d-1} \end{bmatrix}, \quad (9)$$

$$\boldsymbol{y}_k = [\underbrace{y_k, \ldots, y_k}_{p \text{ elements}}]^T,$$

$\Delta \boldsymbol{u}^p = \left[\Delta u_{k-1}, \ldots, \Delta u_{k-p_d}\right]^T$ is a vector of past manipulated variable changes, $\boldsymbol{A} \cdot \Delta \boldsymbol{u}$ is called the forced response, where $\boldsymbol{A}$ is the dynamic matrix of the form

$$\boldsymbol{A} = \begin{bmatrix} a_1 & 0 & \ldots & 0 & 0 \\ a_2 & a_1 & \ldots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_p & a_{p-1} & \ldots & a_{p-s+2} & a_{p-s+1} \end{bmatrix}. \quad (10)$$

Details concerning the formulation of the DMC algorithm can be found, e.g., in (Camacho and Bordons, 1999; Cutler and Ramaker 1979; Maciejowski, 2002; Tatjewski, 2007).

The optimal vector of changes in the manipulated variable is obtained as a solution to the optimization problem (4). From this vector, the first element is taken and applied to the control plant. Then, optimization is repeated at the next sampling instant.

3.2. Fuzzy DMC algorithms. There will be two FDMC algorithms presented, based on Takagi-Sugeno (TS) fuzzy models described in Section 2. The idea of all these algorithms is as follows: At each algorithm iteration (sampling instant):

- the process outputs are measured and a corresponding linear process model is obtained from the nonlinear TS model (linearization);
- using the linearized model for the prediction of the process behaviour, the optimization problem (4) is formulated (which is then a quadratic programming problem) and solved;
- the optimal value $\Delta u_{k|k}$ is applied to the process.

The differences between the algorithms result from different ways in which the approximated, linear plant models are obtained and utilized.

The FDMC algorithm with successive linearization (FDMC-SL) is the simplest one. In this algorithm the same linear model is used for the calculation of the free and forced responses (the prediction of the influence of both past and future values of the manipulated variable on the future plant outputs). This model is obtained at each algorithm iteration for current values of the manipulated and output variables using the nonlinear TS plant model. A detailed description of the algorithm formulation is presented below, and its diagram is given in Fig. 1.

At each algorithm iteration, the following sequence is repeated:

1. Using the TS fuzzy model (1), a linear (linearized) model for the current sampling instant is derived. Let us recall that the TS model is composed of the rules (1), where, for clarity of presentation, the delay was not specified. However, in order to take the delay into consideration, it is sufficient to assume $c_1^i = \ldots = c_d^i = 0$ in the local models, where d is the delay. The output of the model is derived using the standard formula

$$y_{k+1} = \sum_{i=1}^{l} \widetilde{w}_i \cdot y_{k+1}^i, \tag{11}$$

where the weights $\widetilde{w}_i$ are calculated as was described in Section 2. The formula (11) can be written in the following form:

$$\begin{aligned} y_{k+1} &= b_1 \cdot y_k + \cdots + b_{n_B} \cdot y_{k-n_B+1} \\ &\quad + c_1 \cdot u_k + \cdots + c_{m_C} \cdot u_{k-m_C+1}, \end{aligned} \tag{12}$$

where $b_j = \sum_{i=1}^{l} \widetilde{w}_i \cdot b_j^i$, $c_j = \sum_{i=1}^{l} \widetilde{w}_i \cdot c_j^i$.

The next steps of the FDMC-SL algorithm are the same as in the standard DMC algorithm:

2. The derived linear model (12) is used to calculate the step response coefficients. Using these coefficients, a dynamic matrix is generated (Section 3.1).

3. The linear model (12) is used to generate the free response of the plant.

4. The free response and the dynamic matrix are used to formulate the quadratic optimization problem.

5. The optimization problem is solved and, using the obtained solution, the manipulated variable value is generated. Then, the controller passes to the next iteration.

If the control plant is highly nonlinear, the inaccuracy of the approximated model used for prediction may lead to insufficient control performance offered by the FDMC-SL algorithm. An improvement may be then achieved by the application of the modified FDMC-SL algorithm, as presented below.

The FDMC algorithm with successive nonlinear prediction and linearization (FDMC-NPL) calculates the free response of the control plant using a nonlinear (in our case, TS fuzzy) plant model instead of a linearized model calculated at the current process point. Then, only the calculation of the forced response is based on the linear model. This rather simple modification considerably improves the ability to cope with plant nonlinearities. Compared with the FDMC-SL algorithm, Steps 1, 2, 4 and 5 do not change (they are the same as in the FDMC-SL algorithm). The difference occurs in Step 3:

3. The free response of the control plant is derived using the nonlinear process model, available past output measurements and past manipulated variable values. Moreover, the manipulated variable values at present and future sampling instants (not known yet) can be chosen in one of the following ways:

(a) They can be assumed equal to the value of the manipulated variable recently applied to the plant u_{k-1} (the standard free response).

(b) The free response is derived assuming the application of values of the manipulated variable generated by the algorithm in the previous, $(k-1)$-st, sampling instant, i.e., $u_{k|k-1}, u_{k+1|k-1}, \ldots, u_{k+s-2|k-1}$. Such an approach usually gives better results than the first one. This is because the manipulated variable increments obtained after solving the optimization problem are usually

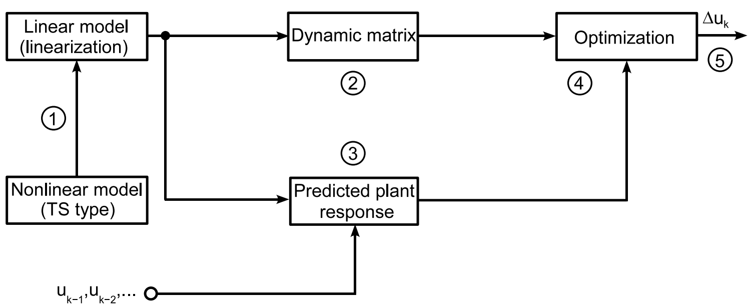


Fig. 1. Block diagram of FDMC-SL algorithm formulation.

smaller than in the first approach. Thus, the modelling inaccuracy caused by the linearization is also smaller.

The second approach will be now described in detail. The first element of the plant response can be derived using the following formulae:

Rule i : (13)

$$\begin{aligned} \text{if}\quad & y_k \text{ is } B_1^i \text{ and } \ldots \text{ and } y_{k-n_P+1} \text{ is } B_{n_P}^i \text{ and} \\ & u_{k|k-1} \text{ is } C_1^i \text{ and } \ldots \text{ and } u_{k-m_P+1} \text{ is } C_{m_P}^i \\ \text{then}\quad & y_{k+1|k}^{fr,i} = b_1^i \cdot y_k + \cdots + b_{n_B}^i \cdot y_{k-n_B+1} \\ & \quad + c_1^i \cdot u_{k|k-1} + \cdots \\ & \quad + c_{m_C}^i \cdot u_{k-m_C+1}, \end{aligned}$$

$$y_{k+1|k}^{fr} = \sum_{i=1}^{l} \widetilde{w}_1^i \cdot y_{k+1|k}^{fr,i}. \quad (14)$$

The next elements of the response (for further sampling instants in the prediction horizon) are obtained recursively using the values of the manipulated variable $u_{k|k-1}, u_{k+1|k-1}, \ldots, u_{k+s-2|k-1}$, derived by the algorithm in the $(k-1)$-st sampling instant and already calculated elements of the plant response. Thus, for the j-th sampling instant ($j = 1, \ldots, p$), one obtains

