

Article

Numerically Efficient Fuzzy MPC Algorithm with Advanced Generation of Prediction—Application to a Chemical Reactor

Piotr M. Marusak 

Institute of Control and Computation Engineering, Warsaw University of Technology, Nowowiejska 15/19, 00-665 Warszawa, Poland; P.Marusak@ia.pw.edu.pl

Received: 5 May 2020; Accepted: 10 June 2020; Published: 14 June 2020



Abstract: In Model Predictive Control (MPC) algorithms, control signals are generated after solving optimization problems. If the model used for prediction is linear then the optimization problem is a standard, easy to solve, quadratic programming problem with linear constraints. However, such an algorithm may offer insufficient performance if applied to a nonlinear control plant. On the other hand, if a model used for prediction is nonlinear, then non-convex optimization problem must be solved at each algorithm iteration. Then the numerical problems may occur during solving it and the time needed to calculate the control signals cannot be determined. Therefore approaches based on linearized models are preferred in practical applications. A fuzzy algorithm with an advanced generation of the prediction is proposed in the article. The prediction is obtained in such a way that the algorithm is formulated as a quadratic optimization problem but offers performance very close to that of the MPC algorithm with nonlinear optimization. The efficiency of the proposed approach is demonstrated in the control system of a nonlinear chemical control plant—a CSTR (Continuous Stirred–Tank Reactor) with van de Vusse reaction.

Keywords: prediction; process control; model predictive control; fuzzy systems; fuzzy control; nonlinear control

1. Introduction

The Model Predictive Control (MPC) algorithms during the calculation of the control signals use the model of the control plant directly, predicting the behavior of the process. Thanks to such an approach they can be successfully applied in control systems of control plants with difficult dynamics (delay, inverse response), with constraints and for MIMO (Multiple–Input Multiple–Output) processes, see for example, References [1–7].

The simplest MPC algorithms are based on linear control plant models; see for example, References [1,8]. Unfortunately, if a control plant is nonlinear and the MPC algorithm should operate for different operating points, as in control system structures with steady–state setpoint optimization (see e.g., References [7,9,10]), then the algorithm based on a linear model may be insufficient in terms of control performance. In such a case operation of the control system can be usually improved using the MPC algorithm based on a nonlinear model.

On the other hand, if a nonlinear process model is used in the MPC algorithm directly (such algorithm is called NMPC—Nonlinear MPC), then the optimization problem solved at each algorithm iteration to generate the control signals is a nonlinear optimization problem; see for example, References [11–13]. Thus, it is usually non-convex, hard to solve, and time-consuming computational task. Moreover, the time needed to find the solution cannot be foreseen in advance. One of the ways to cope with this problem is to apply so called fast NMPC algorithms in which

the solution is generated faster than in the standard approach, but unfortunately, at the expense of suboptimality; see for example, References [14–16]. One of such approaches is the explicit one in which most of calculations are done off-line; see for example, References [17–19]. It is an interesting approach and it is optimal if the linear model is used [20–22]. However it has a serious drawback—the complexity of the controller grows significantly with the number of inequality constraints in the problem. When the fuzzy models are used, then of course the nonlinear optimization can be applied in the algorithm. However the form of the model is such that easier to solve algorithms can be formulated. Among the algorithms exploiting the structure of the fuzzy models there is a group which is based on Linear Matrix Inequalities (LMIs); see for example, References [23–27]. The other group is based on the classical fuzzy Takagi–Sugeno approach in which a few algorithms based on linear models are used to form the fuzzy controller; see for example, References [28,29].

The other approach consists in online linearization of the process model at each MPC algorithm iteration; see for example, References [7,30–36]. Then prediction is formulated in such a way that it is linear relative to decision variables in the optimization problem solved at each iteration of the algorithm. Such form of the prediction leads to the formulation of the optimization problem as the quadratic one. In the linearization-based algorithms the prediction can be based on a classical linearization or a clever method of prediction generation can be used, exploiting structure of the nonlinear model the algorithm is based on. In the algorithms using linearization of the fuzzy Takagi–Sugeno model, described in References [31,34], both—the free response and the dynamic matrix were obtained using the model obtained after the linearization. In References [32,36] the model obtained after linearization was used to obtain the dynamic matrix whereas the (classical) free response was calculated using the nonlinear model.

The MPC algorithms proposed in the article are an extension of the algorithm presented in Reference [36] and, like their predecessor, are based on two models—a nonlinear model and its, easy to obtain, fuzzy approximation. The nonlinear model can have any form of the model generating outputs on the base of input signals. This model is used to obtain an advanced (based on the previously calculated trajectory of the future control signals) free response of the process prediction. The easy-to-obtain fuzzy model is used to derive the dynamic matrix describing influence of the future, calculated control signals on prediction. The prediction has such a form that the proposed algorithms are formulated as the standard quadratic programming problem. Thus, the algorithms combine the computational simplicity of the MPC algorithms based on linear models with control performance almost as good as the one offered by the algorithm with nonlinear optimization. The prediction can be also improved iteratively, if needed. Iterative prediction improvement, though slightly different in details, is also used in the interesting iterative-learning-based approaches to batch control, see for example, References [37–39].

Standard MPC algorithms based on linear and nonlinear models are described in the next section. In Section 3 the MPC algorithms based on fuzzy and nonlinear models are proposed, and mechanisms of their improvements and simplifications are detailed. A simulation example of the application of the proposed algorithms to a nonlinear chemical reactor with the inverse response is presented in Section 4, illustrating the efficiency of the proposed approach. The article is concluded in the last section.

2. Model Predictive Control Algorithms

Future control signal trajectories are generated in the MPC algorithms in such a way that predicted, many sampling instants ahead, future behavior of the control plant fulfills assumed criteria. It is usually formulated as an optimization problem in which, it is demanded that a performance function is minimized subject to the constraints put on control and output signals. The performance function takes into consideration future, predicted control errors and the shape of control signals.

Thus, the optimization problem solved by the MPC algorithm in each iteration has, most often, the following form [1,4,5,7]:

$$\arg \min_{\Delta u} \left\{ J_{\text{MPC}} = (\bar{y} - y)^T \cdot \kappa \cdot (\bar{y} - y) + \Delta u^T \cdot \Lambda \cdot \Delta u \right\} \quad (1)$$

subject to:

$$\Delta u_{\min} \leq \Delta u \leq \Delta u_{\max} , \quad (2)$$

$$u_{\min} \leq u \leq u_{\max} , \quad (3)$$

$$y_{\min} \leq y \leq y_{\max} , \quad (4)$$

where

$$y = [y^1, \dots, y^{n_y}]^T, y^j = [y_{k+1|k}^j, \dots, y_{k+p|k}^j] , \quad (5)$$

$$\Delta u = [\Delta u^1, \dots, \Delta u^{n_u}]^T, \Delta u^m = [\Delta u_{k+1|k}^m, \dots, \Delta u_{k+s-1|k}^m] , \quad (6)$$

$$u = [u^1, \dots, u^{n_u}]^T, u^m = [u_{k+1|k}^m, \dots, u_{k+s-1|k}^m] , \quad (7)$$

$$\kappa = [\kappa_1, \dots, \kappa_{n_y}] \cdot I, \kappa_i = [\kappa_i, \dots, \kappa_i] , \quad (8)$$

$$\Lambda = [\Lambda_1, \dots, \Lambda_{n_u}] \cdot I, \Lambda_i = [\lambda_i, \dots, \lambda_i] , \quad (9)$$

$y_{k+j|k}^j$ is a 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, $\Delta u_{k+i|k}^m$ are future changes in manipulated variables, $\kappa_j \geq 0$ and $\lambda_m \geq 0$ are weighting coefficients for the predicted control errors of the j th output and for the changes of the m th manipulated variable, respectively; p and s denote prediction and control horizons, respectively; n_y, n_u denote number of output and manipulated variables, respectively; $\Delta u_{\min}, \Delta u_{\max}, u_{\min}, u_{\max}, y_{\min}, y_{\max}$ are vectors of lower and upper bounds of changes and values of the control signals and of the values of output variables, respectively; κ_i has p elements, Λ_i has s elements. The reference trajectories are grouped in the following vector:

$$\bar{y} = [\bar{y}^1, \dots, \bar{y}^{n_y}]^T, \bar{y}^j = [\bar{y}_{k+1|k}^j, \dots, \bar{y}_{k+p|k}^j] . \quad (10)$$

It is often assumed that reference trajectories are constant on the prediction horizon, then the vectors \bar{y}^j (each composed of p elements) have the following form:

$$\bar{y}^j = [\bar{y}_k^j, \dots, \bar{y}_k^j] , \quad (11)$$

where \bar{y}_k^j is a setpoint value for the j th output.

As a solution to the optimization problem ((1)–(4)) the optimal vector of control action is obtained in such a way that future, predicted control errors are minimized (the first component of the performance index), control signals do not vary too much (the second component) and the constraints on control and output signals are fulfilled. The $\Delta u_{k|k}^m$ elements from the vector Δu are applied in the control system and the algorithm passes to the next iteration. The predicted values of output signals $y_{k+j|k}^j$ are derived using a control plant model the MPC algorithm is based on.

2.1. MPC Algorithms Based on Nonlinear Models (NMPC)

Assume that we have a nonlinear process model:

$$\hat{y}_k = f(y_{k-1}, y_{k-2}, \dots, y_{k-n_a}, u_{k-1}, u_{k-2}, \dots, u_{k-n_b}) , \quad (12)$$

where $\mathbf{y}_{k-i} = [y_{k-i}^1, \dots, y_{k-i}^{n_y}]^T$ is a vector which contains output values measured at the $(k-i)$ th sampling instant, $\mathbf{u}_{k-i} = [u_{k-i}^1, \dots, u_{k-i}^{n_u}]^T$ is a vector which contains control values applied at the $(k-i)$ th sampling instant; denote output values generated by the model in the $(k+i)$ th sampling instant as $\hat{\mathbf{y}}_{k+i} = [\hat{y}_{k+i}^1, \dots, \hat{y}_{k+i}^{n_y}]$, n_a, n_b determine how many past output and control values the model needs. If one wants to use the model (12) directly in the optimization problem ((1)–(4)), then the prediction takes the form of the following formulas, passed to the optimization problem as a set of equality constraints:

$$\mathbf{y}_{k+i|k} = \mathbf{f}(\hat{\mathbf{y}}_{k+i-1}, \dots, \hat{\mathbf{y}}_{k+1}, \mathbf{y}_k, \dots, \mathbf{y}_{k-n_a+i-1}, \mathbf{u}_{k+i-1|k}, \dots, \mathbf{u}_{k|k}, \mathbf{u}_{k-1}, \dots, \mathbf{u}_{k-n_b+i-1}) + \mathbf{d}_k, \quad (13)$$

where $\mathbf{u}_{k+i|k} = [u_{k+i|k}^1, \dots, u_{k+i|k}^{n_u}]^T$ is a vector containing future control values, depending on decision variables from the vector $\Delta \mathbf{u}$ and

$$\mathbf{d}_k = \mathbf{y}_k - \hat{\mathbf{y}}_k, \quad (14)$$

where $\mathbf{y}_k = [y_k^1, \dots, y_k^{n_y}]^T$ is a vector of recently measured output values; it is assumed that \mathbf{d}_k is the same for all instants in the prediction horizon—an approach proposed in the Dynamic Matrix Control (DMC) algorithm and therefore called the DMC-type disturbance model; see for example, Reference [7].

