

Article

Advanced Construction of the Dynamic Matrix in Numerically Efficient Fuzzy MPC Algorithms

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

Abstract: A method for the advanced construction of the dynamic matrix for Model Predictive Control (MPC) algorithms with linearization is proposed in the paper. It extends numerically efficient fuzzy algorithms utilizing skillful linearization. The algorithms combine the control performance offered by the MPC algorithms with nonlinear optimization (NMPC algorithms) with the numerical efficiency of the MPC algorithms based on linear models in which the optimization problem is a standard, easy-to-solve, quadratic programming problem with linear constraints. In the researched algorithms, the free response obtained using a nonlinear process model and the future trajectory of the control signals is used to construct an advanced dynamic matrix utilizing the easy-to-obtain fuzzy model. This leads to obtaining very good prediction and control quality very close to those offered by NMPC algorithms. The proposed approach is tested in the control system of a nonlinear chemical control plant—a CSTR reactor with the van de Vusse reaction.

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



Citation: Marusak, P.M. Advanced Construction of the Dynamic Matrix in Numerically Efficient Fuzzy MPC Algorithms. *Algorithms* **2021**, *14*, 25. <http://doi.org/10.3390/a14010025>

Received: 7 December 2020
Accepted: 14 January 2021
Published: 17 January 2021

Publisher’s Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 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 (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The Model Predictive Control (MPC) algorithms use the model of the control plant to predict the behavior of the process. Thanks to such an approach, the MPC algorithms can be successfully applied in control systems of processes with delays, with the inverse response, with constraints, and for MIMO (Multiple-Input Multiple-Output) processes; see, e.g., [1–11]. This is because the future control signal trajectories are generated by the MPC algorithms in such a way that the predicted, many sampling instants ahead, future behavior of the control plant and the shape of control signals fulfill the assumed criteria. Thus, the optimization problem, solved in each time step of the MPC algorithm to obtain these control signal trajectories, has the following form [1,4,7,9,12]:

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

subject to:

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

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

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

where:

$$y = \begin{bmatrix} y^1 \\ \vdots \\ y^{n_y} \end{bmatrix}, \quad y^j = \begin{bmatrix} y_{k+1|k}^j \\ \vdots \\ y_{k+p|k}^j \end{bmatrix}, \quad (5)$$

$$\bar{\mathbf{y}} = \begin{bmatrix} \bar{\mathbf{y}}^1 \\ \vdots \\ \bar{\mathbf{y}}^{n_y} \end{bmatrix}, \bar{\mathbf{y}}^j = \begin{bmatrix} \bar{y}_{k+1|k}^j \\ \vdots \\ \bar{y}_{k+p|k}^j \end{bmatrix}, \tag{6}$$

$$\Delta \mathbf{u} = \begin{bmatrix} \Delta \mathbf{u}^1 \\ \vdots \\ \Delta \mathbf{u}^{n_u} \end{bmatrix}, \Delta \mathbf{u}^m = \begin{bmatrix} \Delta u_{k+1|k}^m \\ \vdots \\ \Delta u_{k+s-1|k}^m \end{bmatrix}, \tag{7}$$

$$\mathbf{u} = \begin{bmatrix} \mathbf{u}^1 \\ \vdots \\ \mathbf{u}^{n_u} \end{bmatrix}, \mathbf{u}^m = \begin{bmatrix} u_{k+1|k}^m \\ \vdots \\ u_{k+s-1|k}^m \end{bmatrix}, \tag{8}$$

$$\boldsymbol{\kappa} = [\boldsymbol{\kappa}_1, \dots, \boldsymbol{\kappa}_{n_y}] \cdot \mathbf{I}, \boldsymbol{\kappa}_i = [\kappa_i, \dots, \kappa_i], \tag{9}$$

$$\boldsymbol{\Lambda} = [\boldsymbol{\Lambda}_1, \dots, \boldsymbol{\Lambda}_{n_u}] \cdot \mathbf{I}, \boldsymbol{\Lambda}_i = [\lambda_i, \dots, \lambda_i], \tag{10}$$

where $y_{k+i|k}^j$ denotes a value, predicted using a process model the MPC algorithm is based on, of the j th output for the $(k+i)$ th sampling instant from the prediction horizon, derived at the k th sampling instant, and $\bar{y}_{k+i|k}^j$ is an element of the reference trajectory for the j th output and for the $(k+i)$ th sampling instant from the prediction horizon; if reference trajectories constant on the prediction horizon are used, then:

$$\bar{y}_{k+i|k}^j = \bar{y}_k^j \tag{11}$$

where \bar{y}_k^j denotes a setpoint value for the j th output; $\Delta u_{k+i|k}^m$ are the decision variables of the optimization problem, being future changes in manipulated variables; $\boldsymbol{\kappa}_i$ contains p elements, and $\kappa_j \geq 0$ denote the weighting coefficients for the predicted control errors of the j th output; $\boldsymbol{\Lambda}_i$ contains s elements, and $\lambda_m \geq 0$ denote the weighting coefficients for the changes of the m th manipulated variable; p is the prediction horizon; s is the control horizon; n_y is the number of output variables; n_u is the number of manipulated variables; $\Delta \mathbf{u}_{\min}, \Delta \mathbf{u}_{\max}$ denote the vectors defining the lower and upper bounds of the changes of the control signals; $\mathbf{u}_{\min}, \mathbf{u}_{\max}$ denote vectors defining the lower and upper bounds of the values of the control signals; and $\mathbf{y}_{\min}, \mathbf{y}_{\max}$ denote the vectors defining the lower and upper bounds of the values of the output variables.

The optimization problem (1)–(4) is formulated and solved in each sampling instant, yielding the optimal vector of future control action. From the vector $\Delta \mathbf{u}$, the $\Delta u_{k|k}^m$ elements are extracted and applied in the control system. Then, the described procedure is repeated in the next time step.

1.1. MPC Algorithms Based on Linear Models

The simplest MPC algorithms are based on linear process models; see, e.g., [1,13]. In such a case, the superposition principle holds; therefore, the vector of predicted output values \mathbf{y} can be decomposed into two parts:

$$\mathbf{y} = \tilde{\mathbf{y}} + \mathbf{A} \cdot \Delta \mathbf{u}, \tag{12}$$

$$\tilde{\mathbf{y}} = \begin{bmatrix} \tilde{\mathbf{y}}^1 \\ \vdots \\ \tilde{\mathbf{y}}^{n_y} \end{bmatrix}, \tilde{\mathbf{y}}^j = \begin{bmatrix} \tilde{y}_{k+1|k}^j \\ \vdots \\ \tilde{y}_{k+p|k}^j \end{bmatrix}, \tag{13}$$

where $\tilde{\mathbf{y}}$ is called the free response of the control plant describing the influence of the past values of control signals on the process and $\mathbf{A} \cdot \Delta \mathbf{u}$ is called the forced response, depending

on the future changes of the control signals $\Delta \mathbf{u}$; the matrix A is called the dynamic matrix and has the following form:

$$A = \begin{bmatrix} A^{11} & A^{12} & \dots & A^{1n_u} \\ A^{21} & A^{22} & \dots & A^{2n_u} \\ \vdots & \vdots & \ddots & \vdots \\ A^{n_y 1} & A^{n_y 2} & \dots & A^{n_y n_u} \end{bmatrix}, \tag{14}$$

$$A^{jm} = \begin{bmatrix} a_1^{j,m} & 0 & \dots & 0 & 0 \\ a_2^{j,m} & a_1^{j,m} & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_p^{j,m} & a_{p-1}^{j,m} & \dots & a_{p-s+2}^{j,m} & a_{p-s+1}^{j,m} \end{bmatrix}, \tag{15}$$

where $a_i^{j,m}$ denote the step response coefficients of the control plant, describing the influence of the m th control on the j th output; for details, see, e.g., [9].

After applying the prediction based on a linear model (12), the performance function (1) is transformed into a function that depends quadratically on the decision variables $\Delta \mathbf{u}$:

$$J_{LMPC} = (\bar{\mathbf{y}} - \tilde{\mathbf{y}} - A \cdot \Delta \mathbf{u})^T \cdot \boldsymbol{\kappa} \cdot (\bar{\mathbf{y}} - \tilde{\mathbf{y}} - A \cdot \Delta \mathbf{u}) + \Delta \mathbf{u}^T \cdot \boldsymbol{\Lambda} \cdot \Delta \mathbf{u}. \tag{16}$$

The prediction (12) depends linearly on the decision variables; therefore, after using it in the constraints on the output values (4), the optimization problem (1)–(4) becomes an easy-to-solve, standard quadratic optimization problem, with linear constraints. Unfortunately, the control performance offered by an LMPC algorithm, applied to a nonlinear process, may be unsatisfactory. In order to improve it, one can use an MPC algorithm based on a nonlinear model.

