

Article

Analytical MPC Algorithm Using Steady-State Process Model

Piotr M. Marusak 

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

Abstract: For some classes of control plants (e.g., large time delay or inverse response), the PID controllers may offer unsatisfactory results; on the other hand, a Model Predictive Control (MPC) algorithm based on a linear model may offer insufficient control quality when applied to nonlinear control plants. To improve the MPC algorithm operation, one can use a steady-state process model; this paper describes how to do this skillfully. The obtained algorithm, based on the popular Dynamic Matrix Control (DMC) algorithm, is detailed. The proposed approach consists in modifying the analytical version of the DMC algorithm in such a way that it can still be expressed as the control law. Thus, the algorithm can still be applied to fast control plants, requiring short sampling times. Though the proposed approach does not modify the DMC algorithm much, it offers improvement in the control quality when the algorithm is employed in a nonlinear control plant. Experiments illustrating the efficiency of the proposed approach were conducted in the control system of a nonlinear chemical reactor. The results show improvement in the control quality compared to a case when the classical MPC algorithm is used.

Keywords: prediction; process control; model predictive control (MPC); dynamic matrix control (DMC); nonlinear control; quadratic optimization; steady-state characteristic

1. Introduction

PID control is practically standard in the case of control systems with short sampling times. Moreover, PID is still being researched and developed using modern techniques; see, e.g., [1–4]. However, in the case of control plants with difficult dynamics (e.g., long time delays, inverse response), other control techniques, like the Smith predictor (see, e.g., [5,6]) or the Model Predictive Control (MPC) algorithms, are used. The advantages of the MPC algorithms originate in using the model of the process to predict its behavior [7–12]. It is good to note that if an accurate analytical model is not available or is hard to obtain, an approach based on artificial neural networks can be applied; see, e.g., [13–15]. The MPC algorithms can be divided into two main groups. In the numerical MPC algorithms, the control signal is generated after solving an optimization task online. In the second group of analytical MPC algorithms, the control law is obtained after offline optimization. Numerous studies have been concerned with numerical algorithms; the NMPC (nonlinear model-based MPC) algorithms are based on non-convex, hard to solve, and computationally time-consuming, nonlinear optimization problems solved at each iteration; see, e.g., [16–18]. To limit drawbacks of this approach, the so-called fast NMPC algorithms, generating suboptimal solutions faster than in the standard approach, were developed; see, e.g., [19–21]. In this next group of numerical algorithms, the process model is skillfully used to formulate a prediction that is linear with respect to the control action to simplify the algorithm by formulating the optimization task as the quadratic optimization problem; see, e.g., [12,13,22–28].



Academic Editors: Paulo Moura Oliveira and Ramiro Barbosa

Received: 18 December 2024

Revised: 28 January 2025

Accepted: 31 January 2025

Published: 2 February 2025

Citation: Marusak, P.M. Analytical MPC Algorithm Using Steady-State Process Model. *Algorithms* **2025**, *18*, 79. <https://doi.org/10.3390/a18020079>

Copyright: © 2025 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/>).

The explicit algorithms can be considered as being between numerical and analytical. In the explicit approach, most of the calculations are done offline. The algorithms using a nonlinear process model are suboptimal—see, e.g., [29–31]—but the ones based on the linear model generate optimal solutions [32–34]. The main drawback of the explicit approach is that the complexity of the controller grows rapidly with the number of constraints included in the problem.

The analytical group of MPC algorithms has been researched more and more recently as the computation power of modern microcontrollers has increased, allowing the implementation of analytical MPC algorithms. However, mainly basic versions of them based on linear models have been used [35,36] or researched [37–39], including robust MPC algorithms using Linear Matrix Inequalities (LMIs) to calculate the control signal [40–42]. They offer good control performance for processes that are not highly nonlinear but may work poorly for other ones. Therefore, it is desirable to propose analytical MPC algorithms designed for nonlinear plants; an algorithm based on the fuzzy process model is proposed in [43]. The current paper contains a proposal for such an algorithm, which is a generalization of the classical analytical DMC algorithm, utilizing only a process step response and a steady-state process model.

The algorithm proposed in this article is intended to cope mainly with the nonlinearity of the control plant, but at the same time, it is also capable of dealing with delay and other non-minimum phase behaviors of control plants (for which predictive control algorithms are usually applied). Moreover, the algorithm is designed in such a way that it offers good performance with low computational demand and less computing power required by the algorithm, which means cheaper chips can be used in the embedded systems, resulting in fewer problems in the future when chips start to be rationed and access to them becomes part of the economic rivalry between countries.

Moreover, the proposed algorithm uses a relatively small modification of the DMC algorithm, so it can be easily adopted by someone who knows the popular Dynamic Matrix Control algorithm. However, introducing such a small modification requires an appropriate formulation of the control law; in this case, assuming the parameter λ varies in time and depends on the gain of the control plant at the current operating point was crucial and necessary.

2. Materials and Methods

2.1. MPC Algorithms

The MPC algorithms generate the future control signal trajectories in such a way that, predicting many sampling instants ahead, the future behavior of the control plant and the shape of control signals fulfill assumed criteria, which, for SISO (Single-Input Single-Output) control plants, are expressed in the form of the following optimization task [7,8,10,12]:

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

where $y = [y_{k+1|k}, \dots, y_{k+p|k}]$, $\Delta u = [\Delta u_{k|k}, \dots, \Delta u_{k+s-1|k}]$, $y_{k+j|k}$ is the output value for the $(k+i)^{\text{th}}$ sampling instant predicted at the k^{th} sampling instant using a control plant model, $\Delta u_{k+i|k}$ are future changes in the manipulated variable, and $\lambda \geq 0$ is the weighting coefficient for the changes of the manipulated variable; p and s denote prediction and control horizons, respectively. The following vector defines the reference trajectory:

$$\bar{y} = [\bar{y}_{k+1|k}, \dots, \bar{y}_{k+p|k}]. \quad (2)$$

As a solution to the optimization problem given in (1), 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) and the control signal does not change too much (the second component). The $\Delta u_{k|k}$ element of the vector of decision variables Δu is applied to the control system and the algorithm passes to the next iteration. The predicted values of the output signal $y_{k+j|k}$ are derived using the control plant model the MPC algorithm is based on. In order to take into account modeling uncertainty and unmodeled disturbances, the prediction is calculated, including correction, which is the difference between the measured process output and the output of the model used by the algorithm. If a linear model is used, then (1) becomes quadratic with respect to the changes in the control signal. Moreover, a control law can be obtained if the optimization problem is solved without constraints. The DMC algorithm will now be detailed.

2.2. Analytical DMC Algorithm

The simplest MPC algorithms are based on linear process models; see e.g., [7]. One of them is the DMC algorithm, based on the process model in the form of the step response:

$$\hat{y}_k = \sum_{n=1}^{p_d-1} a_n \cdot \Delta u_{k-n} + a_{p_d} \cdot u_{k-p_d}, \quad (3)$$

where a_n are the coefficients of the control plant step response, and p_d is the horizon of dynamics equal to the number of time instants, after which the step response coefficients can be considered fixed.

The predicted output values, calculated from Model (3), are thus given by the following formula:

$$y_{k+i|k} = \hat{y}_{k+i} + d_k, \quad (4)$$

where $d_k = y_k - \hat{y}_k$ is a model of unmeasured disturbances (it also includes the effect of modeling errors) acting on the process; it is calculated as the difference between the current process output measurement y_k and the output of the model described by (3). If no better information is available, it is assumed that it is constant over the entire prediction horizon. This model of unmeasurable disturbances is called a DMC-type disturbance model. After transformation of the prediction (4), one obtains the following:

$$y_{k+i|k} = \sum_{n=1}^i a_n \cdot \Delta u_{k-n+i} + \sum_{n=i+1}^{p_d-1} a_n \cdot \Delta u_{k-n+i} + a_{p_d} \cdot u_{k-p_d} + d_k, \quad (5)$$

The prediction (5) can be transformed to the following form:

$$\begin{aligned} y_{k+i|k} = y_k + \sum_{n=i+1}^{p_d-1} a_n \cdot \Delta u_{k-n+i} + a_{p_d} \cdot \sum_{n=p_d}^{p_d+i-1} \Delta u_{k-n+i} - \sum_{n=1}^{p_d-1} a_n \cdot \Delta u_{k-n} \\ + \sum_{n=1}^i a_n \cdot \Delta u_{k-n+i|k}. \end{aligned} \quad (6)$$

Note that in Formula (6), only the last component depends on future control changes; the others depend on the past. Thus, the other components are elements of the free response:

$$\tilde{y}_{k+i|k} = y_k + \sum_{n=i+1}^{p_d-1} a_n \cdot \Delta u_{k-n+i} + a_{p_d} \cdot \sum_{n=p_d}^{p_d+i-1} \Delta u_{k-n+i} - \sum_{n=1}^{p_d-1} a_n \cdot \Delta u_{k-n}. \quad (7)$$