Unfortunately, the optimization task with the prediction (13), based on a nonlinear model, is not a quadratic optimization task. The optimization problem ((1)–(4)) becomes, in general, non-convex nonlinear optimization problem. Solving such a problem is difficult and time-consuming task, there is no guarantee to find a global solution and time needed to obtain solution cannot be estimated in advance. Different kinds of nonlinear process models can be used in the algorithm, but they are exploited in a similar way, see for example, References [13,40–42]; such algorithms will be referred to as Nonlinear MPC (NMPC). Due to problems caused by formulation of NMPC algorithms, in practice MPC algorithms based on linear approximations of process models, obtained at each iteration of the algorithm are considered. Such linear approximation is used to formulate the, easy to solve, quadratic optimization problem [35]. Such an optimization problem is also used in the MPC algorithms based on linear models, therefore now their formulation will be reminded. In the next section, the proposed algorithms, based on an advanced approximation of the nonlinear model, will be proposed.

2.2. MPC Algorithms Based on Linear Models (LMPC)

In the case of linear models the superposition principle holds, therefore the vector of predicted output values \mathbf{y} can be decomposed into two parts:

$$\mathbf{y} = \tilde{\mathbf{y}} + \mathbf{A} \cdot \Delta \mathbf{u}, \quad (15)$$

$$\tilde{\mathbf{y}} = [\tilde{y}^1, \dots, \tilde{y}^{n_y}]^T, \quad \tilde{\mathbf{y}}^j = [\tilde{y}_{k+1|k}^j, \dots, \tilde{y}_{k+p|k}^j], \quad (16)$$

where $\tilde{\mathbf{y}}$ is called the free response of the plant, because it contains future values of output signals calculated assuming that the control signal does not change in the future — it describes influence of the past values of control signals; form of the free response depends on the model used for prediction, see for example, References [1,4,5,7]. The second component in the formula (15) depends on future (to be calculated) changes of the control signals and is called the forced response; the matrix \mathbf{A} , called the dynamic matrix, is given by the following formula:

$$A = \begin{bmatrix} A^{11} & A^{12} & \dots & A^{1n_u} \\ A^{21} & A^{22} & \dots & A^{2n_u} \\ \vdots & \vdots & \ddots & \vdots \\ A^{n_y1} & A^{n_y2} & \dots & A^{n_y n_u} \end{bmatrix}, \quad (17)$$

$$A^{jm} = \begin{bmatrix} a_1^{j,m} & 0 & \dots & 0 & 0 \\ a_2^{j,m} & a_1^{j,m} & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_p^{j,m} & a_{p-1}^{j,m} & \dots & a_{p-s+2}^{j,m} & a_{p-s+1}^{j,m} \end{bmatrix}, \quad (18)$$

where $a_i^{j,m}$ are the control plant step response coefficients describing influence of the m th input on the j th output; it can be shown that the dynamic matrix has the same form regardless a type of linear models MPC algorithms are based on [7].

After applying the prediction based on a linear model (15) the performance function (1) is transformed to a function which depends quadratically on decision variables Δu :

$$J_{LMPC} = (\bar{y} - \tilde{y} - A \cdot \Delta u)^T \cdot \kappa \cdot (\bar{y} - \tilde{y} - A \cdot \Delta u) + \Delta u^T \cdot \Lambda \cdot \Delta u. \quad (19)$$

After using prediction (15) also in the constraints (3), all constraints depend linearly on decision variables. Thus, the optimization problem ((1)–(4)) becomes an, easy to solve, standard linear–quadratic optimization problem. Unfortunately, application of an LMPC algorithm to a nonlinear process may result in unsatisfactory control performance, especially if operation in different operating points is demanded. Therefore, the fuzzy MPC (FMPC) algorithm being a combination of the LMPC and NMPC algorithms was proposed in Reference [36]. Now it will be improved by appropriate usage of the nonlinear model. Formulation of the proposed algorithm is presented in the next section. It still uses two models the original, nonlinear one is used to calculate the free response in an advanced way. The fuzzy one, composed of a set of a few step responses, is applied to calculate the dynamic matrix, which is updated at each iteration of the algorithm.

3. Fuzzy MPC Algorithm with Advanced Generation of Prediction

In order to avoid algorithm complications resulting from the need to solve a nonlinear optimization task in each iteration, one can use approximation of the process model carried out in each iteration. Using an easy–to–obtain fuzzy model to get the dynamic matrix and a nonlinear model to obtain the advanced free response leads to prediction which can be generated in a relatively simple way, is linear relative to decision variables, and mimics the nonlinearity of the process very well. Next, in each iteration of the algorithm, an optimization problem ((1)–(4)) which uses this prediction is formulated as a quadratic optimization problem. Such a problem is easy to solve using commonly available optimization routines. Now it will be described how to obtain such an advanced prediction.

3.1. Generation of the Dynamic Matrix

The dynamic matrix, needed to obtain forced response in the prediction can be calculated using an easy–to–obtain fuzzy Takagi–Sugeno model which has local models in the form of step responses [34]. Design process of such a model is very simple, because it is sufficient to collect a few sets of step responses (near a few operating points). These step responses can be obtained directly on the control plant or using a nonlinear model of the process. Next, the premises of the model should be formulated

using expert knowledge, and then tuned, for example, by means of a fuzzy neural network [7]. Thus, the model has the following form:

Rule f : (20)

$$\begin{aligned} &\text{if } y_k^{j_y} \text{ is } B_1^{f,j_y} \text{ and } \dots \text{ and } y_{k-n+1}^{j_y} \text{ is } B_n^{f,j_y} \text{ and} \\ &\quad u_k^{j_u} \text{ is } C_1^{f,j_u} \text{ and } \dots \text{ and } u_{k-m+1}^{j_u} \text{ is } C_m^{f,j_u} \\ &\text{then } \hat{y}_k^{j,f} = \sum_{m=1}^{n_u} \sum_{n=1}^{p_d-1} a_n^{j,m,f} \cdot \Delta u_{k-n}^m + a_{p_d}^{j,m,f} \cdot u_{k-p_d}^m, \end{aligned}$$

where $y_k^{j_y}$ is the value of j_y th output variable at the k th sampling instant, $u_k^{j_u}$ is the value of j_u th manipulated variable at the k th sampling instant, $B_1^{f,j_y}, \dots, B_n^{f,j_y}, C_1^{f,j_u}, \dots, C_m^{f,j_u}$ are fuzzy sets, $a_n^{j,m,f}$ are the coefficients of step responses in the f th local model, $j_y = 1, \dots, n_y$, $j_u = 1, \dots, n_u$, $f = 1, \dots, l$, l is number of fuzzy rules. The output values of the model (20) can be calculated using the following formula:

$$\hat{y}_k^j = \sum_{m=1}^{n_u} \sum_{n=1}^{p_d-1} \tilde{a}_n^{j,m} \cdot \Delta u_{k-n}^m + \tilde{a}_{p_d}^{j,m} \cdot u_{k-p_d}^m, \quad (21)$$

where $\tilde{a}_n^{j,m} = \sum_{f=1}^l \tilde{w}_f \cdot a_n^{j,m,f}$ and \tilde{w}_f are the normalized firing strengths of the fuzzy rules calculated using fuzzy reasoning, see for example, References [43,44]. Note that one can consider the model (21) as the step response describing behavior of the control plant near the current operating point. This model is then used at each sampling instant of the proposed algorithms to obtain a new, updated dynamic matrix:

$$A_k = \begin{bmatrix} A_k^{11} & A_k^{12} & \dots & A_k^{1n_u} \\ A_k^{21} & A_k^{22} & \dots & A_k^{2n_u} \\ \vdots & \vdots & \ddots & \vdots \\ A_k^{n_y 1} & A_k^{n_y 2} & \dots & A_k^{n_y n_u} \end{bmatrix}, \quad (22)$$

$$A_k^{jm} = \begin{bmatrix} \tilde{a}_1^{j,m} & 0 & \dots & 0 & 0 \\ \tilde{a}_2^{j,m} & \tilde{a}_1^{j,m} & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \tilde{a}_p^{j,m} & \tilde{a}_{p-1}^{j,m} & \dots & \tilde{a}_{p-s+2}^{j,m} & \tilde{a}_{p-s+1}^{j,m} \end{bmatrix}. \quad (23)$$

3.2. Generation of the Free Response

The basic idea of the free response generation for an MPC algorithm is to use a non-linear process model assuming that the control signals do not change over the entire prediction horizon—a classic free response; see for example, Reference [36]. In the case of the proposed algorithm, future control increments calculated in the algorithm in the previous iterations are also used when the free response is calculated. Thus, more information is exploited to obtain the improved prediction. Assume that future control values are decomposed as follows:

$$u_{k+i|k}^m = \check{u}_{k+i|k}^m + u_{k+i|k-1}^m, \quad (24)$$

where $\check{u}_{k+i|k}^m$ can be interpreted as a corrections of the control signal $u_{k+i|k-1}^m$ obtained in the last $(k-1)$ st iteration. Similarly, future control increments will be given by the formula:

$$\Delta u_{k+i|k}^m = \Delta \check{u}_{k+i|k}^m + \Delta u_{k+i|k-1}^m. \quad (25)$$

Assume

$$\mathbf{u}_{k+i|k-1} = \left[u_{k+i|k-1}^1, \dots, u_{k+i|k-1}^{n_u} \right]^T, \quad (26)$$

In this case, using the nonlinear model, the free response is iteratively derived as follows:

- First, the nonlinear model (12) is used to obtain

$$\hat{\mathbf{y}}_{k+1} = f(\mathbf{y}_k, \mathbf{y}_{k-1}, \dots, \mathbf{y}_{k-n_a+1}, \mathbf{u}_{k|k-1}, \mathbf{u}_{k-1}, \dots, \mathbf{u}_{k-n_b+1}); \quad (27)$$

- Next, the values $\hat{\mathbf{y}}_{k+1}$ are used when calculating the output values $\hat{\mathbf{y}}_{k+2}$ for the next sampling instant:

$$\hat{\mathbf{y}}_{k+2} = f(\hat{\mathbf{y}}_{k+1}, \mathbf{y}_k, \dots, \mathbf{y}_{k-n_a+2}, \mathbf{u}_{k+1|k-1}, \mathbf{u}_{k|k-1}, \mathbf{u}_{k-1}, \dots, \mathbf{u}_{k-n_b+2}); \quad (28)$$

- For the i th iteration, using the values $\hat{\mathbf{y}}_{k+1}, \dots, \hat{\mathbf{y}}_{k+i-1}$ one obtains:

$$\hat{\mathbf{y}}_{k+i} = f(\hat{\mathbf{y}}_{k+i-1}, \hat{\mathbf{y}}_{k+i-2}, \dots, \mathbf{y}_{k-n_a+i}, \mathbf{u}_{k+i-1|k-1}, \mathbf{u}_{k+i-2|k-1}, \dots, \mathbf{u}_{k-n_b+i}). \quad (29)$$

