Int. J. Appl. Math. Comput. Sci., 2021, Vol. 31, No. 1, 59–71
DOI: 10.34768/amcs-2021-0005

A NUMERICALLY EFFICIENT FUZZY MPC ALGORITHM WITH FAST GENERATION OF THE CONTROL SIGNAL

PIOTR M. MARUSAK [a]

[a]Institute of Control and Computation Engineering
Warsaw University of Technology
Nowowiejska 15/19, 00-665 Warsaw, Poland
e-mail: P.Marusak@ia.pw.edu.pl

Model predictive control (MPC) algorithms are widely used in practical applications. They are usually formulated as optimization problems. If a model used for prediction is linear (or linearized on-line), then the optimization problem is a standard, i.e., quadratic, one. Otherwise, it is a nonlinear, in general, nonconvex optimization problem. In the latter case, numerical problems may occur during solving this problem, and the time needed to calculate control signals cannot be determined. Therefore, approaches based on linear or linearized models are preferred in practical applications. A novel, fuzzy, numerically efficient MPC algorithm is proposed in the paper. It can offer better performance than the algorithms based on linear models, and very close to that of the algorithms based on nonlinear optimization. Its main advantage is the short time needed to calculate the control value at each sampling instant compared with optimization-based numerical algorithms; it is a combination of analytical and numerical versions of MPC algorithms. The efficiency of the proposed approach is demonstrated using control systems of two nonlinear control plants: the first one is a chemical CSTR reactor with a van de Vusse reaction, and the second one is a pH reactor.

Keywords: model predictive control, fuzzy systems, fuzzy control, nonlinear control.

1. Introduction

MPC algorithms offer good control quality resulting from the way they are formulated. Therefore, they can be successfully used in control systems of processes with difficult dynamics and constraints, and of MIMO processes. This is because during the generation of the control signals a model of the control plant is used to predict the behavior of the control system (see, e.g., Camacho and Bordons, 1999; Karimi Pour *et al.*, 2018; Maciejowski, 2002; Rossiter, 2003; Tatjewski, 2007; 2017).

Standard formulations of the MPC algorithms are based on linear control plant models (see, e.g., Camacho and Bordons, 1999; Tatjewski, 2014). However, the operation of the control system of a nonlinear control plant may be usually improved using the MPC algorithm based on a nonlinear model. It is especially important if the MPC algorithm should operate well at different operating points as in control system structures with steady-state set-point optimization (see, e.g., Blevins *et al.*, 2003; Ławryńczuk *et al.*, 2008; Tatjewski, 2007).

If a nonlinear process model is used for prediction, then the optimization problem solved at each iteration by the algorithm may be in general a nonconvex, nonlinear optimization problem—hard to solve and with unpredictable time needed to find the solution; such algorithms will be called NMPC. One of the solutions to cope with the complexity of the optimization task is to allow a slightly suboptimal solution. This idea was proposed, in the case of MPC algorithms based on linear models, by Kouvaritakis *et al.* (2002). In the case of NPMC algorithms, fast NMPC algorithms were designed in which the solution is generated faster than in the standard approach also at the expense of suboptimality (see, e.g., Diehl *et al.*, 2002; Schäfer *et al.*, 2007; Zavala *et al.*, 2008). Another approach, described, e.g., by Dominguez and Pistikopoulos (2010) or Johansen (2002; 2004), is the explicit one, in which most calculations are done off-line. Furthermore, this approach is optimal in the case when a linear model is used (Bemporad *et al.*, 2000; 2002; Pistikopoulos *et al.*, 2000). Unfortunately, an important disadvantage of the approach is that the complexity of the controller grows significantly with the number of inequality constraints included in the problem.

When a fuzzy Takagi–Sugeno model is used in the MPC algorithm, then obviously nonlinear optimization can be applied. The structure of the fuzzy model can be, however, exploited to formulate algorithms which are easier to solve than the standard NMPC. The first group of such algorithms includes classical fuzzy Takagi–Sugeno controllers composed of a few linear controllers which are switched in a fuzzy way (see, e.g., Killian and Kozek, 2017; Marusak and Tatjewski, 2002). Other algorithms, exploiting the structure of fuzzy Takagi–Sugeno models, are based on linear matrix inequalities (LMIs). An interesting review of algorithms using this technique is given by Guerra *et al*. (2009), and recently proposed algorithms can be found in the works of Khooban *et al*. (2016), Kong and Yuan (2019a; 2019b), Shen *et al*. (2020) or Wu *et al*. (2015).

In order to avoid the complexity of NMPC, often on-line linearization at each algorithm iteration is used (see, e.g., Boulkaibet *et al*., 2017; Essien *et al*., 2019; Ławryńczuk, 2014; 2015; 2020; Marusak, 2009a; 2009b; Morari and Lee, 1999; Tatjewski, 2007). After a linear approximation of the nonlinear model is obtained, a prediction linear to decision variables is acquired. Therefore, the optimization problem solved at each iteration of the algorithm is a quadratic one. The prediction can be based on the classical linearization or it can exploit the structure of the model the algorithm is based on. Examples of algorithms using linearization of the fuzzy Takagi–Sugeno model, in which both the free response and the dynamic matrix are obtained using the linearized model, are given by Boulkaibet *et al*. (2017) and Marusak (2009a). Algorithms in which the free response is calculated using the nonlinear model and the model obtained after linearization is used to obtain the dynamic matrix can be found in the works of Essien *et al*. (2019) and Marusak (2009b).

The computational efficiency of MPC algorithms is important when they are implemented in micro-controllers, FPGAs or PLCs. The computational power of these devices achieved such a level that they allow to implement analytical versions of MPC algorithms. An implementation of the DMC algorithm runing on an FPGA and on a PLC was described recently by Wojtulewicz and Ławryńczuk (2018b; 2018a); implementation of the GPC controller for the STM32 ARM micro-controller is detailed by Chaber and Ławryńczuk (2019). Both DMC and GPC algorithms are based on linear models. However, in many cases control performance can be improved after using an MPC algorithm based on a nonlinear (e.g., fuzzy) model. Therefore, fast MPC algorithms based on fuzzy models are desirable. The present paper addresses this issue.

The approach proposed in this work consists in using two models in the algorithm: a nonlinear model and its (easy to obtain) fuzzy approximation. The algorithm is a combination of analytical and numerical versions of MPC algorithms. Thanks to such an approach, the time needed to calculate the control value at each sampling instant is much smaller compared with numerical algorithms. However, thanks to skillful use of nonlinear models, the algorithm offers almost the same performance as that with nonlinear optimization.

The next section contains a description of MPC algorithms. In Section 3 the proposed approach based on fuzzy and nonlinear models is detailed. Example results illustrating the efficacy of the proposed approach are presented in Section 4. The paper is summarized in the last section.

2. Model predictive control algorithms

Model predictive control (MPC) algorithms derive future values of manipulated variables predicting the behavior of the control plant many sampling instants ahead. The values of manipulated variables are calculated in such a way that the prediction fulfills assumed criteria. Usually, the minimization of a performance index is demanded subject to the constraints put on values of manipulated and of output variables (Camacho and Bordons, 1999; Maciejowski, 2002; Rossiter, 2003; Tatjewski, 2007):

$$\arg\min_{\Delta \boldsymbol{u}} \sum_{j=1}^{n_y}\sum_{i=1}^{p} \kappa_j \left(\overline{y}^j_{k+i|k} - y^j_{k+i|k}\right)^2 + \sum_{m=1}^{n_u}\sum_{i=0}^{s} \lambda_m \left(\Delta u^m_{k+i|k}\right)^2 \tag{1}$$

subject to the constraints

$$\Delta \boldsymbol{u}_{\min} \le \Delta \boldsymbol{u} \le \Delta \boldsymbol{u}_{\max}, \tag{2}$$

$$\boldsymbol{u}_{\min} \le \boldsymbol{u} \le \boldsymbol{u}_{\max}, \tag{3}$$

$$\boldsymbol{y}_{\min} \le \boldsymbol{y} \le \boldsymbol{y}_{\max}, \tag{4}$$

where $y^j_{k+i|k}$ is the value of the j-th output for the $(k+i)$-th sampling instant predicted at the k-th sampling instant using a control plant model, $\overline{y}^j_{k+i|k}$ are elements of the reference trajectory for the j-th output, $\Delta u^m_{k+i|k}$ are future changes in manipulated variables, $\kappa_j \ge 0$ and $\lambda_m \ge 0$ are weighting coefficients for the predicted control errors of the j-th output and for the changes in the m-th manipulated variable, respectively; p and s denote prediction and control horizons, respectively; n_y, n_u denote a number of output and manipulated variables, respectively;

$$\boldsymbol{y} = \left[\boldsymbol{y}_k^1, \boldsymbol{y}_k^2, \ldots, \boldsymbol{y}_k^{n_y}\right]^T,$$
$$\boldsymbol{y}_k^j = \left[y_{k+1|k}^j, \ldots, y_{k+p|k}^j\right],$$
$$\Delta\boldsymbol{u} = \left[\Delta\boldsymbol{u}_k^1, \Delta\boldsymbol{u}_k^2, \ldots, \Delta\boldsymbol{u}_k^{n_u}\right]^T,$$
$$\Delta\boldsymbol{u}_k^m = \left[\Delta u_{k|k}^m, \ldots, \Delta u_{k+s-1|k}^m\right],$$
$$\boldsymbol{u} = \left[\boldsymbol{u}_k^1, \boldsymbol{u}_k^2, \ldots, \boldsymbol{u}_k^{n_u}\right]^T,$$
$$\boldsymbol{u}_k^m = \left[u_{k|k}^m, \ldots, u_{k+s-1|k}^m\right],$$

