10.24425/acs.2020.133502

Archives of Control Sciences
Volume 30(LXVI), 2020
No. 2, pages 325–363

Efficient MPC algorithms with variable trajectories of parameters weighting predicted control errors

ROBERT NEBELUK and PIOTR MARUSAK

Model predictive control (MPC) algorithms brought increase of the control system performance in many applications thanks to relatively easily solving issues that are hard to solve without these algorithms. The paper is focused on investigating how to further improve the control system performance using a trajectory of parameters weighting predicted control errors in the performance function of the optimization problem. Different shapes of trajectories are proposed and their influence on control systems is tested. Additionally, experiments checking the influence of disturbances and of modeling uncertainty on control system performance are conducted. The case studies were done in control systems of three control plants: a linear nonminimumphase plant, a nonlinear polymerization reactor and a nonlinear thin film evaporator. Three types of MPC algorithms were used during research: linear DMC, nonlinear DMC with successive linearization (NDMC–SL), nonlinear DMC with nonlinear prediction and linearization (NDMC–NPL). Results of conducted experiments are presented in greater detail for the control system of the polymerization reactor, whereas for the other two control systems only the most interesting results are presented, for the sake of brevity. The experiments in the control system of the linear plant were done as preliminary experiments with the modified optimization problem. In the case of control system of the thin film evaporator the researched mechanisms were used in the control system of a MIMO plant showing possibilities of improving the control system performance.

Key words: model predictive control, nonlinear systems, nonlinear models, nonlinear control, simulation, optimization

1. Introduction

Model Predictive Control (MPC) is an approach where process model is used in order to predict process behavior in the future. For nonlinear plants, with big delays, and bounded inputs or outputs, the use of MPC usually brings a very good system performance whereas controllers like PID would struggle. In practice, the

The Authors are with The Warsaw University of Technology, Institute of Control and Computation Engineering, Nowowiejska 15/19, 00–665 Warsaw, Poland.

Received 5.07.2019.

model used in the MPC algorithm is inaccurate, thus the modeling uncertainty can influence the designed control system performance. Fortunately, the algorithm can assess this uncertainty, include it in the prediction, and reduce its influence on the control system. Usually, all future, predicted control errors have the same weights in the optimization problem solved in each iteration of these algorithms like in [18] or e.g. in a control system of a multi-area interconnected power system described in [4]. The methods of tuning the weights but assuming them constant on the prediction horizon are described e.g. in [14–16, 19] and [20]. The experiments in which all weighting parameters of predicted control errors have been increased by the same amount on prediction horizon while the weighting parameter of future changes of the manipulated variable stays constant were described in [16]. In this case the control quality changes similarly to when the weighting parameter of future changes of the manipulated variable has been changed. MPC tuning strategy for MIMO (Multiple–Input Multiple–Output) control plants described by first order plus dead time (FOPDT) models is proposed in [15]. (The version of the method for SISO plants, in which only the weight for manipulated variable increments is set during the tuning process was proposed in [14].) Another method of MPC algorithm tuning, based on a static optimization is proposed in [20]. The weights are chosen depending on the output or manipulated variable they are related with. A method of tuning the MPC algorithm for MIMO plant is proposed in [19]. Values of both: weighting parameters of predicted control errors and the weighting parameters of future changes of the manipulated variables are appropriately set depending on the output or manipulated variable they are related with and properties of the linearized model of the control plant. The appropriate choice of the weights is used in [11] and [12] to design robust MPC algorithms.

However, the weights of predicted control errors can be different in different time instants from the prediction horizon. Thus, different trajectories of weighting parameters of predicted control errors are proposed and researched in this paper and their influence on the system performance is compared. General benefits offered by this mechanism and sketchy guidelines are mentioned in [13]. However, no detailed information about what shape of these trajectories would give the improved control quality is given. In this article influence of each proposed shape of trajectory on control system performance is described. The general suggestions regarding the shape of trajectories from [13] serve as a starting point for the research described in this paper.

In the field of robotics weighting parameter trajectory is used on a small scale. The most common problem to be solved is finding a reference trajectory for moving the robot's arm from point A to point B that would cost minimum energy and time [3]. An optimization problem with constraints is introduced to solve this problem and in weighting parameters trajectory only the last element has a weighting parameter equal to zero and all the other weighting parameters

are equal to one. This choice of values of weighting parameters is done with the use of expert knowledge of a designer who knows that it will improve the control performance. In this paper influence on the control system performance by assigning values equal to zero to respective weighting parameters is also presented. However, most of the proposed trajectories have nonzero weighting parameters, they are of a different shape, and are used in process control to improve the control system performance, not like in robotics, to improve the set point trajectory following.

The experiments with tuning model predictive controllers by changing the weighting parameter trajectory through an optimization method (e.g. particle swarm optimization) are described in [6] and [9]. The control performance obtained in the experiments described in [6] with weights variable on the prediction horizon (applied to an analytic algorithm) gave results suggesting that it is not necessary to use such an approach in the numerical algorithm. Note that it would complicate the optimization problem solved in order to tune the algorithm comparing to the case with weighting parameters constant on the prediction horizon. It is worth to also notice that in [9] the general formulation of the optimization problem contains weights variable on the prediction horizon. However, in the example it is assumed that they are constant on the prediction horizon and depend only on the output or manipulated variable they are related with. It is a very reasonable approach, because allowing weights variable on the prediction horizon causes the optimization problem to be much more difficult to solve (number of decision variables grows significantly). Therefore in this paper the optimization method is not used as not too practical.

This paper describes the advanced investigation of the topic. It is also checked in the paper, how different shapes of trajectory influence the control quality. It contains recommendations what kind of trajectory shape should be used in a given situation to improve control system performance. This article introduces also a new and easy-to-follow methodology of obtaining the best weighting parameter trajectory for a given problem, improving the control system performance the most.

In the next section the MPC algorithms used during experiments are described. Section 3 contains introductory research done in the control system of the linear, nonminimumphase plant presenting first experiments with the use of the proposed mechanism and showing possibilities offered by it. Section 4 presents detailed results obtained in the control systems of the nonlinear polymerization reactor; it contains many experiments with different shapes of weighting parameters; influence of disturbances and of modeling uncertainty on control system performance is also checked. Section 5 covers the research done in the control system of the nonlinear MIMO thin film evaporator showing how the shapes of weighting parameters trajectory can be used to improve control quality of the control system of a MIMO control plant. The last section summarizes all the obtained results.

2. Model predictive control algorithms

MPC approach was created in 1970s. The advantages such as: achieving great system performance for processes with difficult dynamics or with large number of manipulated and controlled variables, and allowing taking process constraints into consideration are the reason why this approach is used today in many industrial applications. Moreover, measured disturbances and information about model uncertainty can also be used by MPC algorithms to improve control system performance.

In current sampling instant k the MPC algorithms predict the outputs of the given process many sampling instants in the future, using a dynamic process model. The number of these instants is called the prediction horizon and denoted by N. Number of future changes of manipulated variables is called the control horizon and is denoted by N_u. The future changes of manipulated variables are calculated by finding a solution to an optimization problem of minimizing the following cost function [17]:

$$\min_{\Delta u} \left\{ \sum_{i=1}^{N} \left(\boldsymbol{y}^{\mathbf{ref}}_{k+i|k} - \boldsymbol{y}_{k+i|k} \right)^T \boldsymbol{\Psi}_i \left(\boldsymbol{y}^{\mathbf{ref}}_{k+i|k} - \boldsymbol{y}_{k+i|k} \right) + \sum_{i=0}^{N_u-1} \left(\boldsymbol{\Delta u}_{k+i|k} \right)^T \boldsymbol{\Lambda}_i \left(\boldsymbol{\Delta u}_{k+i|k} \right) \right\}, \tag{1}$$

where vectors $\boldsymbol{y}^{\mathbf{ref}}_{k+i|k}$ and $\boldsymbol{y}_{k+i|k}$ are of dimension n_y (n_y is the number of controlled outputs), while the vector of input increments $\boldsymbol{\Delta u}_{k+i|k}$ and matrix of weighting parameters $\boldsymbol{\Lambda}_i$ is of dimension n_u (n_u is the number of manipulated variables). Therefore, minimizing the cost function (1) consists in minimizing two quadratic sums where the first is of predicted control errors, which is calculated as difference between reference trajectories and future process outputs predicted on the prediction horizon, and the second one is of weighted future changes of the manipulated variables on the control horizon. The vectors in (1) are as follows:

$$\boldsymbol{y}^{\mathbf{ref}}_{k+i|k} = \begin{bmatrix} y^{\mathrm{ref\_1}}_{k+i|k} \\ y^{\mathrm{ref\_2}}_{k+i|k} \\ \vdots \\ y^{\mathrm{ref\_n_y}}_{k+i|k} \end{bmatrix}, \qquad \boldsymbol{y}_{k+i|k} = \begin{bmatrix} y^{1}_{k+i|k} \\ y^{2}_{k+i|k} \\ \vdots \\ y^{n_y}_{k+i|k} \end{bmatrix}, \qquad \boldsymbol{\Delta u}_{k+i|k} = \begin{bmatrix} \Delta u^{1}_{k+i|k} \\ \Delta u^{2}_{k+i|k} \\ \vdots \\ \Delta u^{n_u}_{k+i|k} \end{bmatrix},$$

where $y^{\mathrm{ref\_j}}_{k+i|k}$ is the reference value for i-th time instant from the prediction horizon, for j-th output, $y^{j}_{k+i|k}$ is the j-th output value predicted for the i-th time instant

from the prediction horizon, $\Delta u^j_{k+i|k}$ is the increment of the j-th control in the i-th time instant from the control horizon. The matrix $\mathbf{\Lambda}_i$ presented in (1) is as follows:

$$\mathbf{\Lambda}_i = \begin{bmatrix} \lambda_i^1 & 0 & 0 & 0 \\ 0 & \lambda_i^2 & 0 & \vdots \\ \vdots & 0 & \ddots & 0 \\ 0 & 0 & 0 & \lambda_i^{n_u} \end{bmatrix}. \tag{2}$$

The matrix $\mathbf{\Psi}_i$ weights all predicted control errors. It enables to choose which predicted control errors will be the most important and which the least important for the control system performance. This matrix is as follows:

$$\mathbf{\Psi}_i = \begin{bmatrix} \psi_i^1 & 0 & 0 & 0 \\ 0 & \psi_i^2 & 0 & \vdots \\ \vdots & 0 & \ddots & 0 \\ 0 & 0 & 0 & \psi_i^{n_y} \end{bmatrix}, \tag{3}$$

where each $(\psi_i^1, \psi_i^2, \ldots, \psi_i^{n_y})$ is a weighting parameter, their values are, in general, different on the prediction horizon.

For the experiments in the next sections system performance is evaluated by comparing overshoot and rise time of system outputs with the use of different weighting parameter trajectories variable or constant on prediction horizon. Overshoot is defined by the following equation [1]:

$$y_o = \frac{y(t_m) - y(\infty)}{y(\infty)} \cdot 100\%, \tag{4}$$

where: $y(\infty)$ – steady-state value,
t_m – peak time at which the peak value occurs,
$y(t_m)$ – peak value, which is the absolute value of y.

Rise time t_r is the time in which the response rises from 10% to 90% of the steady-state response [2].

2.1. DMC predictive control algorithm

2.1.1. SISO case

Predictive control algorithm DMC (*Dynamic Matrix Control*) was first used in petrochemical industry. The DMC controller uses step response linear model to predict process behavior. This model can be written as [17]:

$$y_k^M = \sum_{i=1}^{D-1} s_i \cdot \Delta u_{k-i} + s_D \cdot u_{k-D}, \tag{5}$$

where: y_k^M – value of model output at sampling instant k,
s_i – step response coefficients ($i = 1, 2, \ldots, D$),
D – dynamics horizon, which is equal to the number of sampling instants after which the step response is fixed.

