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

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 \( H_\infty \) 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 \( H_\infty \) 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 \( H_\infty \) 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 \( H_\infty \) 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
Motivation of this work is to develop a computationally ef- ficient 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. 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 model 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].
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]

\[
\begin{align*}
\frac{dx_1(t)}{dt} &= -0.0018u_2(t)x_1^{0.8}(t) + 0.9u_1(t) - 0.15u_3(t) \\
\frac{dx_2(t)}{dt} &= (0.073u_2(t) - 0.016)x_1^{0.8}(t) - 0.1x_2(t) \\
\frac{dx_3(t)}{dt} &= \frac{141u_2(t) - (1.1u_2(t) - 0.19)x_1(t)}{85}
\end{align*}
\]

where the state variables \(x_1\), \(x_2\), and \(x_3\) denote drum pressure (kg/cm²), electric power output (MW), and fluid density (kg/cm³), respectively. The manipulated inputs, \(u_1\), \(u_2\), and \(u_3\) are the valve positions for fuel flow, steam control, and feed-water flow, respectively. The outputs \(y_1\), \(y_2\), and \(y_3\) of the system are: drum pressure (kg/cm²), electric output (MW) and drum water level deviations (m). The continuous-time output equations are

\[
\begin{align*}
y_1(t) &= x_1(t) \\
y_2(t) &= x_2(t) \\
y_3(t) &= 0.05\left(0.13073x_3(t) + 100\alpha_{cs}(t) + \frac{q_v(t)}{9} - 67.975\right)
\end{align*}
\]

where \(\alpha_{cs}\) and \(q_v\) are steam quality (dimensionless) and evaporation rate (kg/s), respectively, and are given by

\[
\begin{align*}
\alpha_{cs}(t) &= \left(1 - 0.001538x_1(t)\right)\left(0.8x_1(t) - 25.6\right) \\
&\quad x_1(t)(1.0394 - 0.0012304x_1(t)) \\
q_v(t) &= (0.854u_2(t) - 0.147)x_1(t) + 45.59u_1(t) - 2.514u_3(t) - 2.096
\end{align*}
\]

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 \(\hat{x}_3\).

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 (\(y_1 = \hat{x}_1\)), electric power output (\(y_2 = x_2\)) and their set-points (\(y_{1p} = x_{1p}\), \(y_{2p} = x_{2p}\)) are minimised whereas water level deviations should be in the following range [3]

\[-0.1 \leq y_3(t) \leq 0.1\]  

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

\[0 \leq u_1(t) \leq 1\]
\[0 \leq u_2(t) \leq 1\]
\[0 \leq u_3(t) \leq 1\]  

Additionally, rate constraints [3]

\[-0.007 \leq u_1(t) \leq 0.007\]
\[-2 \leq u_2(t) \leq 0.02\]
\[-0.05 \leq u_3(t) \leq 0.05\]  

may be taken into account.

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

2.3. Model discretisation

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

\[
x_1(k+1) = x_1(k) + T_s\left(-0.0018u_2(k)x_1^{0.8}(k) + 0.9u_1(k) - 0.15u_3(k)\right)
\]
\[
x_2(k+1) = x_2(k) + T_s\left(0.073u_2(k) - 0.016)x_1^{0.8}(k) - 0.1x_2(k)\right)
\]
\[
x_3(k+1) = x_3(k) + T_s\left(\frac{141u_2(k) - (1.1u_2(k) - 0.19)x_1(k)}{85}\right)
\]

where \(T_s = 1\) s denotes the sampling time. From Eq. (2), the discrete-time output equations are

\[
y_1(k) = x_1(k)
\]
\[
y_2(k) = x_2(k)
\]
\[
y_3(k) = 0.05\left(0.13073x_3(k) + 100\alpha_{cs}(k) + \frac{q_v(k)}{9} - 67.975\right)
\]

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

\[
\alpha_{cs}(k) = \frac{(1 - 0.001538x_1(k))(0.8x_1(k) - 25.6)}{x_1(k)(1.0394 - 0.0012304x_1(k))}
\]
\[
q_v(k) = (0.854u_2(k) - 0.147)x_1(k) + 45.59u_1(k) - 2.514u_3(k) - 2.096
\]

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

\[x(k+1) = f(x(k), u(k))\]
\( y(k) = g(x(k), u(k)) \) \hspace{1cm} (10b)

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

\[
\begin{align*}
x(k) &= f(x(k-1), u(k-1), u_3(k-1), u_2(k-1), u_1(k-1)) \\
y(k) &= g(x(k), u(k))
\end{align*}
\]

Considering the actual relations between variables, one has

\[
\begin{align*}
x_1(k) &= f_1(x_0(k-1), u_1(k-1), u_2(k-1), u_3(k-1)) \\
x_2(k) &= f_2(x_0(k-1), x_3(k-1), u_2(k-1)) \\
x_3(k) &= f_3(x_0(k-1), x_3(k-1), u_2(k-1), u_3(k-1))
\end{align*}
\]

and

\[
\begin{align*}
y_1(k) &= g_1(x_1(k)) \\
y_2(k) &= g_2(x_2(k)) \\
y_3(k) &= g_3(x_3(k), x_2(k), u_1(k), u_2(k), u_3(k))
\end{align*}
\]

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 \( N_c \) [21,30,39]. Usually, for simplicity of implementation, control increments, not their values, are calculated. The vector of decision variables is

\[
\Delta u(k) = \begin{bmatrix} \Delta u(k) & \vdots & \Delta u(k+N_c-1) \end{bmatrix}
\]