$\boldsymbol{u}_{\min}$, $\boldsymbol{u}_{\max}$ and $\Delta\boldsymbol{u}_{\min}$, $\Delta\boldsymbol{u}_{\max}$ are vectors of lower and upper bounds on the values and on the changes of the manipulated variables, and $\boldsymbol{y}_{\min}$, $\boldsymbol{y}_{\max}$ are the vectors of lower and upper bounds on the values of the output variables. As a solution to the optimization problem (1)–(4) an optimal vector of changes in the manipulated variables is obtained. From this vector, the $\Delta u_{k|k}^m$ elements, corresponding to the current sampling instant, are applied in the control system. The optimization problem (1)–(4) is solved at each sampling instant.

The way the predicted values of output variables $y_{k+i|k}^j$ are derived depends on the dynamic control plant model the predictive algorithm is based on. If a nonlinear process model is used, then the optimization problem (1)–(4) is, in general, a nonconvex nonlinear optimization problem instead of a linear-quadratic one. In such an algorithm, different kinds of process models can be used but they are exploited in a similar way; therefore an algorithm of this kind will be later referred to, in general, as nonlinear MPC (NMPC). In the NMPC algorithm the computational burden needed to solve such a problem can be prohibitive, making practical implementation of the predictive algorithm difficult and unreliable. On the other hand, if the linear model is used, then the problem (1)–(4) is a standard quadratic programming one. Unfortunately, using the algorithm based on a linear plant model for the nonlinear plant may be insufficient if control at a wide range of set-points is needed. A possible solution to these difficulties could be to use algorithms based on an on-line linearization approach; see, e.g., Tatjewski (2007), a survey paper by Mayne *et al*. (2000) and the references therein.

The performance index (1) can be expresssed by

$$J_{\text{MPC}} = \left(\overline{\boldsymbol{y}} - \boldsymbol{y}\right)^T \cdot \boldsymbol{\kappa} \cdot \left(\overline{\boldsymbol{y}} - \boldsymbol{y}\right) + \Delta\boldsymbol{u}^T \cdot \boldsymbol{\lambda} \cdot \Delta\boldsymbol{u}, \quad (5)$$

where

$$\overline{\boldsymbol{y}} = \left[\overline{\boldsymbol{y}}_k^1, \overline{\boldsymbol{y}}_k^2, \ldots, \overline{\boldsymbol{y}}_k^{n_y}\right]^T$$

and every vector $\overline{\boldsymbol{y}}_k^j = \left[\overline{y}_{k+1|k}^j, \ldots, \overline{y}_{k+p|k}^j\right]$ is of length p, $\boldsymbol{\kappa} = \left[\boldsymbol{\kappa}_1, \ldots, \boldsymbol{\kappa}_{n_y}\right] \cdot \boldsymbol{I}$, every vector $\boldsymbol{\kappa}_j = [\kappa_j, \ldots, \kappa_j]$ is of length p (and the matrix $\boldsymbol{I}$ is of dimension $p \cdot n_y \times p \cdot n_y$), $\boldsymbol{\lambda} = [\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_{n_u}] \cdot \boldsymbol{I}$, every vector $\boldsymbol{\lambda}_j = [\lambda_j, \ldots, \lambda_j]$ is of length s (and the matrix $\boldsymbol{I}$ is of dimension $s \cdot n_u \times s \cdot n_u$).

2.1. MPC algorithms based on linear models (LMPC). If the prediction is obtained using a linear process model, then the optimization problem (1)–(4) is a standard quadratic programming one (Camacho and Bordons, 1999; Maciejowski, 2002; Rossiter, 2003; Tatjewski, 2007). This is because the superposition principle applies and the vector of predicted output values $\boldsymbol{y}$ can be described by the following formula:

$$\boldsymbol{y} = \widetilde{\boldsymbol{y}} + \boldsymbol{A} \cdot \Delta\boldsymbol{u}, \quad (6)$$

where

$$\widetilde{\boldsymbol{y}} = \left[\widetilde{\boldsymbol{y}}_k^1, \widetilde{\boldsymbol{y}}_k^2, \ldots, \widetilde{\boldsymbol{y}}_k^{n_y}\right]^T,$$

$$\widetilde{\boldsymbol{y}}_k^j = \left[\widetilde{y}_{k+1|k}^j, \ldots, \widetilde{y}_{k+p|k}^j\right]$$

is called a free response of the control plant, because it contains future values of output variables calculated assuming that the control signals do not change in the prediction horizon; $\boldsymbol{A}$ is called the dynamic matrix, composed of the step response coefficients. It can be shown that the dynamic matrix has the same form in different types of LMPC algorithms using different types of linear models (Tatjewski, 2007).

After applying the prediction (6), the performance index (5) can be transformed to

$$J_{\text{LMPC}} = \left(\overline{\boldsymbol{y}} - \widetilde{\boldsymbol{y}} - \boldsymbol{A} \cdot \Delta\boldsymbol{u}\right)^T \cdot \boldsymbol{\kappa} \cdot \left(\overline{\boldsymbol{y}} - \widetilde{\boldsymbol{y}} - \boldsymbol{A} \cdot \Delta\boldsymbol{u}\right) + \Delta\boldsymbol{u}^T \cdot \boldsymbol{\lambda} \cdot \Delta\boldsymbol{u}. \quad (7)$$

The performance index (7) depends quadratically on decision variables $\Delta\boldsymbol{u}$; if it is minimized without taking constraints into consideration, then the following analytical solution is obtained:

$$\Delta\boldsymbol{u} = \boldsymbol{K} \cdot \left(\overline{\boldsymbol{y}} - \widetilde{\boldsymbol{y}}\right), \quad (8)$$

where the matrix

$$\boldsymbol{K} = \left(\boldsymbol{A}^T \cdot \boldsymbol{\kappa} \cdot \boldsymbol{A} + \boldsymbol{\lambda}\right)^{-1} \cdot \boldsymbol{A}^T \cdot \boldsymbol{\kappa} \quad (9)$$

can be calculated only once, off-line, because the elements of the dynamic matrix remain constant. If only control changes for the current iteration are of interest, then (8) can be simplified:

$$\Delta u_{k|k}^m = \boldsymbol{K}_{(m-1)\cdot s+1} \cdot \left(\overline{\boldsymbol{y}} - \widetilde{\boldsymbol{y}}\right), \quad (10)$$

where $\boldsymbol{K}_{(m-1)\cdot s+1}$ is the $((m-1)\cdot s+1)$-th row of the matrix $\boldsymbol{K}$.

3. Efficient fuzzy MPC algorithm

Application of the LMPC algorithm to a nonlinear process may result in unsatisfactory control performance, especially if operation at different operating points is demanded. However, a skillful usage of a fuzzy model for prediction allows improving the control quality of highly nonlinear processes. An algorithm which offers control quality comparable with the one guaranteed by the NMPC algorithm is the FMPC algorithm, in which the free response is calculated using a nonlinear process model. This FMPC algorithm is detailed by Marusak (2009b) and is a numerical, optimization-based one.

The algorithm proposed in this section is a combination of analytical LMPC, described in Section 2.1, and of the FMPC algorithm by Marusak (2009b). Thanks to such an approach it generates the control signal fast and, at the same time, offers control quality close to the one guaranteed by NMPC algorithms. This is because in the proposed numerically efficient FMPC (EFMPC) algorithm solving the optimization problem in each iteration is avoided; the control action is generated in a much simpler way.

In the EFMPC algorithm two models are used. The original, nonlinear one is used to calculate the free response, like in the FMPC algorithm, whereas the fuzzy model, with step responses used as local models, is employed in a skillful generation of the control law. The proposed EFMPC algorithm will be now described.