Predicted output values can be calculated as follows [13]:

$$y_{k+i|k} = \sum_{n=1}^{i} s_n \cdot \Delta u_{k+i-n} + \sum_{n=i+1}^{D-1} s_n \cdot \Delta u_{k+i-n} + s_D \cdot u_{k-D+i} + d_k \,, \tag{6}$$

where: $d_k = y_k - y_k^M$ describes influence of model uncertainty and of unmeasured disturbances on the control plant. By expanding d_k the equation (6) is transformed to [13]:

$$\begin{aligned} y_{k+i|k} = y_k + \sum_{n=i+1}^{D-1} s_n \cdot \Delta u_{k+i-n} + s_D \cdot \sum_{n=D}^{D+i-1} \Delta u_{k+i-n} - \sum_{n=1}^{D-1} s_n \cdot \Delta u_{k-n} \\ + \sum_{n=1}^{i} s_n \cdot \Delta u_{k+i-n|k} \,. \end{aligned} \tag{7}$$

2.1.2. MIMO case

The formulas (6) and (7) describe process model and predicted output values for SISO processes. For MIMO processes a multivariable step response matrix has to be defined. This matrix denoted as $\boldsymbol{S_l}$ consists of a set of coefficients s_l^{ij} of all the finite step responses at the sampling instant $l, i = 1, 2, \ldots, n_y, j = 1, 2, \ldots, n_u$. By performing a step change of one input with all other inputs kept unchanged the values of outputs can be registered and by repeating this procedure for all the other inputs $j = 1, 2, \ldots, n_u$ the set of step responses can be obtained. Therefore, the matrix $\boldsymbol{S}_l$ consists of all these step responses and can be written as [17]:

$$\boldsymbol{S}_l = \begin{bmatrix} s_l^{11} & s_l^{12} & s_l^{13} & \ldots & s_l^{1n_u} \\ s_l^{21} & s_l^{22} & s_l^{23} & \ldots & s_l^{2n_u} \\ s_l^{31} & s_l^{32} & s_l^{33} & \ldots & s_l^{3n_u} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ s_l^{n_y 1} & s_l^{n_y 2} & s_l^{n_y 3} & \ldots & s_l^{n_y n_u} \end{bmatrix}, \qquad l = 1, 2, \ldots, D. \tag{8}$$

With the use of matrix (8) the formulas (6) and (7) can be rewritten [17]:

$$\boldsymbol{y}_k^{\boldsymbol{M}} = \sum_{i=1}^{D-1} \boldsymbol{S}_i \cdot \boldsymbol{\Delta u}_{k-i} + \boldsymbol{S}_D \cdot \boldsymbol{u}_{k-D} \,, \tag{9}$$

$$\boldsymbol{y}_{k+i|k} = \boldsymbol{y}_k + \sum_{n=i+1}^{D-1} \boldsymbol{S}_n \cdot \Delta\boldsymbol{u}_{k+i-n} + \boldsymbol{S}_D \cdot \sum_{n=D}^{D+i-1} \Delta\boldsymbol{u}_{k+i-n} - \sum_{n=1}^{D-1} \boldsymbol{S}_n \cdot \Delta\boldsymbol{u}_{k-n} + \sum_{n=1}^{i} \boldsymbol{S}_n \cdot \Delta\boldsymbol{u}_{k+i-n|k} \,. \tag{10}$$

The predicted values of outputs can be written in the vector–matrix format [17]:

$$\boldsymbol{Y}(k) = \boldsymbol{Y}^{\boldsymbol{0}}(k) + \boldsymbol{M} \cdot \Delta\boldsymbol{U}(k), \tag{11}$$

where $\boldsymbol{M} \cdot \Delta\boldsymbol{U}(k)$ is the process forced response dependent on future changes of the manipulated variables and $\boldsymbol{Y}^{\boldsymbol{0}}(k)$ is the process free response describing the behavior of the process in response to the previous control changes. Process forced response is described using dynamic matrix $\boldsymbol{M}$ which includes the process step response coefficients. This matrix is of dimension $n_y N \times n_u N_u$ and is as follows [17]:

$$\boldsymbol{M} = \begin{bmatrix} \boldsymbol{S}_1 & \boldsymbol{0} & \dots & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{S}_2 & \boldsymbol{S}_1 & \dots & \boldsymbol{0} & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \boldsymbol{S}_N & \boldsymbol{S}_{N-1} & \dots & \boldsymbol{S}_{N-N_u+2} & \boldsymbol{S}_{N-N_u+1} \end{bmatrix}. \tag{12}$$

Dimension of $\boldsymbol{Y}(k)$, $\boldsymbol{Y}^{\boldsymbol{0}}(k)$ is $n_y \cdot N$ while of $\Delta\boldsymbol{U}(k)$ – is $n_u \cdot (N_u - 1)$ [17]:

$$\boldsymbol{Y}(k) = \begin{bmatrix} \boldsymbol{y}_{k+1|k} \\ \vdots \\ \boldsymbol{y}_{k+N|k} \end{bmatrix}, \qquad \boldsymbol{Y}^{\boldsymbol{0}}(k) = \begin{bmatrix} \boldsymbol{y}^{\boldsymbol{0}}_{k+1|k} \\ \vdots \\ \boldsymbol{y}^{\boldsymbol{0}}_{k+N|k} \end{bmatrix}, \qquad \Delta\boldsymbol{U}(k) = \begin{bmatrix} \Delta\boldsymbol{u}_{k|k} \\ \vdots \\ \Delta\boldsymbol{u}_{k+N_u-1|k} \end{bmatrix},$$

where

$$\boldsymbol{y}^{\boldsymbol{0}}_{k+i|k} = \begin{bmatrix} y^{0\_1}_{k+i|k} \\ y^{0\_2}_{k+i|k} \\ \vdots \\ y^{0\_n_y}_{k+i|k} \end{bmatrix},$$

$y^{0\_j}_{k+i|k}$ is the element of the free response predicted for i-th time instant from the prediction horizon, for j-th output. The process free response is calculated as follows [17]:

$$\boldsymbol{Y}^{\boldsymbol{0}}(k) = \boldsymbol{Y}(k) + \boldsymbol{M}^{\boldsymbol{P}} \cdot \Delta\boldsymbol{U}^{\boldsymbol{P}}(k), \tag{13}$$

where matrix $\boldsymbol{M}^{\boldsymbol{P}}$ is used for calculations of the outputs predictions depending only on past increments of the process control inputs $\Delta\boldsymbol{U}^{\boldsymbol{P}}(k)$; the matrix is of

dimension $n_y \cdot N \times n_u \cdot (D-1)$ and has the following form [17]:

$$\boldsymbol{M^P} = \begin{bmatrix} \boldsymbol{S}_2 - \boldsymbol{S}_1 & \boldsymbol{S}_3 - \boldsymbol{S}_2 & \dots & \boldsymbol{S}_{D-1} - \boldsymbol{S}_{D-2} & \boldsymbol{S}_D - \boldsymbol{S}_{D-1} \\ \boldsymbol{S}_3 - \boldsymbol{S}_2 & \boldsymbol{S}_4 - \boldsymbol{S}_2 & \dots & \boldsymbol{S}_D - \boldsymbol{S}_{D-2} & \boldsymbol{S}_D - \boldsymbol{S}_{D-1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \boldsymbol{S}_{N+1} - \boldsymbol{S}_1 & \boldsymbol{S}_{N+2} - \boldsymbol{S}_2 & \dots & \boldsymbol{S}_D - \boldsymbol{S}_{D-2} & \boldsymbol{S}_D - \boldsymbol{S}_{D-1} \end{bmatrix}. \tag{14}$$

Dimension of $\boldsymbol{\Delta U^P}(k)$ is $n_u \cdot (D-1)$ [13]:

$$\boldsymbol{\Delta U^P}(k) = \begin{bmatrix} \boldsymbol{\Delta u}_{k-1} \\ \vdots \\ \boldsymbol{\Delta u}_{k-D+1} \end{bmatrix},$$

where:

$$\boldsymbol{\Delta u}_{k-p} = \begin{bmatrix} \Delta u^1_{k-p} \\ \Delta u^2_{k-p} \\ \vdots \\ \Delta u^{n_u}_{k-p} \end{bmatrix},$$

Δu^j_{k-p} is the increment of the j-th control in the past $(k-p)$-th time instant. Analytic solution to the optimization problem for DMC controller is as follows [17]:

$$\boldsymbol{\Delta U}(k) = \left(\boldsymbol{M}^T \boldsymbol{\Psi} \boldsymbol{M} + \boldsymbol{\Lambda}\right)^{-1} \boldsymbol{M}^T \boldsymbol{\Psi} \left(\boldsymbol{Y}^{\mathbf{ref}}(k) - \boldsymbol{Y}^{\mathbf{0}}(k)\right), \tag{15}$$

where $\boldsymbol{\Lambda}$ is the matrix of weighting parameters of dimension $N_u \cdot n_u \times N_u \cdot n_u$ weighting input increments and $\boldsymbol{\Psi}$ is the matrix of weighting parameters of dimension $N \cdot n_y \times N \cdot n_y$ weighting predicted control errors. Matrix $\boldsymbol{\Psi}$ can be written as follows:

$$\boldsymbol{\Psi} = \begin{bmatrix} \boldsymbol{\Psi}_1 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Psi}_2 & \mathbf{0} & \vdots & \vdots & \vdots \\ \vdots & \mathbf{0} & \ddots & \mathbf{0} & \vdots & \vdots \\ \vdots & \vdots & \mathbf{0} & \ddots & \mathbf{0} & \vdots \\ \vdots & \vdots & \vdots & \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{\Psi}_N \end{bmatrix}, \tag{16}$$

where $\boldsymbol{\Psi}_i$ for $i = 1, 2, \ldots, N$ are defined in (3). Similarly matrix $\boldsymbol{\Lambda}$ [17]:

$$\boldsymbol{\Lambda} = \begin{bmatrix} \boldsymbol{\Lambda}_0 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Lambda}_1 & \mathbf{0} & \vdots & \vdots & \vdots \\ \vdots & \mathbf{0} & \ddots & \mathbf{0} & \vdots & \vdots \\ \vdots & \vdots & \mathbf{0} & \ddots & \mathbf{0} & \vdots \\ \vdots & \vdots & \vdots & \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{\Lambda}_{N_u-1} \end{bmatrix}, \tag{17}$$

where $\boldsymbol{\Lambda}_i$ for $i = 0, 1, \ldots, N_u - 1$ are defined in (2). $\boldsymbol{Y}^{\mathbf{ref}}(k)$ is the vector containing reference trajectory, of the same dimension as $\boldsymbol{Y}^{\mathbf{0}}(k)$ [17]:

$$\boldsymbol{Y}^{\mathbf{ref}}(k) = \begin{bmatrix} \boldsymbol{y}^{\mathbf{ref}}_{k+1|k} \\ \vdots \\ \boldsymbol{y}^{\mathbf{ref}}_{k+N|k} \end{bmatrix}.$$

The reference trajectory is a user-specified trajectory based on a set point trajectory. It can be chosen in different ways in order to influence the control system performance.

When the analytic formula (15) is used, the constraints on increments and on values of manipulated variables can be taken into consideration by projecting the increments of manipulated variables onto the admissible set [17]. Therefore only the control increments from the first time instant of the control horizon can be modified. Thus, the following constraints can be taken into consideration:

$$\boldsymbol{U}_{\mathbf{min}} \leqslant \boldsymbol{U}(k|k) \leqslant \boldsymbol{U}_{\mathbf{max}}, \tag{18}$$

$$\boldsymbol{\Delta U}_{\mathbf{min}} \leqslant \boldsymbol{\Delta U}(k|k) \leqslant \boldsymbol{\Delta U}_{\mathbf{max}}, \tag{19}$$

where

$$\boldsymbol{U}(k|k) = \begin{bmatrix} u^1_{k|k} \\ u^2_{k|k} \\ \vdots \\ u^{n_u}_{k|k} \end{bmatrix}, \qquad \boldsymbol{\Delta U}(k|k) = \begin{bmatrix} \Delta u^1_{k|k} \\ \Delta u^2_{k|k} \\ \vdots \\ \Delta u^{n_u}_{k|k} \end{bmatrix},$$

$\boldsymbol{U}_{\mathbf{min}}$, $\boldsymbol{U}_{\mathbf{max}}$, $\boldsymbol{\Delta U}_{\mathbf{min}}$, $\boldsymbol{\Delta U}_{\mathbf{max}}$ are vectors of length n_u each, containing constraints on values and increments of process inputs, respectively. Such an approach often leads to results close to optimal. However, it is much more difficult to take into account constraints on the process outputs in this way. To include these

constraints and the control constraints on the whole control horizon it is easier to use a numeric solution of the following quadratic optimization problem [7]:

$$\min_{\Delta \boldsymbol{U}(k)} \left\{ \left(\boldsymbol{Y}^{\mathbf{ref}}(k) - \boldsymbol{Y}^{\mathbf{0}}(k) - \boldsymbol{M} \cdot \Delta \boldsymbol{U}(k) \right)^T \boldsymbol{\Psi} \left(\boldsymbol{Y}^{\mathbf{ref}}(k) - \boldsymbol{Y}^{\mathbf{0}}(k) - \boldsymbol{M} \cdot \Delta \boldsymbol{U}(k) \right) + \boldsymbol{\Lambda} \cdot \Delta \boldsymbol{U}^T(k) \cdot \Delta \boldsymbol{U}(k) \right\} \tag{20}$$

subject to [7]:

$$\Delta \boldsymbol{U}_{\mathbf{min}} \leqslant \Delta \boldsymbol{U}(k) \leqslant \Delta \boldsymbol{U}_{\mathbf{max}}, \tag{21}$$

$$\boldsymbol{U}_{\mathbf{min}} \leqslant \boldsymbol{U}(k-1) + \boldsymbol{J} \cdot \Delta \boldsymbol{U}(k) \leqslant \boldsymbol{U}_{\mathbf{max}}, \tag{22}$$

$$\boldsymbol{Y}_{\mathbf{min}} \leqslant \boldsymbol{Y}^{\mathbf{0}}(k) + \boldsymbol{M} \cdot \Delta \boldsymbol{U}(k) \leqslant \boldsymbol{Y}_{\mathbf{max}}, \tag{23}$$

where $\boldsymbol{U}_{\mathbf{min}}$, $\boldsymbol{U}_{\mathbf{max}}$, $\boldsymbol{U}(k-1)$, $\Delta \boldsymbol{U}_{\mathbf{min}}$, $\Delta \boldsymbol{U}_{\mathbf{max}}$ are of dimension $n_u \cdot N_u$ while $\boldsymbol{Y}_{\mathbf{min}}$, $\boldsymbol{Y}_{\mathbf{max}}$ are of dimension $n_y \cdot (N - N_1 + 1)$. Matrix $\boldsymbol{J}$ is of dimension $n_u \cdot N_u \times n_u \cdot N_u$ and is defined as follows [7]:

$$\boldsymbol{J} = \begin{bmatrix} \boldsymbol{I}_u & \boldsymbol{0} & \dots & \boldsymbol{0} \\ \boldsymbol{I}_u & \boldsymbol{I}_u & \dots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{I}_u & \boldsymbol{I}_u & \boldsymbol{I}_u & \boldsymbol{I}_u \end{bmatrix}, \tag{24}$$

where every identity matrix $\boldsymbol{I}_u$ is of dimension $n_u \times n_u$. A numerical DMC controller solves the presented quadratic optimization problem at each iteration. As a result the vector of future increments of the control inputs is obtained. In the analytic version it is sufficient to apply the formula (15). However, the numeric version solves the optimization problem taking into account input and output constraints on the whole prediction horizon and gives the optimal results. In every iteration of the algorithm in either version the first n_u elements of the obtained solution vector $\Delta \boldsymbol{U}(k)$ are applied to the process. In the next iteration of the algorithm the procedure repeats and the updated solution vector $\Delta \boldsymbol{U}(k)$ is obtained.