- After taking into account the estimation of unmeasured disturbances $\mathbf{d}_k = \mathbf{y}_k - \hat{\mathbf{y}}_{k|k-1}$, containing also influence of modeling errors, the final form of the formula describing the elements of the free response is obtained:

$$\tilde{\mathbf{y}}_{k+i|k} = \hat{\mathbf{y}}_{k+i} + \mathbf{d}_k. \quad (30)$$

3.3. Formulation of the Optimization Problem

Now the dynamic matrix (22) will be used to predict the influence of control corrections on the control plant outputs. Therefore, after using the free response (30) and the dynamic matrix (22) one obtains the following prediction:

$$\mathbf{y} = \tilde{\mathbf{y}} + \mathbf{A}_k \cdot \Delta \tilde{\mathbf{u}}. \quad (31)$$

Due to the change of decision variables, optimization task ((1)–(4)), which is solved by the control algorithm in each iteration can be written as:

$$\arg \min_{\Delta \tilde{\mathbf{u}}} \left\{ J_{MPC+} = (\bar{\mathbf{y}} - \tilde{\mathbf{y}} - \mathbf{A}_k \cdot \Delta \tilde{\mathbf{u}})^T \cdot \boldsymbol{\kappa} \cdot (\bar{\mathbf{y}} - \tilde{\mathbf{y}} - \mathbf{A}_k \cdot \Delta \tilde{\mathbf{u}}) + \Delta \mathbf{u}^T \cdot \boldsymbol{\Lambda} \cdot \Delta \mathbf{u} \right\} \quad (32)$$

subject to:

$$\Delta \mathbf{u}_{\min} \leq \Delta \mathbf{u} \leq \Delta \mathbf{u}_{\max}, \quad (33)$$

$$\mathbf{u}_{\min} \leq \mathbf{u} \leq \mathbf{u}_{\max}, \quad (34)$$

$$\mathbf{y}_{\min} \leq \tilde{\mathbf{y}} + \mathbf{A}_k \cdot \Delta \tilde{\mathbf{u}} \leq \mathbf{y}_{\max}, \quad (35)$$

where

$$\Delta \tilde{\mathbf{u}} = \left[\Delta \tilde{\mathbf{u}}^1, \dots, \Delta \tilde{\mathbf{u}}^{n_u} \right]^T, \quad \Delta \tilde{\mathbf{u}}^m = \left[\Delta \tilde{u}_{k+1|k}^m, \dots, \Delta \tilde{u}_{k+s-1|k}^m \right]. \quad (36)$$

Note that the performance function in (32) depends quadratically on decision variables $\Delta \tilde{\mathbf{u}}$ and all constraints depend linearly on decision variables. Thus, a standard, easy-to-solve linear-quadratic optimization problem was obtained.

Remark 1. In the proposed algorithm, one can also use a slightly modified form of the performance index, in which the second component depends only on the corrections $\Delta \tilde{\mathbf{u}}$ of the control signals determined in the previous iteration of the algorithm, that is,

$$J_{MPC+v2} = (\bar{\mathbf{y}} - \tilde{\mathbf{y}} - \mathbf{A}_k \cdot \Delta \tilde{\mathbf{u}})^T \cdot \boldsymbol{\kappa} \cdot (\bar{\mathbf{y}} - \tilde{\mathbf{y}} - \mathbf{A}_k \cdot \Delta \tilde{\mathbf{u}}) + \Delta \tilde{\mathbf{u}}^T \cdot \boldsymbol{\Lambda} \cdot \Delta \tilde{\mathbf{u}}. \quad (37)$$

It should be emphasized, however, that such modification causes a change in the meaning of the λ parameter in relation to the classical formulation. Consequently, algorithms with a modified form of the performance function generate faster responses, but influence of the λ parameter on system robustness becomes significantly limited.

3.4. Iterative Improvement of the Prediction

In practice, the proposed approach gives results very similar to those offered by the NMPC algorithm. However it is possible to further improve the control quality, if necessary, thanks to iterative improvement of the prediction determined in the algorithm. Then smaller corrections of control trajectories in each subsequent iteration are obtained, what is beneficial because impact of them on the process outputs is predicted using the dynamic matrix instead of a more accurate non-linear process model.

The first, internal iteration of the algorithm is the same as in the basic version of the algorithm. After receiving the future control trajectories, one can either apply them in the process or try to improve the resulting solution. In such a case, the second internal iteration of the algorithm should be executed. Denote the values of future control signal trajectories obtained in the first iteration as $u_{k+i|k(1)}^m$ and future control increments as $\Delta u_{k+i|k(1)}^m$.

In subsequent iterations, because subsequent control signal trajectories are available, future control values are decomposed as follows:

$$u_{k+i|k(i_w)}^m = \check{u}_{k+i|k(i_w)}^m + u_{k+i|k(i_w-1)}^m, \quad (38)$$

where i_w —index of the internal iteration of the algorithm, $i_w = 2, 3, \dots$. Similarly, future control increments are decomposed:

$$\Delta u_{k+i|k(i_w)}^m = \Delta \check{u}_{k+i|k(i_w)}^m + \Delta u_{k+i|k(i_w-1)}^m. \quad (39)$$

Assume

$$\mathbf{u}_{k+i|k(i_w-1)} = \left[u_{k+i|k(i_w-1)}^1, \dots, u_{k+i|k(i_w-1)}^{n_u} \right]^T, \quad (40)$$

Thus, using the non-linear process model, the free response is iteratively derived as follows:

- First, the model is used to obtain values

$$\hat{\mathbf{y}}_{k+1} = f(\mathbf{y}_k, \mathbf{y}_{k-1}, \dots, \mathbf{y}_{k-n_a+1}, \mathbf{u}_{k|k(i_w-1)}, \mathbf{u}_{k-1}, \dots, \mathbf{u}_{k-n_b+1}); \quad (41)$$

- Next, the values $\hat{\mathbf{y}}_{k+1}$ are used to calculate the output values $\hat{\mathbf{y}}_{k+2}$ for the next sampling instant:

$$\hat{\mathbf{y}}_{k+2} = f(\hat{\mathbf{y}}_{k+1}, \mathbf{y}_k, \dots, \mathbf{y}_{k-n_a+2}, \mathbf{u}_{k+1|k(i_w-1)}, \mathbf{u}_{k|k(i_w-1)}, \mathbf{u}_{k-1}, \dots, \mathbf{u}_{k-n_b+2}); \quad (42)$$

- For the i th iteration, using the values $\hat{\mathbf{y}}_{k+1}, \dots, \hat{\mathbf{y}}_{k+i-1}$ one obtains:

$$\hat{\mathbf{y}}_{k+i} = f(\hat{\mathbf{y}}_{k+i-1}, \hat{\mathbf{y}}_{k+i-2}, \dots, \mathbf{y}_{k-n_a+i}, \mathbf{u}_{k+i-1|k(i_w-1)}, \mathbf{u}_{k+i-2|k(i_w-1)}, \dots, \mathbf{u}_{k-n_b+i}). \quad (43)$$

- After taking into account the estimation of unmeasured disturbances $\mathbf{d}_k = \mathbf{y}_k - \hat{\mathbf{y}}_{k|k-1}$, containing also influence of modeling errors, like in (30), the final formula describing the elements of the free response is obtained:

$$\tilde{\mathbf{y}}_{k+i|k} = \hat{\mathbf{y}}_{k+i} + \mathbf{d}_k. \quad (44)$$

As in the basic algorithm, the dynamic matrix is used to predict the influence of control corrections on the control plant outputs. Thus, after using the free response (44) and the dynamic matrix (22) one obtains the following prediction:

$$\mathbf{y} = \tilde{\mathbf{y}} + \mathbf{A}_k \cdot \Delta \check{\mathbf{u}}. \quad (45)$$

where

$$\Delta \tilde{u} = [\Delta \tilde{u}^1, \dots, \Delta \tilde{u}^{n_u}]^T, \Delta \tilde{u}^m = [\Delta \tilde{u}_{k+1|k(i_w)}^m, \dots, \Delta \tilde{u}_{k+s-1|k(i_w)}^m]. \quad (46)$$

This prediction should then be used to formulate the optimization task in the same way as in the basic algorithm. After solving this task, one can either use the result to control the process or to perform another internal iteration of the algorithm.

Note that the lower the values of the corrections of the future control trajectories are obtained, the better the prediction is obtained, because it will be closer to the one obtained fully using a nonlinear model. The number of internal iterations of the algorithm can therefore be selected depending on how large the corrections of the future control trajectories are (provided that the time for generating control signals has not expired). If these adjustments become small enough, one should stop doing further internal iterations.

Due to the lack of proof of the convergence of the described procedure, it is advisable to check after each internal iteration whether the performance function in the problem solved by the algorithm and control corrections decrease. If there is no improvement in the value of the performance function (especially when there is a large increase of it) or control adjustments begin to increase the internal iterations should be interrupted.

3.5. Fast Generation of the Control Action—Analytical Approach

Note that if in each iteration of the proposed approach the following optimization problem without constraints is solved:

$$\arg \min_{\Delta \tilde{u}} \left\{ J_{MPC+} = (\bar{y} - \tilde{y} - A_k \cdot \Delta \tilde{u})^T \cdot \kappa \cdot (\bar{y} - \tilde{y} - A_k \cdot \Delta \tilde{u}) + \Delta u^T \cdot \Lambda \cdot \Delta u \right\}$$

then it has the following solution given by the analytical formula:

$$\Delta \tilde{u} = \left(A_k^T \cdot \kappa \cdot A_k + \Lambda \right)^{-1} \cdot \left(A_k^T \cdot \kappa \cdot (\bar{y} - \tilde{y}) + \Delta \vec{u} \right). \quad (47)$$

where $\Delta \vec{u}$ contains elements of future control increments trajectory, corrections of which are collected in the vector $\Delta \tilde{u}$.

Remark 2. If one uses the performance index (37), that is, the modified version:

$$J_{MPC+v2} = (\bar{y} - \tilde{y} - A_k \cdot \Delta \tilde{u})^T \cdot \kappa \cdot (\bar{y} - \tilde{y} - A_k \cdot \Delta \tilde{u}) + \Delta \tilde{u}^T \cdot \Lambda \cdot \Delta \tilde{u}, \quad (48)$$

then the following analytical formula should be used:

$$\Delta \tilde{u} = \left(A_k^T \cdot \kappa \cdot A_k + \Lambda \right)^{-1} \cdot A_k^T \cdot \kappa \cdot (\bar{y} - \tilde{y}). \quad (49)$$

It should be once again emphasized, however, that this modification changes the meaning of the parameter matrix Λ comparing to the classical formulation and algorithms with a modified form of the performance index generate faster responses than the algorithm using the control law (47), but influence of the λ parameter on system robustness becomes significantly limited.

Due to the fact that the dynamic matrix A_k changes, in general, in every iteration, then in each iteration of the algorithm the formulas (47) (or (49)) allowing to calculate the control action should be used. As a consequence, appropriate rows of the matrix $(A_k^T \cdot \kappa \cdot A_k + \Lambda)^{-1}$ (corresponding to the control increments for the current sampling instant $\Delta u_{k|k}^m$) should be calculated in each iteration of the algorithm.