In such a case, the prediction of the process output values y expressed in the matrix–vector form can be decomposed into two parts:

$$\mathbf{y} = \tilde{\mathbf{y}} + \mathbf{A} \cdot \Delta \mathbf{u} , \quad (8)$$

where $\tilde{\mathbf{y}} = [\tilde{y}_{k+1|k}, \dots, \tilde{y}_{k+p|k}]$ is the free response of the plant, which describes the influence of the past values of control signals on the process output; the form of the free response depends on the model used for prediction—see, e.g., [7,8,10,12]. The second component in Formula (8), the forced response, depends on future changes in the control signal. The dynamic matrix $\mathbf{A}$ is given by the following formula:

$$\mathbf{A} = \begin{bmatrix} a_1 & 0 & \dots & 0 & 0 \\ a_2 & a_1 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_p & a_{p-1} & \dots & a_{p-s+2} & a_{p-s+1} \end{bmatrix} . \quad (9)$$

The free response (7) written in vector–matrix form is as follows:

$$\tilde{\mathbf{y}} = \overleftarrow{\mathbf{y}} + \overleftarrow{\mathbf{A}} \cdot \Delta \overleftarrow{\mathbf{u}} , \quad (10)$$

where

$$\overleftarrow{\mathbf{A}} = \begin{bmatrix} a_2 - a_1 & a_3 - a_2 & \dots & a_{p_d} - a_{p_d-1} \\ a_3 - a_1 & a_4 - a_2 & \dots & a_{p_d} - a_{p_d-1} \\ \vdots & \vdots & \ddots & \vdots \\ a_{p+1} - a_1 & a_{p+2} - a_2 & \dots & a_{p_d} - a_{p_d-1} \end{bmatrix} , \quad (11)$$

$$\Delta \overleftarrow{\mathbf{u}} = [\Delta u_{k-1}, \dots, \Delta u_{k-p_d+1}] , \quad (12)$$

$$\overleftarrow{\mathbf{y}} = [y_k, \dots, y_k] , \quad (13)$$

where $\overleftarrow{\mathbf{y}}$ vector has p elements.

After applying the prediction (8), the performance function (1) is transformed into a function that depends quadratically on the decision variables $\Delta \mathbf{u}$:

$$J_{LMPC} = (\bar{\mathbf{y}} - \tilde{\mathbf{y}} - \mathbf{A} \cdot \Delta \mathbf{u})^T \cdot (\bar{\mathbf{y}} - \tilde{\mathbf{y}} - \mathbf{A} \cdot \Delta \mathbf{u}) + \lambda \cdot \Delta \mathbf{u}^T \cdot \Delta \mathbf{u} . \quad (14)$$

If, in each iteration, the optimization problem with performance function (14) is solved without constraints, then it has the following solution given by the analytical formula:

$$\Delta \mathbf{u} = (\mathbf{A}^T \cdot \mathbf{A} + \lambda \cdot \mathbf{I})^{-1} \cdot \mathbf{A}^T \cdot (\bar{\mathbf{y}} - \tilde{\mathbf{y}}) . \quad (15)$$

After substituting the free response (10) into (15), we obtain

$$\Delta \mathbf{u} = \mathbf{K} \cdot (\bar{\mathbf{y}} - \tilde{\mathbf{y}}) = \mathbf{K} \cdot (\bar{\mathbf{y}} - \overleftarrow{\mathbf{y}} - \overleftarrow{\mathbf{A}} \cdot \Delta \overleftarrow{\mathbf{u}}) , \quad (16)$$

where $\mathbf{K} = (\mathbf{A}^T \cdot \mathbf{A} + \lambda \cdot \mathbf{I})^{-1} \cdot \mathbf{A}^T$. Note that if $\Delta \mathbf{u}$ is calculated only for control, it is sufficient to calculate only the value of the control increment for the current instant $\Delta u_{k|k}$, which is described by the following formula:

$$\Delta u_{k|k} = \mathbf{K}_1 \cdot (\bar{\mathbf{y}} - \overleftarrow{\mathbf{y}} - \overleftarrow{\mathbf{A}} \cdot \Delta \overleftarrow{\mathbf{u}}) , \quad (17)$$

where $\mathbf{K}_1$ is the 1st row of the matrix $\mathbf{K}$. In this case, the value of the future control increment $\Delta u_{k|k}$ depends in the analytical DMC algorithm on the current value of the output and the past control increments. In addition, the matrix $\mathbf{K}$ and its individual rows remain the same in each iteration. In this case, it is sufficient to determine this matrix once in the controller design stage. As a result, a large part of the calculations are performed offline.

If one assumes that reference trajectory is constant on the prediction horizon, then the vector $\bar{y}$ from (2) becomes

$$\bar{y} = [\bar{y}_k, \dots, \bar{y}_k], \quad (18)$$

where $\bar{y}_k$ is the current setpoint value. In such a case, the following control law can be obtained:

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

where $r_0 = \sum_{j=1}^p K_{1j}$, $[r_1, \dots, r_{p_d-1}] = K_1 \cdot \overleftarrow{A}$, $K_1 = [K_{11}, \dots, K_{1p}]$.

The output of the analytical DMC controller is the control increment. The following formula describes the control value:

$$u_{k|k} = u_{k-1} + \Delta u_{k|k}. \quad (20)$$

Applying an LMPC algorithm to a nonlinear process may result in unsatisfactory control performance, especially if the MPC algorithm should operate at different operating points. In such a case, the operation of the control system can be improved using an MPC algorithm based on a nonlinear model.

2.3. Efficient SSDMC Algorithm

The algorithm proposed in this article is formulated in such a way that solving the optimization task online is avoided, because, due to the assumed form of the linear approximation used to obtain the dynamic matrix and the appropriate choice of the tuning parameter λ depending on time, it is possible to simplify the control law. Let us assume that one has a static model of the process and the step response of the process. These models will be used skillfully to formulate the proposed analytical algorithm. Of course, such an algorithm will probably perform worse than an algorithm based on a well-identified model, like the one proposed, e.g., in [26,27]. However, at a relatively low cost (assuming one has a steady-state process model, which is often developed for the setpoint optimization layer), one can obtain an algorithm that should offer better control quality than the LMPC algorithm; this is especially true when the process has strong static nonlinearity.

Assume that in each iteration of the algorithm, for the current operating point, a linear approximation of the process model is obtained using the following formula:

$$y_k^L = K_k \cdot \left(\sum_{i=1}^{p_d-1} a_i \cdot \Delta u_{k-i} + a_{p_d} \cdot u_{k-p_d} \right), \quad (21)$$

where K_k is the slope of the steady-state characteristic of the process (the gain of the control plant) around the current operating point; it is also assumed that the coefficients of the step response a_i are normalized in such a way that a_{p_d} is equal to 1. Due to the form of the adopted linear approximation of the process model, the proposed algorithm is an extension of the DMC algorithm. Thus, it will be abbreviated as the SSDMC algorithm (Steady-State Model-Based DMC—DMC algorithm using the steady-state characteristic of the process). The dynamic matrix obtained in each iteration will thus be given by the following formula:

$$A_k = K_k \cdot A, \quad (22)$$

where A is described by Equation (9).

The free response is calculated using the following formula:

$$\tilde{y} = \overleftarrow{y} + \overleftarrow{A}_k \cdot \Delta \overleftarrow{u}, \quad (23)$$

where this time, $\overleftarrow{A}_k$ changes at each sampling instant but depends on $\overleftarrow{A}$ as described by Equation (11):

$$\overleftarrow{A}_k = K_k \cdot \overleftarrow{A}. \quad (24)$$

Then, using the obtained dynamic matrix described by Equation (22) and the free response given by Equation (23), one obtains the output prediction described by the following formula:

$$y = \tilde{y} + A_k \cdot \Delta u = \overleftarrow{y} + \overleftarrow{A}_k \cdot \Delta \overleftarrow{u} + A_k \cdot \Delta u. \quad (25)$$

Next, the prediction given by (25) is used to formulate the quadratic optimization problem, with the following performance index:

$$J_{SSDMC} = (\tilde{y} + A_k \cdot \Delta u - \bar{y})^T \cdot (\tilde{y} + A_k \cdot \Delta u - \bar{y}) + \lambda_k \cdot \Delta u^T \cdot \Delta u. \quad (26)$$

As a result of solving the optimization task with the quadratic performance index defined in Equation (26) without constraints, the following result is obtained:

$$\Delta u = \left(A_k^T \cdot A_k + \lambda_k \cdot I \right)^{-1} \cdot A_k^T \cdot (\bar{y} - \tilde{y}). \quad (27)$$

It is important to note that in Equation (27), the dynamic matrix A_k changes at each sampling instant due to the nonlinear nature of the control plant. Therefore, an additional transformation must be made to account for these changes. Note that $A_k^T \cdot A_k = K_k^2 \cdot A^T \cdot A$. Then, after assuming the coefficient $\lambda_k = K_k^2 \cdot \lambda$ (which is crucial for obtaining a compact control law), the following relation is obtained:

$$\left(A_k^T \cdot A_k + \lambda_k \cdot I \right) = K_k^2 \cdot \left(A^T \cdot A + \lambda \cdot I \right). \quad (28)$$

Therefore, after substituting Equations (22)–(24) into (27), along with performing the appropriate mathematical transformations, one obtains the following result:

$$\Delta u = \frac{1}{K_k} \cdot K \cdot \left(\bar{y} - \overleftarrow{y} + K_k \cdot \overleftarrow{A} \cdot \Delta \overleftarrow{u} \right), \quad (29)$$

where the matrix $K = (A^T \cdot A + \lambda \cdot I)^{-1} \cdot A^T$ has the same form as in the classical DMC algorithm. Then, as in the classical DMC algorithm, the main part of the calculations in the SSDMC algorithm can be efficiently performed once at the algorithm design stage (in the offline phase). This approach allows for a significant reduction in the computational demand required by the algorithm during its execution.