Rule i : (15)

$$\begin{aligned} \text{if}\quad & y_{k+j-1|k}^{fr} \text{ is } B_1^i \text{ and } \ldots \text{ and } y_{k-n_P+j} \text{ is } B_{n_P}^i \text{ and} \\ & u_{k+j-1|k-1} \text{ is } C_1^i \text{ and } \ldots \text{ and } u_{k-m_P+j} \text{ is } C_{m_P}^i \\ \text{then}\quad & y_{k+j|k}^{fr,i} = b_1^i \cdot y_{k+j-1|k}^{fr} + \cdots + b_{n_B}^i \cdot y_{k-n_B+j} \\ & \quad + c_1^i \cdot u_{k+j-1|k-1} + \cdots \\ & \quad + c_{m_C}^i \cdot u_{k-m_C+j}, \end{aligned}$$

$$y_{k+j|k}^{fr} = \sum_{i=1}^{l} \widetilde{w}_j^i \cdot y_{k+j|k}^{fr,i}. \quad (16)$$

The predicted output values are described by the following equation:

$$y_{k+j|k} = \sum_{i=1}^{j} a_i \cdot \Delta\widetilde{u}_{k-i+j|k} + y_{k+j|k}^{fr} + d_k, \quad (17)$$

where a_i are step response coefficients, $\Delta\widetilde{u}_{k-i+j|k}$ are future manipulated variable increments used during full plant response derivation in the optimization problem, $y_{k+j|k}^{fr}$ are elements of the system response derived using the procedure described above, d_k is the DMC disturbance estimate. The block diagram of the FDMC-NPL algorithm formulation is shown in Fig. 2.

There is a possibility to iteratively improve the nonlinear free and linear forced responses in the FDMC-NPL algorithm, according to the general NPL+ structure, as presented in (Tatjewski, 2007). The idea consists in repeating the linearization and the optimization several times during one iteration of the algorithm (sub-iterations are performed). Then the obtained plant response is changing each time the optimization is repeated. As a result, the obtained manipulated variable increments (calculated using a linear model) become smaller at each sub-iteration. Thus, the modelling inaccuracy caused by the linearization becomes smaller at each sub-iteration.

3.3. Explicit, unconstrained fuzzy DMC control laws.

The idea of the DMC algorithm is to minimize the performance index (4) at each sampling instant, see Section 3.1. The problem without inequality constraints (4a–c) has a unique solution, which can be expressed as (Camacho and Bordons, 1999; Tatjewski, 2007)

$$\Delta\boldsymbol{u} = \left(\boldsymbol{A}^T \cdot \boldsymbol{A} + \lambda \cdot \boldsymbol{I}\right)^{-1} \cdot \boldsymbol{A}^T \cdot \left(\boldsymbol{y}^{sp} - \boldsymbol{y}^{fr}\right), \quad (18)$$

where $\boldsymbol{I}$ is the identity matrix,

$$\boldsymbol{y}^{sp} = [\underbrace{y_k^{sp}, \ldots, y_k^{sp}}_{p \text{ elements}}]^T.$$

Only the first element of the vector $\Delta\boldsymbol{u}$ is applied to the process. Therefore, it is possible to define the structure of the controller. The control law is as follows:

$$\Delta u_k = -r_0 \cdot e_k + \sum_{i=1}^{p_d-1} r_i \cdot \Delta u_{k-i}, \quad (19)$$

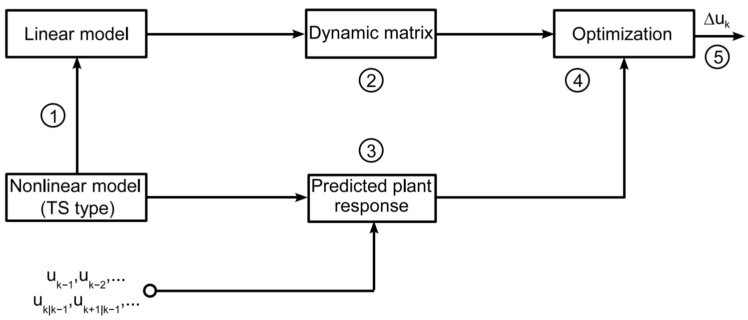


Fig. 2. Block diagram of FDMC-NPL algorithm formulation.

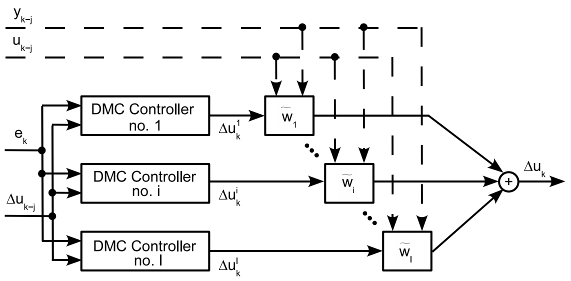


Fig. 3. Block diagram of the analytical FDMC controller.

where $e_k = y_k^{sp} - y_k$ is a control error at the k-th sampling instant, $r_0 = \sum_{j=1}^{p} b_{1j}$, $[r_1, \ldots, r_{p_d}] = \boldsymbol{K}_1 \cdot \boldsymbol{A}^p$, $\boldsymbol{K}_1 = [b_{11}, \ldots, b_{1p}]$ is the first row of the matrix $\boldsymbol{K} = \left(\boldsymbol{A}^T \cdot \boldsymbol{A} + \lambda \cdot \boldsymbol{I}\right)^{-1} \cdot \boldsymbol{A}^T$.

The fuzzy DMC controller based on an explicit, unconstrained formulation consists of several linear DMC controllers of the form (19) (Marusak and Tatjewski, 2000; Tatjewski, 2007). The idea of the algorithm consists in obtaining a vector of step response coefficients for every linear local model in the TS fuzzy model (1). Then, using these responses, coefficients of local controllers described by (19) are calculated. The weights for the local controllers are derived, at each iteration, using fuzzy reasoning. Then, the output of the controller is calculated as the following weighted sum of the outputs of the local controllers:

$$\Delta u_k = \sum_{i=1}^{l} \widetilde{w}_i \cdot \Delta u_k^i, \qquad (20)$$

where Δu_k^i is the output of the i-th local controller given by (19), $\widetilde{w}_i$ are the normalized weights (activating levels of the corresponding fuzzy rules), $i = 1, \ldots, l$, and l means the number of local controllers. The block diagram of the analytical FDMC controller is shown in Fig. 3.

It is possible to analyze the stability of the nonlinear control system with the discussed explicit unconstrained fuzzy DMC controller (Marusak and Tatjewski, 2001; 2002). The method of stability analysis is based on a transformation of the control system description to the appropriate form using a quasi-state vector. Then, the Tanaka-Sugeno stability criterion (Tanaka and Sugeno, 1992) can be applied, consisting in solving a set of Lyapunov type linear matrix inequalities. Using this procedure, a Lyapunov matrix for the nonlinear control system under consideration can be found. Further in the paper, $\boldsymbol{P}$ denotes this Lyapunov matrix.

4. Stable dual-mode fuzzy DMC algorithms

FDMC algorithms with guaranteed nominal stability, which combine several different elements applied to stabilize control systems with predictive controllers, will be presented in this section. The algorithms, the idea of which was generally presented in (Marusak and Tatjewski, 2003), use the suboptimal dual-mode approach as discussed in (Scokaert *et al.*, 1999), being a mutation of the dual-mode algorithm introduced in (Michalska and Mayne, 1993). In fact, in this approach two controllers are designed. The first one, a constrained MPC controller, should bring the trajectory of the control system into a (convex) target set W, which contains the equilibrium point in its interior. This set and the second (stabilizing) unconstrained feedback controller, which is used if the state of the process is inside the set W, should be chosen in such a way that the following conditions are fulfilled:

1. The set W is inside the admissible set.
2. The control system (with the stabilizing controller) is asymptotically stable in this set.
3. Any trajectory of the control system starting in the set W remains there (W is an invariant set).