Because the boiler-turbine system has 3 inputs, there are \( 3N_c \) decision variables. The control increments are defined as

\[
\Delta u_p(k+p|k) = \begin{cases}
\Delta u_p(k+p|k) - \Delta u_p(k-1|k) & \text{if } p = 0 \\
\Delta u_p(k+p|k) - \Delta u_p(k+p-1|k) & \text{if } p \geq 1
\end{cases}
\]

where \( n = 1, 2, 3 \). It is assumed that \( \Delta u_p(k+p|k) = 0 \) for \( p \geq N_c \). 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 \( x_1^p \), as well as discrepancies between predicted values of electric power output \( x_2 \) and its set-point trajectory \( x_2^p \). These discrepancies are considered on the prediction horizon \( N_c \). Thus, the following quadratic cost function is minimised on-line

\[
J(k) = \sum_{n=1}^{N_c} \sum_{p=1}^{N_p} \mu_{n,p} \left( x_1^p(k+p|k) - \hat{x}_1(k+p|k) \right)^2
\]

\[
+ \sum_{n=1}^{N_c} \sum_{p=0}^{N_p} \lambda_{n,p} \left( \Delta u_p(k+p|k) \right)^2
\]

where \( \mu_{n,p} \geq 0 \) and \( \lambda_{n,p} > 0 \) are weighting coefficients. The predictions \( \hat{x}_1(k+p|k) \) 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 \( \Delta u(k) \), i.e.

\[
\begin{align*}
u_1(k) &= \Delta u_1(k|k) + u_1(k-1) \\
u_2(k) &= \Delta u_2(k|k) + u_2(k-1) \\
u_3(k) &= \Delta u_3(k|k) + u_3(k-1)
\end{align*}
\]

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

\[
-0.1 \leq \hat{y}_3(k+p|k) \leq 0.1, \ p = 1, \ldots, N_c
\]

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

\[
\begin{align*}
0 \leq u_1(k+p|k) \leq 1, \ p = 0, \ldots, N_c - 1 \\
0 \leq u_2(k+p|k) \leq 1, \ p = 0, \ldots, N_c - 1 \\
0 \leq u_3(k+p|k) \leq 1, \ p = 0, \ldots, N_c - 1
\end{align*}
\]

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

\[
\begin{align*}
-0.007 \leq \Delta u_1(k+p|k) \leq 0.007, \ p = 0, \ldots, N_c - 1 \\
-1 \leq \Delta u_2(k+p|k) \leq 0.02, \ p = 0, \ldots, N_c - 1 \\
-0.05 \leq \Delta u_3(k+p|k) \leq 0.05, \ p = 0, \ldots, N_c - 1
\end{align*}
\]

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

\[
\begin{align*}
\min_{\Delta u(k), \mu_{n,p}, \lambda_{n,p}} & \sum_{p=1}^{N_c} \left\| \hat{x}_1^p(k+p|k) - \hat{x}_1(k+p|k) \right\|_{M_1} + \sum_{p=0}^{N_c-1} \| \Delta u(k+p|k) \|^2_{M_p} \\
+ \rho_{\min} \left( (\rho_{\min}(k))^2 + (\rho_{\max}(k))^2 \right)
\end{align*}
\]

subject to

\[
\begin{align*}
u_{\min} \leq u(k+p|k) \leq u_{\max}, \ p = 0, \ldots, N_c - 1 \\
\Delta u_{\min} \leq \Delta u(k+p|k) \leq \Delta u_{\max}, \ p = 0, \ldots, N_c - 1 \\
y_3^p(k) - \hat{y}_3(k+p|k) \leq y_3^p(k) + \epsilon(k), \ p = 0, \ldots, N_c - 1
\end{align*}
\]

\[
\begin{align*}
\epsilon_{\min}(k) \geq 0, \ \epsilon_{\max}(k) \geq 0
\end{align*}
\]

The input constraints are defined by the vectors:

\[
\begin{align*}
u_{\min} &= [0 \ 0 \ 0]^T, \ u_{\max} = [111]^T, \ \Delta u_{\min} = [-0.007 \ -2 \ -0.05]^T, \ \Delta u_{\max} = [0.007 \ 0.02 \ 0.05]^T, \ \mu_{n,p} \geq 0, \ \lambda_{n,p} > 0
\end{align*}
\]