1.2. MPC Algorithms Based on Nonlinear Models

Assume that we have a nonlinear process model:

$$\hat{\mathbf{y}}_k = f(\mathbf{y}_{k-1}, \mathbf{y}_{k-2}, \dots, \mathbf{y}_{k-n_a}, \mathbf{u}_{k-1}, \mathbf{u}_{k-2}, \dots, \mathbf{u}_{k-n_b}), \tag{17}$$

where $\mathbf{y}_{k-i} = [y_{k-i}^1, \dots, y_{k-i}^{n_y}]^T$ is a vector that contains output values measured at the $(k-i)$ th sampling instant and $\mathbf{u}_{k-i} = [u_{k-i}^1, \dots, u_{k-i}^{n_u}]^T$ is a vector that contains control values applied at the $(k-i)$ th sampling instant; denote the output values generated by the model in the $(k+i)$ th sampling instant as $\hat{\mathbf{y}}_{k+i} = [\hat{y}_{k+i}^1, \dots, \hat{y}_{k+i}^{n_y}]^T$; n_a, n_b determine how many past output and control values the model needs. If one wants to use the model (17) 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:

$$\mathbf{y}_{k+i|k} = f(\hat{\mathbf{y}}_{k+i-1}, \dots, \hat{\mathbf{y}}_{k+1}, \mathbf{y}_k, \dots, \mathbf{y}_{k-n_a+i-1}, \mathbf{u}_{k+i-1|k}, \dots, \mathbf{u}_{k|k}, \mathbf{u}_{k-1}, \dots, \mathbf{u}_{k-n_b+i-1}) + \mathbf{d}_k, \tag{18}$$

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 the decision variables from the vector $\Delta \mathbf{u}$ and:

$$\mathbf{d}_k = \mathbf{y}_k - \hat{\mathbf{y}}_k, \tag{19}$$

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, e.g., [9].

Unfortunately, the optimization problem (1)–(4) with the prediction (18), based on a nonlinear model, is, in general, a non-convex nonlinear optimization problem; see, e.g., [14–16]. Thus, it is a hard-to-solve, time-consuming computational task, and there is no guarantee of finding a global solution, while the time needed to obtain the solution cannot be foreseen in advance. One of the methods to overcome this problem is to use so-called fast NMPC algorithms in which a suboptimal solution is generated faster than in the standard approach; see, e.g., [17–19]. To this group of methods belongs also the explicit approach in which most of the calculations are done off-line; see, e.g., [20–22] (it is interesting that this approach is optimal if the linear model is used [23–25]). Unfortunately, in this approach, the complexity of the controller grows significantly with the number of constraints taken into consideration.

If the process behavior is described by means of a fuzzy model, then the standard NMPC approach can be used, but also the structure of the model can be exploited to formulate algorithms that are easier to solve. One group of such algorithms is based on Linear Matrix Inequalities (LMIs); see, e.g., [26–30]. The other group is based on the classical fuzzy Takagi–Sugeno approach in which a few algorithms in the form of control laws, based on linear models, are used to obtain the fuzzy controller; see, e.g., [31,32].

In the other method, the linearization of the process model for the MPC algorithm is obtained in each time step, and the linear prediction relative to control changes is formulated; see, e.g., [9,33–39]. As a result, the optimization problem solved by the MPC algorithm in each time step is formulated as the quadratic one (like in LMPC algorithms). In the linearization-based algorithms, the prediction can be based on a classical linearization or a method of prediction generation can exploit the structure of the nonlinear model on which it is based. In the algorithms using the fuzzy Takagi–Sugeno model, described in [36,39], both the free response and the dynamic matrix are obtained using the model obtained after the linearization. In the algorithms detailed in [33,37], the (classical) free response is calculated using the nonlinear model. In [12], the advanced free response, calculated using the nonlinear model (which can have any form of the model generating outputs on the basis of input signals), takes into consideration the previously calculated trajectory of the future control signals (it can be improved iteratively if needed; the approach is similar, though slightly different in details, to the iterative prediction improvement in the iterative learning-based approaches to batch control described in [40–42]); the dynamic matrix is generated using the easy-to-obtain fuzzy model.

The approach proposed in the article is an extension of the algorithms presented in [12]. It is done in such a way that the optimization problem solved by the MPC algorithm in each time step is the quadratic one. The modification is introduced in the method of dynamic matrix construction. In [12], elements in the dynamic matrix were obtained taking into consideration the current operation point. Changes of the operating point on the prediction horizon are taken into consideration in the proposed approach. This is done using the free response, generated using the nonlinear control plant model. The obtained prediction is, however, still linear with respect to control changes. Thus, the algorithms utilizing the proposed approach combine the computational simplicity of the LMPC algorithms with the control performance offered by the NMPC algorithms.

The next section details the formulation of the Fuzzy MPC (FMPC) algorithms based on fuzzy and nonlinear models, with the advanced construction of the dynamic matrix. In Section 3, the operation of the FMPC algorithms exploiting the proposed advanced dynamic matrix is tested in a simulation example of the control system of a nonlinear chemical reactor with the inverse response. Conclusions are presented in the last section.

2. Efficient Fuzzy MPC Algorithms with the Advanced Construction of the Dynamic Matrix

The complications resulting from the need to solve a nonlinear optimization task by the MPC algorithm in each time step can be avoided by using an approximation of the

process model carried out in each time step. Then, a nonlinear model is used to obtain the free response in such a way that the prediction is linear relative to decision variables, but mimics the nonlinearity of the process very well. An easy-to-obtain fuzzy model is used to get the dynamic matrix.

The dynamic matrix can be constructed in different ways. In Section 2.2, it is described how to do it in such a way that it fits the nonlinearity of the process better than in the standard approach. The method is based on a skillful utilization of the free response (generated using the nonlinear process model). Then, using both elements needed to obtain the prediction, the free response and the dynamic matrix, the optimization problem (1)–(4), solved in each time step, is formulated as a quadratic optimization problem. Such a problem is easy to solve using commonly available optimization routines. Moreover, the simplified versions of the algorithms, in which a control law is obtained, are also discussed.

The proposed algorithms are the generalization of their counterparts proposed in [12]. They use the same free response generation (reminded in Section 2.1) and the same formulation of the optimization problems solved at each time step (reminded in Section 2.3). The difference is in the way the dynamic matrix is constructed (the topic detailed in Section 2.2.3).

2.1. Generation of the Free Response

The method of advanced free response generation, based on a nonlinear process model, proposed in [12], is reminded in this subsection. Define the following vectors:

$$\mathbf{u}_{k+i|k-1} = \left[u_{k+i|k-1}^1, \dots, u_{k+i|k-1}^{n_u} \right]^T, \tag{20}$$

where $u_{k+i|k-1}^m$ are elements of control signal trajectories obtained in the last $(k - 1)$ st time step.

First, the components of the free response are derived iteratively using the nonlinear model (17) and the vectors (20):

$$\begin{aligned} \hat{\mathbf{y}}_{k+1} &= f(\mathbf{y}_k, \mathbf{y}_{k-1}, \dots, \mathbf{y}_{k-n_a+1}, \mathbf{u}_{k|k-1}, \mathbf{u}_{k-1}, \dots, \mathbf{u}_{k-n_b+1}); \\ \hat{\mathbf{y}}_{k+2} &= f(\hat{\mathbf{y}}_{k+1}, \mathbf{y}_k, \dots, \mathbf{y}_{k-n_a+2}, \mathbf{u}_{k+1|k-1}, \mathbf{u}_{k|k-1}, \mathbf{u}_{k-1}, \dots, \mathbf{u}_{k-n_b+2}); \\ &\vdots \\ \hat{\mathbf{y}}_{k+i} &= f(\hat{\mathbf{y}}_{k+i-1}, \hat{\mathbf{y}}_{k+i-2}, \dots, \mathbf{y}_{k-n_a+i}, \mathbf{u}_{k+i-1|k-1}, \mathbf{u}_{k+i-2|k-1}, \dots, \mathbf{u}_{k-n_b+i}); \\ &\vdots \end{aligned} \tag{21}$$

Note that the values $\hat{\mathbf{y}}_{k+1}$ are used when calculating the output values $\hat{\mathbf{y}}_{k+2}$, and in general, in the i th iteration, the values $\hat{\mathbf{y}}_{k+1}, \dots, \hat{\mathbf{y}}_{k+i-1}$ are used to obtain $\hat{\mathbf{y}}_{k+i}$.

Next, the final form of the free response is obtained, after taking into account the estimation of unmeasured disturbances $\mathbf{d}_k = \mathbf{y}_k - \hat{\mathbf{y}}_{k|k-1}$:

$$\tilde{\mathbf{y}}_{k+i|k} = \hat{\mathbf{y}}_{k+i} + \mathbf{d}_k. \tag{22}$$

Note that the free response can be iteratively improved using the nonlinear model. Moreover, it can be modified to include information about measured disturbances. These topics are detailed in [12,43].

2.2. Generation of the Dynamic Matrix

2.2.1. Fuzzy Model Used to Obtain the Dynamic Matrix

The fuzzy Takagi–Sugeno model used to generate the dynamic matrix has the following form:

Rule f : (23)

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

where $y_{k-i}^{j_y}$ is the value of the j_y th output variable at the $(k-i)$ th sampling instant, $u_{k-i}^{j_u}$ is the value of j_u th manipulated variable at the $(k-i)$ 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, $j_y = 1, \dots, n_y$, $j_u = 1, \dots, n_u$, $f = 1, \dots, l$, and l is number of fuzzy rules. The model is composed of local models in the form of step responses [39]; thus, it can be obtained relatively easily, because it is sufficient to collect a few sets of step responses, near a few operating points, e.g., using a nonlinear model of the process. Next, the premises of the model should be formulated using expert knowledge and then tuned, e.g., by means of a fuzzy neural network [9].