Suppose that one has a nonlinear process model which generates the output values

$$\widehat{\boldsymbol{y}}_{k+1|k} = \boldsymbol{f}(\boldsymbol{y}_k, \boldsymbol{y}_{k-1}, \ldots, \boldsymbol{y}_{k-n_a}, \boldsymbol{u}_{k-1}, \boldsymbol{u}_{k-2}, \ldots, \boldsymbol{u}_{k-n_b}), \tag{11}$$

where $\boldsymbol{y}_{k-i} = \left[y^1_{k-i}, \ldots, y^{n_y}_{k-i}\right]^T$ is the vector of the measured values of the output variables at the $(k-i)$-th sampling instant, $\boldsymbol{u}_{k-i} = \left[u^1_{k-i}, \ldots, u^{n_u}_{k-i}\right]^T$ is the vector of the values of manipulated variables at the $(k-i)$-th sampling instant; let us also denote the outputs of the model at the $(k+i)$-th sampling instant as $\widehat{\boldsymbol{y}}_{k+i|k} = \left[\widehat{y}^1_{k+i|k}, \ldots, \widehat{y}^{n_y}_{k+i|k}\right]^T$, while n_a, n_b determine how many past output and control values are used by the model.

3.1. Generation of the free response.

The model (11) is then employed to obtain the free response, like in the work of Marusak (2009b), for the whole prediction horizon, iteratively, i.e.,

- First, the process model is used to obtain $\widehat{\boldsymbol{y}}_{k+1|k}$ (formula (11)).
- Then the values $\widehat{\boldsymbol{y}}_{k+1|k}$ are used as the output values for the $(k+1)$-th sampling instant, to obtain the output values for the next sampling instant $\widehat{\boldsymbol{y}}_{k+2|k}$. Moreover, the assumption that the control signal does not change (the free response is calculated) is utilized:

$$\widehat{\boldsymbol{y}}_{k+2|k} = \boldsymbol{f}(\widehat{\boldsymbol{y}}_{k+1|k}, \boldsymbol{y}_k, \ldots, \boldsymbol{y}_{k-n_a+1}, \boldsymbol{u}_{k-1}, \boldsymbol{u}_{k-1}, \ldots, \boldsymbol{u}_{k-n_b+1}); \tag{12}$$

- thus, in general, in the i-th step, using the values $\widehat{\boldsymbol{y}}_{k+1|k}, \ldots, \widehat{\boldsymbol{y}}_{k+i-1|k}$ and assuming that the control signal does not change, one obtains

$$\widehat{\boldsymbol{y}}_{k+i|k} = \boldsymbol{f}(\widehat{\boldsymbol{y}}_{k+i-1|k}, \widehat{\boldsymbol{y}}_{k+i-2|k}, \ldots, \boldsymbol{y}_{k-n_a+i-1}, \boldsymbol{u}_{k-1}, \boldsymbol{u}_{k-1}, \ldots, \boldsymbol{u}_{k-n_b+i-1}). \tag{13}$$

- Then the free response is calculated taking into consideration the estimated disturbances (containing also the influence of modeling errors). The final formula describing the elements of the free response is then as follows:

$$\widetilde{\boldsymbol{y}}_{k+i|k} = \widehat{\boldsymbol{y}}_{k+i|k} + \boldsymbol{d}_k \ , \tag{14}$$

where $\widetilde{\boldsymbol{y}}_{k+i|k} = \left[\widetilde{y}^1_{k+i|k}, \ldots, \widetilde{y}^{n_y}_{k+i|k}\right]^T$ and $\boldsymbol{d}_k$ is the DMC-type disturbance model, i.e., it is assumed the same for all instants in the prediction horizon and

$$\boldsymbol{d}_k = \boldsymbol{y}_k - \widehat{\boldsymbol{y}}_{k|k-1}. \tag{15}$$

3.2. Formulation of the control law.

In order to simplify the algorithm and make the calculation of the control signal faster, the latter part of the algorithm is done in a way described below. Assume that a Takagi–Sugeno fuzzy model composed of local models in the form of step responses was obtained:

$$\begin{aligned}
&\text{Rule}f: \\
&\text{if } y^j_{k-1} \text{ is } B^{f,j}_1 \text{ and } \ldots \text{ and } y^j_{k-i} \text{ is } B^{f,j}_i \text{ and} \\
&\quad u^m_{k-1} \text{ is } C^{f,m}_1 \text{ and } \ldots \text{ and } u^m_{k-i} \text{ is } C^{f,m}_i \\
&\text{then } \hat{y}^{j,f}_k = \sum_{m=1}^{n_u}\sum_{i=1}^{p-1} a^{j,m,f}_i \cdot \Delta u^m_{k-i} + a^{j,m,f}_p \cdot u^m_{k-p},
\end{aligned} \tag{16}$$

where y^j_k is the j-th output variable value at the k-th sampling instant, u^m_k is the m-th manipulated variable value at the k-th sampling instant, $B^{f,j}_1, \ldots, B^{f,j}_i$, $C^{f,m}_1, \ldots, C^{f,m}_i$ are fuzzy sets, $a^{j,m,f}_i$ are the coefficients of step responses in the f-th local model, $j = 1, \ldots, n_y$, $m = 1, \ldots, n_u$, $f = 1, \ldots, l$; here l is the number of rules.

The design process of such a model is rather simple. It is sufficient to collect a few sets of step responses (around a few operating points). Then, using expert knowledge, the premises can be formulated and, subsequently, they can be tuned using, e.g., a fuzzy neural network.

Now, for each local model a dynamic matrix is obtained:

$$\boldsymbol{A}^f = \begin{bmatrix} \boldsymbol{A}^f_{11} & \boldsymbol{A}^f_{12} & \dots & \boldsymbol{A}^f_{1n_u} \\ \boldsymbol{A}^f_{21} & \boldsymbol{A}^f_{22} & \dots & \boldsymbol{A}^f_{2n_u} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{A}^f_{n_y1} & \boldsymbol{A}^f_{n_y2} & \dots & \boldsymbol{A}^f_{n_yn_u} \end{bmatrix}, \tag{17}$$

$$\boldsymbol{A}^f_{jm} = \begin{bmatrix} a_1^{j,m,f} & 0 & \dots & 0 & 0 \\ a_2^{j,m,f} & a_1^{j,m,f} & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_p^{j,m,f} & a_{p-1}^{j,m,f} & \dots & a_{p-s+2}^{j,m,f} & a_{p-s+1}^{j,m,f} \end{bmatrix}. \tag{18}$$

Next, for each dynamic matrix $\boldsymbol{A}^f$ the appropriate matrix $\boldsymbol{K}^f$ is calculated using the formula (cf. (9))

$$\boldsymbol{K}^f = \left(\boldsymbol{A^f}^T \cdot \boldsymbol{\kappa} \cdot \boldsymbol{A^f} + \boldsymbol{\lambda}\right)^{-1} \cdot \boldsymbol{A^f}^T \cdot \boldsymbol{\kappa}. \tag{19}$$

Each matrix $\boldsymbol{K}^f$ can be calculated off-line. Thus, the following fuzzy controller can be obtained (compare the local controllers with (8)):

$$\begin{aligned} &\text{Rule} f: \qquad (20)\\ &\text{if } y^j_{k-1} \text{ is } B^{f,j}_1 \text{ and } \dots \text{ and } y^j_{k-i} \text{ is } B^{f,j}_i \text{ and}\\ &\quad u^m_{k-1} \text{ is } C^{f,m}_1 \text{ and } \dots \text{ and } u^m_{k-i} \text{ is } C^{f,m}_i\\ &\text{then } \Delta \boldsymbol{u}^f = \boldsymbol{K}^f \cdot (\overline{\boldsymbol{y}} - \widetilde{\boldsymbol{y}}) . \end{aligned}$$

If only control changes for the current iteration are of interest, then (20) can be simplified:

$$\begin{aligned} &\text{Rule} f: \qquad (21)\\ &\text{if } y^j_{k-1} \text{ is } B^{f,j}_1 \text{ and } \dots \text{ and } y^j_{k-i} \text{ is } B^{f,j}_i \text{ and}\\ &\quad u^m_{k-1} \text{ is } C^{f,m}_1 \text{ and } \dots \text{ and } u^m_{k-i} \text{ is } C^{f,m}_i\\ &\text{then } \Delta u^{f,m}_{k|k} = \boldsymbol{K}^f_{(m-1)\cdot s+1} \cdot (\overline{\boldsymbol{y}} - \widetilde{\boldsymbol{y}}) , \end{aligned}$$

where $\boldsymbol{K}^f_{(m-1)\cdot s+1}$ are the $((m-1)\cdot s+1)$-th rows of the matrices $\boldsymbol{K}^f$. The output of the fuzzy controller (21) is given by

$$\Delta u^m_{k|k} = \sum_{f=1}^{l} \widetilde{w}^f \cdot \boldsymbol{K}^f_{(m-1)\cdot s+1} \cdot (\overline{\boldsymbol{y}} - \widetilde{\boldsymbol{y}}) , \tag{22}$$

where $\widetilde{w}^f$ is the normalized firing strength of the f-th rule, obtained using fuzzy reasoning.