3.6. Disturbance Measurement Utilization

The proposed algorithms are formulated in such a way that the mechanisms of disturbance measurement utilization presented in Reference [45] can be applied in them easily. The proposed algorithms are based on two models, therefore two such mechanisms can be applied. They will be now shortly reminded.

3.6.1. Employing Fuzzy Model

The first method consists in extending local models of the fuzzy model (20) using the models of disturbance influence on the control plant in the form of step responses. It results then in extending the prediction (31) with a term dependent on measured disturbance; for details see References [34,45]:

$$\mathbf{y} = \tilde{\mathbf{y}} + \mathbf{E}_k \cdot \Delta \mathbf{u}^d + \mathbf{A}_k \cdot \Delta \tilde{\mathbf{u}} \quad , \quad (50)$$

where

$$\mathbf{E}_k = \begin{bmatrix} \mathbf{E}_k^1 \\ \mathbf{E}_k^2 \\ \vdots \\ \mathbf{E}_k^{n_y} \end{bmatrix}, \mathbf{E}_k^j = \begin{bmatrix} \tilde{e}_1^j & \tilde{e}_2^j - \tilde{e}_1^j & \tilde{e}_3^j - \tilde{e}_2^j & \dots & \tilde{e}_{p_z-1}^j - \tilde{e}_{p_z-2}^j & \tilde{e}_{p_z}^j - \tilde{e}_{p_z-1}^j \\ \tilde{e}_2^j & \tilde{e}_3^j - \tilde{e}_1^j & \tilde{e}_4^j - \tilde{e}_2^j & \dots & \tilde{e}_{p_z}^j - \tilde{e}_{p_z-2}^j & \tilde{e}_{p_z}^j - \tilde{e}_{p_z-1}^j \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \tilde{e}_p^j & \tilde{e}_{p+1}^j - \tilde{e}_1^j & \tilde{e}_{p+2}^j - \tilde{e}_2^j & \dots & \tilde{e}_{p_z}^j - \tilde{e}_{p_z-2}^j & \tilde{e}_{p_z}^j - \tilde{e}_{p_z-1}^j \end{bmatrix},$$

and

$$\Delta \mathbf{u}^d = \left[\Delta u_k^d, \Delta u_{k-1}^d, \Delta u_{k-2}^d, \dots, \Delta u_{k-p_z+2}^d, \Delta u_{k-p_z+1}^d \right]^T,$$

where $\tilde{e}_n^j = \sum_{f=1}^l \tilde{w}_f \cdot e_i^{j,f}$, $e_i^{j,f}$ ($i = 1, \dots, p_z$, $j = 1, \dots, n_y$, $f = 1, \dots, l$) are the coefficients of step responses to the change of disturbance, in the f th local model, u_{k-i}^d are values of the disturbance measured in the $(k-i)$ th sampling instant.

Then the following performance function in the optimization problem (1)–(4) should be now used:

$$\begin{aligned} J_{MPC+} = & (\bar{\mathbf{y}} - \tilde{\mathbf{y}} - \mathbf{E}_k \cdot \Delta \mathbf{u}^d - \mathbf{A}_k \cdot \Delta \tilde{\mathbf{u}})^T \cdot \boldsymbol{\kappa} \cdot (\bar{\mathbf{y}} - \tilde{\mathbf{y}} - \mathbf{E}_k \cdot \Delta \mathbf{u}^d - \mathbf{A}_k \cdot \Delta \tilde{\mathbf{u}}) \\ & + \Delta \mathbf{u}^T \cdot \boldsymbol{\Lambda} \cdot \Delta \mathbf{u} \quad . \end{aligned} \quad (51)$$

or the modified one:

$$\begin{aligned} J_{MPC+v2} = & (\bar{\mathbf{y}} - \tilde{\mathbf{y}} - \mathbf{E}_k \cdot \Delta \mathbf{u}^d - \mathbf{A}_k \cdot \Delta \tilde{\mathbf{u}})^T \cdot \boldsymbol{\kappa} \cdot (\bar{\mathbf{y}} - \tilde{\mathbf{y}} - \mathbf{E}_k \cdot \Delta \mathbf{u}^d - \mathbf{A}_k \cdot \Delta \tilde{\mathbf{u}}) \\ & + \Delta \tilde{\mathbf{u}}^T \cdot \boldsymbol{\Lambda} \cdot \Delta \tilde{\mathbf{u}} \quad . \end{aligned} \quad (52)$$

In the example presented in the next section, this mechanism was simplified by assuming that only one model of disturbance influence on the process is obtained, that is, $e_i^{j,f} = e_i^j$ for $f = 1, \dots, l$, then the prediction transforms to the following formula:

$$\mathbf{y} = \tilde{\mathbf{y}} + \mathbf{E} \cdot \Delta \mathbf{u}^d + \mathbf{A}_k \cdot \Delta \tilde{\mathbf{u}} \quad , \quad (53)$$

where

$$\mathbf{E} = \begin{bmatrix} \mathbf{E}^1 \\ \mathbf{E}^2 \\ \vdots \\ \mathbf{E}^{n_y} \end{bmatrix}, \mathbf{E}^j = \begin{bmatrix} e_1^j & e_2^j - e_1^j & e_3^j - e_2^j & \dots & e_{p_z-1}^j - e_{p_z-2}^j & e_{p_z}^j - e_{p_z-1}^j \\ e_2^j & e_3^j - e_1^j & e_4^j - e_2^j & \dots & e_{p_z}^j - e_{p_z-2}^j & e_{p_z}^j - e_{p_z-1}^j \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ e_p^j & e_{p+1}^j - e_1^j & e_{p+2}^j - e_2^j & \dots & e_{p_z}^j - e_{p_z-2}^j & e_{p_z}^j - e_{p_z-1}^j \end{bmatrix}.$$

and the matrix E is constant in each iteration, what simplifies the calculations. Thus, disturbance measurement utilization mechanism becomes actually the same as in the DMC algorithm.

3.6.2. Employing Nonlinear Model

The second approach assumes that the disturbance is one of the inputs of the model used to obtain the free response:

$$\tilde{u}_{k-i} = [u_{k-i}^1, \dots, u_{k-i}^{n_u}, u_{k-i}^d]^T, \quad (54)$$

where u_{k-i}^d is value of disturbance in the $(k-i)$ th sampling instant. Then the nonlinear process model (12) becomes:

$$\hat{y}_{k+1|k} = f(y_k, y_{k-1}, \dots, y_{k-n_a}, \tilde{u}_{k-1}, \tilde{u}_{k-2}, \dots, \tilde{u}_{k-n_b}). \quad (55)$$

Then it is sufficient to use measured values of the disturbance during calculation of the free response to include this information in the prediction. During calculation of the free response, as future values of the disturbance its estimates can be used or, if one does not have any, they should be assumed unchanged, like in the case of classical free response.

4. Example

The control plant is a nonlinear, isothermal CSTR in which the van de Vusse reaction takes place. The steady-state characteristic of the control plant is depicted in Figure 1.

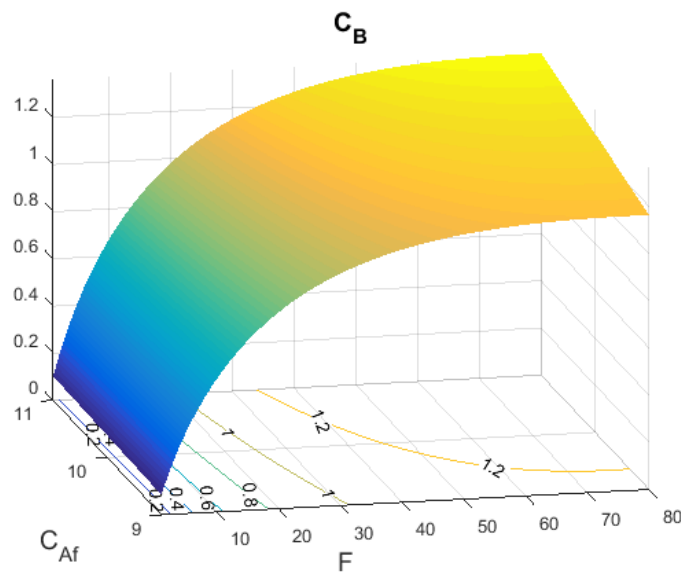


Figure 1. Steady-state characteristic of the control plant.

The CSTRs in which the van de Vusse reaction takes place are a popular benchmark due to their nonlinearity and difficult dynamics, willingly used to test newly developed control algorithms; see for example, References [46–51]. The process model of the reactor is composed of two composition balance equations; see for example, Reference [52]:

$$\begin{aligned} \frac{dC_A}{dt} &= -k_1 \cdot C_A - k_3 \cdot C_A^2 + \frac{F}{V} (C_{Af} - C_A), \\ \frac{dC_B}{dt} &= k_1 \cdot C_A - k_2 \cdot C_B - \frac{F}{V} C_B, \end{aligned} \quad (56)$$

where C_A , C_B are the concentrations of components A and B, respectively, F is the inlet (and also outlet) flow rate, V is the volume in which the reaction takes place (assumed constant and $V = 1$ L), C_{Af} is the concentration of component A in the inlet flow stream (if not declared otherwise, it is assumed

that $C_{Af} = 10$ mol/L). The values of parameters are: $k_1 = 50$ 1/h, $k_2 = 100$ 1/h, $k_3 = 10$ L/(h · mol). The output variable is the concentration C_B of substance B, the manipulated variable is the inlet flow rate F , C_{Af} is the disturbance variable. The control plant has the inverse response, thus it is a control plant difficult to control using a standard algorithm. Therefore, it is natural to use an MPC algorithm in this case.

Experiments

For the considered control plant a few MPC algorithms were designed—an NMPC one based on the nonlinear model and nonlinear optimization, an LMPC one based on a linear model in the form of a step response (of DMC type) and the proposed FMPC algorithms using a fuzzy model, different methods of prediction improvements, and different forms of the performance index (FMPC1—with classical performance index, and FMPC2—with the modified performance index). The fuzzy model used in the FMPC algorithms is composed of three step responses obtained near the following operating points:

- R1 $C_{B0} = 0.91$ mol/L, $C_{A0} = 2.18$ mol/L, $F = 20$ L/h;
- R2 $C_{B0} = 1.12$ mol/L, $C_{A0} = 3$ mol/L, $F = 34.3$ L/h;
- R3 $C_{B0} = 1.22$ mol/L, $C_{A0} = 3.66$ mol/L, $F = 50$ L/h.

The model used to obtain the dynamic matrix is thus composed of three rules; the membership functions, same as in Reference [34], chosen after analysis of the steady-state characteristic of the control plant, are shown in Figure 2.

