ISA Transactions 67 (2017) 476–495

Contents lists available at ScienceDirect

ISA Transactions

journal homepage: www.elsevier.com/locate/isatrans

Research article

Nonlinear predictive control of a boiler-turbine unit: A state-space approach with successive on-line model linearisation and quadratic optimisation

Maciej Ławryńczuk

Institute of Control and Computation Engineering, Warsaw University of Technology, ul. Nowowiejska 15/19, 00-665 Warsaw, Poland

ARTICLE INFO

Article history:
Received 7 July 2015
Received in revised form
27 December 2016
Accepted 16 January 2017
Available online 30 January 2017

Keywords:
Process control
Model predictive control
Boiler-turbine unit
On-line successive linearisation

ABSTRACT

This paper details development of a Model Predictive Control (MPC) algorithm for a boiler-turbine unit, which is a nonlinear multiple-input multiple-output process. The control objective is to follow set-point changes imposed on two state (output) variables and to satisfy constraints imposed on three inputs and one output. In order to obtain a computationally efficient control scheme, the state-space model is successively linearised on-line for the current operating point and used for prediction. In consequence, the future control policy is easily calculated from a quadratic optimisation problem. For state estimation the extended Kalman filter is used. It is demonstrated that the MPC strategy based on constant linear models does not work satisfactorily for the boiler-turbine unit whereas the discussed algorithm with on-line successive model linearisation gives practically the same trajectories as the truly nonlinear MPC controller with nonlinear optimisation repeated at each sampling instant.

1. Introduction

A boiler-turbine unit is a crucial system in coal-fired power plants. It is used to convert fuel's chemical energy into mechanical one and then into electrical energy. The boiler generates high-pressure and high-temperature steam to drive the turbine which generates electricity. The boiler-turbine model most frequently considered in literature was developed by R.D. Bell and K.J. Åström [1], alternative models are described e.g. in [4,24]. The boiler-turbine unit is a nonlinear multiple-input multiple-output process with strong cross-coupling. The control objective is to follow the set-points and to satisfy some constraints imposed on process variables. Therefore, it usually cannot be efficiently controlled efficiently enough by simple, single-loop PID controllers [37], in particular when the operating point must be changed fast and in wide range. That is why more advanced control strategies should be applied for the boiler-turbine unit.

A number of multivariable control strategies have been applied to the boiler-turbine system, both linear and nonlinear, the state-space model or its input-output approximations may be used for controller synthesis. A multivariable PI controller and a linear quadratic regulator (LQR) are discussed in [6], the controllers' parameters are adjusted off-line by a genetic algorithm. The H_∞ controller and its transformation to a multivariable PI controller are considered in [37]. A technique which allows to respect constraints in linear control schemes such as multivariable PI and H_∞ is considered in [36]. The gain-scheduled optimal controller which takes into account the existing constraints is presented in [3], a linear parameter varying state-space model is used. A multi-objective control of the process is discussed in [2], the constraints are taken into account by a linear matrix inequality algorithm while tracking capabilities are optimised in terms of H_∞ performance. A robust adaptive sliding mode controller is presented in [11], input-output linearisation is performed to eliminate process nonlinearity. A two-level hierarchical control scheme is presented in [10] in which a supervisory fuzzy reference governor generates the set-points while a feedforward-feedback controller is used. Three single loop PID controllers are used together with a nonlinear fuzzy feedforward correctors.

As far as nonlinear control strategies are considered, a fuzzy H_∞ tracking state feedback controller based on a fuzzy state-space models may be used as described in [43]. An alternative is to use a simplified fuzzy multiple-model controller which consists of three single-loop fuzzy controllers [27]. An optimal multiple-model state-space controller is described in [13]. An approach to enforce constraints in a nonlinear state feedback controller is presented in [19].

Model Predictive Control (MPC) strategy [21,30,39] is frequently used to efficiently control numerous processes, e.g. manipulators [7], plastic injection moulding [8], distillation columns [9], chemical reactors [32], cruise control [33], air conditioning [35], and electric arc furnaces [40]. Due to its inherent ability to take into account constraints and deal with multiple-input multiple-output processes with couplings, MPC is a straightforward option for the boiler-turbine system. In the simplest case pure

E-mail address: M.Lawrynczuk@ia.pw.edu.pl

http://dx.doi.org/10.1016/j.isatra.2017.01.016
0019-0578/

Nomenclature

Fundamental model

f, f_1, f_2, f_3 model state functions
g, g_1, g_2, g_3 model output functions
q_e evaporation rate (kg/s)
u_1 valve positions for fuel flow (0–1)
u_2 valve positions for steam control (0–1)
u_3 valve positions for feedwater flow (0–1)
$x_1 = y_1$ drum pressure (kg/cm^2)
$x_2 = y_2$ electric power output (MW)
x_3 fluid density (kg/cm^3)
y_3 drum water level deviations (m)
α_{cs} steam quality (–)

Model predictive control

$\mathbf{0}_{m\times n}$ zeros matrix of dimensionality $m \times n$
$\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$ matrices of constant linear model for MPC
$\mathbf{A}(k), \mathbf{B}(k), \mathbf{C}(k), \mathbf{D}(k)$ matrices of time-varying linearised model for MPC
$\widetilde{\mathbf{A}}(k), \widetilde{\mathbf{B}}(k), \widetilde{\mathbf{C}}(k), \widetilde{\mathbf{D}}(k), \mathbf{P}(k), \mathbf{V}(k)$ auxiliary matrices based on parameters of linearised model
$d(k)$ estimation of output disturbance
$\mathbf{I}_{m\times n}$ identity matrix of dimensionality $m \times n$
$\mathbf{J}, \mathbf{J}^{\mathrm{N}}$ auxiliary constant matrices
k discrete time (the current sampling instant)
N, N_{u} prediction and control horizons
$\mathrm{SSE_x}$ sum of squared errors of controlled state variables
$\mathrm{SSE_y}$ sum of squared errors of deviations of constrained output variable
$\bar{u}(k-1)$ values of manipulated variables applied to process
$u^{\min}, u^{\max}, \mathbf{u}^{\min}, \mathbf{u}^{\max}$ definition of range constraints imposed on manipulated variables
$\bar{x}(k), \bar{x}(k-1)$ measurements (estimation) of state variables
$x_1^{\mathrm{sp}}(k+p|k), x_2^{\mathrm{sp}}(k+p|k), x_3^{\mathrm{sp}}(k+p|k)$ set-point values of state variables
$x^{\mathrm{sp}}(k+p|k), \mathbf{x}^{\mathrm{sp}}(k)$ state set-point trajectories
$\hat{x}_1(k+p|k), \hat{x}_2(k+p|k), \hat{x}_3(k+p|k)$ predicted values of state variables
$\hat{x}(k+p|k), \hat{\mathbf{x}}(k)$ predicted state trajectories
$\hat{y}_3(k+p|k)$ predicted value of output
$y_3^{\min}, y_3^{\max}, \mathbf{y}_3^{\min}, \mathbf{y}_3^{\max}$ definition of range constraints imposed on controlled variable y_3
$\hat{\mathbf{y}}(k)$ predicted output trajectory
$\mu_{1,p}, \mu_{2,p}, \mu_{3,p}, \mathbf{M}_p, \mathbf{M}$ weights in MPC related to predicted errors of controlled variables
$\lambda_{1,p}, \lambda_{2,p}, \lambda_{3,p}, \mathbf{\Lambda}_p, \mathbf{\Lambda}$ weights in MPC related to increments of manipulated variables
$\triangle u_1(k+p|k), \triangle u_2(k+p|k), \triangle u_3(k+p|k)$ future increments of manipulated variables
$\triangle u(k+p|k), \triangle \mathbf{u}(k)$ trajectory of future increments of manipulated variables
$\triangle u^{\max}, \triangle \mathbf{u}^{\max}$ definition of rate constraints imposed on manipulated variables
$\varepsilon^{\min}(k), \varepsilon^{\max}(k)$ slack variables in MPC optimisation when the output constraints are treated in a suboptimal way
$\varepsilon^{\min}(k+p), \varepsilon^{\max}(k+p), \boldsymbol{\varepsilon}^{\min}(k), \boldsymbol{\varepsilon}^{\max}(k)$ slack variables in MPC optimisation when the output constraints are treated in the optimal way
$\rho^{\min}, \rho^{\max}$ weighting coefficients (penalties)
$\nu(k)$ estimation of state disturbance

State estimation

$\mathbf{F}(k-1), \mathbf{H}(k)$ matrices of linearised model for state estimation
$\mathbf{F}^{\mathrm{a}}(k-1), \mathbf{H}^{\mathrm{a}}(k)$ matrices of augmented linearised model for state estimation
$f^{\mathrm{a}}, g^{\mathrm{a}}$ state and output function of augmented model
$\mathbf{K}(k)$ filter gain matrix
$\mathbf{P}(k|k-1), \mathbf{P}(k-1|k-1)$ covariance matrices of estimated vectors $\hat{x}(k|k-1), \hat{x}(k-1|k-1)$
$\mathbf{Q}(k-1), \mathbf{R}(k)$ noise covariance matrices
$\mathbf{S}(k)$ residual covariance matrix
$u^{\mathrm{a}}(k)$ augmented input vector
$v(k), w(k)$ process and observation noises
$x^{\mathrm{a}}(k)$ augmented state vector
$\tilde{x}_3$ estimated value of fluid density (kg/cm^3)
z measured variable
$\tilde{z}$ measurement residual

linear models may be used. An application of Generalized Predictive Control (GPC) algorithm based on a linearised model using an input-output feedback linearisation method and an MPC algorithm based on a fuzzy model are presented in [18]. Multivariable linear step-response input-output models obtained by two methods and corresponding Dynamic Matrix Control (DMC) algorithms are described in [26]. Unfortunately, the linear models of the boiler-turbine system are inaccurate and the controller performance is not satisfactory. Quite frequently fuzzy models or multi-model structures are used in MPC due to their computational simplicity. An adaptive DMC algorithm with fuzzy step-response models, which are recalculated on-line for the current operating point of the process, is presented in [25]. A fuzzy input-output model and a resulting MPC controller is discussed in [17]. A conceptually similar control scheme based on a bank of local state-space linear models used in MPC is studied in [28]. An MPC algorithm based on a fuzzy state-space model is considered in [41]. A piecewise affine state-space model of the process is discussed in [14] together with an explicit MPC algorithm which calculates on-line the control law as an affine function of system states, the control laws for a number of regions are precalculated off-line.

More advanced MPC schemes use the full nonlinear model of the process or its transformation. An application of the extended DMC technique based on a constant input-output step-response model with an additional time-varying disturbances which accounts for nonlinearity of the process is described in [12]. Constrained nonlinear MPC schemes with nonlinear optimisation repeated at each sampling instant are reported in [19], state-space models are used for prediction. An application of a genetic algorithm for optimisation in nonlinear MPC based on a fuzzy state-space model is described in [16,19]. A predictive control scheme based on a state-space hybrid model is discussed in [31], the resulting optimisation problem is of a mixed integer linear or quadratic type. Hierarchical set-point optimisation cooperating with an MPC algorithm based on state-space a fuzzy model is considered in [42].

Motivation of this work is to develop a computationally efficient nonlinear MPC algorithm for the boiler-turbine unit. In contrast to the MPC approaches with nonlinear on-line numerical optimisation discussed in [16,19], in the presented MPC scheme the original nonlinear model is successively linearised on-line for the current operating point of the process and used for prediction.

Linearisation makes it possible to easily calculate on-line the future control policy from a quadratic optimisation problem. In order to show advantages of the discussed MPC algorithm, it is compared with the fully-fledged nonlinear MPC in terms of control accuracy and computational burden. In both approaches the control objective is to follow set-point changes imposed on two state (output) variables and to satisfy constraints imposed on three inputs and one output.

This paper is organised as follows. First, Section 2 describes the state-space model of the process and control objectives. Next, Section 3 shortly reminds the general idea of MPC and gives a mathematical formulation of the MPC optimisation problem resulting from the control objectives. The main part of the paper, given in Section 4, details derivation of the computationally efficient MPC algorithm with on-line model linearisation for the boiler-turbine unit and discusses the state estimation problem. Section 5 presents simulation results. In particular, the discussed MPC strategy is compared with a simple MPC algorithm based on constant linear models and with a computationally demanding MPC algorithm with full nonlinear optimisation repeated at each sampling instant. Finally, Section 6 concludes the paper.

2. Boiler-turbine unit

2.1. Continuous-time process description

The model considered in this paper is based on the basic conservation laws and their parameters have been estimated from the data measured from the Synvendska Kraft AB Plant in Malmö, Sweden. The process is described by the following continuous-time state equations [1]

$$\begin{aligned}\frac{\mathrm{d}x_1(t)}{\mathrm{d}t} &= -0.0018u_2(t)x_1^{9/8}(t)+0.9u_1(t)-0.15u_3(t)\\ \frac{\mathrm{d}x_2(t)}{\mathrm{d}t} &= (0.073u_2(t)-0.016)x_1^{9/8}(t)-0.1x_2(t)\\ \frac{\mathrm{d}x_3(t)}{\mathrm{d}t} &= \frac{141u_3(t)-(1.1u_2(t)-0.19)x_1(t)}{85}\end{aligned} \tag{1}$$

where the state variables x_1, x_2, and x_3 denote drum pressure ($\mathrm{kg/cm^2}$), electric power output (MW), and fluid density ($\mathrm{kg/cm^3}$), respectively. The manipulated inputs, u_1, u_2, and u_3 are the valve positions for fuel flow, steam control, and feedwater flow, respectively. The outputs y_1, y_2, and y_3 of the system are: drum pressure ($\mathrm{kg/cm^2}$), electric output (MW) and drum water level deviations (m). The continuous-time output equations are

$$\begin{aligned}y_1(t) &= x_1(t)\\ y_2(t) &= x_2(t)\\ y_3(t) &= 0.05\left(0.13073x_3(t)+100\alpha_{\mathrm{cs}}(t)+\frac{q_{\mathrm{e}}(t)}{9}-67.975\right)\end{aligned} \tag{2}$$

where α_{cs} and q_{e} are steam quality (dimensionless) and evaporation rate (kg/s), respectively, and are given by

$$\alpha_{\mathrm{cs}}(t)=\frac{(1-0.001538x_3(t))(0.8x_1(t)-25.6)}{x_3(t)(1.0394-0.0012304x_1(t))} \tag{3a}$$

$$q_{\mathrm{e}}(t)=(0.854u_2(t)-0.147)x_1(t)+45.59u_1(t)-2.514u_3(t)-2.096 \tag{3b}$$

It is assumed that all the output signals are measured, which means that the first two state variables x_1 and x_2 are also measured, whereas the third state variable x_3 is not available for measurement. Therefore, for state estimation an observer must be used. The estimated state variable is denoted by $\tilde{x}_3$.

2.2. Control objectives

The boiler-turbine controller must calculate the manipulated variables u_1, u_2 and u_3 in such a way that the discrepancy between drum pressure ($y_1=x_1$), electric power output ($y_2=x_2$) and their set-points $\left(y_1^{\mathrm{sp}}=x_1^{\mathrm{sp}},\ y_2^{\mathrm{sp}}=x_2^{\mathrm{sp}}\right)$ are minimised whereas water level deviations should be in the following range [3]

$$-0.1\le y_3(t)\le 0.1 \tag{4}$$

Due to actuators' limitations, the process inputs are subject to the following magnitude constraints [3]

$$\begin{aligned}0&\le u_1(t)\le 1\\ 0&\le u_2(t)\le 1\\ 0&\le u_3(t)\le 1\end{aligned} \tag{5}$$

Additionally, rate constraints [3]

$$\begin{aligned}-0.007&\le \dot{u}_1(t)\le 0.007\\ -2&\le \dot{u}_2(t)\le 0.02\\ -0.05&\le \dot{u}_3(t)\le 0.05\end{aligned} \tag{6}$$

may be taken into account.

In this paper two notation methods are used: vectors and scalars. For compactness of presentation the vectors $u=\left[u_1\ u_2\ u_3\right]^{\mathrm{T}}$, $x=\left[x_1\ x_2\ x_3\right]^{\mathrm{T}}$, $y=\left[y_1\ y_2\ y_3\right]^{\mathrm{T}}$ are used. When it is necessary or convenient, the elements of these vectors are also used, i.e. the scalars u_n, x_n and y_n, where $n=1,2,3$.