After further transformations, one obtains the following formula, which describes the current increment of the manipulated variable:

$$\Delta u_{k|k} = \frac{1}{K_k} \cdot K_1 \cdot \left(\bar{y} - \overleftarrow{y} - K_k \cdot \overleftarrow{A} \cdot \Delta \overleftarrow{u} \right) = \frac{1}{K_k} \cdot K_1 \cdot (\bar{y} - \overleftarrow{y}) - K_1 \cdot \left(\overleftarrow{A} \cdot \Delta \overleftarrow{u} \right). \quad (30)$$

Then, if a constant reference trajectory as in Equation (18) is assumed, the following control law can be finally obtained:

$$\Delta u_{k|k} = \Delta u_k = r_{0,k} \cdot e_k + \sum_{i=1}^{p_d-1} r_i \cdot \Delta u_{k-i}, \quad (31)$$

where $r_{0,k} = \frac{1}{K_k} \cdot r_0$ and $r_0, r_1, \dots, r_{p_d}$ are derived in the same way as in the control law presented in Equation (19) of the analytical DMC algorithm.

Although the control law obtained in the SSDMC algorithm is very close to that of the standard Dynamic Model Control algorithm, it is important to note that the SSDMC algorithm offers considerably better control quality when applied to nonlinear processes compared to its predecessor.

Tuning of the parameters of the SSDMC controller can be achieved in the same way as in the case of the DMC controller. The sampling period should be chosen in such a way that the delay or inverse response is included in the step-response model. The prediction horizon can be chosen to be equal to the dynamics horizon; t_{j0} is the safest approach, and in the case of the SSDMC algorithm, the control law is obtained. Therefore, the length of the prediction horizon (and also the control horizon) has no influence on the online computational demand of the controller. Both horizons can thus be chosen during offline simulation experiments without any limitations. Moreover, the SSDMC algorithm was designed in such a way that the λ parameter has a similar influence on the behavior of the control system as in the case of the standard DMC algorithm.

3. Results

3.1. Control Plant

The control plant is a nonlinear, isothermal CSTR in which the van de Vusse reaction takes place—a popular benchmark with difficult dynamics, used in much research to test newly developed control algorithms (see, e.g., [26,27,44–47]) as well as in MIMO configuration, where not only concentration, but also temperature [48,49], is stabilized. Also, other CSTRs with less demanding dynamics that are controlled in the MIMO configuration can be found in the literature; see, e.g., [50,51]. The process model of the reactor is as follows [52]:

$$\begin{aligned} \frac{dC_A}{dt} &= -k_1 \cdot C_A - k_3 \cdot C_A^2 + \frac{F}{V}(C_{Af} - C_A) , \\ \frac{dC_B}{dt} &= k_1 \cdot C_A - k_2 \cdot C_B - \frac{F}{V}C_B , \end{aligned} \quad (32)$$

where C_A, C_B —the concentrations of components A and B, respectively; F —the flow rate, where it is assumed that its upper bound is equal to $F_{max} = 60$ L/h; V —the volume in which the reaction takes place (assumed constant and $V = 1$ L); and C_{Af} —the concentration of component A in the inlet flow stream ($C_{Af} = 10$ mol/L if not declared otherwise). The parameters of the model have the following values: $k_1 = 50$ L/h, $k_2 = 100$ L/h, and $k_3 = 10$ L/(h · mol). The output variable is the concentration C_B of substance B, the manipulated variable is the flow rate F , and C_{Af} is the disturbance variable.

The control plant under consideration is highly nonlinear. Its step responses obtained for different flow rate values ($F_1 = 54.3$ L/h, $F_2 = 44.3$ L/h, $F_3 = 34.3$ L/h, $F_4 = 24.3$ L/h, $F_5 = 14.3$ L/h) are shown in Figure 1. Although the difference between the subsequent values of the flow rate is the same ($\Delta F = 10$ L/h) the obtained responses are asymmetric; it is clear that the control plant has both nonlinear dynamics and statics. Moreover, inverse response is present in the step responses obtained.

The same control plant was also used during the tests in [26,27], where advanced numerical versions of the Fuzzy MPC algorithms were proposed and tested; the nonlinear steady-state characteristic of the process is depicted in Figure 2. The algorithm proposed in the current article exploits this characteristic.

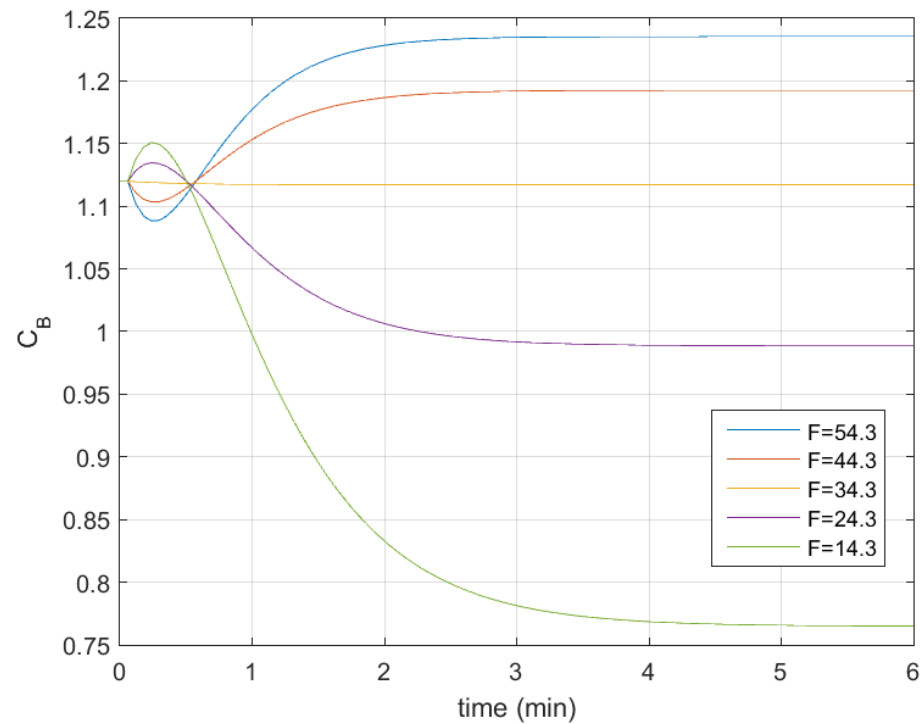


Figure 1. Step responses of control plant.

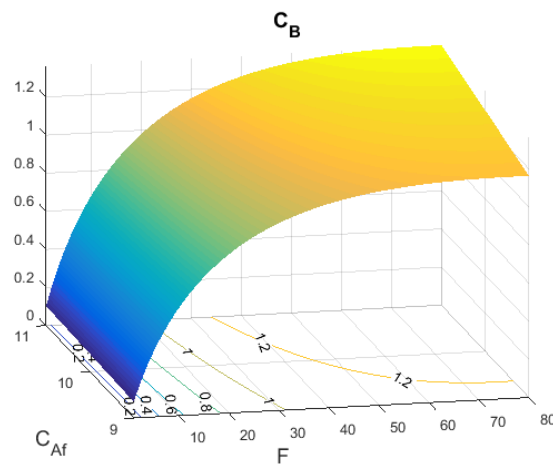


Figure 2. Steady-state characteristic of control plant.

To calculate the current gain K_k , the equation that describes the steady-state characteristic was obtained from Equation (32). The equations were rewritten for the steady-state; the constant $V = 1$ L was also taken into account. Moreover, it is important to note that all variables representing the steady-state are denoted by an overline to clearly distinguish them from their dynamic counterparts. The first equation from (32), therefore, transforms into the following equation:

$$0 = -k_1 \cdot \bar{C}_A - k_3 \cdot \bar{C}_A^2 + \bar{F}(\bar{C}_{Af} - \bar{C}_A) . \quad (33)$$

And the second one transforms into the following equation:

$$0 = k_1 \cdot \bar{C}_A - k_2 \cdot \bar{C}_B - \bar{F}\bar{C}_B . \quad (34)$$

After rewriting Equation (33), one obtains

$$-k_3 \cdot \bar{C}_A^2 - (k_1 + \bar{F})\bar{C}_A + \bar{F} \cdot \bar{C}_{Af} = 0 ; \quad (35)$$

and after transforming Formula (34), the following expression that describes $\bar{C}_B$ is obtained:

$$\bar{C}_B = \frac{k_1 \cdot \bar{C}_A}{k_2 + \bar{F}} . \quad (36)$$

After conducting the necessary and appropriate transformations, and assuming that the operating point would be positioned on the left-hand side of the steady-state characteristic curve, the following equation was obtained as a result of this analysis:

$$\bar{C}_B(\bar{F}) = \frac{k_1 + \bar{F} - \sqrt{(k_1 + \bar{F})^2 + 4 \cdot k_3 \cdot \bar{F} \cdot \bar{C}_{Af}}}{-2 \cdot k_3 \cdot \frac{k_2 + \bar{F}}{k_1}} . \quad (37)$$

Then, the approximate value of the derivative was calculated using the equation, which takes into account the change in the function's value relative to a small change in the input variable:

$$K_k = \frac{\bar{C}_B(\bar{F} + \Delta\bar{F}) - \bar{C}_B(\bar{F})}{\Delta\bar{F}} , \quad (38)$$

where $\Delta\bar{F} = 0.01$ was assumed.