2.2. NDMC–SL predictive control algorithm

The nonlinear DMC algorithm with successive linearization (NDMC–SL) has a similar structure to DMC, but with one modification. In every iteration of the algorithm a linear model is extracted from nonlinear process model and a new set of step responses is calculated from the new model. This model is extracted by using current values of process inputs and outputs. As a result new dynamic

matrix $\boldsymbol{M}$ and matrix $\boldsymbol{M}^{\boldsymbol{P}}$ are obtained and are later used to obtain analytic (15) or numeric (20)–(23) solution of the optimization problem.

Algorithm steps for the current sampling instant k are as follows [17]:

1. Initialization.
2. Linearization of nonlinear model.
3. Obtaining dynamic matrices $\boldsymbol{M}(k)$ and $\boldsymbol{M}^{\boldsymbol{P}}(k)$:

$$\boldsymbol{M}(k) = \begin{bmatrix} \boldsymbol{S}_{N_1}(k) & \boldsymbol{S}_{N_1-1}(k) & \dots & \boldsymbol{S}_1(k) & \boldsymbol{0} & \dots & \boldsymbol{0} \\ \boldsymbol{S}_{N_1+1}(k) & \boldsymbol{S}_{N_1}(k) & \dots & \boldsymbol{S}_2(k) & \boldsymbol{S}_1(k) & \dots & \boldsymbol{0} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{S}_{N_u}(k) & \boldsymbol{S}_{N_u-1}(k) & \dots & \boldsymbol{S}_{N_u-N_1+1}(k) & \boldsymbol{S}_{N_u-N_1}(k) & \dots & \boldsymbol{S}_1(k) \\ \boldsymbol{S}_{N_u+1}(k) & \boldsymbol{S}_{N_u}(k) & \dots & \boldsymbol{S}_{N_u-N_1+2}(k) & \boldsymbol{S}_{N_u-N_1+1}(k) & \dots & \boldsymbol{S}_2(k) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{S}_N(k) & \boldsymbol{S}_{N-1}(k) & \dots & \boldsymbol{S}_{N-N_1+1}(k) & \boldsymbol{S}_{N-N_1}(k) & \dots & \boldsymbol{S}_{N-N_u+1}(k) \end{bmatrix}, \tag{25}$$

$$\boldsymbol{M}^{\boldsymbol{P}}(k) = \begin{bmatrix} \boldsymbol{S}_{1+N_1}(k)-\boldsymbol{S}_1(k) & \boldsymbol{S}_{2+N_1}(k)-\boldsymbol{S}_2(k) & \dots & \boldsymbol{S}_{D-1+N_1}(k)-\boldsymbol{S}_{D-1}(k) \\ \boldsymbol{S}_{2+N_1}(k)-\boldsymbol{S}_1(k) & \boldsymbol{S}_{3+N_1}(k)-\boldsymbol{S}_2(k) & \dots & \boldsymbol{S}_{D+N_1}(k)-\boldsymbol{S}_{D-1}(k) \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{S}_{N+1}-\boldsymbol{S}_1(k) & \boldsymbol{S}_{N+2}(k)-\boldsymbol{S}_2(k) & \dots & \boldsymbol{S}_{N+D-1}(k)-\boldsymbol{S}_{D-1}(k) \end{bmatrix}. \tag{26}$$

4. Calculating free response (13).
5. Defining the optimization problem and solving it using (15) in the unconstrained case or solving (20)–(23) in the constrained case.
6. Calculating new process inputs.

In the next sampling instant the algorithm goes back to step 2 repeating linearization of the nonlinear model for current values of variables, then it follows steps 3–6.

2.3. NDMC–NPL predictive control algorithm

In the nonlinear DMC algorithm with nonlinear prediction and linearization (NDMC–NPL), just like in NDMC–SL algorithm, in every iteration a linearization of the nonlinear model is conducted. However the difference is in calculating process free response. Here every element of the free response is calculated using the nonlinear model and not the linearized one like in algorithm NDMC–SL.

Algorithm steps for the current sampling instant k are as follows [17]:

1. Initialization.
2. Linearization of the nonlinear model.
3. Designing the matrices $\boldsymbol{M}(k)$ and $\boldsymbol{M}^{\boldsymbol{P}}(k)$ with the use of linearized model as in (25) and (26).

4. Calculating free response iteratively with the use of nonlinear model as follows:

$$\begin{aligned}
\boldsymbol{y}^{\boldsymbol{0}}_{k+1|k} &= g\left(\boldsymbol{y}_k, \boldsymbol{y}_{k-1}, \ldots, \boldsymbol{y}_{k-n_A}, \boldsymbol{u}_{k-\tau}, \ldots, \boldsymbol{u}_{k-n_B}\right) + \boldsymbol{d}_k\,, \\
\boldsymbol{y}^{\boldsymbol{0}}_{k+2|k} &= g\left(\boldsymbol{y}^{\boldsymbol{0}}_{k+1|k}, \boldsymbol{y}_k, \ldots, \boldsymbol{y}_{k-n_A+1}, \boldsymbol{u}_{k-\tau+1}, \ldots, \boldsymbol{u}_{k-n_B+1}\right) + \boldsymbol{d}_k\,, \\
&\vdots \\
\boldsymbol{y}^{\boldsymbol{0}}_{k+N|k} &= g\Big(\boldsymbol{y}^{\boldsymbol{0}}_{k+N-1|k}, \boldsymbol{y}^{\boldsymbol{0}}_{k+N-2|k}, \ldots, \boldsymbol{y}_{k-n_A+N-1}, \boldsymbol{u}_{k-\tau+N-1}, \\
&\qquad \ldots, \boldsymbol{u}_{k-n_B+N-1}\Big) + \boldsymbol{d}_k\,,
\end{aligned} \tag{27}$$

where $\boldsymbol{d}_k = \boldsymbol{y}_k - g(\boldsymbol{y}_{k-1}, \ldots, \boldsymbol{y}_{k-n_A-1}, \boldsymbol{u}_{k-\tau-1}, \ldots, \boldsymbol{u}_{k-n_B-1})$.

5. Defining the optimization problem and solving it using (15) in the unconstrained case or solving (20)–(23) in the constrained case.

6. Calculating new process inputs.

In step 4 a nonlinear process model and the previous values of process outputs and inputs are used to calculate the free response. During the calculations future output values are replaced with the previously calculated elements of the free response. In the next sampling instant the algorithm goes back to step 2 with current values of variables and then follows the next steps.

3. Preliminary experiments

3.1. Control plant description

First experiments were conducted in the control system of a linear plant described by the following transfer function [8]:

$$G(s) = \frac{-s+1}{4s^2+2s+1} e^{-4s}\,. \tag{28}$$

By using the Z-transform and assuming sample time $T_p = 1$ [s] the model can be presented as the following discrete transfer function:

$$G(z) = \frac{-0.08427z + 0.277}{z^2 - 1.414z + 0.6065} z^{-4}\,. \tag{29}$$

In order to implement a DMC controller a step response of (29) was obtained; it is shown in Fig. 1.

As seen in Fig. 1 the plant has dynamics difficult to control. It has a delay $\tau = 4$ [s], an inverse–response starting at $t = 5$ [s] and later an oscillating

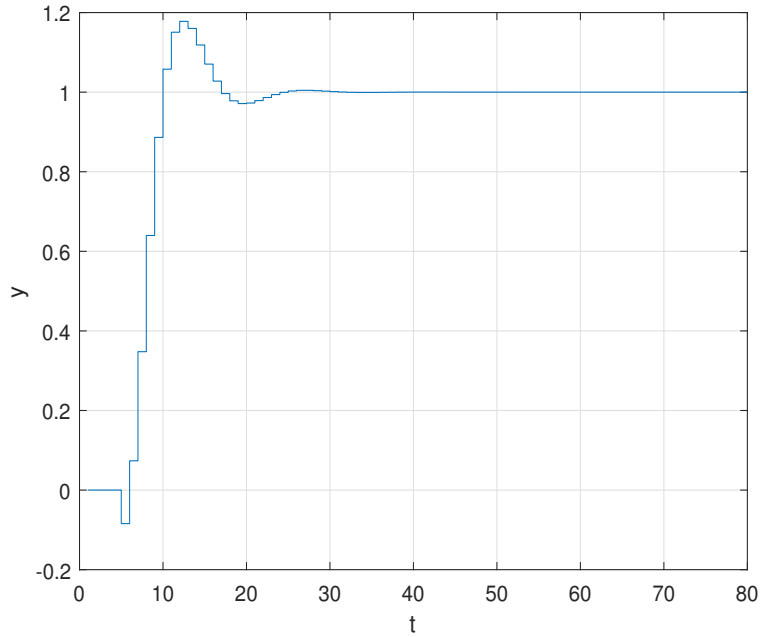

Figure 1: Step response of a linear model

response. A linear second order discrete differential equation describing the plant is as follows:

$$y(k) = -0.0843u(k-5)+0.277u(k-6)+1.4138y(k-1)-0.6065y(k-2). \quad (30)$$

3.2. Influencing control performance with different shapes of the trajectory of weighting parameters ψ_i

For the considered control plant a DMC controller in the analytic version was designed. The following controller parameters were chosen: $N = 20$, $N_u = 10$, $\lambda = 2$; the simulation method was used to tune the controller. The parameters were chosen in such a way that the steady-state is reached by the process in the shortest time. These values of parameters remain unchanged for all experiments in this subsection. The first trajectory under consideration is:

$$\psi_i = \frac{1}{N} i, \quad (31)$$

where the biggest value of the weighting parameter is $\psi_N = 1$. This trajectory reduced the overshoot but increased the rise time of the controlled system output. Similar results have been obtained with the next trajectories:

$$\psi_i = \frac{1}{N^2} i^2, \quad (32)$$

$$\psi_i = \frac{N}{N+i-1}, \tag{33}$$

$$\psi_i = -\frac{1}{N}i + 1, \tag{34}$$

where similarly to trajectory (31) the biggest value of the weighting parameter is 1. However, these trajectories have different shapes i.e. trajectory (32) rises asymptotically while the descending trajectories (33) and (34) have different descending ratios. Trajectory (32) reduced the overshoot even more but also increased the rise time further in comparison to trajectory (31). Trajectory (33) has decreased slightly the overshoot without any change in the rise time in comparison when using a trajectory of constant weighting parameters. Trajectory (34) has slightly more reduced the overshoot but also slightly increased the rise time in comparison with trajectory (33). These experiments show that the shape of the trajectory has influence on the overall system performance. The first experiments have shown that by choosing the appropriate shape of trajectory of the weighting parameters ψ_i it is possible to reduce the overshoot comparing to the case when the weighting parameters ψ_i are the same on the entire prediction horizon.