2.2.2. Standard Dynamic Matrix

In order to calculate the output values of the model (23), the normalized firing strengths of the fuzzy rules should be calculated using fuzzy reasoning; see, e.g., [44,45]. These values are calculated using the previous values of the output and control signals. In the k th sampling instant, the normalized firing strengths \tilde{w}_k^f are obtained, then the output values of the model are as follows:

$$\hat{y}_k^j = \sum_{m=1}^{n_u} \sum_{n=1}^{p_d-1} \tilde{a}_n^{j,m} \cdot \Delta u_{k-n}^m + \tilde{a}_{p_d}^{j,m} \cdot u_{k-p_d}^m, \tag{24}$$

where:

$$\tilde{a}_n^{j,m} = \sum_{f=1}^l \tilde{w}_k^f \cdot a_n^{j,m,f}. \tag{25}$$

In the standard approach to the dynamic matrix generation, exploited, e.g., in [12,46,47], the parameters $\tilde{a}_n^{j,m}$ are used in the construction of the dynamic matrix in the same way as in the LMPC algorithm, i.e., they are used at each sampling instant from the prediction horizon despite being calculated using the firing strengths \tilde{w}_k^f obtained in the current (k)th sampling instant. Thus, they are obtained for the current operating point, which in general, changes on the prediction horizon. Therefore, now the improved version of the dynamic matrix, adopting the nonlinearity of the process on the prediction horizon, will be proposed. It can be relatively easily obtained using the elements of the free response and of the trajectory of future control signals. The method is detailed below.

2.2.3. Advanced Dynamic Matrix Generation

In the first time step from the prediction horizon, the fuzzy model (generating outputs for the $(k+1)$ st time step) has the following form:

Rule f : (26)

$$\begin{aligned} &\text{if } y_k^{jy} \text{ is } B_1^{f,jy} \text{ and } y_{k-1}^{jy} \text{ is } B_2^{f,jy} \text{ and } \dots \text{ and } y_{k-n+1}^{jy} \text{ is } B_n^{f,jy} \text{ and} \\ &\quad u_k^{ju} \text{ is } C_1^{f,ju} \text{ and } u_{k-1}^{ju} \text{ is } C_2^{f,ju} \text{ and } \dots \text{ and } u_{k-m+1}^{ju} \text{ is } C_m^{f,ju} \\ &\text{then } \hat{y}_{k+1}^{j,f} = \sum_{m=1}^{n_u} \sum_{n=1}^{p_d-1} a_n^{j,m,f} \cdot \Delta u_{k-n+1}^m + a_{p_d}^{j,m,f} \cdot u_{k-p_d+1}^m, \end{aligned}$$

However, u_k^{ju} are not known yet, but the approach used in the free response generation can be applied here, i.e., the appropriate element of the future trajectory of the control signals can be used, namely: $u_{k|k-1}^{ju}$. Therefore, the premises used to obtain the firing strengths of the fuzzy rules will be as follows:

Rule f : (27)

$$\begin{aligned} &\text{if } y_k^{jy} \text{ is } B_1^{f,jy} \text{ and } y_{k-1}^{jy} \text{ is } B_2^{f,jy} \text{ and } \dots \text{ and } y_{k-n+1}^{jy} \text{ is } B_n^{f,jy} \text{ and} \\ &\quad u_{k|k-1}^{ju} \text{ is } C_1^{f,ju} \text{ and } u_{k-1}^{ju} \text{ is } C_2^{f,ju} \text{ and } \dots \text{ and } u_{k-m+1}^{ju} \text{ is } C_m^{f,ju} \end{aligned}$$

As a result, the normalized firing strengths of the fuzzy rules $\tilde{w}_{k+1|k}^f$ will be obtained for the $(k + 1)$ st time step from the prediction horizon (calculated at the k th sampling instant, using the fuzzy reasoning); the appropriate elements of the dynamic matrix will be then calculated using the formula:

$$\tilde{a}_{n,k+1}^{j,m} = \sum_{f=1}^l \tilde{w}_{k+1|k}^f \cdot a_n^{j,m,f}. \tag{28}$$

In the second time step from the prediction horizon, the fuzzy model has the following form:

Rule f : (29)

$$\begin{aligned} &\text{if } y_{k+1}^{jy} \text{ is } B_1^{f,jy} \text{ and } y_k^{jy} \text{ is } B_2^{f,jy} \text{ and } \dots \text{ and } y_{k-n+2}^{jy} \text{ is } B_n^{f,jy} \text{ and} \\ &\quad u_{k+1}^{ju} \text{ is } C_1^{f,ju} \text{ and } u_k^{ju} \text{ is } C_2^{f,ju} \text{ and } \dots \text{ and } u_{k-m+2}^{ju} \text{ is } C_m^{f,ju} \\ &\text{then } \hat{y}_{k+2}^{j,f} = \sum_{m=1}^{n_u} \sum_{n=1}^{p_d-1} a_n^{j,m,f} \cdot \Delta u_{k-n+2}^m + a_{p_d}^{j,m,f} \cdot u_{k-p_d+2}^m \end{aligned}$$

This time, however, not only u_k^{ju} are not known yet, but also u_{k+1}^{ju} and y_{k+1}^{jy} . One can use the approach from the previous step, i.e., instead of u_{k+1}^{ju} , $u_{k+1|k-1}^{ju}$ is used. The other problem is that also the values y_{k+1}^{jy} are not known yet. However, one can use the appropriate elements of the free response here, namely $\hat{y}_{k+1|k}^{jy}$. Therefore, the premises used to obtain the firing strengths in the next $(k + 2)$ nd time step from the prediction horizon will have the following form:

Rule f : (30)

$$\begin{aligned} &\text{if } \hat{y}_{k+1|k}^{jy} \text{ is } B_1^{f,jy} \text{ and } y_k^{jy} \text{ is } B_2^{f,jy} \text{ and } \dots \text{ and } y_{k-n+2}^{jy} \text{ is } B_n^{f,jy} \text{ and} \\ &\quad u_{k+1|k-1}^{ju} \text{ is } C_1^{f,ju} \text{ and } u_{k|k-1}^{ju} \text{ is } C_2^{f,ju} \text{ and } \dots \text{ and } u_{k-m+2}^{ju} \text{ is } C_m^{f,ju} \end{aligned}$$

the normalized firing strengths $\tilde{w}_{k+2|k}^f$ obtained for the $(k + 2)$ nd time step from the prediction horizon, using (30), are then utilized to calculate the elements of the dynamic matrix corresponding to the $(k + 2)$ nd time step from the prediction horizon:

$$\tilde{a}_{n,k+2}^{j,m} = \sum_{f=1}^l \tilde{w}_{k+2|k}^f \cdot a_n^{j,m,f} . \tag{31}$$

This procedure is repeated iteratively in the next time steps from the prediction horizon, resulting in obtaining the normalized firing strengths of the fuzzy rules $\tilde{w}_{k+i|k}^f$ in each $(k+i)$ th, time step from the prediction horizon. Next, the appropriate elements of the dynamic matrix, for each $(k+i)$ th time step from the prediction horizon, will be calculated using the formula:

$$\tilde{a}_{n,k+i}^{j,m} = \sum_{f=1}^l \tilde{w}_{k+i|k}^f \cdot a_n^{j,m,f} . \tag{32}$$

Now, the dynamic matrix can be generated (and updated in each sampling instant):

$$\mathbf{A}_k = \begin{bmatrix} \tilde{\mathbf{A}}_k^{11} & \tilde{\mathbf{A}}_k^{12} & \dots & \tilde{\mathbf{A}}_k^{1n_u} \\ \tilde{\mathbf{A}}_k^{21} & \tilde{\mathbf{A}}_k^{22} & \dots & \tilde{\mathbf{A}}_k^{2n_u} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{\mathbf{A}}_k^{n_y 1} & \tilde{\mathbf{A}}_k^{n_y 2} & \dots & \tilde{\mathbf{A}}_k^{n_y n_u} \end{bmatrix} , \tag{33}$$

$$\tilde{\mathbf{A}}_k^{jm} = \begin{bmatrix} \tilde{a}_{1,k+1}^{j,m} & 0 & \dots & 0 & 0 \\ \tilde{a}_{2,k+2}^{j,m} & \tilde{a}_{1,k+2}^{j,m} & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \tilde{a}_{p,k+p}^{j,m} & \tilde{a}_{p-1,k+p}^{j,m} & \dots & \tilde{a}_{p-s+2,k+p}^{j,m} & \tilde{a}_{p-s+1,k+p}^{j,m} \end{bmatrix} . \tag{34}$$

Note that in the i th row of each matrix $\tilde{\mathbf{A}}_k^{jm}$, the elements calculated for the $(k+i)$ th time step from the prediction horizon are used.

Assume that the firing strengths for a given fuzzy rule are grouped in the following vector:

$$\tilde{\mathbf{w}}^f = [\tilde{w}_{k+1|k}^f, \tilde{w}_{k+2|k}^f, \dots, \tilde{w}_{k+p|k}^f] \tag{35}$$

and define:

$$\mathbf{A}^{jmf} = \begin{bmatrix} a_1^{j,m,f} & 0 & \dots & 0 & 0 \\ a_2^{j,m,f} & a_1^{j,m,f} & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_p^{j,m,f} & a_{p-1}^{j,m,f} & \dots & a_{p-s+2}^{j,m,f} & a_{p-s+1}^{j,m,f} \end{bmatrix} . \tag{36}$$

Then, the matrices $\tilde{\mathbf{A}}_k^{jm}$ can be calculated using the relatively simple formula:

$$\tilde{\mathbf{A}}_k^{jm} = \sum_{f=1}^l \tilde{\mathbf{w}}^f \cdot \mathbf{I} \cdot \mathbf{A}^{jmf} , \tag{37}$$

where \mathbf{I} is the identity matrix of dimension $p \times p$, and the matrices \mathbf{A}^{jmf} remain the same in each time step.