2.3. Model discretisation

The continuous-time state equations of the boiler-turbine model (Eqs. (1)) are discreticised using the Euler's method. The discrete-time state equations are

$$\begin{aligned}x_1(k+1)&=x_1(k)+T_{\mathrm{s}}\left(-0.0018u_2(k)x_1^{9/8}(k)+0.9u_1(k)-0.15u_3(k)\right)\\ x_2(k+1)&=x_2(k)+T_{\mathrm{s}}\left((0.073u_2(k)-0.016)x_1^{9/8}(k)-0.1x_2(k)\right)\\ x_3(k+1)&=x_3(k)+T_{\mathrm{s}}\left(\frac{141u_3(k)-(1.1u_2(k)-0.19)x_1(k)}{85}\right)\end{aligned} \tag{7}$$

where $T_{\mathrm{s}}=1$ s denotes the sampling time. From Eq. (2), the discrtete-time output equations are

$$\begin{aligned}y_1(k)&=x_1(k)\\ y_2(k)&=x_2(k)\\ y_3(k)&=0.05\left(0.13073x_3(k)+100\alpha_{\mathrm{cs}}(k)+\frac{q_{\mathrm{e}}(k)}{9}-67.975\right)\end{aligned} \tag{8}$$

where, from Eqs. (3a) and (3b), one has

$$\alpha_{\mathrm{cs}}(k)=\frac{(1-0.001538x_3(k))(0.8x_1(k)-25.6)}{x_3(k)(1.0394-0.0012304x_1(k))} \tag{9a}$$

$$q_{\mathrm{e}}(k)=(0.854u_2(k)-0.147)x_1(k)+45.59u_1(k)-2.514u_3(k)-2.096 \tag{9b}$$

The discrete-time state-space model may be described by the general equations

$$x(k+1)=f(x(k),u(k)) \tag{10a}$$

$$y(k) = g(x(k), u(k)) \tag{10b}$$

where $f: \mathbb{R}^6 \to \mathbb{R}^3$ and $g: \mathbb{R}^6 \to \mathbb{R}^3$. The state equations may be rewritten for the sampling instant k, which gives

$$x(k) = f(x(k-1), u(k-1)) \tag{11a}$$

$$y(k) = g(x(k), u(k)) \tag{11b}$$

Considering the actual relations between variables, one has

$$\begin{aligned} x_1(k) &= f_1(x_1(k-1), u_1(k-1), u_2(k-1), u_3(k-1)) \\ x_2(k) &= f_2(x_1(k-1), x_2(k-1), u_2(k-1)) \\ x_3(k) &= f_3(x_1(k-1), x_3(k-1), u_2(k-1), u_3(k-1)) \end{aligned} \tag{12}$$

and

$$\begin{aligned} y_1(k) &= g_1(x_1(k)) \\ y_2(k) &= g_2(x_2(k)) \\ y_3(k) &= g_3(x_1(k), x_3(k), u_1(k), u_2(k), u_3(k)) \end{aligned} \tag{13}$$

3. Model predictive control problem formulation for the boiler-turbine unit

MPC is a computer control technique which calculates repeatedly on-line not only the current values of control signals (i.e. for the current sampling instant k), but the whole future control policy, defined on some control horizon N_u [21,30,39]. Usually, for simplicity of implementation, control increments, not their values, are calculated. The vector of decision variables is

$$\triangle \boldsymbol{u}(k) = \begin{bmatrix} \triangle u(k|k) \\ \vdots \\ \triangle u(k+N_u-1|k) \end{bmatrix}$$

Because the boiler-turbine system has 3 inputs, there are $3N_u$ decision variables. The control increments are defined as

$$\triangle u_n(k+p|k) = \begin{cases} u_n(k|k) - u_n(k-1) & \text{if } p = 0 \\ u_n(k+p|k) - u_n(k+p-1|k) & \text{if } p \geq 1 \end{cases}$$

where $n = 1, 2, 3$. It is assumed that $\triangle u_n(k+p|k) = 0$ for $p \geq N_u$. Considering the control objectives detailed in Section 2, the objective of the MPC algorithm is to minimise discrepancies between predicted values of drum pressure x_1 and its set-point trajectory x_1^{sp} as well as discrepancies between predicted values of electric power output x_2 and its set-point trajectory x_2^{sp}. These discrepancies are considered on the prediction horizon $N \geq N_u$. Thus, the following quadratic cost function is minimised on-line

$$J(k) = \sum_{n=1}^{2} \sum_{p=1}^{N} \mu_{n,p} \left(x_n^{sp}(k+p|k) - \hat{x}_n(k+p|k) \right)^2 + \sum_{n=1}^{3} \sum_{p=0}^{N_u-1} \lambda_{n,p} (\triangle u_n(k+p|k))^2 \tag{14}$$

where $\mu_{n,p} \geq 0$ and $\lambda_{n,p} > 0$ are weighting coefficients. The predictions $\hat{x}_n(k+p|k)$ are calculated from a dynamic model of the process. Additionally, the second part of the MPC cost-function minimises excessive control increments. Having calculated the decision variables at the sampling instant k, the manipulated variables are updated using the first 3 elements of the solution vector $\triangle \boldsymbol{u}(k)$, i.e.

$$u_1(k) = \triangle u_1(k|k) + u_1(k-1), \ u_2(k) = \triangle u_2(k|k) + u_2(k-1), \ u_3(k) = \triangle u_3(k|k) + u_3(k-1)$$

In the next sampling instant, the measurements of the outputs and available state variables are updated, and the whole optimisation procedure is repeated.

In the discrete-time domain and in the MPC framework, the water level deviations constraints (4) may be rewritten as the constraints imposed on the third output variable over the prediction horizon

$$-0.1 \leq \hat{y}_3(k+p|k) \leq 0.1, \ p = 1, \ldots, N \tag{15}$$

The predictions of the output variable, $\hat{y}_3(k+p|k)$, are calculated from the model. From Eq. (5), the constraints imposed on magnitude of the inputs are used in MPC over the control horizon

$$\begin{aligned} 0 &\leq u_1(k+p|k) \leq 1, \ p = 0, \ldots, N_u - 1 \\ 0 &\leq u_2(k+p|k) \leq 1, \ p = 0, \ldots, N_u - 1 \\ 0 &\leq u_3(k+p|k) \leq 1, \ p = 0, \ldots, N_u - 1 \end{aligned} \tag{16}$$

From Eq. (6), taking into account that the sampling time is $T_s = 1$ s and the magnitude constraints imposed in MPC on the second manipulated variable are $0 \leq u_2(k+p|k) \leq 1$ for $p = 0, \ldots, N_u - 1$, the rate constraints used in MPC are

$$\begin{aligned} -0.007 &\leq \triangle u_1(k+p|k) \leq 0.007, \ p = 0, \ldots, N_u - 1 \\ -1 &\leq \triangle u_2(k+p|k) \leq 0.02, \ p = 0, \ldots, N_u - 1 \\ -0.05 &\leq \triangle u_3(k+p|k) \leq 0.05, \ p = 0, \ldots, N_u - 1 \end{aligned} \tag{17}$$

Taking into account Eqs. (14), (15), (16) and (17), the constrained MPC optimisation problem which must be solved at each sampling instant on-line may be formulated in the following way

$$\min_{\substack{\triangle \boldsymbol{u}(k) \\ \varepsilon^{\min}(k) \\ \varepsilon^{\max}(k)}} \left\{ \sum_{p=1}^{N} \left\| x^{sp}(k+p|k) - \hat{x}(k+p|k) \right\|_{\boldsymbol{M}_p}^2 + \sum_{p=0}^{N_u-1} \| \triangle u(k+p|k) \|_{\Lambda_p}^2 + \rho^{\min} ((\varepsilon^{\min}(k))^2 + \rho^{\max} ((\varepsilon^{\max}(k))^2 \right\}$$

subject to

$$\begin{aligned} & u^{\min} \leq u(k+p|k) \leq u^{\max}, \ p = 0, \ldots, N_u - 1 \\ & \triangle u^{\min} \leq \triangle u(k+p|k) \leq \triangle u^{\max}, \ p = 0, \ldots, N_u - 1 \\ & y_3^{\min} - \varepsilon^{\min}(k) \leq \hat{y}_3(k+p|k) \leq y_3^{\max} + \varepsilon^{\max}(k), \ p = 1, \ldots, N \\ & \varepsilon^{\min}(k) \geq 0, \ \varepsilon^{\max}(k) \geq 0 \end{aligned} \tag{18}$$

The input constraints are defined by the vectors: $u^{\min} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T$, $u^{\max} = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}^T$, $\triangle u^{\min} = \begin{bmatrix} -0.007 & -2 & -0.05 \end{bmatrix}^T$, $\triangle u^{\max} = \begin{bmatrix} 0.007 & 0.02 & 0.05 \end{bmatrix}^T$, the output constraints by the scalars $y_3^{\min} = -0.1$, $y_3^{\max} = 0.1$. The matrices $\boldsymbol{M}_p = \mathrm{diag}(\mu_{1,p}, \mu_{2,p}, \mu_{3,p}) \geq 0$ (with $\mu_{3,p} = 0$), $\boldsymbol{M}_p = \mathrm{diag}(\lambda_{1,p}, \lambda_{2,p}, \lambda_{3,p}) > 0$ are of dimensionality 3×3. In order to guarantee that the feasible set of the MPC optimisation problem is not empty, not the hard output constraints (15), but their soft versions are used. The hard constraints may be temporarily violated, but it enforces the existence of the feasible set. The degree of constraint violation is minimised by the last two parts of the cost-function. The soft approach to output variables increases the number of decision variables of the MPC optimisation problem to $3N_u + 2$, the number of constraints is $12N_u + 4$, $\varepsilon^{\min}(k)$ and $\varepsilon^{\max}(k)$ are additional decision variables, $\rho^{\min}, \rho^{\max} > 0$ are penalty coefficients.

The MPC optimisation problem (18) may be further reformulated using the vector-matrix notation. The set-point trajectory and the predicted state trajectory vectors

$$\boldsymbol{x}^{\rm sp}(k)=\begin{bmatrix}x^{\rm sp}(k+1|k)\\ \vdots\\ x^{\rm sp}(k+N|k)\end{bmatrix},\quad \hat{\boldsymbol{x}}(k)=\begin{bmatrix}\hat{x}(k+1|k)\\ \vdots\\ \hat{x}(k+N|k)\end{bmatrix}$$

are of length $3N$, the weighting matrices $\boldsymbol{M}=\mathrm{diag}(\boldsymbol{M}_1,\ldots,\boldsymbol{M}_N)$ and $\boldsymbol{\Lambda}=\mathrm{diag}(\boldsymbol{\Lambda}_0,\ldots,\boldsymbol{\Lambda}_{N_{\rm u}-1})$ are of dimensionality $3N\times 3N$ and $3N_{\rm u}\times 3N_{\rm u}$, respectively. The input constraint vectors

$$\boldsymbol{u}^{\min}=\begin{bmatrix}u^{\min}\\ \vdots\\ u^{\min}\end{bmatrix},\quad \boldsymbol{u}^{\max}=\begin{bmatrix}u^{\max}\\ \vdots\\ u^{\max}\end{bmatrix},\quad \triangle\boldsymbol{u}^{\min}=\begin{bmatrix}\triangle u^{\min}\\ \vdots\\ \triangle u^{\min}\end{bmatrix},\quad \triangle\boldsymbol{u}^{\max}=\begin{bmatrix}\triangle u^{\max}\\ \vdots\\ \triangle u^{\max}\end{bmatrix}$$

are of length $3N_{\rm u}$, the output constraint vectors

$$\boldsymbol{y}_3^{\min}=\begin{bmatrix}y_3^{\min}\\ \vdots\\ y_3^{\min}\end{bmatrix},\quad \boldsymbol{y}_3^{\max}=\begin{bmatrix}y_3^{\max}\\ \vdots\\ y_3^{\max}\end{bmatrix}$$

are of length N. The general MPC optimisation problem defined by Eq. (18) can be expressed as

$$\min_{\substack{\triangle\boldsymbol{u}(k)\\ \varepsilon^{\min}(k)\\ \varepsilon^{\max}(k)}}\left\{\left\|\boldsymbol{x}^{\rm sp}(k)-\hat{\boldsymbol{x}}(k)\right\|_{\boldsymbol{M}}^2+\|\triangle\boldsymbol{u}(k)\|_{\boldsymbol{\Lambda}}^2+\rho^{\min}(\varepsilon^{\min}(k))^2+\rho^{\max}(\varepsilon^{\max}(k))^2\right\}$$

subject to

$$\begin{aligned}&\boldsymbol{u}^{\min}\le\boldsymbol{u}(k)\le\boldsymbol{u}^{\max}\\ &\triangle\boldsymbol{u}^{\min}\le\triangle\boldsymbol{u}(k)\le\triangle\boldsymbol{u}^{\max}\\ &\boldsymbol{y}_3^{\min}-\varepsilon^{\min}(k)\boldsymbol{I}_{N\times1}\le\hat{\boldsymbol{y}}_3(k)\le\boldsymbol{y}_3^{\max}+\varepsilon^{\max}(k)\boldsymbol{I}_{N\times1}\\ &\varepsilon^{\min}(k)\ge0,\quad \varepsilon^{\max}(k)\ge0\end{aligned}\tag{19}$$

One may note that in the MPC optimisation problems (18) and (19) the constraints imposed on the predicted output variable y_3 are used in a simple, suboptimal version, because the same coefficients $\varepsilon^{\min}(k)$ and $\varepsilon^{\max}(k)$ are used over the whole prediction horizon to soften the original hard constraints (15). In the general optimal approach the vectors of additional variables are not constant over the prediction horizon, but the degree of hard constraints violation is adjusted independently for consecutive sampling instants over the prediction horizon. In place of the MPC optimisation problem (18) one has

$$\min_{\substack{\triangle\boldsymbol{u}(k)\\ \varepsilon^{\min}(k+p)\\ \varepsilon^{\max}(k+p)}}\left\{\sum_{p=1}^{N}\left\|x^{\rm sp}(k+p|k)-\hat{x}(k+p|k)\right\|_{\boldsymbol{M}_p}^2+\sum_{p=0}^{N_{\rm u}-1}\|\triangle u(k+p|k)\|_{\boldsymbol{\Lambda}_p}^2+\rho^{\min}\sum_{p=1}^{N}\|\varepsilon^{\min}(k+p)\|^2+\rho^{\max}\sum_{p=1}^{N}\|\varepsilon^{\max}(k+p)\|^2\right\}$$

subject to

$$\begin{aligned}&u^{\min}\le u(k+p|k)\le u^{\max},\quad p=0,\ldots,N_{\rm u}-1\\ &\triangle u^{\min}\le\triangle u(k+p|k)\le\triangle u^{\max},\quad p=0,\ldots,N_{\rm u}-1\\ &y_3^{\min}-\varepsilon^{\min}(k+p)\le\hat{y}_3(k+p|k)\le y_3^{\max}+\varepsilon^{\max}(k+p),\quad p=1,\ldots,N\\ &\varepsilon^{\min}(k+p)\ge0,\quad \varepsilon^{\max}(k+p)\ge0,\quad p=1,\ldots,N\end{aligned}\tag{20}$$

The number of decision variables is $3N_{\rm u}+2N$, the number of constraints is $12N_{\rm u}+4N$. Using the vectors of additional variables

$$\boldsymbol{\varepsilon}^{\min}(k)=\begin{bmatrix}\varepsilon^{\min}(k+1)\\ \vdots\\ \varepsilon^{\min}(k+N)\end{bmatrix},\quad \boldsymbol{\varepsilon}^{\max}(k)=\begin{bmatrix}\varepsilon^{\max}(k+1)\\ \vdots\\ \varepsilon^{\max}(k+N)\end{bmatrix}$$

of length N, the MPC optimisation problem (20) may be also rewritten using the vector-matrix notation in the following way

$$\min_{\substack{\triangle\boldsymbol{u}(k)\\ \boldsymbol{\varepsilon}^{\min}(k)\\ \boldsymbol{\varepsilon}^{\max}(k)}}\left\{\left\|\boldsymbol{x}^{\rm sp}(k)-\hat{\boldsymbol{x}}(k)\right\|_{\boldsymbol{M}}^2+\|\triangle\boldsymbol{u}(k)\|_{\boldsymbol{\Lambda}}^2+\rho^{\min}\left\|\boldsymbol{\varepsilon}^{\min}(k)\right\|^2+\rho^{\max}\|\boldsymbol{\varepsilon}^{\max}(k)\|^2\right\}$$

subject to

$$\begin{aligned}&\boldsymbol{u}^{\min}\le\boldsymbol{u}(k)\le\boldsymbol{u}^{\max}\\ &\triangle\boldsymbol{u}^{\min}\le\triangle\boldsymbol{u}(k)\le\triangle\boldsymbol{u}^{\max}\\ &\boldsymbol{y}_3^{\min}-\boldsymbol{\varepsilon}^{\min}(k)\le\hat{\boldsymbol{y}}_3(k)\le\boldsymbol{y}_3^{\max}+\boldsymbol{\varepsilon}^{\max}(k)\\ &\boldsymbol{\varepsilon}^{\min}(k)\ge\boldsymbol{0}_{N\times1},\quad \boldsymbol{\varepsilon}^{\max}(k)\ge\boldsymbol{0}_{N\times1}\end{aligned}\tag{21}$$

4. Nonlinear MPC algorithm with successive on-line model linearisation (MPC-SL) for the boiler-turbine unit

If the nonlinear state-space model defined by Eqs. (7), (8), (9a) and (9b) is used for prediction calculation in MPC, the predictions of the state variables x_1, x_2 and of the output variable y_3 are nonlinear functions of the calculated on-line future control moves $\triangle\boldsymbol{u}(k)$. In consequence, the MPC optimisation problems (19) or (21) are nonlinear tasks which must be solved on-line in real time.

In order to obtain a computationally efficient MPC algorithm, the MPC scheme with Successive Linearisation (MPC-SL) is developed for the boiler-turbine system. A linear approximation of the model is repeatedly, at each sampling instant, calculated on-line for the current operating point of the process and used for prediction, which makes it possible to calculate the decision variables from an easy to solve quadratic optimisation problem (by the active set method or the interior point method [23]). The general formulation of the MPC-SL algorithm based on input-output models is given in [39], algorithm implementation for neural models is given in [20], the description for state-space models is presented in [15], but in this work a simple approach to prediction which guarantees offset-free control [38] is used.

4.1. On-line model linearisation

Using the Taylor series expansion method, the local linear approximation of the nonlinear state-space model described by Eqs. (11a) and (11b) is

$$\begin{aligned}x(k)=f(\bar{x}(k-1),\bar{u}(k-1))&+\boldsymbol{A}(k)(x(k-1)-\bar{x}(k-1))\\ &+\boldsymbol{B}(k)(u(k-1)-\bar{u}(k-1))\end{aligned}\tag{22a}$$

$$\begin{aligned}y(k)=g(\bar{x}(k),\bar{u}(k-1))&+\boldsymbol{C}(k)(x(k)-\bar{x}(k))\\ &+\boldsymbol{D}(k)(u(k)-\bar{u}(k-1))\end{aligned}\tag{22b}$$

where the vectors

$$\bar{x}(k-1)=\begin{bmatrix}\bar{x}_1(k-1)\\ \bar{x}_2(k-1)\\ \tilde{x}_3(k-1)\end{bmatrix},\quad \bar{x}(k)=\begin{bmatrix}\bar{x}_1(k)\\ \bar{x}_2(k)\\ \tilde{x}_3(k)\end{bmatrix},\quad \bar{u}(k-1)=\begin{bmatrix}\bar{u}_1(k-1)\\ \bar{u}_2(k-1)\\ \bar{u}_3(k-1)\end{bmatrix}$$

define the current operating point of the process. It is described by real measurements of the first two state variables ($\bar{x}_1$, $\bar{x}_2$) and by the estimated state of the third state variable ($\tilde{x}_3$) as well as by the values of the manipulated variables applied to the process at the previous sampling instant. The vectors $x(k-1)$, $x(k)$ and $u(k)$ are arguments (independent variables) of the linearised model. Because linearisation precedes optimisation of the future control increments at each sampling instant, the current operating point is defined in the output Eq. (22b) not by unavailable signals $\bar{u}_1(k)$, $\bar{u}_2(k)$, $\bar{u}_3(k)$, but by the most recent values from the previous sampling instant. The linearised model (22a)–(22b) may be expressed in the following form

$$x(k) = \boldsymbol{A}(k)x(k-1) + \boldsymbol{B}(k)u(k-1) + \delta_{\mathrm{x}}(k) \tag{23a}$$

$$y(k) = \boldsymbol{C}(k)x(k) + \boldsymbol{D}(k)u(k) + \delta_{\mathrm{y}}(k) \tag{23b}$$

where

$$\begin{aligned} \delta_{\mathrm{x}}(k) &= f(\bar{x}(k-1), \bar{u}(k-1)) - \boldsymbol{A}(k)\bar{x}(k-1) - \boldsymbol{B}(k)\bar{u}(k-1) \\ \delta_{\mathrm{y}}(k) &= g(\bar{x}(k), \bar{u}(k-1)) - \boldsymbol{C}(k)\bar{x}(k) - \boldsymbol{D}(k)\bar{u}(k-1) \end{aligned} \tag{24}$$