3.3. Taking constraints into consideration. In order to take control constraints into consideration in the proposed algorithm, the control projection onto the constraint set can be applied (see, e.g., Tatjewski, 2007). The mechanism consists in application of the following rules of $\Delta u^m_{k|k}$ modification, leading to the fulfillment of the constraints on control changes:

- for lower bound constraints:
  —if $\Delta u^m_{k|k} < \Delta u^m_{\min}$, then $\Delta u^m_{k|k} = \Delta u^m_{\min}$,
- for upper bound constraints:
  —if $\Delta u^m_{k|k} > \Delta u^m_{\max}$, then $\Delta u^m_{k|k} = \Delta u^m_{\max}$;

and leading to fulfillment of the constraints on control values:

- for lower bound constraints:
  —if $u^m_{k-1} + \Delta u^m_{k|k} < u^m_{\min}$, then
  $$\Delta u^m_{k|k} = u^m_{\min} - u^m_{k-1},$$
- for upper bound constraints:
  —if $u^m_{k-1} + \Delta u^m_{k|k} > u^m_{\max}$, then
  $$\Delta u^m_{k|k} = u^m_{\max} - u^m_{k-1}.$$

Using this mechanism it is also possible to take constraints on predicted output values into consideration; for details see the work of Marusak (2010).

3.4. Properties of the algorithm. The diagram of the algorithm is depicted in Fig. 1. Many actions, also the most computationally demanding calculation of the matrices $\boldsymbol{K}^f$ (using (19)), are done off-line; these include the following:

I. Development of the fuzzy model (16).

II. Construction of the dynamic matrices $\boldsymbol{A}^f$ from (17), for each of the local models from (16).

III. Calculation of the matrices $\boldsymbol{K}^f$ using (19), for each dynamic matrix obtained in the previous step.

IV. Construction of the fuzzy controller (21) using appropriate elements of the $\boldsymbol{K}^f$ matrices.

Thanks to the proposed approach, in each iteration of the EFDMC algorithm (in each sampling instant) only the following actions should be done (see Fig. 1):

1. The nonlinear control plant model is used to generate the free response (14) of the control plant.
2. For each fuzzy rule from the model (16) the value of $\widetilde{w}^f$, the normalized firing strength, is calculated.
3. The obtained free response $\widetilde{\boldsymbol{y}}$, the firing strengths $\widetilde{w}^f$ and the appropriate rows of the matrices $\boldsymbol{K}^f$ are used in the control law (22) to calculate the control signals. No optimization routines are used.

The proposed algorithm usually gives better results than the standard LMPC one and generates results very close to those obtained with a numerical, optimization-based FMPC algorithm, in which the free response is generated the same way. However, it needs much fewer computations to obtain the control values than its counterpart, which uses quadratic optimization.

The proposed algorithm, like other algorithms based on linearization, is an approximate one compared with

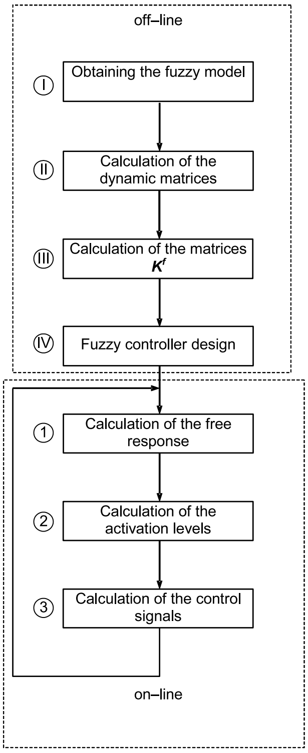

Fig. 1. Diagram of the EFMPC algorithm.

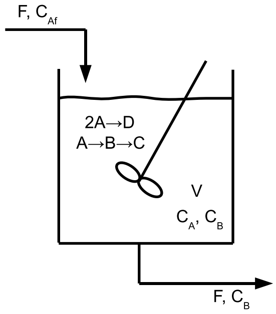

Fig. 2. Diagram of the isothermal CSTR with a van de Vusse reaction.

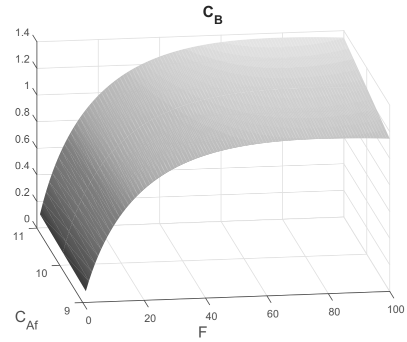

Fig. 3. Steady-state characteristic of the control plant.

an NMPC algorithm. However, in the example control systems it will be demonstrated that both algorithms can offer comparable results. Moreover, the algorithm can be used not only as a stand-alone one, but also as support for NMPC algorithms to improve their numerical properties. In the latter case the numerically efficient algorithm can be employed to generate the starting control trajectory for a nonlinear optimization routine, using (20). Next, the NMPC algorithm tries to improve the initial trajectory. If it manages to do so during one sampling instant, then the newly derived control action is applied to the process. If it fails to improve the approximate solution, then the control signal generated by the proposed algorithm can be used, as it offers performance very close to the optimal one, anyway.

Remark 1. The model given by (16) is obtained from the model (11). The membership functions can be chosen by a designer after analyzing the shape of the steady-state characteristic of the process. It is done in the examples detailed in the next section. Such an approach is relatively simple and allows obtaining the EFMPC controller offering better control performance than the LMPC one, fast. However, parameters of the membership functions can be also tuned using optimization.

Remark 2. The premises of the obtained controller (21) can be changed if needed to improve the control system performance. This can be done either by using a trial-and-error method or by optimizing the parameters of the membership functions. In the case of the local controllers the main tuning parameters are κ_j and λ_m. An increase in λ_m in relation to κ_j increases robustness of the controller but, at the same time, slows it down. In the MIMO case, if one needs to improve stabilization of the j-th output, then κ_j should be increased. More advanced methods of tuning allow varying the values of κ_j and λ_m at different instants of the prediction horizon. Such an approach allows obtaining interesting results but demands significant effort during the tuning process; for details, see the work of Nebeluk and Marusak (2020).

4. Simulation experiments

4.1. SISO control plant. The control plant under consideration is an isothermal CSTR in which a van de Vusse reaction is carried out (Fig. 2). The steady-state characteristic of the control plant is shown in Fig. 3.

The process model of the reactor contains two composition balance equations (Doyle *et al.*, 1995):

$$\begin{aligned} \frac{\mathrm{d}C_\mathrm{A}}{\mathrm{d}t} &= -k_1 \cdot C_\mathrm{A} - k_3 \cdot C_\mathrm{A}^2 + \frac{F}{V}\left(C_\mathrm{Af} - C_\mathrm{A}\right), \\ \frac{\mathrm{d}C_\mathrm{B}}{\mathrm{d}t} &= k_1 \cdot C_\mathrm{A} - k_2 \cdot C_\mathrm{B} - \frac{F}{V} C_\mathrm{B}, \end{aligned} \tag{23}$$

where C_{A}, C_{B} are the concentrations of components A and B, respectively, F is the inlet flow rate (equal to the outlet flow rate; it is assumed that it is constrainted, and $F_{\min} = 0\,\mathrm{l/h}$, $F_{\max} = 60\,\mathrm{l/h}$), V is the volume in which the reaction takes place (it is assumed constant and $V = 1$ l), C_{Af} is the concentration of component A in the inlet flow stream (it is assumed that $C_{\mathrm{Af}} = 10$ mol/l). The values of the parameters are $k_1 = 50$ 1/h, $k_2 = 100$ 1/h, $k_3 = 10\,\mathrm{l/(h \cdot mol)}$.

The output variable is the concentration C_{B} of substance B, the manipulated variable is the inlet flow rate F of the raw substance, C_{Af} concentration is the disturbance variable.

The fuzzy model is composed of three step responses, with membership functions shown in Fig. 4. The step responses were obtained in a vicinity of the following operating points (Marusak, 2009a):

1. $C_{\mathrm{B0}} = 0.91\,\mathrm{mol/l}$, $C_{\mathrm{A0}} = 2.18\,\mathrm{mol/l}$, $F = 20\,\mathrm{l/h}$;
2. $C_{\mathrm{B0}} = 1.12\,\mathrm{mol/l}$, $C_{\mathrm{A0}} = 3\,\mathrm{mol/l}$, $F = 34.3\,\mathrm{l/h}$;
3. $C_{\mathrm{B0}} = 1.22\,\mathrm{mol/l}$, $C_{\mathrm{A0}} = 3.66\,\mathrm{mol/l}$, $F = 50\,\mathrm{l/h}$.

Operation of the proposed EFMPC algorithm will be compared with that of three other MPC algorithms: an NMPC one with nonlinear optimization, an LMPC one with a linear model and the optimization-based FMPC by Marusak (2009b). The sampling time was assumed equal to $T_{\mathrm{s}} = 3.6$ s; tuning parameters of all three algorithms were as follows: prediction horizon $p = 70$, control horizon $s = 35$, $\lambda_1 = 0.001$ and $\kappa_1 = 1$. Responses generated with the predictive algorithms will be also compared with the ones obtained with the PID controller proposed by Krishna *et al.* (2012) (called SA-PID) and considered the best there.