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 \( 3N_c + 2 \), the number of constraints is \( 12N_c + 4 \), \( \epsilon_{\min}(k) \) and \( \epsilon_{\max}(k) \) are additional decision variables, \( \mu_{\min}, \mu_{\max} > 0 \) 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
\[ x^{\text{yp}}(k) = \begin{bmatrix} x^{\text{yp}}(k + 1) \\ \vdots \\ x^{\text{yp}}(k + Nk) \end{bmatrix}, \quad \hat{x}(k) = \begin{bmatrix} \hat{x}(k + 1) \\ \vdots \\ \hat{x}(k + Nk) \end{bmatrix} \]
are of length 3N, the weighting matrices \( M = \text{diag}(M_0, \ldots, M_\ell) \) and \( \Lambda = \text{diag}(\Lambda_0, \ldots, \Lambda_{Nk-1}) \) are of dimensionality 3N x 3N and 3Nk x 3Nk, respectively. The input constraint vectors
\[ u^{\text{in}} = \begin{bmatrix} u_{\text{min}}^1 \\ \vdots \\ u_{\text{min}}^N \end{bmatrix}, \quad u^{\text{max}} = \begin{bmatrix} u_{\text{max}}^1 \\ \vdots \\ u_{\text{max}}^N \end{bmatrix}, \quad \Delta u^{\text{min}} = \begin{bmatrix} \Delta u_{\text{min}}^1 \\ \vdots \\ \Delta u_{\text{min}}^N \end{bmatrix}, \quad \Delta u^{\text{max}} = \begin{bmatrix} \Delta u_{\text{max}}^1 \\ \vdots \\ \Delta u_{\text{max}}^N \end{bmatrix} \]
are of length 3Nk, the output constraint vectors
\[ y_3^{\text{min}} = \begin{bmatrix} y_3^{\text{min}} \\ \vdots \\ y_3^{\text{min}} \end{bmatrix}, \quad y_3^{\text{max}} = \begin{bmatrix} y_3^{\text{max}} \\ \vdots \\ y_3^{\text{max}} \end{bmatrix} \]
are of length N. The general MPC optimisation problem defined by Eq. (18) can be expressed as
\[
\begin{align*}
\min_{u^{\text{min}}(k), u^{\text{max}}(k)} & \left\{ \frac{1}{2} \sum_{p=1}^{N} \| x_{\text{yp}}^k - \hat{x}^k(\cdot) \|^2 + \| \Lambda u^k \|^2 + \rho_{\text{min}} (\delta_{\text{min}}^k)^2 \\
+ & \rho_{\text{max}} (\delta_{\text{max}}^k)^2 \right\} \\
\text{subject to} & \\
u^{\text{in}}(k) \leq u^k \leq u^{\text{max}} \\
\Delta u^{\text{min}} \leq \Delta u^k \leq \Delta u^{\text{max}} \\
y_3^{\text{min}} - \delta_{\text{min}}(k) N_{k} \leq y_3^k \leq y_3^{\text{max}} + \delta_{\text{max}}(k) N_{k} \\
\delta_{\text{min}}(k) \geq 0, \quad \delta_{\text{max}}(k) \geq 0
\end{align*}
\]
(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 \( \delta_{\text{min}}(k) \) and \( \delta_{\text{max}}(k) \) 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
\[
\begin{align*}
\min_{u^{\text{min}}(k+p), u^{\text{max}}(k+p)} & \left\{ \frac{1}{2} \sum_{p=1}^{N} \| x_{\text{yp}}^k - \hat{x}^k(\cdot) \|^2 + \| \Lambda u^k \|^2 + \rho_{\text{min}} (\delta_{\text{min}}^k)^2 \\
+ & \rho_{\text{max}} (\delta_{\text{max}}^k)^2 \right\} \\
\text{subject to} & \\
u^{\text{in}}(k) \leq u^k \leq u^{\text{max}} \\
\Delta u^{\text{min}} \leq \Delta u^k \leq \Delta u^{\text{max}} \\
y_3^{\text{min}} - \delta_{\text{min}}(k+p) \leq y_3^k \leq y_3^{\text{max}} + \delta_{\text{max}}(k+p) \\
\delta_{\text{min}}(k+p) \geq 0, \quad \delta_{\text{max}}(k+p) \geq 0
\end{align*}
\]
(20)
The number of decision variables is 3Nk + 2N, the number of constraints is 12Nk + 4N. Using the vectors of additional variables
\[ \varepsilon_{\text{min}}(k) = \begin{bmatrix} \varepsilon_{\text{min}}^1(k+1) \\ \vdots \\ \varepsilon_{\text{min}}^{Nk}(k+N) \end{bmatrix}, \quad \varepsilon_{\text{max}}(k) = \begin{bmatrix} \varepsilon_{\text{max}}^1(k+1) \\ \vdots \\ \varepsilon_{\text{max}}^{Nk}(k+N) \end{bmatrix} \]
of length N, the MPC optimisation problem (20) may be also re-written using the vector-matrix notation in the following way
\[
\begin{align*}
\min_{\varepsilon_{\text{min}}(k), \varepsilon_{\text{max}}(k)} & \left\{ \frac{1}{2} \| x_{\text{yp}}^k - \hat{x}^k(\cdot) \|^2 + \| \Lambda u^k \|^2 + \rho_{\text{min}} (\varepsilon_{\text{min}}(k))^2 \\
+ & \rho_{\text{max}} (\varepsilon_{\text{max}}(k))^2 \right\} \\
\text{subject to} & \\
v^{\text{in}}(k) \leq u^k \leq u^{\text{max}} \\
\Delta u^{\text{min}} \leq \Delta u^k \leq \Delta u^{\text{max}} \\
y_3^{\text{min}} - \varepsilon_{\text{min}}(k) \leq y_3^k \leq y_3^{\text{max}} + \varepsilon_{\text{max}}(k) \\
\varepsilon_{\text{min}}(k) \geq 0, \quad \varepsilon_{\text{max}}(k) \geq 0
\end{align*}
\]
(21)

\[ \text{4. Nonlinear MPC algorithm with successive on-line model linearisation (MPC-SL) for the boiler-turbine unit} \]

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 \( \Delta u^k \). 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.

\[ \text{4.1. On-line model linearisation} \]