The time-varying matrices of the linearised model, all of dimensionality 3×3, are calculated analytically on-line by differentiating the nonlinear model (11a)–(11b)

$$\begin{aligned} \boldsymbol{A}(k) &= \left.\frac{\mathrm{d}f(x(k-1), u(k-1))}{\mathrm{d}x(k-1)}\right|_{\substack{x(k-1)=\bar{x}(k-1)\\u(k-1)=\bar{u}(k-1)}} \\ \boldsymbol{B}(k) &= \left.\frac{\mathrm{d}f(x(k-1), u(k-1))}{\mathrm{d}u(k-1)}\right|_{\substack{x(k-1)=\bar{x}(k-1)\\u(k-1)=\bar{u}(k-1)}} \\ \boldsymbol{C}(k) &= \left.\frac{\mathrm{d}h(x(k), u(k))}{\mathrm{d}x(k)}\right|_{\substack{x(k)=\bar{x}(k)\\u(k)=\bar{u}(k-1)}} \\ \boldsymbol{D}(k) &= \left.\frac{\mathrm{d}h(x(k), u(k))}{\mathrm{d}u(k)}\right|_{\substack{x(k)=\bar{x}(k)\\u(k)=\bar{u}(k-1)}} \end{aligned} \tag{25}$$

Taking into account the linearisation rules (25) and the model state Eq. (7) of the boiler-turbine system, one obtains

$$\boldsymbol{A}(k) = \begin{bmatrix} 1 + \left(-0.002025\bar{u}_2(k-1)\bar{x}_1^{1/8}(k-1)\right)T_{\mathrm{s}} & 0 & 0 \\ \left(0.082125\bar{u}_2(k-1)\bar{x}_1^{1/8}(k-1)\right. & 1-0.1T_{\mathrm{s}} & 0 \\ \left.-0.018\bar{x}_1^{1/8}(k-1)\right)T_{\mathrm{s}} & & \\ \dfrac{-1.1\bar{u}_2(k-1)+0.19}{85}T_{\mathrm{s}} & 0 & 1 \end{bmatrix} \tag{26}$$

and

$$\boldsymbol{B}(k) = \begin{bmatrix} 0.9T_{\mathrm{s}} & T_{\mathrm{s}}\left(-0.0018\bar{x}_1^{9/8}(k-1)\right) & -0.15T_{\mathrm{s}} \\ 0 & \left(0.073\bar{x}_1^{9/8}(k-1)\right)T_{\mathrm{s}} & 0 \\ 0 & \dfrac{-1.1\bar{x}_1(k-1)}{85}T_{\mathrm{s}} & \dfrac{141}{85}T_{\mathrm{s}} \end{bmatrix} \tag{27}$$

Taking into account Eqs. (25) and the model output Eq. (8) of the process, one has

$$\boldsymbol{C}(k) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 5\left.\dfrac{\partial\alpha_{\mathrm{cs}}(k)}{\partial x_1(k)}\right|_{\substack{x_1(k)=\bar{x}_1(k)\\x_3(k)=\tilde{x}_3(k)}} + \dfrac{0.0427\bar{u}_2(k-1)-0.00735}{9} & 0 & 0.0065365 + 5\left.\dfrac{\partial\alpha_{\mathrm{cs}}(k)}{\partial x_3(k)}\right|_{\substack{x_1(k)=\bar{x}_1(k)\\x_3(k)=\tilde{x}_3(k)}} \end{bmatrix} \tag{28}$$

where, from Eq. (9a)

$$\left.\frac{\partial\alpha_{\mathrm{cs}}(k)}{\partial x_1(k)}\right|_{\substack{x_1(k)=\bar{x}_1(k)\\x_3(k)=\tilde{x}_3(k)}} = \frac{N_1(k)}{D(k)}, \quad \left.\frac{\partial\alpha_{\mathrm{cs}}(k)}{\partial x_3(k)}\right|_{\substack{x_1(k)=\bar{x}_1(k)\\x_3(k)=\tilde{x}_3(k)}} = \frac{N_3(k)}{D(k)} \tag{29}$$

and

$$\begin{aligned} N_1(k) &= (0.8 - 0.0012304\tilde{x}_3(k))(1.0394\tilde{x}_3(k) - 0.0012304\bar{x}_1(k)\tilde{x}_3(k)) - (0.8\bar{x}_1(k) \\ &\quad - 25.6 - 0.0012304\bar{x}_1(k)\tilde{x}_3(k) + 0.0393728\tilde{x}_3(k))(-0.0012304\tilde{x}_3(k)) \\ N_3(k) &= (-0.0012304\bar{x}_1(k) + 0.0393728)(1.0394\tilde{x}_3(k) - 0.0012304\bar{x}_1(k)\tilde{x}_3(k)) - (0.8\bar{x}_1(k) \\ &\quad - 25.6 - 0.0012304\bar{x}_1(k)\tilde{x}_3(k) + 0.0393728\tilde{x}_3(k))(1.0394 - 0.0012304\bar{x}_1(k)) \\ D(k) &= (1.0394\tilde{x}_3(k) - 0.0012304\bar{x}_1(k)\tilde{x}_3(k))^2 \end{aligned} \tag{30}$$

From Eqs. (25), (8) and (9b), it follows that

$$\boldsymbol{D}(k) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \dfrac{2.2795}{9} & \dfrac{0.0427}{9}\bar{x}_1(k-1) & -\dfrac{0.1257}{9} \end{bmatrix} \tag{31}$$

4.2. State and output prediction

In this work the state disturbance model discussed in [38,39] is used. The predicted state vector for the sampling instant $k+p$ is calculated at the current instant k from

$$\hat{x}(k+p|k) = x(k+p|k) + \nu(k) \tag{32}$$

where $x(k+p|k)$ denotes the state vector obtained from the model and it is assumed that the same state disturbance $\nu(k)$ acts on the process over the whole prediction horizon. The unknown disturbance vector may be assessed as the difference between the state variables $x_1(k)$, $x_2(k)$ (measured), $\tilde{x}_3(k)$ (estimated) and the state variables calculated from the linearised state Eq. (23a) for the sampling instant k

$$\nu(k) = \begin{bmatrix} \nu_1(k) \\ \nu_2(k) \\ \nu_3(k) \end{bmatrix} = x(k) - \boldsymbol{A}(k)x(k-1) - \boldsymbol{B}(k)u(k-1) - \delta_{\mathrm{x}}(k) \tag{33}$$

The first two state variables are measured whereas the third one is observed, which means that $x(k) = \left[x_1(k)\ x_2(k)\ \tilde{x}_3(k)\right]^{\mathrm{T}}$, $x(k-1) = \left[x_1(k-1)\ x_2(k-1)\ \tilde{x}_3(k-1)\right]^{\mathrm{T}}$. Similary to Eq. (32), the predicted output for the sampling instant $k+p$ is calculated at the current instant k from

$$\hat{y}_3(k+p|k) = y_3(k+p|k) + d(k) \tag{34}$$

where $y_3(k+p|k)$ denotes the output obtained from the linearised model and it is assumed that the same output disturbance $d(k)$ acts on the process over the whole prediction horizon. Taking into account Eq. (23b), the output disturbance is assessed from

$$d(k) = y_3(k) - \boldsymbol{C}(k)x(k) - \boldsymbol{D}(k)u(k-1) - \delta_{\mathrm{y}}(k) \tag{35}$$

Because the signals $u(k)$ are not known before optimisation, the vector $u(k-1)$ from the previous sampling instant is used. The state and output predictions are calculated in a relatively straightforward way from Eqs. (32) and (34), the state and output disturbances are estimated from Eqs. (33) and (35). The prediction equations are very similar to those used in MPC in which input-output models are used for prediction. The discussed approach, although simple, guarantees offset-free control, even in the case of deterministic constant-type disturbances acting on the process and modelling errors, as proved and demonstrated in [38]. The alternative offset-free formulations of MPC based on state-space models require augmenting the process state by the states of deterministic disturbances [21,22,29] or using the velocity form state-space model, in which the extended state consists of state increments and the output signals [21].

From the state prediction Eq. (32) and using the linearised state model (23a), state predictions for the consecutive sampling instants are

$$\begin{aligned}
\hat{x}(k+1|k) &= \boldsymbol{A}(k)x(k) + \boldsymbol{B}(k)u(k|k) + \delta_{\mathrm{x}}(k) + \nu(k)\\
\hat{x}(k+2|k) &= \boldsymbol{A}(k)\hat{x}(k+1|k) + \boldsymbol{B}(k)u(k+1|k) + \delta_{\mathrm{x}}(k) + \nu(k)\\
\hat{x}(k+3|k) &= \boldsymbol{A}(k)\hat{x}(k+2|k) + \boldsymbol{B}(k)u(k+2|k) + \delta_{\mathrm{x}}(k) + \nu(k)\\
&\vdots
\end{aligned}$$

The state predictions may be expressed as functions of the increments of the future control increments (i.e. the decision variables of MPC)

$$\begin{aligned}
\hat{x}(k+1|k) &= \boldsymbol{A}(k)x(k) + \boldsymbol{B}(k)\triangle u(k|k) + \boldsymbol{B}(k)u(k-1) + \delta_{\mathrm{x}}(k) + \nu(k)\\
\hat{x}(k+2|k) &= \boldsymbol{A}^2(k)x(k) + (\boldsymbol{A}(k)+\boldsymbol{I}_{3\times 3})\boldsymbol{B}(k)\triangle u(k|k) + \boldsymbol{B}(k)\triangle u(k+1|k)\\
&\quad + (\boldsymbol{A}(k)+\boldsymbol{I}_{3\times 3})\boldsymbol{B}(k)u(k-1) + (\boldsymbol{A}(k)+\boldsymbol{I}_{3\times 3})\nu(k)\\
&\quad + (\boldsymbol{A}(k)+\boldsymbol{I}_{3\times 3})\delta_{\mathrm{x}}(k)\\
\hat{x}(k+3|k) &= \boldsymbol{A}^3(k)x(k) + (\boldsymbol{A}^2(k)+\boldsymbol{A}(k)+\boldsymbol{I}_{3\times 3})\boldsymbol{B}(k)\triangle u(k|k)\\
&\quad + (\boldsymbol{A}(k)+\boldsymbol{I}_{3\times 3})\boldsymbol{B}(k)\triangle u(k+1|k) + \boldsymbol{B}(k)\triangle u(k+2|k)\\
&\quad + (\boldsymbol{A}^2(k)+\boldsymbol{A}(k)+\boldsymbol{I}_{3\times 3})\boldsymbol{B}(k)u(k-1)\\
&\quad + (\boldsymbol{A}^2(k)+\boldsymbol{A}(k)+\boldsymbol{I}_{3\times 3})\nu(k) + (\boldsymbol{A}^2(k)+\boldsymbol{A}(k)+\boldsymbol{I}_{3\times 3})\delta_{\mathrm{x}}(k)\\
&\quad \vdots
\end{aligned}$$

where $\boldsymbol{I}_{3\times 3}$ stands for an identity matrix of dimensionality 3×3. The predicted state trajectory over the prediction horizon (the vector of length $3N$) may be expressed in the following compact manner

$$\hat{\boldsymbol{x}}(k) = \begin{bmatrix}\hat{x}(k+1|k)\\ \vdots \\ \hat{x}(k+N|k)\end{bmatrix} = \widetilde{\boldsymbol{A}}(k)x(k) + \widetilde{\boldsymbol{B}}(k)u(k-1) + \boldsymbol{V}(k)(\nu(k)+\delta_{\mathrm{x}}(k)) + \boldsymbol{P}(k)\triangle\boldsymbol{u}(k) \tag{36}$$

The matrices $\widetilde{\boldsymbol{A}}(k)$, $\widetilde{\boldsymbol{B}}(k)$ and $\boldsymbol{V}(k)$, of dimensionality $3N\times 3$, are

$$\widetilde{\boldsymbol{A}}(k) = \begin{bmatrix}\boldsymbol{A}(k)\\ \vdots\\ \boldsymbol{A}^N(k)\end{bmatrix},\quad \boldsymbol{V}(k) = \begin{bmatrix}\boldsymbol{I}_{3\times 3}\\ \boldsymbol{A}(k)+\boldsymbol{I}_{3\times 3}\\ \vdots\\ \sum_{i=1}^{N-1}\boldsymbol{A}^i(k)+\boldsymbol{I}_{3\times 3}\end{bmatrix},\quad \widetilde{\boldsymbol{B}}(k) = \boldsymbol{V}(k)\boldsymbol{B}(k) \tag{37}$$

whereas the matrix $\boldsymbol{P}(k)$, of dimensionality $3N\times 3N_{\mathrm{u}}$, is

$$\boldsymbol{P}(k) = \begin{bmatrix}
\boldsymbol{B}(k) & \dots & \boldsymbol{0}_{3\times 3}\\
(\boldsymbol{A}(k)+\boldsymbol{I}_{3\times 3})\boldsymbol{B}(k) & \dots & \boldsymbol{0}_{3\times 3}\\
\vdots & \ddots & \vdots\\
\left(\sum_{i=1}^{N_{\mathrm{u}}-1}\boldsymbol{A}^i(k)+\boldsymbol{I}_{3\times 3}\right)\boldsymbol{B}(k) & \dots & \boldsymbol{B}(k)\\
\left(\sum_{i=1}^{N_{\mathrm{u}}}\boldsymbol{A}^i(k)+\boldsymbol{I}_{3\times 3}\right)\boldsymbol{B}(k) & \dots & (\boldsymbol{A}(k)+\boldsymbol{I}_{3\times 3})\boldsymbol{B}(k)\\
\vdots & \ddots & \vdots\\
\left(\sum_{i=1}^{N-1}\boldsymbol{A}^i(k)+\boldsymbol{I}_{3\times 3}\right)\boldsymbol{B}(k) & \dots & \left(\sum_{i=1}^{N-N_{\mathrm{u}}}\boldsymbol{A}^i(k)+\boldsymbol{I}_{3\times 3}\right)\boldsymbol{B}(k)
\end{bmatrix} \tag{38}$$

where $\boldsymbol{0}_{3\times 3}$ stands for a zeros matrix of dimensionality 3×3.

From the output prediction Eq. (34), using the linearised output model (23b) and the predicted state trajectory (36), predictions of the third output variable for the consecutive sampling instants over the prediction horizon are

$$\hat{\boldsymbol{y}}_3(k) = \begin{bmatrix}\hat{y}_3(k+1|k)\\ \vdots\\ \hat{y}_3(k+N|k)\end{bmatrix} = \widetilde{\boldsymbol{C}}(k)\hat{\boldsymbol{x}}(k) + \widetilde{\boldsymbol{D}}(k)\boldsymbol{u}^{\mathrm{N}}(k) + (\delta_{\mathrm{y}}(k)+d(k))\boldsymbol{I}_{N\times 1} \tag{39}$$

where the block-diagonal matrices are

$$\widetilde{\boldsymbol{C}}(k) = \begin{bmatrix}\boldsymbol{C}_3(k) & \dots & \boldsymbol{0}_{1\times 3}\\ \vdots & \ddots & \vdots\\ \boldsymbol{0}_{1\times 3} & \dots & \boldsymbol{C}_3(k)\end{bmatrix},\quad \widetilde{\boldsymbol{D}}(k) = \begin{bmatrix}\boldsymbol{D}_3(k) & \dots & \boldsymbol{0}_{1\times 3}\\ \vdots & \ddots & \vdots\\ \boldsymbol{0}_{1\times 3} & \dots & \boldsymbol{D}_3(k)\end{bmatrix}$$

and the submatices $\boldsymbol{C}_3(k)$ and $\boldsymbol{D}_3(k)$ contains the third row of the matrices $\boldsymbol{C}(k)$ and $\boldsymbol{D}(k)$, respectively. The vector $\boldsymbol{u}^{\mathrm{N}}(k)$, of length $3N$, consists of all future control values over the prediction horizon. It may be easily found from the future control increments defined over the control horizon

$$\boldsymbol{u}^{\mathrm{N}}(k) = \begin{bmatrix}u(k+1|k)\\ \vdots\\ u(k+N|k)\end{bmatrix} = \boldsymbol{J}^{\mathrm{N}}\triangle\boldsymbol{u}(k) + \boldsymbol{u}^{\mathrm{N}}(k-1)$$

The entries of the matrix $\boldsymbol{J}^{\mathrm{N}}$, of dimensionality $3N\times 3N_{\mathrm{u}}$, are

$$J^{\mathrm{N}}_{i,j} = \begin{cases}1 & \text{if } j < i+2\\ 0 & \text{if } j \geq i+2\end{cases}$$

and the vector $\boldsymbol{u}^{\mathrm{N}}(k-1)$, of length $3N$, is comprised of the values of the manipulated variables from the previous sampling instant

$$\boldsymbol{u}^{\mathrm{N}}(k-1) = \begin{bmatrix}u(k-1)\\ \vdots\\ u(k-1)\end{bmatrix}$$

Using the state prediction Eq. (36), the output predictions given by Eq. (39) may be rewritten in the form

$$\hat{\boldsymbol{y}}_3(k) = \widetilde{\boldsymbol{C}}(k)\left[\widetilde{\boldsymbol{A}}(k)x(k) + \widetilde{\boldsymbol{B}}(k)u(k-1) + \boldsymbol{V}(k)(\nu(k)+\delta_{\mathrm{x}}(k))\right] + (\widetilde{\boldsymbol{C}}(k)\boldsymbol{P}(k) + \widetilde{\boldsymbol{D}}(k)\boldsymbol{J}^{\mathrm{N}})\triangle\boldsymbol{u}(k) + \widetilde{\boldsymbol{D}}(k)\boldsymbol{u}^{\mathrm{N}}(k-1) + (\delta_{\mathrm{y}}(k)+d(k))\boldsymbol{I}_{N\times 1} \tag{40}$$

