Explicit output-feedback nonlinear predictive control based on black-box models☆

Alexandra Grancharova[a,*], Juš Kocijan[b,c], Tor A. Johansen[d]

[a]*Institute of Control and System Research, Bulgarian Academy of Sciences, Acad. G. Bonchev str., Bl.2, P.O.Box 79, Sofia 1113, Bulgaria*

[b]*Department of Systems and Control, Jozef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia*

[c]*Centre for systems and information technologies, University of Nova Gorica, Vipavska 13, 5000 Nova Gorica, Slovenia*

[d]*Department of Engineering Cybernetics, Norwegian University of Science and Technology, 7491 Trondheim, Norway*

Abstract

Nonlinear Model Predictive Control (NMPC) algorithms are based on various nonlinear models. A number of on-line optimization approaches for output-feedback NMPC based on various black-box models can be found in the literature. However, NMPC involving on-line optimization is computationally very demanding. On the other hand, an explicit solution to the NMPC problem would allow efficient on-line computations as well as verifiability of the implementation. This paper applies an approximate multi-parametric Nonlinear Programming approach to explicitly solve output-feedback NMPC problems for constrained nonlinear systems described by black-box models. In particular, neural network models are used and the optimal regulation problem is considered. A dual-mode control strategy is employed in order to achieve an offset-free closed-loop response in the presence of bounded disturbances and/or model errors. The approach is applied to design an explicit NMPC for regulation of a pH maintaining system. The verification of the

☆This work was financed by the National Science Fund of the Ministry of Education, Youth and Science of Republic of Bulgaria, contract DO02-94/14.12.2008 and the Slovenian Research Agency, contract BI-BG/09-10-005 ("Application of Gaussian processes to the modeling and control of complex stochastic systems").

*Corresponding author A. Grancharova. Tel. +359 889 625010. Fax +3592 870 33 61.

Email addresses: alexandra.grancharova@abv.bg (A. Grancharova), jus.kocijan@ijs.si (J. Kocijan), Tor.Arne.Johansen@itk.ntnu.no (T. A. Johansen)

NMPC controller performance is based on simulation experiments.

Keywords: Model predictive control, Black-box models, Multi-parametric nonlinear programming.

1. Introduction

Nonlinear Model Predictive Control (NMPC) involves the solution at each sampling instant of a finite horizon optimal control problem subject to nonlinear system dynamics and state and input constraints [1, 2, 3]. A survey of the numerical methods for on-line solution of NMPC problems is given in [4]. Most recently, an advanced-step NMPC controller with reduced on-line computational costs has been proposed in [5]. The NMPC algorithms are based on various nonlinear models. Often these models are developed as first-principles models, but other approaches, like black-box identification approaches are also popular. In this paper we focus on explicit solution of output-feedback NMPC problems based on black-box models.

There exists a number of NMPC approaches based on various black-box models e.g. based on neural network models (e.g. [6]), fuzzy models (e.g. [7]), local model networks (e.g. [8]), Gaussian Process models (e.g. [9]). The common feature of these NMPC approaches is that an on-line optimization needs to be performed in order to compute the optimal control input. Consequently, the computation is time consuming and the real-time NMPC implementation is limited to processes where the sampling time is sufficient to support the computational needs. However, the on-line computational complexity can be circumvented with an explicit approach to NMPC, where the only computation performed on-line would be a simple function evaluation.

It has been shown that the explicit solution to linear constrained MPC problems has an explicit representation as a piece-wise linear (PWL) state feedback law defined on a polyhedral partition of the state space [10]. The benefits of an explicit solution, in addition to the efficient on-line computations, include also verifiability of the implementation, which is an essential issue in safety-critical applications. In [11], the main contributions on explicit MPC, which have appeared in the scientific literature, are reviewed. For nonlinear MPC, the prospects of explicit solutions are even higher than for linear MPC, since the benefits of computational efficiency and verifiability are even more important. Recently, several approaches to explicit solution of NMPC problems have been suggested. An approach for efficient on-line computation of NMPC for constrained input-affine nonlinear systems has been suggested in [12]. In [13, 14, 15], approaches for off-line computation of explicit sub-optimal PWL predictive controllers for general nonlinear systems with state and input constraints have been developed, based on the multi-parametric Nonlinear Programming (mp-NLP) ideas [16]. It has been shown that for convex mp-NLP problems, it is straightforward to impose tolerances on the level of approximation such that theoretical properties like asymptotic stability of the sub-optimal feedback controller can be ensured [14, 17]. In [15], practical computational methods to handle non-convex mp-NLP problems have been suggested that not necessarily lead to guaranteed properties, but when combined with verification and analysis methods give a practical tool for development and implementation of explicit NMPC. Algorithms for solving mp-NLP problems, including the non-convex case, are described also in [18]. It should be noted that the mentioned methods for explicit NMPC are based on first-principles models of the systems and they assume that the state variables can be measured. Further, in [19], an approach for off-line computation of explicit *stochastic* NMPC controller for constrained nonlinear systems based on a stochastic black-box model (Gaussian process model) has been proposed. In addition to the mentioned methods, there exists another group of approaches for off-line computation of sub-optimal controllers, where the optimal solution is approximated by means of neural networks [20, 21, 22, 23].

This paper suggests an approximate mp-NLP approach to explicit solution of *deterministic* NMPC problems for constrained nonlinear systems described by black-box models (NARX models [24]). In particular, neural network NARX models are considered [25]. The approach builds an orthogonal search tree structure of the *regressor* space partition and consists in constructing a PWL approximation to the optimal control sequence by applying the approximate mp-NLP algorithm in [15]. A dual-mode control strategy is proposed in order to achieve an offset-free closed-loop response in the presence of bounded disturbances and/or model errors. It is similar to the dual-mode receding horizon control concept developed in [26] (based on state space models), however here black-box models are considered and an explicit solution of the NMPC problem is sought. Thus, the suggested strategy consists in using the explicit NMPC (based on NARX model) when the output variable is far from the origin and applying an LQR in a neighborhood of the origin. The LQR design is based on an augmented linear ARX model which takes into account the integral regulation error. The main motivations behind the dual-mode control strategy are the following. First, it may be beneficial to use a separate linear model in a neighborhood of the equilibrium, since the nonlinear black-box model may not have accurate linearizations unlike a first-principles model, and the requirement for accurate control is highest at the equilibrium. Second, it leads to a reduced complexity of the explicit NMPC compared to augmenting the nonlinear model with an integrator to achieve an integral action directly in the NMPC.

The following abbreviation and notation will be used in the paper. The nonlinear model predictive control problem based on black-box model will be referred to as BB-NMPC problem. $A \succ 0$ means that the square matrix A is positive definite. For $x \in \mathbb{R}^n$, the Euclidean norm is $\|x\| = \sqrt{x^T x}$ and the weighted norm is defined for some symmetric matrix $A \succ 0$ as $\|x\|_A = \sqrt{x^T A x}$.

2. Formulation of the BB-NMPC problem as an mp-NLP problem

2.1. Modelling of dynamic systems with NARX models

The black-box identification of nonlinear systems is an area which is quite diverse. It covers topics from mathematical approximation theory, estimation theory, non-parametric regression and concepts like neural networks, fuzzy models, wavelets etc. A unified overview of this topic is given in [27].

Consider a nonlinear dynamical system with input $u \in \mathbb{R}^m$ and output $y \in \mathbb{R}^p$ and let $U = [u(1), u(2), \dots, u(M)]$ and $Y = [y(1), y(2), \dots, y(M)]$ be sets of observed values of u and y to the number of M. Based on these data, the dynamics of the system can be described with a NARX model, where the future predicted output $y(i+1)$ depends on previous estimated outputs, as well as on previous control inputs:

$$y(i+1) = f(z(i), \theta) \tag{1}$$

$$z(i) = [y(i),\, y(i-1),\, \dots,\, y(i-L),\, u(i),\, u(i-1),\, \dots,\, u(i-L)] \tag{2}$$

Here, L is a given lag, i denotes the consecutive index of data samples ($i \geq L$), $z(i)$ is the so called *regressor* vector, f is the function realized by the black-box model, and θ is a finite-dimensional vector of parameters. Thus, the function f is a concatenation of two mappings: one that takes the increasing number of the past values of the observed inputs and outputs and maps them into the finite dimensional *regressor* vector and one that takes this vector to the space of the outputs. The nonlinear mapping from the *regressor* space to the output space can be of various kinds. In our case we will use neural network with sigmoid basis functions in the hidden layer and linear basis functions in the output layer. This form of neural network is called Multilayer Perceptron (MLP), which is probably the most frequently considered member of the neural network family (e.g. Norgaard et al. (2000)) and can be used as an universal approximator. This particular choice was subjective. Any other choice of *regressor* vector composition or any other choice of mapping is possible until it enables satisfactory description of the modelled dynamic system. The results given in the continuation of the paper are not limited to MLP approach only.