The responses obtained after changes in the set-point value are shown in Figs. 5 and 6. In the case when the set-point was changed to $\overline{C}_B = 1.25$ (Fig. 5), the responses obtained in the control system with the proposed EFMPC algorithm (solid lines) are almost the same as those generated with the optimization-based FMPC algorithm (dotted lines). Despite significant simplification of calculations, the obtained responses are practically the same. The responses obtained with FMPC algorithms are faster than those obtained with the NMPC algorithm (dashed lines). All the algorithms based on nonlinear models outperform the standard LMPC algorithm (dash-dotted lines), which works very slow—the control time is much longer than in the case of other algorithms. The PID controller is the slowest among all tested algorithms. In all cases there is no overshoot.

The responses obtained after the change in the set-point value to $\overline{C}_B = 0.8$ are shown in Fig. 6. In this experiment the responses obtained in the control system with the EFMPC algorithm (solid lines) are also very close to those obtained with optimization-based FMPC (dotted lines). Moreover, they are also similar to the responses generated with the NMPC algorithm (dashed lines). In all three cases there is practically no overshoot; this time the NMPC algorithm is slightly faster than its fuzzy counterparts. All the algorithms based on nonlinear models outperform the standard LMPC (dotted lines) and PID (dashed lines with points) algorithms. The LMPC algorithm generates significant overshoot and control time is longer than in the case of other algorithms. In this experiment it is the worst among all tested algorithms.

It was also tested how the algorithms respond to the disturbance change by 10% from $C_{\mathrm{Af0}} = 10\,\mathrm{mol/l}$ to $C_{\mathrm{Af1}} = 11\,\mathrm{mol/l}$ (Fig. 7). The responses obtained

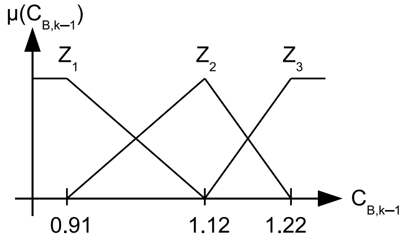

Fig. 4. Membership functions of the fuzzy model.

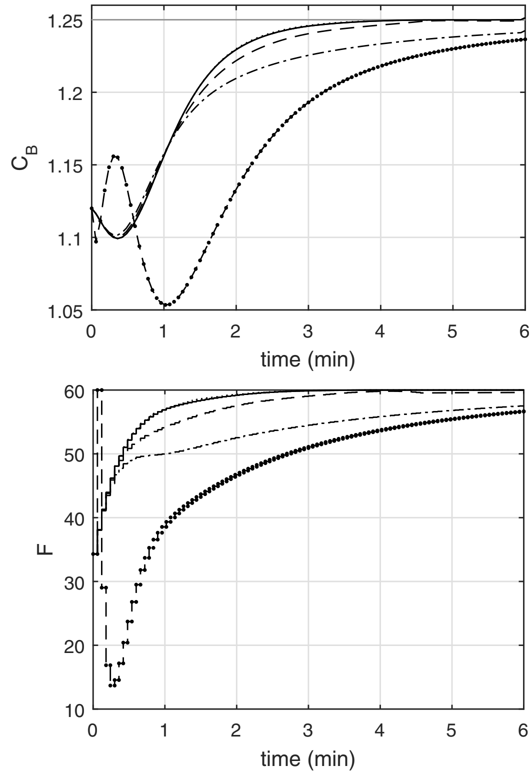

Fig. 5. Responses of control systems to the change in the set-point value to $\overline{C}_B = 1.25$; EFMPC—solid lines, optimization-based FMPC—dotted lines, NMPC—dashed lines, LMPC—dash-dotted lines and PID—dashed lines with points.

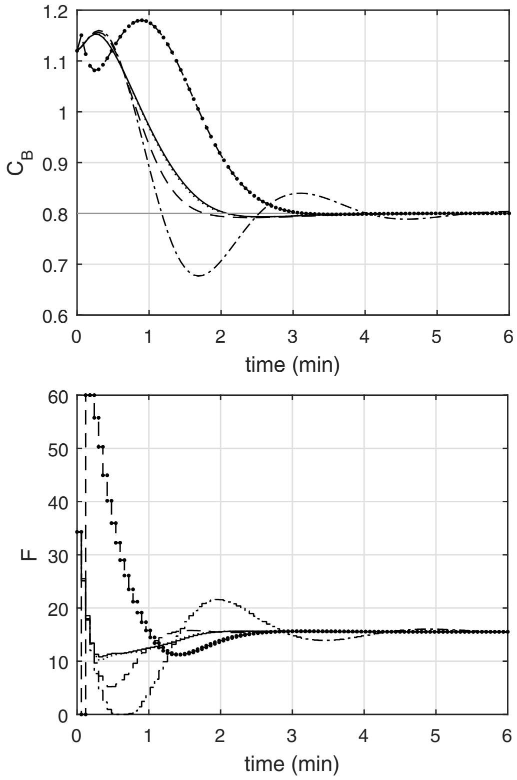

Fig. 6. Responses of control systems to the change of the set-point value to $\overline{C}_B = 0.8$; EFMPC—solid lines, optimization-based FMPC—dotted lines, NMPC—dashed lines, LMPC—dash-dotted lines and PID—dashed lines with points.

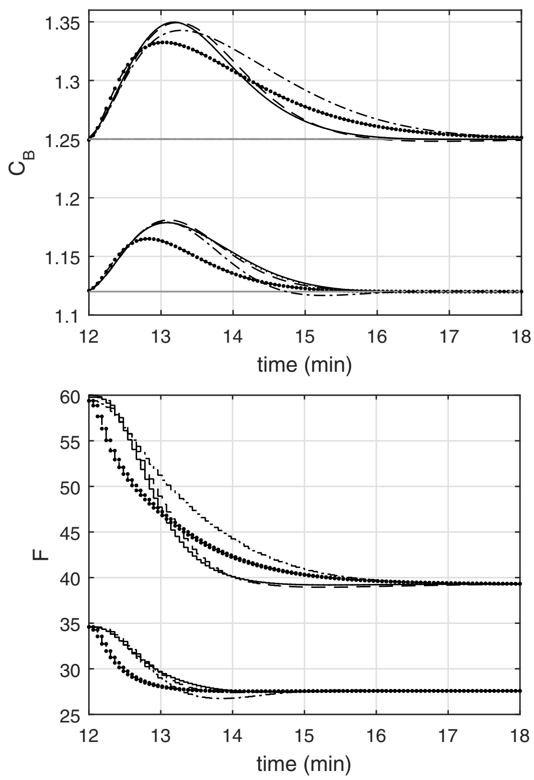

Fig. 7. Responses of control systems to the change in the disturbance by 10% to $C_{\mathrm{Af1}} = 11\ \mathrm{mol/l}$: EFMPC—solid lines, optimization-based FMPC—dotted lines, NMPC—dashed lines, LMPC—dash-dotted lines and PID—dashed lines with points.

in control systems with the EFMPC (solid lines) and FMPC (dotted lines) algorithms are almost the same and very similar to these obtained with the NMPC algorithm (dashed lines). The LMPC and PID algorithms (dash-dot lines and dashed lines with points, respectively) compensate the disturbance faster than MPC algorithms based on nonlinear models when operating near $\overline{C}_B = 1.12$, but slower when operating near $\overline{C}_B = 1.25$. In the latter case the maximal control error generated with the LMPC algorithm is lower than in the case when MPC algorithms based on nonlinear models are used. Near both operating points the maximal control error is the smallest when the PID controller is used in the control system.

There was also an experiment conducted to show the effectiveness in the constraint handling mechanism. The algorithms were tuned to work faster at the expense of a more aggressive control action (λ decreased to 0.0001); see Fig. 8. The responses obtained in control systems with the EFMPC (solid lines) and FMPC (dotted lines) algorithms are almost the same. The difference can be noticed in control signal between the 1st and 2nd minute of the experiment; the output responses are almost the same (the difference is barely perceptible). For comparison, also the responses obtained in a control system with the EFMPC algorithm in the unconstrained case (dashed lines) are shown. In such a case the rise time is faster, but at the expense of small overshooting.

There were also experiments conducted with changes of the parameters of the control plant—reaction rate constants k_i $(i = 1, 2, 3)$. The responses obtained after a decrease by 10% are shown in Fig. 9 and the ones obtained after an increase by 10% are shown in Fig. 10. The changes in reaction rate constants significantly influence the behavior of the control plant. Despite that, in all cases the control system is stable. Moreover, in most cases the obtained responses are good. In some cases, the set-point value $\overline{C}_B = 1.25$ cannot be achieved due to the upper bound of the control signal, but in these cases the output signal stabilizes on the value which is as close to the set-point as possible.