Using the Taylor series expansion method, the local linear approximation of the nonlinear state-space model described by Eqs. (11a) and (11b) is
\[
\begin{align*}
x(k) &= f(\hat{x}(k-1), \hat{u}(k-1)) + A(k)(x(k-1) - \hat{x}(k-1)) \\
+ & B(k)(u(k-1) - \hat{u}(k-1)) \\
y(k) &= g(\hat{x}(k), \hat{u}(k-1)) + C(k)(x(k) - \hat{x}(k)) \\
+ & D(k)(u(k) - \hat{u}(k-1))
\end{align*}
\]
(22a)
\[ \text{where the vectors} \]
\[
\begin{align*}
x(k-1) &= \begin{bmatrix} \hat{x}_1(k-1) \\ \hat{x}_2(k-1) \end{bmatrix}, \quad \hat{x}(k) = \begin{bmatrix} \hat{x}_1(k) \\ \hat{x}_2(k) \end{bmatrix}, \quad \hat{u}(k-1) = \begin{bmatrix} \hat{u}_1(k-1) \\ \hat{u}_2(k-1) \end{bmatrix}, \quad \hat{u}(k) = \begin{bmatrix} \hat{u}_1(k) \\ \hat{u}_2(k) \end{bmatrix}
\end{align*}
\]
define the current operating point of the process. It is described by real measurements of the first two state variables \( \hat{x}_1, \hat{x}_2 \) and by the estimated state of the third state variable \( \hat{x}_3 \) 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 \( \hat{u}_1(k), \hat{u}_2(k), \hat{u}_3(k) \), but by the most recent values from the previous
The linearised model (22a)–(22b) may be expressed in the following form

\[ x(k) = A(k)x(k-1) + B(k)u(k-1) + \delta_x(k) \quad \text{(23a)} \]

\[ y(k) = C(k)x(k) + D(k)u(k) + \delta_y(k) \quad \text{(23b)} \]

where

\[ \delta_x(k) = f(x(k-1), u(k-1)) - A(k)x(k-1) - B(k)u(k-1) \]

\[ \delta_y(k) = g(x(k), u(k-1)) - C(k)x(k) - D(k)u(k-1) \quad \text{(24)} \]

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)

\[ A(k) = \frac{df(x(k-1), u(k-1))}{dx(k-1)}_{|x(k-1)=x(k-1), u(k-1)=u(k-1)} \]

\[ B(k) = \frac{df(x(k-1), u(k-1))}{du(k-1)}_{|x(k-1)=x(k-1), u(k-1)=u(k-1)} \]

\[ C(k) = \frac{dh(x(k), u(k))}{dx(k)}_{|x(k)=x(k), u(k)=u(k)} \]

\[ D(k) = \frac{dh(x(k), u(k))}{du(k)}_{|x(k)=x(k), u(k)=u(k)} \quad \text{(25)} \]

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

\[
A(k) = \begin{bmatrix}
1 + \left(-0.002025\omega_2(k-1)\right)\bar{T}_i & 0 & 0 \\
0.082125\omega_2(k-1)\bar{T}_i & 1 & 0.1\bar{T}_i \\
-0.0181\bar{T}_i & 0 & 1 \\
-1.1\bar{T}_i & 0.19\bar{T}_i & 0 & 1
\end{bmatrix}
\]

\[
B(k) = \begin{bmatrix}
0.9\bar{T}_i & -0.0181\bar{T}_i & -0.15\bar{T}_i \\
0.073\bar{T}_i & 1 & 0 \\
-1.1\bar{T}_i & 0 & 141 \bar{T}_i & 85 \bar{T}_i & -1.1\bar{T}_i & 0.19\bar{T}_i & 0 & 1
\end{bmatrix}
\]

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

\[
C(k) = \begin{bmatrix}
\frac{\partial x_1(k)}{\partial x_1(k)} & 1 & 0 & 0 & 0 \\
\frac{\partial x_1(k)}{\partial x_2(k)} & 0 & 0 & 1 & 0 \\
\frac{\partial x_1(k)}{\partial x_3(k)} & 0 & 0 & 0 & 1
\end{bmatrix}
\]

\[
D(k) = \begin{bmatrix}
0 & 0 & 0 & 0 \\
9 & 0.0427 & 0 & 0.0065365 & 0.00123043 \bar{T}_i & -0.1257 \bar{T}_i
\end{bmatrix}
\]

\[
\partial_x(k) = \begin{bmatrix}
\frac{\partial x_1(k)}{\partial x_1(k)} & \frac{\partial x_1(k)}{\partial x_2(k)} & \frac{\partial x_1(k)}{\partial x_3(k)} & \frac{\partial x_2(k)}{\partial x_1(k)} & \frac{\partial x_2(k)}{\partial x_2(k)} & \frac{\partial x_2(k)}{\partial x_3(k)} & \frac{\partial x_3(k)}{\partial x_1(k)} & \frac{\partial x_3(k)}{\partial x_2(k)} & \frac{\partial x_3(k)}{\partial x_3(k)}
\end{bmatrix}
\]

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

\[
D(k) = \begin{bmatrix}
0 & 0 & 0 & 0 \\
2.2795 & 0.0427 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.1257 \bar{T}_i
\end{bmatrix}
\]

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

\[
\hat{x}(k + p|k) = x(k + p|k) + \nu(k)
\]

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 \( x_i(k) \), \( x_i(k) \) (measured), \( \hat{x}_i(k) \) (estimated) and the state variables calculated from the linearised state Eq. (23a) for the sampling instant \( k \)

\[
\nu(k) = \begin{bmatrix}
\nu_1(k) \\
\nu_2(k) \\
\nu_3(k)
\end{bmatrix} = x(k) - A(k)x(k-1) - B(k)u(k-1) - \delta_x(k)
\]

The first two state variables are measured whereas the third one is observed, which means that \( x_i(k) = [x_i(k) x_i(k) x_i(k)]^T \)

\( x(k-1) = [x_i(k-1) x_i(k-1) x_i(k-1)]^T \) Similarly to Eq. [32], the predicted output for the sampling instant \( k + p \) is calculated at the current instant \( k \) from

\[
\hat{y}_i(k + p|k) = y_i(k + p|k) + \nu_i(k)
\]