The idea of the dual-mode approach is reminded in Fig. 4. In the proposed approach, outside the set W a numerical FDMC algorithm with an additional stabilizing constraint put on manipulated variable values is used. As a stabilizing controller, standard or fuzzy unconstrained DMC algorithms, recalled in Section 3.3, are used. This

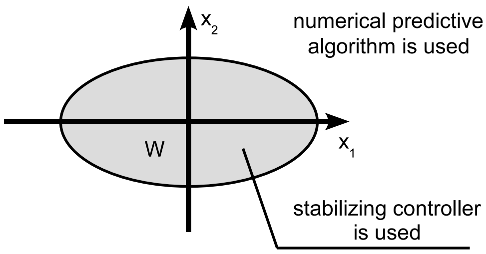


Fig. 4. Idea of the dual-mode approach.

leads to a simplification of the proposed stable FDMC algorithms, in the way explained later. Any version of the numerical FDMC algorithm can then be used as a first stage algorithm.

4.1. Preliminary assumptions. For further analysis, it is assumed that the control plant is stable. It is a standard assumption in DMC and FDMC algorithms (a detailed discussion was presented in Introduction). Moreover, it is assumed that the control plant is described by a Takagi-Sugeno fuzzy model and that there are no modelling errors (the nominal case).

In the sequel, the following quasi-state vector consisting of past output and input values is used:

$$\begin{aligned}\overline{\boldsymbol{x}}_k = [&y_k - y^{sp} \ \dots \ y_{k-n+1} - y^{sp} \\ &u_{k-d-1} - u_s \ \dots \ u_{k-d-p_d} - u_s]^T, \qquad (21)\end{aligned}$$

where u_s is the manipulated variable value in a steady-state for y^{sp}, d is the time delay in the control plant and n is the number of the past output values used in the control plant model. It is also assumed that the set-point value remains constant during the stability analysis, i.e., $y_k^{sp} = y^{sp}$.

4.2. FDMC algorithms with a stabilizing constraint. Let us consider the following nonlinear optimization problem:

$$\min_{\boldsymbol{u}} \phi_k \qquad (22)$$

$$= \sum_{j=0}^{p-1} \theta(\overline{\boldsymbol{x}}_{k+j|k}) \left(\overline{\boldsymbol{x}}_{k+j|k}^T \boldsymbol{Q} \overline{\boldsymbol{x}}_{k+j|k} + R(u_{k+j|k} - u_s)^2 \right)$$

subject to the constraints

$$\begin{aligned}y_{k+1} = f(y_k, \dots, y_{k-a}, u_k, \dots, u_{k-b}), &\qquad (22a)\\ \overline{\boldsymbol{x}}_{k+p} \in W, &\qquad (22b)\\ \Delta u^{\min} \le \Delta u_{k+j|k} \le \Delta u^{\max}, &\qquad (22c)\\ u^{\min} \le u_{k+j|k} \le u^{\max}, &\qquad (22d)\\ y^{\min} \le y_{k+j|k} \le y^{\max}, &\qquad (22e)\end{aligned}$$

where

$$\theta(\overline{\boldsymbol{x}}_{k+j|k}) = \begin{cases} 0 & \text{for} \quad \overline{\boldsymbol{x}}_{k+j|k} \in W, \\ 1 & \text{for} \quad \overline{\boldsymbol{x}}_{k+j|k} \notin W, \end{cases}$$

and (22a) is a Takagi-Sugeno fuzzy model of the form (1),

$$\begin{aligned}a &= \max\{k - n_P + 1, k - n_B + 1\},\\ b &= \max\{k - m_P + 1, k - m_C + 1\}.\end{aligned}$$

In the standard nonlinear dual-mode approach, the problem (22) is typically used in the algorithm that acts outside the set W and generates a control signal that should bring the plant state into the set W. However, this problem is in general nonconvex and difficult to solve. This difficulty can be omitted using the following approach: Namely, if the state of the system is outside the set W, the control signal is obtained using one of the numerical FDMC algorithms (as described in Section 3.2) solving the quadratic optimization problem (4). However, the FDMC algorithms used must be slightly modified by adding the following stabilizing constraint to the problem (4):

$$u_s = u_{k-1} + \sum_{i=0}^{s-1} \Delta u_{k+i|k}. \qquad (23)$$

This constraint is an analogue of the constraint (22b) in the problem (22). However, it is a different constraint than a direct demand for the state to approach the set W. Because a stable control plant was assumed, the state will asymptotically approach the equilibrium point (by assumption located inside W) if the constraint (23) is fulfilled at the end of the control horizon. This means that it will approach the set W.

4.3. Stabilizing controller. Typically, in dual-mode predictive controllers, a linear state-feedback controller is used inside the set W as a stabilizing controller. But different stabilizing controllers ensuring stability inside W can be used as well. In particular, a nonlinear feedback controller, which could perform better, can be used. Therefore, we will further assume that an unconstrained explicit FDMC controller (Section 3.3) is used. Let this controller be denoted as $u = h_{\mathrm{FDMC}}(\overline{\boldsymbol{x}})$.

Now, only the set W must be derived. It is worth noticing that the explicit FDMC controller is a natural nonlinear generalization of a linear controller and actually consists of several such controllers. This fact will be used to find the set W. In the case of a linear controller, the method presented below is simpler because then a smaller number of optimization problems must be solved to calculate W.

4.4. Efficient algorithm for target set derivation. In dual-mode predictive controller design, the main problem is to obtain the set W. Typically, in order to find the set W, a complicated method from (Michalska and Mayne, 1993) is used. This method is sufficient if one has a good procedure for solving nonlinear optimization problems. But this

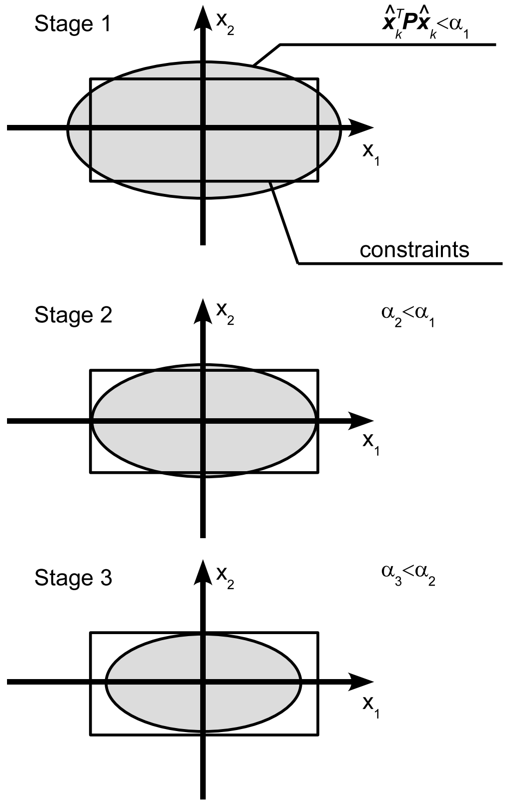


Fig. 5. Idea of the approach applied to find the set W.

condition is not always easy to fulfil. In this paper, a simpler method was developed, taking advantage of the structure of the proposed algorithms. In this method, which is much simpler than the mentioned original one, it is sufficient to solve a number of simple quadratic-programming problems, each with one linear equality constraint only. The W set is to be found among the hyper-ellipsoidal sets, defined by $\overline{\boldsymbol{x}}_k^T \boldsymbol{P} \overline{\boldsymbol{x}}_k \leq \alpha$. Thus, the proper value of α should be chosen, in such a way that the hyper-ellipsoid is contained inside the admissible set. The set W calculated this way is invariant because $\boldsymbol{P}$ is the Lyapunov matrix of the control system with the stabilizing controller.

The proposed algorithm is based on the fact that constraints (4a–c) are linear and uses the observation that it is sufficient to check if a particular constraint (defining a hyper-plain) has common points with the hyper-ellipsoid given by the constraint $\overline{\boldsymbol{x}}_k^T \boldsymbol{P} \overline{\boldsymbol{x}}_k \leq \alpha$, for an assumed value of a parameter α. If this is true, the value of α is decreased. If not, then the condition for a next constraint is checked. At the end of the procedure, when there are no common points, the desired value of α is found. The idea of this approach is shown in Fig. 5.