The parameters of the MLP are the weights of its units. After the structure (number of layers and units) is determined, the model parameters are obtained with optimization, based on a chosen cost function. This cost function is most frequently a least squares combination of errors between estimated and measured output signals:

$$E = \frac{1}{2M} \sum_{i=1}^{M} \|y(i) - \hat{y}(i|\theta)\|^2 \tag{3}$$

where $\hat{y}(i|\theta)$ is estimated output signal, θ is a vector containing the weights, and M is the number of measured output signals $y(i)$. The quality of prediction can be assessed with evaluation of residuals, estimation of the average prediction error or visualization of the network model's ability to predict. The reader is referred to [6] for more details.

2.2. Formulation of the BB-NMPC problem

Consider the discrete-time nonlinear system:

$$x(t+1) = h(x(t), u(t)) \tag{4}$$

$$y(t) = g(x(t), u(t)) \tag{5}$$

where $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$, and $y(t) \in \mathbb{R}^p$ are the state, input and output vectors, and $h : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ and $g : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^p$ are nonlinear functions. The following input and output constraints are imposed on the system (4)–(5):

$$u_{\min} \leq u(t) \leq u_{\max} \ , \ y_{\min} \leq y(t) \leq y_{\max} \tag{6}$$

Assume that the dynamics of the nonlinear system (4)–(5) is approximated with an MLP neural network with NARX structure of the form (1)–(2). Then for $t \geq L$, define a *modified regressor* vector:

$$\tilde{z}(t) = \begin{cases} [y(t),\, y(t-1),\, \dots,\, y(t-L), \\ \qquad u(t-1),\, \dots,\, u(t-L)]\,, \text{ if } L > 0 \\ y(t)\,, \text{ if } L = 0 \end{cases}, \tag{7}$$

where $u(t-1)$, ..., $u(t-L)$ and $y(t)$, $y(t-1)$, ..., $y(t-L)$ are the measured values of the input u and the output y. Thus, $\tilde{z}(t) \in \mathbb{R}^q$ with $q = (L+1)p + Lm$ if $L > 0$ and $q = p$ if $L = 0$. Then, the NARX model, used to obtain one-step ahead prediction of the output for $t \geq L$, is represented:

$$\hat{y}(t+1|\theta) = f_{NN}(\tilde{z}(t), u(t), \theta)\,, \tag{8}$$

where f_{NN} is the function realized by the neural network (NN) and θ contains the network weights. Suppose the initial *regressor* vector $\tilde{z}(t) = \tilde{z}_{t|t}$ is known and the control inputs $u(t+k) = u_{t+k}\,,\ k = 0,\, 1,\, \dots,\, N-1$ are given. Then, the model (8) can be used to obtain the predicted output $y_{t+k+1|t}$, $k = 0,\, 1,\, \dots,\, N-1$ through iterative one-step ahead predictions, where at each step the predicted output value is fed back to the *regressor* vector:

$$y_{t+k+1|t} = f_{NN}(\tilde{z}_{t+k|t}, u_{t+k}, \theta) \tag{9}$$

$$\tilde{z}_{t+k|t} = \begin{cases} [y_{t+k|t}, y_{t+k-1|t},\, \cdots, y_{t+k-L|t}, \\ \qquad u_{t+k-1},\, \dots, u_{t+k-L}], \text{ if } L > 0 \\ y_{t+k|t}, \text{ if } L = 0 \end{cases} \tag{10}$$

The following assumptions are made:

A1. There exists $u_{st}^{\mathrm{NN}} \in \mathbb{R}^m$ satisfying $u_{\min} \leq u_{st}^{\mathrm{NN}} \leq u_{\max}$, and such that $f_{NN}(\tilde{z}_0, u_{st}^{\mathrm{NN}}, \theta) = 0$, where $\tilde{z}_0$ is obtained from (10) with $y_{t+k|t} = y_{t+k-1|t} = \dots = y_{t+k-L|t} = 0$, $u_{t+k-1} = \dots = u_{t+k-L} = u_{st}^{\mathrm{NN}}$.

A2. $y_{\min} < 0 < y_{\max}$.

Assumption A1 means that the point $y = 0$, $u = u_{st}^{\mathrm{NN}}$, is an equilibrium point for the NARX model (8), and Assumption A2 means that it is feasible for (6).

We consider the optimal regulation problem where the goal is to steer the output variable y to the origin by minimizing certain performance criterion. Suppose that a full measurement of the *modified regressor* vector $\tilde{z}(t)$ is

available at the current time $t \geq L$. Then, for the current $\tilde{z}(t)$, the regulation BB-NMPC solves the following optimization problem:
Problem P1:

$$V^*(\tilde{z}(t)) = \min_U J(U, \tilde{z}(t)) \tag{11}$$

subject to $\tilde{z}_{t|t} = \tilde{z}(t)$ and:

$$y_{\min} \leq y_{t+k|t} \leq y_{\max}, \ k = 1, ..., N \tag{12}$$

$$u_{\min} \leq u_{t+k} \leq u_{\max}, \ k = 0, 1, ..., N-1 \tag{13}$$

$$y_{t+N|t} \in \Omega \tag{14}$$

$$y_{t+k+1|t} = f_{NN}(\tilde{z}_{t+k|t}, u_{t+k}, \theta), \ k = 0, 1, ..., N-1 \tag{15}$$

$$\tilde{z}_{t+k|t} = \begin{cases} [y_{t+k|t}, y_{t+k-1|t}, \ \dots, y_{t+k-L|t}, u_{t+k-1}, \ \dots, u_{t+k-L}], \text{ if } L > 0 \\ y_{t+k|t}, \text{ if } L = 0 \ , \end{cases} \tag{16}$$

$$k = 0, 1, ..., N-1$$

with $U = [u_t, u_{t+1}, \ \dots, u_{t+N-1}]$ and the cost function given by:

$$J(U, \tilde{z}(t)) = \sum_{k=0}^{N-1} \left[\left\| y_{t+k|t} \right\|_Q^2 + \left\| u_{t+k} - u_{st}^{\mathrm{NN}} \right\|_R^2 \right] + \left\| y_{t+N|t} \right\|_P^2 \tag{17}$$

Here, N is a finite horizon and P, Q, $R \succ 0$. In (14), Ω is the terminal set defined by $\Omega = \{y \in \mathbb{R}^p \,|\, \|y\|^2 \leq \delta\}$ with $\delta > 0$. From a stability point of view it is desirable to choose δ as small as possible [28]. If the system is asymptotically stable (or pre-stabilized) and N is large, then it is more likely that the choice of a small δ will be possible.

Let $\tilde{z}$ be the value of the *modified regressor* vector at the current time t. Then, the optimization problem P1 can be formulated in a compact form as follows:
Problem P2:

$$V^*(\tilde{z}) = \min_U J(U, \tilde{z}) \quad \text{subject to} \quad G(U, \tilde{z}) \leq 0 \tag{18}$$

The BB-NMPC problem defines an mp-NLP, since it is NLP in U parameterized by $\tilde{z}$. We remark that the constraints function $G(U, \tilde{z})$ in (18) is implicitly defined by (12)–(16). An optimal solution to this problem is denoted $U^* = [u_t^*, \ u_{t+1}^*, \ \dots, \ u_{t+N-1}^*]$ and the control input is chosen according to the receding horizon policy $u(t) = u_t^*$. Define the set of N-step feasible initial *regressor* vectors as follows:

$$Z_f = \{\tilde{z} \in \mathbb{R}^q \,|\, G(U, \tilde{z}) \leq 0 \text{ for some } U \in \mathbb{R}^{Nm}\} \tag{19}$$

In parametric programming problems one seeks the solution $U^*(\tilde{z})$ as an explicit function of the parameters $\tilde{z}$ in some set $\underline{Z} \subseteq Z_f \subseteq \mathbb{R}^q$ [16]. The explicit solution allows us to replace the computationally expensive real-time optimization with a simple function evaluation.

3. Approximate mp-NLP approach to explicit BB-NMPC

Definition 1 (Feasibility on a discrete set). Let $\overline{Z} \subset \mathbb{R}^q$ be a hyper-rectangle and $V_{\overline{Z}} = \{v_1, v_2, \dots, v_Q\} \subset \overline{Z}$ be a discrete set. A function $U(\tilde{z})$ is feasible on $V_{\overline{Z}}$ if $G(U(v_i), v_i) \leq 0$, $i \in \{1, 2, \dots, Q\}$.