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

\[
y_i(k + p) = C(k)x(k) - D(k)u(k) - \delta_y(k)
\]

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
\[
\hat{x}(k+1k) = A(k)x(k) + B(k)u(k) + \delta_x(k) + v(k) \\
\hat{x}(k+2k) = A(k)\hat{x}(k+1k) + B(k)u(k+1) + \delta_x(k) + v(k) \\
\hat{x}(k+3k) = A(k)\hat{x}(k+2k) + B(k)u(k+2) + \delta_x(k) + v(k)
\]

The state predictions may be expressed as functions of the increments of the future control inputs (i.e. the decision variables of MPC)
\[
\hat{x}(k+1k) = A(k)x(k) + B(k)\Delta u(k) + B(k)u(k-1) + \delta_x(k) + v(k) \\
\hat{x}(k+2k) = A(k)\hat{x}(k+1k) + B(k)\Delta u(k) + B(k)u(k) + \delta_x(k) + v(k) \\
\hat{x}(k+3k) = A(k)\hat{x}(k+2k) + B(k)u(k+1) + A(k) + I_{3x3}) \delta_x(k)
\]

where \(I_{3x3}\) stands for an identity matrix of dimensionality \(3 \times 3\). The predicted state trajectory over the prediction horizon (the length of length \(N\)) may be expressed in the following compact manner
\[
\hat{x}(k) = \begin{bmatrix} \hat{x}(k+1k) \\ \vdots \\ \hat{x}(k+Nk) \end{bmatrix} = \begin{bmatrix} A(k) & B(k) \end{bmatrix} u(k) + \begin{bmatrix} I_{3x3} \\ A(k) + I_{3x3} \end{bmatrix} \delta_x(k) + P(k)\Delta u(k) + P(k) \delta v(k) (36)
\]


The matrices \(\bar{A}(k), \bar{B}(k)\) and \(V(k)\), of dimensionality \(3N \times 3\), are
\[
\bar{A}(k) = \begin{bmatrix} A(k) \\ \vdots \\ A^N(k) \end{bmatrix}, \quad V(k) = \begin{bmatrix} I_{3x3} \\ A(k) + I_{3x3} \\ \vdots \\ \sum_{i=1}^{N-1} A^i(k) + I_{3x3} \end{bmatrix}, \quad \bar{B}(k) = V(k)B(k)
\]

whereas the matrix \(P(k)\), of dimensionality \(3N \times 3N\), is
\[
P(k) = \begin{bmatrix} B(k) \\ (A(k) + I_{3x3}) B(k) \\ \vdots \\ (N-1) A^i(k) + I_{3x3} B(k) \\ \vdots \\ N A^i(k) + I_{3x3} B(k) \end{bmatrix}
\]

where \(B_{3x3}\) stands for a zeros matrix of dimensionality \(3 \times 3\)

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
\[
\hat{y}_3(k) = \begin{bmatrix} \hat{y}_3(k+1k) \\ \vdots \\ \hat{y}_3(k+Nk) \end{bmatrix} = \bar{C}(k)\hat{x}(k) + \bar{D}(k)u(k) + \delta y(k) + d(k)I_{Nv1}
\]

where the block-diagonal matrices are
\[
\bar{C}(k) = \begin{bmatrix} C_1(k) & \cdots & 0_{1x3} \\ \vdots & \ddots & \vdots \\ 0_{3x3} & \cdots & C_N(k) \end{bmatrix}, \quad \bar{D}(k) = \begin{bmatrix} D_1(k) & \cdots & 0_{1x3} \\ \vdots & \ddots & \vdots \\ 0_{3x3} & \cdots & D_N(k) \end{bmatrix}
\]
and the submatrices \(C_i(k)\) and \(D_i(k)\) contains the third row of the matrices \(C(k)\) and \(D(k)\), respectively. The vector \(u(k)\), of length \(N\), 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
\[
u(k) = \begin{bmatrix} u(k+1k) \\ \vdots \\ u(k+Nk) \end{bmatrix} = J^N \Delta u(k) + u(k-1)
\]

The entries of the matrix \(J^N\), of dimensionality \(3N \times 3N\), are
\[
J^N_{ij} = \begin{cases} 1 & \text{if } j < i + 2 \\ 0 & \text{if } j \geq i + 2 \end{cases}
\]
and the vector \(u(k-1)\), of length \(3N\), is comprised of the values of the manipulated variables from the previous sampling instant
\[
u(k-1) = \begin{bmatrix} u(k-1) \\ \vdots \\ u(k-1) \end{bmatrix}
\]

Using the state prediction Eq. (36), the output predictions given by Eq. (39) may be rewritten in the form
\[
\hat{y}_3(k) = \bar{C}(k)\hat{x}(k) + \bar{D}(k)u(k-1) + \bar{V}(k)u(k-1) + \delta y(k) + d(k)I_{Nv1}
\]