4.3. Quadratic programming MPC-SL optimisation problem

It is interesting to notice that as a result of model linearisation, predictions of state and output variables (Eqs. (36) and (40)) are linear functions of the future control increments $\triangle\boldsymbol{u}(k)$, i.e. the decision variables of MPC. In consequence, it is possible to obtain MPC quadratic optimisation problems, in which the minimised cost-function is quadratic and the constraints are linear with respect to the decision variables. The first version of the rudimentary MPC optimisation problem (19) can be transformed to the following quadratic optimisation task

$$\min_{\substack{\triangle \boldsymbol{u}(k)\\ \varepsilon^{\min}(k)\\ \varepsilon^{\max}(k)}} \left\{ \left\| \boldsymbol{x}^{\mathrm{sp}}(k) - \widetilde{\boldsymbol{A}}(k)x(k) - \widetilde{\boldsymbol{B}}(k)u(k-1) - \boldsymbol{V}(k)(\nu(k)+\delta_{\mathrm{x}}(k)) - \boldsymbol{P}(k)\triangle\boldsymbol{u}(k) \right\|_{\boldsymbol{M}}^2 + \|\triangle\boldsymbol{u}(k)\|_{\Lambda}^2 + \rho^{\min}(\varepsilon^{\min}(k))^2 + \rho^{\max}(\varepsilon^{\max}(k))^2 \right\}$$

subject to

$$\begin{aligned}
&\boldsymbol{u}^{\min} \le \boldsymbol{J}\triangle\boldsymbol{u}(k) + \boldsymbol{u}(k-1) \le \boldsymbol{u}^{\max}\\
&\triangle\boldsymbol{u}^{\min} \le \triangle\boldsymbol{u}(k) \le \triangle\boldsymbol{u}^{\max}\\
&\boldsymbol{y}_3^{\min} - \varepsilon^{\min}(k)\boldsymbol{I}_{N\times 1} \le \widetilde{\boldsymbol{C}}(k)\Big[\widetilde{\boldsymbol{A}}(k)x(k) + \widetilde{\boldsymbol{B}}(k)u(k-1) + \boldsymbol{V}(k)(\nu(k)+\delta_{\mathrm{x}}(k))\Big]\\
&\qquad + (\widetilde{\boldsymbol{C}}(k)\boldsymbol{P}(k) + \widetilde{\boldsymbol{D}}(k)\boldsymbol{J}^{\mathrm{N}})\triangle\boldsymbol{u}(k) + \widetilde{\boldsymbol{D}}(k)\boldsymbol{u}^{\mathrm{N}}(k-1)\\
&\qquad + (\delta_{\mathrm{y}}(k) + d(k))\boldsymbol{I}_{N\times 1} \le \boldsymbol{y}_3^{\max} + \varepsilon^{\max}(k)\boldsymbol{I}_{N\times 1}\\
&\varepsilon^{\min}(k) \ge 0,\ \ \varepsilon^{\max}(k) \ge 0
\end{aligned} \tag{41}$$

where the vector

$$\boldsymbol{u}(k-1) = \begin{bmatrix} u(k-1)\\ \vdots \\ u(k-1)\end{bmatrix} \tag{42}$$

is of length $3N_{\mathrm{u}}$ and the matrix

$$\boldsymbol{J} = \begin{bmatrix} \boldsymbol{I}_{3\times3} & \boldsymbol{0}_{3\times3} & \dots & \boldsymbol{0}_{3\times3}\\ \boldsymbol{I}_{3\times3} & \boldsymbol{I}_{3\times3} & \dots & \boldsymbol{0}_{3\times3}\\ \vdots & \vdots & \ddots & \vdots\\ \boldsymbol{I}_{3\times3} & \boldsymbol{I}_{3\times3} & \dots & \boldsymbol{I}_{3\times3}\end{bmatrix} \tag{43}$$

is of dimensionality $3N_{\mathrm{u}} \times 3N_{\mathrm{u}}$. When different violations of the predicted output constraints may be adjusted for consecutive sampling instants of the prediction horizon, the general MPC optimisation problem (21) becomes

$$\min_{\substack{\triangle \boldsymbol{u}(k)\\ \boldsymbol{\varepsilon}^{\min}(k)\\ \boldsymbol{\varepsilon}^{\max}(k)}} \left\{ \left\| \boldsymbol{x}^{\mathrm{sp}}(k) - \widetilde{\boldsymbol{A}}(k)x(k) - \widetilde{\boldsymbol{B}}(k)u(k-1) - \boldsymbol{V}(k)(\nu(k)+\delta_{\mathrm{x}}(k)) - \boldsymbol{P}(k)\triangle\boldsymbol{u}(k) \right\|_{\boldsymbol{M}}^2 + \|\triangle\boldsymbol{u}(k)\|_{\Lambda}^2 + \rho^{\min}\left\|\boldsymbol{\varepsilon}^{\min}(k)\right\|^2 + \rho^{\max}\left\|\boldsymbol{\varepsilon}^{\max}(k)\right\|^2 \right\}$$

subject to

$$\begin{aligned}
&\boldsymbol{u}^{\min} \le \boldsymbol{J}\triangle\boldsymbol{u}(k) + \boldsymbol{u}(k-1) \le \boldsymbol{u}^{\max}\\
&\triangle\boldsymbol{u}^{\min} \le \triangle\boldsymbol{u}(k) \le \triangle\boldsymbol{u}^{\max}\\
&\boldsymbol{y}_3^{\min} - \boldsymbol{\varepsilon}^{\min}(k) \le \widetilde{\boldsymbol{C}}(k)\Big[\widetilde{\boldsymbol{A}}(k)x(k) + \widetilde{\boldsymbol{B}}(k)u(k-1) + \boldsymbol{V}(k)(\nu(k)+\delta_{\mathrm{x}}(k))\Big]\\
&\qquad + (\widetilde{\boldsymbol{C}}(k)\boldsymbol{P}(k) + \widetilde{\boldsymbol{D}}(k)\boldsymbol{J}^{\mathrm{N}})\triangle\boldsymbol{u}(k) + \widetilde{\boldsymbol{D}}(k)\boldsymbol{u}^{\mathrm{N}}(k-1)\\
&\qquad + (\delta_{\mathrm{y}}(k) + d(k))\boldsymbol{I}_{N\times 1} \le \boldsymbol{y}_3^{\max} + \boldsymbol{\varepsilon}^{\max}(k)\\
&\boldsymbol{\varepsilon}^{\min}(k) \ge \boldsymbol{0}_{N\times1},\ \ \boldsymbol{\varepsilon}^{\max}(k) \ge \boldsymbol{0}_{N\times1}
\end{aligned} \tag{44}$$

4.4. MPC-SL algorithm summary

The following steps are repeated at each sampling instant k of the MPC-SL algorithm:

1. The output values $y_1(k) = x_1(k)$, $y_2(k) = x_2(k)$, $y_3(k-1)$ are measured, the state variable $\tilde{x}_3(k)$ is estimated using a state observer.
2. A local linear approximation (Eqs. (22a), (22b)) of the nonlinear model (Eqs. (7), (8), (9a), (9b)) of the process is calculated for the current operating point, i.e. the matrices $\boldsymbol{A}(k)$, $\boldsymbol{B}(k)$, $\boldsymbol{C}(k)$, $\boldsymbol{D}(k)$ and are found from Eqs. (26), (27), (28), (29), (30), (31), whereas the quantities $\delta_{\mathrm{x}}(k)$, $\delta_{\mathrm{y}}(k)$ are found from Eq. (24).
3. The matrices $\widetilde{\boldsymbol{A}}(k)$, $\boldsymbol{V}(k)$ and $\widetilde{\boldsymbol{B}}(k)$ are calculated from Eqs. (37), the matrix $\boldsymbol{P}(k)$ is calculated from Eq. (38).
4. The state disturbance vector $\nu(k)$ is estimated from Eq. (33), the output disturbance vector $d(k)$ is calculated from Eq. (35).
5. The quadratic optimisation problem (41) or (44) is solved to calculate the decision variables of the algorithm.
6. The first 3 elements of the calculated sequence $\triangle\boldsymbol{u}(k)$ are applied to the process, i.e. $u_1(k) = \triangle u_1(k|k) + u_1(k-1)$, $u_2(k) = \triangle u_2(k|k) + u_2(k-1)$, $u_3(k) = \triangle u_3(k|k) + u_3(k-1)$.
7. At the next sampling instant ($k := k+1$) the algorithm goes to step 1.

4.5. State estimation

When state-space models are used in MPC of the boiler-turbine process, state estimation is necessary. In this work the extended Kalman filter [34] is used for state estimation because it is characterised by excellent estimation quality and low computational complexity. Because the same state-space model is used for process simulation and in MPC, at the sampling instant k the current value of the output signal $y_3(k)$ is not available for measurement, the most recent available signal comes from the previous sampling instant, $k-1$. Considering the model (12), it is because the output $y_3(k)$ depends on the inputs $u_1(k)$, $u_2(k)$, $u_3(k)$, which are calculated at the sampling instant k after state estimation. That is why the measurement vector is

$$z(k) = \begin{bmatrix} z_1(k)\\ z_2(k)\\ z_3(k)\end{bmatrix} = \begin{bmatrix} y_1(k)\\ y_2(k)\\ y_3(k-1)\end{bmatrix} = \begin{bmatrix} x_1(k)\\ x_2(k)\\ g_3(x_1(k-1), x_3(k-1), u_1(k-1), u_2(k-1), u_3(k-1))\end{bmatrix}$$

In order to deal with delayed measurement, the state augmentation technique discussed in [5] is used. For state estimation, the following augmented model is used

$$x^{\mathrm{a}}(k+1) = f^{\mathrm{a}}(x^{\mathrm{a}}(k), u^{\mathrm{a}}(k)) + w(k) \tag{45a}$$

$$z^{\mathrm{a}}(k) = g^{\mathrm{a}}(x^{\mathrm{a}}(k), u^{\mathrm{a}}(k)) + v(k) \tag{45b}$$

where $w(k)$ and $v(k)$ are the process and observation (measurement) noises, respectively. They are assumed to be zero mean, Gaussian and uncorelated noises with covariance matrices $\boldsymbol{Q}(k) = E([w(k)w^{\mathrm{T}}(k)])$ and $\boldsymbol{R}(k) = E([v(k)v^{\mathrm{T}}(k)])$, respectively. The augmented state and input vectors are

$$x^{\mathrm{a}}(k) = \begin{bmatrix} x_1(k)\\ x_2(k)\\ x_3(k)\\ x_1(k-1)\\ x_2(k-1)\\ x_3(k-1)\end{bmatrix},\quad u^{\mathrm{a}}(k) = \begin{bmatrix} u_1(k)\\ u_2(k)\\ u_3(k)\\ u_1(k-1)\\ u_2(k-1)\\ u_3(k-1)\end{bmatrix}$$

Taking into account Eqs. (12), the augmented state Eq. (45a) becomes

$$x^{\mathrm{a}}(k+1) = f^{\mathrm{a}}(x^{\mathrm{a}}(k), u^{\mathrm{a}}(k)) = \begin{bmatrix} f_1(x_1(k-1), u_1(k-1), u_2(k-1), u_3(k-1))\\ f_2(x_1(k-1), x_2(k-1), u_2(k-1))\\ f_3(x_1(k-1), x_3(k-1), u_2(k-1), u_3(k-1))\\ x_1(k)\\ x_2(k)\\ x_3(k)\end{bmatrix}$$

State estimation requires successive on-line model linearisation

$$\boldsymbol{F}(k-1) = \left.\frac{\partial f(x(k-1), u(k-1))}{\partial x(k-1)}\right|_{x(k-1)=\tilde{x}(k-1|k-1)}$$

$$\boldsymbol{H}(k) = \left.\frac{\partial g(x(k), u(k))}{\partial x(k)}\right|_{x(k)=\tilde{x}(k|k-1)}$$

One may notice that linearisation for the MPC-SL algorithm and for state estimation is performed in a very similar way. The matrix $\boldsymbol{F}(k-1) = \boldsymbol{A}(k)$ is calculated from Eq. (26). The matrix $\boldsymbol{H}(k-1) = \boldsymbol{C}(k)$ is calculated from Eqs. (28), (29) and (30), but for calculation of the matrix $\boldsymbol{H}(k-1)$ the operating point is defined by the quantities $x_1(k-1)$, $x_2(k-1)$ and $\tilde{x}_3(k-1)$ whereas for model linearisation in the MPC-SL algorithm by the quantities $x_1(k)$, $x_2(k)$ and $\tilde{x}_3(k)$. When constant matrices $\boldsymbol{F}$ and $\boldsymbol{H}$ are used, one obtains the Kalman filter for linear systems. The matrices of the linearised model with the augmented state are easily calculated from the matrices which describe a linear approximation of the rudimentary model used in the MPC-SL algorithm:

$$\boldsymbol{F}^{\mathrm{a}}(k) = \begin{bmatrix} \boldsymbol{A}(k) & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} \end{bmatrix}, \quad \boldsymbol{H}^{\mathrm{a}}(k) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \boldsymbol{H}_{3,1}(k-1) & \boldsymbol{H}_{3,1}(k-1) & \boldsymbol{H}_{3,1}(k-1) \end{bmatrix}$$

The following steps are repeated for state estimation at each sampling instant k (before execution of the MPC-SL algorithm)

1. The predicted state estimate is found from $\tilde{x}(k|k-1) = f(\tilde{x}(k-1|k-1), u(k-1))$.
2. The predicted covariance estimate is calculated from $\boldsymbol{P}(k|k-1) = \boldsymbol{F}^{\mathrm{a}}(k-1)\boldsymbol{P}(k-1|k-1)(\boldsymbol{F}^{\mathrm{a}}(k-1))^{\mathrm{T}} + \boldsymbol{Q}(k-1)$.
3. The measurement residual is determined from $\tilde{z}(k) = z(k) - g(\tilde{x}(k|k-1), u(k-1))$.
4. The residual covariance is found from $\boldsymbol{S}(k) = \boldsymbol{H}^{\mathrm{a}}(k)\boldsymbol{P}(k|k-1)(\boldsymbol{H}^{\mathrm{a}}(k))^{\mathrm{T}} + \boldsymbol{R}(k)$.
5. The filter gain is calculated from $\boldsymbol{K}(k) = \boldsymbol{P}(k|k-1)(\boldsymbol{H}^{\mathrm{a}}(k))^{\mathrm{T}}\boldsymbol{S}^{-1}(k)$.
6. The state estimate is updated from $\tilde{x}(k|k) = \tilde{x}(k|k-1) + \boldsymbol{K}(k)\tilde{z}(k)$.
7. The covariance estimate is updated from $\boldsymbol{P}(k|k) = (\boldsymbol{I} - \boldsymbol{K}(k)\boldsymbol{H}^{\mathrm{a}}(k))\boldsymbol{P}(k|k-1)$.

5. Simulation results

The discrete-time dynamic model described by Eqs. (7), (8), (9a) and (9b) is used for simulation of the process (i.e. as the simulated process) and in MPC (for on-line linearisation and prediction). All simulations are carried out in Matlab. Parameters of all algorithms are the same: $\mu_{1,p} = 30$, $\mu_{2,p} = 1$, $\mu_{3,p} = 0$ for $p = 1, \ldots, N$, $\lambda_{n,p} = 1$ for $n = 1, 2, 3$ and $p = 0, \ldots, N_{\mathrm{u}} - 1$, $\rho^{\min} = \rho^{\max} = 10^6$. The extended Kalman filter has the following parameters: $\boldsymbol{P}(1|0) = \boldsymbol{I}_{6\times6}$, $\boldsymbol{Q} = 100\boldsymbol{I}_{6\times6}$, $\boldsymbol{R} = 0.0001\boldsymbol{I}_{3\times3}$.

Different operating points of the process are detailed in [1]. For further simulations the operating points #1, #3 and #7 are chosen. Considering control objectives described in Section 2, one may easily check that the output constraints (4) are not satisfied at the operating points described in [1]. That is why three modified operating points are used in this study, the values of state, input and output variables of these points are given in Table 1. The values of the first two state variables are the same as in [1], but different values of the third state variable and different values of process inputs are necessary to guarantee that the output constraints are satisfied. The chosen operating conditions are named operating points A, B and C, respectively.

Table 1
Operating points of the boiler-turbine.

Variable	Operating point A	Operating point B	Operating point C
$x_1 = y_1$	7.56×10^1	9.72×10^1	1.4×10^2
$x_2 = y_2$	1.53×10^1	5.05×10^1	1.28×10^2
x_3	5.0897×10^2	4.6951×10^2	3.2368×10^2
u_1	1.1926×10^{-1}	2.7049×10^{-1}	5.9589×10^{-1}
u_2	3.8063×10^{-1}	6.2082×10^{-1}	8.9447×10^{-1}
u_3	1.2262×10^{-1}	3.3979×10^{-1}	7.8829×10^{-1}
y_3	9.8414×10^{-2}	9.7038×10^{-2}	9.7970×10^{-2}

5.1. Model simulations

At first it is interesting to compare effectiveness of successive model linearisation. Fig. 1 compares state and output trajectories caused by step changes of the manipulated variables u_1, u_2, u_3 by $\delta = \pm 0.1, \pm 0.2, \pm 0.3$ occurring at $k=100$ of two models: the full nonlinear model and the successively linearised one are considered (both in the open-loop mode, i.e. with no controller). The initial operating point is B. From the presented simulations two conclusions may be draw. Firstly, the successively linearised model gives practically the same trajectories as the full nonlinear model, which means that local linearisation may be efficiently used for model approximation. Secondly, it is necessary to notice that the responses for positive and negative changes in the manipulated variables have different shapes and final values, which means that both dynamic and steady-state properties of the process are nonlinear. It motivates development of nonlinear MPC algorithms.