The detailed algorithm is as follows:

1. An initial value of α is chosen (it should not be too small).

2. Optimization problems

$$\min_{\overline{\boldsymbol{x}}_k} \left(\overline{\boldsymbol{x}}_k^T \boldsymbol{P} \overline{\boldsymbol{x}}_k \right), \tag{24}$$

subject to

$$\boldsymbol{G}^q \overline{\boldsymbol{x}}_k = g_q, \tag{25}$$

are solved subsequently, where $\boldsymbol{G}^q \overline{\boldsymbol{x}}_k = g_q$ $(q = 1, \dots, 2m + 4l)$ is one of the linear constraints from the following set of $2m + 4l$ (in the single-input single-output case) constraints:

$$\boldsymbol{A}^j \overline{\boldsymbol{x}}_k = \boldsymbol{A}^j \overline{\boldsymbol{x}}_k^{\min}, \qquad j = 1, \dots, m, \tag{25a}$$

$$\boldsymbol{A}^j \overline{\boldsymbol{x}}_k = \boldsymbol{A}^j \overline{\boldsymbol{x}}_k^{\max}, \qquad j = 1, \dots, m, \tag{25b}$$

$$\boldsymbol{K}^i \overline{\boldsymbol{x}}_k = u_k^{\min}, \qquad i = 1, \dots, l, \tag{25c}$$

$$\boldsymbol{K}^i \overline{\boldsymbol{x}}_k = u_k^{\max}, \qquad i = 1, \dots, l, \tag{25d}$$

$$\widetilde{\boldsymbol{K}}^i \overline{\boldsymbol{x}}_k = \Delta u_k^{\min}, \qquad i = 1, \dots, l, \tag{25e}$$

$$\widetilde{\boldsymbol{K}}^i \overline{\boldsymbol{x}}_k = \Delta u_k^{\max}, \qquad i = 1, \dots, l, \tag{25f}$$

where

$$\boldsymbol{A}^j = [\overbrace{0 \ \dots \ 0}^{j-1} \ 1 \ 0 \ \dots \ 0]^T$$

is a vector with the same number of elements as the quasi-state vector (21), with 1 at the j-th place, m is the length of the quasi-state vector, $\overline{\boldsymbol{x}}_k^{\min}$ and $\overline{\boldsymbol{x}}_k^{\max}$ are lower and upper bounds of the quasi-state vector, $u_k^{\min}$ and $u_k^{\max}$ are lower and upper bounds on manipulated variable values, $\Delta u_k^{\min}$ and $\Delta u_k^{\max}$ are lower and upper bounds on manipulated variable changes, $\boldsymbol{K}^i$ is the vector of coefficients of the i-th local controller, obtained after a suitable transformation of the local control law given by (19), $\widetilde{\boldsymbol{K}}^i$ is the vector of coefficients of the i-th local controller, obtained after a suitable transformation of the incremental form of the local control law.

Remark 2. The first n constraints in (25a) and (25b) are appropriately transformed constraints imposed on the output variable, the next p_d constraints are imposed on the manipulated variable, as well as the constraints (25c) and (25d). The constraints (25e) and (25f) are imposed on the manipulated variable changes.

3. After solving the problem (24), (25), the condition $\hat{\boldsymbol{x}}_k^T \boldsymbol{P} \hat{\boldsymbol{x}}_k < \alpha$ is checked, where $\hat{\boldsymbol{x}}_k^T$ denotes a solution to the optimization problem. If this is true, then the constraint has common points with the hyper-ellipsoid defined by this condition. In such a case, the value of the parameter α is decreased. If the condition is not fulfilled, then the next problem from the set is solved $(q := q + 1)$ and so on until $\hat{\boldsymbol{x}}_k^T \boldsymbol{P} \hat{\boldsymbol{x}}_k \geq \alpha$ for all single-constrained problems (24), (25).

Let us notice that the described algorithm of finding the set W uses an effective optimization method and produces the solution very fast, which is important in the case considered, because after a set-point change a new set W must be found again. The effectiveness is due to the fact that the optimization problems (24), (25) have analytical solutions (which can be formulated analogously as a solution to the optimization problem in the explicit version of the DMC algorithm in Section 3.3).

Using the proposed algorithm and decreasing, gradually, the value of α, the set W will be found. A desired strictly positive value of α exists, due to the assumption that the equilibrium point belongs to the controller admissible set.

If the set W is small, then the design of a nonlinear stabilizing controller is simpler because fewer local models in the plant model can be active. Moreover, it might turn out that the linear stabilizing controller can be sufficient.

It is also worth noticing that the new set W must be found only after the change in the set-point value. If such changes can be foreseen, it is possible to calculate the sets W off-line, in advance.

4.5. Dual-mode FDMC algorithm and its stability analysis. When the set W is obtained, the following dual-mode algorithm, with the structure corresponding to that proposed in (Scokaert *et al.*, 1999), can be used:

1. Let $\mu \in (0, 1]$.

2. At time $k = 0$, the state $\overline{\boldsymbol{x}}_0$ is known. If $\overline{\boldsymbol{x}}_0 \in W$, then $u_0 = h_{\text{FDMC}}(\overline{\boldsymbol{x}}_0)$. In the opposite case, using the state $\overline{\boldsymbol{x}}_0$, the future sequence of control increments $\Delta\boldsymbol{\pi}_0 = \{\Delta u_{0|0}, \Delta u_{1|0}, \ldots, \Delta u_{s-1|0}\}$ and a corresponding sequence of states $\{\overline{\boldsymbol{x}}_0, \overline{\boldsymbol{x}}_{1|0}, \ldots, \overline{\boldsymbol{x}}_{p|0}\}$ fulfilling the assumed constraints are obtained using one of the numerical FDMC algorithms (as described in Section 3.2) solving, at each iteration, the quadratic optimization problem (4) with the stabilizing constraint (23). Then $\Delta u_0 = \Delta u_{0|0}$ is applied to the control plant.

3. At the k-th sampling instant, if $\overline{\boldsymbol{x}}_k \in W$, then $u_k = h_{\text{FDMC}}(\overline{\boldsymbol{x}}_k)$. In the opposite case, the sequence of future control increments

$$\Delta\boldsymbol{\pi}_k = \{\Delta u_{k|k}, \Delta u_{k+1|k}, \ldots, \Delta u_{k+s-1|k}\}$$

and suitable state sequence $\{\overline{\boldsymbol{x}}_k, \overline{\boldsymbol{x}}_{k+1|k}, \ldots, \overline{\boldsymbol{x}}_{k+p|k}\}$ fulfilling constraints assumed in the optimization problem (4) are obtained. Moreover, the following condition is checked:

$$\phi_k(\overline{\boldsymbol{x}}_k, \boldsymbol{\pi}_k) \le \phi_{k-1}(\overline{\boldsymbol{x}}_{k-1}\boldsymbol{\pi}_{k-1}) - \mu\left(\overline{\boldsymbol{x}}_{k-1}^T \boldsymbol{Q}\overline{\boldsymbol{x}}_{k-1} + R(u_{k-1} - u_s)^2\right). \tag{26}$$

If this condition is true, then $\Delta u_k = \Delta u_{k|k}$ is used. If not, then the next control increment is taken from the previously obtained sequence $\Delta u_k = \Delta u_{k|k-1}$. As an initial sequence for the optimization problem (4), the sequence $\boldsymbol{\pi} = \{u_{k|k-1}, u_{k+1|k-1}, \ldots, \Delta u_{k+s-2|k-1}, u_s\}$ can be used. It is an admissible sequence because it is, in fact, a continuation of the realization of the control sequence obtained in the previous step of the algorithm, and nominal stability is considered. Thus, after the application of the control sequence $\boldsymbol{\pi} = \{u_{k|k-1}, u_{k+1|k-1}, \ldots, \Delta u_{k+s-2|k-1}, u_s\}$, the constraints will be satisfied. The same will be done if in the next step the condition (26) is not fulfilled. The block diagram of the algorithm is shown in Fig. 6.