Trajectories tested so far were able to reduce the overshoot, but not the rise time. In the next experiments other trajectories are inspected in order to find the one that can effectively reduce the rise time as well. The first one is as follows:

$$\psi_i = \begin{cases} 1 & \text{for } i = k, \\ 0 & \text{for } i \neq k, \end{cases} \tag{35}$$

where all weighting parameter except one have values equal to 0; k is the tuning parameter of this trajectory. For different values of k (where $k = 1, 2, \ldots, N$) different system performance was observed. The trajectory (35) reduced the overshoot greatly but also increased the rise time greatly. However, for different values of the parameter k, different values of rise time were obtained; for $k = 9$ the least one. The next trajectory is the opposite to the trajectory (35):

$$\psi_i = \begin{cases} 0 & \text{for } i = k, \\ 1 & \text{for } i \neq k, \end{cases} \tag{36}$$

where this time all weighting parameters except one have values equal to 1. In this case the rise time remained unchanged, but overshoot had either decreased or increased, depending on value of k. For $k = 8$ the overshoot was reduced. An additional experiment was conducted where weighting parameters for $i = \{1, \ldots, 5\}$ had values 0 and the rest – 1. The purpose of this test was to eliminate the influence of the delay and of the inverse–response and, as a result,

to improve the system performance. However, it increased the overshoot while the rise time was the same as when using the constant trajectory. The next tested trajectory was as follows:

$$\psi_i = \begin{cases} K & \text{for } i = k, \\ 1 & \text{for } i \neq k, \end{cases} \qquad K > 1, \tag{37}$$

where one of the weighting parameters had a higher value than the rest. Several experiments have been conducted for different values of tuning parameters k and K. For $k = 8$ and $K = 150$ the best results were obtained; in this case both: rise time and overshoot were reduced. The value of parameter k is exactly the same as for trajectory (36) when the best control performance was obtained. It is also close to parameter $k = 9$ for trajectory (35) where also the best results were achieved. In the next experiments other trajectories were tested to further reduce the rise time by changing the shape of trajectory (37). The first one is in the shape of the Gauss function:

$$\psi_i = K \cdot e^{-\left(\frac{i-k}{a}\right)^2}, \tag{38}$$

where a is a parameter defining the shape of the trajectory. This parameter influences the width of the trajectory. Depending on the value of parameter a the overshoot can be increased or reduced. When parameter a is increased, as a result the width of the trajectory is increased, and the overshoot is reduced. However, the rise time can be increased as well if the width of the trajectory is increased too much. If parameter a is reduced then the opposite happens. The rise time is reduced, but the overshoot is increased. However, if the width of the trajectory is reduced too much the rise time can also get increased. Nevertheless, trajectory (38) reduced the rise time effectively in comparison to trajectory (37). For $a = 4$ the best results were obtained. Also other functions, like bell-shaped, triangular and trapezoidal ones were tested:

$$\psi_i = K \cdot \frac{1}{1 + \left(\dfrac{i-k}{b}\right)^{2a}}, \tag{39}$$

$$\psi_i = \begin{cases} 0 & \text{for } i \leqslant a, \\ K \cdot \dfrac{i-a}{k-a} & \text{for } a \leqslant i \leqslant k, \\ K \cdot \dfrac{c-i}{c-k} & \text{for } k \leqslant i \leqslant c, \\ 0 & \text{for } \geqslant c; \end{cases} \tag{40}$$

$$\psi_i = \begin{cases} 0 & \text{for } i \leqslant a, \\ K \cdot \dfrac{i-a}{b-a} & \text{for } a \leqslant i \leqslant b, \\ K & \text{for } b \leqslant i \leqslant c, \\ K \cdot \dfrac{d-i}{d-c} & \text{for } c \leqslant i \leqslant d, \\ 0 & \text{for } i \geqslant d, \end{cases} \tag{41}$$

where a, b, c and d are tuning parameters defining shapes of trajectories. Experiments with changing the shapes of trajectories (39), (40) and (41) were started by making their width close to the width of the trajectory (38) which gave the best control system performance. Next, all the parameters were increased and reduced to check their influence on the obtained results. Similarly to trajectory (38) if the width is increased overshoot is reduced and if it is increased too much the rise time is also increased. The opposite like in trajectory (38) also happens i.e. when width is reduced then the rise time is reduced and the overshoot increased. If it is reduced too much then the rise time can get increased as well. Nevertheless, for different shapes of trajectories, even if the width is similar, different control system performance is obtained. Therefore, it is important to perform these experiments to achieve the best control system performance.

Table 1 and Fig. 2 summarize the best and the most interesting results from conducted experiments. The best control system performance was obtained with the bell-shaped trajectory with the parameters $a = 4$ and $b = 3.5$. As seen also in Fig. 2 the overshoot and rise time were significantly reduced comparing to the case with the constant trajectory. However, due to using the proposed trajectory the influence of inverse–response was increased. Moreover, manipulated variable has higher values at the beginning of the experiment in order to guarantee this system performance. This section introduced the use of differently shaped trajectories of weighting parameters ψ_i in a control system of a linear plant with difficult dynamics. The next sections explore the influence of these trajectories on the control systems of nonlinear plants.

Table 1: Comparison of control system performance with different trajectories of weighting parameters ψ_i

Trajectory number	y_o [%]	t_r [s]	Parameters
– (constant)	3.3405	4	–
32	0	6	–
37	2.6268	2	$k = 8, K = 100$
39	2.0372	2	$k = 8, K = 100, a = 4, b = 3.5$

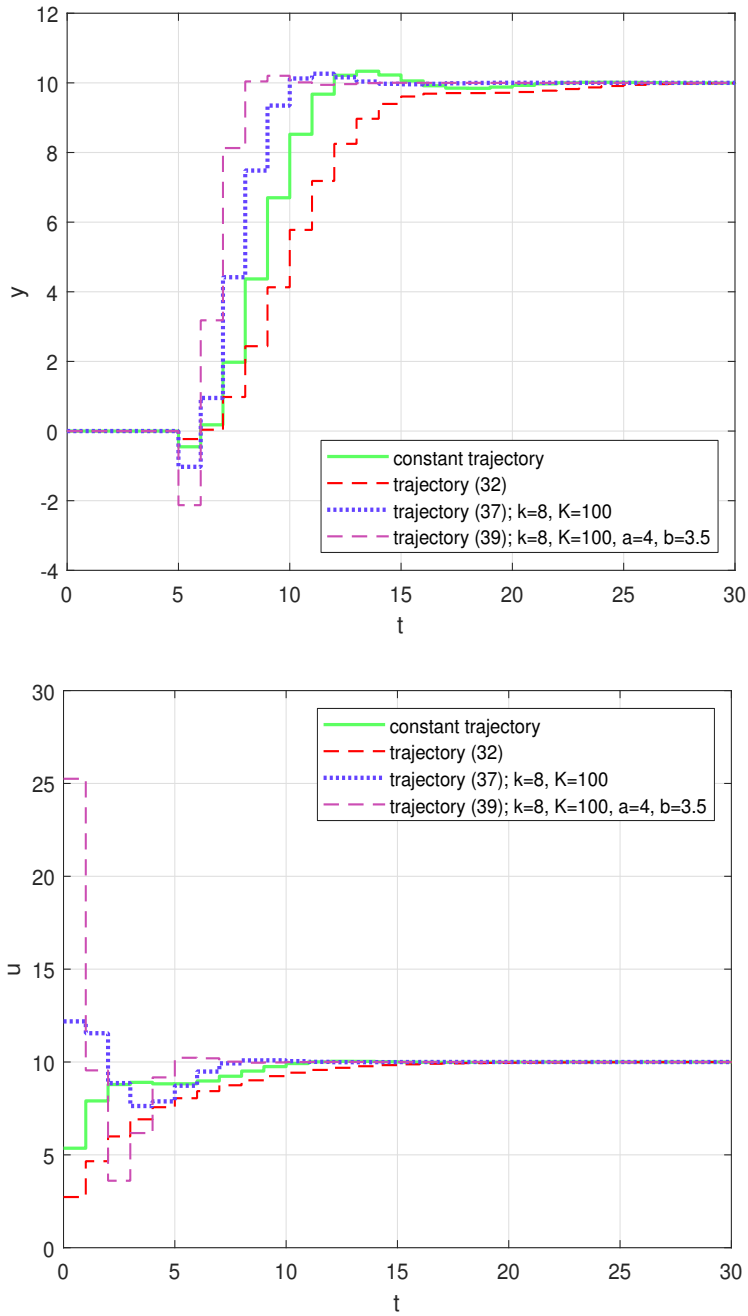

Figure 2: Responses of the control system with DMC controller to the change of the set point value to $y^{set} = 10$ with different trajectories of weighting parameters ψ_i; output – top, input – bottom

4. Experiments in the control system of a chemical reactor

4.1. Control plant description

The second control plant is a polymerization reactor in which a chemical reaction of combining monomer molecules together in order to produce large polymer chains or three-dimensional networks is carried out. This polymerization reactor is a SISO nonlinear plant that is described by the following differential equations model [5]:

$$\dot{x}_1 = 10(6 - x_1) - 2.4568x_1\sqrt{x_2}, \tag{42}$$

$$\dot{x}_2 = 80u - 10.1022x_2, \tag{43}$$

$$\dot{x}_3 = 0.0024121x_1\sqrt{x_2} + 0.112191x_2 - 10x_3, \tag{44}$$

$$\dot{x}_4 = 245.978x_1\sqrt{x_2} - 10x_4, \tag{45}$$

where the output is calculated as:

$$y = \frac{x_4}{x_3}. \tag{46}$$

The input is the flowrate of initiator denoted as u and the output is the number-average molecular weight (NAMW) of the product denoted as y. The operating conditions are as follows: $x_{10} = 5.50677$ kmol/m^3, $x_{20} = 0.132906$ kmol/m^3,

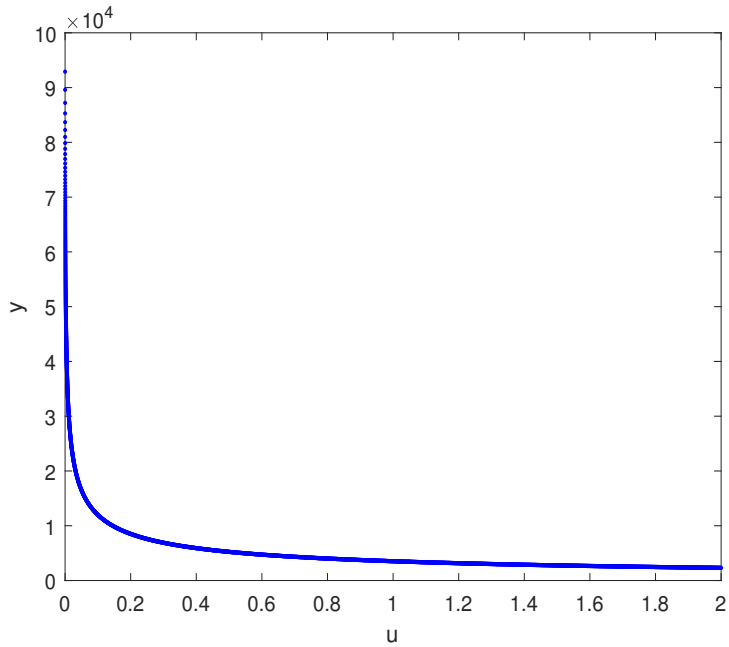

Figure 3: Steady-state characteristic

$x_{30} = 0.0019752$ kmol/m^3, $x_{40} = 49.3818$ kmol/m^3, $u_0 = 0.016783$ m^3/h, $y_0 = 25000.5$. Steady-state characteristic of the plant is presented in Fig. 3. This characteristic is highly nonlinear for $u \leqslant 0.2$.

4.2. Influencing control system performance with different shapes of the trajectory of weighting parameters ψ_i

Three MPC algorithms were designed for the considered control plant. These are: DMC, NDMC–SL and NDMC–NPL. DMC and NDMC–SL algorithms were implemented in analytic version while NDMC–NPL algorithm – in numeric version. All three algorithms were thoroughly tested with the same set of parameters $N = 50$, $N_u = 10$ and $\lambda = 2e12$ chosen in such a way that the best system performance was obtained. For these parameters the steady-state is reached by the process in the shortest time. A linear model was extracted from the nonlinear process model (42)–(46) for the DMC algorithm. The process model was then implemented as discrete differential equations with an assumed sample time $T_p = 0.01$ [h]. Input constraint was also taken into consideration in the algorithms – negative values of manipulated variable are prohibited.

Comparison of the responses obtained with all three algorithms is presented in Fig. 4. Test simulations show that the control system with NDMC–NPL algorithm offers the best control quality in comparison to the other algorithms – the settling time and the overshoot are the smallest. Control system with the NDMC–SL algorithm operates better than the one with the DMC algorithm – generates smaller settling time. However in both control systems oscillating responses were obtained. Moreover, due to nonlinearity of the plant the response has a different character depending on the value of set point. If the set point has a value higher than the initial output value the output trajectory has an oscillating character with slow reduction of oscillations, but if the initial output value has a higher value than the set point the output trajectory has barely visible oscillations. This difference can influence the control performance during the experiments. The control system with the NDMC–NPL algorithm has no oscillating character of the response and has shown the best control quality. Therefore it was used in next experiments with different shapes of weighting parameters trajectories. The numerical results of all experiments have been collected in Table 2; superscripts of overshoot and of the rise time denote what was the value of the set point, namely 1 denotes the set point equal to 29500.5, and 2 – 20500.5.