In general, the exact solution of problem P2 can not be found. In [15], a computational method for constructing an explicit piecewise linear (PWL) approximate solution of *state-space* NMPC problems has been suggested. Here, the approximate mp-NLP approach is applied to explicitly solve the *output-feedback* NMPC problem formulated in the previous section and it is given only in brief. Let $Z \subset \mathbb{R}^q$ be a hyper-rectangle where we seek to approximate the optimal solution $U^*(\tilde{z})$ to problem P2. It is required that the *regressor* space partition is orthogonal and can be represented as a $k-d$ tree. The main idea of the approximate mp-NLP approach is to construct a feasible on a discrete set PWL approximation $\widehat{U}(\tilde{z})$ to $U^*(\tilde{z})$ on Z, where the constituent affine functions are defined on hyper-rectangles covering Z. The computation of an affine *regressor* feedback associated to a given region Z_0 includes the following steps. First, a close-to-global solution of problem P2 is computed at a set of points in Z_0. Then, based on the solutions at these points, a local linear approximation $\widehat{U}_0(\tilde{z}) = K_0\tilde{z} + g_0$ to the close-to-global solution $U^*(\tilde{z})$, feasible at these points and valid in the whole hyper-rectangle Z_0, is determined by applying the following procedure [15]:

Procedure 1. (Computation of explicit approximate solution).
Consider any hyper-rectangle $Z_0 \subseteq Z$ with a set of points $V_0 = \{v_0, v_1, v_2, \dots, v_{N_1}\} \subset Z_0$. Compute K_0 and g_0 by solving the following NLP:
Problem P3:

$$\min_{K_0, g_0} \sum_{i=0}^{N_1} \left(J(K_0 v_i + g_0, v_i) - V^*(v_i) + \alpha \left\| K_0 v_i + g_0 - U^*(v_i) \right\|^2 \right) \tag{20}$$

$$\textit{subject to} \;\; G\left(K_0 v_i + g_0, v_i\right) \leq 0 \;, \; \forall v_i \in V_0 \tag{21}$$

In (20), $J(K_0 v_i + g_0, v_i)$ is the sub-optimal cost, $V^*(v_i)$ denotes the cost corresponding to the close-to-global solution $U^*(v_i)$, i.e. $V^*(v_i) = J(U^*(v_i), v_i)$, and the parameter α is a weighting coefficient. The details about the generation of a set of points $V_0 = \{v_0,\, v_1,\, v_2,\, \dots,\, v_{N_1}\}$ associated to a given region Z_0 and the computation of a close-to-global solution of problem P2 at these points can be found in [15].

After a linear *regressor* feedback $\widehat{U}_0(\tilde{z}) = K_0\tilde{z} + g_0$ that is feasible on the set $V_0 \subset Z_0$ has been determined, an estimate $\widehat{\varepsilon}_0$ of the maximal cost function approximation error in Z_0 is computed as follows:

$$\widehat{\varepsilon}_0 = \max_{i \in \{0,\,1,\,2,\,\dots\,,\,N_1\}} \left(J(K_0 v_i + g_0,\, v_i) - V^*(v_i)\right) \tag{22}$$

Assume the tolerance $\bar{\varepsilon} > 0$ of the cost function approximation error is given. Denote with S_{Z_0} the volume of a given hyper-rectangular region $Z_0 \subset Z \subset \mathbb{R}^q$, i.e. $S_{Z_0} = \prod_{i=1}^{q} \Delta z^i$, where Δz^i is the size of Z_0 along the dimension z^i. Let $S_{\min}$ be the minimal allowed volume of the regions in the partition of Z. The following algorithm is used to compute the explicit approximate output-feedback NMPC controller on the *regressor* space Z [15]:

Algorithm 1 (Explicit approximate BB-NMPC)

1. Initialize the partition to the whole hyper-rectangle, i.e. $\Pi = \{Z\}$. Mark the hyper-rectangle Z as unexplored.

2. Select any unexplored hyper-rectangle $Z_0 \in \Pi$. If no such hyper-rectangle exists, terminate.

3. Compute a close-to-global solution to problem P2 at the center point v_0 of Z_0. If problem P2 has a feasible solution, go to step 4. Otherwise, go to step 7.

4. Define a set of points $V_0 = \{v_0, v_1, v_2,\, \dots, v_{N_1}\}$ associated to the region Z_0. Compute a close-to-global solution to problem P2 for $\tilde{z}$ fixed to each of the points v_i, $i = 1,\, \dots, N_1$. If problem P2 has a feasible solution at all these points, go to step 5. Otherwise, go to step 7.

5. Compute an affine feedback $\widehat{U}_0(\tilde{z})$ using Procedure 1, as an approximation to be used in Z_0. If a feasible solution of problem P3 was found, go to step 6. Otherwise, go to step 7.

6. Compute an estimate $\widehat{\varepsilon}_0$ of the error bound in Z_0 according to (22). If $\widehat{\varepsilon}_0 \leq \bar{\varepsilon}$, mark Z_0 as *explored* and *feasible* and go to step 2. Otherwise, split Z_0 into two hyper-rectangles Z_1 and Z_2. Mark Z_1 and Z_2 unexplored, remove Z_0 from Π, add Z_1 and Z_2 to Π, and go to step 2.

7. Compute the volume S_{Z_0} of the hyper-rectangle Z_0. If $S_{Z_0} < S_{\min}$, mark Z_0 as *explored* and *infeasible* and go to step 2. Otherwise, split Z_0 into hyper-rectangles $Z_1, \ldots, Z_{N_s}$. Mark $Z_1, \ldots, Z_{N_s}$ unexplored, remove Z_0 from Π, add $Z_1, \ldots, Z_{N_s}$ to Π, and go to step 2.

The heuristic rules used to split a region in steps 6 and 7 of the algorithm can be found in [15]. This algorithm will terminate with a PWL function $\widehat{U}(\tilde{z}) = [\widehat{u}_0(\tilde{z}), \widehat{u}_1(\tilde{z}), \ldots, \widehat{u}_{N-1}(\tilde{z})]$ that is defined on an inner approximation Z_Π of the set $Z \cap Z_f$.

4. Design of feedback control law in a neighborhood of the equilibrium

Generally, it will be difficult to guarantee that the local linearization at a nominal equilibrium point of an NN ARX model is accurate (see the example in section 5). The inaccuracies of the model may result in a steady-state offset of the explicit BB-NMPC controller. Here, a dual-mode control strategy is proposed which aims at achieving an offset-free closed-loop response in the presence of bounded disturbances and/or model errors. With this strategy, the control is performed by the explicit BB-NMPC controller when the system is far from equilibrium, and by a Linear Quadratic Regulator (LQR) with integral action when it is close to equilibrium. In order to design the LQR, a linear ARX model of the system needs to be obtained in a neighborhood of the equilibrium.

4.1. Modelling of dynamic systems with linear ARX models

The purpose is to obtain a linear ARX model [29]:

$$\begin{aligned} y(t+1) = A_1 y(t) + A_2 y(t-1) + \ldots + A_{l+1} y(t-l) + B_1(u(t) - u^*_{st}) \\ + B_2(u(t-1) - u^*_{st}) + \ldots + B_{l+1}(u(t-l) - u^*_{st}) \,, \end{aligned} \tag{23}$$

that will be valid in a neighborhood of the equilibrium $y = 0$, $u = u^*_{st}$ of the considered nonlinear dynamical system (4)–(5). In (23), the matrices $A_i \in \mathbb{R}^{p\times p}$ and $B_i \in \mathbb{R}^{p\times m}$, $i = 1, 2, \ldots, l+1$ contain the coefficients of the model, and l is a given lag. To estimate the parameters of the model (23), the least squares estimation method or the four-stage instrumental variable method can be applied [29].

4.2. Design of Linear Quadratic Regulator with integral action

The purpose is to design an LQR that will regulate the system (23) to the origin. In order to achieve an offset-free performance, the model (23) is augmented with the following output $y_{int} \in \mathbb{R}^p$, which takes into account the integral regulation error:

$$y_{int}(t+1) = y_{int}(t) + T_s y(t) \tag{24}$$

where T_s is the sampling time. Let $u_e(t) \equiv u(t) - u^*_{st}$. Then, the extended system with input u_e and output $y_e = [y,\, y_{int}]$ is described by the linear ARX model:

$$\begin{aligned} y_e(t+1) = A^e_1 y_e(t) + A^e_2 y_e(t-1) + ... + A^e_{l+1} y_e(t-l) + \\ B^e_1 u_e(t) + B^e_2 u_e(t-1) + ... + B^e_{l+1} u_e(t-l)\,, \end{aligned} \tag{25}$$

where $A^e_1 = \begin{bmatrix} A_1 & 0_p \\ T_s I_p & I_p \end{bmatrix}$, $A^e_i = \begin{bmatrix} A_i & 0_p \\ 0_p & 0_p \end{bmatrix}$, $i = 2, 3, \dots, l+1$, $B^e_i = \begin{bmatrix} B_i \\ 0_{p,m} \end{bmatrix}$, $i = 1, 2, \dots, l+1$. Here, I_p is the p-dimensional identity matrix, 0_p is the p-dimensional square zero matrix, and $0_{p,m}$ is the $p \times m$-dimensional zero matrix. The following *regressor* vector is introduced:

$$\tilde{z}_e(t) = \begin{cases} [y_e(t), y_e(t-1), ..., y_e(t-l), \\ u_e(t-1), u_e(t-2), ..., u_e(t-l)], \text{if}\, l > 0 \\ y_e(t)\,,\ \text{if}\ \ l = 0 \end{cases}, \tag{26}$$