5.2. Selection of horizons

Selection of proper prediction and control horizons in MPC is an important task. On the one hand, the horizons must be long enough to give good control quality, on the other hand, they must be short to minimise computational complexity. It is an interesting issue to demonstrate that for the considered process it is sufficient to use quite short prediction and control horizons in MPC. In this work the horizons are determined experimentally, i.e. simulations are carried out for different horizons and quality of control is assessed. For this purpose two Sum of Squared Errors (SSE) performance indices are used. They are calculated after completing simulations of MPC in order to compare the influence of the tuning parameters. The first performance index measures control errors of the first two state variables

$$\mathrm{SSE_x} = \sum_{n=1}^{2} \sum_{p=1}^{N_{\mathrm{sim}}} (x_n^{\mathrm{sp}}(k) - x_n(k))^2 \tag{46}$$

where $x_n^{\mathrm{sp}}(k)$ denotes the state set-point whereas $x_n(k)$ is the actual value of the corresponding state variable calculated during simulations, N_{sim} is the simulation horizon. It is necessary to point out that the $\mathrm{SSE_x}$ index is actually not minimised in MPC. The second index measures the degree of hard output constraints (Eq. (15)) violation

$$\mathrm{SSE_y} = \sum_{p=1}^{N_{\mathrm{sim}}} (0.1 - \mathrm{abs}(y_3(k)))^2 \,\mathrm{sgn}(\max(\mathrm{abs}(y3) - 0.1, 0)) \tag{47}$$

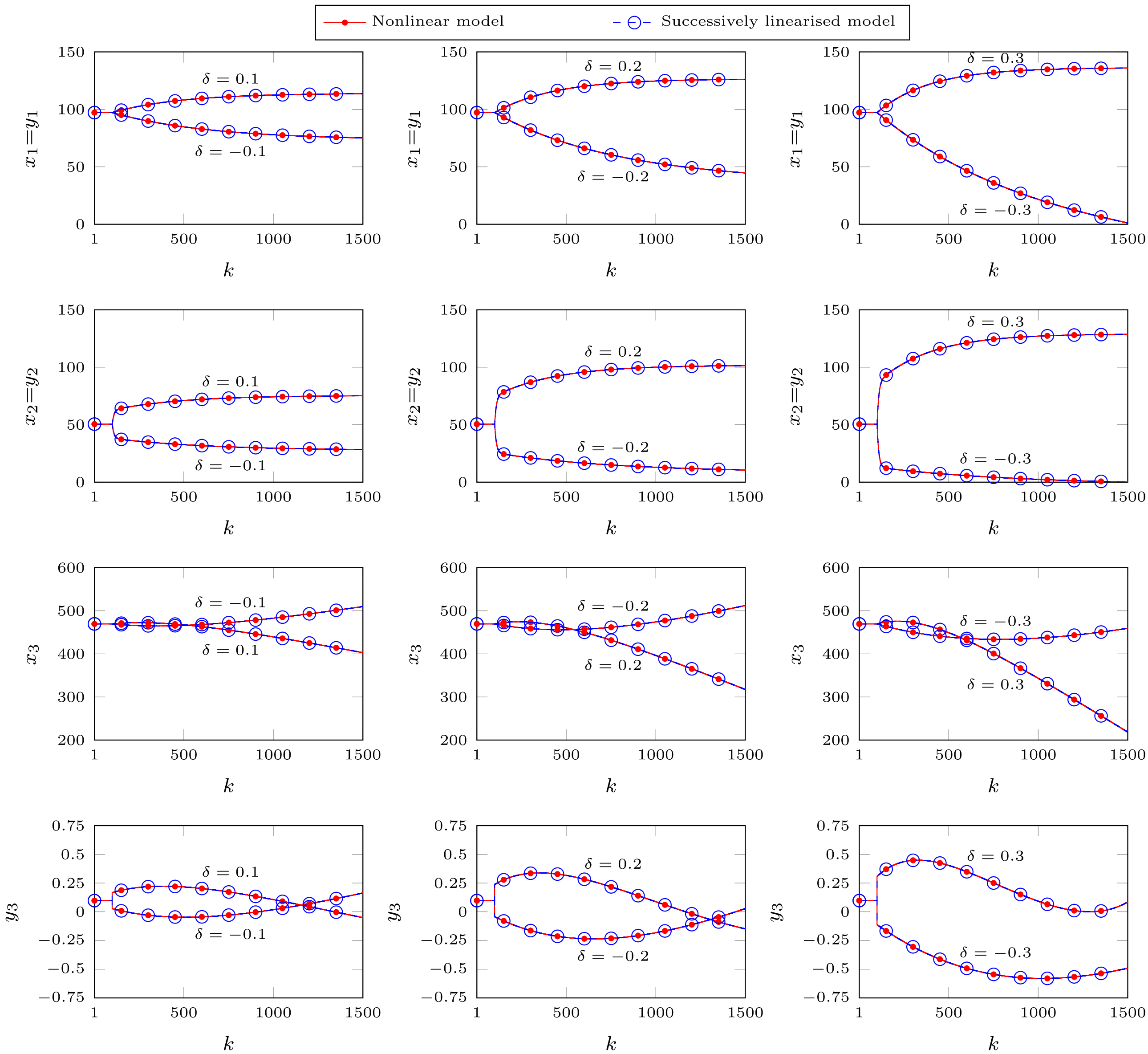

Fig. 1. State and output trajectories caused by step changes of the manipulated variables u_1, u_2, u_3 by $\delta = \pm 0.1, \pm 0.2, \pm 0.3$ at $k=100$: the nonlinear model (*solid line with dots*) vs. the successively linearised model (*dashed line with circles*).

Firstly, the influence of the prediction horizon on control quality and constraint satisfaction is assessed. Table 2 shows accuracy of the MPC-SL algorithm in terms of the performance indices $\mathrm{SSE_x}$ and $\mathrm{SSE_y}$ for different prediction horizons for set-point change from the operating point B to C (for different set-point changes the conclusions are the same). A sufficiently long control horizon $N_u = 5$ is used (its choice is explained next), real state measurements are used (i.e. no observer is necessary), no constraints imposed on the rate of change of the manipulated variables are taken into account, the output constraints are softened by means of two additional variables $\varepsilon^{\min}(k)$ and $\varepsilon^{\max}(k)$, as in the MPC-SL optimisation problem (19). Fig. 2 depicts selected simulation results, for prediction horizons $N=5$, $N=10$ and $N=50$. The best control quality is obtained for the shortest horizon $N=5$, but for the steady-state, starting from the sampling instant $k=200$,

Table 2
Accuracy of the MPC-SL algorithm in terms of the performance indices $\mathrm{SSE_x}$ and $\mathrm{SSE_y}$ for different prediction horizons N for set-point change from the operating point B to C; $N_u = 5$ (real state measurements are used, no constraints imposed on the rate of change of the manipulated variables are taken into account).

N	$\mathrm{SSE_x}$	$\mathrm{SSE_y}$
5	1.0858×10^5	7.7740
10	1.0918×10^5	7.6435
20	1.0957×10^5	7.7306
30	1.0959×10^5	7.7328
40	1.3034×10^5	7.7352
50	1.5222×10^5	8.0571

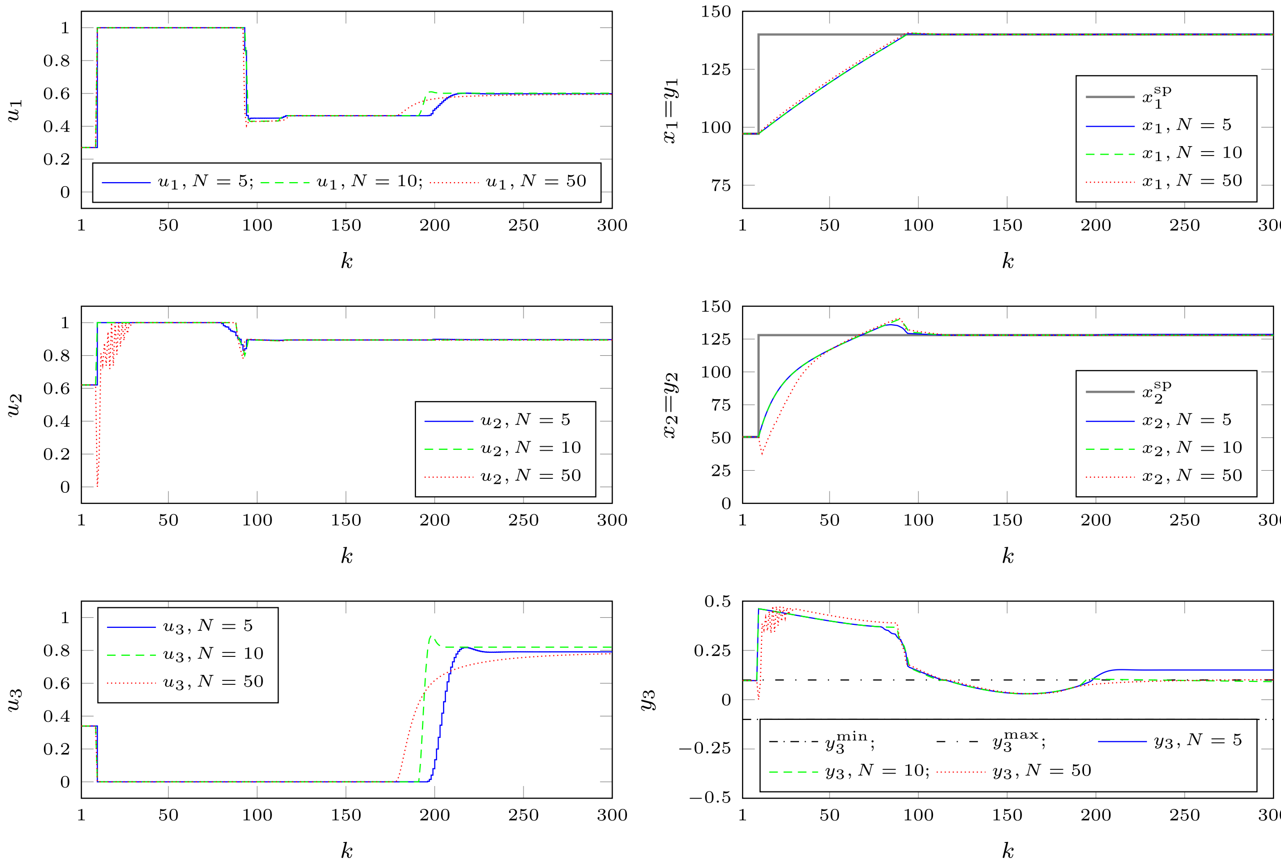

Fig. 2. Simulation results: the nonlinear MPC-SL algorithm for set-point change from the operating point B to C for different prediction horizons: $N=5$ (*solid line*), $N=10$ (*dashed line*), $N=50$ (*dotted line*); $N_u=5$ (real state measurements are used, no constraints imposed on the rate of change of the manipulated variables are taken into account).

the algorithm is unable to satisfy the constraints imposed on the output variable y_3. The best constraint satisfaction is possible for $N=10$, moderate lengthening the horizon results in practically the same performance, but for very long horizons, e.g. $N=50$, the quality deteriorates. It is because for long-range prediction the same linear approximation of the model is used and for long horizons a significant discrepancy between the prediction calculated by a locally linearised model and the real process trajectory is possible. Because for the prediction horizons $10<N<45$ there is no important difference in quality of control and constraint satisfaction, the quite short horizon $N=10$ is finally chosen. Although the very short horizons, e.g. $N=5$, are also possible, they are not recommended as the MPC algorithm with a too short horizon may only work in the ideal situation with no disturbances, modelling errors and noise.

Secondly, the influence of the control horizon on control quality and constraint satisfaction is assessed. Table 3 shows accuracy of the MPC-SL algorithm in terms of the performance indices $\mathrm{SSE_x}$ and $\mathrm{SSE_y}$ for different control horizons for set-point change from the operating point B to C. The prediction horizon is fixed, $N=10$, real state measurements are used, no constraints imposed on the rate of change of the manipulated variables are taken into account. Fig. 3 depicts selected simulation results, for control horizons $N_u=1$, $N_u=2$ and $N_u=5$. For the shortest horizon $N_u=1$ the trajectories are completely different from those obtained for $N_u\geq 2$. It is interesting to notice that increasing the length of the control horizon (i.e. for $N_u>2$) does not lead to any improvement of control performance and control satisfaction ability. That is why, in order to reduce computational complexity of MPC, the control horizon is fixed to $N_u=2$. Taking into account the simulations discussed so far, in the following part of the paper the horizons are: $N=10$, $N_u=2$.

Table 3
Accuracy of the MPC-SL algorithm in terms of the performance indicies $\mathrm{SSE_x}$ and $\mathrm{SSE_y}$ for different control horizons N_u for set-point change from the operating point B to C; $N=10$ (real state measurements are used, no constraints imposed on the rate of change of the manipulated variables are taken into account).

N_u	$\mathrm{SSE_x}$	$\mathrm{SSE_y}$
1	7.566243×10^5	0.5865
2	1.089781×10^5	7.6243
3	1.090982×10^5	7.6390
4	1.091857×10^5	7.6457
5	1.091822×10^5	7.6435
10	1.091564×10^5	7.6556

5.3. Classical MPC based on linear models

Poor performance of classical linear MPC algorithms applied to the considered boiler-turbine process is demonstrated elsewhere, e.g. simulation results for the DMC algorithm are given in [26]. Nevertheless, it is interesting to evaluate the linear

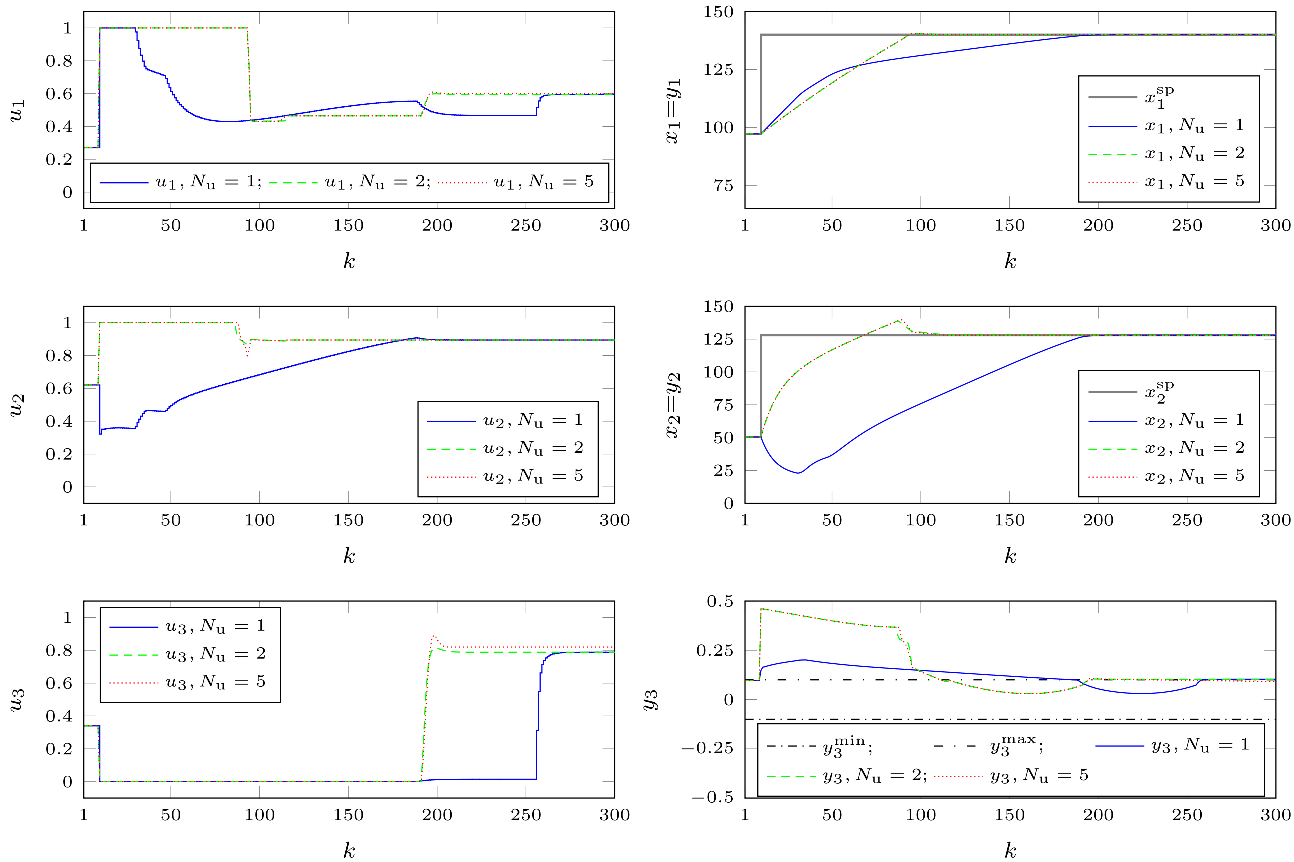

Fig. 3. Simulation results: the nonlinear MPC-SL algorithm for set-point change from the operating point B to C for different control horizons: $N_u = 1$ (*solid line*), $N_u = 2$ (*dashed line*), $N_u = 5$ (*dotted line*); $N=10$ (real state measurements are used, no constraints imposed on the rate of change of the manipulated variables are taken into account).

state-space MPC algorithm with the output constraints. Satisfaction of constraints imposed on the values and on the rate of change of the manipulated variables is quite simple as such constraints do not lead to an empty feasible set. Because a linear model is only a very rough approximation of the process, when it is used for prediction of the output y_3, serious problems are likely. The MPC algorithm based on linear models described in [38] is used, it uses the same disturbance models as the MPC-SL algorithm described in this work, but it is entirely based on parameter-constant linear models. For the chosen operating points (Table 1), using Eqs. (26), (27), (28), (29), (30), (31), the matrices of the linear model are easily calculated off-line. For the operating point A one obtains the matrices

$$\boldsymbol{A} = \begin{bmatrix} 9.9868 \times 10^{-1} & 0 & 0 \\ 2.2768 \times 10^{-2} & 0.9 & 0 \\ -2.6904 \times 10^{-3} & 0 & 1 \end{bmatrix},$$

$$\boldsymbol{B} = \begin{bmatrix} 0.9 & -2.3367 \times 10^{-1} & -0.15 \\ 0 & 9.4768 & 0 \\ 0 & -9.7836 \times 10^{-1} & 1.6588 \end{bmatrix},$$

$$\boldsymbol{C} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 2.8951 \times 10^{-3} & 0 & 5.8251 \times 10^{-3} \end{bmatrix},$$

$$\boldsymbol{D} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 2.5328 \times 10^{-1} & 3.5868 \times 10^{-1} & -1.3967 \times 10^{-2} \end{bmatrix}$$

for the point B

$$\boldsymbol{A} = \begin{bmatrix} 9.9777 \times 10^{-1} & 0 & 0 \\ 5.8449 \times 10^{-2} & 0.9 & 0 \\ -5.7989 \times 10^{-3} & 0 & 1 \end{bmatrix},$$

$$\boldsymbol{B} = \begin{bmatrix} 0.9 & -3.1002 \times 10^{-1} & -0.15 \\ 0 & 1.2573 \times 10^{1} & 0 \\ 0 & -1.2579 & 1.658823 \end{bmatrix},$$

$$\boldsymbol{C} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 4.927322 \times 10^{-3} & 0 & 5.250236 \times 10^{-3} \end{bmatrix},$$

$$\boldsymbol{D} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 2.53278 \times 10^{-1} & 4.6116 \times 10^{-1} & -1.3967 \times 10^{-2} \end{bmatrix}$$

and for the point C

$$\boldsymbol{A} = \begin{bmatrix} 9.9664 \times 10^{-1} & 0 & 0 \\ 1.0286 \times 10^{-1} & 0.9 & 0 \\ -9.3402 \times 10^{-3} & 0 & 1 \end{bmatrix},$$

$$\boldsymbol{B} = \begin{bmatrix} 0.9 & -4.6738 \times 10^{-1} & -0.15 \\ 0 & 1.8955 \times 10^{1} & 0 \\ 0 & -1.8118 \times 10^{-1} & 1.6588 \end{bmatrix},$$

$$\boldsymbol{C} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1.800770 \times 10^{-2} & 0 & -2.584602 \times 10^{-3} \end{bmatrix},$$

$$\boldsymbol{D} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 2.5328 \times 10^{-1} & 6.6422 \times 10^{-1} & -1.3967 \times 10^{-2} \end{bmatrix}$$

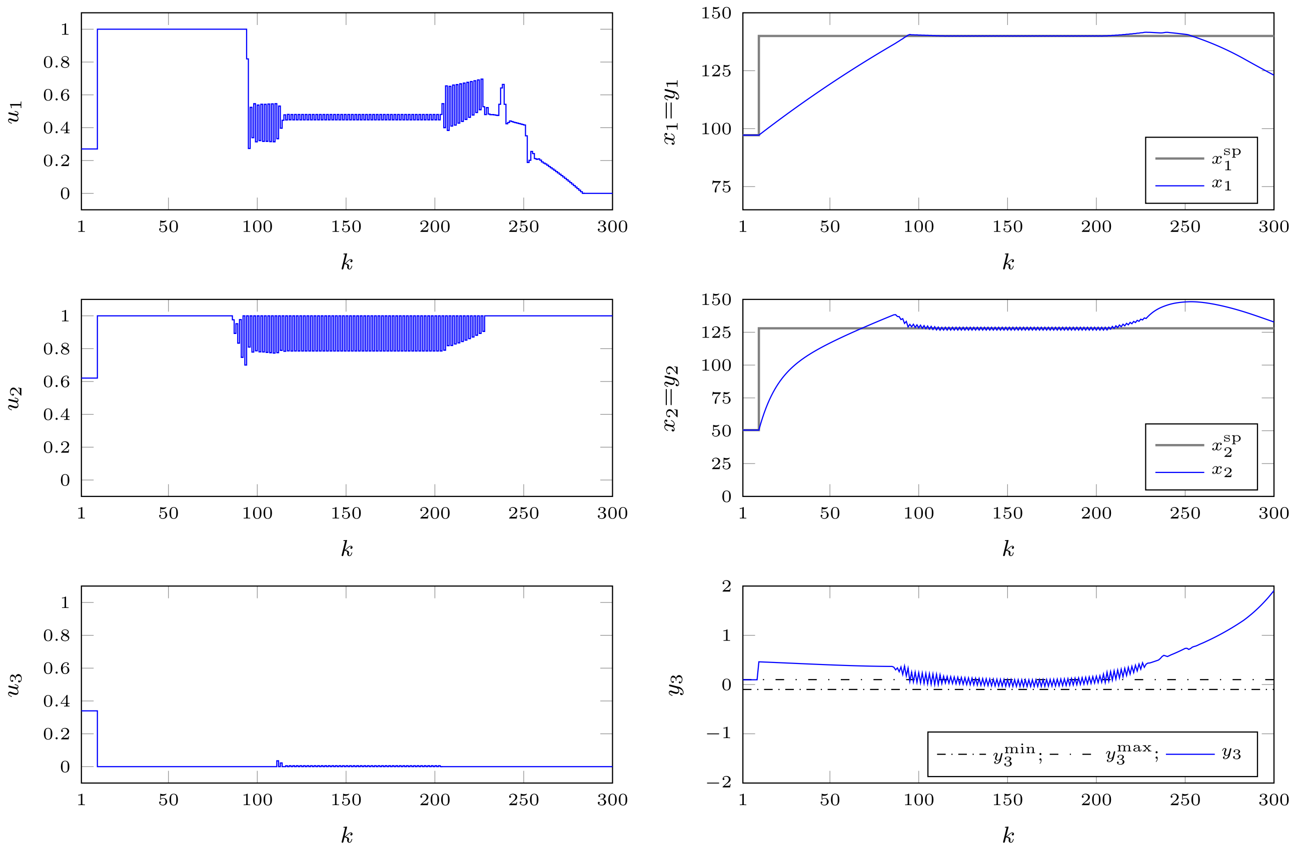

Fig. 4. Simulation results: the linear MPC algorithm for set-point change from the operating point B to C (the linear model for the operating point B is used, real state measurements are used, no constraints imposed on the rate of change of the manipulated variables are taken into account).