First experiments were done with trajectories (31)–(34). These trajectories caused reduction of overshoot y_o^1 in comparison to using the constant trajectory of weighting parameters. However, the rise times t_r^1, t_r^2 and the overshoot y_o^2 have increased. This difference in the influence of trajectories on the overshoot is caused by the nonlinear characteristic of the plant, as pointed out in the previous subsection. The example shows that the plant is hard to control and that finding the best shape of the trajectory of weighting parameters is not an easy task. Next

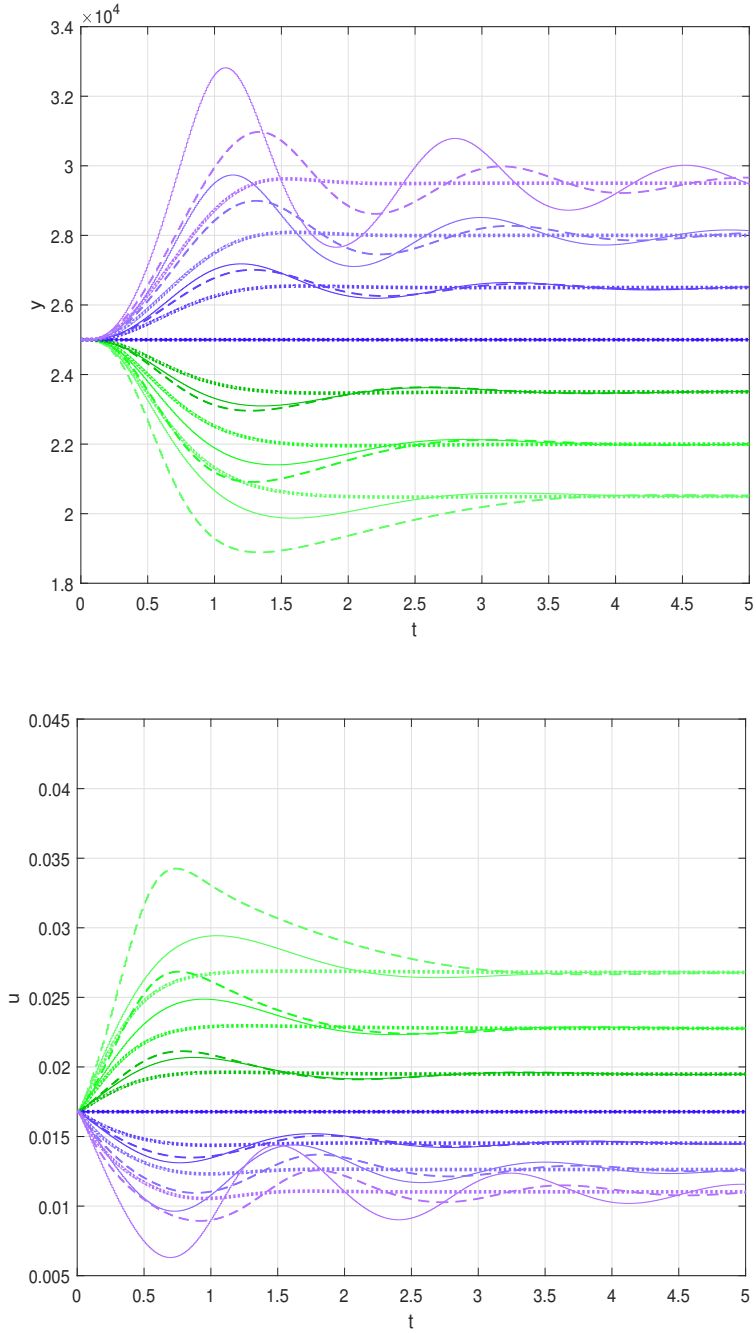

Figure 4: Responses of the control systems of the polymerization reactor to the set point changes with controllers: DMC – solid line, NDMC–SL – dashed line, NDMC–NPL – dotted line; output – top, input – bottom

Table 2: Comparison of control system performance with different trajectories of weighting parameters ψ_i

Trajectory number	y_o^1 [%]	y_o^2 [%]	t_r^1 [h]	t_r^2 [h]	Parameters
– (constant)	0.4203	0.1141	0.78	0.89	–
31	0.2329	0.2012	0.9700	1.5300	–
32	0.12	0.6474	1.1600	2.3800	–
33	0.3116	0.2826	1.0300	1.6200	–
34	0.1147	2.0566	1.6500	4.3400	–
37	0.4764	1.4361	0.7700	0.6600	$K = 100,\ k = 5$
37	0.7615	2.9039	0.4800	0.3900	$K = 100,\ k = 25$
37	0.0658	1.2179	0.5100	0.4300	$K = 100,\ k = 40$
37	0.7210	2.1789	0.5500	0.5000	$K = 50,\ k = 25$
37	0.6493	1.5461	0.6100	0.5800	$K = 30,\ k = 25$
37	0.5129	0.6019	0.7200	0.7500	$K = 10,\ k = 25$
37	0.7211	3.0340	0.4400	0.3500	$K = 150,\ k = 25$
37	0.6567	2.9248	0.4100	0.3200	$K = 200,\ k = 25$
38	0.0623	0.0869	0.3700	0.3000	$K = 150,\ k = 25,\ a = 10$
38	0.1279	0.3024	0.3500	0.2800	$K = 150,\ k = 25,\ a = 5$
38	0.4296	1.7296	0.3500	0.2600	$K = 150,\ k = 25,\ a = 2$
38	1.4833	0.0766	0.2200	0.3100	$K = 150,\ k = 25,\ a = 12$
39	0.3747	1.4705	0.3400	0.2700	$K = 150,\ k = 25,\ a = 10,\ b = 2$
39	0.1233	0.2242	0.3400	0.2800	$K = 150,\ k = 25,\ a = 10,\ b = 5$
39	1.0052	0.0740	0.2300	0.3000	$K = 150,\ k = 25,\ a = 10,\ b = 10$
40	1.8616	0.0715	0.2100	0.3100	$K = 150,\ k = 25,\ a = 0,\ c = 50$
40	0.0696	0.1063	0.3700	0.3000	$K = 150,\ k = 25,\ a = 10,\ c = 40$
40	0.2719	1.0263	0.3400	0.2600	$K = 150,\ k = 25,\ a = 20,\ c = 30$
41	0.0662	0.0893	0.3700	0.3000	$K = 150,\ a = 10,\ b = 23,\ c = 27,\ d = 40$
41	0.0952	0.1518	0.3600	0.2800	$K = 150,\ a = 15,\ b = 23,\ c = 27,\ d = 35$
41	0.1843	0.5210	0.3400	0.2700	$K = 150,\ a = 20,\ b = 23,\ c = 27,\ d = 30$

experiments were done by using trajectory (37) that generated good results in the previous section. Firstly optimal value of parameter k should be found with an assumed value of parameter K. For the linear plant the greater the value of parameter K the more visible rise time reduction was obtained; therefore $K = 100$

was assumed initially. For $k = 5$ rise times t_r^1, t_r^2 have been slightly reduced, but the overshoots y_o^1, y_o^2 have increased. Increasing parameter k has caused further reduction of the rise times and increase in the overshoots. Above $k = 25$ the rise times could not be reduced more. When increasing k above this limit the rise times increase and the overshoots become less visible. Parameter $k = 25$ was chosen for further experiments, because of the maximum reduction of the rise times. After choosing the value of parameter k experiments with different values of the parameter K were done. For values of the parameter lower than 100 the rise times t_r^1, t_r^2 and the overshoot y_o^1 have increased, but overshoot y_o^2 has been reduced. For values of the parameter K greater than 100, the rise times t_r^1, t_r^2 and the overshoots y_o^1, y_o^2 have been reduced. However for $K = 150$ overshoot y_o^2 has increased slightly in comparison to $K = 100$, but for $K = 200$ it was reduced. Nevertheless, $K = 150$ has been chosen as the optimal value of the parameter, because of two reasons. The first was the small difference in control quality between $K = 150$ and $K = 200$. The second one was the significant increase of the manipulated variable at the beginning of the experiment with the increase of parameter K. Afterwards, $K = 150$ and $k = 25$ were used to implement other shapes of trajectories.

For trajectory (38) the rise times and the overshoots were further reduced. Experiments were conducted by changing the shape parameter a. If the width of the trajectory was increased then overshoots y_o^1, y_o^2 were reduced, but rise times t_r^1, t_r^2 increased. Fig. 5 shows an example of control system responses obtained by changing the shape of trajectory (38). For $a = 10$ the best results were achieved therefore this value of the parameter was used as a reference for designing trajectories (39)–(41). If parameter b in the bell–shaped trajectory got increased then rise time t_r^1 and overshoot y_o^2 got reduced, but unfortunately rise time t_r^2 increased. Overshoot y_o^1 got reduced for $b < 10$ and got increased for $b \geqslant 10$. For trajectory (40) when the width of the trajectory was reduced then the rise time t_r^2 was also reduced. However, overshoots y_o^1, y_o^1 and the rise time t_r^1 got either reduced or increased depending on the values of parameters. In the case of trajectory (41) when width of the trajectory was reduced then rise times t_r^1, t_r^2 got reduced. The overshoots y_o^1, y_o^2 had either got increased or reduced. Many experiments have been conducted and the most interesting responses are presented in Fig. 6.

The least overshoot y_o^1 was obtained for trajectory (38) with parameters $K = 150$, $k = 25$ and $a = 10$. The lowest overshoot y_o^2 and the lowest rise time t_r^1 were obtained for trajectory (40) with parameters $K = 150$, $k = 25$, $a = 0$ and $c = 50$. The lowest rise time t_r^1 was obtained also for trajectory (40), but with parameters $K = 150$, $k = 25$, $a = 20$ and $c = 30$. The results show that the best results are achieved when using Gaussian, bell–shaped, triangular or trapezoidal–shaped trajectories. This was also true for the control system of the linear plant, discussed in the previous section. Trajectory (39) with parameters

$K = 150$, $k = 25$, $a = 10$ and $b = 5$ was chosen as the optimal one, because of the most reduced values of the rise times t_r^1, t_r^2 and of the overshoots y_o^1, y_o^2, in comparison to other results.

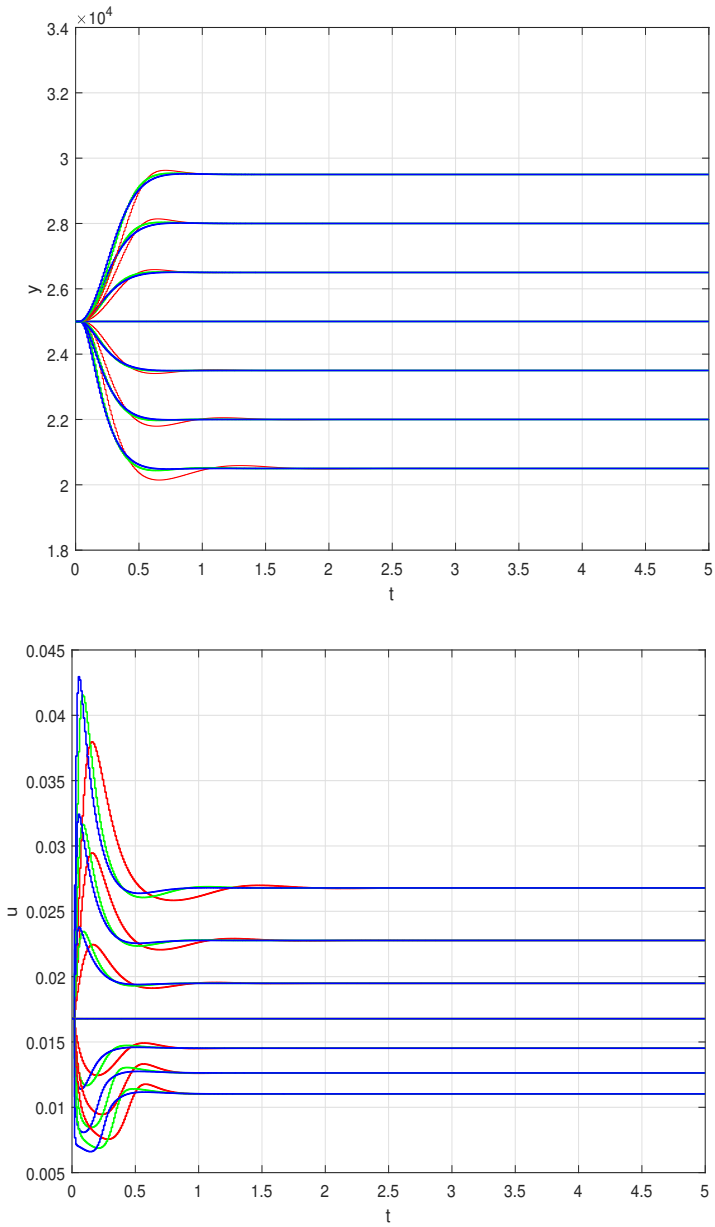

Figure 5: Responses of the control system with NDMC–NPL algorithm for different set point values and different shape of trajectory (38) with parameters $K = 150$, $k = 25$ and: red – $a = 2$, green – $a = 5$, blue – $a = 10$; output – top, input – bottom

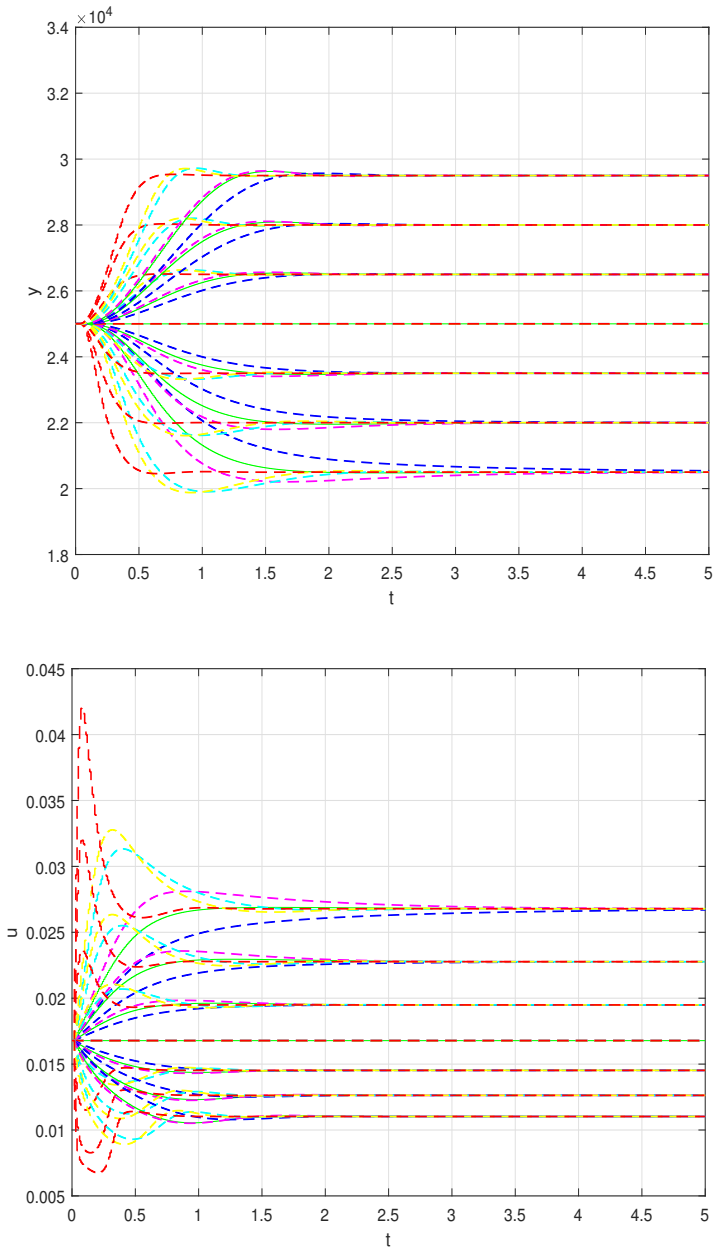

Figure 6: Responses of the control system with NDMC–NPL algorithm for different set point values and different shapes of trajectory of weighting parameters ψ_i: green – constant trajectory, blue – trajectory (31), magenta – trajectory (37) with K=100, k=5, cyan – trajectory (37) with $K = 100$, $k = 25$, yellow – trajectory (37) with $K = 150$, $k = 25$, red – trajectory (39) with $K = 150$, $k = 25$, $a = 10$, $b = 5$; output – top, input – bottom

4.3. Checking influence of disturbances and of modeling uncertainty on control quality

Additional research was done to test influence of disturbances and of modeling uncertainty on control system performance. First experiments were conducted to check the influence of the measurement noise v. This disturbance was generated as a random signal v^r with an assumed amplitude K_v:

$$v_i = K_v \cdot v_i^r , \tag{47}$$

where i is a time instant of the simulation time t.