3.2. Simulation Results

The performance of the SSDMC algorithm was compared with the classical DMC algorithm's performance in the chemical reactor's control system. In both algorithms, the following values of parameters were assumed: prediction horizon $p = 70$, control horizon $s = 35$, and coefficient $\lambda = 0.001$. Moreover, a constant reference trajectory on the prediction horizon was assumed. The control systems were simulated in Matlab on a PC equipped with the AMD Athlon II X3 460 processor running at 3.40 GHz. A sampling time equal to $T_s = 3.6$ s was assumed. The control plant was simulated using the Dormand–Prince method implemented in the *ode45* solver. The structure of the control system is shown in Figure 3.

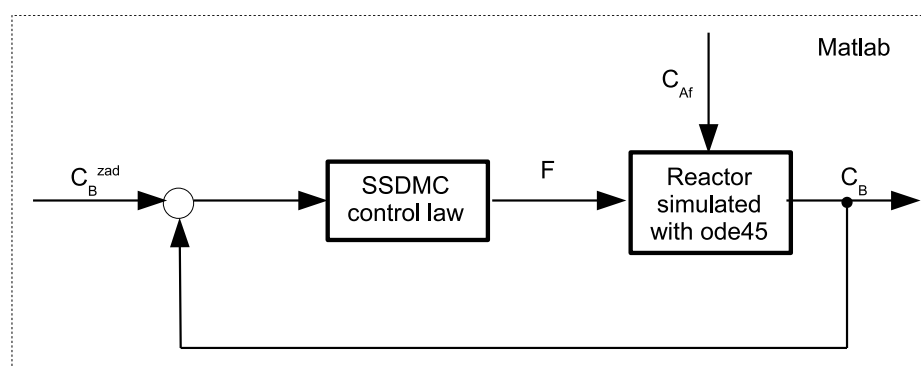


Figure 3. Structure of simulated control system.

The responses of control systems with DMC and SSDMC controllers are shown in Figure 4. The response to the change of the setpoint to $\bar{C}_{B1} = 1$ mol/L obtained in the control system with the SSDMC controller (red lines) practically does not have overshoot; therefore, the control time is slightly shorter than in the case when the DMC controller (blue lines) is used. In the case of disturbance response, the SSDMC controller does not generate overshoot, but the maximal temporary control error is slightly higher than in the control system with the DMC controller.

Moreover, the DMC controller compensates the disturbance faster. On the other hand, the control signal achieves much smaller values than with the SSDMC controller, generating the calm control signal. In the case of responses to the change of the setpoint to $\bar{C}_{B2} = 1.25$ mol/L, the SSDMC controller outperforms the DMC controller. It is significantly faster than the DMC controller and generates responses without overshoot. This time, the SSDMC controller compensates the disturbance faster than the DMC controller. However, the maximal temporary control error is slightly bigger in the control system with the SSDMC controller.

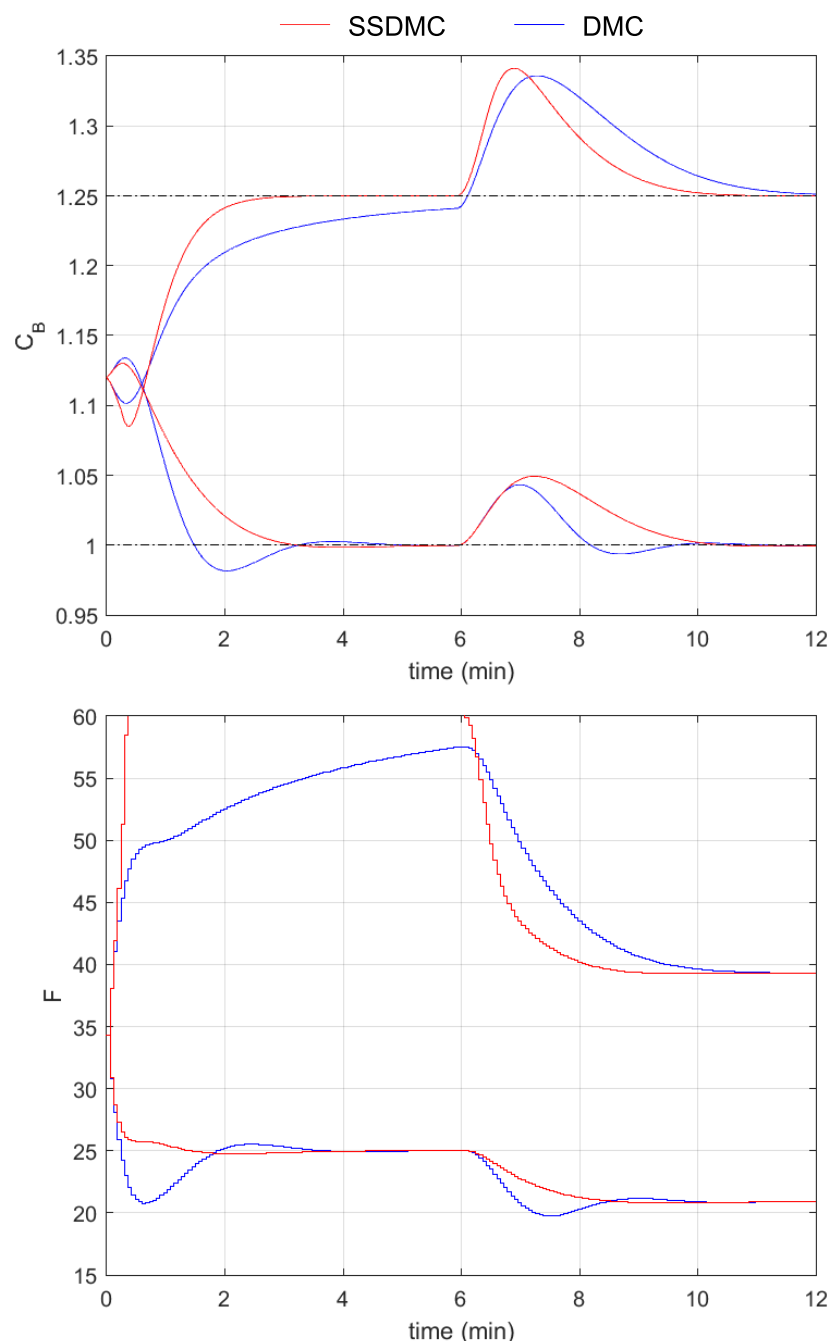


Figure 4. Responses of control system to changes in setpoint to $\bar{C}_{B1} = 1$ mol/L and $\bar{C}_{B2} = 1.25$ mol/L and to change in disturbance by 10% in 6th minute of experiment; $\lambda = 0.001$; SSDMC—red lines, DMC—blue lines.

To improve the operation of the SSDMC controller, experiments with decreasing λ coefficient were conducted; the results are depicted in Figure 5. Initial responses (for $\lambda = 0.001$) are drawn with red lines; the ones obtained for λ decreased 10 times—with blue lines, and for λ

decreased 100 times—with magenta lines. Decreasing λ did not influence the overshoot—it is not present at all in responses to the change of the setpoint to $\bar{C}_{B1} = 1.25$ mol/L and almost not present in responses to the change of the setpoint to $\bar{C}_{B1} = 1$ mol/L. Decreasing λ in all cases brought a decrease in the maximal temporary control error in disturbance responses. Decreasing λ 10 times visibly accelerated the control system's responses. However, the further decrease in λ to 10^{-5} almost did not influence the control time. Moreover, in the latter case and in the case of the change of the setpoint to $\bar{C}_{B1} = 1$ mol/L, oscillations of the control signal at the beginning of the experiment occurred. For the next experiment, $\lambda = 10^{-4}$ was chosen.