Fig. 4 depicts simulation results of the linear MPC algorithm for set-point change from the operating point B to C when for prediction the linear model for the operating point B is used. For simplicity real state measurements are used and no constraints imposed on the rate of change of the manipulated variables are taken into account. Unfortunately, due to modelling errors the algorithm is unable to guarantee that the state variables x_1 and x_2 follow precisely their set-points x_1^{sp} and x_2^{sp}, respectively. Furthermore, the output variable y_3 does not satisfy its constraints. Unwanted oscillations may be eliminated by increasing the penalty coefficients $\lambda_{n,p}$ from their default value 1 to 100, but it does not lead to any better set-point tracking and satisfaction of the output constraints. Fig. 5 depicts simulation results of the linear MPC algorithm when the linear model for the operating point C is used with the same simplifications. When the output constraints imposed on the variable y_3 are taken into account, the algorithm is unable to provide good state set-point tracking (in particular of the variable x_2) and the output constrains are not satisfied. Better state set-point tracking may be achieved by removing the output constraints from the MPC optimisation problem, but they are crucial from the technological point of view. The same problems are encountered when the set-point is changed from the operating point A to C. For example, Fig. 6 shows simulation results of the linear MPC algorithm based on the linear model for the operating point C. When compared to Fig. 5, due to big set-point change, input, state and output variables are characterised by bigger damped oscillations. All things considered, the MPC algorithm based on linear models are unable to provide offset-free control and satisfaction of the output constraints.

5.4. Nonlinear MPC based on nonlinear model

The next simulations are concerned with two nonlinear MPC strategies:

a) the MPC scheme with Nonlinear Optimisation (MPC-NO) repeated at each sampling instant, in which the full nonlinear model of the process is used (for optimisation the Sequential Quadratic Programming (SQP) method is used [23]),
b) the discussed MPC-SL scheme with on-line model linearisation, which means that a quadratic optimisation problem is solved at each sampling instant (for optimisation the active set method is used [23]).

Both algorithms use the same model of the process, but in a different way. The figures presented next show the trajectories of MPC algorithms in which the output constraints are softened by means of two additional variables $\varepsilon^{\min}(k)$ and $\varepsilon^{\max}(k)$, as in the MPC-SL optimisation problem (19). Fig. 7 depicts simulation results of MPC-NO and MPC-SL algorithms for set-point change from the operating point B to C. For simplicity the same assumptions as in the case of the linear MPC algorithm are used, i.e. real state measurements are used and no constraints imposed on the rate of change of the manipulated variables are taken into account. Unlike the results of the linear MPC algorithm (Figs. 4 and 5), the state set-points are tracked precisely, with no steady-state error, there are no oscillations. Furthermore, the constraints of the output variable y_3 are satisfied in some 2/3 period of the simulation. In the first 1/3 period of simulation, when the state operating point changes quickly, the hard output constraints are not satisfied, but it is necessary to enforce existence of a feasible set of the MPC optimisation task (i.e. the additional variables $\varepsilon^{\min}(k) > 0$

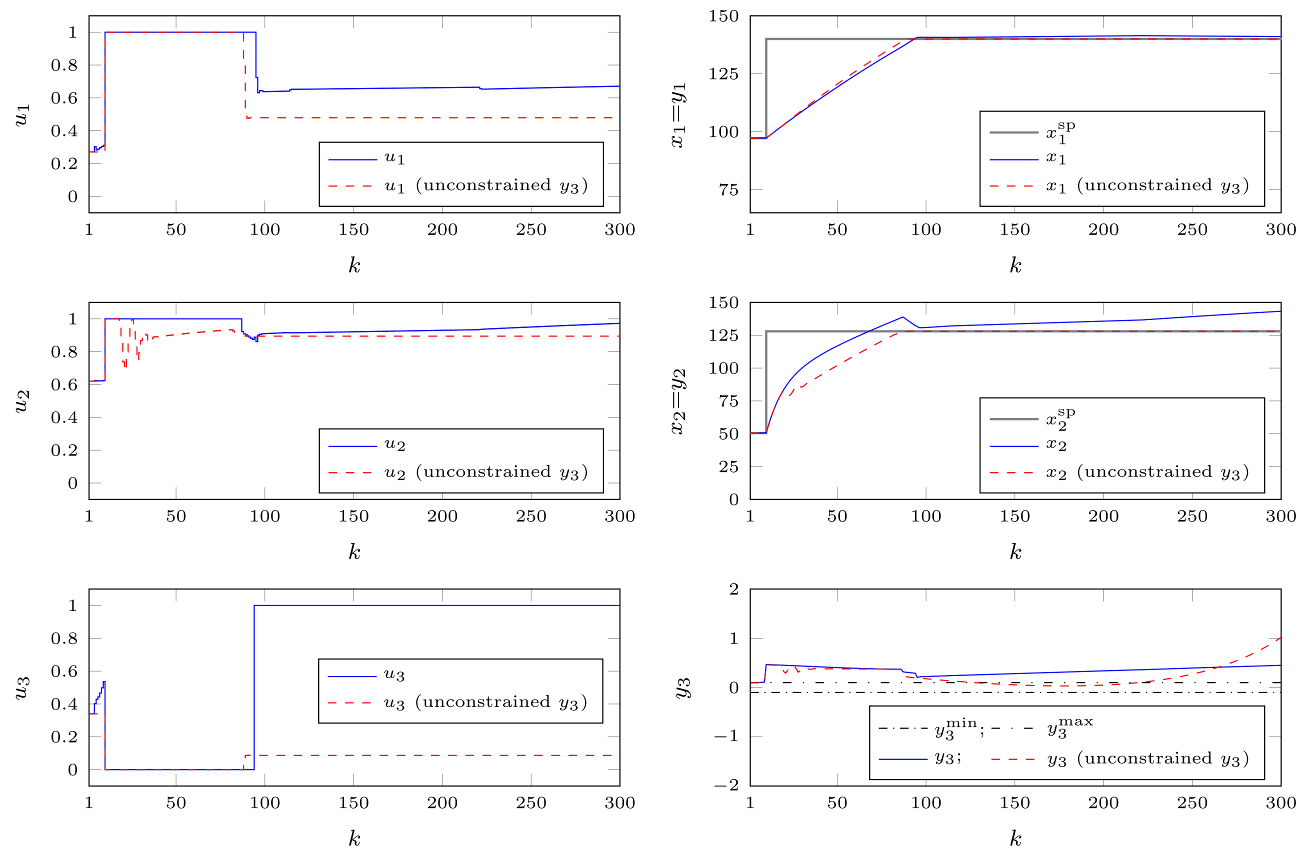

Fig. 5. Simulation results: the linear MPC algorithm (*solid line*) and the linear MPC algorithm when the predicted output variable y_3 is not constrained (*dashed line*) for set-point change from the operating point B to C (the linear model for the operating point C is used, real state measurements are used, no constraints imposed on the rate of change of the manipulated variables are taken into account).

and $\varepsilon^{\min}(k) > 0$). It is necessary to point out the fact that the MPC-SL algorithm with on-line model linearisation and nonlinear optimisation gives practically the same trajectories as the computationally demanding "ideal" MPC-NO algorithm with nonlinear optimisation. Fig. 8 compares trajectories of both nonlinear MPC control schemes for set-point change from the operating point A to C. Unlike the linear algorithm (Fig. 6), they give good set-point tracking and satisfaction of the output constraints. Moreover, the trajectories of the MPC-SL algorithm are the same as in the MPC-NO one. It is interesting to notice that for some portion of the simulations the optimal values of the manipulated variables reach their magnitude constraints which is necessary for fast transition from the initial to the final operating point.

Figs. 9 and 10 compare MPC-NO and MPC-SL algorithms for set-point changes from the operating point B to C and from the point A to C, respectively, but the constraints imposed on the rate of change of the manipulated variables are additionally taken into account. For simplicity real state measurements are still used. In comparison with Figs. 7 and 8 the state trajectories are significantly slower, but it is caused by quite slow changes of the input trajectories, which is enforced by the input rate constraints. Also in the case of the additional rate constraints the MPC-SL algorithm gives trajectories very similar to those possible in the MPC-NO one. Comparing the situation when no rate constraints of the manipulated variables are taken into account, i.e. Figs. 7 and 8, introduction of such constraints reduces the time when the manipulated variables reach their magnitude constraints.

In all the simulation presented so far real state (undisturbed) measurements are used. In reality state estimation of the variable x_3 is necessary. For this purpose the Extended Kalman Filter described in Section 4.5 is used. Figs. 11 and 12 compare MPC-NO and MPC-SL algorithms for set-point changes from the operating point B to C and from the point A to C, respectively. All the constraints imposed on the inputs, on the rate of change of the inputs and on the output y_3 are taken into account. It is assumed that the initial condition of the process is not known, the initial state of the filter is $\tilde{x} = \left[50\ 25\ 300\right]^{\mathrm{T}}$. Additionally, the process is affected by unmeasured step disturbances which act on its inputs: at the sampling instant $k=250$ the step applied to the first input changes from 0 to 0.3, at the instant $k=350$ the step applied to the second input changes from 0 to 0.2. Moreover, the process outputs y_1, y_2 and y_3 are affected by noise with normal distribution with zero mean and standard deviation 0.01, 0.01 and 0.001, respectively. The simulation horizon is lengthen. It is evident that the nonlinear MPC algorithms are able co compensate for disturbances and noise, the wrong initial condition of the state observer does not lead to any significant deterioration of set-point tracking and constraints satisfaction. Also in this case the MPC-SL algorithm with quadratic optimisation gives the trajectories almost the same as the MPC-NO one.

Fig. 13 compares real and estimated state trajectories in the nonlinear MPC-SL algorithm for two considered set-point changes and state estimation error. Because the initial condition of the process is not known precisely, the observer needs a few (6-7) time steps to find the real state with quite a small error. The error increases when the process is affected by the unmeasured input step disturbances (in particular the second one, which starts for $k = 350$).

Finally, the two nonlinear MPC algorithms are compared in terms of the performance indicies $\mathrm{SSE_x}$ (Eq. (46)) and $\mathrm{SSE_y}$ (Eq. (47)). Their values are given in Tables 4 and 5 for different set-point changes and configurations: with or without the constraints

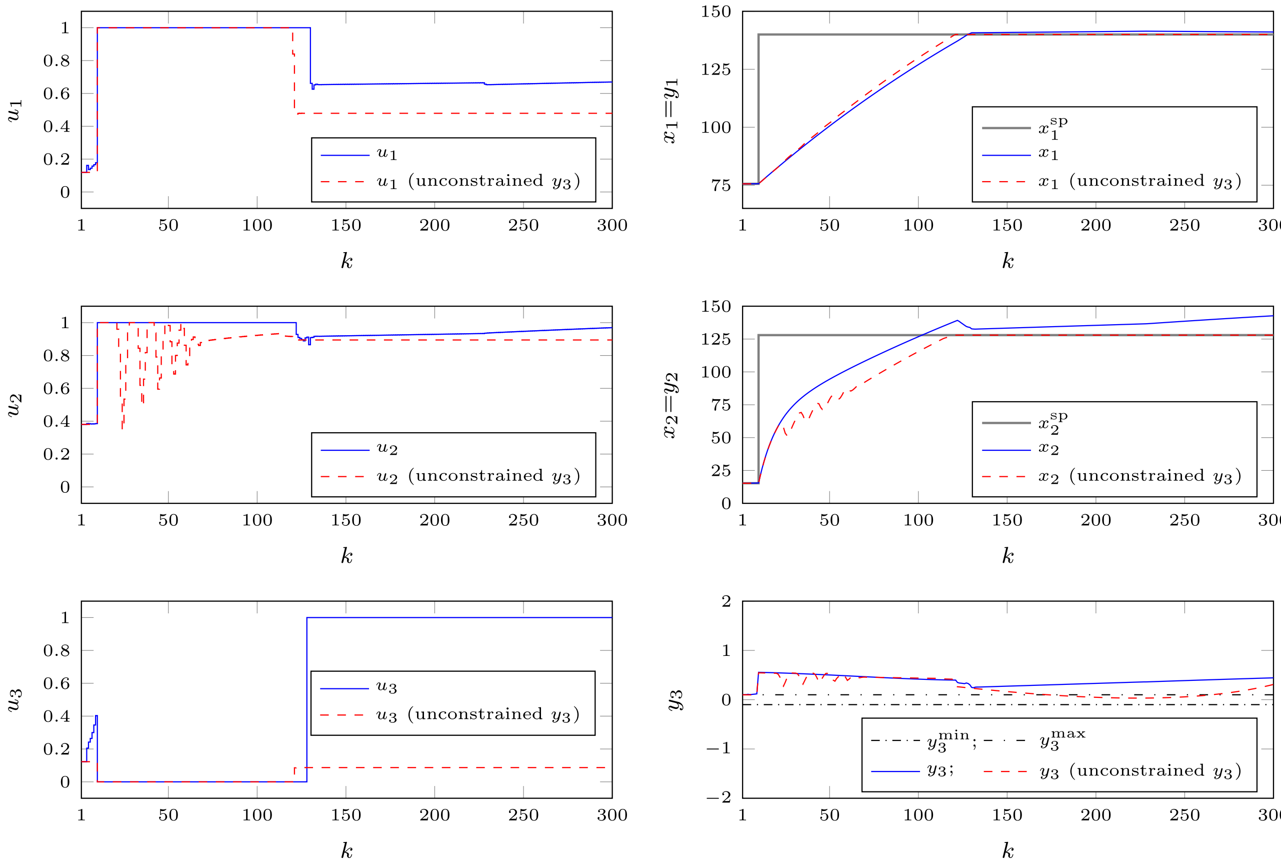

Fig. 6. Simulation results: the linear MPC algorithm (*solid line*) and the linear MPC algorithm when the predicted output variable y_3 is not constrained (*dashed line*) for set-point change from the operating point A to C (the linear model for the operating point C is used, real state measurements are used, no constraints imposed on the rate of change of the manipulated variables are taken into account).

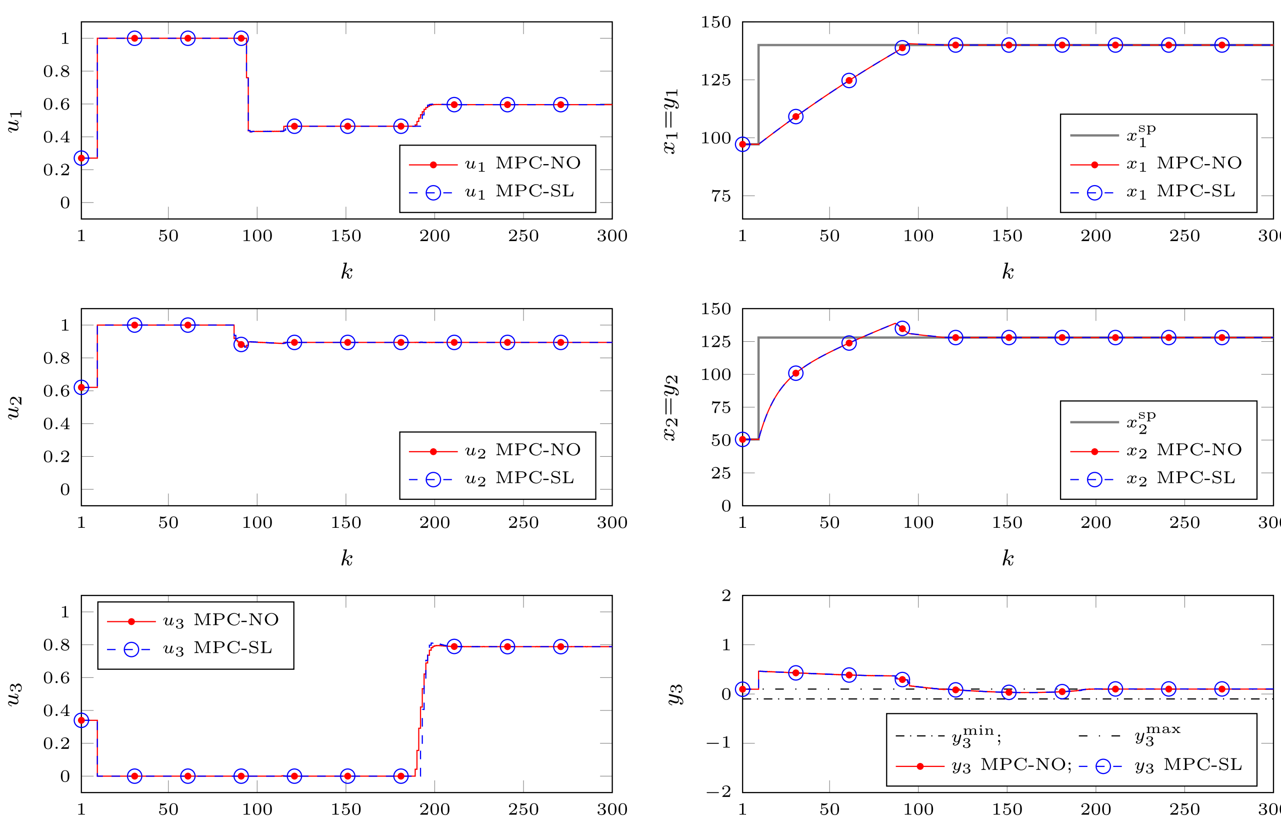

Fig. 7. Simulation results: the nonlinear MPC-NO algorithm (*solid line with dots*) and the nonlinear MPC-SL algorithm (*dashed line with circles*) for set-point change from the operating point B to C (real state measurements are used, no constraints imposed on the rate of change of the manipulated variables are taken into account).