Experiments were conducted for different values of parameter K_v and with the constant trajectory of weighting parameter ψ_i and with the trajectory (39) with parameters $K = 150$, $k = 25$, $a = 10$ and $b = 5$ used in the predictive algorithm. The influence of the disturbance on control system performance was tested for $K_v = \{50, 100, 300\}$. The purpose of these experiments was to check how disturbance influences the control system performance when the algorithm uses the mechanism of variable trajectory of weighting parameter ψ_i.

Starting from $K_v = 50$ the noise is visible in the input as well as in the output of the control system. When comparing the responses of the control system with constant or variable trajectories for increasing K_v one can observe that the disturbance influences the control system operation in the same way regardless of which trajectories are applied (Fig. 7 shows the responses obtained for $K_v = 100$). This influence is rather small, the rise times have not changed, the algorithm with variable trajectory still generates faster response, thus the implemented mechanism works as intended.

Influence of model uncertainty on control system performance when using a variable trajectory of weighting parameters ψ_i was tested in the next experiments. Model uncertainty was included by changing gain of the control plant by appropriately modifying equation (43), where the input is amplified by a constant value. This value will be changed by using a parameter K_{dk}. The modified equation is as follows:

$$\dot{x}_2 = K_{dk} \cdot 80u - 10.1022x_2 . \tag{48}$$

The experiments were done for four different values of K_{dk}: $K_{dk} = \{1.1, 1.3, 1.6, 1.8\}$ which means increasing the gain by 10%, 30%, 60% and 80%, respectively. For $K_{dk} = 1.1$ and constant trajectory the rise time increases what is more visible when the set point is increased. However the rise time when using variable trajectory (39) has not changed and it remains not influenced by modeling uncertainty. The rise time increases with each increase in parameter K_{dk} when constant trajectory is used. It also causes an inverse–response to appear starting from $K_{dk} = 1.3$. On the other hand, the rise time decreases when variable trajectory is used. Unfortunately, the overshoot increases then. Inverse–response is not present in this case. When parameter is set to $K_{dk} = 1.8$ the overshoot is

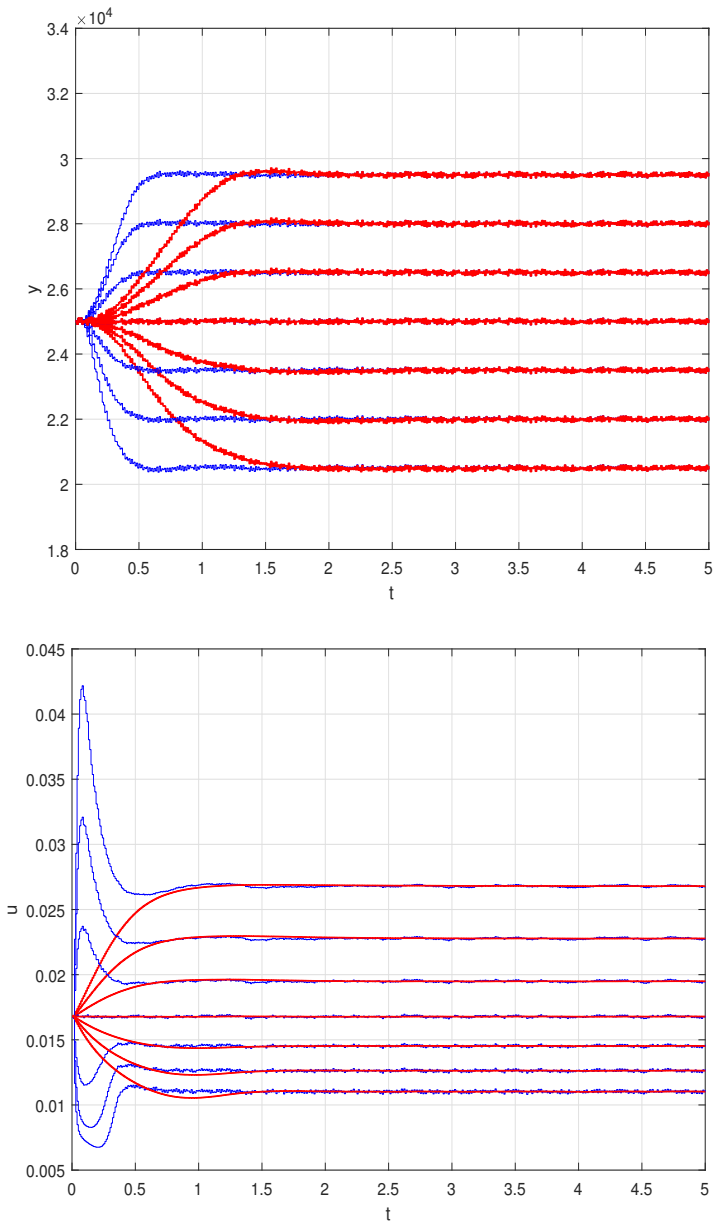

Figure 7: Responses of the control system with NDMC–NPL algorithm for different set point values, with constant (red) or variable (blue) trajectories of weighting parameters ψ_i, and with added disturbance for $K_v = 100$; output – top, input – bottom

much increased, but the overall control system performance is greatly improved in comparison when the constant trajectory is employed. Fig. 8 shows the responses obtained for $K_{dk} = 1.3$. These experiments show that when the process

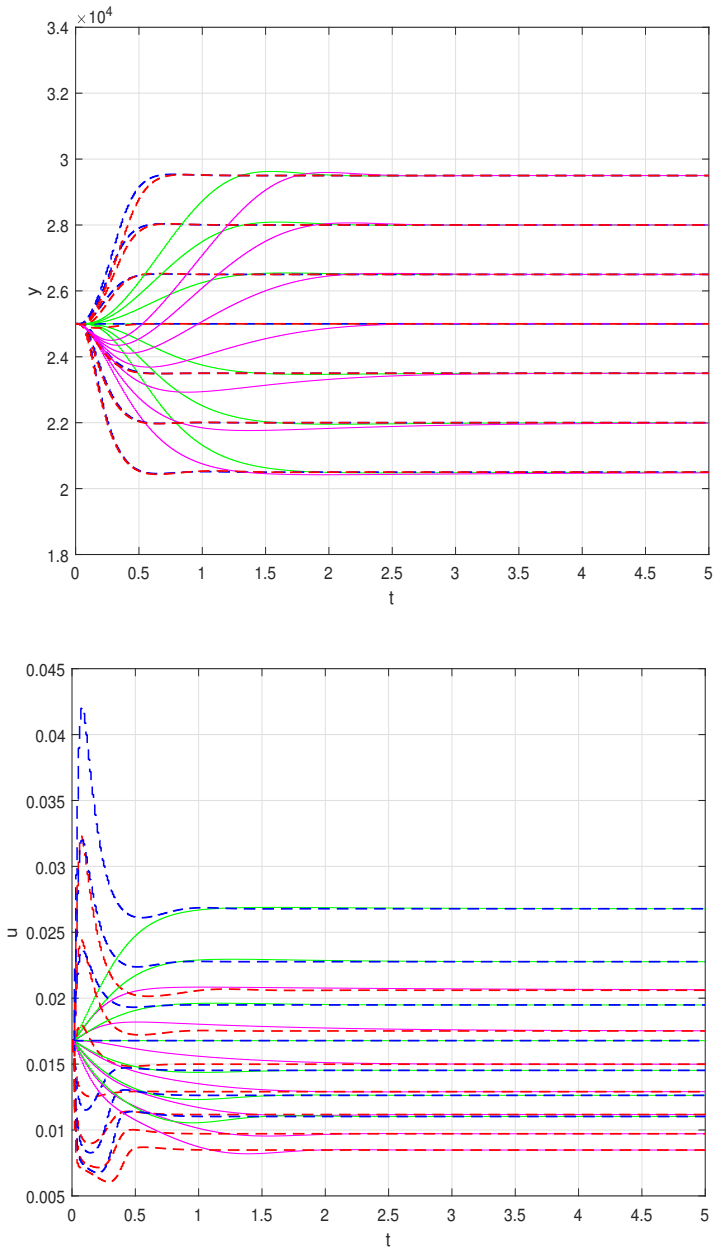

Figure 8: Responses of the control system with NDMC–NPL algorithm to different set point values with constant and variable trajectories of weighting parameters ψ_i, and without or with modeling uncertainty for $K_{dk} = 1.3$; green – nominal model and constant trajectory, blue – nominal model and variable trajectory; magenta – modeling uncertainty and constant trajectory; red – modeling uncertainty and variable trajectory; output – top, input – bottom

model implemented in the NDMC–NPL algorithm is not very accurate then by implementing a weighting parameter trajectory it is possible to reduce the influence of the model uncertainty on the control system performance. However, the variable trajectory should be chosen carefully using the methodology presented in the previous subsection so that these results could be achieved.

5. Application in the MIMO control system of an evaporator

5.1. Control plant description

The final experiments were done in the control system of a thin film evaporator which is a nonlinear MIMO plant with 3 inputs and 3 outputs. Evaporators are used to evaporate liquid substances like water. They are used to produce condensed milk, jams, caramel mass, fruit juices and also in many other applications where the final product needs to undergo an evaporation process. The nonlinear model of the control plant is described by the following equations [10]:

$$\rho A \frac{\mathrm{d}L_2}{\mathrm{d}t} = F_1 - F_4 - F_2, \tag{49}$$

$$\frac{\mathrm{d}X_2}{\mathrm{d}t} = \frac{F_1 X_1 - F_2 X_2}{20}, \tag{50}$$

$$\frac{\mathrm{d}P_2}{\mathrm{d}t} = \frac{F_4 - F_5}{4}, \tag{51}$$

$$T_2 = 0.5616 P_2 + 0.3126 X_2 + 48.43, \tag{52}$$

$$T_3 = 0.507 P_2 + 55, \tag{53}$$

$$F_4 = \frac{Q_{100} - 0.07 F_1 (T_2 - T_1)}{38.5}, \tag{54}$$

$$T_{100} = 0.1538 P_{100} + 90, \tag{55}$$

$$Q_{100} = 0.16 \, (F_1 + F_3) \, (T_{100} - T_2), \tag{56}$$

$$F_{100} = \frac{Q_{100}}{36.6}, \tag{57}$$

$$Q_{200} = \frac{0.9576 F_{200} (T_3 - T_{200})}{0.14 F_{200} + 6.84}, \tag{58}$$

$$T_{201} = T_{200} + \frac{13.68 (T_3 - T_{200})}{0.14 F_{200} + 6.84}, \tag{59}$$

$$F_5 = \frac{Q_{200}}{38.5}. \tag{60}$$

1. Process inputs are:
   - F_2 – product flowrate
   - F_{200} – cooling water flowrate
   - P_{100} – steam pressure
2. Process outputs are:
   - L_2 – separator level
   - X_2 – product composition
   - P_2 – operating pressure
3. Process disturbances are:
   - F_1 – feed flowrate
   - X_1 – feed composition
   - T_1 – feed temperature
   - T_{200} – inlet cooling water temperature
   - F_3 – circulating flowrate
4. Other process variables are:
   - F_4 – vapor flowrate
   - F_5 – condensate flowrate
   - T_2 – product temperature
   - T_3 – vapor temperature
   - F_{100} – steam flowrate
   - T_{100} – steam temperature
   - Q_{100} – heat duty
   - T_{201} – outlet cooling water temperature
   - Q_{200} – condenser duty

The constant $\rho A = 20$. Process operating conditions are as follows: $F_1 = 10$ kg/min, $F_2 = 2$ kg/min, $F_3 = 50$ kg/min, $F_4 = 8$ kg/min, $F_5 = 8$ kg/min, $X_1 = 5\%$, $X_2 = 25\%$, $T_1 = 40°$C, $T_2 = 84°$C, $T_3 = 80.6°$C, $L_2 = 1$ m, $P_2 = 50.5$ kPa, $F_{100} = 9.3$ kg/min, $T_{100} = 119.9°$C, $P_{100} = 194.7$ kPa, $Q_{100} = 339$ kW, $F_{200} = 208$ kg/min, $T_{200} = 25°$C, $T_{201} = 46.1°$C and $Q_{200} = 307.9$ kW.