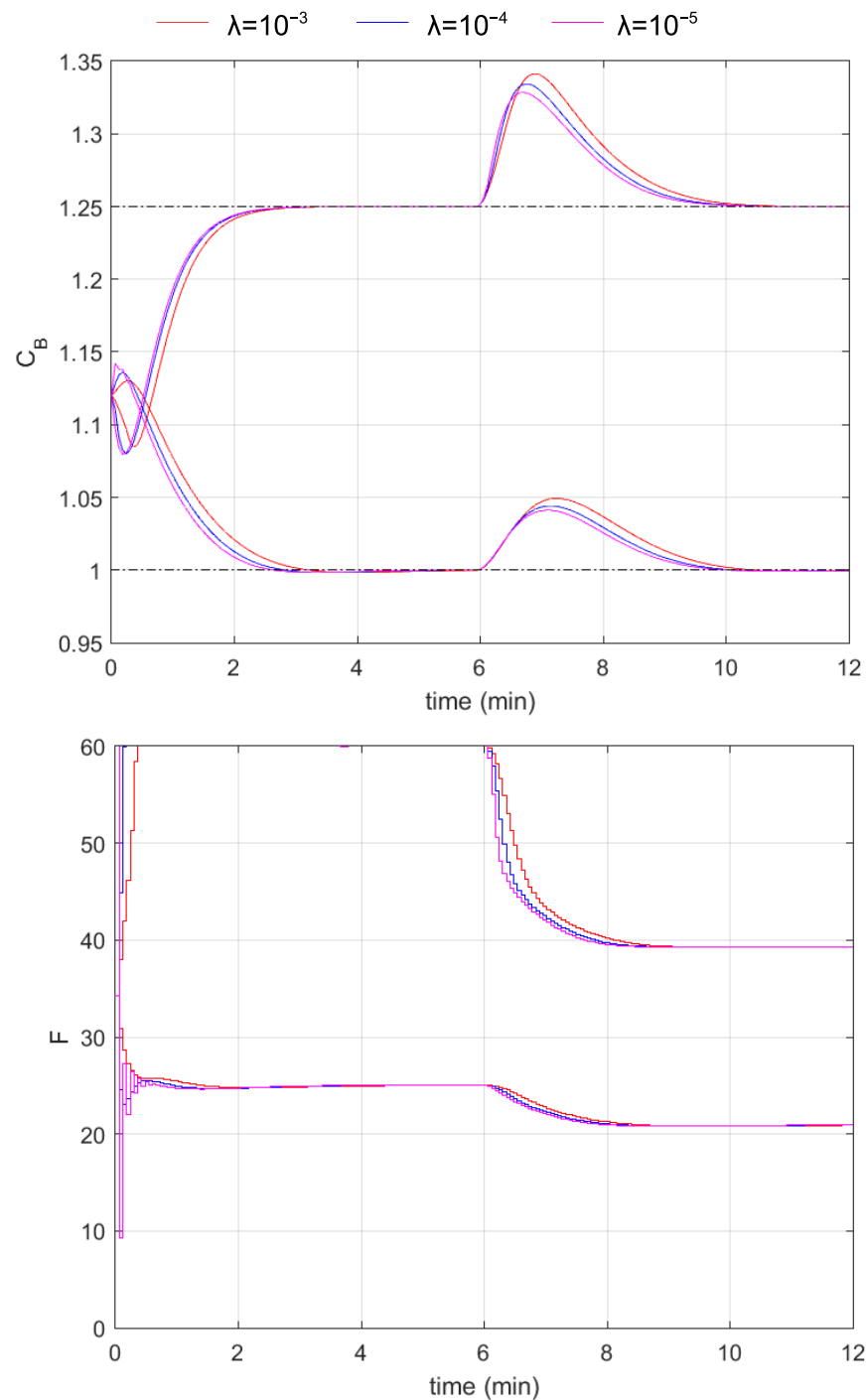


Figure 5. Responses of the control system with the SSDMC controller to changes in the setpoint to $\bar{C}_{B1} = 1$ mol/L and $\bar{C}_{B2} = 1.25$ mol/L and to the change of the disturbance by 10% in the 6th minute of the experiment. $\lambda = 10^{-3}$ —red lines; $\lambda = 10^{-4}$ —blue lines; $\lambda = 10^{-5}$ —magenta lines.

In addition, experiments with different values of the control horizon s were performed; the results are shown in Figure 6. The control horizon gradually decreased from $s = 35$. Interestingly, the responses obtained remained unchanged even for $s = 4$ (they are drawn with the red lines). A tiny change occurred for $s = 3$ (blue lines); the response was a little slower. The further decrease in the control horizon to $s = 2$ (green lines) and $s = 1$ (black dashed lines) continued this trend; the slowest responses were obtained for $s = 1$. In this last case, the maximal temporary control error in the disturbance response near $\bar{C}_{B1} = 1$ mol/L slightly increased, and oscillations occurred in the control signal at the beginning of the experiment. However, in all cases, the response to the setpoint change to $\bar{C}_{B2} = 1.25$ mol/L remained the same.

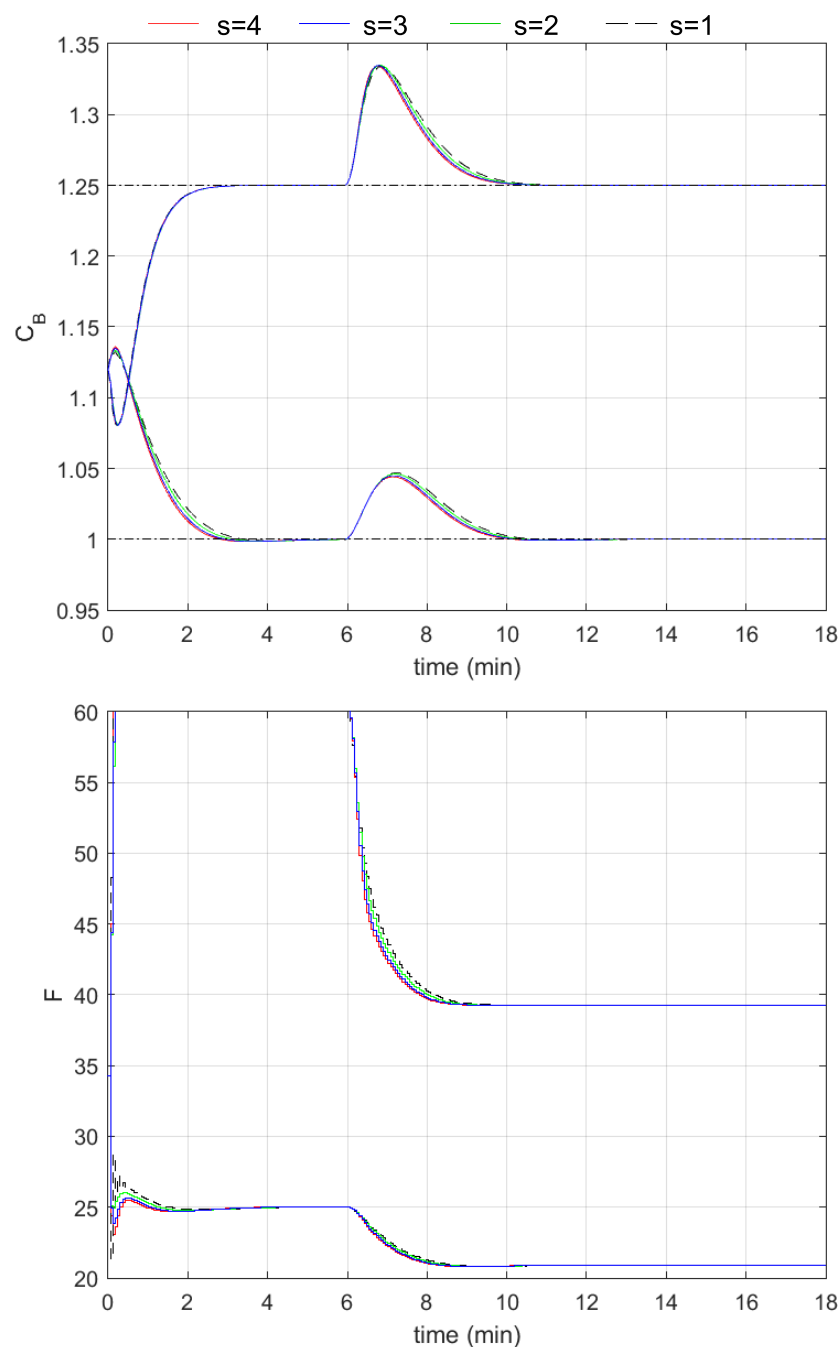


Figure 6. Responses of the control system with the SSDMC controller to changes in the setpoint to $\bar{C}_{B1} = 1$ mol/L and $\bar{C}_{B2} = 1.25$ mol/L and to the change of the disturbance by 10% in the 6th minute of the experiment. $\lambda = 10^{-4}$; s between 4 and 35—red lines; $s = 3$ —blue lines; $s = 2$ —green lines; $s = 1$ —black dashed lines.

Experiments were also conducted to test the robustness of the proposed algorithm. In Figure 7, responses of the control system to the change in the setpoint to $\bar{C}_{B1} = 1$ mol/L and $\bar{C}_{B2} = 1.25$ mol/L and to the change in the disturbance by 10% in the 6th minute of the experiment are shown; the responses are shown when the output was subject to a delay of one sampling time that was not included in the model the algorithm was designed with (blue lines). Despite the delay not being known to the controller, it operates only slightly worse than in the ideal case. The responses to setpoint changes are slightly slower. The overshoot in the response to the setpoint change to $\bar{C}_{B1} = 1$ mol/L increased a little. The responses to the disturbance change worsened. Also, the maximal control error was larger especially when the setpoint was equal to $\bar{C}_{B2} = 1.25$ mol/L.