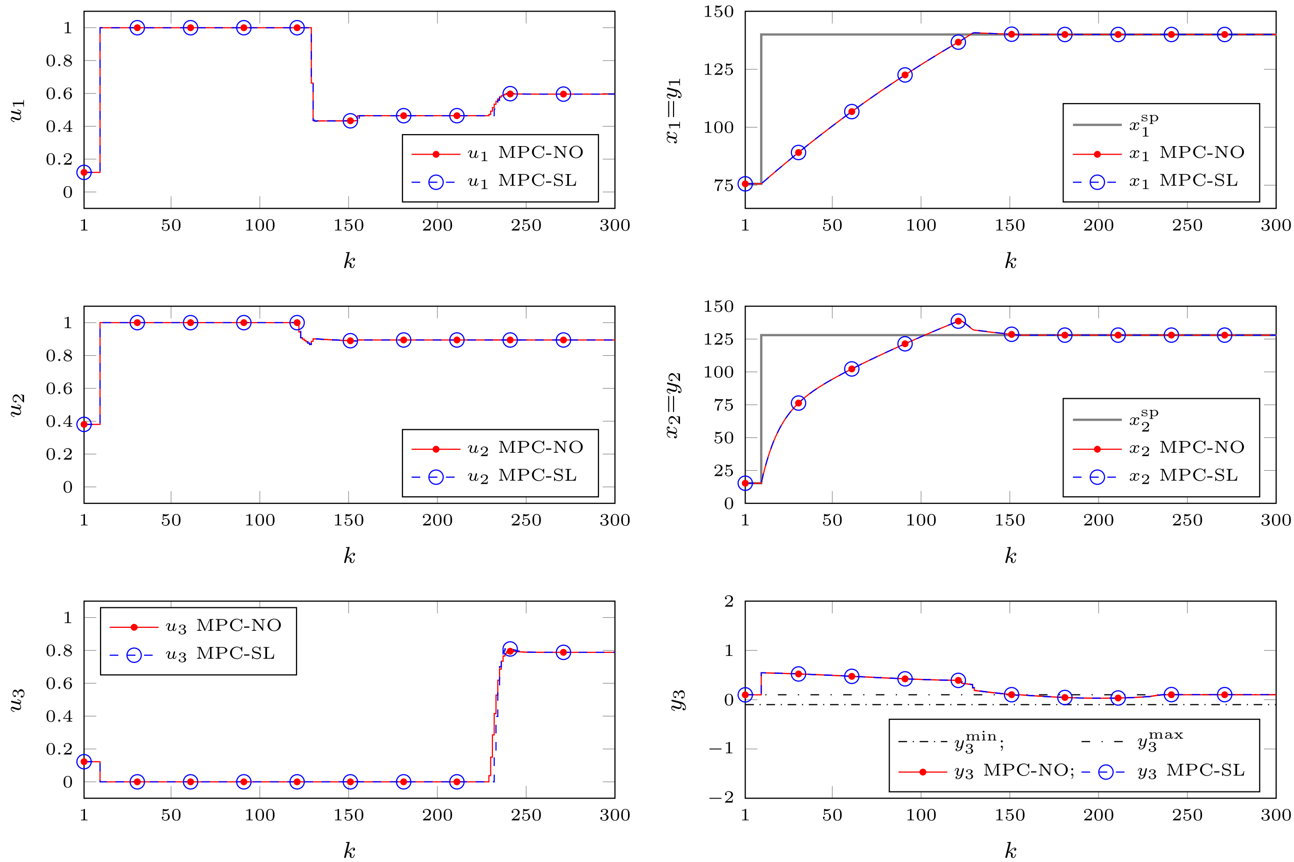

Fig. 8. Simulation results: the nonlinear MPC-NO algorithm (*solid line with dots*) and the nonlinear MPC-SL algorithm (*dashed line with circles*) for set-point change from the operating point A to C (real state measurements are used, no constraints imposed on the rate of change of the manipulated variables are taken into account).

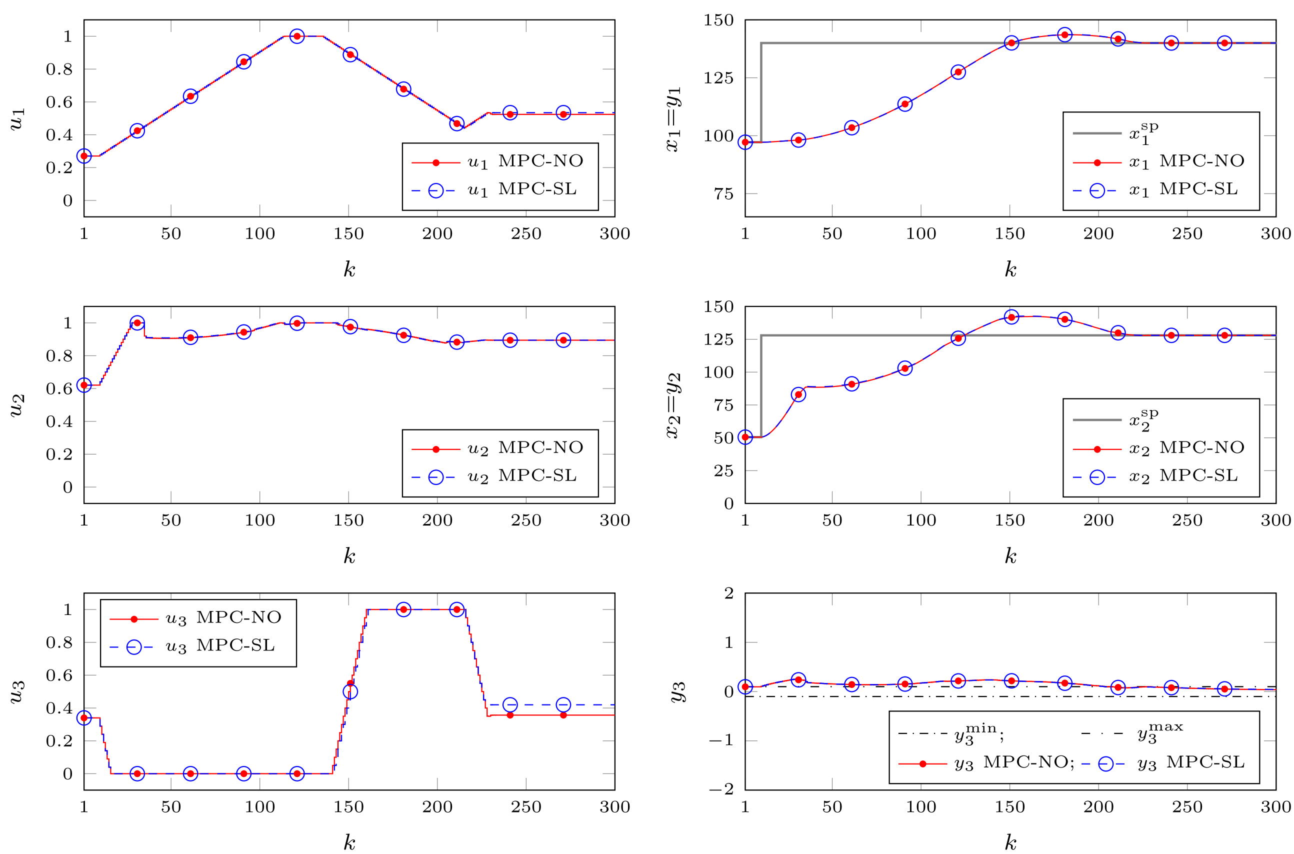

Fig. 9. Simulation results: the nonlinear MPC-NO algorithm (*solid line with dots*) and the nonlinear MPC-SL algorithm (*dashed line with circles*) for set-point change from the operating point B to C (real state measurements are used, the constraints imposed on the rate of change of the manipulated variables are taken into account).

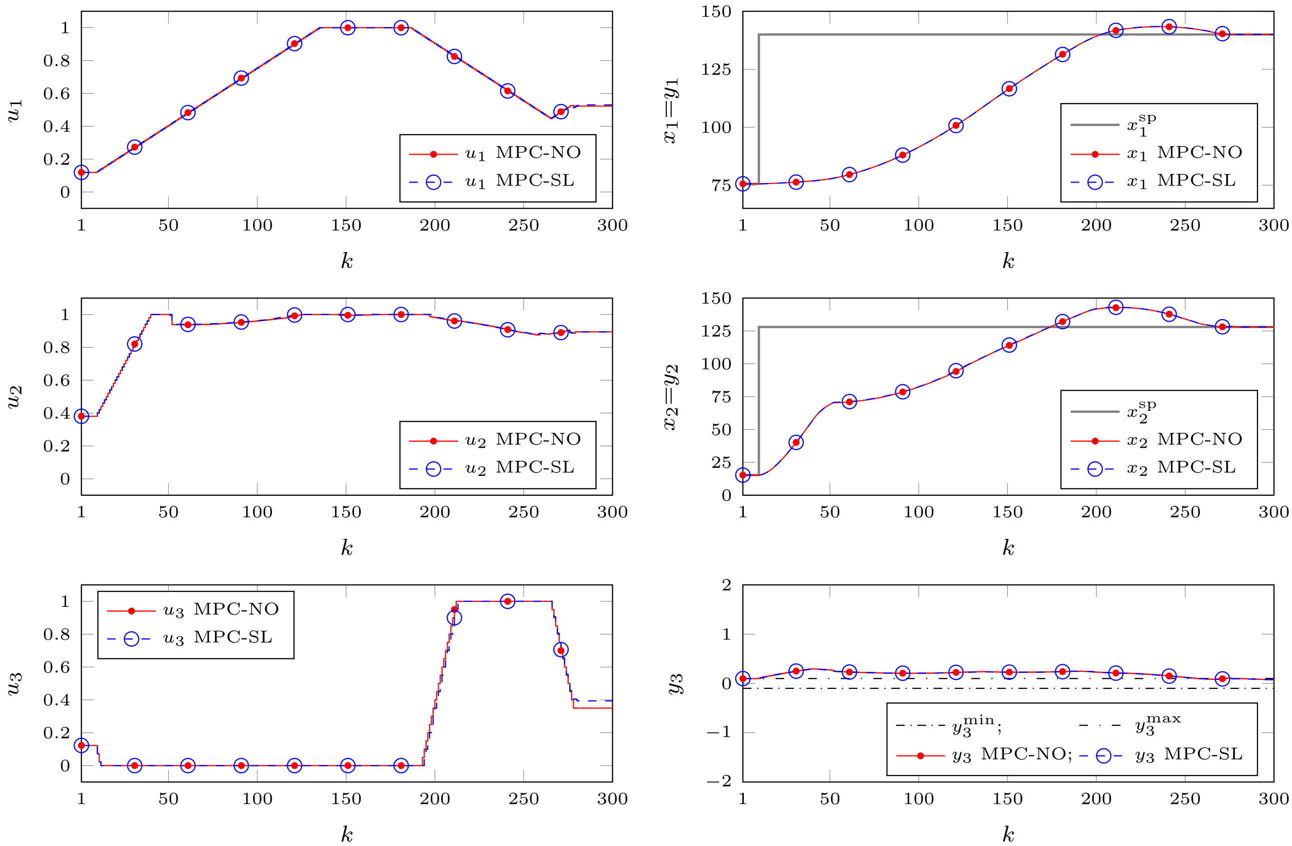

Fig. 10. Simulation results: the nonlinear MPC-NO algorithm (*solid line with dots*) and the nonlinear MPC-SL algorithm (*dashed line with circles*) for set-point change from the operating point A to C (real state measurements are used, the constraints imposed on the rate of change of the manipulated variables are taken into account).

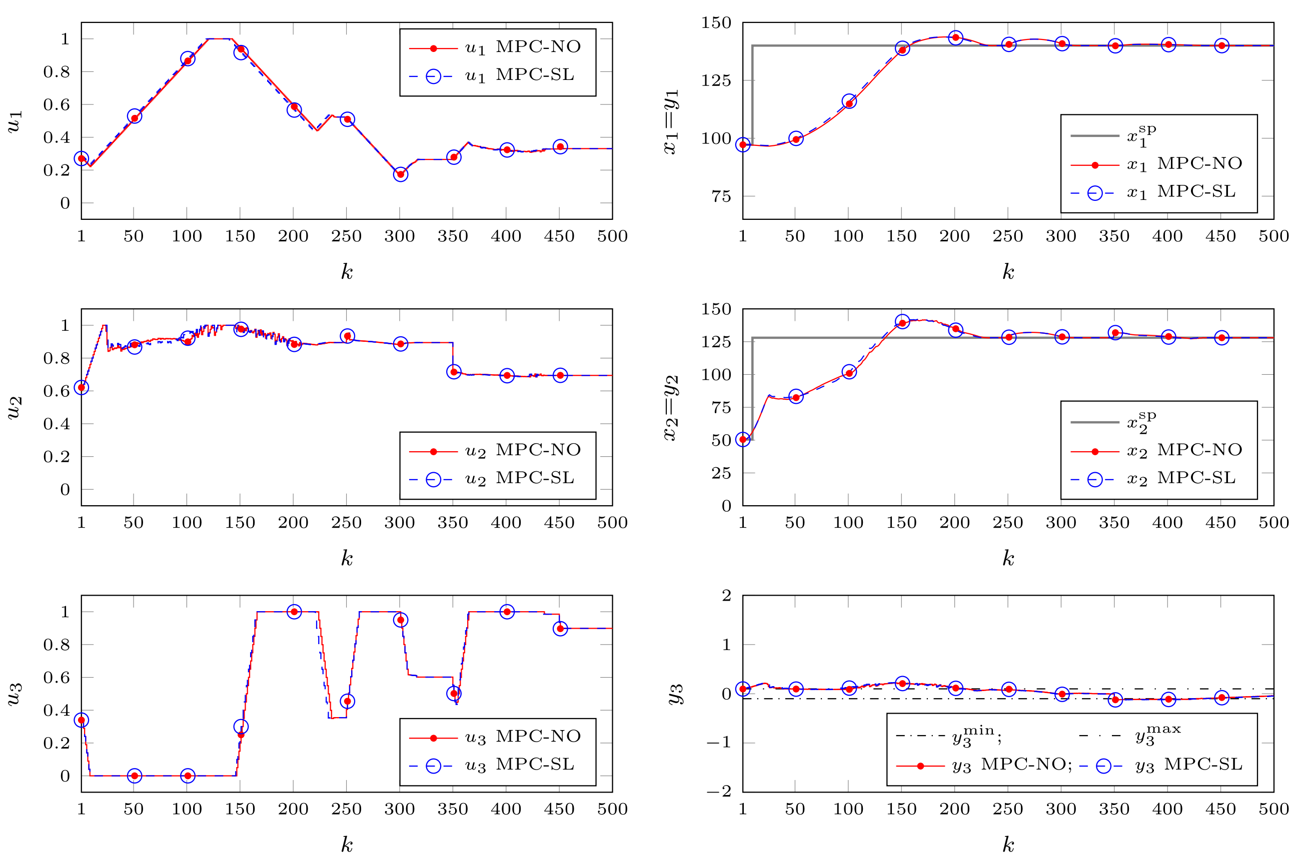

Fig. 11. Simulation results: the nonlinear MPC-NO algorithm (*solid line with dots*) and the nonlinear MPC-SL algorithm (*dashed line with circles*) for set-point change from the operating point B to C (the state observer is used with a wrong initial condition, the constraints imposed on the rate of change of the manipulated variables are taken into account, the inputs of the process are affected by step disturbances, the outputs are affected by measurement noise).

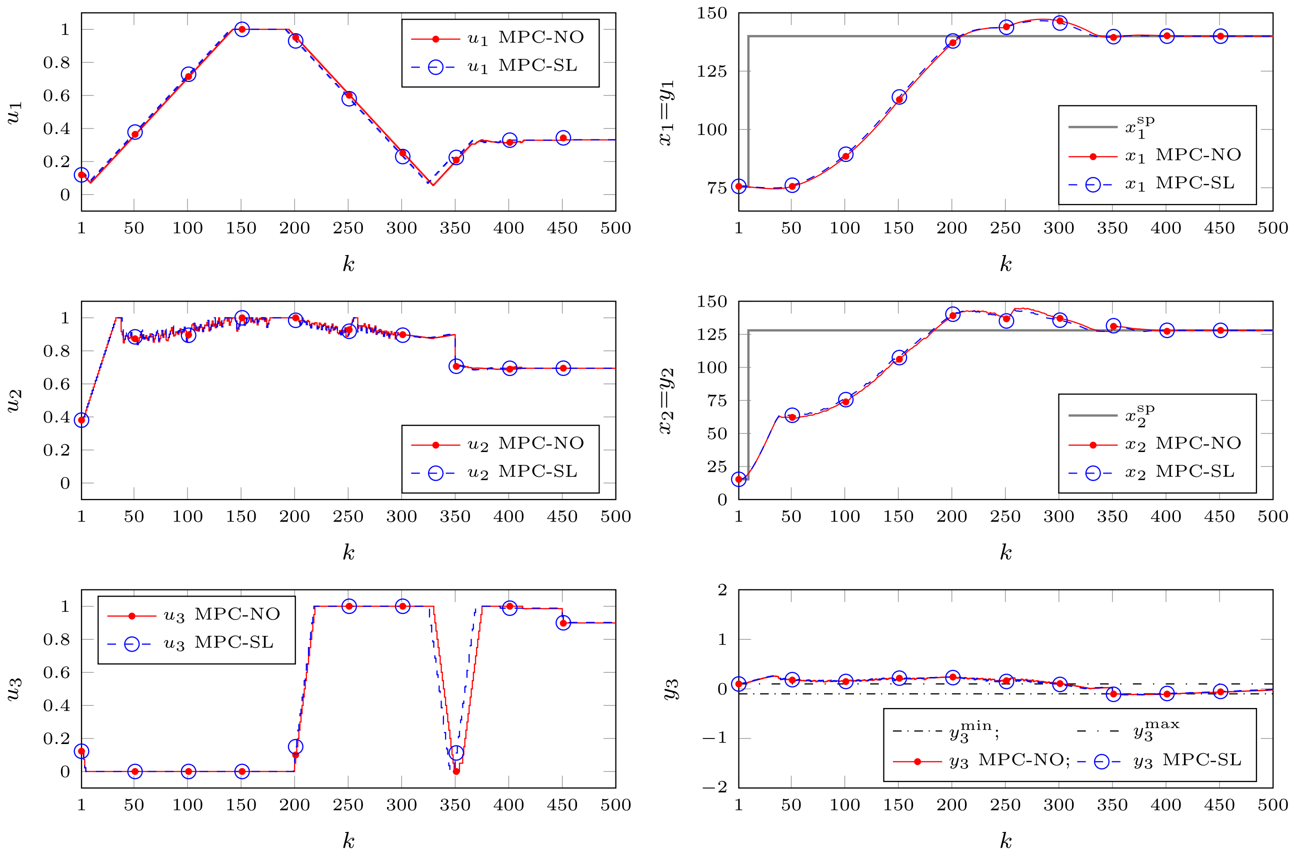

Fig. 12. Simulation results: the nonlinear MPC-NO algorithm (*solid line with dots*) and the nonlinear MPC-SL algorithm (*dashed line with circles*) for set-point change from the operating point A to C (the state observer is used with a wrong initial condition, the constraints imposed on the rate of change of the manipulated variables are taken into account, the inputs of the process are affected by step disturbances, the outputs are affected by measurement noise).