For controlling the output L_2 a PI controller was implemented as recommended e.g. in [10]. A control loop F_2–L_2 was created and after several tests parameters $K_p = 8$ and $T_I = 6.8$ [min] were chosen to obtain the most reduced rise time and overshoot. The problem to control the MIMO plant with 3 inputs and 3 outputs was thus reduced and the MPC algorithm was designed to control the remaining 2 outputs, using 2 remaining manipulated variables.

5.2. Influencing control system performance with different shapes of the trajectory of weighting parameters ψ_i

The DMC controller in analytic version was used to control the evaporator. The parameters were chosen after several tests and their values are as follows: $N = 50$, $N_u = 5$, $\lambda_1 = 1$ and $\lambda_2 = 1$. For these parameters the best overall control system performance was obtained – the rise time and overshoot had low values. Sample time $T_p = 0.1$ [min] was assumed. The controller takes input constraints into consideration so they cannot be negative.

In the considered control system two separate trajectories ψ_i^1 and ψ_i^2 had to be found which would guarantee the best control performance. Both implemented variable trajectories should reduce the rise time without increasing the overshoot of all process controlled variables. Many experiments have been conducted with different shapes of trajectories. Simulation tests were performed for trajectories where: ψ_i^1 is variable and ψ_i^2 is constant; ψ_i^1 is constant and ψ_i^2 is variable; ψ_i^1 is variable and ψ_i^2 is variable. The final experiments with both trajectories variable were performed after gaining experience about the influence of each of these trajectories on control system performance. All tests were done for one set point change of either X_2 or P_2 at once; both: positive and negative changes of each set point were done. Thus, for X_2 the set point values were $X_2 = 28$ [%] and $X_2 = 22$ [%] while for $P_2 - P_2 = 53.5$ [kPa] and $P_2 = 47.5$ [kPa]. Additionally to rise time and overshoot, also the maximum control error of the stabilized variable (the one for which the set point was not changed) was measured. The values of these three parameters were used to compare control system performance.

First experiments were conducted with trajectories (31) and (34) used as either ψ_i^1 or ψ_i^2 trajectory. Applying the trajectory (31) to ψ_i^1, ψ_i^2 or both gave similar results and the improvement of control system performance was small in comparison to applying the constant trajectory. Trajectory (34) gave different results depending on the set point value; they are as follows:

- for variable ψ_i^1 and constant ψ_i^2
  - when X_2 set point was changed to $X_2 = 28$ [%] the overshoot as well as the rise time increased, but the maximum control error got reduced;
  - when X_2 set point was changed to $X_2 = 22$ [%] the overshoot and the maximum control error decreased, but the rise time increased;
  - when P_2 set point was changed to $P_2 = 53.5$ [kPa] the overshoot as well as the rise time increased, but the maximum control error got reduced;
  - when P_2 set point was changed to $P_2 = 47.5$ [kPa] the overshoot and the maximum control error got reduced, but the rise time increased;

- for constant ψ_i^1 and variable ψ_i^2
  - when X_2 set point was changed the overshoot got reduced, but the maximum control error as well as the rise time got increased;
  - when P_2 set point was changed to $P_2 = 53.5$ [kPa] the overshoot got reduced, but the maximum control error and the rise time got increased;
  - when P_2 set point was changed to $P_2 = 47.5$ [kPa] the overshoot as well as the maximum control error got reduced, but the rise time increased;
- for variable ψ_i^1 and variable ψ_i^2
  - when X_2 set point was changed to $X_2 = 28$ [%] the overshoot got reduced, but the maximum control error as well as the rise time got increased;
  - when X_2 set point was changed to $X_2 = 22$ [%] the overshoot and the maximum control error got reduced, but the rise time increased;
  - when P_2 set point was changed to $P_2 = 53.5$ [kPa] the overshoot as well as the rise time got increased, but the maximum control error got reduced;
  - when P_2 set point was changed to $P_2 = 47.5$ [kPa] the overshoot and the maximum control error got reduced, but the rise time increased;

The experiments show that these trajectories do not improve the control system performance as much as in the control system of the linear plant. Nonlinearity and the number of controlled variables make the control system not easy to tune which means it is necessary to test other shapes of trajectories to improve control quality.

In the next experiments trajectory (37) was implemented where values of parameters k and K were found using the methodology presented in the previous section. The experiments, in which the best parameters k and K were obtained, were conducted for a set point change of X_2 to $X_2 = 28$ [%]. The parameters for trajectories ψ_i^1 and ψ_i^2 were chosen as described in the following steps:

- for variable ψ_i^1 and constant ψ_i^2
  1. At the beginning $K = 50$ was assumed and changes of value of the parameter k were applied. The rise time decreased until $k = 30$, then it started to increase. However, the overshoot was getting reduced for growing k, unfortunately maximum control error was getting increased as well. Parameter $k = 30$ was chosen for the next experiments, because of the least value of the rise time.

2. For parameter $k = 30$ tests with different values of parameter K were performed. When increasing the parameter K the rise time and the overshoot were reduced. Unfortunately, the maximum control error was further increased. Finally, parameter $K = 50$ was chosen, because of the small reduction of rise time and due to increase of inputs amplitude at the beginning of control system operation with the further increase of the parameter.

- for constant ψ_i^1 and variable ψ_i^2

1. At the beginning parameter $K = 50$ was assumed for tests with different values of parameter k. The rise time and the maximum control error were constantly getting reduced with the increase of parameter k. However, the overshoot was getting increased. Parameter $k = 50$ was chosen for the next experiments, because of the least value of rise time.
2. For parameter $k = 50$ next tests were done for changes in parameter K. When the parameter got increased the rise time and the overshoot got reduced while the maximum control error got increased. However, the reduction of the rise time was very insignificant and the amplitude of inputs at the beginning of control system operation increased, therefore parameter $K = 50$ was finally chosen for this trajectory.

When the trajectory (37) was used for both ψ_i^1 and ψ_i^2, with the chosen parameters, the following results were obtained:

- when X_2 set point was changed the rise time as well as the maximum control error got reduced, but the overshoot got increased;
- when P_2 set point was changed the rise time got reduced, but the overshoot as well as the maximum control error got increased;

The best results were obtained for trajectory (37) used for both ψ_i^1 and ψ_i^2.

In the last experiments just like in previously considered control systems of SISO plants trajectories (38)–(41) were implemented using the values of parameters k and K obtained in the previous tests. Each of the trajectories ψ_i^1 and ψ_i^2 was chosen separately to give the best control performance. During the tuning process the width of each trajectory was changed according to the methodology presented in the previous section. The trajectories were found for a set point change of X_2 to $X_2 = 28$ [%]. Application of these trajectories greatly reduced the rise time in comparison to trajectory (37), in all cases. The best results obtained during these experiments are described below:

- for variable ψ_i^1 and constant ψ_i^2
  - for trajectory (38) with parameter $a = 10$ the overshoot got reduced the most;

  - for trajectory (40) with parameters $a = 10,\ c = 40$ and for trajectory (41) with $a = 10,\ b = 25,\ c = 35,\ d = 40$ the rise time got reduced the most;

- for constant ψ_i^1 and variable ψ_i^2
  - for trajectory (41) with parameters $a = 0,\ b = 45,\ c = 55,\ d = 60$ the rise time as well as the maximum control error got reduced the most;

For variable trajectory ψ_i^1 and constant ψ_i^2 the maximum control error got always increased while for variable trajectory ψ_i^2 and constant ψ_i^1 – the overshoot increased. The trajectory (40) was chosen for ψ_i^1, because of the reduced rise time and less overshoot and maximum control error than in the case of the trajectory (41). For ψ_i^2 the best trajectory (41) was chosen for which, as stated before, the rise time and the maximum control error got reduced the most.

The combination of the best trajectories found separately is thus as follows:

$$\psi_i^1 = \begin{cases} 0 & \text{for } i \leqslant a_1, \\ K \cdot \dfrac{i - a_1}{k - a_1} & \text{for } a_1 \leqslant i \leqslant k, \\ K \cdot \dfrac{c_1 - i}{c_1 - b} & \text{for } k \leqslant i \leqslant c_1, \\ 0 & \text{for } i \geqslant c_1, \end{cases}$$

$$\psi_i^2 = \begin{cases} 0 & \text{for } i \leqslant a_2, \\ K \cdot \dfrac{i - a_2}{b - a_2} & \text{for } a_2 \leqslant i \leqslant b, \\ K & \text{for } b \leqslant i \leqslant c_2, \\ K \cdot \dfrac{d - i}{d - c_2} & \text{for } c_2 \leqslant i \leqslant d, \\ 0 & \text{for } i \geqslant d, \end{cases} \tag{61}$$

where $K = 50,\ k = 30,\ a_1 = 10,\ c_1 = 40,\ a_2 = 0,\ b = 45,\ c_2 = 55,\ d = 60$.

The control system operation obtained with trajectories (61) is detailed in Table 3 and Fig. 9; changes of X_2 set point were done. Measurements of overshoots, maximum control errors and rise times of respective process variables are collected in the table; they are denoted as: $y^s_{o\_out}$, $e^s_{m\_out}$, $t^s_{r\_out}$, respectively, where *out* means the appropriate output and the superscript s indicates the value of the set point change for which it was measured:

- $s = 1$ means X_2 set point change to $X_2 = 28$ [%],
- $s = 2$ means X_2 set point change to $X_2 = 22$ [%].

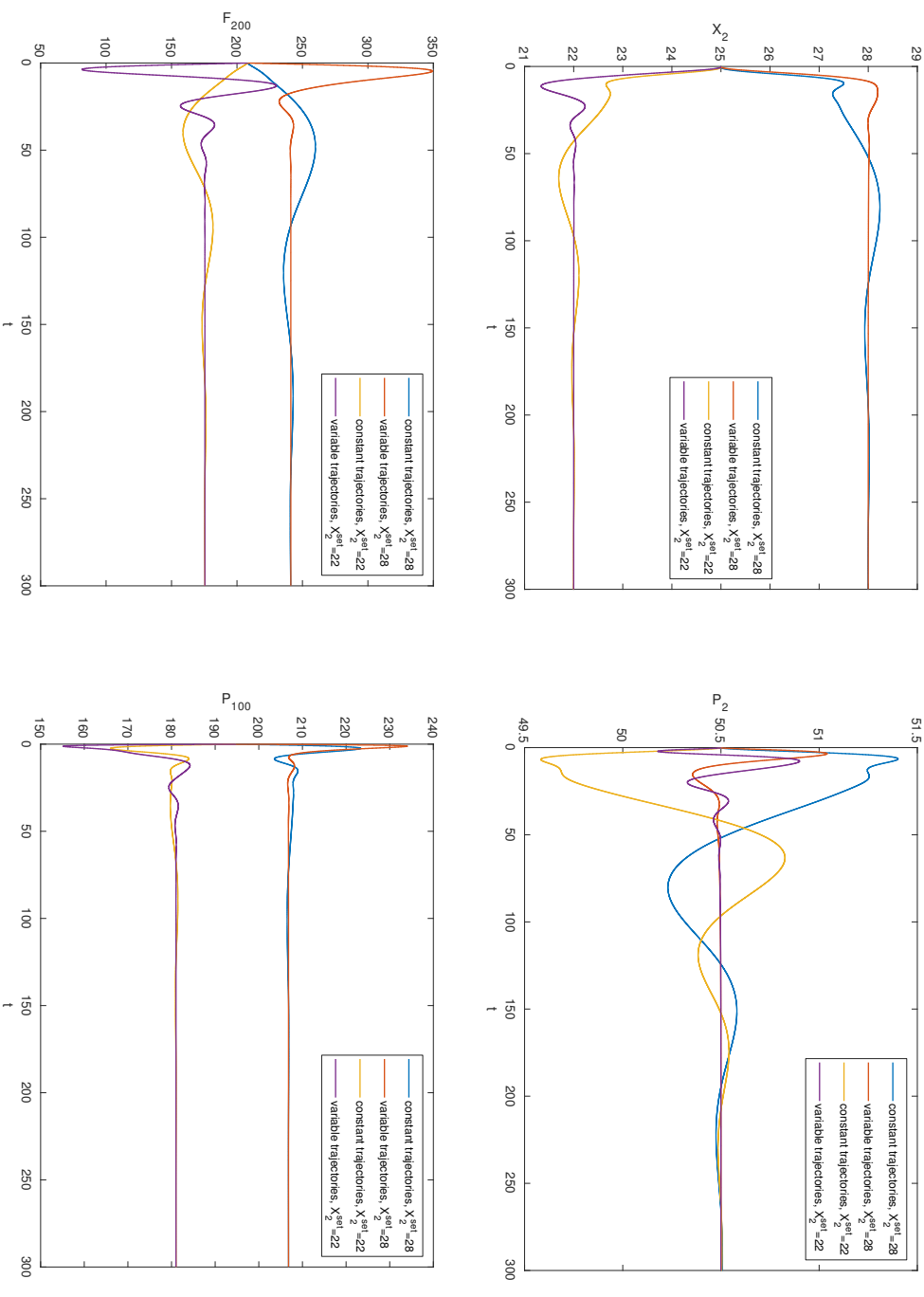

Figure 9: Responses of the control system with constant and with variable trajectories to change in set point value of X_2

Table 3: Comparison of responses obtained with constant or variable trajectories for change in set point value of variable X_2

Trajectory number	$y^1_{o\_X_2}$ [%]	$e^1_{m\_P_2}$ [%]	$y^2_{o\_X_2}$ [%]	$e^2_{m\_P_2}$ [%]	$t^1_{r\_X_2}$ [min]	$t^2_{r\_X_2}$ [min]
– (constant)	0.8502	1.7856	1.3849	0.6457	34.7	31.2
61	0.6878	1.0711	3.015	0.7931	5.2	4.7

Trajectories (61) greatly reduced the rise times $t^1_{r\_X_2}$ and $t^2_{r\_X_2}$, the overshoot $y^1_{o\_X_2}$, and the maximum control error $e^1_{m\_P_2}$. However, the overshoot $y^2_{o\_X_2}$ and the maximum control error $e^2_{m\_P_2}$ increased. Similar results were obtained for changes of P_2 set point. The results are presented in Table 4 and Fig. 10.