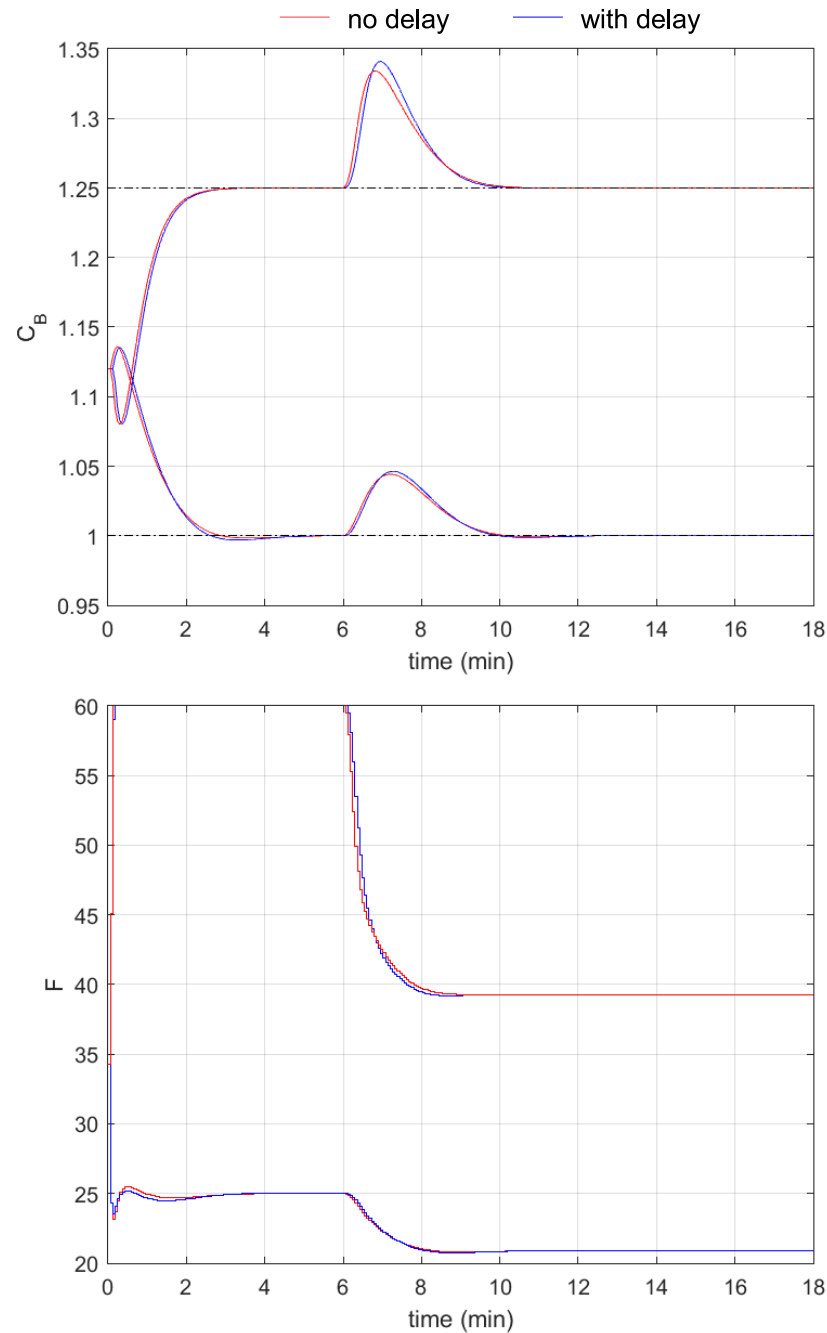


Figure 7. Responses of the control system with the SSDMC controller to changes in the setpoint to $\bar{C}_{B1} = 1$ mol/L and $\bar{C}_{B2} = 1.25$ mol/L and to the change of the disturbance by 10% in the 6th minute of the experiment; $\lambda = 10^{-4}$; $s = 4$. No measurement delay—red lines; with additional measurement delay—blue lines.

In Figure 8, responses of the control system to the change of 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 are shown; the responses are given when the output was subject to a measurement noise. This was simulated using random numbers from the interval $[-0.005, 0.005]$ added to the output measurement and sent to the controller. The proposed SSDMC controller copes well with the measurement noise. It achieves the setpoints without problem, even $\bar{C}_{B2} = 1.25$ mol/L though small oscillations of the control signal occurred when it hit the constraint. The change of disturbance was also compensated without problems.

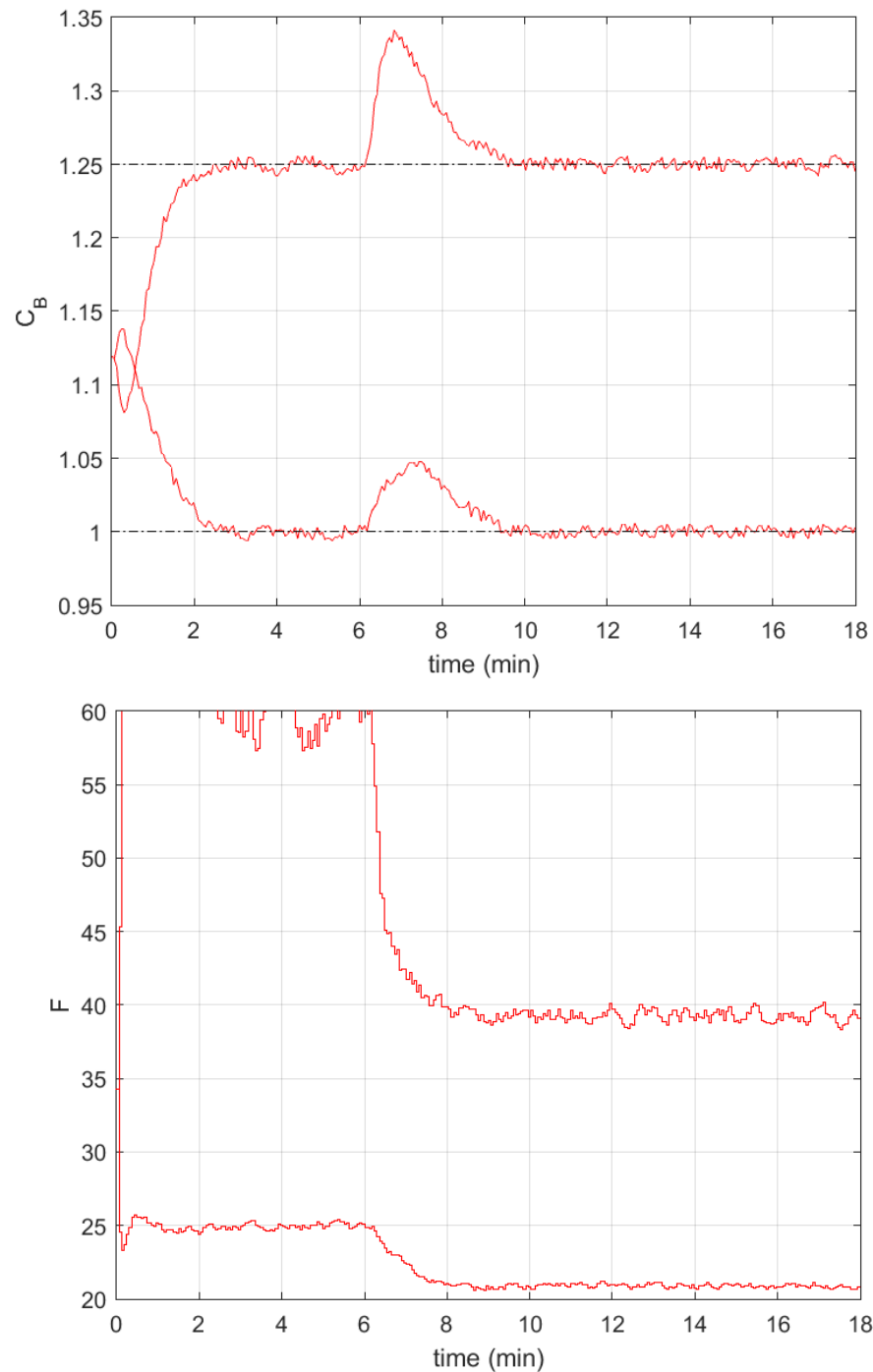


Figure 8. Responses of the control system with the SSDMC controller to changes in the setpoint to $\bar{C}_{B1} = 1$ mol/L and $\bar{C}_{B2} = 1.25$ mol/L and to the change of the disturbance by 10% in the 6th minute of the experiment; $\lambda = 10^{-4}$; $s = 4$; measurement noise added.