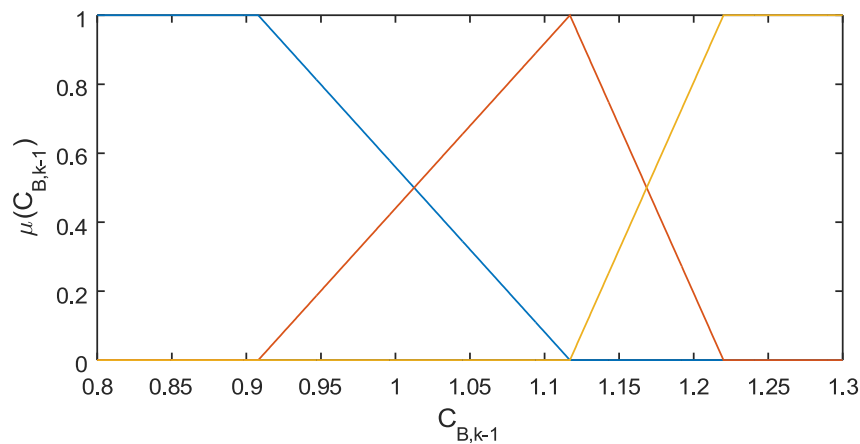


Figure 2. Membership functions of the fuzzy model used to obtain the dynamic matrix.

The simulation experiments were done using Matlab. The sampling time equal to $T_s = 3.6$ s was assumed; the values of tuning parameters were as follows: prediction horizon $p = 70$, control horizon $s = 35$, and at the beginning weighting coefficient $\lambda = 0.001$. During the experiments operation of control systems with NMPC, LMPC and FMPC algorithms was compared. The LMPC algorithm was designed using the step response obtained near the R2 point. The FMPC algorithms use two models: nonlinear one in the form of state equations, to generate a free response and the fuzzy one with local models in the form of step responses, to obtain the dynamic matrix.

Responses of the control system to changes in the setpoint and to the change of the disturbance by 10% in the 6th minute of the experiment are shown in Figure 3. If the setpoint was changed to $\bar{C}_{B1} = 1$ mol/L, responses obtained in the control system with the FMPC1 algorithm (blue lines in Figure 3) are very close to those obtained with NMPC algorithm with nonlinear optimization (red lines in Figure 3). In both cases there is almost no overshoot and both algorithms work much better than their LMPC counterpart (magenta lines in Figure 3). In the latter case there is a significant overshoot.

Moreover, settling time is much longer than in the case of NMPC and FMPC algorithms. The FMPC1 algorithm is slightly slower than the NMPC one, whereas the FMPC2 is practically as fast as NMPC.

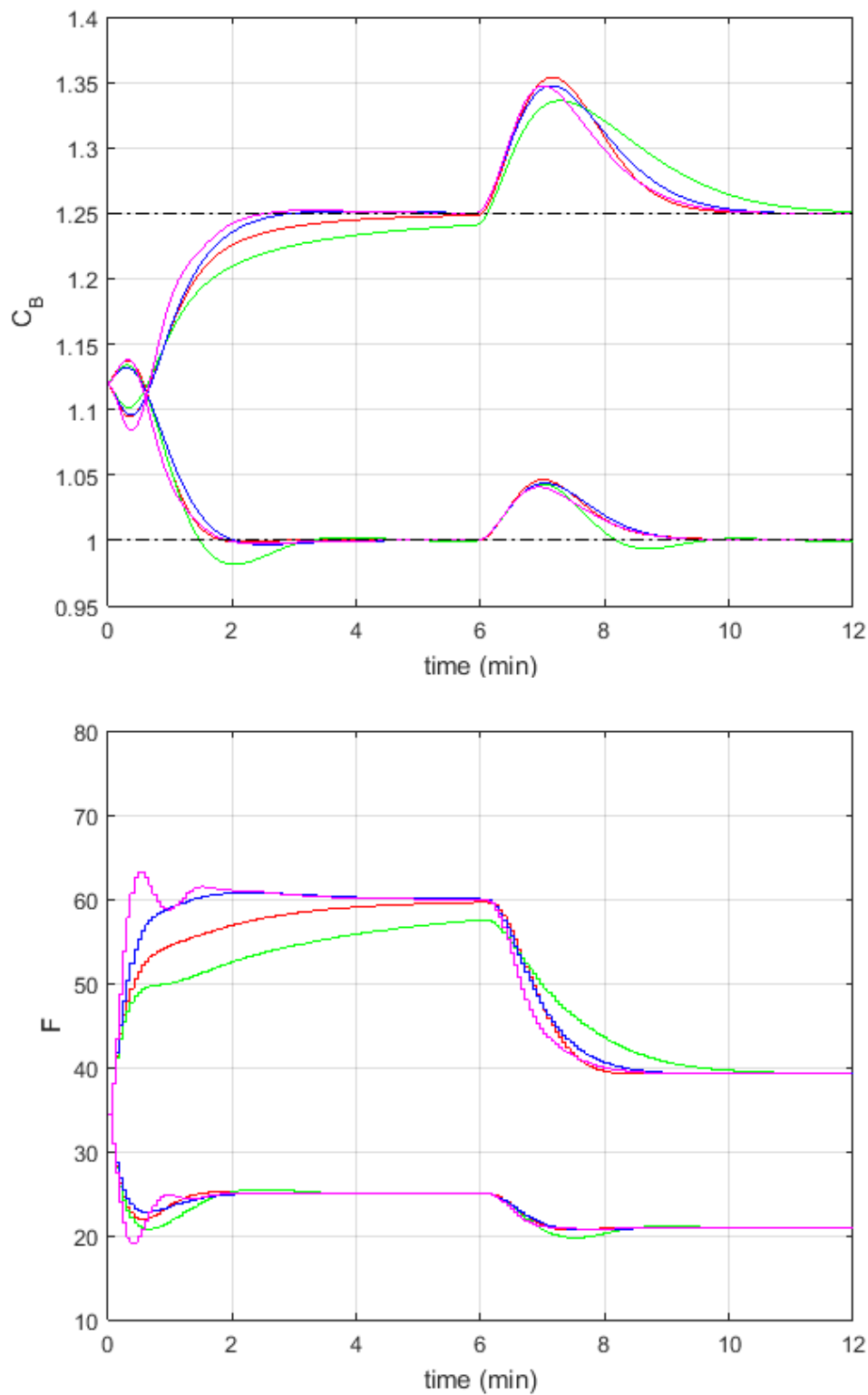


Figure 3. Responses of the control system to changes in the setpoint to $\bar{C}_{B1} = 1$ mol/L and $\bar{C}_{B2} = 1.25$ mol/L and to the change of the disturbance by 10% in the 6th minute of the experiment; $\lambda = 0.001$; Nonlinear Model Predictive Control (NMPC)—red lines, linear MPC (LMPC)—green lines, fuzzy MPS (FMPC)1—blue lines, FMPC2—magenta lines.

In the case when setpoint changed to $\bar{C}_{B2} = 1.25$ mol/L, the responses obtained with the FMPC algorithms are even faster than the one generated with NMPC algorithm. In the case of all the algorithms the setpoint is reached without overshoot, but LMPC works much slower than the algorithms based on nonlinear models.

Disturbance responses obtained near $C_B = 1$ mol/L with FMPC and NMPC algorithms are almost the same. There is no overshoot which is present in the response of the LMPC algorithm. The algorithms based on nonlinear models slightly faster compensate the change in disturbance. In the case of operation near $C_B = 1.25$ mol/L, the NMPC is the fastest in compensation of the disturbance change. The FMPC1 is slightly slower and the LMPC is much slower than its counterparts.

It can be noticed that the FMPC algorithms offer control performance close to the one which can be obtained with the NMPC algorithm and with nonlinear optimization. In one case (setpoint change to $\bar{C}_{B2} = 1.25$ mol/L) they are even faster. It should be, however, emphasized that the FMPC algorithms use reliable quadratic programming routine to generate control action.

Duration of action of the researched algorithms in the experiments depicted in Figure 3 is compared in Table 1. For each algorithm and for each setpoint five measurements of duration were done (time needed to simulate the control plant was subtracted) and averaged. Then these average durations were summed in the last row of the table. Both FDMC algorithms work much (around 20 times) faster than the NMPC algorithm.

Table 1. Average duration of action of the algorithms in the experiments depicted in Figure 3.

\bar{C}_B	NMPC	FMPC1	FMPC2
1.00 mol/L	29.1671	1.8175	1.8118
1.25 mol/L	47.0518	1.8116	1.8137
sum	76.2189	3.6291	3.6254

There was also done an experiment with $\lambda = 0.0001$ (10 times smaller than in the first experiment; Figure 4). All the algorithms work now faster; the control action in all cases is more aggressive, but the relations between FMPC, NMPC, and LMPC algorithms remain unchanged; they are the same as in the previous experiment, for $\lambda = 0.001$. However, as the λ parameter is closer to zero, the differences between FMPC1 and FMPC2 algorithms became much smaller comparing to the previous experiment. The FMPC1 algorithm is practically as fast as the FMPC2 one. Such a phenomenon is reasonable, because for $\lambda = 0$ both performance indexes (32) and (37) are the same. Thus, the closer value of the λ coefficient to 0 is, the closer responses generated with FMPC1 and FMPC2 are.

There were also done experiments with shortening the control horizon s (Figures 5 and 6). Note, that the smaller the control horizon is the less number of decision variables in the optimization problem. Therefore, it is simpler to solve and can be solved faster. In the case of both algorithms FMPC1 and FMPC2, with different values of λ for many values of the control horizon the responses generated with the algorithms do not change too much. Thus, the control horizon can be shorted significantly. In the considered examples for FMPC1 algorithm the responses generated for $s = 35$ and for $s = 10$ are almost the same; in the case of control system with FMPC2 algorithm—the responses generated for $s = 35$ and for $s = 5$ are almost the same. Further decreasing of the control horizon brings changes in the responses.

In the case of the control system with FMDC1 algorithm, for $s = 5$ the disturbance compensation near $C_B = 1.25$ mol/L becomes faster, but unfortunately achieving the setpoint $\bar{C}_{B1} = 1$ mol/L lasts longer. Further decrease of the control horizon to $s = 1$ causes that the disturbance compensation near $C_B = 1.25$ mol/L becomes even faster as well as reaching the setpoint $\bar{C}_{B2} = 1.25$ mol/L, however at the cost of slower response to the setpoint change to $\bar{C}_{B1} = 1$ mol/L and worsened disturbance compensation near $C_B = 1$ mol/L.