The stability of the control system with the presented algorithm is ensured by the stability of the controller working in the neighbourhood of the equilibrium point and by the decrease in the performance index enforced by the constraint (26), which implies that the quasi-state will approach the set W in finite time. The proof is analogous to that presented in (Scokaert *et al.*, 1999), and therefore the reader is referred there for detailed reasoning.

Remark 3. The theorem proved in (Scokaert *et al.*, 1999) considers nominal stability of the control system with manipulated variables generated by solving the nonlinear optimization problem (22) with a nonlinear model and was used mainly in order to cope with problems that might occur during nonlinear optimization. The suboptimal algorithm detailed in Section 4.5 is applied in order to design stable versions of the FDMC algorithms described in Section 3.2. Thanks to such an approach, nonlinear optimization is avoided and the optimization problem solved on-line by the algorithm is relaxed to the quadratic one.

5. Simulation examples

The proposed algorithms were tested in the control systems of two plants. The preliminary tests were performed in a control system of a plant given in (Setnes and Roubos, 2000). The obtained results are a good illustration of the efficacy of the proposed stabilizing mechanisms. The next experiments were performed on a control system of an ethylene distillation column DA-303 from petrochemical plant in Płock, Poland. In this case, various types of stable FDMC algorithms were tested and the control quality offered by them was compared.

5.1. Preliminary example. The discrete-time TS plant model is as follows (Setnes and Roubos, 2000):

Rule 1: if y_{k-1} is *Low* and y_{k-2} is *Low* then (27)

$$y_k^1 = 0.5402 \cdot y_{k-1} + 0.1686 \cdot y_{k-2} + u_{k-1} + 0.1413;$$

Rule 2: if y_{k-1} is *Low* and y_{k-2} is *High* then

$$y_k^2 = -0.4193 \cdot y_{k-1} + 0.1575 \cdot y_{k-2} + u_{k-1} - 0.0937;$$

Rule 3: if y_{k-1} is *High* and y_{k-2} is *Low* then

$$y_k^3 = -0.2699 \cdot y_{k-1} + 0.0890 \cdot y_{k-2} + u_{k-1} + 0.1327;$$

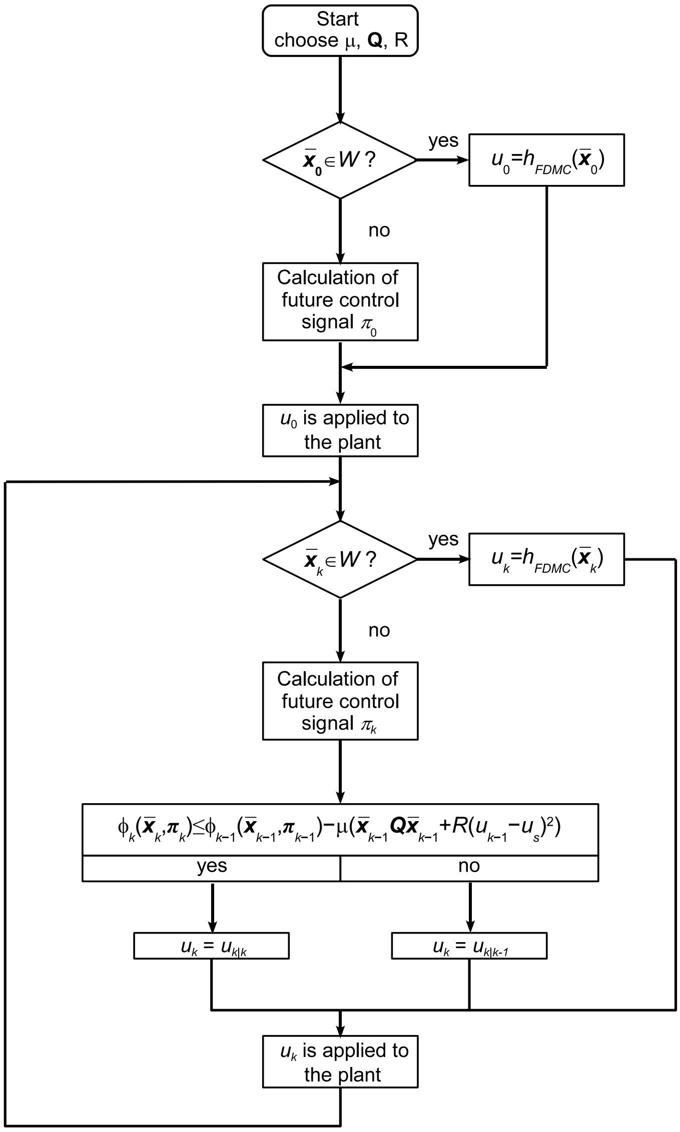


Fig. 6. Block diagram of the dual-mode algorithm.

Rule 4: if y_{k-1} is *High* and y_{k-2} is *High* then

$$y_k^4 = 0.3213 \cdot y_{k-1} + 0.0584 \cdot y_{k-2} + u_{k-1} - 0.1716,$$

where y is the output of the plant, u is the input to the plant (a manipulated variable). The parameters of the triangular membership functions are as follows (for the middle process variable value, the membership function value is equal to 1):

$$\begin{aligned} \text{— for } y_{k-1}: \quad & Low = (-4.7899, -1.4475, 1.2972), \\ & High = (-0.6048, 0.9765, 4.7889); \\ \text{— for } y_{k-2}: \quad & Low = (-3.1795, -0.6248, 0.6600), \\ & High = (-0.7942, 0.9789, 2.7312). \end{aligned}$$

As the conjunction operator the product was used as in (3). The weights were normalized. It was assumed that the manipulated variable was constrained, $-1 \leq u \leq 1$.

During the experiments the efficacy of the stabilizing mechanism was demonstrated. First, the FDMC-SL algorithm with no stabilizing mechanism was designed using the control plant model (27); $\lambda = 1$ was chosen in such a way that unstable control system behaviour was obtained (the dashed line in Fig. 7).

Next, the stabilization mechanism proposed in the paper was used. First, an analytical FDMC controller was designed for $\lambda = 40$. Then, a procedure presented in (Marusak and Tatjewski, 2001; 2002) was applied and a Lyapunov matrix of the control system with the analytical controller was found using the Matlab LMI toolbox. The analytical FDMC controller was then applied as a stabilizing controller operating near the equilibrium point.

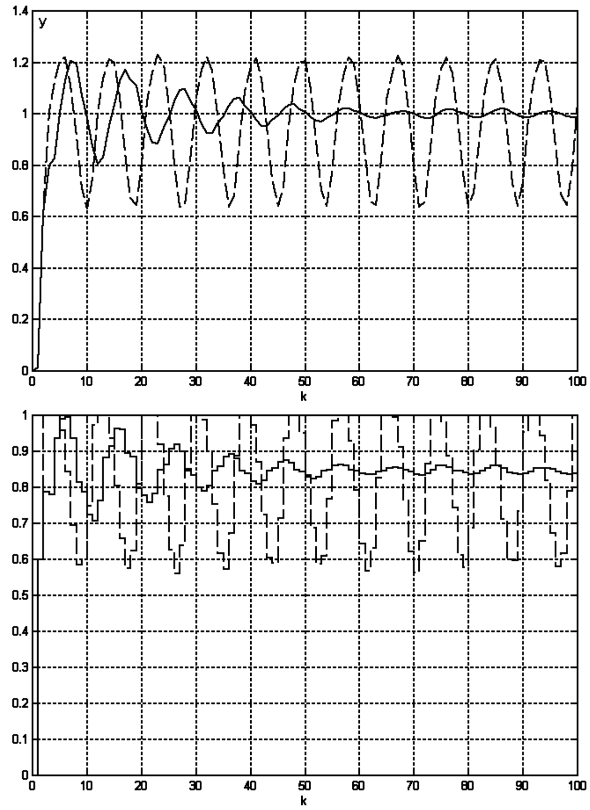


Fig. 7. Responses of control systems with FDMC-SL algorithms with (solid line) and without (dashed line) stabilizing modification to a set-point change from $y_0 = 0$ to $y^{sp} = 1$; top—output signal, bottom—control signal.