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 \(\Delta u(k)\), 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
\[
\begin{align*}
\min_{\Delta u(k), \epsilon(k)} & \left\{ \left\| y_p(k) - \hat{A}(k)x(k) - \hat{B}(k)u(k-1) - V(k)\delta(k) - P(k)\Delta u(k) \right\|^2_{2m} \\
& + \epsilon(k)^2 + \rho_{\text{min}} \epsilon(k)^2 + \rho_{\text{max}} \epsilon(k)^2 \right\} \\
\text{subject to} & \quad u^{\text{min}} \leq f_x u(k) + u(k-1) \leq u^{\text{max}} \\
& \quad \Delta u^{\text{min}} \leq \Delta u(k) \leq \Delta u^{\text{max}} \\
& \quad y_3^{\text{min}} - \epsilon(k) I_{N_x} \leq \hat{C}(k)\left[ \hat{A}(k)x(k) + \hat{B}(k)u(k-1) + V(k)\delta(k) - P(k)\Delta u(k) \right] \\
& \quad + \left( \hat{C}(k)P(k) + \hat{D}(k)F(k) \right)\epsilon(k) I_{N_{x,1}} \leq y_3^{\text{max}} + \epsilon(k) I_{N_{x,1}} \\
& \quad \epsilon(k) \geq 0, \quad \epsilon(k) \geq 0 \\
\end{align*}
\]

where the vector
\[
\begin{bmatrix}
I_{3N_x} \\
I_{3N_x} I_{3N_x} \ldots I_{3N_x}
\end{bmatrix}
\]
is of length \(3N_x\) and the matrix
\[
J =
\begin{bmatrix}
I_{3x3} & 0_{3x3} & \ldots & 0_{3x3} \\
I_{3x3} & I_{3x3} & \ldots & 0_{3x3} \\
\vdots & \vdots & \ddots & \vdots \\
I_{3x3} & I_{3x3} & \ldots & I_{3x3}
\end{bmatrix}
\]
is of dimensionality \(3N_x \times 3N_x\). When different violations of the predicted output constraints may be adjusted for consecutive sampling instances of the prediction horizon, the general MPC optimisation problem (21) becomes

\[
\begin{align*}
\min_{\Delta u(k), \epsilon(k)} & \left\{ \left\| y_p(k) - \hat{A}(k)x(k) - \hat{B}(k)u(k-1) - V(k)\delta(k) - P(k)\Delta u(k) \right\|^2_{2m} \\
& + \epsilon(k)^2 + \rho_{\text{min}} \epsilon(k)^2 + \rho_{\text{max}} \epsilon(k)^2 \right\} \\
\text{subject to} & \quad u^{\text{min}} \leq f_x u(k) + u(k-1) \leq u^{\text{max}} \\
& \quad \Delta u^{\text{min}} \leq \Delta u(k) \leq \Delta u^{\text{max}} \\
& \quad y_3^{\text{min}} - \epsilon(k) I_{N_x} \leq \hat{C}(k)\left[ \hat{A}(k)x(k) + \hat{B}(k)u(k-1) + V(k)\delta(k) - P(k)\Delta u(k) \right] \\
& \quad + \left( \hat{C}(k)P(k) + \hat{D}(k)F(k) \right)\epsilon(k) I_{N_{x,1}} \leq y_3^{\text{max}} + \epsilon(k) I_{N_{x,1}} \\
& \quad \epsilon(k) \geq 0, \quad \epsilon(k) \geq 0 \\
\end{align*}
\]

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 \(y_j(k)\) 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 \(y_j(k)\) 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

\[
\begin{bmatrix}
z(k) \\
\end{bmatrix} =
\begin{bmatrix}
z_1(k) \\
z_2(k) \\
z_3(k) \\
z_j(k-1)
\end{bmatrix}
\]

\[
\begin{bmatrix}
\hat{x}_1(k-1) \\
\hat{x}_2(k-1) \\
\hat{x}_3(k-1) \\
\hat{x}_j(k-1)
\end{bmatrix}
\]

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

\[
x^a(k+1) = f^a(x^a(k), u^a(k)) + w(k)
\]

\[
z^a(k) = g^a(x^a(k), u^a(k)) + v(k)
\]

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

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

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

\[
x^a(k+1) = f^a(x^a(k), u^a(k))
\]

\[
z^a(k) = g^a(x^a(k), u^a(k)) + v(k)
\]

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 \(\hat{x}_3(k)\) 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 \(\hat{A}(k), \hat{B}(k), \hat{C}(k), \hat{D}(k)\) and are found from Eqs. (26), (27), (28), (29), (30), (31), whereas the quantities \(\delta(k), \hat{\delta}(k)\) are found from Eq. (24).
3. The matrices \(\hat{A}(k), \hat{V}(k)\) and \(\hat{B}(k)\) are calculated from Eqs. (37), the matrix \(\hat{P}(k)\) is calculated from Eq. (38).
4. The state disturbance vector \(\epsilon(k)\) 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 \(\Delta u(k)\) are applied to the process, i.e. \(u_1(k) = \Delta u_1(k) + u_1(k-1), u_2(k) = \Delta u_2(k) + u_2(k-1), u_3(k) = \Delta u_3(k) + u_3(k-1)\).
7. At the next sampling instant \(k = k + 1\) the algorithm goes to step 1.
\[ F(k-1) = \frac{\partial f(x(k-1), u(k-1))}{\partial x(k-1)} \]

\[ H(k) = \frac{\partial g(x(k), u(k))}{\partial x(k)} \]

One may notice that linearisation for the MPC-SL algorithm and for state estimation is performed in a very similar way. The matrix \( F(k-1) = A(k) \) is calculated from Eq. (26). The matrix \( H(k-1) = C(k) \) is calculated from Eqs. (28), (29) and (30), but for calculation of the matrix \( H(k) \) the operating point is defined by the quantities \( x_1(k-1) \), \( x_2(k-1) \) and \( x_3(k-1) \) whereas for model linearisation in the MPC-SL algorithm by the quantities \( \tilde{x}_1(k), \tilde{x}_2(k) \) and \( \tilde{x}_3(k) \). When constant matrices \( F \) and \( H \) 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:

\[ F^a(k) = \begin{bmatrix} A(k) & 0_{1\times 3} \\ 0_{1\times 3} & H^a(k) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & H_{1,1}(k-1) & H_{1,1}(k-1) & H_{1,1}(k-1) \end{bmatrix} \]

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

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: \( \mu_1, \mu_2 = 30, \mu_3, p = 1, \ldots, N, \epsilon_{\text{d}} = 1 \) for \( n = 1, 2, 3 \) and \( p = 0, \ldots, N_{\text{d}} - 1, \rho_{\text{min}} = \rho_{\text{max}} = 10^6 \). The extended Kalman filter has the following parameters: \( P(10) = I_{6\times 6}, Q = 100I_{6\times 6}, R = 0.0001I_{9\times 9} \).