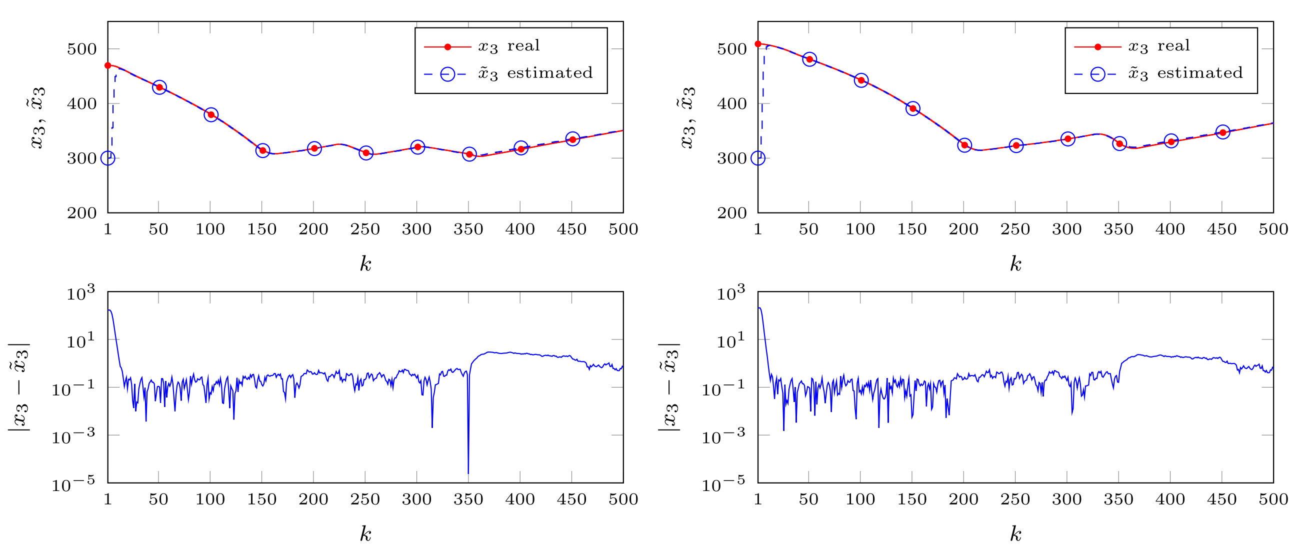

Fig. 13. Top panels: real state trajectories (*solid line with dots*) and the estimated state trajectories (*dashed line*) in the nonlinear MPC-SL algorithm (*dashed line with circles*), bottom panels: state estimation errors, for set-point change from the operating point B to C (*left panels*) and from A to C (*right panels*), (the state observer is used with a wrong initial condition, the constraints imposed on the rate of change of the manipulated variables are taken into account, the inputs of the process are affected by step disturbances, the outputs are affected by measurement noise).

imposed on the rate of change of the manipulated variables, with or without the state observer. When the observer is used, it is also assumed that its initial condition is wrong, the process output measurements are characterised by noise and two input step disturbances act on the process. Two approaches to soft output constraints are also compared, i.e. the suboptimal method in which the same coefficients $\varepsilon^{\min}(k)$ and $\varepsilon^{\max}(k)$ are used over the whole prediction horizon (as in the MPC-SL optimisation problem (41) with $3N_{\rm u}+2$ decision variables) and the optimal method in which different violations $\varepsilon^{\min}(k+1), \ldots, \varepsilon^{\min}(k+N)$ and $\varepsilon^{\max}(k+1), \ldots, \varepsilon^{\max}(k+N)$ may be adjusted independently for consecutive sampling instants (as in the MPC-SL optimisation

Table 4
Accuracy of compared nonlinear MPC algorithms in terms of the performance index SSE_x.

Set-point change	State observer	Constraints $\triangle u^{min}$, $\triangle u^{max}$	Algorithm			
			Suboptimal output soft constraints		Optimal output soft constraints	
			MPC-SL	MPC-NO	MPC-SL	MPC-NO
B → C	No	No	1.0898×10^5	1.0894×10^5	1.0864×10^5	1.0895×10^5
B → C	No	Yes	3.1675×10^5	3.1772×10^5	3.3168×10^5	3.1664×10^5
B → C	Yes	Yes	3.3328×10^5	3.4133×10^5	3.3065×10^5	3.4102×10^5
A → C	No	No	3.3110×10^5	3.3106×10^5	3.3077×10^5	3.3106×10^5
A → C	No	Yes	9.3900×10^5	9.4059×10^5	9.8272×10^5	9.3635×10^5
A → C	Yes	Yes	9.9041×10^5	1.0263×10^6	1.0374×10^6	1.0233×10^6

Table 5
Accuracy of compared nonlinear MPC algorithms in terms of the performance index SSE_y.

Set-point change	State observer	Constraints $\triangle u^{min}$, $\triangle u^{max}$	Algorithm			
			Suboptimal output soft constraints		Optimal output soft constraints	
			MPC-SL	MPC-NO	MPC-SL	MPC-NO
B → C	No	No	7.6243	7.6154	7.5697	7.6157
B → C	No	Yes	1.5203	1.5335	1.6464	1.5224
B → C	Yes	Yes	9.2890×10^{-1}	8.2773×10^{-1}	9.4550×10^{-1}	8.2173×10^{-1}
A → C	No	No	1.5753×10^1	1.5743×10^1	1.5696×10^1	1.5743×10^1
A → C	No	Yes	3.6320	3.6576	4.2028	3.6008
A → C	Yes	Yes	2.5186	2.4817	3.1905	2.4484

Table 6
Computational burden of the MPC-SL and MPC-NO algorithms; for set-point change from the operating point B to C (the state observer is used with a wrong initial condition, the constraints imposed on the rate of change of the manipulated variables are taken into account, the inputs of the process are affected by step disturbances, the outputs are affected by measurement noise).

Algorithm	Output soft constraints	$N_u = 1$	$N_u = 2$	$N_u = 3$	$N_u = 4$	$N_u = 5$	$N_u = 10$
MPC-SL	Suboptimal	5.67	5.69	6.11	6.55	7.06	10.06
MPC-SL	Optimal	5.66	6.39	7.11	8.08	20.41	37.31
MPC-NO	Suboptimal	38.66	83.56	91.78	113.61	130.61	256.51
MPC-NO	Optimal	241.22	453.16	590.44	687.55	1052.60	4733.30

Table 7
Computational burden of the MPC-SL and MPC-NO algorithms; for set-point change from the operating point A to C (the state observer is used with a wrong initial condition, the constraints imposed on the rate of change of the manipulated variables are taken into account, the inputs of the process are affected by step disturbances, the outputs are affected by measurement noise).

Algorithm	Output soft constraints	$N_u = 1$	$N_u = 2$	$N_u = 3$	$N_u = 4$	$N_u = 5$	$N_u = 10$
MPC-SL	Suboptimal	6.56	6.81	7.23	7.44	8.33	10.61
MPC-SL	Optimal	5.80	6.36	6.89	7.84	20.55	34.75
MPC-NO	Suboptimal	47.76	51.78	61.95	73.51	126.30	325.34
MPC-NO	Optimal	184.50	523.59	604.30	666.50	948.23	4742.80

problem (44) with $3N_u + 2N$ decision variables). For the simulations carried out, two conclusions may be drawn: performance (set-point accuracy) of the MPC-SL algorithm is very similar to that of the MPC-NO one and the suboptimal approach to soft constraints does not lead to any significant worsening of constraint satisfaction.

It is interesting to compare computational burden of the discussed MPC-SL and MPC-NO algorithms. For this purpose the cputime command in Matlab is used. Tables 6 and 7 give computational effort of both algorithms for set-point changes from the operating point B to C and from A to C, corresponding to the trajectories depicted in Figs. 11 and 12. For comparison different control horizons are taken into account ($N_u = 1, 2, 3, 4, 5, 10$) and two methods of treating the soft output constraints. First of all, it may be easily noticed that the MPC-SL algorithm is many times less computationally demanding than the MPC-NO one. For example, for the first considered trajectory and the suboptimal soft output constraints, the complexity reduction factor is 14.68 when $N_u = 2$, 18.50 when $N_u = 5$ and 25.50 when $N_u = 10$. One may also notice that as the control horizon is lengthened computational burden of the MPC-SL algorithm grows, but the growth is quite moderate, whereas in case of the MPC-NO algorithm lengthening of the control horizon leads to a dramatic growth of the computational burden. Secondly, the suboptimal soft output constraints, when compared with the optimal ones, are very computationally efficient, in particular in the case of MPC-SL algorithm.

6. Conclusions

When the full nonlinear state-space model of the boiler-turbine system is used in MPC, it is necessary to solve on-line a nonlinear optimisation problem with constraints at each sampling instant [16,19]. Nonlinear on-line optimisation in real time may be impossible in practice because of its significant computational burden. This paper details derivation of the suboptimal MPC-SL algorithm for the process and discusses simulation results. In the MPC-SL algorithm the state-space model is successively linearised on-line for the current operating point of the process and the obtained linear approximation is used for prediction of state and output variables. The MPC-SL algorithm has the following advantages:

1. Linearisation makes it possible to obtain a computationally not demanding MPC quadratic optimisation problem. Its computational burden is many times lower than that of the MPC with nonlinear optimisation.
2. Unlike the MPC algorithm based on linear models, the MPC-SL technique gives good control accuracy and leads to satisfaction of the output constraints.
3. For different set-point changes, different constraint configurations and unmeasured disturbances, the suboptimal MPC-SL algorithm gives process trajectories very similar to those obtained in the MPC scheme with nonlinear optimisation.
4. The MPC-SL algorithm uses quite simple prediction method discussed in [38], which guarantees offset-free control. It means that the MPC-SL algorithm is able to compensate for unmeasured disturbances, i.e. the controlled variables of the process follow their set-point trajectories even when the process is affected by some unmeasured disturbances and noise.

Moreover, two approaches to treating the soft output constraints are described. The first one is suboptimal, but it introduces only two additional decision variables and four constraints whereas in the second one (optimal) the number of decision variables and the number of constraints are linearly dependent on the prediction horizon. Taking into account the presented simulation results, it is found that for the boiler-turbine process the suboptimal approach to soft output constraints gives practically the same results as the optimal one.

References

[1] Bell RD, Åström KJ. Dynamic models for boiler-turbine alternator units: data logs and parameter estimation for a 160 MW unit. Tech. Rep. LUTFD2/(TFRT-3192)/1-137, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden, 1987.
[2] Chen P-C. Multi-objective control of nonlinear boiler-turbine dynamics with actuator magnitude and rate constraints. ISA Trans 2013;52:115–28.
[3] Chen P-C, Shamma JS. Gain-scheduled l_1-optimal control for boiler-turbine dynamics with actuator saturation. J Proc Control 2004;14:263–77.
[4] Cheres E. Small and medium size drum boiler models suitable for long term dynamic response. IEEE Trans Energy Convers 1990;5:686–92.
[5] Choi M, Choi J. W.K. State estimation with delayed measurements incorporating time-delay uncertainty. IET Control Theory Appl 2012;6:2351–61.
[6] Dimeo R, Lee KY. Boiler-turbine control system design using a genetic algorithm. IEEE Trans Energy Convers 1995;10:752–9.
[7] Dubay R, Hassan M, Li C, Charest M. Finite element based model predictive control for active vibration suppression of a one-link flexible manipulator. ISA Trans 2014;53:1609–19.
[8] Dubay R. Self-optimizing MPC of melt temperature in injection moulding. ISA Trans 2002;41:81–94.
[9] Dutta P, Rhinehart RR. Application of neural network control to distillation and an experimental comparison with other advanced controllers. ISA Trans 1999;38:251–78.
[10] Garduno-Ramirez R, Lee KY. Wide range operation of a power unit via feedforward fuzzy control. IEEE Trans Energy Convers 1995;15:421–6.
[11] Ghabraei S, Moradi H, Vossoughi G. Multivariable robust adaptive sliding mode control of an industrial boiler-turbine in the presence of modeling imprecisions and external disturbances: a comparison with type-I servo controller. ISA Trans 2015;58:398–408.
[12] Haeri M, Beik HZ. Application of extended DMC for nonlinear MIMO systems. Comp Chem Eng 2005;29:1867–74.
[13] Jalali AA, Golmohammad H. An optimal multiple-model strategy to design a controller for nonlinear processes: a boiler-turbine unit. Comp Chem Eng 2012;46:48–58.
[14] Keshavarz M, Yazdi MB, Jahed-Motlagh MR. Piecewise affine modeling and control of a boiler-turbine unit. Appl Therm Eng 2010;30:781–91.
[15] Lee JH, Ricker NL. Extended Kalman filter based nonlinear model predictive control. Ind Eng Chem Res 1994;33:1530–41.
[16] Li Y, Shen J, Lee KY, Liu X. Offset-free fuzzy model predictive control of a boiler-turbine system based on genetic algorithm. Simul Model Pract Theory 2012;26:77–95.
[17] Liu X, Kong X. Nonlinear fuzzy model predictive iterative learning control for drum-type boiler-turbine system. J Proc Control 2013;23:1023–40.
[18] Liu X, Guan P, Chan CW. Nonlinear multivariable power plant coordinate control by constrained predictive scheme. IEEE Trans Control Syst Technol 2010;18:1116–25.
[19] Lu Z, Lin W, Feng G, Wan F. A study of nonlinear control schemes for a boiler-turbine unit. In: Proceedings of the 8th IFAC symposium on nonlinear control systems, NLCOS 2010, Bologna, Italy; 2010.
[20] Ławryńczuk M. Computationally efficient model predictive control algorithms: a neural network approach, studies in systems, decision and control, Springer, Cham; 2014 vol. 3.
[21] Maciejowski JM. Predictive control with constraints.Englewood Cliffs: Prentice, Hall; 2002.
[22] Muske K, Badgwell T. Disturbance modeling for offset-free linear model predictive control. J Proc Control 2002;5:617–32.
[23] Nocedal J, Wright SJ. Numerical optimization.Berlin: Springer; 2006.
[24] Maffezzoni C. Boiler-trubine dynamics in power-plant control. Control Eng Pract 1997;5:301–12.
[25] Moon U-C, Lee KY. An adaptive dynamic matrix control with fuzzy-interpolated step-response model for a drum-type boiler-turbine system. IEEE Trans Energy Convers 2011;26:393–401.
[26] Moon U-C, Lee KY. Step-response model development for Dynamic Matrix Control of a drum-type boiler-turbine system. IEEE Trans Energy Convers 2009;24:423–30.
[27] Moon U-C, Lee KY. A boiler-turbine system control using a fuzzy auto-regressive moving average (FARMA) model. IEEE Trans Energy Convers 2003;18:142–8.
[28] Novak J, Chalupa P. Predictive control of a boiler-turbine system. In: V.E. Balas, M. Koksal (Eds.). In: Proceedings of the 16th WESAS international conference on systems, Kos, Greece, 2013, pp. 383–8.
[29] Pannocchia G, Bemporad A. Combined design of disturbance model and observer for offset-free model predictive control. IEEE Trans Aut Control 2007;52:1048–53.
[30] Rossiter JA. Model-based predictive control.Boca Raton: CRC Press; 2003.
[31] Sarailoo M, Rahmani Z, Rezaie B. Fuzzy predictive control of a boiler-turbine system based on a hybrid model system. Ind Eng Chem Res 2014;53:2362–81.
[32] Schnelle PD, Rollins DL. Industrial model predictive control technology as applied to continuous polymerization processes. ISA Trans 1998;36:281–92.
[33] Shakouri P, Ordys A, Askari MR. Adaptive cruise control with stop & go function using the state-dependent nonlinear model predictive control approach. ISA Trans 2012;51:622–31.
[34] Simon D. Optimal state estimation: kalman, H_∞ and nonlinear approaches. Hoboken: John Wiley & Sons; 2006.
[35] Škrjanc I, Blažič S, Oblak S, Richalet J. An approach to predictive control of multivariable time-delayed plant: stability and design issues. ISA Trans 2004;43:585–95.
[36] Tan W, Marquez HJ, Chen To, Liu J. Analysis and control of a nonlinear boiler-turbine unit. J Proc Control 2005;15:883–91.
[37] Tan W, Niu YG, Liu JZ. H_∞ control for a boiler-turbine unit. In: Proceedings of the IEEE conference on control applications, Hawaii, USA, 1999, pp. 807–10.
[38] Tatjewski P. Disturbance modeling and state estimation for offset-free predictive control with state-space models. Int J Appl Math Comp Sci 2014;24:313–23.
[39] Tatjewski P. Advanced control of industrial processes, structures and algorithms.London: Springer; 2007.
[40] Wei D, Craig IK, Bauer M. Multivariate economic performance assessment of an MPC controlled electric arc furnace. ISA Trans 2007;46:429–36.
[41] Wu X, Shen J, Li Y, Lee KY. Data-driven modeling and predictive control for boiler-turbine unit using fuzzy clustering and subspace methods. ISA Trans 2014;53:699–708.
[42] Wu X, Shen J, Li Y, Lee KY. Hierarchical optimization of boiler-turbine unit using fuzzy stable model predictive control. Control Eng Pract 2014;30:112–23.
[43] Wu J, Nguang SK, Shen J, Liu G, Li YG. Robust H_∞ tracking control of boiler-turbine systems. ISA Trans 2010;49:369–75.