Then, the stable version of the FDMC-SL algorithm was used (with the stabilizing constraint (23) and the following values of parameters: $\mu = 1$, $R = \lambda = 1$ and $\boldsymbol{Q} = \boldsymbol{I}$). This time, the control system is stable (solid lines in Fig. 7). The obtained result illustrates the efficiency of the stabilizing mechanism introduced into the FDMC algorithm.

5.2. Control system of a distillation column.

A control plant is the ethylene distillation column DA-303 from the petrochemical plant in Płock. It is a highly nonlinear plant with a large time delay. The presented model was designed at the Institute of Control and Computation Engineering of the Warsaw University of Technology jointly with specialists from the Institute of Industrial Chemistry. It was assumed that the model has Hammerstein's structure. This means that it consists of a nonlinear static and a linear dynamic block. The structure of this model is shown in Fig. 8, where y is the product impurity counted in ppm, u denotes the reflux to the product ratio and x_f is the feed composition (time constants in Fig. 8 are given in minutes). The control plant under consideration has a large time delay and is highly nonlinear. It is well illustrated by its static characteristics shown in Fig. 9. The manipulated variable u was constrained, $4.05 \le u \le 4.4$.

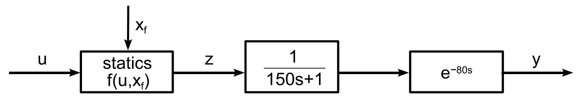


Fig. 8. Block diagram of the control plant model; u—manipulated variable, x_f—measurable disturbance, y—output variable.

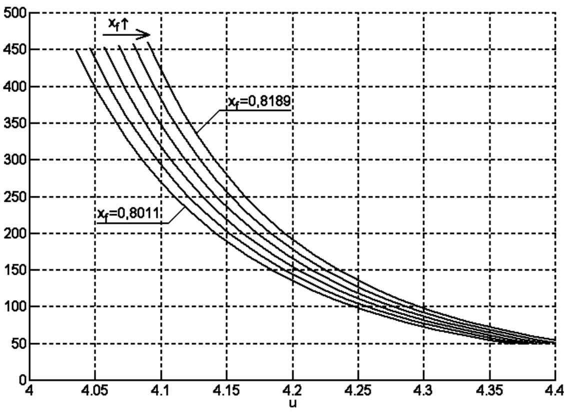


Fig. 9. Static characteristics of the control plant.

Attempts were made at designing a conventional DMC controller for the control plant; however, the controller designed to work well for large set-point values worked too slowly for smaller set-point values, and the controller designed for smaller set-point values caused a lack of stability for larger set-point values. Satisfactory results were obtained after the application of the FDMC controllers described in Section 3, see (Marusak and Tatjewski, 2000) for a detailed description of these experiments.

The discrete-time TS plant model for sampling time $T_p = 40$ minutes is as follows:

Rule 1: if u_{k-2} is Z_1, then (29)

$$y^1_{k+1} = 0.7659 \cdot y_k - 520.2638 \cdot u_{k-2} + 2220.9067;$$

Rule 2: if u_{k-2} is Z_2, then

$$y^2_{k+1} = 0.7659 \cdot y_k - 253.5771 \cdot u_{k-2} + 1102.4471;$$

Rule 3: if u_{k-2} is Z_3, then

$$y^3_{k+1} = 0.7659 \cdot y_k - 125.1030 \cdot u_{k-2} + 563.8767,$$

with membership functions shown in Fig. 10.

In order to illustrate how the proposed stabilizing mechanism can improve the control performance, the following experiment was done. The FDMC-NPL algorithm without the stabilizing mechanism was designed

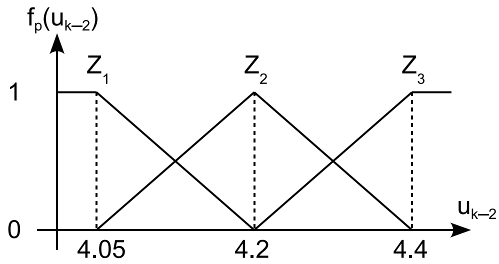


Fig. 10. Membership functions of the control plant model.

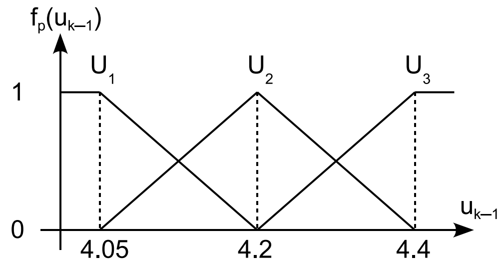


Fig. 12. Membership functions in controllers.

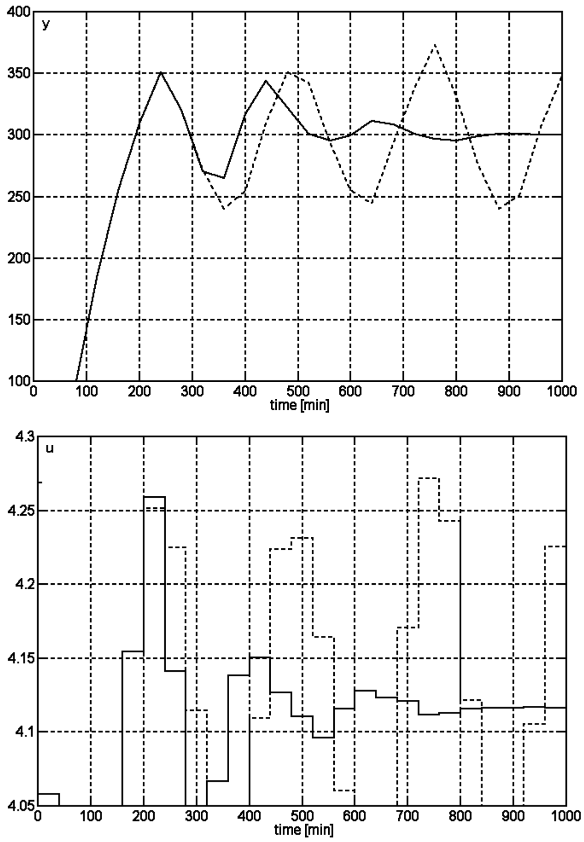


Fig. 11. Responses of control systems with FDMC-NPL algorithms with (solid line) and without (dashed line) stabilizing modification to set-point change from $y_0 =$ 100 ppm to $y^{sp} = 300$ ppm; top—output signal, bottom—control signal.

(using the discrete-time plant model) and the parameter $\lambda = 4e{+}5$ was chosen, which is a value resulting in an unstable control system response (the dashed line in Fig. 11). Then the stable FDMC-NPL algorithm was used.

First, an FDMC controller in an analytical version was designed for $\lambda = 8e + 6$. The membership functions shown in Fig. 12 were assumed. Then, utilizing the procedure presented in (Marusak and Tatjewski, 2001; 2002), the Lyapunov matrix of the control system with this controller was found using the Matlab LMI toolbox. The analytical FDMC controller was applied as a stabilizing controller operating near the equilibrium point.