This *regressor* vector can be also represented as $\tilde{z}_e(t) = [z_1(t), z_2(t), ..., z_{l+l+1}(t)]$, where $z_1(t), ..., z_{l+1}(t)$ are the shifted values of y_e and $z_{l+2}(t), ..., z_{l+l+1}(t)$ are the shifted values of u_e. The relations between these elements are:

$$\begin{aligned} & y_e(t+1) = z_1(t+1) \\ z_1(t) \quad &= \quad y_e(t) \quad = z_2(t+1) \\ z_2(t) \quad &= \quad y_e(t-1) \quad = z_3(t+1) \\ &\vdots \\ z_l(t) \quad &= y_e(t-l+1) = z_{l+1}(t+1) \\ z_{l+1}(t) &= y_e(t-l) \end{aligned} \tag{27}$$

$$
\begin{aligned}
&\qquad\qquad u_e(t) = z_{l+2}(t+1) \\
z_{l+2}(t) &= u_e(t-1) = z_{l+3}(t+1) \\
z_{l+3}(t) &= u_e(t-2) = z_{l+4}(t+1) \\
&\vdots \\
z_{l+l}(t) &= u_e(t-l+1) = z_{l+l+1}(t+1) \\
z_{l+l+1}(t) &= u_e(t-l)
\end{aligned}
\tag{28}
$$

Then, the system (25) can be represented:

$$\tilde{z}_e(t+1) = \tilde{A}^e \tilde{z}_e(t) + \tilde{B}^e u_e(t) \tag{29}$$

For $l > 0$, the matrices $\tilde{A}^e$ and $\tilde{B}^e$ in (29) are given by:

$$
\tilde{A}^e = \begin{bmatrix}
A_1^e & A_2^e & \dots & A_l^e & A_{l+1}^e & B_2^e & \dots & B_l^e & B_{l+1}^e \\
I_{2p} & 0_{2p} & \dots & 0_{2p} & 0_{2p} & 0_{2p,m} & \dots & 0_{2p,m} & 0_{2p,m} \\
0_{2p} & I_{2p} & \dots & 0_{2p} & 0_{2p} & 0_{2p,m} & \dots & 0_{2p,m} & 0_{2p,m} \\
\vdots & & & & & & & & \\
0_{2p} & 0_{2p} & \dots & I_{2p} & 0_{2p} & 0_{2p,m} & \dots & 0_{2p,m} & 0_{2p,m} \\
0_{m,2p} & 0_{m,2p} & \dots & 0_{m,2p} & 0_{m,2p} & 0_m & \dots & 0_m & 0_m \\
0_{m,2p} & 0_{m,2p} & \dots & 0_{m,2p} & 0_{m,2p} & I_m & \dots & 0_m & 0_m \\
\vdots & & & & & & & & \\
0_{m,2p} & 0_{m,2p} & \dots & 0_{m,2p} & 0_{m,2p} & 0_m & \dots & I_m & 0_m
\end{bmatrix}
\tag{30}
$$

$$\tilde{B}^e = [B_1^e \; 0_{2p,m} \; 0_{2p,m} \; \dots \; 0_{2p,m} \; I_m \; 0_m \; \dots \; 0_m]^T \tag{31}$$

In (30), (31), I_{2p} and I_m are identity matrices, 0_{2p} and 0_m are square zero matrices, and $0_{2p,m}$ and $0_{m,2p}$ are zero matrices with dimensions $2p \times m$ and $m \times 2p$ respectively. If $l = 0$, then $\tilde{A}^e = A_1^e$ and $\tilde{B}^e = B_1^e$.

The unconstrained LQR problem for system (29) solves the following optimization problem:

$$\min_{\{u_e(t),\, u_e(t+1),\, \dots\}} \sum_{k=0}^{\infty} \left[\|\tilde{z}_e(t+k)\|_{Q_e}^2 + \|u_e(t+k)\|_{R_e}^2 \right] \tag{32}$$

where Q_e, $R_e \succ 0$. The solution to (32) is the linear feedback control law:

$$u_e(t+k) = -K\tilde{z}_e(t+k)\,, \; k \geq 0\,, \tag{33}$$

where the controller gain matrix K is given by [30]:

$$K = \left(\tilde{B}^{eT} P \tilde{B}^{e} + R_e\right)^{-1} \tilde{B}^{eT} P \tilde{A}^{e} \tag{34}$$

In (34), P is the solution of the discrete-time algebraic Riccati equation [30]:

$$P = \tilde{A}^{eT} P \tilde{A}^{e} + Q_e - \tilde{A}^{eT} P \tilde{B}^{e} \left(\tilde{B}^{eT} P \tilde{B}^{e} + R_e\right)^{-1} \left(\tilde{A}^{eT} P \tilde{B}^{e}\right)^{T} \tag{35}$$

By taking into account that $u_e(t) \equiv u(t) - u^*_{st}$, it follows from (33) that the control input applied to the system is:

$$u(t+k) = -K\tilde{z}_e(t+k) + u^*_{st}\,,\ \ k \geq 0 \tag{36}$$

4.3. Explicit dual-mode controller

Consider the closed-loop system:

$$\tilde{z}_e(t+k) = (\tilde{A}^e - \tilde{B}^e K)\tilde{z}_e(t+k-1),\ \ k \geq 0\,, \tag{37}$$

where $\tilde{z}_e(t+k)$ is defined by (26) if t is replaced by $t+k$. Further, note that the *regressor* vectors $\tilde{z}(t)$ and $\tilde{z}_e(t)$ (defined by (7) and (26)) are related by:

$$\tilde{z}(t) \ = \ \begin{bmatrix} \Psi_{p,2p} & \dots & 0_{p,2p} & 0_m & \dots & 0_m \\ \vdots & & & & & \\ 0_{p,2p} & \dots & \Psi_{p,2p} & 0_m & \dots & 0_m \\ 0_{p,2p} & \dots & 0_{p,2p} & I_m & \dots & 0_m \\ \vdots & & & & & \\ 0_{p,2p} & \dots & 0_{p,2p} & 0_m & \dots & I_m \end{bmatrix} \tilde{z}_e(t) \tag{38}$$

where $\Psi_{p,2p} = [I_p \ \ 0_p]$, $0_{p,2p}$ is zero matrix with dimensions $p \times 2p$, and I_p, 0_p, I_m, 0_m are defined above. Let $\Gamma_1 = \{\tilde{z} \in \mathbb{R}^q \,|\, -\gamma_1 \leq \tilde{z} \leq \gamma_1\}$ with $\gamma_1 \in \mathbb{R}^q$, $\gamma_1 > 0$ and $\Gamma_2 = \{\tilde{z} \in \mathbb{R}^q \,|\, -\gamma_2 \leq \tilde{z} \leq \gamma_2\}$ with $\gamma_2 \in \mathbb{R}^q$, $\gamma_2 > 0$, be two sets such that $\Gamma_2 \subseteq \Gamma_1 \subset Z_\Pi$ (recall that the explicit approximate BB-NMPC controller is defined on the set Z_Π). Suppose that $\forall \tilde{z}(t) \in \Gamma_2$, $\tilde{z}(t+k)$ (associated to the closed-loop system (37) by the relation (38)) is a sufficiently accurate prediction of the dynamics of the nonlinear system (4)–(5) and the following conditions are satisfied:

$$\tilde{z}(t+k) \in \Gamma_1,\ \ k > 0 \tag{39}$$

$$y_{\min} < [I_p\, 0_p \dots 0_p\, 0_m \dots 0_m]\tilde{z}(t+k) < y_{\max},\ \ k \geq 0 \tag{40}$$

$$u_{\min} < -K\tilde{z}_e(t+k) + u^*_{st} < u_{\max},\ \ k \geq 0 \tag{41}$$

Let $\tilde{z}$ and $\tilde{z}_e$ be the values of the *regressor* vectors (7) and (26) at the current time t. Also, let i_s be a switch index indicating if $\tilde{z}$ has entered the set Γ_2. Then, the explicit dual-mode controller is defined as follows:

$$u_d \triangleq \begin{cases} \widehat{u}_0(\tilde{z})\,,\ i_s = 0\,,\ \text{ if } \tilde{z} \notin \Gamma_1 \\ -K\tilde{z}_e + u^*_{st}\,,\ i_s = 1\,,\ \text{ if } \tilde{z} \in \Gamma_2 \\ -K\tilde{z}_e + u^*_{st}\,,\ \text{ if } i_s = 1 \end{cases} \tag{42}$$

The expression in the first row of (42) means that the control is performed by the explicit BB-NMPC controller when the system is far from equilibrium, while the expressions in the second and third rows imply that the control will be switched to the LQR when $\tilde{z}$ enters the set Γ_2 and the LQR will continue controlling the system afterwords. It should be noted that the integrator output y_{int} is used only when $\tilde{z} \in \Gamma_1$. In the case when $\tilde{z} \notin \Gamma_1$, y_{int} is set to zero and not used. From (40) and (41), it is also observed that the output and input constraints never become active for the closed-loop system (37).

5. Design of an explicit output-feedback NMPC for regulation of a pH maintaining system

The dual-mode approach to explicit output-feedback NMPC, described in the previous two sections, is applied to design an explicit NMPC for regulation of a pH maintaining system. The motivation for this particular example is not to suggest that the mp-NLP approach is particularly suitable for this kind of process, but rather to demonstrate a potential engineering applications of the mp-NLP approach to processes which are modelled with higher order black-box models. Particularly attractive for suggested control method from engineering applications aspect is a benefit to be able to execute the NMPC code in a low-cost PLC type of hardware.