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.

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 both 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

\[
\text{SSE}_\epsilon = \sum_{n=1}^{N_{\text{sim}}} \sum_{p=1}^{N_{\text{exp}}} (x_{\text{exp}}^p(k) - x_n(k))^2
\]

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

\[
\text{SSE}_y = \sum_{p=1}^{N_{\text{exp}}} (0.1 - \text{abs}(y_3)) \cdot \text{sign} (\text{max} (\text{abs}(y_3) - 0.1, 0))
\]
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 $S_{Ex}$ and $S_{Ey}$ for different prediction horizons $N$ 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 $N_0 = 5$ 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 $\epsilon_{\text{min}}(k)$ and $\epsilon_{\text{max}}(k)$, 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 $k=200$.

\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|}
\hline
$N$ & $S_{Ex}$ & $S_{Ey}$ \\
\hline
5 & $1.0858 \times 10^5$ & 7.7740 \\
10 & $1.0918 \times 10^5$ & 7.6435 \\
20 & $1.0957 \times 10^5$ & 7.7306 \\
30 & $1.0959 \times 10^5$ & 7.7328 \\
40 & $1.3034 \times 10^5$ & 7.7352 \\
50 & $1.5222 \times 10^5$ & 8.0571 \\
\hline
\end{tabular}
\caption{Accuracy of the MPC-SL algorithm in terms of the performance indices $S_{Ex}$ and $S_{Ey}$ for different prediction horizons $N$ for set-point change from the operating point B to C; $N_0 = 5$ (real state measurements are used, no constraints imposed on the rate of change of the manipulated variables are taken into account).}
\end{table}

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).
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 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 $SSE_x$ and $SSE_y$ 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. 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).](image)

### Table 3

<table>
<thead>
<tr>
<th>$N_u$</th>
<th>$SSE_x$</th>
<th>$SSE_y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$7.566243 \times 10^5$</td>
<td>0.5865</td>
</tr>
<tr>
<td>2</td>
<td>$1.089781 \times 10^5$</td>
<td>7.6243</td>
</tr>
<tr>
<td>3</td>
<td>$1.090982 \times 10^5$</td>
<td>7.6390</td>
</tr>
<tr>
<td>4</td>
<td>$1.091857 \times 10^5$</td>
<td>7.6457</td>
</tr>
<tr>
<td>5</td>
<td>$1.091822 \times 10^5$</td>
<td>7.6435</td>
</tr>
<tr>
<td>10</td>
<td>$1.091564 \times 10^5$</td>
<td>7.6556</td>
</tr>
</tbody>
</table>

$N_u > 2$) 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: $N = 10$, $N_u = 2$.

### 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
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

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

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

$$C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 2.8951 \times 10^{-3} & 0.58251 \times 10^{-3} \end{bmatrix},$$

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

for the point B.

and for the point C

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

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

$$C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 4.927322 \times 10^{-3} & 0.5250236 \times 10^{-3} \end{bmatrix},$$

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

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).
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 $\lambda_{np}$ 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 constraints 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 $\epsilon_{\text{min}}(k)$ and $\epsilon_{\text{max}}(k)$, 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 $\epsilon_{\text{min}}(k) > 0$ and $\epsilon_{\text{max}}(k) < 0$).
and $\epsilon_{\min}(k) > 0$. 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 this case the MPC-SL algorithms give trajectories very similar to those possible in the MPC-NO one.

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 $\tilde{x} = [50 ~ 25 ~ 300]^T$. 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 lengthened. It is evident that the nonlinear MPC algorithms are able to 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 indices $S_{Ex}$ (Eq. (46)) and $S_{Ey}$ (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).
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 characterized 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 $\varepsilon_{\text{min}}(k)$ and $\varepsilon_{\text{max}}(k)$ are used over the whole prediction horizon (as in the MPC-SL optimisation problem \ref{eq:MPC-SL}) with $3N_u + 2$ decision variables and the optimal method in which different violations $\varepsilon_{\text{min}}(k+1), \ldots, \varepsilon_{\text{min}}(k+N)$ and $\varepsilon_{\text{max}}(k+1), \ldots, \varepsilon_{\text{max}}(k+N)$ may be adjusted independently for consecutive sampling instants (as in the MPC-SL optimisation problem \ref{eq:MPC-SL}).
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 ($N_h = 1, 2, 3, 4, 5, 10$) 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 $N_h = 2, 18.50$ when $N_h = 5$ and 25.50 when $N_h = 10$. 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 and the obtained linear approximation is used for prediction of state and output variables. The MPC-SL algorithm has the following advantages:

---

Table 4

<table>
<thead>
<tr>
<th>Set-point change</th>
<th>State observer</th>
<th>Constraints $\Delta u_{\text{min}}, \Delta u_{\text{max}}$</th>
<th>Algorithm</th>
<th>Suboptimal output soft constraints</th>
<th>Optimal output soft constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>$B \rightarrow C$</td>
<td>No</td>
<td>No</td>
<td>MPC-SL</td>
<td>$1.0898 \times 10^5$</td>
<td>$1.0894 \times 10^5$</td>
</tr>
<tr>
<td>$B \rightarrow C$</td>
<td>Yes</td>
<td>Yes</td>
<td>MPC-NO</td>
<td>$3.1675 \times 10^5$</td>
<td>$3.1772 \times 10^5$</td>
</tr>
<tr>
<td>$A \rightarrow C$</td>
<td>No</td>
<td>No</td>
<td>MPC-SL</td>
<td>$3.3328 \times 10^5$</td>
<td>$3.4133 \times 10^5$</td>
</tr>
<tr>
<td>$A \rightarrow C$</td>
<td>Yes</td>
<td>Yes</td>
<td>MPC-NO</td>
<td>$3.3110 \times 10^5$</td>
<td>$3.3106 \times 10^5$</td>
</tr>
<tr>
<td>$A \rightarrow C$</td>
<td>No</td>
<td>Yes</td>
<td>MPC-SL</td>
<td>$9.3900 \times 10^5$</td>
<td>$9.4095 \times 10^5$</td>
</tr>
<tr>
<td>$A \rightarrow C$</td>
<td>Yes</td>
<td>Yes</td>
<td>MPC-NO</td>
<td>$9.9041 \times 10^5$</td>
<td>$1.0263 \times 10^6$</td>
</tr>
</tbody>
</table>

Table 5

<table>
<thead>
<tr>
<th>Set-point change</th>
<th>State observer</th>
<th>Constraints $\Delta u_{\text{min}}, \Delta u_{\text{max}}$</th>
<th>Algorithm</th>
<th>Suboptimal output soft constraints</th>
<th>Optimal output soft constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>$B \rightarrow C$</td>
<td>No</td>
<td>No</td>
<td>MPC-SL</td>
<td>$7.6243$</td>
<td>$7.6154$</td>
</tr>
<tr>
<td>$B \rightarrow C$</td>
<td>Yes</td>
<td>Yes</td>
<td>MPC-NO</td>
<td>$1.5203$</td>
<td>$1.5335$</td>
</tr>
<tr>
<td>$A \rightarrow C$</td>
<td>No</td>
<td>No</td>
<td>MPC-SL</td>
<td>$9.2890 \times 10^{-1}$</td>
<td>$8.2773 \times 10^{-1}$</td>
</tr>
<tr>
<td>$A \rightarrow C$</td>
<td>Yes</td>
<td>Yes</td>
<td>MPC-NO</td>
<td>$1.5753 \times 10^1$</td>
<td>$1.5743 \times 10^1$</td>
</tr>
<tr>
<td>$A \rightarrow C$</td>
<td>No</td>
<td>Yes</td>
<td>MPC-SL</td>
<td>$3.6320$</td>
<td>$3.6576$</td>
</tr>
<tr>
<td>$A \rightarrow C$</td>
<td>Yes</td>
<td>Yes</td>
<td>MPC-NO</td>
<td>$2.5186$</td>
<td>$2.4817$</td>
</tr>
</tbody>
</table>

Table 6

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Output soft constraints</th>
<th>$N_h = 1$</th>
<th>$N_h = 2$</th>
<th>$N_h = 3$</th>
<th>$N_h = 4$</th>
<th>$N_h = 5$</th>
<th>$N_h = 10$</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPC-SL Suboptimal</td>
<td></td>
<td>5.67</td>
<td>6.69</td>
<td>6.11</td>
<td>6.55</td>
<td>7.06</td>
<td>10.06</td>
</tr>
<tr>
<td>MPC-SL Optimal</td>
<td></td>
<td>5.66</td>
<td>6.39</td>
<td>7.11</td>
<td>8.08</td>
<td>20.41</td>
<td>37.31</td>
</tr>
<tr>
<td>MPC-NO Suboptimal</td>
<td></td>
<td>38.66</td>
<td>83.56</td>
<td>91.78</td>
<td>113.61</td>
<td>130.61</td>
<td>256.51</td>
</tr>
<tr>
<td>MPC-NO Optimal</td>
<td></td>
<td>241.22</td>
<td>453.16</td>
<td>590.44</td>
<td>687.55</td>
<td>1052.60</td>
<td>4733.30</td>
</tr>
</tbody>
</table>

Table 7

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Output soft constraints</th>
<th>$N_h = 1$</th>
<th>$N_h = 2$</th>
<th>$N_h = 3$</th>
<th>$N_h = 4$</th>
<th>$N_h = 5$</th>
<th>$N_h = 10$</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPC-SL Suboptimal</td>
<td></td>
<td>6.56</td>
<td>6.81</td>
<td>7.23</td>
<td>7.44</td>
<td>8.33</td>
<td>10.61</td>
</tr>
<tr>
<td>MPC-SL Optimal</td>
<td></td>
<td>5.80</td>
<td>6.36</td>
<td>6.89</td>
<td>7.84</td>
<td>20.55</td>
<td>34.75</td>
</tr>
<tr>
<td>MPC-NO Suboptimal</td>
<td></td>
<td>47.76</td>
<td>51.78</td>
<td>61.95</td>
<td>73.51</td>
<td>126.30</td>
<td>325.34</td>
</tr>
<tr>
<td>MPC-NO Optimal</td>
<td></td>
<td>184.50</td>
<td>523.59</td>
<td>604.30</td>
<td>686.50</td>
<td>948.23</td>
<td>4742.80</td>
</tr>
</tbody>
</table>
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