Table 4: Comparison of responses obtained with constant or variable trajectories for change in set point value of variable P_2

Trajectory number	$e^1_{m\_X_2}$ [%]	$y^1_{o\_P_2}$ [%]	$e^2_{m\_X_2}$ [%]	$y^2_{o\_P_2}$ [%]	$t^1_{r\_P_2}$ [min]	$t^2_{r\_P_2}$ [min]
– (constant)	7.732	1.7278	2.2201	1.3412	23.9	33.7
61	5.0692	2.545	1.8389	0.7886	3.1	12.7

Measurements are denoted as follows: $y^s_{o\_out}$, $e^s_{m\_out}$, $t^s_{r\_out}$ where the superscript s indicates the value of the set point change for which it was measured and this time:

- $s = 1$ means P_2 set point change to $P_2 = 53.5$ [kPa],
- $s = 2$ means P_2 set point change to $P_2 = 47.5$ [kPa].

After using the variable trajectories (61), the rise times $t^1_{r\_P_2}$ and $t^2_{r\_P_2}$ were greatly reduced also the overshoot $y^1_{o\_P_2}$ and the maximum control errors $e^1_{m\_X_2}$ and $e^2_{m\_X_2}$ got reduced. Only the overshoot $y^2_{o\_P_2}$ got increased. Thus, the overall control system performance has greatly improved. The experiments show that the methodology proposed for control systems of SISO plants can be also successfully applied for control systems of MIMO plants. However, finding the best trajectories is harder in this case, because of the influence each ψ^j_i $(j = 1, \ldots, n_y)$, has on each controlled variable.

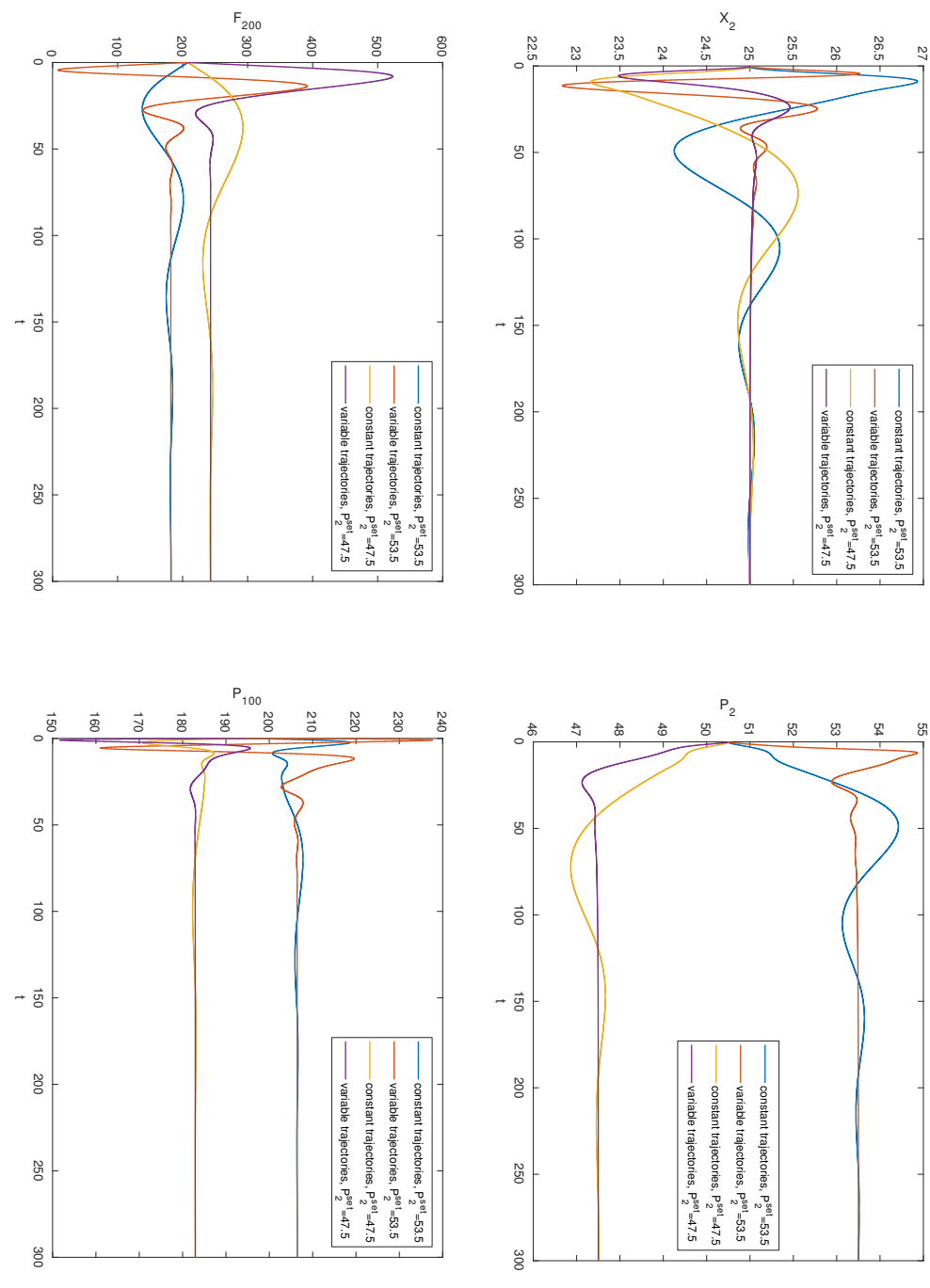

Figure 10: Responses of the control system with constant and with variable trajectories to change in set point value of P_2

6. Summary

This paper presents many shapes of trajectories of weighting parameters ψ_i which can improve control system performance when applied. The experiments were done in control systems of three control plants: the linear nonminimumphase plant, the nonlinear polymerization reactor and the nonlinear evaporator. For all three plants a DMC controller was implemented. However, for the polymerization reactor also NDMC–SL and NDMC–NPL control algorithms were designed. Simple and more complex shapes of trajectories were investigated and also their influence on each of the three designed control systems described in the paper. The tests included many changes in shape parameters and analyzing the obtained results. Moreover, for control system of the polymerization reactor influence of disturbances and model uncertainty was tested. Comparison of control systems performance was performed by comparing the values of rise times, overshoots and maximum control errors in cases when constant or variable trajectories were applied.

The best results for all control systems were obtained when using trajectories (38)–(41). For the linear plant as well as the chemical reactor a bell–shaped function improved control system performance the most. However, for the evaporator a combination of triangular and trapezoidal functions gave the best control quality. If this mechanism is used correctly it can greatly reduce either the rise time or the overshoot of the controlled process as shown in this paper. Moreover, the control system performance can be improved even if process model is not very accurate or if the measurement noise is present.

In order to find the trajectory with least rise time and overshoot, it is recommended to start with finding the best trajectory (37), i.e. to find the best values of its 2 parameters: k and K that define this trajectory. The best value of parameter k can be found by performing a few experiments with an assumed constant value of the parameter K and then changing the value of the parameter k until the least rise time is achieved. It is recommended to start the tests from $k = \frac{N}{2}$, where N is the prediction horizon, and analyze the results obtained near this value, first. The experiments described in the paper show that usually the k for which the least rise time is obtained is near this value. Once the best parameter k is found then experiments with changing the value of parameter K can begin. It is recommended to increase this parameter until its influence on control quality is barely visible or the amplitude of the input at the beginning of control system operation is too high. When both parameters are correctly found then tests with trajectories (38)–(41) can begin. It is recommended to start from a thin shape of a chosen trajectory and slowly increase its width. The experiments in the paper show that usually the best results are obtained with thinner rather than thicker trajectories.

This paper presents many trajectories that can be already used in the proposed form. Using these trajectories in control systems of plants similar to the ones described in the paper should improve control system performance. These weighting parameter trajectories can be applied to any type of predictive control algorithm and by following the presented methodology it is easy to find those that can fulfill the objectives of the control system designer.

References

[1] I.Z. Ahmetzyanov and D.N. Dem'yanov: Determination of the Overshoot Scalar Control Systems with Transfer Zero and Binomial Law of Poles Distribution, *Indian Journal of Science and Technology*, **10**(1) (2017).

[2] A.A. Aly and F.A. Salem: A New Accurate Analytical Expression for Rise Time Intended for Mechatronics Systems Performance Evaluation and Validation, *International Journal of Automation, Control and Intelligent Systems*, **1**(2) (2015), 51–60

[3] M.G. Ardakani, B. Olofsson, A. Robertsson, and R. Johansson: Real-Time Trajectory Generation using Model Predictive Control, Proceedings of 2015 IEEE Conference on Automation Science and Engineering (CASE): Automation for a Sustainable Future. *IEEE–Institute of Electrical and Electronics Engineers Inc.*, (2015), 942–948.

[4] S. Daniar, M. Shiroei, and R. Aazami: Multivariable predictive control considering time delay for load-frequency control in multi-area power systems, *Archives of Control Sciences*, **26**(4) (2016), 527–549.

[5] F.J. Doyle III, B.A. Ogunnaike, and R.K. Pearson: Nonlinear Model–based Control Using Second order Volterra Models, *Automatica*, **31**(5) (1995), 697–714.

[6] V. Exadaktylos and C.J. Taylor: Multi–objective performance optimisation for model predictive control by goal attainment, *International Journal of Control*, **83**(7) (2010), 1374–1386.

[7] C.E. Garcia and A.M. Morshedi: Quadratic Programming Solution of Dynamic Matrix Control (QDMC), *Chemical Engineering Communications*, **46**(1-3) (1986), 73–87.

[8] P. Marusak: Predictive controllers with presumed trajectory of control changes and efficient mechanism of output constraints handling (in Polish), *Pomiary Automatyka Robotyka*, 2/2008, 581–590.

[9] G.A. Nery Júnior, M.A.F. Martins, and R. Kalid: A PSO-based optimal tuning strategy for constrained multivariable predictive controllers with model uncertainty, *ISA Transactions*, **53**(2) (2014), 560–567

[10] M.F. Nor Shah, M.A. Zainal, A. Faruq, and S.S. Abdullah: Metamodeling Approach for PID Controller Optimization in an Evaporator Process, Proceedings of Fourth International Conference on Modeling, Simulation and Applied Optimization, 19–21 April 2011, Kuala Lumpur, Malaysia (2011).

[11] C. Rowe and J. Maciejowski: Tuning MPC using H_∞ Loop Shaping, In *Proceedings of the 2000 American Control Conference*, 2 (2000), 1332–1336.

[12] C. Rowe and J.M. Maciejowski: Tuning Robust Model Predictive Controllers using LQG/LTR, In *Proceedings of the 14th Triennial World Congress*, **32**(2), (1999), 1231–1236.

[13] D.E. Seborg, T.F. Edgar, D.A. Mellichamp, and F.J. Doyle III: *Process dynamics and control*, John Wiley &Sons, 2011.

[14] R. Shridhar and D.J. Cooper: A tuning strategy for unconstrained SISO model predictive control. *Industrial and Engineering Chemistry Research*, **36**(3) (1997), 729–746.

[15] R. Shridhar and D.J. Cooper: A tuning strategy for unconstrained multivariable model predictive control, *Industrial and Engineering Chemistry Research*, **37**(10) (1998), 4003–4016.

[16] M.A. Stephens, C. Manzie, and M.C. Good: Model Predictive Control for Reference Tracking on an Industrial Machine Tool Servo Drive, *IEEE Transactions on Industrial Informatics*, **9**(2) (2013), 808–816.

[17] P. Tatjewski: *Advanced Control of Industrial Processes. Structures and Algorithms*, Springer-Verlag, London, 2007.

[18] P. Tatjewski: Offset-free nonlinear Model Predictive Control with state-space process models, *Archives of Control Sciences*, **27**(4), (2017), 595–615.

[19] J.O. Trierweiler and L.A. Farina: RPN Tuning Strategy for Model Predictive Control, *Journal of Process Control*, **13**(7) (2003), 591–598.

[20] A.S. Yamashita, P.M. Alexandre, A.C. Zanin, and D. Odloak: Reference trajectory tuning of model predictive control, *Control Engineering Practice*, **50** (2016), 1–11.