Experiments were also conducted with changes in values of the parameters k_1 , k_2 , and k_3 in the control plant model—a 10% change was made to each value so that the constraint was not active, and each change was made separately. The obtained responses of the control system to the change of 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 are shown in Figure 9. In all cases, the control system remains stable, though the obtained responses became worse. When $k_1 = 55$ L/h was assumed (the parameters k_2 and k_3 having nominal values), the responses drawn with the red lines were obtained. A significant overshoot occurred in the response to the setpoint change to $\bar{C}_{B2} = 1.25$ mol/L, and a small overshoot occurred in the case of the setpoint change to $\bar{C}_{B2} = 1$ mol/L. The response to the disturbance change near $\bar{C}_{B2} = 1$ mol/L did not change, and near $\bar{C}_{B2} = 1.25$ mol/L, it became better (a smaller maximal control error was obtained compared to the nominal case).

When $k_2 = 90$ L/h was assumed (k_1 and k_3 parameters having nominal values), the responses drawn with the blue lines were obtained. The biggest overshoot occurred in the response to the setpoint change to $\bar{C}_{B2} = 1.25$ mol/L, a slightly larger overshoot than that obtained in the previous case for the setpoint change to $\bar{C}_{B2} = 1$ mol/L. The responses are the slowest among all the ones depicted in Figure 9. The responses to the disturbance change are almost the same as in the case where only the k_1 parameter was changed.

When $k_3 = 9$ l/(h · mol) was assumed (k_1 and k_2 parameters having nominal values), the responses drawn with the magenta lines were obtained. The smallest overshoot occurred in the response to the setpoint change to $\bar{C}_{B2} = 1.25$ mol/L, as well as for the setpoint change to $\bar{C}_{B2} = 1$ mol/L. Unfortunately, in the response to the setpoint change to $\bar{C}_{B2} = 1.25$ mol/L, oscillations occurred. The response to the disturbance change near $\bar{C}_{B2} = 1$ mol/L is almost the same as in the other cases, while that near $\bar{C}_{B2} = 1$ mol/L is the worst; the maximal control error is the largest among the cases considered.

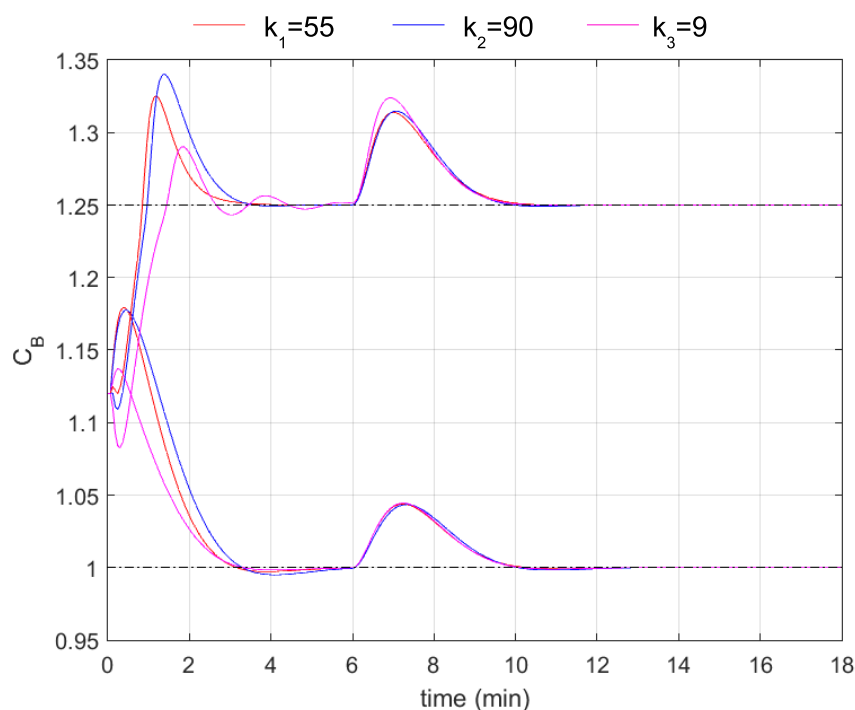


Figure 9. Cont.

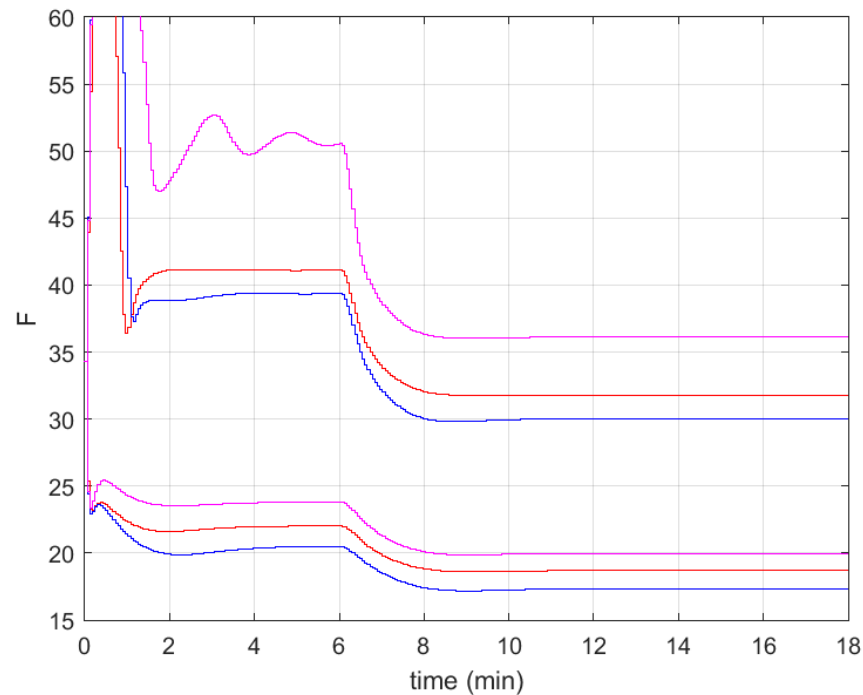


Figure 9. Responses of the control system with the SSDMC controller to changes in the setpoint to $\bar{C}_{B1} = 1$ mol/L and $\bar{C}_{B2} = 1.25$ mol/L and to the change of the disturbance by 10% in the 6th minute of the experiment. $\lambda = 10^{-4}$; $s = 4$; k_1 changed to 55 L/h ($k_2 = 100$ 1/h, $k_3 = 10$ L/(h · mol))—red lines; k_2 changed to 90 1/h ($k_1 = 50$ L/h, $k_3 = 10$ 1/(h · mol))—blue lines, k_3 changed to 9 L/(h · mol) ($k_1 = 50$ L/h, $k_2 = 100$ L/h)—magenta lines.

4. Discussion and Conclusions

It is worth noting that the computational complexity of both DMC and SSDMC controllers is similar. In the SSDMC algorithm, in addition to the operations performed in the DMC algorithm, only the slope of the steady-state characteristic (the derivative of a specific function) needs to be calculated additionally, which does not significantly complicate the proposed algorithm. Moreover, in the example under consideration, determining the derivative is particularly easy due to the relative simplicity of obtaining the analytical formula for calculating it.

The main tuning parameter, λ , is still available in the SSDMC controller and has the same meaning as in the standard algorithms. It decides on the size of the penalty for the control signal changes. However, inside the algorithm, it depends on the process gain at the current operating point to obtain the compact control law.

The next main tuning parameter, the control horizon s , does not influence the responses obtained as much as the parameter λ . Moreover, its decrease slightly slows down the responses obtained, worsening the control quality. On the other hand, thanks to the formulation of the SSDMC algorithm in the form of the control law, the longer control horizon does not influence the numerical complexity of the algorithm.

An important advantage of the proposed algorithm is its robustness to measurement noise, measurement delay, and the uncertainty of process parameters. Even when the parameters of the control plant were significantly changed, the controller managed to stabilize the reactor and maintain satisfactory performance.

The use of a dynamic model in the form of the process step response and static characteristic of the process, combined in a skillful way in order to synthesize the MPC algorithm, resulted in a computationally efficient predictive control algorithm in which the main part of the calculations needed to determine the control value is performed offline

as in a conventional, analytical DMC algorithm. However, note that despite the relative simplicity of the controller obtained and the relatively straightforward way of receiving it, it can offer much better control quality than the conventional DMC controller based on the linear model.

Funding: This research received no external funding.

Data Availability Statement: The data are contained within the article.