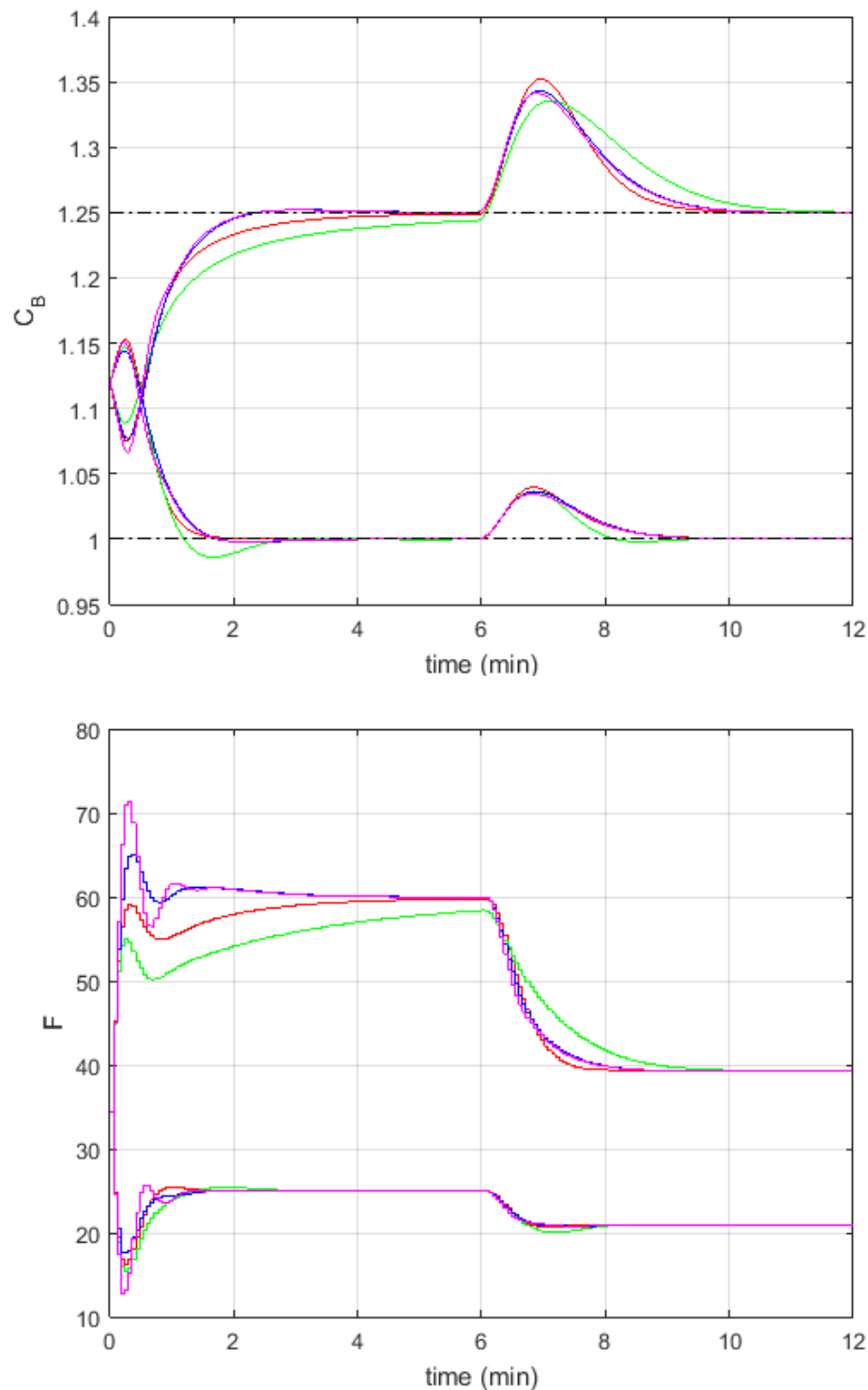


Figure 4. Responses of the control system to changes in the setpoint to $\bar{C}_{B1} = 1$ mol/L and $\bar{C}_{B2} = 1.25$ mol/L and to the change of the disturbance by 10% in the 6th minute of the experiment; $\lambda = 0.0001$; NMPC—red lines, LMPC—green lines, FMPC1—blue lines, FMPC2—magenta lines.

In the control system with FMDC2 algorithm, for $s = 2$ and $s = 1$ the disturbance compensation near $C_B = 1.25$ mol/L becomes slightly faster, but unfortunately achieving both the setpoints $\bar{C}_{B1} = 1$ mol/L and $\bar{C}_{B2} = 1.25$ mol/L lasts longer. Moreover disturbance compensation near $C_B = 1$ mol/L is now slower. Thus, in this case, as in the case of FMPC1 algorithm, decreasing

the control horizon should be done also carefully and there is also a threshold which should not be crossed in order to avoid control performance deterioration.

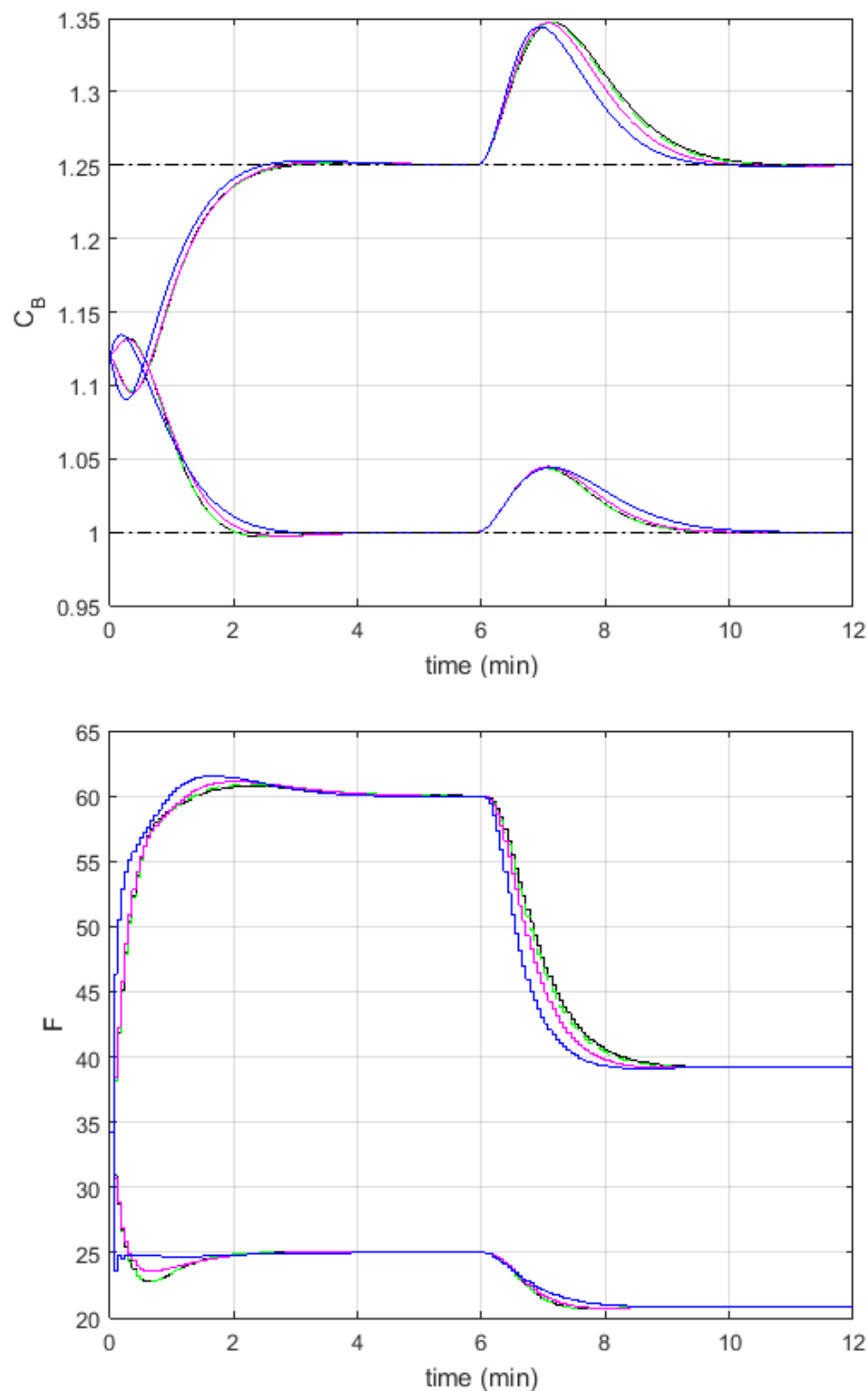


Figure 5. Responses of the control system with FMPC1 algorithm and $\lambda = 0.001$ to changes in the setpoint to $\bar{C}_{B1} = 1$ mol/L and $\bar{C}_{B2} = 1.25$ mol/L and to the change of the disturbance by 10% in the 6th minute of the experiment; $s = 35$ —black lines, $s = 10$ —green dashed lines, $s = 5$ —magenta lines, $s = 1$ —blue lines.

To sum up, during designing of an algorithm it is advisable to check if the control horizon can be decreased (and optimization problem solved by the algorithm simplified). However, it should be done carefully, because for too short control horizon control performance can get worse.

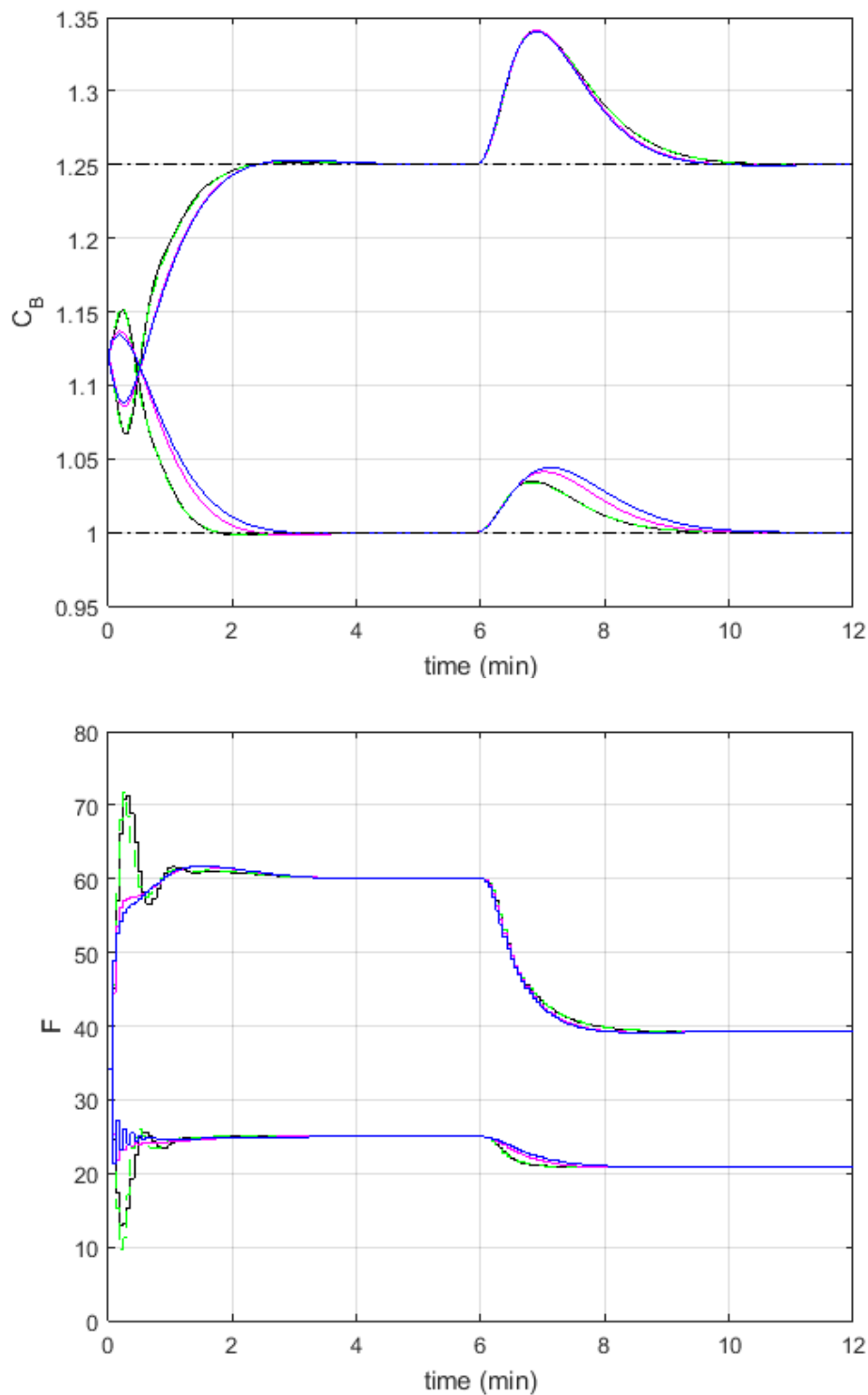


Figure 6. Responses of the control system with FMPC2 algorithm and $\lambda = 0.001$ to changes in the setpoint to $\bar{C}_{B1} = 1$ mol/L and $\bar{C}_{B2} = 1.25$ mol/L and to the change of the disturbance by 10% in the 6th minute of the experiment; $s = 35$ —black lines, $s = 5$ —green dashed lines, $s = 2$ —magenta lines, $s = 1$ —blue lines.