4.2. MIMO control plant. The next control plant considered is a pH reactor (Fig. 11). The process model is given by the set of the following equations

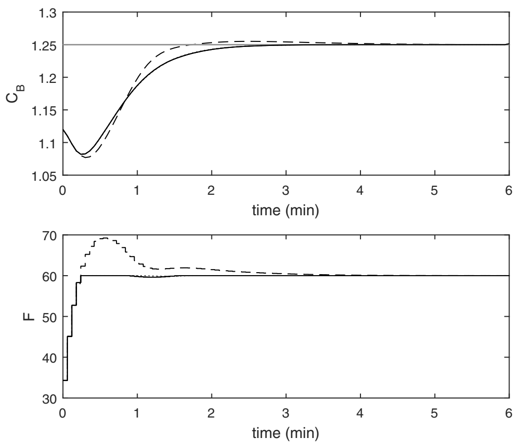

Fig. 8. Responses of control systems to the change in the set-point value to $\overline{C}_B = 1.25$, $\lambda = 0.0001$: EFMPC—solid lines, optimization-based FMPC—dotted lines, EFMPC and no constraints—dashed lines.

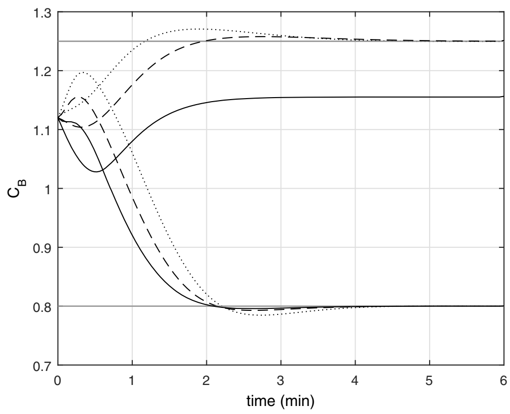

Fig. 9. Responses obtained in the control system for $k_1 = 45$ (solid line), $k_2 = 90$ (dotted line) and $k_3 = 9$ (dashed line).

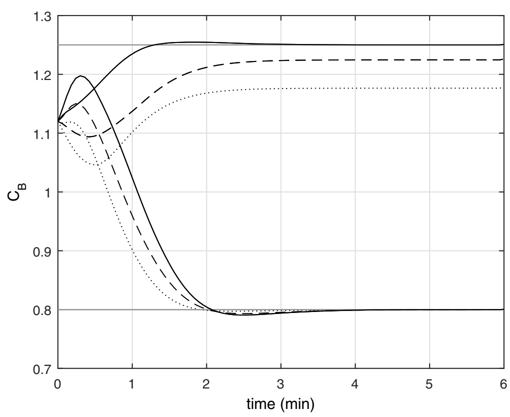

Fig. 10. Responses obtained in the control system for $k_1 = 55$ (solid line), $k_2 = 110$ (dotted line) and $k_3 = 11$ (dashed line).

(Chen and Huang, 2004; Dougherty and Cooper, 2003; Hu *et al.*, 2000; Marusak, 2009a):

$$\frac{\mathrm{d}h}{\mathrm{d}t} = \frac{1}{A} Q_1 + Q_2 + Q_3 + Q_4, \quad Q_4 = C_V \sqrt{h}, \tag{24}$$

$$\begin{aligned}\frac{\mathrm{d}W_{a4}}{\mathrm{d}t} =& \frac{1}{A \cdot h}\Big[(W_{a1} - W_{a4}) \cdot Q_1 \\ &+ (W_{a2} - W_{a4}) \cdot Q_2 + (W_{a3} - W_{a4}) \cdot Q_3\Big],\end{aligned} \tag{25}$$

$$\begin{aligned}\frac{\mathrm{d}W_{b4}}{\mathrm{d}t} =& \frac{1}{A \cdot h}\Big[(W_{b1} - W_{b4}) \cdot Q_1 \\ &+ (W_{b2} - W_{b4}) \cdot Q_2 + (W_{b3} - W_{b4}) \cdot Q_3\Big],\end{aligned} \tag{26}$$

$$\begin{aligned}&W_{a4} + 10^{pH-14} \\ &+ W_{b4} \frac{1 + 2 \cdot 10^{pH - pK_2}}{1 + 10^{pK_1 - pH} + 10^{pK_2 - pH}} - \frac{1}{10^{pH}} = 0,\end{aligned} \tag{27}$$

where Q_1 is the flow rate of acid (HNO_3), Q_2 is the flow rate of buffer ($NaHCO_3$), Q_3 is the flow rate of base (NaOH), Q_4 is the gravitational outflow of the product, h is the level of the liquid in the reactor, pH indicates the composition of the product. The values of the parameters in the pH reactor model are as follows:

$$\begin{aligned} W_{a1} &= 3 \cdot 10^{-3}\ \mathrm{M}, & W_{a2} &= -3 \cdot 10^{-2}\ \mathrm{M}, \\ W_{a3} &= -3.05 \cdot 10^{-3}\ \mathrm{M}, & W_{b1} &= 0\ \mathrm{M}, \\ W_{b2} &= 3 \cdot 10^{-2}\ \mathrm{M}, & W_{b3} &= 5 \cdot 10^{-5}\ \mathrm{M}, \\ A &= 207\ \mathrm{cm}^2, & C_V &= 8.75\ \tfrac{\mathrm{ml}}{\mathrm{cm \cdot s}}, \\ pK_1 &= 6.35, & pK_2 &= 10.25. \end{aligned}$$

Moreover, there is a delay of pH measurement, which is equal to $T_d = 30$ s. The control task is thus difficult because of the delay and high nonlinearity of the control plant; the steady-state characteristic $pH(Q_1, Q_3)$ is shown in Fig. 12.

It is assumed that the manipulated variables are the flow rates: of acid—Q_1 and of base—Q_3; the controlled variables are: the level of the liquid in the reactor—h and the composition of the product—pH. It is assumed that the liquid level should be stabilized on the fixed value $h_0 = 14$ cm.

The fuzzy model by Marusak (2009a) is used; it is composed of five sets of step responses, with membership functions shown in Fig. 13. The step responses were obtained in vicinity of the following operating points:

1. $h_0 = 14$ cm, $pH_0 = 4$,
   $Q_{10} = 19.48\ \frac{\mathrm{ml}}{\mathrm{s}}$, $Q_{30} = 12.71\ \frac{\mathrm{ml}}{\mathrm{s}}$;

2. $h_0 = 14\,\mathrm{cm},\ pH_0 = 5,$
   $Q_{10} = 18.89\,\frac{\mathrm{ml}}{\mathrm{s}},\ Q_{30} = 13.30\,\frac{\mathrm{ml}}{\mathrm{s}};$
3. $h_0 = 14\,\mathrm{cm},\ pH_0 = 6.5,$
   $Q_{10} = 17.29\,\frac{\mathrm{ml}}{\mathrm{s}},\ Q_{30} = 14.90\,\frac{\mathrm{ml}}{\mathrm{s}};$
4. $h_0 = 14\,\mathrm{cm},\ pH_0 = 8,$
   $Q_{10} = 16.14\,\frac{\mathrm{ml}}{\mathrm{s}},\ Q_{30} = 16.05\,\frac{\mathrm{ml}}{\mathrm{s}};$
5. $h_0 = 14\,\mathrm{cm},\ pH_0 = 10,$
   $Q_{10} = 14.51\,\frac{\mathrm{ml}}{\mathrm{s}},\ Q_{30} = 17.68\,\frac{\mathrm{ml}}{\mathrm{s}}.$

Operation of the proposed EFMPC algorithm will be compared with that of optimization-based FMPC and of LMPC. The sampling time was assumed equal to $T_\mathrm{s} = 10$ s; tuning parameters of all three algorithms are assumed the same as in the work of Marusak (2009a): prediction horizon $p = 100$, control horizon $s = 50$, $\lambda_1 = 1$, $\lambda_2 = 1$, $\kappa_1 = 1$ and $\kappa_2 = 1$. During the experiments the value of the flow rate of buffer Q_2 in the equations used to simulate the control plant was increased by 10%. The responses obtained after changes of the set-point value are shown in Figs. 14 and 15. The LMPC algorithm (dash-dot lines), based on a single set of step responses, obtained near the 4th operating point, works well near this point—for the set-point change to 9. For other set-point values, the LMPC algorithm is much slower than fuzzy algorithms. Both EFMPC (solid lines) and optimization-based FMPC (dotted lines) offer much better control performance. Fuzzy algorithms generate responses very similar to each other.

The output variable pH reaches the set-point values much faster in the control systems with fuzzy algorithms. There is some overshoot, although it is small. The responses obtained for different set-point values have shapes similar to each other—the influence of control plant nonlinearities on control quality is then significantly reduced, thanks to the application of control algorithms based on nonlinear models. The second output variable—the liquid level in the reactor h is stabilized properly by all the algorithms.

5. Summary

The proposed algorithm is computationally efficient and, at the same time, offers very good control performance. It uses the nonlinear model to derive the free response of the control plant and the approximate, easy to obtain, fuzzy model to calculate the control signal in a fast and efficient way. Thanks to such an approach, repetition of on-line optimization at each iteration of the algorithm is avoided. Despite that, the proposed algorithm offers control performance very close to that granted by the algorithms with optimization.