The FDMC-NPL algorithm with $\mu = 10^{-10}$, $R = \lambda = 0.4{\cdot}10^6$ and $\boldsymbol{Q} = \boldsymbol{I}$ was then used in the control system. Responses obtained with the stable FDMC-NPL algorithm are marked in Fig. 11 with the solid line.

The obtained result illustrates the efficiency of the stabilizing mechanism introduced into the FDMC algorithm. The control system was stabilized. The large overshoot is implied by the value of the parameter λ assumed in order to illustrate how the proposed mechanism works. However, in the case when the controller is designed to fulfil typically assumed criteria, a different value of λ parameter should be presumed. In the discussed case, $\lambda = 0.8 \cdot 10^7$ yields satisfactory control system behaviour. That is why such a value was assumed in the analytical FDMC controller.

In Figs. 13 and 14, responses of the control systems with stable FDMC-SL and FDMC-NPL algorithms are shown; $\lambda = 0.8 \cdot 10^7$ was assumed. The responses marked with the solid line were obtained with the stable FDMC-NPL algorithm and these marked with the dashed line—with the stable FDMC-SL algorithm.

The obtained responses are practically unchanged compared to the case without stabilizing modifications except when the set-point changes from 400 ppm with the FDMC-SL algorithm. Then the algorithm had to use manipulated variable values obtained in the previous iteration in order to ensure a decrease in the performance index.

It is worth comparing the proposed algorithms with the algorithm that consists in nonlinear optimization. In Fig. 15, responses of the control system with such an algorithm are shown; the same value of $\lambda = 0.8 \cdot 10^7$ was assumed. It can be noticed that these responses are similar to those obtained in the control system with the FDMC-NPL algorithm. However, the latter are a little bit faster at the cost of a slightly bigger overshoot and control signal changes. As was already mentioned in Section 3, the application of FDMC algorithms can result in obtaining responses close to those generated in the control system with a nonlinear optimization based algorithm. However, it should be noticed that the FDMC-NPL algorithm is much more computationally efficient than the one with

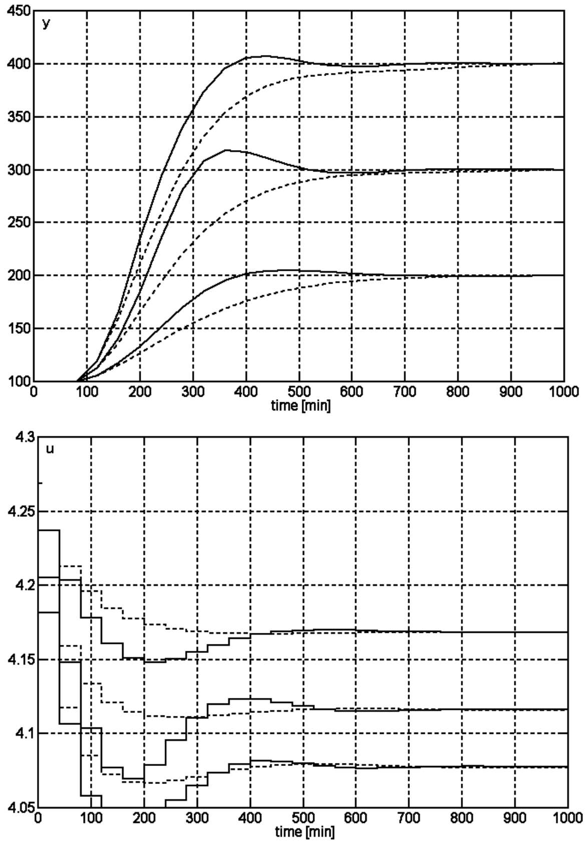


Fig. 13. Responses of control systems with FDMC-SL (dashed line) and FDMC-NPL (solid line) algorithms to set-point changes from $y_0 = 100$ ppm; top—output signal, bottom—control signal.

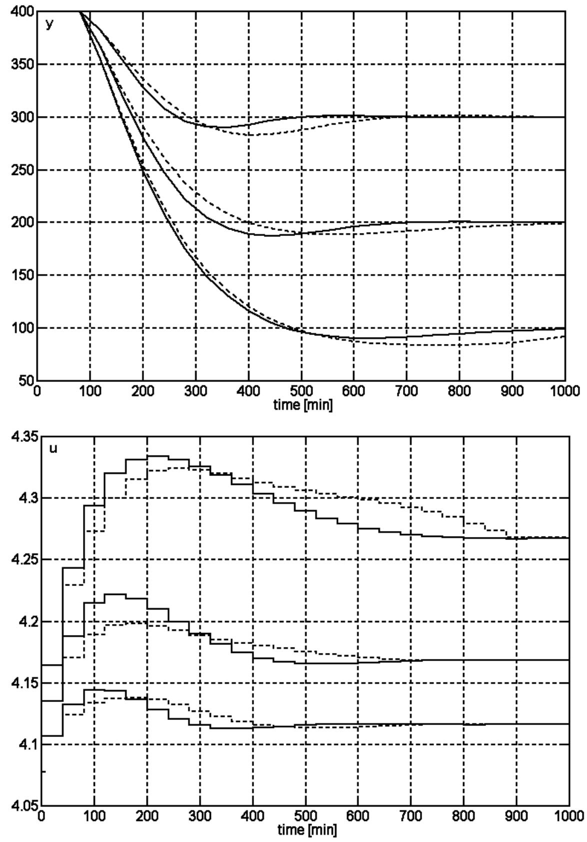


Fig. 14. Responses of control systems with FDMC-SL (dashed line) and FDMC-NPL (solid line) algorithms to set-point changes from $y_0 = 400$ ppm; top—output signal, bottom—control signal.

nonlinear optimization solved at each algorithm iteration.

6. Conclusions

Dual-mode fuzzy DMC (FDMC) algorithms with mechanisms ensuring nominal closed-loop stability have been proposed. This is one of the first results concerning nominal stability of predictive algorithms with constraints, based on nonlinear models and successive linearization, and the first one for predictive algorithms based on fuzzy models and successive linearization. The algorithms are of dual-mode type but are computationally effective, solving only quadratic optimization problems at each sampling instant. The stabilization mechanisms introduced are easy to implement and can be used in conjunction with any FDMC algorithm. Thus, the algorithm version most suitable for a given nonlinear control plant can be selected. Comparative simulation results show the efficiency of the proposed dual-mode extensions, supplementing the theoretical deliberations of the paper.

The algorithm of derivation of the invariant set W is a key issue in dual-mode type algorithms applied for output tracking problems. In the case of systems considered in the paper, the structure of the fuzzy Takagi-Sugeno model was exploited in such a way that a fast and simple algorithm for determining the invariant set W was proposed. The proposed algorithm of finding the set W relies on solving quadratic optimization problems which are formulated in such a way that they can be solved analytically (no numerical optimization is necessary), analogously to the problem of manipulated variable calculation by the explicit controller described in Section 3.3. Thus, the algorithm produces the solution very fast. This is important because after a set-point change a new set W must be found.

Acknowledgement

This work was supported by the Polish state budget funds for science.