The mechanisms of disturbance measurement utilization were used in the next experiments. Both FMPC algorithms generated similar responses. The first method of disturbance measurement utilization (red lines in Figures 7 and 8) is based on the same approach as in the standard DMC algorithm, the second one employs the nonlinear process model (magenta lines in Figures 7 and 8). Disturbance compensation near $C_B = 1.25$ mol/L is almost the same in the case of both methods; differences in responses are really small. It is because the model of disturbance influence on the control plant for the first method was obtained near this output value.

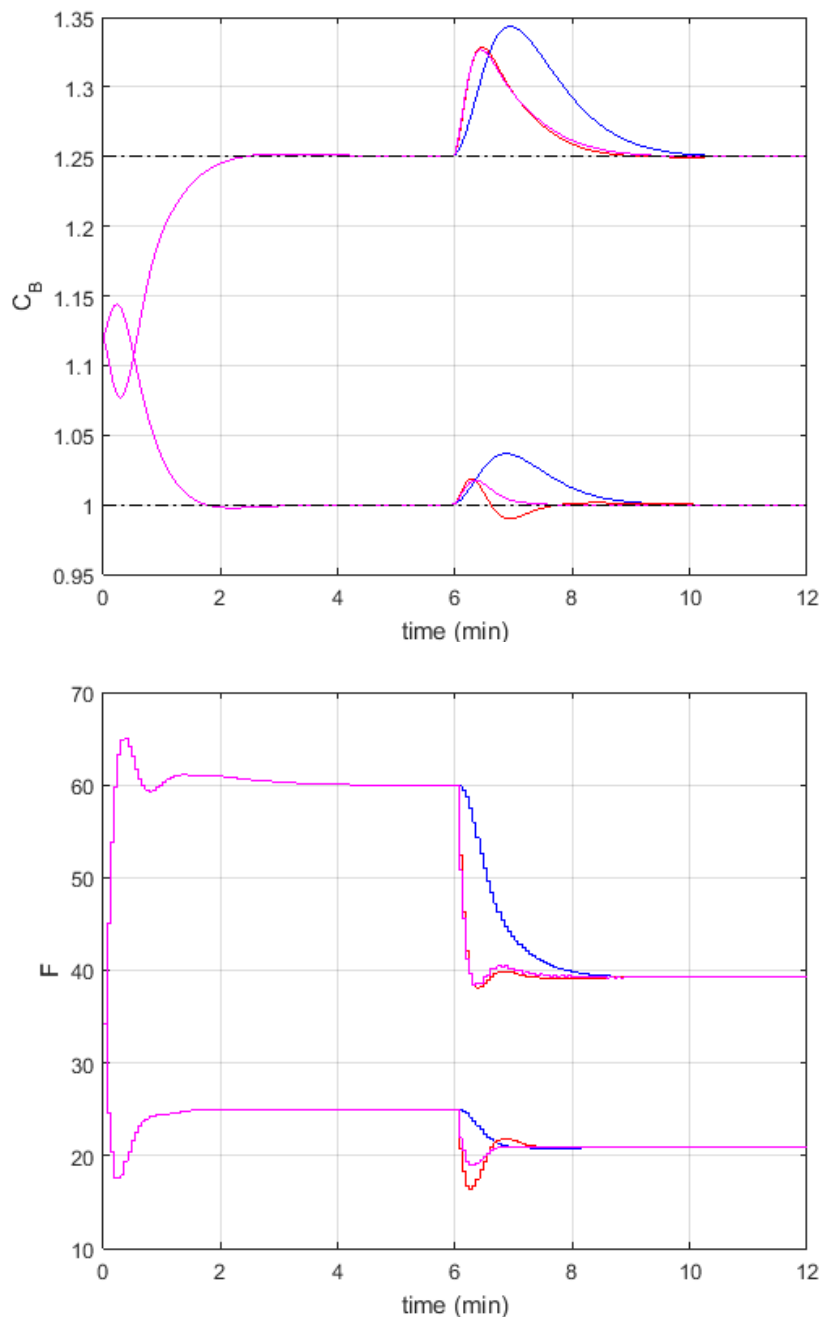


Figure 7. Responses of the control system with FMPC1 algorithm and $\lambda = 0.0001$ to changes in the setpoint to $\bar{C}_{B1} = 1$ mol/L and $\bar{C}_{B2} = 1.25$ mol/L and to the change of the disturbance by 10% in the 6th minute of the experiment; mechanism of disturbance measurement utilization: not used—blue lines, based on the linear model—red lines, based on the nonlinear model—magenta lines.

Application of both methods of disturbance measurement utilization near $C_B = 1$ mol/L causes reduction of the maximal control error by around 50% comparing to the case when disturbance measurement was not exploited (blue lines in Figures 7 and 8). Both methods offer very good disturbance compensation. However, in the case of the one based on the nonlinear model (magenta lines in Figures 7 and 8) the output value achieves the setpoint slightly faster than the one based on the linear model (red lines in Figures 7 and 8) while changes of the control signal are smaller.

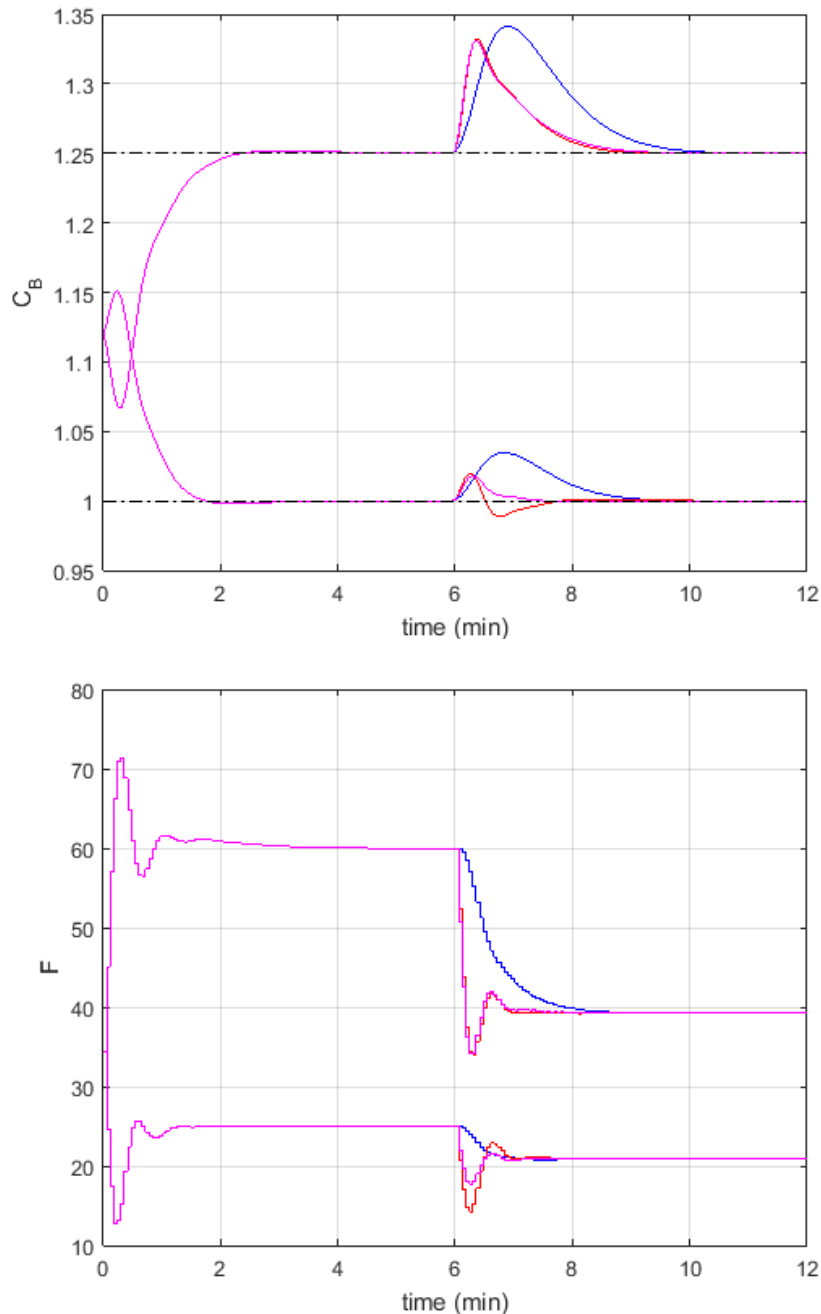


Figure 8. Responses of the control system with FMPC2 algorithm and $\lambda = 0.0001$ to changes in the setpoint to $\bar{C}_{B1} = 1$ mol/L and $\bar{C}_{B2} = 1.25$ mol/L and to the change of the disturbance by 10% in the 6th minute of the experiment; mechanism of disturbance measurement utilization: not used—blue lines, based on the linear model—red lines, based on the nonlinear model—magenta lines.

5. Conclusions

The numerically efficient FMPC algorithms proposed in the article use the easy-to-obtain fuzzy model to calculate the dynamic matrix, which describes the influence of future control action on prediction, and the nonlinear model to derive the free response of the control plant in a clever way which allows obtaining very good prediction. As a result, the algorithms offer control performance very close to the one offered by the NMPC algorithm, based on the nonlinear optimization. The FMPC algorithms, however, despite they are also based on nonlinear models, thanks to a clever way of prediction generation are formulated as the easy-to-solve quadratic optimization problem.

Modifications of the algorithms proposed to improve their operation can be applied by a system designer if needed. Thus the algorithm can be tailored to a particular application. Thus, it can offer a good balance between control performance which is needed in the problem at hand, and control algorithm complication. One can adjust the algorithm to a given problem so the computational demand is as small possible.

Funding: This research received no external funding.

Conflicts of Interest: The author declares no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

CSTR	Continuous Stirred–Tank Reactor
DMC	Dynamic Matrix Control
FMPC	Fuzzy Model Predictive Control
LMPC	Linear Model Predictive Control
LMIs	Linear Matrix Inequalities
MPC	Model Predictive Control
MIMO	Multiple–Input Multiple–Output
NMPC	Nonlinear Model Predictive Control

References

1. Camacho, E.F.; Bordons, C. *Model Predictive Control*; Springer: London, UK, 1999.
2. Domański, P.D. Performance Assessment of Predictive Control—A Survey. *Algorithms* **2020**, *13*, 97. [\[CrossRef\]](#)
3. El Youssef, J.; Castle, J.; Ward, W.K. A Review of Closed–Loop Algorithms for Glycemic Control in the Treatment of Type 1 Diabetes. *Algorithms* **2009**, *2*, 518–532. [\[CrossRef\]](#)
4. Rossiter, J.A. *Model–Based Predictive Control: A Practical Approach*; CRC Press: Boca Raton, FL, USA, 2003.
5. Maciejowski, J.M. *Predictive Control with Constraints*; Prentice Hall: Harlow, UK, 2002.
6. Sands, T. Comparison and Interpretation Methods for Predictive Control of Mechanics. *Algorithms* **2019**, *12*, 232. [\[CrossRef\]](#)
7. Tatjewski, P. *Advanced Control of Industrial Processes; Structures and Algorithms*; Springer: London, UK, 2007.
8. Tatjewski, P. Disturbance modeling and state estimation for offset–free predictive control with state–spaced process models. *Int. J. Appl. Math. Comput. Sci.* **2014**, *24*, 313–323. [\[CrossRef\]](#)
9. Ławryńczuk, M.; Marusak, P.; Tatjewski, P. Cooperation of model predictive control with steady–state economic optimisation. *Control. Cybern.* **2008**, *37*, 133–158.
10. Blevins, T.L.; McMillan, G.K.; Wojsznis, W.K.; Brown, M.W. *Advanced Control Unleashed*; ISA: Fort Belvoir, VA, USA, 2003.
11. Abdelaal, M.; Schön, S. Predictive Path Following and Collision Avoidance of Autonomous Connected Vehicles. *Algorithms* **2020**, *13*, 52. [\[CrossRef\]](#)
12. Chen, H.; Allgöwer, F. A Quasi-Infinite Horizon Nonlinear Model Predictive Control Scheme with Guaranteed Stability. *Automatica* **1998**, *34*, 1205–1217. [\[CrossRef\]](#)
13. Tatjewski, P. Offset–free nonlinear model predictive control with state–space process models. *Arch. Control. Sci.* **2017**, *27*, 595–615. [\[CrossRef\]](#)

14. Zavala, V.M.; Laird, C.D.; Biegler, L.T. A fast moving horizon estimation algorithm based on nonlinear programming sensitivity. *J. Process Control* **2008**, *18*, 876–884. [\[CrossRef\]](#)
15. Diehl, M.; Bock, H.G.; Schlöder, J.P.; Findeisen, R.; Nagy, Z.; Allgöwer, F. Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations. *J. Process Control* **2002**, *12*, 577–585. [\[CrossRef\]](#)
16. Schäfer, A.; Kühn, P.; Diehl, M.; Schlöder, J.; Bock, H.G. Fast reduced multiple shooting methods for nonlinear model predictive control. *Chem. Eng. Process.* **2007**, *46*, 1200–1214. [\[CrossRef\]](#)
17. Johansen, T.A. On multi-parametric nonlinear programming and explicit nonlinear model predictive control. In Proceedings of the 41st IEEE Conference on Decision and Control, Las Vegas, NV, USA, 10–13 December 2002; Volume 3, pp. 2768–2773.
18. Johansen, T.A. Approximate explicit receding horizon control of constrained nonlinear systems. *Automatica* **2004**, *40*, 293–300. [\[CrossRef\]](#)
19. Dominguez, L.F.; Pistikopoulos, E.N. A Novel mp-NLP Algorithm for Explicit/Multi-parametric NMPC. *IFAC Proc. Vol.* **2010**, *43*, 539–544. [\[CrossRef\]](#)
20. Bemporad, A.; Borrelli, F.; Morari, M. Piecewise linear optimal controllers for hybrid systems. In Proceedings of the 2000 American Control Conference. ACC (IEEE Cat. No.00CH36334), Chicago, IL, USA, 28–30 June 2000; pp. 1190–1194.
21. Bemporad, A.; Morari, M.; Dua, V.; Pistikopoulos, E.N. The explicit linear quadratic regulator for constrained systems. *Automatica* **2002**, *38*, 3–20. [\[CrossRef\]](#)
22. Pistikopoulos, E.N.; Dua, V.; Bozinis, N.A.; Bemporad, A.; Morari, M. On-line optimization via off-line parametric optimization tools. *Comput. Chem. Eng.* **2002**, *26*, 175–185. [\[CrossRef\]](#)
23. Wu, X.; Shen, J.; Li, Y.; Lee, K.Y. Fuzzy modeling and predictive control of superheater steam temperature for power plant. *ISA Trans.* **2015**, *56*, 241–251. [\[CrossRef\]](#)
24. Khooban, M.H.; Vafam, N.; Niknam, T. Optimal partitioning of a boiler-turbine unit for Fuzzy model predictive control. *ISA Trans.* **2016**, *64*, 231–240. [\[CrossRef\]](#)
25. Kong, L.; Yuan, J. Disturbance-observer-based fuzzy model predictive control for nonlinear processes with disturbances and input constraints. *ISA Trans.* **2019**, *90*, 74–88. [\[CrossRef\]](#)
26. Kong, L.; Yuan, J. Generalized Discrete-time Nonlinear Disturbance Observer Based Fuzzy Model Predictive Control for Boiler-Turbine Systems. *ISA Trans.* **2019**, *90*, 89–106. [\[CrossRef\]](#)
27. Shen, D.; Lim, C.-C.; Shi, P. Robust fuzzy model predictive control for energy management systems in fuel cell vehicles. *Control. Eng. Pract.* **2020**, *98*, 104364. [\[CrossRef\]](#)
28. Killian, M.; Kozek, M. T-S fuzzy model predictive speed control of electrical vehicles. *IFAC-Papers On Line* **2017**, *50*, 2011–2016. [\[CrossRef\]](#)
29. Marusak, P.; Tatjewski, P. Stability analysis of nonlinear control systems with unconstrained fuzzy predictive controllers. *Arch. Control. Sci.* **2002**, *12*, 267–288.
30. Ławryńczuk, M. Nonlinear state-space predictive control with on-line linearisation and state estimation. *Int. J. Appl. Math. Comput. Sci.* **2015**, *25*, 833–847. [\[CrossRef\]](#)
31. Boulkaibet, I.; Belarbi, K.; Bououden, S.; Marwala, T.; Chadli, M. A new T-S fuzzy model predictive control for nonlinear processes. *Expert Syst. Appl.* **2017**, *88*, 132–151. [\[CrossRef\]](#)
32. Essien, E.; Ibrahim, H.; Mehrandezh, M.; Idem, R. 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. *Therm. Sci. Eng. Prog.* **2019**, *9*, 148–161. [\[CrossRef\]](#)
33. Ławryńczuk, M. *Computationally Efficient Model Predictive Control Algorithms: A Neural Network Approach*; Springer: Heidelberg, Germany, 2014.
34. Marusak, P. Advantages of an easy to design fuzzy predictive algorithm in control systems of nonlinear chemical reactors. *Appl. Soft Comput.* **2009**, *9*, 1111–1125. [\[CrossRef\]](#)
35. Morari, M.; Lee, J.H. Model predictive control: Past, present and future. *Comput. Chem. Eng.* **1999**, *23*, 667–682. [\[CrossRef\]](#)
36. Marusak, P. Efficient model predictive control algorithm with fuzzy approximations of nonlinear models. *LNCS* **2009**, *5495*, 448–457.
37. Lu, J.; Cao, Z.; Zhao, C.; Gao, F. Multipoint Iterative Learning Model Predictive Control. *IEEE Trans. Ind. Electron.* **2019**, *66*, 6230–6240. [\[CrossRef\]](#)

38. Lu, J.; Cao, Z.; Zhang, R.; Gao, F. Nonlinear Monotonically Convergent Iterative Learning Control for Batch Processes. *IEEE Trans. Ind. Electron.* **2018**, *65*, 5826–5836. [\[CrossRef\]](#)
39. Lu, J.; Cao, Z.; Gao, F. 110th Anniversary: An Overview on Learning-Based Model Predictive Control for Batch Processes. *Ind. Eng. Chem. Res.* **2019**, *58*, 17164–17173. [\[CrossRef\]](#)
40. Fink, A.; Fischer, M.; Nelles, O.; Isermann, R. Supervision of nonlinear adaptive controllers based on fuzzy models. *Control. Eng. Pract.* **2000**, *8*, 1093–1105. [\[CrossRef\]](#)
41. Foss, B.A.; Johansen, T.A.; Sorensen, A.V. Nonlinear predictive control using local models—applied to a batch fermentation process. *Control. Eng. Pract.* **1995**, *3*, 389–396. [\[CrossRef\]](#)
42. Babuska, R.; Te, Braake, H.A.B.; Van Can, H.J.L.; Krijgsman, A.J.; Verbruggen, H.B. Comparison of intelligent control schemes for real-time pressure control. *Control. Eng. Pract.* **1996**, *4*, 1585–1592. [\[CrossRef\]](#)
43. Takagi, T.; Sugeno, M. Fuzzy identification of systems and its application to modeling and control. *IEEE Trans. Syst. Man Cybern.* **1985**, *15*, 116–132. [\[CrossRef\]](#)
44. Piegat, A. *Fuzzy Modeling and Control*; Physica-Verlag: Heidelberg, Germany, 2001.
45. Marusak, P. Disturbance Measurement Utilization in the Efficient MPC Algorithm with Fuzzy Approximations of Nonlinear Models. *LNCS* **2013**, *7824*, 307–316.
46. Ribeiro, L.M.; Secchi, A.R. A methodology to obtain analytical models that reduce the computational complexity faced in real time implementation of NMPC controllers. *Braz. J. Chem. Eng.* **2019**, *36*, 1255–1277. [\[CrossRef\]](#)
47. Nguyen, T.S.; Hoang, N.H.; Hussain, M.A. Tracking error plus damping injection control of non-minimum phase processes. *IFAC-Pap. OnLine* **2018**, *51*, 643–648. [\[CrossRef\]](#)
48. Uçak, K. A Runge–Kutta neural network-based control method for nonlinear MIMO systems. *Soft Comput.* **2019**, *23*, 7769–7803. [\[CrossRef\]](#)
49. Jain, A.; Taparia, R. Laguerre function based model predictive control for van–de–vusse reactor. In Proceedings of the 2018 2nd IEEE International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES), Delhi, India, 22–24 October 2018; Volume 3, pp. 1010–1015.
50. Uçak, K. A Novel Model Predictive Runge–Kutta Neural Network Controller for Nonlinear MIMO Systems. *Neural Process. Lett.* **2020**, *51*, 1789–1833. [\[CrossRef\]](#)
51. Mate, S.; Kodamana, H.; Bhartiya, S.; Nataraj, P.S.V. A Stabilizing Sub–Optimal Model Predictive Control for Quasi–Linear Parameter Varying Systems. *IEEE Control. Syst. Lett.* **2020**, *4*, 402–407. [\[CrossRef\]](#)
52. Doyle, F.; Ogunnaike, B.A.; Pearson, R.K. Nonlinear model–based control using second–order Volterra models. *Automatica* **1995**, *31*, 697–714. [\[CrossRef\]](#)



© 2020 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).