5.1. The pH maintaining system

A simplified schematic diagram of the pH maintaining system taken from [31] is given in Figure B.1. The process consists of an acid stream (Q_1), buffer stream (Q_2) and base stream (Q_3) that are mixed in a tank T_1. Prior to mixing, the acid stream enters the tank T_2. The acid and buffer flow rates are assumed to be constant. The effluent pH is the measured variable, which is controlled by manipulating the base flow rate.

In [31], a dynamic model of the pH maintaining system is derived, which is given in details in Appendix A. The following state, input and output variables are defined [31]:

$$x = [W_{a4}\ W_{b4}\ h_1]^T\ ,\ u = Q_3\ ,\ y = \mathrm{pH}\ , \tag{43}$$

where W_{a4} and W_{b4} are the effluent reaction invariants, and h_1 is the liquid level in tank T_1. The obtained state space model has the form [31]:

$$\dot{x} = \tilde{f}(x) + \tilde{g}(x)u \tag{44}$$

$$c(x, y) = 0\ , \tag{45}$$

where $\tilde{f}(x)$, $\tilde{g}(x)$ and $c(x, y)$ are given in Appendix A.

5.2. ARX model identification

5.2.1. Neural network ARX model identification

The identification and the validation of the NN model of the pH maintaining system is based on simulation data, generated with the model (44)–(45), where the liquid level h_1 in tank T_1 is assumed to be constant. Thus, it is presumed that a controller has been already designed to keep the level h_1 on the nominal value $h_1^* = 14$ [cm] by manipulating the exit flow rate Q_4. To get an idea about the system dynamics, necessary for sampling time and regressor vector selection, some preliminary tests were pursued. The process model (44)–(45) was excited with a combination of step-like signals for estimation of the dominant time constant and settling time. The dominant time constant was estimated in range between 65 [s] and 185 [s] and settling time between 135 [s] and 325 [s]. This 'provisional' dynamics is necessary for the estimation of appropriate sampling time. Based on responses and iterative cut-and-try procedure, a sampling time of 25 [s] was selected for these tests. Based on these preliminary tests, the chosen identification signal (400 samples) was generated from a uniform random distribution and a rate of change of the signal of 50 [s]. The validation signal was obtained using a generator of random noise with uniform distribution and a rate of change of the signal of 500 [s], so it has lower magnitude and frequency components than the identification signal. The rationale behind this is that if the model was identified using a rich signal, then it should respond well to a signal with less components.

The NN model represents a NARX model of the form (7)–(8). The hidden layer has sigmoid activation functions and the output layer has linear activation function. The choice of regressors is a difficult one and is common to all black-box modelling approaches. The number of regressors (delayed inputs and outputs) was determined by the method described in [32]. A trade-off between modelling error and complexity was taken into the account. The final selection was that the system model has the form:

$$y(t+1) = f_{NN}(\tilde{z}(t), u(t), \theta) \tag{46}$$

$$\tilde{z}(t) = [y(t),\ y(t-1),\ y(t-2),\ u(t-1),\ u(t-2)] \tag{47}$$

It should be noted that in difference to the state space model (44)–(45) where $y = \mathrm{pH}$, in the NN model (46)–(47) the variable y represents the deviation of the pH from the desired set point $\mathrm{pH}_{sp} = 4.8$, i.e. $y = \mathrm{pH} - \mathrm{pH}_{sp}$. In general, any other value for pH_{sp} can be pursued if the developed black-box model describes the specified operating range. Also, while in [31] the goal is to keep the pH at value 7 (a pH neutralization system), here the task is to maintain the pH at value 4.8 (a pH maintaining system). The data used for identification of the NN model (46)–(47) and for validation of its performance were scaled to zero mean and variance 1. This means that $u(t)$ and $y(t)$ can take both positive and negative values.

The optimal number of neurons in the hidden layer was determined systematically by optimization in parallel. The network was optimized for each possible number of hidden neurons in a certain range. The Levenberg-Marquardt method was used for minimization of the mean-square error criteria (3), due to its rapid convergence properties and robustness. At the end of this lengthy procedure and after removing the unimportant weights, the optimal parameters of the model (46)–(47) were obtained, with thirteen neurons in the hidden layer. More about systematic network structure selection, pruning and other issues regarding neural networks modelling can be found in various literature describing this topic and its applications (e.g. [6], [25], [32], [33], [34], [35]).

Figure B.2 depicts a comparison between the simulated NN response and the process response to the identification and the validation input signals. From the validation, it can be concluded that the black-box model captures the dynamics of the pH maintaining system relatively well. The resulting black-box model is not too large to be handled and was relatively routinely obtained with the selected software tool.

5.2.2. Linear ARX model identification

The equilibrium point of the pH maintaining system (the model (44)–(45)) is $y = 0$, $u^*_{st} = 0.1732$. A validation of the obtained NN ARX model near this point clearly shows that it is not accurate (see Figure B.3).

In order to obtain accurate predictions when the output variable is close to zero, the following 1-st order linear ARX model is identified:

$$y(t+1) = 0.7704y(t) + 0.0539(u(t) - u_{st}^*) \tag{48}$$

Higher order linear ARX models have been also obtained, however simulations have shown that the dynamics of the pH maintaining system around the equilibrium is captured best by the 1-st order model (48). The simulated response of the ARX model (48) is depicted in Figure B.3.

5.3. Design of explicit dual-mode controller

The approach described in sections 3 and 4 is applied to design an explicit dual-mode controller for the pH maintaining system based on its NN model (46)–(47) and linear ARX model (48). Recall that due to scaling, the variables u and y can take both positive and negative values.

First, Algorithm 1 is applied to design an explicit approximate BB-NMPC controller. The following control input constraint is imposed on the system:

$$-0.4 \leq u \leq 0.4 \tag{49}$$

The horizon is $N = 8$ and the terminal constraint in problem P1 is:

$$y_{t+N|t} \in \Omega \ , \tag{50}$$

where $\Omega = \{y \in \mathbb{R} \,|\, y^2 \leq 0.001\}$. The weighting matrices in the cost function (17) are $Q = 10$, $R = 1$, $P = 10$. The BB-NMPC minimizes the cost function (17) subject to the model (46)–(47) and the constraints (49), (50). In (20), it is chosen $\alpha = 10$. The *regressor* space to be partitioned is defined by $Z = ([-1.2; 1.2] \times [-1.2; 1.2] \times [-1.2; 1.2] \times [-0.4; 0.4] \times [-0.4; 0.4])$. The cost function approximation tolerance is chosen as $\bar{\varepsilon}(Z_0) = \max(\bar{\varepsilon}_a, \, \bar{\varepsilon}_r \min_{\tilde{z} \in Z_0} V^*(\tilde{z}))$, where $\bar{\varepsilon}_a = 0.005$ and $\bar{\varepsilon}_r = 0.1$ are the absolute and the relative tolerances, respectively. The partition has 5512 regions and 23 levels of search. Totally, 33 arithmetic operations are needed in real-time to compute the control input (23 comparisons, 5 multiplications and 5 additions).

Further, an unconstrained LQR is designed, which is used in a neighborhood of the origin. For this purpose, consider the extended linear system, where an integral error is added to the linear ARX model (48):

$$y(t+1) = 0.7704y(t) + 0.0539u_e(t) \tag{51}$$

$$y_{int}(t+1) = y_{int}(t) + T_s y(t) \tag{52}$$

Here, $u_e(t) \equiv u(t) - u^*_{st}$. Thus, we obtain the following system:

$$\tilde{z}_e(t+1) = \tilde{A}^e \tilde{z}_e(t) + \tilde{B}^e u_e(t) \ , \tag{53}$$

which is characterized with *regressor* vector $\tilde{z}_e(t) = y_e(t) = [y(t),\ y_{int}(t)]$ and matrices $\tilde{A}^e = \left[\begin{smallmatrix} 0.7704 & 0 \\ T_s & 1 \end{smallmatrix}\right]$ and $\tilde{B}^e = \left[\begin{smallmatrix} 0.0539 \\ 0 \end{smallmatrix}\right]$. The computed LQR law for the system (53) is:

$$u_e = -K\tilde{z}_e = -k_1 y - k_2 y_{int}\ , \text{ where } K = [0.7994,\ 0.0069] \tag{54}$$

This control law solves the optimization problem (32) with weighting matrices $Q_e = \text{diag}\{10,\ 0.0005\}$, $R_e = 10$.

Then, the explicit dual-mode controller for the pH maintaining system is defined according to (42) with $\Gamma_2 \equiv \Gamma_1 = \{\tilde{z} \in \mathbb{R}^q \,|\, -\gamma_1 \leq \tilde{z} \leq \gamma_1\}$, $\gamma_1 = [0.2\ \ 0.6\ \ 0.7\ \ 0.4\ \ 0.4]$.