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

Abbreviations

The following abbreviations are used in this manuscript:

CSTR	Continuous Stirred-Tank Reactor
DMC	Dynamic Matrix Control
LMI	Linear Matrix Inequality
LMPC	Linear Model-Based Model Predictive Control
MPC	Model Predictive Control
NMPC	Nonlinear Model-Based MPC
PID	Proportional–Integral–Derivative
SISO	Single-Input Single-Output
SSDMC	Steady-State Model-Based Dynamic Matrix Control

References

1. Arteaga, M.A.; Guajardo-Benavides, E.J.; Sánchez-Sánchez, P. Stability Analysis and Experimental Validation of Standard Proportional-Integral-Derivative Control in Bilateral Teleoperators with Time-Varying Delays. *Algorithms* **2024**, *17*, 580. [\[CrossRef\]](#)
2. Candan, F.; Dik, O.F.; Kumbasar, T.; Mahfouf, M.; Mihaylova, L. Real-Time Interval Type-2 Fuzzy Control of an Unmanned Aerial Vehicle with Flexible Cable-Connected Payload. *Algorithms* **2023**, *16*, 273. [\[CrossRef\]](#)
3. Kong, W.; Zhang, H.; Yang, X.; Yao, Z.; Wang, R.; Yang, W.; Zhang, J. PID control algorithm based on multistrategy enhanced dung beetle optimizer and back propagation neural network for DC motor control. *Sci. Rep.* **2024**, *14*, 28276. [\[CrossRef\]](#)
4. Mao, W.-L.; Chen, S.-H.; Kao, C.-Y. Fractional-Order Fuzzy PID Controller with Evolutionary Computation for an Effective Synchronized Gantry System. *Algorithms* **2024**, *17*, 58. [\[CrossRef\]](#)
5. Guo, X.; Shirkhani, M.; Ahmed, E.M. Machine-Learning-Based Improved Smith Predictive Control for MIMO Processes. *Mathematics* **2022**, *10*, 3696. [\[CrossRef\]](#)
6. Mohammadzahari, G.M.; Ghodsi, M. An Enhanced Smith-Predictor-based Control System for delayed MIMO Processes and its Use on a CSTR with Multiple Time-delays. In Proceedings of the 2023 28th International Conference on Automation and Computing (ICAC), Birmingham, UK, 30 August–1 September 2023; pp. 1–6.
7. Camacho, E.F.; Bordons, C. *Model Predictive Control*; Springer: London, UK, 1999.
8. Maciejowski, J.M. *Predictive Control with Constraints*; Prentice Hall: Harlow, UK, 2002.
9. Nebeluk, R.; Marusak, P. Efficient MPC algorithms with variable trajectories of parameters weighting predicted control errors. *Arch. Control Sci.* **2020**, *30*, 325–363.
10. Rossiter, J.A. *Model-Based Predictive Control: A Practical Approach*; CRC Press: Boca Raton, FL, USA, 2003.
11. Sands, T. Comparison and Interpretation Methods for Predictive Control of Mechanics. *Algorithms* **2019**, *12*, 232. [\[CrossRef\]](#)
12. Tatjewski, P. *Advanced Control of Industrial Processes; Structures and Algorithms*; Springer: London, UK, 2007.
13. Ławryńczuk, M. *Computationally Efficient Model Predictive Control Algorithms: A Neural Network Approach*; Springer: Berlin/Heidelberg, Germany, 2014.
14. Mohamed, M.A.-E.-H.; Abbas, H.S.; Shouran, M.; Kamel, S. A Neuro-Predictive Controller Scheme for Integration of a Basic Wind Energy Generation Unit with an Electrical Power System. *Energies* **2022**, *15*, 5839. [\[CrossRef\]](#)
15. Mohammadzahari, M.; Chen, L.; Mirsepahi, A.; Ghanbari, M.; Tafreshi, R. Neuro-Predictive Control of an Infrared Dryer with a Feedforward-Feedback Approach. *Asian J. Control* **2015**, *17*, 1972–1977. [\[CrossRef\]](#)
16. Abdelaal, M.; Schön, S. Predictive Path Following and Collision Avoidance of Autonomous Connected Vehicles. *Algorithms* **2020**, *13*, 52. [\[CrossRef\]](#)
17. Chen, H.; Allgöwer, F. A Quasi-Infinite Horizon Nonlinear Model Predictive Control Scheme with Guaranteed Stability. *Automatica* **1998**, *34*, 1205–1217. [\[CrossRef\]](#)

18. Tatjewski, P. Offset-free nonlinear model predictive control with state-space process models. *Arch. Control Sci.* **2017**, *27*, 595–615. [\[CrossRef\]](#)
19. 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. [\[CrossRef\]](#)
20. 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. [\[CrossRef\]](#)
21. 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. [\[CrossRef\]](#)
22. Ławryńczuk, M. Nonlinear state-space predictive control with on-line linearisation and state estimation. *Int. J. Appl. Math. Comput. Sci.* **2015**, *25*, 833–847. [\[CrossRef\]](#)
23. Morari, M.; Lee, J.H. Model predictive control: Past, present and future. *Comput. Chem. Eng.* **1999**, *23*, 667–682. [\[CrossRef\]](#)
24. Marusak, P. Efficient model predictive control algorithm with fuzzy approximations of nonlinear models. *LNCS* **2009**, *5495*, 448–457.
25. 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. [\[CrossRef\]](#)
26. Marusak, P. Numerically Efficient Fuzzy MPC Algorithm with Advanced Generation of Prediction: Application to a Chemical Reactor. *Algorithms* **2020**, *13*, 143. [\[CrossRef\]](#)
27. Marusak, P. Advanced Construction of Dynamic Matrix in Numerically Efficient Fuzzy MPC Algorithms. *Algorithms* **2021**, *14*, 25. [\[CrossRef\]](#)
28. Nebeluk, R.; Ławryńczuk, M. Fast Nonlinear Predictive Control Using Classical and Parallel Wiener Models: A Comparison for a Neutralization Reactor Process. *Sensors* **2023**, *23*, 9539. [\[CrossRef\]](#)
29. 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.
30. 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.
31. Johansen, T.A. Approximate explicit receding horizon control of constrained nonlinear systems. *Automatica* **2004**, *40*, 293–300. [\[CrossRef\]](#)
32. Bemporad, A.; Borrelli, F.; Morari, M. Piecewise linear optimal controllers for hybrid systems. In Proceedings of the 2000 American Control Conference, Chicago, IL, USA, 28–30 June 2000; Volume 2; pp. 1190–1194.
33. Bemporad, A.; Morari, M.; Dua, V.; Pistikopoulos, E.N. The explicit linear quadratic regulator for constrained systems. *Automatica* **2002**, *38*, 3–20. [\[CrossRef\]](#)
34. 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. [\[CrossRef\]](#)
35. Chaber, P.; Ławryńczuk, M. Fast analytical model predictive controllers and their implementation for STM32 ARMmicrocontroller. *IEEE Trans. Ind. Inform.* **2019**, *15*, 4580–4590. [\[CrossRef\]](#)
36. Wojtulewicz, A.; Ławryńczuk, M. Implementation of multiple-input multiple-output dynamic matrix control algorithm for fast processes using field programmable gate array. *IFAC-PapersOnLine* **2018**, *51*, 324–329. [\[CrossRef\]](#)
37. Bagheri, P.; Sedigh, A.K. Analytical approach to tuning of model predictive control for first-order plus dead time models. *IET Control Theory Appl.* **2013**, *7*, 1806–1817. [\[CrossRef\]](#)
38. Short, M. Move suppression calculations for well-conditioned MPC. *ISA Trans.* **2017**, *67*, 371–381. [\[CrossRef\]](#)
39. Zheng, H.; Zou, T.; Hu, J.; Yu, H. An offline optimization and online table look-up strategy of two-layer model predictive control. *IEEE Access* **2018**, *6*, 47433–47441. [\[CrossRef\]](#)
40. Abbas, H.S.; Hanema, J.; Tóth, R.; Mohammadpour, J.; Meskin, N. An improved robust model predictive control for linear parameter-varying input-output models. *Int. J. Robust Nonlinear Control* **2018**, *28*, 859–880. [\[CrossRef\]](#)
41. Ding, B.; Zou, T. A synthesis approach for output feedback robust model predictive control based-on input-output model. *J. Process Control* **2014**, *24*, 60–72. [\[CrossRef\]](#)
42. Yan, Y.; Zhang, L. Robust Model Predictive Control with Almost Zero Online Computation. *Mathematics* **2021**, *9*, 242. [\[CrossRef\]](#)
43. Marusak, P. A numerically efficient fuzzy MPC algorithm with fast generation of the control signal. *Int. J. Appl. Math. Comput. Sci.* **2021**, *31*, 59–71. [\[CrossRef\]](#)
44. Jain, A.; Taparia, R. Laguerre function based model predictive control for van-de-vusse reactor. In Proceedings of the 2nd IEEE International Conference on Power Electronics, Intelligent Control and Energy Systems, ICPEICES 2018, Delhi, India, 22–24 October 2018; Volume 3, pp. 1010–1015.
45. 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. [\[CrossRef\]](#)

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-Papers 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. Uçak, K. A Novel Model Predictive Runge–Kutta Neural Network Controller for Nonlinear MIMO Systems. *Neural Process. Lett.* **2020**, *51*, 1789–1833. [[CrossRef](#)]
50. Banerjee, S.; Khan, O.; El Mistiri, M.; Nandola, N.N.; Rivera, D.E. Data-Driven Control of Highly Interactive Systems using 3DoF Model-On-Demand MPC: Application to a MIMO CSTR. *IFAC-PapersOnLine* **2024**, *58*, 420–425. [[CrossRef](#)]
51. Mohammadzaheri, M.; Chen, L. Double-command fuzzy control of a nonlinear CSTR. *Korean J. Chem. Eng.* **2010**, *27*, 19–31. [[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](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.