The control signal in the proposed algorithm is generated fast. Therefore the algorithm, although it is based on a nonlinear model, can be used with relatively small sampling times. The proposed algorithm can be used not only as a stand-alone one, but also in the control systems with NMPC algorithms, to improve the numerical properties of the latter ones. It is also possible to obtain the free response using other methods than the one exploited in the paper, for example, methods dedicated to particular types of models; the proposed approach is independent of the technique of free response generation.

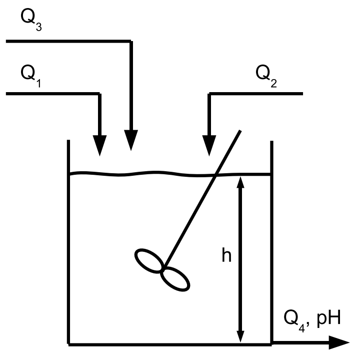

Fig. 11. Diagram of the pH reactor.

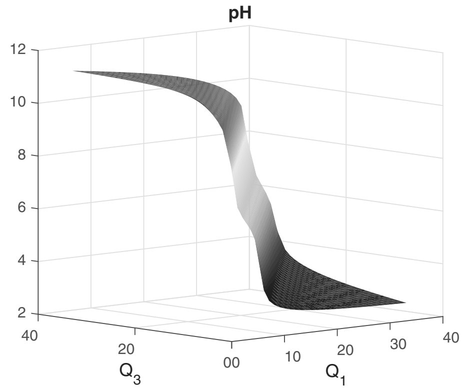

Fig. 12. Steady-state characteristic $pH(Q_1, Q_3)$.

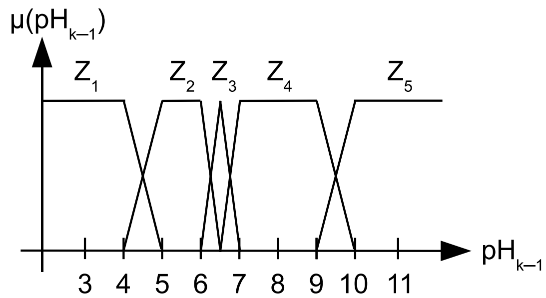

Fig. 13. Membership functions of the fuzzy model.

References

Bemporad, A., Borrelli, F. and Morari, M. (2000). Piecewise linear optimal controllers for hybrid systems, *Proceedings of the 2000 American Control Conference, ACC, Chicago, USA*, Vol. 2, pp. 1190–1194.

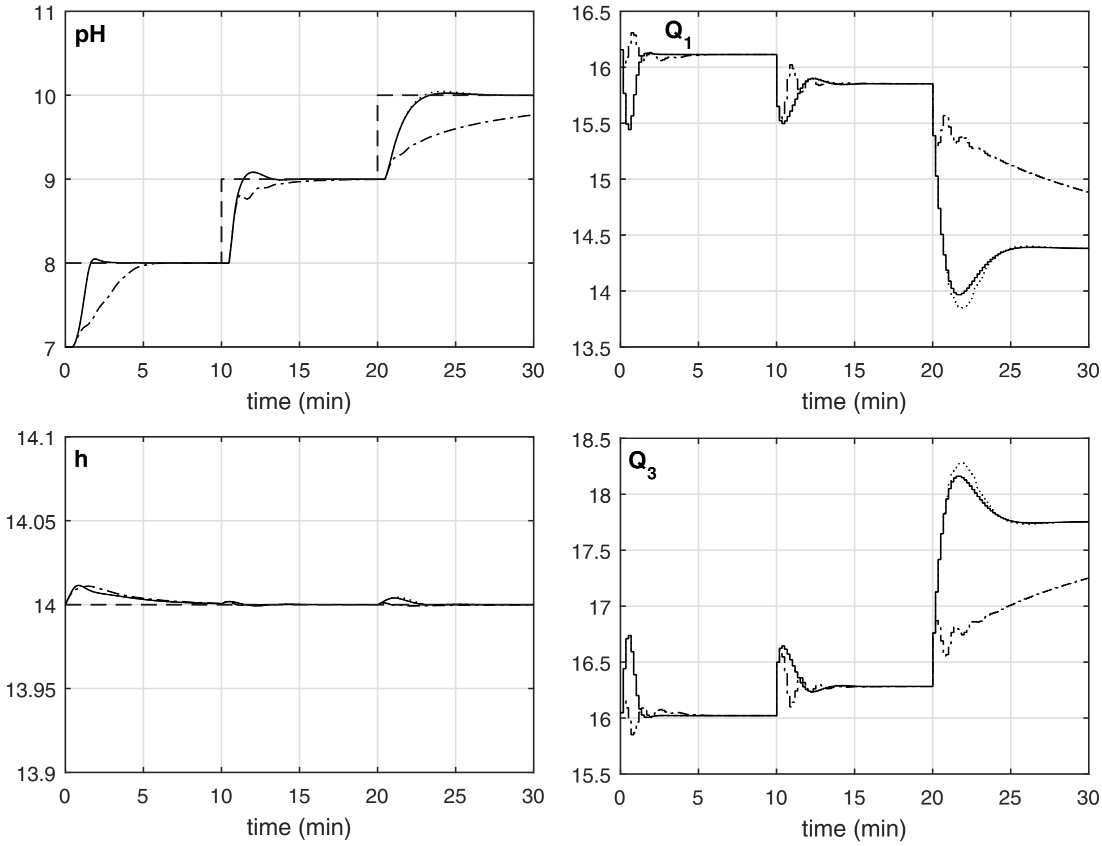

Fig. 14. Responses of control systems to the change of pH set-points from 7 to 8, 9 and 10: EFMPC—solid lines, optimization-based FMPC—dotted lines and LMPC—dash-dotted lines, set-point trajectory—dashed lines; left—output variables pH and h, right—manipulated variables Q_1 and Q_3.

Bemporad, A., Morari, M., Dua, V. and Pistikopoulos, E.N. (2002). The explicit linear quadratic regulator for constrained systems, *Automatica* **38**(1): 3–20.

Blevins, T., McMillan, G., Wojsznis, W. and Brown, M. (2003). *Advanced Control Unleashed*, ISA—The Instrumentation, Systems, and Automation Society, Research Triangle Park.

Boulkaibet, I., Belarbi, K., Bououden, S., Marwala, T. and Chadli, M. (2017). A new T–S fuzzy model predictive control for nonlinear processes, *Expert Systems with Applications* **88**: 132–151.

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

Chaber, P. and Ławryńczuk, M. (2019). Fast analytical model predictive controllers and their implementation for STM32 ARM microcontroller, *IEEE Transactions on Industrial Informatics* **15**(8): 4580–4590.

Chen, J. and Huang, T. (2004). Applying neural networks to on-line updated PID controllers for nonlinear process control, *Journal of Process Control* **14**(2): 211–230.

Diehl, M., Bock, H., Schlöder, J.P., Findeisen, R., Nagy, Z. and Allgöwer, F. (2002). Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations, *Journal of Process Control* **12**(4): 577–585.

Dominguez, L.F. and Pistikopoulos, E.N. (2010). A novel mp-NLP algorithm for explicit/multi-parametric NMPC, *IFAC Proceedings Volumes* **43**(14): 539–544.

Dougherty, D. and Cooper, D. (2003). A practical multiple model adaptive strategy for single-loop MPC, *Control Engineering Practice* **11**(2): 141–159.

Doyle, F., Ogunnaike, B. and Pearson, R. (1995). Nonlinear model-based control using second-order Volterra models, *Automatica* **31**(5): 697–714.

Essien, E., Ibrahim, H., Mehrandezh, M. and Idem, R. (2019). Adaptive neuro-fuzzy inference system (ANFIS)-based model predictive control (MPC) for carbon dioxide reforming of methane (CDRM) in a plug flow tubular reactor for hydrogen production, *Thermal Science and Engineering Progress* **9**: 148–161.

Guerra, T.-M., Kruszewski, A. and Lauber, J. (2009). Discrete Tagaki–Sugeno models for control: Where are we?, *Annual Reviews in Control* **33**(1): 37–47.

Hu, Q., Saha, P. and Rangaiah, G. (2000). Experimental evaluation of an augmented IMC for nonlinear systems, *Control Engineering Practice* **8**(10): 1167–1176.