References

Blevins, T., McMillan, G., Wojsznis, W. and Brown, M. (2003). *Advanced Control Unleashed*, ISA—The Instrumentation,

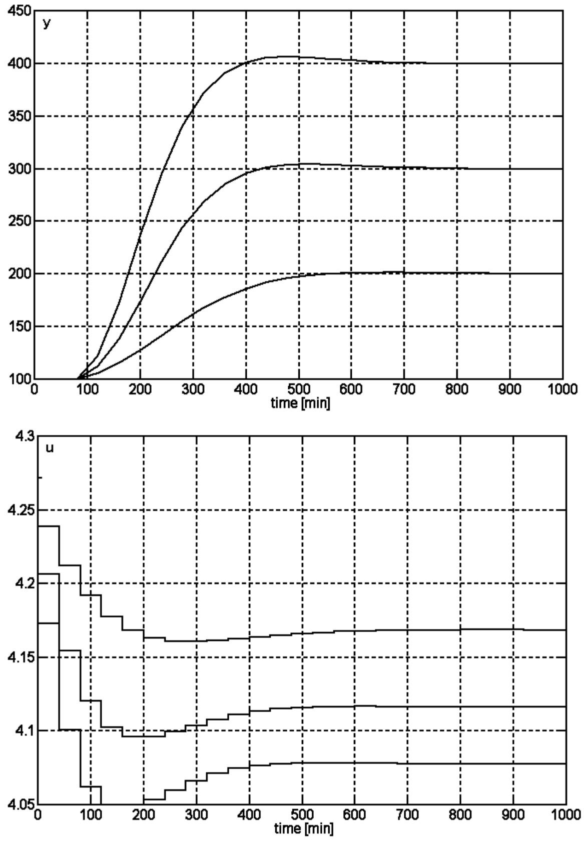


Fig. 15. Responses of control system with the algorithm with nonlinear optimization to set-point changes from $y_0 =$ 100 ppm; top—output signal, bottom—control signal.

Systems, and Automation Society, Research Triangle Park, NC.

Camacho, E. and Bordons, C. (1999). *Model Predictive Control*, Springer-Verlag, London.

Cao, S., Rees, N. and Feng, G. (1997). Analysis and design for a class of complex control systems. Part I: Fuzzy modelling and identification, *Automatica* **33**(6): 1017–1028.

Chen, J., Xi, Y. and Zhang, Z. (1998). A clustering algorithm for fuzzy model identification, *Fuzzy Sets and Systems* **98**(3): 319–329.

Cutler, C. and Ramaker, B. (1980). Dynamic matrix control—A computer control algorithm, *Proceedings of the Joint Automatic Control Conference*, San Francisco, CA, USA, paper no. WP5–B.

Driankov, D., Hellendoorn, H. and Reinfrank, M. (1993). *An Introduction to Fuzzy Control*, Springer-Verlag, Berlin.

Garcia, C. (1984). Quadratic dynamic matrix control of nonlinear processes: An application to a batch reaction process, *Proceedings of the AIChE Annual Meeting*, San Francisco, CA, USA, paper no. 82f.

Garcia, C. and Morshedi, A. (1986). Quadratic programming solution of dynamic matrix control (QDMC), *Chemical Engineering Communications* **46**(1–3): 73–87.

Gattu, G. and Zafiriou, E. (1992). Nonlinear quadratic dynamic matrix control with state estimation, *Industrial and Engineering Chemistry Research* **31**(4): 1096–1104.

Lee, J. and Ricker, N. (1994). Extended Kalman filter based nonlinear model predictive control, *Industrial and Engineering Chemistry Research* **33**(6): 1530–1541.

Li, W. and Biegler, L. (1989). Multistep, Newton-type control strategies for constrained, nonlinear processes, *Chemical Engineering Research and Design* **67**(Nov.): 562–577.

Maciejowski, J. (2002). *Predictive Control with Constraints*, Prentice Hall, Harlow.

Marusak, P. (2002). *Predictive control of nonlinear plants using dynamic matrix and fuzzy modeling*, Ph.D. thesis, Warsaw University of Technology, Warsaw, (in Polish).

Marusak, P. and Tatjewski, P. (2000). Fuzzy dynamic matrix control algorithms for nonlinear plants, *Proceedings of the 6-th International Conference on Methods and Models in Automation and Robotics MMAR 2000*, Międzyzdroje, Poland, pp. 749–754.

Marusak, P. and Tatjewski, P. (2001). Stability analysis of nonlinear control systems with fuzzy DMC controllers, *Proceedings of the IFAC Workshop on Advanced Fuzzy and Neural Control, AFNC'01*, Valencia, Spain, pp. 21–26.

Marusak, P. and Tatjewski, P. (2002). Stability analysis of nonlinear control systems with unconstrained fuzzy predictive controllers, *Archives of Control Sciences* **12**(3): 267–288.

Marusak, P. and Tatjewski, P. (2003). Stable, effective fuzzy DMC algorithms with on-line quadratic optimization, *Proceedings of the American Control Conference, ACC 2003*, Denver, CO, USA, pp. 3513–3518.

Mayne, D., Rawlings, J., Rao, C. and Scokaert, P. (2000). Constrained model predictive control: Stability and optimality, *Automatica* **36**(6): 789–814.

Michalska, H. and Mayne, D. (1993). Robust receding horizon control of constrained nonlinear systems, *IEEE Transactions on Automatic Control* **38**(11): 1623–1632.

Morari, M. and Lee, J. (1999). Model predictive control: Past, present and future, *Computers and Chemical Engineering* **23**(4): 667–682.

Mutha, R., Cluett, W. and Penlidis, A. (1997). Nonlinear model-based predictive control of control nonaffine systems, *Automatica* **33**(5): 907–913.

Mutha, R., Cluett, W. and Penlidis, A. (1998). Modifying the prediction equation for nonlinear model-based predictive control, *Automatica* **34**(10): 1283–1287.

Piegat, A. (2001). *Fuzzy Modeling and Control*, Physica-Verlag, Berlin.

Rossiter, J. (2003). *Model-Based Predictive Control*, CRC Press, Boca Raton, FL.

Scokaert, P., Mayne, D. and Rawlings, J. (1999). Suboptimal model predictive control (feasibility implies stability), *IEEE Transactions on Automatic Control* **44**(3): 648–654.

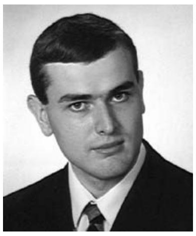
Setnes, M. and Roubos, H. (2000). GA-fuzzy modeling and classification: Complexity and performance, *IEEE Transactions on Fuzzy Systems* **8**(5): 509–522.

Takagi, T. and Sugeno, M. (1985). Fuzzy identification of systems and its application to modeling and control, *IEEE Transactions on Systems, Man and Cybernetics* **15**(1): 116–132.

Tanaka, K. and Sugeno, M. (1992). Stability analysis and design of fuzzy control systems, *Fuzzy Sets and Systems* **45**(2): 135–156.

Tatjewski, P. (2007). *Advanced Control of Industrial Processes; Structures and Algorithms*, Springer-Verlag, London.

Yager, R. and Filev, D. (1994). *Essentials of Fuzzy Modeling and Control*, Wiley, New York, NY.



Piotr Marusak was born in Warsaw, Poland, in 1974. He received the M.Sc. degree in computer science and the Ph.D. degree in automatic control from the Warsaw University of Technology in 1997 and 2003, respectively. He is with the Institute of Control and Computation Engineering of the Warsaw University of Technology. His main research interests include predictive control algorithms based on linear and nonlinear (especially fuzzy) process models, as well as the application of these algorithms in fault-tolerant control systems and the cooperation of predictive control algorithms with set-point optimization.

Piotr Tatjewski was born in Warsaw, Poland, in 1949. He received his M.Sc. (in electronic engineering), Ph.D. and D.Sc. degrees (in automatic control) from the Warsaw University of Technology in 1972, 1976 and 1998, respectively. Since 1972 he has been with the Institute of Control and Computation Engineering of the Warsaw University of Technology (1996–2008: the director of the Institute, 1993: a professor, 2006: a full professor). He is the author or a co-author of more than 30 journal papers, five books, including the recent *Advanced Control of Industrial Processes. Structures and Algorithms* (Springer, 2007) and *Iterative Algorithms of Multilayer Optimizing Control* (Imperial College Press/World Scientific, 2005, co-authored by M.A. Brdys). He is a member of the Committee on Automatic Control and Robotics of the Polish Academy of Sciences, and a member of the EUCA (European Union Control Association). His current interests involve multilayer control, predictive control of linear and nonlinear processes, set-point optimization, decomposition techniques in optimization and control, and soft computing methods.

Received: 4 September 2007
Revised: 18 March 2008