Table B.1 shows statistics about the performance decrease ε_V as well as the error ε_u in the control input with the approximate explicit BB-NMPC controller, based on simulations for a set of initial regressor vectors $\tilde{z}_i = [y(t),\ y(t-1),\ y(t-2),\ u(t-1),\ u(t-2)]$, $i = 1, 2, \dots, 7776$. The errors ε_V and ε_u are computed as the relative deviations (measured in percentage) of the sub-optimal cost function and control input value (respectively, $\widehat{V}(\tilde{z})$ and $\widehat{u}(\tilde{z})$) from the optimal ones (respectively, $V^*(\tilde{z})$ and $u^*(\tilde{z})$):

$$\varepsilon_V = \frac{\widehat{V}(\tilde{z}) - V^*(\tilde{z})}{V_{\max}} \times 100\%\ ,\ \ \varepsilon_u = \frac{|\widehat{u}(\tilde{z}) - u^*(\tilde{z})|}{u_{\max} - u_{\min}} \times 100\% \tag{55}$$

In (55), $V_{\max} = \max\limits_{i \in \{1,2,\dots,7776\}} V^*(\tilde{z}_i)$.

In order to study the robustness of the explicit dual-mode controller against model inaccuracies, its performance is simulated in closed-loop with the first-principles model (44)–(45). Further, it is assumed that there are persistent disturbances in the acid and the buffer flow rates, which have the following values $\tilde{Q}_1 = 16.8$ [ml/s], $\tilde{Q}_2 = 0.53$ [ml/s] (different from the nominal values $Q^*_1 = 16.6$ [ml/s], $Q^*_2 = 0.55$ [ml/s]). In addition to the explicit dual-mode controller which maintains the pH on the required set point, a second controller (an LQR) is applied, which keeps the liquid level h_1 on the nominal value $h^*_1 = 14$ [cm] by manipulating the exit flow rate Q_4. The obtained trajectories of the control input u and the output variable y are shown in Figure B.4, while the trajectories of the exit flow rate Q_4 and the liquid level h_1 are depicted in Figure B.5.

It can be seen from Figure B.4 that the output variable is steered to the origin despite of the presence of persistent disturbances and the control input achieves a new equilibrium value $\tilde{u}_{st} = 0.2380$ (recall that the equilibrium value corresponding to the nominal model parameters is $u^*_{st} = 0.1732$). It would be necessary to distinguish how the exact NMPC and the approximate explicit NMPC trajectories in Figures B.4 and B.5 are obtained. The exact NMPC response is computed by solving at each time instant of an open-loop NMPC problem formulated for the first-principles model (44)–(45). In contrast, the approximate explicit NMPC solution is first computed off-line as an approximation to problem P1, in which the NN ARX model by itself represents another approximation. Then, its performance is simulated in closed-loop with the first-principles model (44)–(45). Thus, the performance degradation far from the origin is due to the approximations in the model and in the NMPC solution, while near the origin it is related to the use of LQR (pursuing an offset-free response) which differs from the exact NMPC (where no integral action is taken). It also should be noted that the response depicted in Figures B.4 and B.5 has a typical amount of performance degradation being representative for other initial conditions and scenarios.

6. Conclusions

In this paper, an approximate mp-NLP approach to explicit solution of output-feedback NMPC problems for constrained nonlinear systems, based on black-box models, is developed. In particular, neural network ARX models are considered, but the approach can be easily applied to nonlinear systems modelled by other types of black-box models. The explicit controller employs a dual-mode control concept in order to avoid the steady state offset. The approach is illustrated by designing an explicit dual-mode controller for a pH maintaining system. Simulation results show that the dual-mode control concept achieves a strict regulation of the output variable to the origin. The off-line computational complexity with the suggested approach could be circumvented by the application of parallel processing techniques [36].

References

[1] D.Q. Mayne, H. Michalska, Receding horizon control of nonlinear systems, IEEE Transactions on Automatic Control. 35 (1990) 814–824.

[2] F. Allgöwer, A. Zheng, Nonlinear Model Predictive Control, Progress in System and Control Theory, Vol.26, Birkhäuser Verlag, Basel, 2000.

[3] B. Kouvaritakis, M. Cannon (Eds.), Nonlinear Predictive Control, Theory and Practice, IEE Control Engineering Series, vol.61, IEE, 2001.

[4] M. Diehl, H.J. Ferreau, N. Haverbeke, Efficient numerical methods for nonlinear MPC and moving horizon estimation, in: L. Magni, D. M. Raimondo and F. Allgöwer (Eds.), Nonlinear Model Predictive Control: Towards New Challenging Applications, Lecture Notes in Control and Information Sciences, vol.384, Springer-Verlag, Berlin/Heidelberg, Germany, 2009, pp.391–418.

[5] V.M. Zavala, L.T. Biegler, The advanced-step NMPC controller: Optimality, stability and robustness, Automatica. 45 (2009) 86–93.

[6] M. Nørgaard, O. Ravn, N.K. Poulsen, L.K. Hansen, Neural Networks for Modelling and Control of Dynamic Systems, Springer, London, 2000.

[7] J. Espinosa, J. Vandewalle, V. Wertz, Fuzzy Logic, Identification and Predictive Control, Springer-Verlag London, 2005.

[8] S. Garcia-Nieto, M. Martinez, X. Blasco, J. Sanchis, Nonlinear predictive control based on local model networks for air management in diesel engines, Control Engineering Practice. 16 (2008) 1399–1413.

[9] J. Kocijan, R. Murray-Smith, Nonlinear predictive control with a Gaussian process model, in: R. Murray-Smith, R. Shorten (Eds.), Switching and Learning in Feedback Systems, Lecture Notes in Computer Science, vol.3355, Springer-Verlag, Heidelberg, Germany, 2005, pp.185–200.

[10] A. Bemporad, M. Morari, V. Dua, E.N. Pistikopoulos, The explicit linear quadratic regulator for constrained systems, Automatica. 38 (2002) 3–20.

[11] A. Alessio, A. Bemporad, A survey on explicit model predictive control, in: L. Magni, D. M. Raimondo and F. Allgöwer (Eds.), Nonlinear Model Predictive Control: Towards New Challenging Applications, Lecture Notes in Control and Information Sciences, vol.384, Springer-Verlag, Berlin/Heidelberg, Germany, 2009, pp.345–370.

[12] M. Bacic, M. Cannon, B. Kouvaritakis, Constrained NMPC via state-space partitioning for input affine non-linear systems, International Journal of Control. 76 (2003) 1516–1526.

[13] T.A. Johansen, On multi-parametric nonlinear programming and explicit nonlinear model predictive control, Proceedings of the IEEE Conference on Decision and Control, 2002, Las Vegas, vol.3, pp.2768–2773.

[14] T.A. Johansen, Approximate explicit receding horizon control of constrained nonlinear systems, Automatica. 40 (2004) 293–300.

[15] A. Grancharova, T.A. Johansen, P. Tøndel, Computational aspects of approximate explicit nonlinear model predictive control, in: R. Findeisen, F. Allgöwer, L. Biegler (Eds.), Assessment and Future Directions of Nonlinear Model Predictive Control, Lecture Notes in Control and Information Sciences, vol.358, Springer-Verlag, Berlin/Heidelberg, Germany, 2007, pp.181–192.

[16] A.V. Fiacco, Introduction to Sensitivity and Stability Analysis in Nonlinear Programming, Academic Press, Orlando, Florida, 1983.

[17] A. Bemporad, C. Filippi, An algorithm for approximate multiparametric convex programming, Computational Optimization and Applications. 35 (2006) 87–108.

[18] E.N. Pistikopoulos, M.C. Georgiadis, V. Dua, Multi-parametric Programming: Theory, Algorithms, and Applications, Wiley-VCH, 2007.

[19] A. Grancharova, J. Kocijan, T.A. Johansen, Explicit stochastic nonlinear predictive control based on Gaussian process models, Proceedings of European Control Conference, Kos, Greece, 2007, pp.2340–2347.

[20] T. Parisini, R. Zoppoli, A receding-horizon regulator for nonlinear systems and a neural approximation, Automatica. 31 (1995) 1443–1451.

[21] T. Parisini, S. Sacone, Stable hybrid control based on discrete-event automata and receding-horizon neural regulators, Automatica. 37 (2001) 1279–1292.

[22] D.P. Bertsekas, J.N. Tsitsiklis, Neuro-Dynamic Programming, Athena Scientific, Belmont, MA, 1998.

[23] B.M. Åkesson, H.T. Toivonen, A neural network model predictive controller, Journal of Process Control. 16 (2006) 937–946.

[24] S. Chen, S.A. Billings, Representation of non-linear systems: The NARMAX model, International Journal of Control. 49 (1989) 1013–1032.

[25] S. Chen, S.A. Billings, P.M. Grant, Non-linear system identification using neural networks, International Journal of Control. 51 (1990) 1191–1214.

[26] H. Michalska, D.Q. Mayne, Robust receding horizon control of constrained nonlinear systems, IEEE Transactions on Automatic Control. 38 (1993) 1623–1633.

[27] J. Sjöberg, Q. Zhang, L. Ljung, A. Benveniste, B. Delyon, P.Y. Glorennec, H. Hjalmarsson, A. Juditsky, Non-linear black-box modeling in system identification: A unified overview, Automatica. 31 (1995) 1691–1724.

[28] D.Q. Mayne, J.B. Rawlings, C.V. Rao, P.O.M. Scokaert, Constrained model predictive control: Stability and optimality, Automatica. 36 (2000) 789–814.