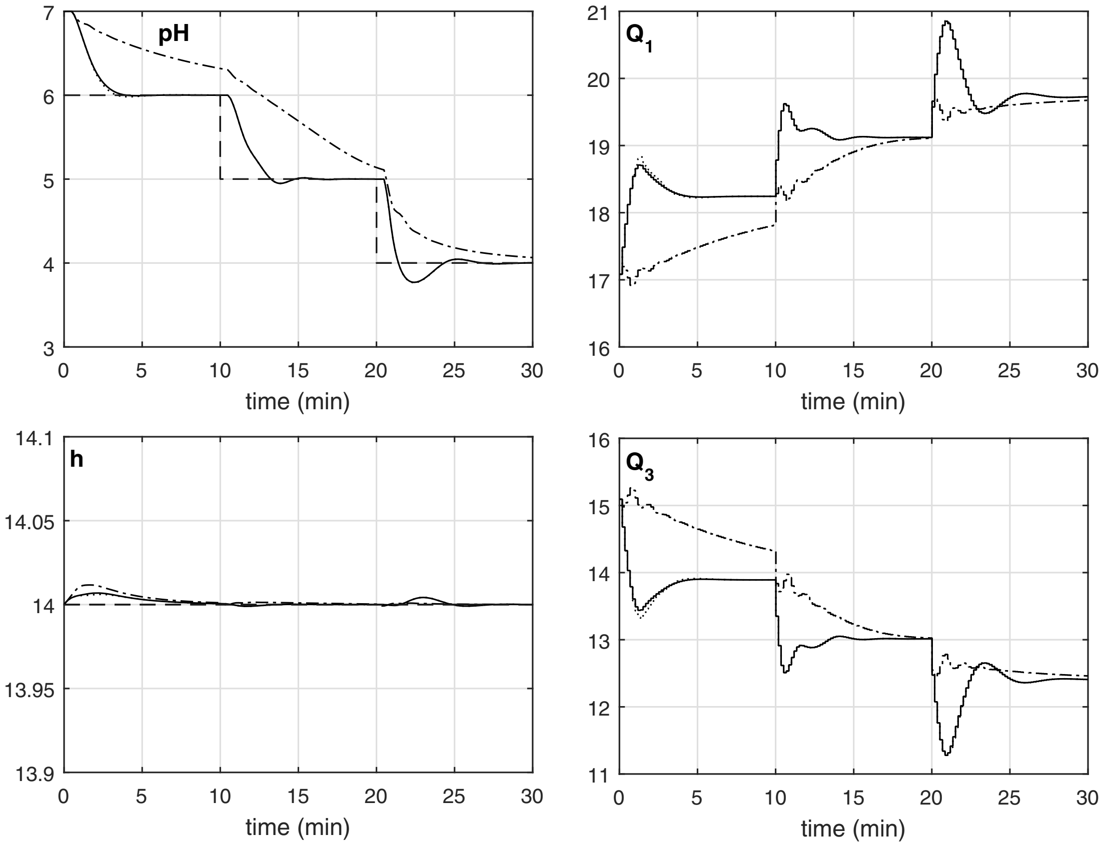

Fig. 15. Responses of control systems to the change of pH set-points from 7 to 6, 5 and 4: EFMPC—solid lines, optimization-based FMPC—dotted lines and LMPC—dash-dotted lines; set-point trajectory—dashed lines, left—output variables pH and h, right—manipulated variables Q_1 and Q_3.

Johansen, T.A. (2002). On multi-parametric nonlinear programming and explicit nonlinear model predictive control, *Proceedings of the 41st IEEE Conference on Decision and Control, Las Vegas, USA*, Vol. 3, pp. 2768–2773.

Johansen, T.A. (2004). Approximate explicit receding horizon control of constrained nonlinear systems, *Automatica* **40**(2): 293–300.

Karimi Pour, F., Puig, V. and Ocampo-Martinez, C. (2018). Multi-layer health-aware economic predictive control of a pasteurization pilot plant, *International Journal of Applied Mathematics and Computer Science* **28**(1): 97–110, DOI: 10.2478/amcs-2018-0007.

Khooban, M.H., Vafamand, N. and Niknam, T. (2016). T–S fuzzy model predictive speed control of electrical vehicles, *ISA Transactions* **64**: 231–240.

Killian, M. and Kozek, M. (2017). Optimal partitioning of a boiler-turbine unit for fuzzy model predictive control, *IFAC-PapersOnLine* **50**(1): 2011–2016.

Kong, L. and Yuan, J. (2019a). Disturbance-observer-based fuzzy model predictive control for nonlinear processes with disturbances and input constraints, *ISA Transactions* **90**: 74–88.

Kong, L. and Yuan, J. (2019b). Generalized discrete-time nonlinear disturbance observer based fuzzy model predictive control for boiler-turbine systems, *ISA Transactions* **90**: 89–106.

Kouvaritakis, B., Cannon, M. and Rossiter, J. (2002). Who needs QP for linear MPC anyway?, *Automatica* **38**(5): 879–884.

Krishna, D., Suryanarayana, K., Aparna, G. and Padma Sree, R. (2012). Tuning of PID controllers for continuous stirred tank reactors, *Indian Chemical Engineer* **54**(3): 157–179.

Ławryńczuk, M. (2014). *Computationally Efficient Model Predictive Control Algorithms: A Neural Network Approach*, Springer, Heidelberg.

Ławryńczuk, M. (2015). Nonlinear state-space predictive control with on-line linearisation and state estimation, *International Journal of Applied Mathematics and Computer Science* **25**(4): 833–847, DOI: 10.1515/amcs-2015-0060.

Ławryńczuk, M. (2020). Nonlinear model predictive control for processes with complex dynamics: A parameterisation approach using Laguerre functions, *International Journal of Applied Mathematics and Computer Science* **30**(1): 35–46, DOI: 10.34768/amcs-2020-0003.

Ławryńczuk, M., Marusak, P. and Tatjewski, P. (2008). Cooperation of model predictive control with steady-state economic optimisation, *Control and Cybernetics* **37**(1): 133–158.

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

Marusak, P. (2009a). Advantages of an easy to design fuzzy predictive algorithm in control systems of nonlinear chemical reactors, *Applied Soft Computing* **9**(3): 1111–1125.

Marusak, P. (2009b). Efficient model predictive control algorithm with fuzzy approximations of nonlinear models, *in* M. Kolehmainen *et al.* (Eds), *Adaptive and Natural Computing Algorithms, ICANNGA 2009*, Lecture Notes in Computer Science, Vol. 5495, Springer, Berlin/Heidelberg, pp. 448–457.

Marusak, P. (2010). A mechanism of output constraint handling for analytical fuzzy controllers, *in* N. García-Pedrajas *et al.* (Eds), *Trends in Applied Intelligent Systems, IEA/AIE 2010*, Lecture Notes in Computer Science, Vol. 6097, Springer, Berlin/Heidelberg, pp. 222–227.

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.

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

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

Nebeluk, R. and Marusak, P. (2020). Efficient MPC algorithms with variable trajectories of parameters weighting predicted control errors, *Archives of Control Sciences* **30**(2): 325–363.

Pistikopoulos, E.N., Dua, V., Bozinis, N.A., Bemporad, A. and Morari, M. (2000). On-line optimization via off-line parametric optimization tools, *Computers and Chemical Engineering* **24**(2): 183–188.

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

Schäfer, A., Kühl, P., Diehl, M., Schlöder, J. and Bock, H. G. (2007). Fast reduced multiple shooting methods for nonlinear model predictive control, *Chemical Engineering and Processing: Process Intensification* **46**(11): 1200–1214.

Shen, D., Lim, C.-C. and Shi, P. (2020). Robust fuzzy model predictive control for energy management systems in fuel cell vehicles, *Control Engineering Practice* **98**: 104364.

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

Tatjewski, P. (2014). Disturbance modeling and state estimation for offset-free predictive control with state-spaced process models, *International Journal of Applied Mathematics and Computer Science* **24**(2): 313–323, DOI: 10.2478/amcs-2014-0023.

Tatjewski, P. (2017). Offset-free nonlinear model predictive control with state-space process models, *Archives of Control Sciences* **27**(4): 595–615.

Wojtulewicz, A. and Ławryńczuk, M. (2018a). Computationally efficient implementation of dynamic matrix control algorithm for very fast processes using programmable logic controller, *23rd International Conference on Methods and Models in Automation and Robotics (MMAR), Międzyzdroje, Poland*, pp. 579–584.

Wojtulewicz, A. and Ławryńczuk, M. (2018b). Implementation of multiple-input multiple-output dynamic matrix control algorithm for fast processes using field programmable gate array, *IFAC-PapersOnLine* **51**(6): 324–329.

Wu, X., Shen, J., Li, Y. and Lee, K.Y. (2015). Fuzzy modeling and predictive control of superheater steam temperature for power plant, *ISA Transactions* **56**: 241–251.

Zavala, V.M., Laird, C.D. and Biegler, L.T. (2008). A fast moving horizon estimation algorithm based on nonlinear programming sensitivity, *Journal of Process Control* **18**(9): 876–884.

Piotr M. Marusak was born in Warsaw, Poland, in 1974. He received his MSc degree in computer science and his PhD 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 of nonlinear systems, the application of these algorithms in fault-tolerant control systems, the cooperation of the predictive control algorithms with set-point optimization, process modeling and simulation, and fuzzy control.

Received: 31 December 2019
Revised: 23 June 2020
Re-revised: 4 October 2020
Accepted: 5 October 2020