2.3. Optimization Problem in the Numerical and Analytical Versions of the Algorithms

Now, assume that future control values are decomposed as follows:

$$u_{k+i|k}^m = \check{u}_{k+i|k}^m + u_{k+i|k-1}^m , \tag{38}$$

where $\check{u}_{k+i|k}^m$ can be interpreted as the corrections of the control signal $u_{k+i|k-1}^m$ obtained in the last $(k - 1)$ st time step. Thus, future control increments are described by the following, similar formula:

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

Note that the values $u_{k+i|k-1}^m$ and $\Delta u_{k+i|k-1}^m$ are known, as they were calculated in the last time step, and their influence on the output variables is contained in the free response (22), described in Section 2.1. Therefore, now the dynamic matrix (33) will be used to predict the influence of control corrections on the control plant outputs, and after using the free response (22) and the dynamic matrix (33), one obtains the following prediction:

$$\mathbf{y} = \tilde{\mathbf{y}} + \mathbf{A}_k \cdot \Delta \check{\mathbf{u}}. \tag{40}$$

where:

$$\Delta \check{\mathbf{u}} = \begin{bmatrix} \Delta \check{u}^1 \\ \vdots \\ \Delta \check{u}^{n_u} \end{bmatrix}, \Delta \check{u}^m = \begin{bmatrix} \Delta \check{u}_{k+1|k}^m \\ \vdots \\ \Delta \check{u}_{k+s-1|k}^m \end{bmatrix}. \tag{41}$$

Note that the prediction depends linearly on corrections $\Delta \check{\mathbf{u}}$.

2.3.1. Optimization with the Classical Performance Index

Optimization task (1)–(4), which is solved by the control algorithm in each time step, changes now to the following form with corrections $\Delta \check{\mathbf{u}}$ being the decision variables:

$$\arg \min_{\Delta \check{\mathbf{u}}} \left\{ J_{MPC+} = (\bar{\mathbf{y}} - \tilde{\mathbf{y}} - \mathbf{A}_k \cdot \Delta \check{\mathbf{u}})^T \cdot \boldsymbol{\kappa} \cdot (\bar{\mathbf{y}} - \tilde{\mathbf{y}} - \mathbf{A}_k \cdot \Delta \check{\mathbf{u}}) + \Delta \mathbf{u}^T \cdot \boldsymbol{\Lambda} \cdot \Delta \mathbf{u} \right\} \tag{42}$$

subject to:

$$\Delta \mathbf{u}_{\min} \leq \Delta \mathbf{u} \leq \Delta \mathbf{u}_{\max}, \tag{43}$$

$$\mathbf{u}_{\min} \leq \mathbf{u} \leq \mathbf{u}_{\max}, \tag{44}$$

$$\mathbf{y}_{\min} \leq \tilde{\mathbf{y}} + \mathbf{A}_k \cdot \Delta \check{\mathbf{u}} \leq \mathbf{y}_{\max}, \tag{45}$$

where $\Delta \mathbf{u} = \overset{\rightarrow}{\Delta \mathbf{u}} + \Delta \check{\mathbf{u}}$, $\mathbf{u} = \overset{\rightarrow}{\mathbf{u}} + \check{\mathbf{u}}$, and:

$$\overset{\rightarrow}{\Delta \mathbf{u}} = \begin{bmatrix} \overset{\rightarrow}{\Delta \mathbf{u}}^1 \\ \vdots \\ \overset{\rightarrow}{\Delta \mathbf{u}}^{n_u} \end{bmatrix}, \overset{\rightarrow}{\Delta \mathbf{u}}^m = \begin{bmatrix} \Delta u_{k+1|k-1}^m \\ \vdots \\ \Delta u_{k+s-1|k-1}^m \end{bmatrix}, \overset{\rightarrow}{\mathbf{u}} = \begin{bmatrix} \overset{\rightarrow}{\mathbf{u}}^1 \\ \vdots \\ \overset{\rightarrow}{\mathbf{u}}^{n_u} \end{bmatrix}, \overset{\rightarrow}{\mathbf{u}}^m = \begin{bmatrix} u_{k+1|k-1}^m \\ \vdots \\ u_{k+s-1|k-1}^m \end{bmatrix}$$

contain elements of the future control increments' trajectory and of the future control trajectory.

Note that the performance function in (42) depends quadratically on decision variables $\Delta \check{\mathbf{u}}$, and all constraints depend linearly on decision variables. Thus, a standard, easy-to-solve linear-quadratic optimization problem is obtained.

If in each time step, the optimization problem with performance function from (42) is solved without constraints, then it has the following solution given by the analytical formula:

$$\Delta \check{\mathbf{u}} = \left(\mathbf{A}_k^T \cdot \boldsymbol{\kappa} \cdot \mathbf{A}_k + \boldsymbol{\Lambda} \right)^{-1} \cdot \left(\mathbf{A}_k^T \cdot \boldsymbol{\kappa} \cdot (\bar{\mathbf{y}} - \tilde{\mathbf{y}}) + \overset{\rightarrow}{\Delta \mathbf{u}} \right). \tag{46}$$

2.3.2. Optimization with the Modified Performance Index

In the optimization task (42)–(45), a slightly modified performance index can be used:

$$J_{MPC+v2} = (\bar{y} - \tilde{y} - A_k \cdot \Delta \tilde{u})^T \cdot \kappa \cdot (\bar{y} - \tilde{y} - A_k \cdot \Delta \tilde{u}) + \Delta \tilde{u}^T \cdot \Lambda \cdot \Delta \tilde{u}. \quad (47)$$

The change consists of the modification of the second component of the performance index, which now depends only on the corrections of the control signals $\Delta \tilde{u}$. This modification leads to algorithms that generate faster responses, but the influence of the λ parameter on system robustness becomes significantly limited.

If the performance index (47) is minimized without constraints, then the following analytical solution is obtained:

$$\Delta \tilde{u} = \left(A_k^T \cdot \kappa \cdot A_k + \Lambda \right)^{-1} \cdot A_k^T \cdot \kappa \cdot (\bar{y} - \tilde{y}). \quad (48)$$

2.3.3. Utilization of Analytical Versions of the Algorithms

Due to the fact that the dynamic matrix A_k changes, in general, in every time step, then in each time step, the formula (46) or (48) allowing calculating the control action should be used by the algorithm. 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 by the algorithm in each time step.

3. Example

3.1. Control Plant

The control plant is a nonlinear, isothermal CSTR in which the van de Vusse reaction takes place. These CSTRs are a popular benchmark due to their nonlinearity and difficult dynamics, willingly used to test newly developed control algorithms; see, e.g., [48–53]. The process model of the reactor is composed of two composition balance equations; see, e.g., [54]:

$$\begin{aligned} \frac{dC_A}{dt} &= -k_1 \cdot C_A - k_3 \cdot C_A^2 + \frac{F}{V} (C_{Af} - C_A), \\ \frac{dC_B}{dt} &= k_1 \cdot C_A - k_2 \cdot C_B - \frac{F}{V} C_B, \end{aligned} \quad (49)$$

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), and 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 the parameters are: $k_1 = 50$ 1/h, $k_2 = 100$ 1/h, $k_3 = 10$ 1/(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 described control plant was used during the tests also in [12]. Its nonlinear steady-state characteristic is reminded in Figure 1. The control plant has the inverse response; thus, it is natural to use an MPC algorithm in this case.

The fuzzy model used to generate the dynamic matrix, in the researched algorithms, is the same as in [12]. It 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 assumed membership functions are reminded in Figure 2.

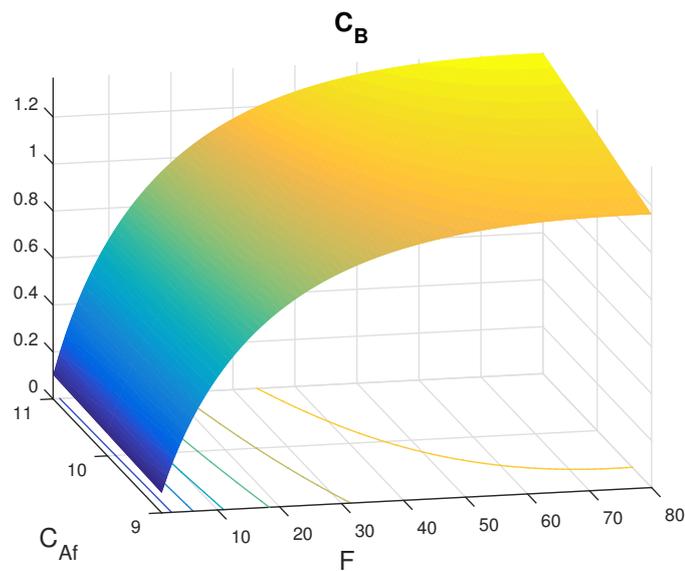


Figure 1. Steady-state characteristic of the control plant.

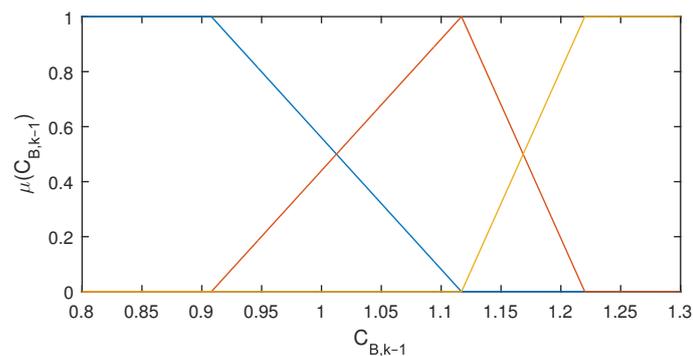


Figure 2. Membership functions of the fuzzy model used to obtain the dynamic matrix.

3.2. Experiments

For the considered control plant, a few MPC algorithms were designed:

- NMPC—based on the nonlinear model and nonlinear optimization and numerically efficient FMPC algorithms with advanced free response generation, proposed in [12], and:
- FMPC1 with the conventional dynamic matrix and the classical performance index,
- FMPC2 with the conventional dynamic matrix and the modified performance index,
- FMPC1a with the advanced dynamic matrix and the classical performance index,
- FMPC2a with the advanced dynamic matrix and the modified performance index.

The simulation experiments were done using MATLAB. During the experiments, the operation of the control systems with the NMPC and FMPC algorithms was compared. The FMPC algorithms use the nonlinear model in the form of state equations, to generate the free response and the fuzzy model (23) to obtain the dynamic matrix. A sampling time equal to $T_s = 3.6$ s was assumed; the values of tuning parameters were as follows (if not declared otherwise): prediction horizon $p = 70$, control horizon $s = 35$, and weighting coefficient $\lambda = 0.001$.

The responses of the control systems to changes in the setpoint and to the change of the disturbance by 10% in the 6th minute of the experiment are shown in Figures 3 and 4. If the setpoint was changed to $\bar{C}_{B1} = 1$ mol/L, the responses obtained in the control system with the FMPC1a algorithm (magenta lines in Figure 3) and with the FMPC1 algorithm (blue lines in Figure 3) were very close to those obtained with the NMPC algorithm with

nonlinear optimization (red lines in Figure 3). In all cases, there was almost no overshoot. The FMPC algorithms were slightly faster than the NMPC algorithm.

In the case when the setpoint changed to $\bar{C}_{B2} = 1.25$ mol/L, the response obtained with the FMPC1 algorithm was the fastest one. The one generated with the NMPC algorithm was the slowest, and the response obtained with the FMPC1a algorithm was between these two responses; it was closer to the response generated with the NMPC algorithm than the response obtained with the FMPC1 algorithm. This is because the prediction used in the FMPC1a algorithm, thanks to using the advanced construction of the dynamic matrix, is more accurate. In the case of all the algorithms, the setpoint was reached without overshoot.

Disturbance responses obtained with the FMPC1 and the FMPC1a algorithms were practically the same. Near $C_B = 1$ mol/L, the response generated with the NMPC algorithm was very close to the other responses. In all three cases, there was no overshoot. In the case of operation near $C_B = 1.25$ mol/L, the NMPC was faster in the compensation of the disturbance change than the FMPC algorithms.

The FMPC2 and FMPC2a algorithms worked faster than FMPC1 and FMPC1a. However, the same relations between the algorithms could be observed when the responses generated with FMPC2 and FMPC2a algorithms were compared with the ones obtained with the NMPC algorithm (Figure 4). If the setpoint was changed to $\bar{C}_{B1} = 1$ mol/L, the responses obtained with all the algorithms (FMPC2, FMPC2a, and NMPC) were very close to each other. There was almost no overshoot, and the FMPC2 algorithm was slightly slower than the other algorithms.

In the case when the setpoint changed to $\bar{C}_{B2} = 1.25$ mol/L, the response obtained with the FMPC2 algorithm was the fastest one. The one generated with the NMPC algorithm was the slowest, and the response obtained with the FMPC2a algorithm, for the same reason as in the case of the FMPC1a algorithm, was between these two responses. In the case of all the algorithms, the setpoint was reached almost without overshoot.

The disturbance responses obtained with the FMPC2 and FMPC2a algorithms were practically the same (the same phenomenon was observed in the case of the FMPC1 and FMPC1a algorithms). Near $C_B = 1$ mol/L and $C_B = 1.25$ mol/L, the NMPC algorithm generated a bigger maximal error than the FMPC algorithms. In all three cases, there was no overshoot.

It can be noticed that the FMPC1a and FMPC2a algorithms generated responses closer to the ones generated by the NMPC algorithm than the FMPC1 and FMPC2 algorithms. This is because in the algorithms with the advanced construction of the dynamic matrix, the prediction was more accurate, thanks to using the proposed mechanism. It should be, however, emphasized that all the FMPC algorithms used a reliable quadratic programming routine to generate the control action instead of the non-convex, nonlinear optimization utilized in the NMPC algorithm. When comparing the SSE (Sum of Squared Errors), one can notice that the smallest value was obtained with the FMPC2 and FMPC2a algorithms. This is because they used the modified performance index in the optimization problem solved by the algorithm in each time step.

We also performed experiments with the changes of the parameters (λ coefficient and the control horizon) of the algorithms, like the ones in [12]. First, the λ coefficient was changed to 0.0001 (Figures 5 and 6). All the algorithms now worked faster than in the previous case (for $\lambda = 0.001$); the control action was more aggressive, and the maximal errors at the beginning of the experiment were now bigger. However, the relations between the FMPC and NMPC algorithms remained unchanged. It can be also noticed that the differences between the FMPC1a and FMPC2a (and also FMPC1 and FMPC2) algorithms became much smaller compared to the previous experiments. The FMPC1a algorithm was practically as fast as the FMPC2a one. This is because for $\lambda = 0$, both performance indexes (42) and (47) were the same. Thus, the closer the value of the λ coefficient to zero is, the closer the responses generated with the FMPC1a and FMPC2a algorithms should be. The

SSE values obtained for all the algorithms were smaller than in the previous case, i.e., for $\lambda = 0.001$.

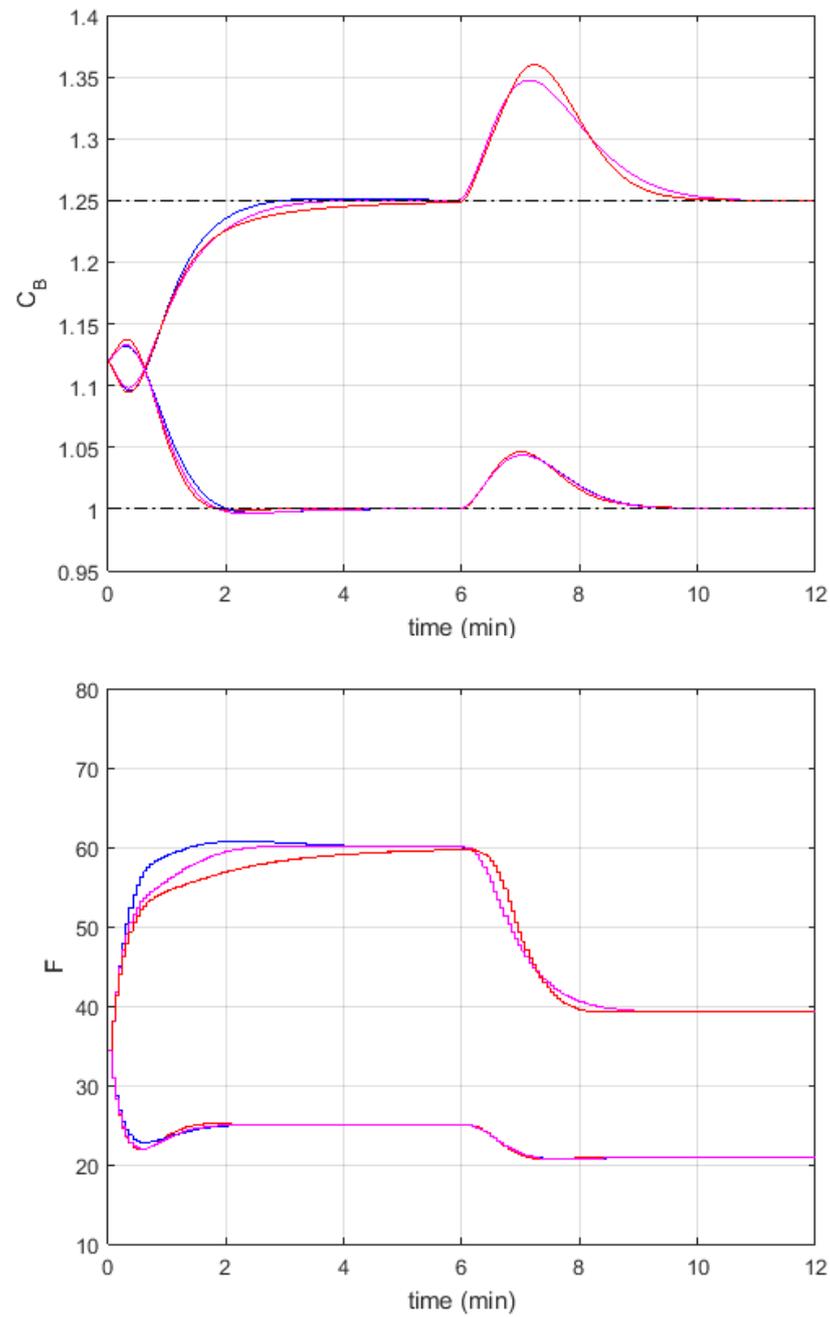


Figure 3. Responses of the control system to changes in the setpoint to $\bar{C}_{B1} = 1$ mol/L and $\bar{C}_{B2} = 1.25$ mol/L and to the change of the disturbance by 10% in the 6th minute of the experiment; $\lambda = 0.001$; NMPC—red lines (SSE = 1.2900), FMPC1—blue lines (SSE = 1.2764), FMPC1a—magenta lines (SSE = 1.2740).

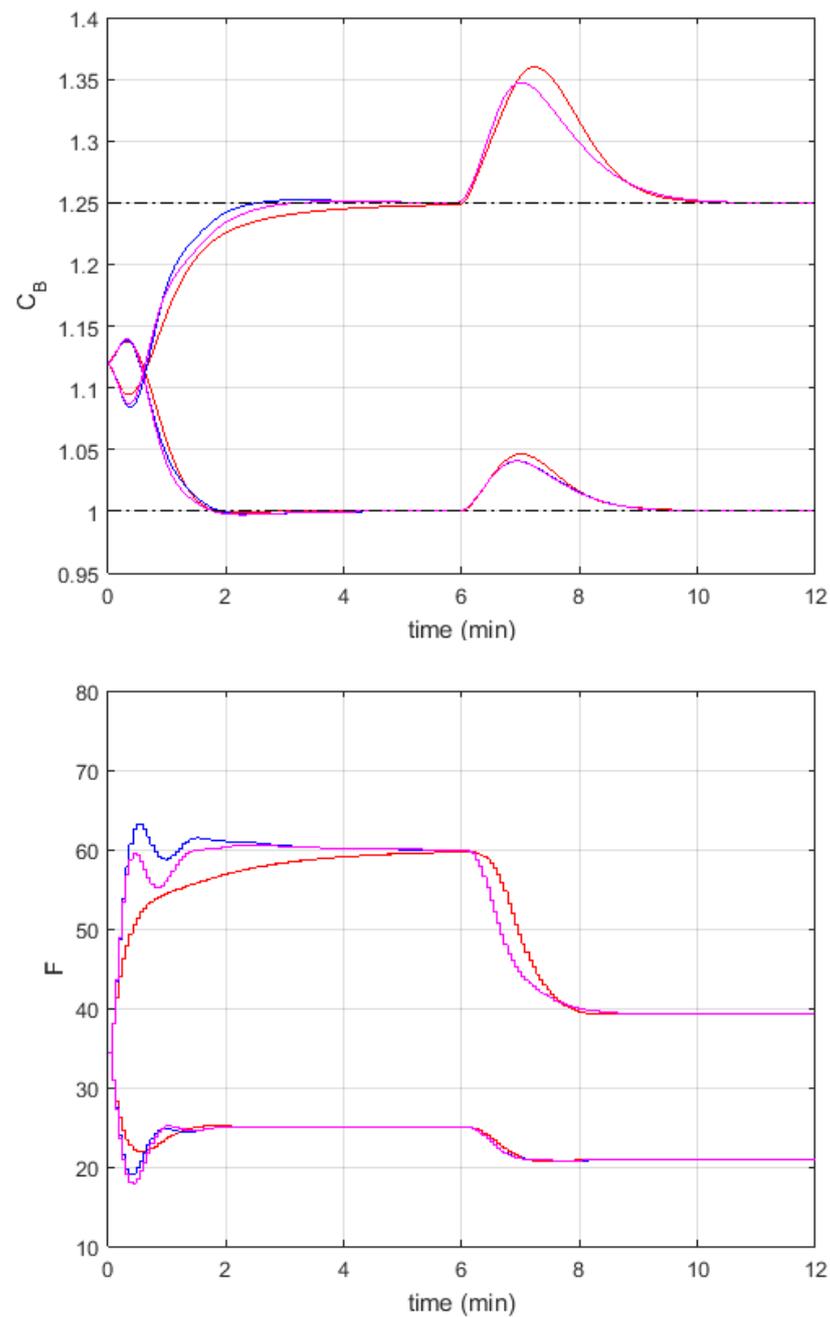


Figure 4. Responses of the control system to changes in the setpoint to $\bar{C}_{B1} = 1$ mol/L and $\bar{C}_{B2} = 1.25$ mol/L and to the change of the disturbance by 10% in the 6th minute of the experiment; $\lambda = 0.001$; NMPC—red lines (SSE = 1.2900), FMPC2—blue lines (SSE = 1.2230), FMPC2a—magenta lines (SSE = 1.2183).

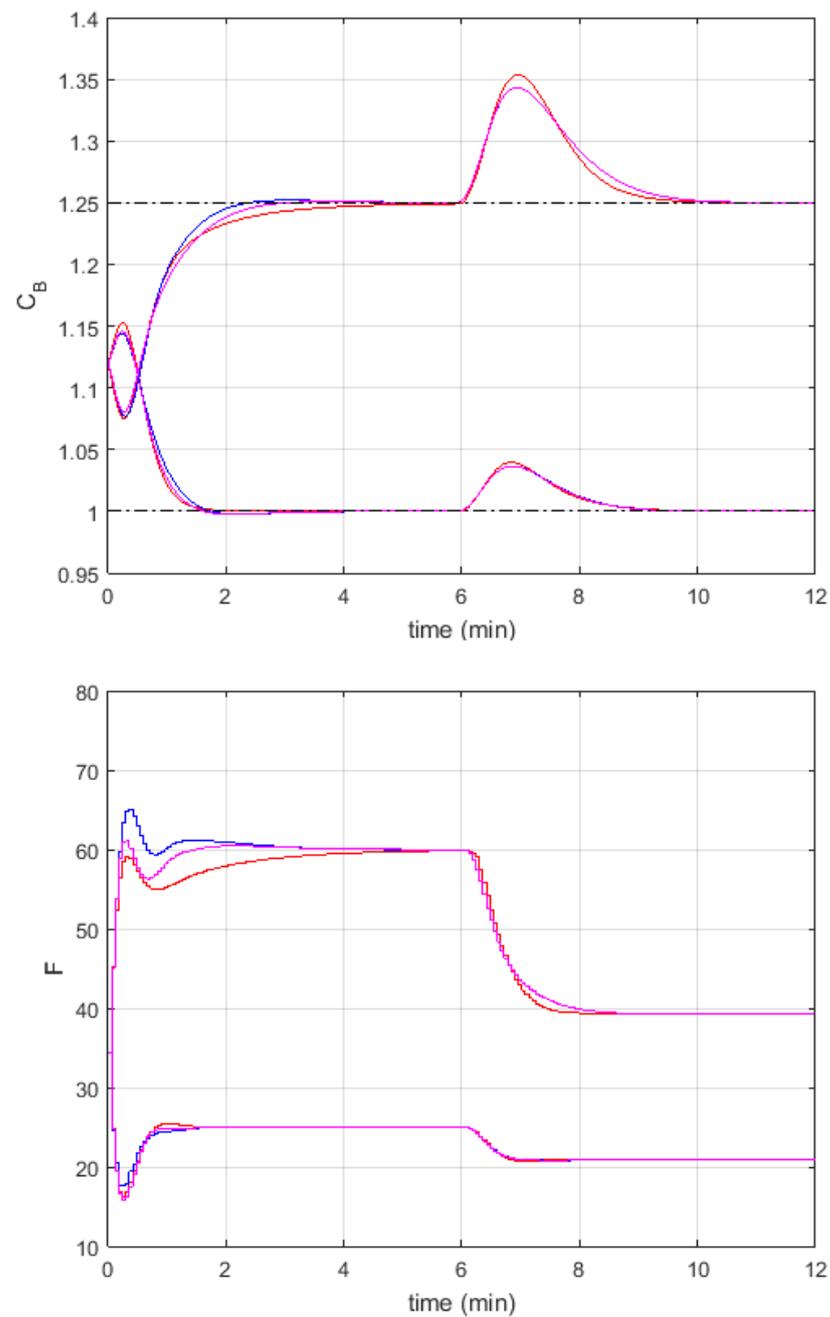


Figure 5. Responses of the control system to changes in the setpoint to $\bar{C}_{B1} = 1$ mol/L and $\bar{C}_{B2} = 1.25$ mol/L and to the change of the disturbance by 10% in the 6th minute of the experiment; $\lambda = 0.0001$; NMPC—red lines (SSE = 1.1913), FMPC1—blue lines (SSE = 1.1679), FMPC1a—magenta lines (SSE = 1.1629).

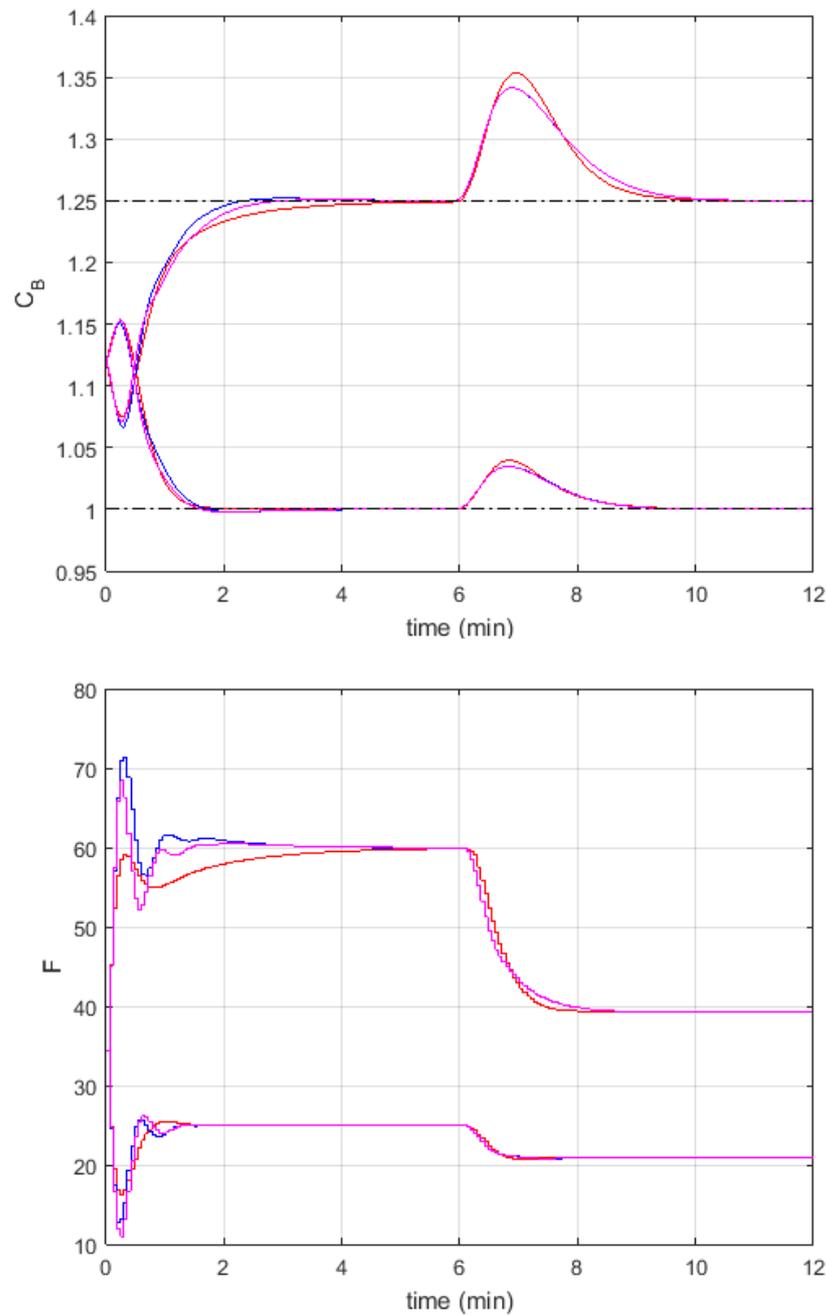


Figure 6. Responses of the control system to changes in the setpoint to $\bar{C}_{B1} = 1$ mol/L and $\bar{C}_{B2} = 1.25$ mol/L and to the change of the disturbance by 10% in the 6th minute of the experiment; $\lambda = 0.0001$; NMPC—red lines (SSE = 1.1913), FMPC2—blue lines (SSE = 1.1502), FMPC2a—magenta lines (SSE = 1.1450).

To simplify the optimization problem by reducing the number of decision variables, one can decrease the control horizon s . The responses obtained for different values of the control horizon are shown in Figures 7 and 8 for the FMPC1a and FMPC2a algorithms, respectively. In the case of both algorithms, the control horizon can be shortened significantly, because the obtained responses do not change too much when the control horizon is between $s = 10$ and $s = 35$. However, reduction of the control horizon should be done carefully, because for a too short control horizon, the control performance can get worse.

The FMPC1a algorithm, for $s = 5$, compensated the disturbance near $C_B = 1.25$ mol/L faster, but achieved the setpoint $\bar{C}_{B1} = 1$ mol/L slightly more slowly. A further decrease

of the control horizon to $s = 1$ brought faster disturbance compensation near $C_B = 1.25$ mol/L, but worsened the compensation near $C_B = 1$ mol/L. Reaching the setpoint $\bar{C}_{B2} = 1.25$ mol/L was also faster, but at the cost of a slower response to the setpoint change to $\bar{C}_{B1} = 1$ mol/L. The value of the SSE decreased with the decrease of the control horizon, but the value of the SSE obtained for $s = 1$ was noticeably smaller than in the other cases.

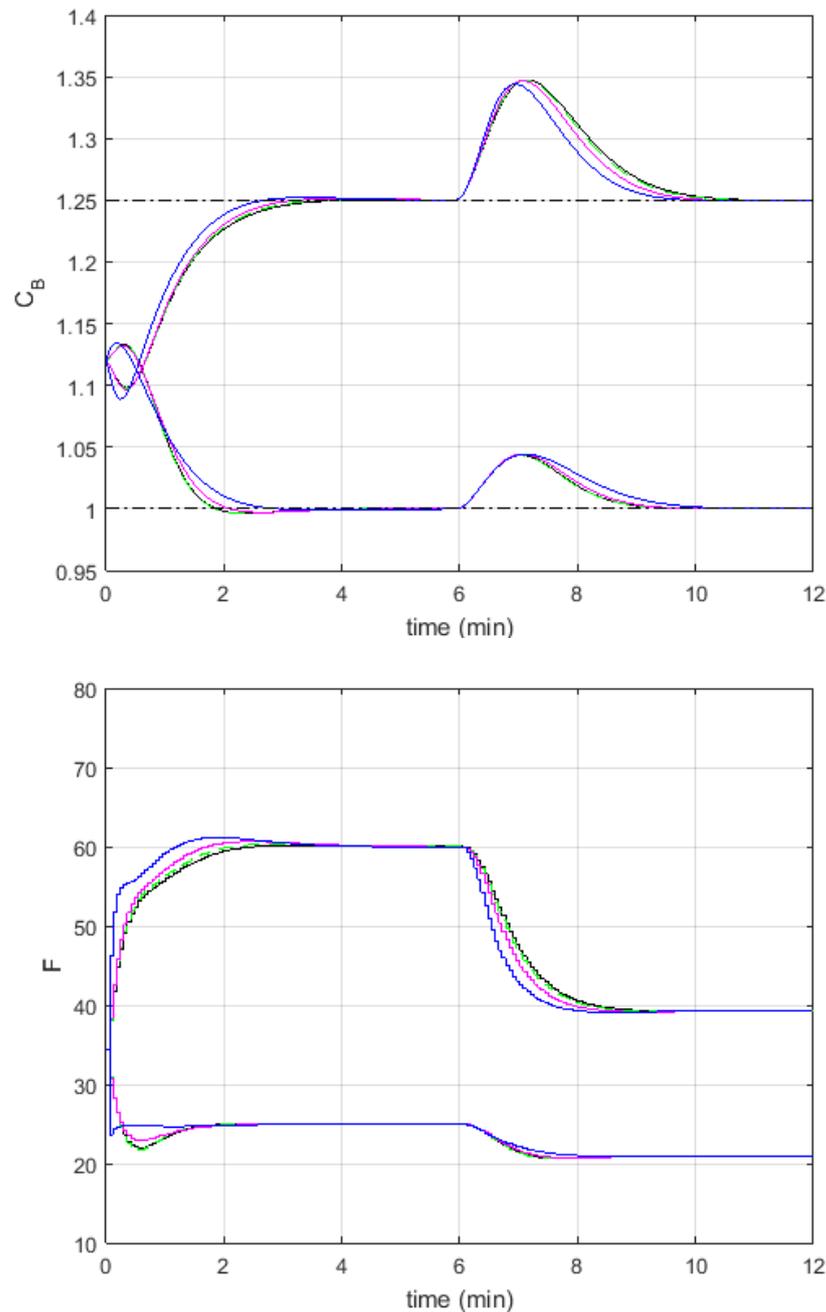


Figure 7. Responses of the control system with FMPC1a algorithm and $\lambda = 0.001$ to changes in the setpoint to $\bar{C}_{B1} = 1$ mol/L and $\bar{C}_{B2} = 1.25$ mol/L and to change of the disturbance by 10% in the 6th minute of the experiment; $s = 35$ —black lines (SSE = 1.2740), $s = 10$ —green dashed lines (SSE = 1.2675), $s = 5$ —magenta lines (SSE = 1.2631), $s = 1$ —blue lines (SSE = 1.2218).

The FMPC2a algorithm, for $s = 5$ and $s = 1$, compensates the disturbance near $C_B = 1.25$ mol/L slightly faster, but unfortunately reaching the setpoint $\bar{C}_{B1} = 1$ mol/L took longer, especially for $s = 1$. Disturbance compensation near $C_B = 1$ mol/L was also

visibly slower for $s = 1$. The smallest value of the SSE was obtained for $s = 10$; however, the differences in the SSE values in all cases were very small.

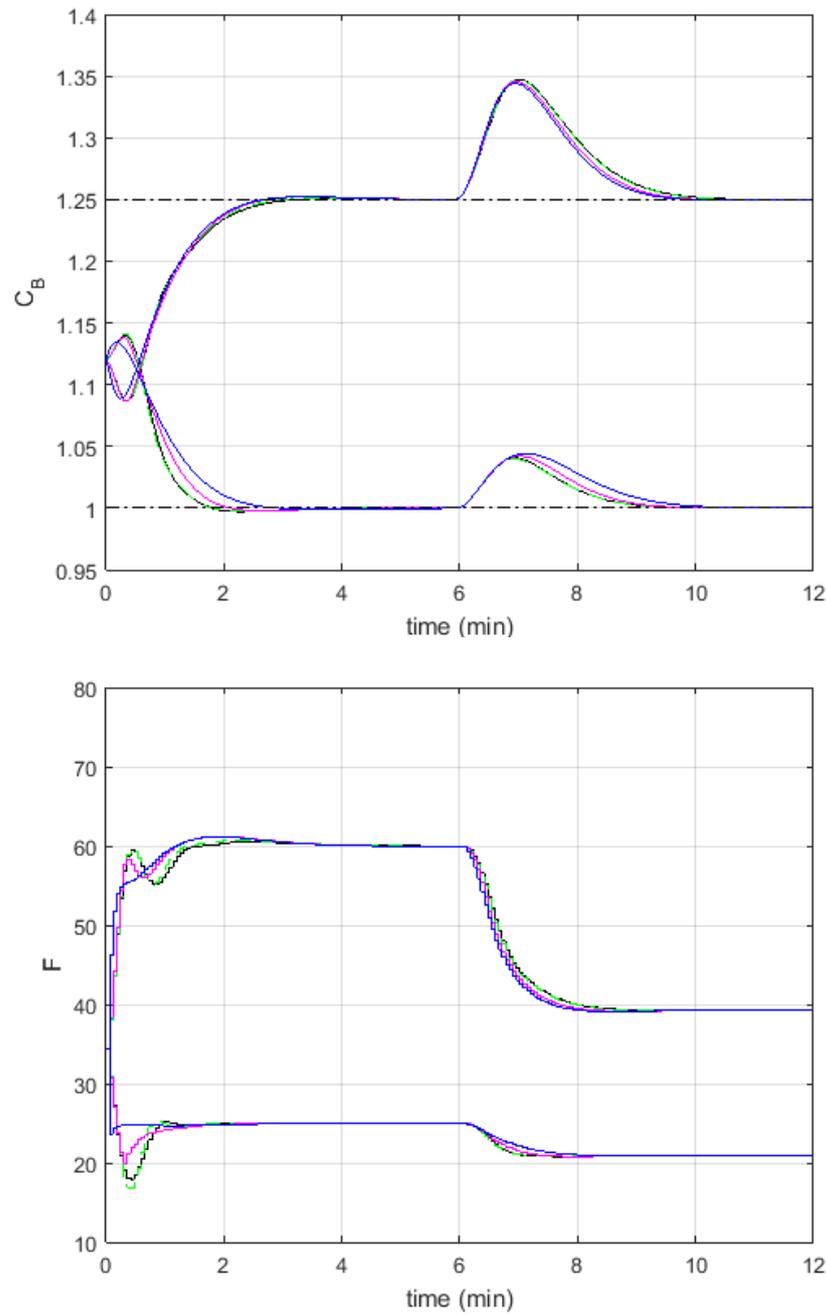


Figure 8. Responses of the control system with FMPC2a algorithm and $\lambda = 0.001$ to changes in the setpoint to $\bar{C}_{B1} = 1$ mol/L and $\bar{C}_{B2} = 1.25$ mol/L and to the change of the disturbance by 10% in the 6th minute of the experiment; $s = 35$ —black lines (SSE = 1.2183), $s = 10$ —green dashed lines (SSE = 1.2142), $s = 5$ —magenta lines (SSE = 1.2220), $s = 1$ —blue lines (SSE = 1.2218).

4. Conclusions

The mechanism for the advanced construction of the dynamic matrix proposed in the article uses the easy-to-obtain fuzzy model and advanced free response to generate the dynamic matrix skillfully. Thanks to such an approach, the changes of the operating point on the prediction horizon are taken into consideration during the construction of the dynamic matrix. This leads to obtaining a very good prediction, which is, however, linear with respect to the control changes. As a result, the FMPC algorithms can offer control

performance very close to the one characteristic for the NMPC algorithms (based on the nonlinear optimization) and, at the same time, numerical efficiency resulting from the fact that the proposed mechanism is designed in such a way that the FMPC algorithms are formulated as the easy-to-solve quadratic optimization problems.

The proposed mechanism can be applied by a control system designer also in the case of simpler FMPC algorithms, e.g., with the classical free response. It should improve the operation of such a controller, but a better prediction can be obtained in the case considered in the article when the advanced free response is used. The quality of the advanced dynamic matrix and of the prediction can be increased even more if iterative improvement of the free response is applied.

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
SSE	Sum of Squared Errors

References

- Camacho, E.F.; Bordons, C. *Model Predictive Control*; Springer: London, UK, 1999.
- Domański, P.D. Performance Assessment of Predictive Control—A Survey. *Algorithms* **2020**, *13*, 97.
- 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.
- Maciejowski, J.M. *Predictive Control with Constraints*; Prentice Hall: Harlow, UK, 2002.
- Nebeluk, R.; Marusak, P. Efficient MPC algorithms with variable trajectories of parameters weighting predicted control errors. *Arch. Control Sci.* **2020**, *30*, 325–363.
- Plamowski, S.; Kephart, R.W. The Model Order Reduction Method as an Effective Way to Implement GPC Controller for Multidimensional Objects. *Algorithms* **2020**, *13*, 178.
- Rossiter, J.A. *Model-Based Predictive Control: A Practical Approach*; CRC Press: Boca Raton, FL, USA, 2003.
- Sands, T. Comparison and Interpretation Methods for Predictive Control of Mechanics. *Algorithms* **2019**, *12*, 232.
- Tatjewski, P. *Advanced Control of Industrial Processes; Structures and Algorithms*; Springer: London, UK, 2007.
- Blevins, T.L.; McMillan, G.K.; Wojsznis, W.K.; Brown, M.W. *Advanced Control Unleashed*; The ISA Society: Research Triangle Park, NC, USA, 2003.
- Ławryńczuk, M.; Marusak, P.; Tatjewski, P. Cooperation of model predictive control with steady-state economic optimisation. *Control Cybern.* **2008**, *37*, 133–158.
- Marusak, P. Numerically Efficient Fuzzy MPC Algorithm with Advanced Generation of Prediction: Application to a Chemical Reactor. *Algorithms* **2020**, *13*, 143.
- 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.
- Abdelaal, M.; Schön, S. Predictive Path Following and Collision Avoidance of Autonomous Connected Vehicles. *Algorithms* **2020**, *13*, 52.
- Chen, H.; Allgöwer, F. A Quasi-Infinite Horizon Nonlinear Model Predictive Control Scheme with Guaranteed Stability. *Automatica* **1998**, *34*, 1205–1217.
- Tatjewski, P. Offset-free nonlinear model predictive control with state-space process models. *Arch. Control Sci.* **2017**, *27*, 595–615.
- 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.
- Schäfer, A.; Kühl, 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.

19. 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.
20. Dominguez, L.F.; Pistikopoulos, E.N. A Novel mp-NLP Algorithm for Explicit/Multi-parametric NMPC. In Proceedings of the 8th IFAC Symposium on Nonlinear Control Systems, Bologna, Italy, 1–3 September 2010.
21. Johansen, T.A. On multi-parametric nonlinear programming and explicit nonlinear model predictive control. In Proceedings of the 41st IEEE Conf Decision and Control, Las Vegas, NV, USA, 10–13 December 2002; Volume 3, pp. 2768–2773.
22. Johansen, T.A. Approximate explicit receding horizon control of constrained nonlinear systems. *Automatica* **2004**, *40*, 293–300.
23. 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.
24. Bemporad, A.; Morari, M.; Dua, V.; Pistikopoulos, E.N. The explicit linear quadratic regulator for constrained systems. *Automatica* **2002**, *38*, 3–20.
25. Bemporad, A.; Borrelli, F.; Morari, M. Piecewise linear optimal controllers for hybrid systems. *2000 Am. Control Conf.* **2000**, *2*, 1190–1194.
26. 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.
27. 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.
28. 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.
29. 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.
30. 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.
31. Marusak, P.; Tatjewski, P. Stability analysis of nonlinear control systems with unconstrained fuzzy predictive controllers. *Arch. Control Sci.* **2002**, *12*, 267–288.
32. Killian, M.; Kozek, M. T–S fuzzy model predictive speed control of electrical vehicles. *IFAC-Pap. Line* **2017**, *50*, 2011–2016.
33. Marusak, P. Efficient model predictive control algorithm with fuzzy approximations of nonlinear models. *LNCS* **2009**, *5495*, 448–457.
34. Ławryńczuk, M. *Computationally Efficient Model Predictive Control Algorithms: A Neural Network Approach*; Springer: Heidelberg, Germany, 2014.
35. Morari, M.; Lee, J.H. Model predictive control: Past, present and future. *Comput. Chem. Eng.* **1999**, *23*, 667–682.
36. 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.
37. Essien, E.; Ibrahim, H.; Mehrendezh, 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.
38. Ławryńczuk, M. Nonlinear state–space predictive control with on–line linearisation and state estimation. *Int. J. Appl. Math. Comput. Sci.* **2015**, *25*, 833–847.
39. 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.
40. 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.
41. 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.
42. Lu, J.; Cao, Z.; Zhao, C.; Gao, F. Multipoint Iterative Learning Model Predictive Control. *IEEE Trans. Ind. Electron.* **2019**, *66*, 6230–6240.
43. Marusak, P. Disturbance Measurement Utilization in the Efficient MPC Algorithm with Fuzzy Approximations of Nonlinear Models. *LNCS* **2013**, *7824*, 307–316.
44. Takagi, T.; Sugeno, M. Fuzzy identification of systems and its application to modeling and control. *IEEE Trans. Syst. Man Cybern.* **1985**, *15*, 116–132.
45. Piegat, A. *Fuzzy Modeling and Control*; Physica–Verlag: Heidelberg, Germany, 2001.
46. Marusak, P. Efficient fuzzy predictive algorithms with integrated economic optimization: A case study. *IFAC Proc. Vol.* **2007**, *40*, 61–66.
47. Marusak, P. Easily reconfigurable analytical fuzzy predictive controllers: Actuator faults handling. *LNCS* **2008**, *5370*, 396–405.
48. 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.
49. 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.

50. Jain, A.; Taparia, R. Laguerre function based model predictive control for van-de-vusse reactor. In Proceedings of the 2nd IEEE Int. Conf. Power Electronics, Intelligent Control and Energy Systems, ICPEICES 2018, Delhi, India, 22–24 October 2018; Volume 3, pp. 1010–1015.
51. Uçak, K. A Runge–Kutta neural network-based control method for nonlinear MIMO systems. *Soft Comput.* **2019**, *23*, 7769–7803.
52. Uçak, K. A Novel Model Predictive Runge–Kutta Neural Network Controller for Nonlinear MIMO Systems. *Neural Process. Lett.* **2020**, *51*, 1789–1833.
53. Sang Nguyen, T.; Hoang, N.H.; Hussain, M.A. Tracking error plus damping injection control of non-minimum phase processes. *IFAC-Pap. Line* **2018**, *51*, 643–648.
54. Doyle, F.; Ogunnaike, B.A.; Pearson, R.K. Nonlinear model-based control using second-order Volterra models. *Automatica* **1995**, *31*, 697–714.