[29] L. Ljung, T. Glad, Modeling of Dynamic Systems, Prentice Hall, Englewood Cliffs, N.J., 1994.

[30] K. Ogata, Discrete-time control systems, Englewood Cliffs, Prentice-Hall, N.J., 1995.

[31] M.A. Henson, D.E. Seborg, Adaptive nonlinear control of a pH neutralization process, IEEE Transactions on Control System Technology. 2 (1994) 169–183.

[32] X. He, H. Asada, A new method for identifying orders of input-output models for nonlinear dynamic systems, Proceedings of the American Control Conference, San Francisco, CA, 1993, pp.2520–2523.

[33] J. Han, C. Moraga, S. Sinne, Optimization of feedforward neural networks, Engineering Applications of Artificial Intelligence. 9 (1996) 109–119.

[34] J.C. Principe, N.R. Euliano, W.C. Lefebvre, Neural and Adaptive systems, John Wiley & Sons, New York, 2000.

[35] Neural network applications in chemical engineering, Special issue of Computers & Chemical Engineering, Eds. V. Venkatasubramanian, T.J. McAvoy, volume 16, Issue 4, 1992.

[36] R. Trobec, M. Vajteršic, P. Zinterhof (Eds.), Parallel Computing: Numerics, Applications, and Trends, Springer-Verlag, London, 2009.

Appendix A. First-principles model of the pH maintaining system

The dynamic model of the pH maintaining system is derived using conservation equations and equilibrium relations [31]. The model also includes hydraulic relationships for the tank outlet flows. Modeling assumptions include perfect mixing, constant density, and complete solubility of the ions involved. The model is presented briefly below according to [31].

The chemical reactions for the system are:

$$\begin{aligned} \mathrm{H_2CO_3} &\longleftrightarrow \mathrm{HCO_3^-} + \mathrm{H^+} &\quad (A.1)\\ \mathrm{HCO_3^-} &\longleftrightarrow \mathrm{CO_3^=} + \mathrm{H^+} &\quad (A.2)\\ \mathrm{H_2O} &\longleftrightarrow \mathrm{OH^-} + \mathrm{H^+} &\quad (A.3)\end{aligned}$$

The corresponding equilibrium constants are:

$$K_{a1} = \frac{[\mathrm{HCO_3^-}][\mathrm{H^+}]}{[\mathrm{H_2CO_3}]}\ ,\ K_{a2} = \frac{[\mathrm{CO_3^=}][\mathrm{H^+}]}{[\mathrm{HCO_3^-}]}\ ,\ K_w = [\mathrm{H^+}][\mathrm{OH^-}] \quad (A.4)$$

The chemical equilibria is modelled by defining two reaction invariants for each of the streams Q_i, $i \in \{1, 2, 3, 4\}$ [31]:

$$\begin{aligned} W_{ai} &= [\mathrm{H^+}]_i - [\mathrm{OH^-}]_i - [\mathrm{HCO_3^-}]_i - 2[\mathrm{CO_3^=}]_i &\quad (A.5)\\ W_{bi} &= [\mathrm{H_2CO_3}]_i + [\mathrm{HCO_3^-}]_i + [\mathrm{CO_3^=}]_i &\quad (A.6)\end{aligned}$$

The invariant W_a is a charge related quantity, while W_b represents the concentration of the $\mathrm{CO_3^=}$ ion. The pH can be determined from W_a and W_b using the following relations [31]:

$$W_b \frac{\frac{K_{a1}}{[\mathrm{H^+}]} + \frac{2K_{a1}K_{a2}}{[\mathrm{H^+}]^2}}{1 + \frac{K_{a1}}{[\mathrm{H^+}]} + \frac{K_{a1}K_{a2}}{[\mathrm{H^+}]^2}} + W_a + \frac{K_w}{[\mathrm{H^+}]} - [\mathrm{H^+}] = 0 \quad (A.7)$$

$$\mathrm{pH} = -\log([\mathrm{H^+}]) \quad (A.8)$$

In [31], a simplified model of the pH maintaining system is developed, where the dynamics of the pH transmitter and the flow dynamics of tank T_2 are neglected. The mass balance on tank T_1 yields:

$$A_1 \frac{dh_1}{dt} = Q_{1e} + Q_2 + Q_3 - Q_4 \ , \tag{A.9}$$

where h_1 is the liquid level and A_1 is the cross-sectional area of tank T_1. The exit flow rate Q_4 is modelled as:

$$Q_4 = C_v (h_1 + l)^s \ , \tag{A.10}$$

where C_v is a constant valve coefficient, s is a constant valve exponent, and l is the vertical distance between the bottom of tank T_1 and the outlet for Q_4. By combining mass balances on each of the ionic species in the system, the following differential equations for the effluent reaction invariants W_{a4} and W_{b4} are derived [31]:

$$A_1 h_1 \frac{dW_{a4}}{dt} = Q_{1e}(W_{a1} - W_{a4}) + Q_2(W_{a2} - W_{a4}) + Q_3(W_{a3} - W_{a4}) \tag{A.11}$$
$$A_1 h_1 \frac{dW_{b4}}{dt} = Q_{1e}(W_{b1} - W_{b4}) + Q_2(W_{b2} - W_{b4}) + Q_3(W_{b3} - W_{b4}) \tag{A.12}$$

Based on the above relations, a state space model of the pH maintaining system is obtained by defining the following state, input and output variables:

$$x = [W_{a4} \ W_{b4} \ h_1]^T \ , \ u = Q_3 \ , \ y = \mathrm{pH} \tag{A.13}$$

The state space model has the form [31]:

$$\dot{x} = \tilde{f}(x) + \tilde{g}(x) u \tag{A.14}$$
$$c(x, y) = 0 \ , \tag{A.15}$$

where:

$$\tilde{f}(x) = \begin{bmatrix} \frac{Q_1(W_{a1} - x_1) + Q_2(W_{a2} - x_1)}{A_1 x_3} \\ \frac{Q_1(W_{b1} - x_2) + Q_2(W_{b2} - x_2)}{A_1 x_3} \\ \frac{Q_1 - C_v(x_3 + l)^s + Q_2}{A_1} \end{bmatrix} \ , \ \tilde{g}(x) = \begin{bmatrix} \frac{W_{a3} - x_1}{A_1 x_3} \\ \frac{W_{b3} - x_2}{A_1 x_3} \\ \frac{1}{A_1} \end{bmatrix} \tag{A.16}$$

$$c(x, y) = x_1 + 10^{y-14} - 10^{-y} + \frac{x_2(1 + 2 \times 10^{y - pK_2})}{1 + 10^{pK_1 - y} + 10^{y - pK_2}} \tag{A.17}$$

The relation between the constants K_{a1}, K_{a2} in (A.7) and the constants K_1, K_2 in (A.17) is:

$$K_{a1} = 10^{-pK_1} \ , \ \ K_{a2} = 10^{-pK_2} \ , \ \ p > 0 \ . \tag{A.18}$$

The parameters of the model (A.14)–(A.18) are given in [31].

Appendix B. Tables and figures

List of Tables

List of Figures

Table B.1:

	Error in cost function ε_V, %	Error in control input ε_u, %
Average	0.068	0.991
Maximal	5.219	9.089

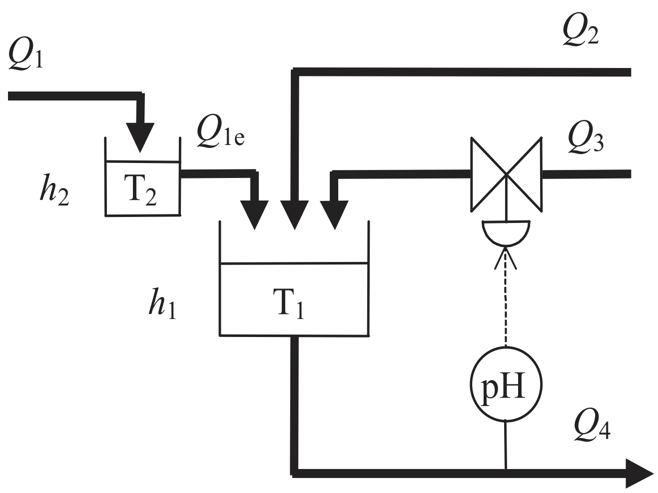
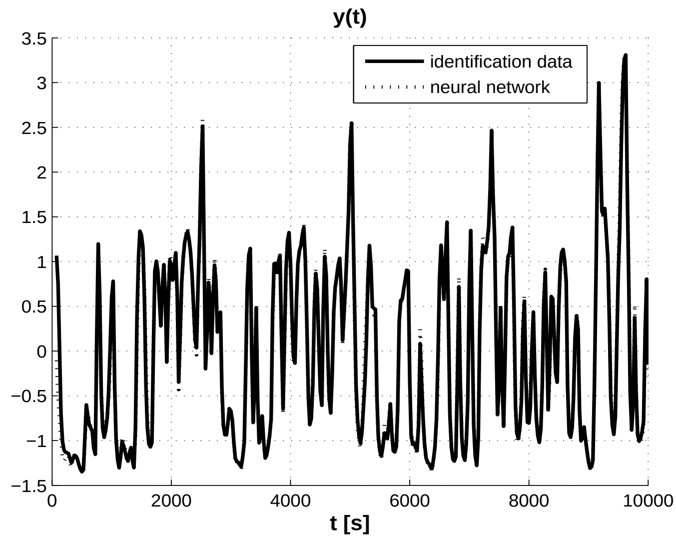


Figure B.1:


y(t)
identification data
neural network
3.5
3
2.5
2
1.5
1
0.5
0
−0.5
−1
−1.5
0
2000
4000
6000
8000
10000
t [s]

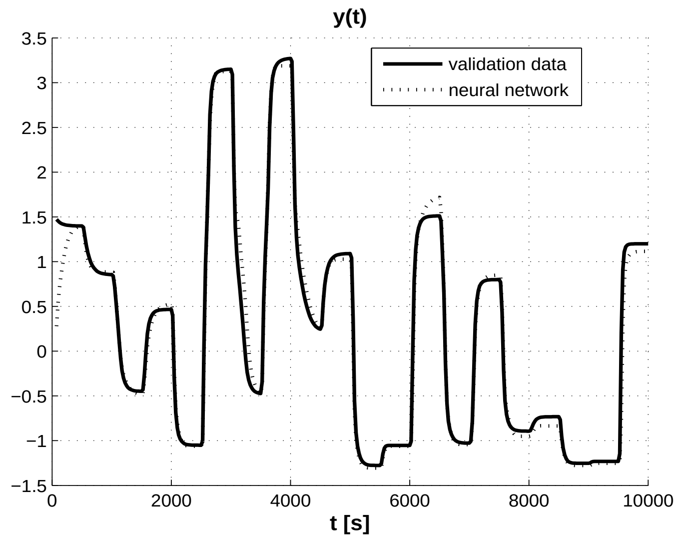

y(t)
validation data
neural network
3.5
3
2.5
2
1.5
1
0.5
0
−0.5
−1
−1.5
0
2000
4000
6000
8000
10000
t [s]

Figure B.2:

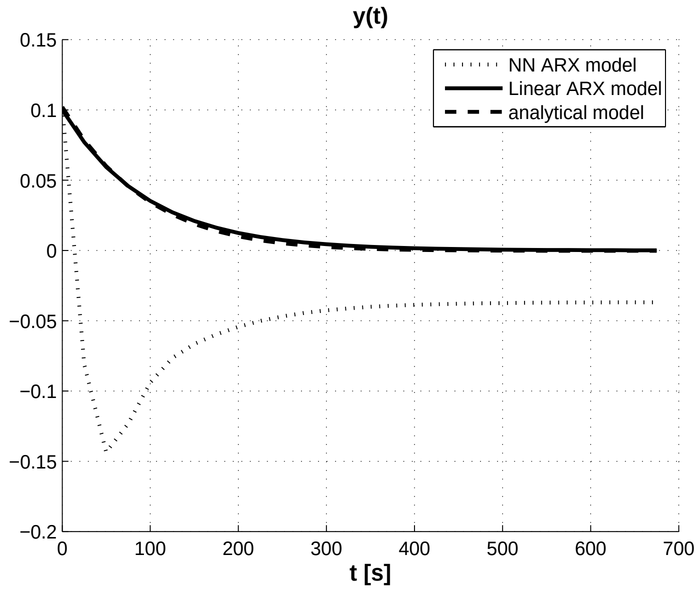
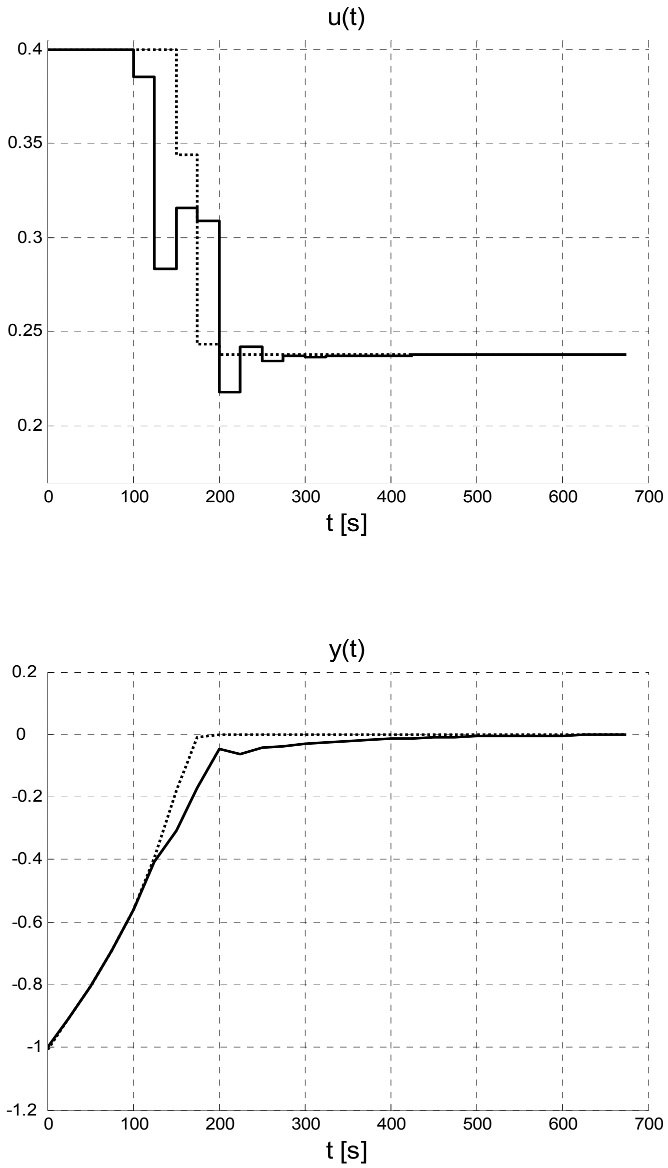
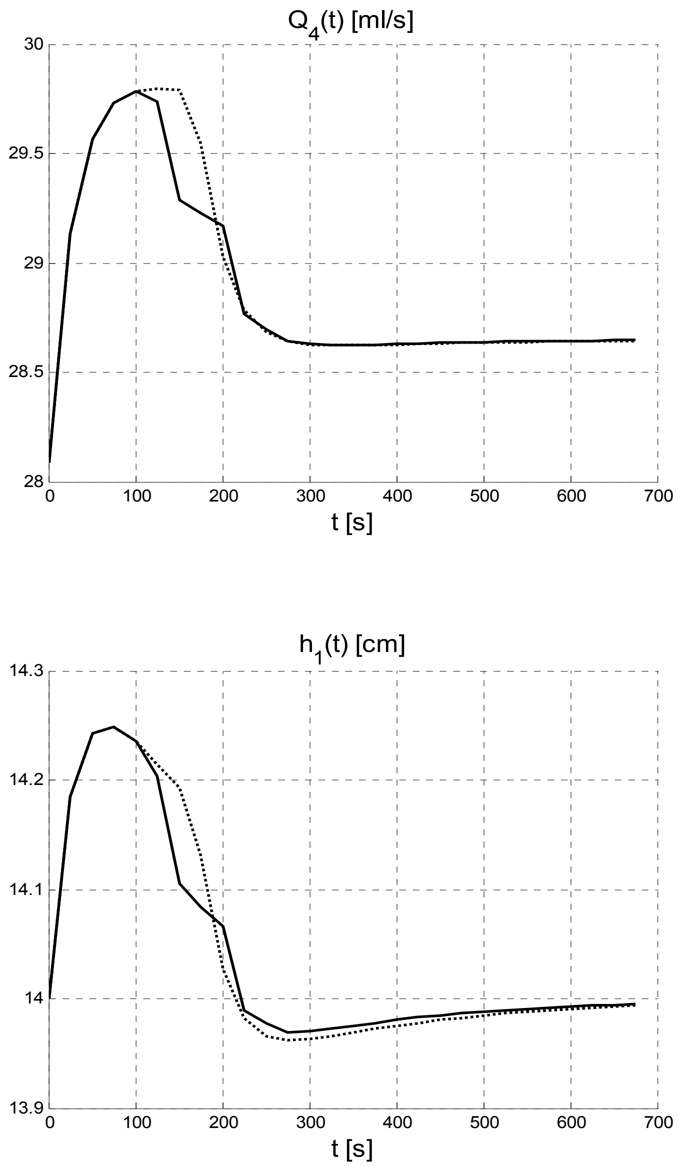

y(t)
NN ARX model
Linear ARX model
analytical model
0.15
0.1
0.05
0
−0.05
−0.1
−0.15
−0.2
0
100
200
300
400
500
600
700
t [s]

Figure B.3:



u(t)
0.4
0.35
0.3
0.25
0.2
0
100
200
300
400
500
600
700
t [s]
y(t)
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-1.2
0
100
200
300
400
500
600
700
t [s]

Figure B.4:



$Q_4(t)$ [ml/s]
30
29.5
29
28.5
28
0
100
200
300
400
500
600
700
t [s]
$h_1(t)$ [cm]
14.3
14.2
14.1
14
13.9
0
100
200
300
400
500
600
700
t [s]

Figure B.5: