Chapter 8
Explicit NMPC Based on Neural Network Models

Abstract. This chapter considers an approximate mp-NLP approach to explicit solution of deterministic NMPC problems for constrained nonlinear systems described by neural network NARX models. The approach builds an orthogonal search tree structure of the *regressor* space partition and consists in constructing a piecewise linear (PWL) approximation to the optimal control sequence. 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 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 approximate mp-NLP approach and the dual-mode approach are applied to design an explicit output-feedback NMPC for regulation of a pH maintaining system.

8.1 Introduction

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 Chapters 3, 4, 5, 6 and in Section 7.2, approaches to explicit solution of NMPC problems based on *first-principles* models were presented, which assume that the state variables can be measured.

Alternatively, there exists a number of NMPC approaches based on various *black-box* models e.g. based on neural network models (e.g. [16, 22]), fuzzy models (e.g. [12]), local model networks (e.g. [18]), Gaussian process models (e.g. [11]). 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

A. Grancharova & T.A. Johansen: Explicit Nonlinear Model Predictive Ctrl., LNCIS 429, pp. 187–207.
springerlink.com

be a simple function evaluation. Thus, in Section 7.3, 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) was described.

In this chapter, the approximate mp-NLP approach [5, 6] to explicit solution of deterministic NMPC problems for constrained nonlinear systems described by neural network NARX models [2] is considered. The NMPC problem based on neural network model will be referred to as NN-NMPC problem. The approach builds an orthogonal search tree structure of the *regressor* space partition and consists in constructing a piecewise linear (PWL) approximation to the optimal control sequence. 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 [15] (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.

8.2 Formulation of the NN-NMPC Problem as an mp-NLP Problem

8.2.1 Modeling of Dynamic Systems with Neural Networks

The black-box identification of nonlinear systems is an area which is quite diverse. It covers topics from mathematical approximation theory, estimation theory, nonparametric regression and concepts like neural networks, fuzzy models, wavelets etc. A unified overview of this topic is given in [20].

Consider a nonlinear dynamical system with input $u \in \mathbb{R}^m$ and output $y \in \mathbb{R}^p$ and let $U = [u(1), u(2), \ldots, u(M)]$ and $Y = [y(1), y(2), \ldots, y(M)]$ 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 neural network NARX model [2], where the future predicted output $y(i+1)$ depends on previous estimated outputs, as well as on previous control inputs:

$$y(i+1) = f(z(i), \theta) \tag{8.1}$$

$$z(i) = [y(i), y(i-1), \ldots, y(i-L), u(i), u(i-1), \ldots, u(i-L)] \tag{8.2}$$

Here, L is a given lag, i denotes the consecutive index of data samples ($i \geq L$), $z(i)$ is the *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. [16]) 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 modeled dynamic system. It should be noted that the results given in [5, 6] 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:

$$E = \frac{1}{2M} \sum_{i=1}^{M} \|y(i) - \hat{y}(i|\theta)\|^2 \tag{8.3}$$

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 [16] for more details.

8.2.2 Formulation of the NN-NMPC Problem

Consider the discrete-time nonlinear system:

$$x(t+1) = h(x(t), u(t)) \tag{8.4}$$

$$y(t) = g(x(t), u(t)) \tag{8.5}$$

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 (8.4)–(8.5):

$$u_{\min} \leq u(t) \leq u_{\max} \ , \ y_{\min} \leq y(t) \leq y_{\max} \tag{8.6}$$

Assume that the dynamics of the nonlinear system (8.4)–(8.5) is approximated with an MLP neural network with NARX structure of the form (8.1)–(8.2). Then for $t \geq L$, define a *modified regressor* vector:

$$\tilde{z}(t) = \begin{cases} [y(t), y(t-1), \ldots, y(t-L), u(t-1), \ldots, u(t-L)], \text{ if } L > 0 \\ y(t), \text{ if } L = 0 \end{cases}, \tag{8.7}$$

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

$$\hat{y}(t+1|\theta) = f_{NN}(\tilde{z}(t), u(t), \theta), \tag{8.8}$$

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 $u(t+k) = u_{t+k}$, $k = 0, 1, \ldots, N-1$ are given. Then, the model (8.8) can be used to obtain the predicted output $y_{t+k+1|t}$, $k = 0, 1, \ldots, N-1$ through iterative one-step ahead predictions, where at each step the predicted output value is fed back to the *regressor* vector:

$$y_{t+k+1|t} = f_{NN}(\tilde{z}_{t+k|t}, u_{t+k}, \theta) \tag{8.9}$$

$$\tilde{z}_{t+k|t} = \begin{cases} [y_{t+k|t}, y_{t+k-1|t}, \ldots, y_{t+k-L|t}, u_{t+k-1}, \ldots, u_{t+k-L}], \text{ if } L > 0 \\ y_{t+k|t}, \text{ if } L = 0 \end{cases} \tag{8.10}$$

The following assumptions are made [5, 6]:

Assumption 8.1. *There exists $u_{st}^{NN} \in \mathbb{R}^m$ satisfying $u_{\min} \leq u_{st}^{NN} \leq u_{\max}$, and such that $f_{NN}(\tilde{z}_0, u_{st}^{NN}, \theta) = 0$, where $\tilde{z}_0$ is obtained from (8.10) with $y_{t+k|t} = y_{t+k-1|t} = \ldots = y_{t+k-L|t} = 0$, $u_{t+k-1} = \ldots = u_{t+k-L} = u_{st}^{NN}$.*

Assumption 8.2. $y_{\min} < 0 < y_{\max}$.

Assumption 8.1 means that the point $y = 0$, $u = u_{st}^{\mathrm{NN}}$, is an equilibrium point for the NARX model (8.8), and Assumption 8.2 means that it is feasible for (8.6).

We consider the optimal regulation problem where the goal is to steer the output variable y to the origin by minimizing a 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 NN-NMPC solves the following optimization problem [5, 6]:

Problem 8.1:

$$V^*(\tilde{z}(t)) = \min_U J(U, \tilde{z}(t)) \tag{8.11}$$

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

$$y_{\min} \le y_{t+k|t} \le y_{\max},\; k = 1, \ldots, N \tag{8.12}$$

$$u_{\min} \le u_{t+k} \le u_{\max},\; k = 0, 1, \ldots, N-1 \tag{8.13}$$

$$\tilde{z}^c_{t+N|t} \in \Omega \tag{8.14}$$

$$y_{t+k+1|t} = f_{NN}(\tilde{z}_{t+k|t}, u_{t+k}, \theta),\; k = 0, 1, \ldots, N-1 \tag{8.15}$$

$$\tilde{z}_{t+k|t} = \begin{cases} [y_{t+k|t}, y_{t+k-1|t}, \ldots, y_{t+k-L|t}, u_{t+k-1}, \ldots, u_{t+k-L}], \text{if} L > 0 \\ y_{t+k|t}, \text{if} L = 0, \end{cases} \quad k = 0, 1, \ldots, N-1 \tag{8.16}$$

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

$$J(U, \tilde{z}(t)) = \sum_{k=0}^{N-1} \left[\|y_{t+k|t}\|_Q^2 + \|u_{t+k} - u_{st}^{\mathrm{NN}}\|_R^2\right] + \|y_{t+N|t}\|_F^2 \tag{8.17}$$

and $\tilde{z}^c = \tilde{z} - [0^T_{(L+1)p}\; u_{st}^{NN^T} \;\ldots\; u_{st}^{NN^T}]^T$, where $0_{(L+1)p}$ is a zero vector with dimension $(L+1)p$. In (8.17), N is a finite horizon and Q, R, $F \succ 0$. In (8.14), Ω is the terminal set defined by $\Omega = \{\tilde{z}^c \in \mathbb{R}^q \,|\, \|\tilde{z}^c\|^2 \le \delta\}$ with $\delta > 0$. From a stability point of view it is desirable to choose δ as small as possible [14]. 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 8.1 can be formulated in a compact form as follows [5, 6]:

Problem 8.2:

$$V^*(\tilde{z}) = \min_U J(U, \tilde{z}) \;\text{ subject to }\; G(U, \tilde{z}) \le 0 \tag{8.18}$$

The NN-NMPC problem defines an mp-NLP, since it is an NLP in U parameterized by $\tilde{z}$. We remark that the constraints function $G(U, \tilde{z})$ in (8.18) is implicitly defined by (8.12)–(8.16), and that all equality constraints are eliminated due to the direct single shooting strategy. An optimal solution to this problem is denoted $U^* = [u_t^*, u_{t+1}^*, \ldots, u_{t+N-1}^*]$ 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:

$$Z_f = \{\tilde{z} \in \mathbb{R}^q \,|\, G(U, \tilde{z}) \le 0 \text{ for some } U \in \mathbb{R}^{Nm}\} \tag{8.19}$$

In parametric programming problems one seeks the solution $U^*(\tilde{z})$ as an explicit function of the parameters $\tilde{z}$ in some set $\underline{Z} \subseteq Z_f \subseteq \mathbb{R}^q$ [3].

8.3 Approximate mp-NLP Approach to Explicit NN-NMPC

In [5, 6], an approximate mp-NLP approach is proposed to explicitly solve the *output-feedback* NN-NMPC problem formulated in the previous section. It is similar to the approximate mp-NLP approach to explicit solution of *state-space* NMPC problems. Let $Z \subset \mathbb{R}^q$ be a hyper-rectangle where we seek to approximate the optimal solution $U^*(\tilde{z})$ to Problem 8.2. It is required that the *regressor* space partition is orthogonal and can be represented as a $k-d$ tree. The idea of the approximate mp-NLP approach is to construct a 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 [5, 6]. First, a close-to-global solution of Problem 8.2 is computed at a set of points $V_0 = \{v_0, v_1, v_2, \dots, v_{N_1}\} \subset Z_0$. 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 $U^*(\tilde{z})$, valid in the whole hyper-rectangle Z_0, is determined by applying the following procedure [5, 6]:

Procedure 8.1 (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:*

$$\min_{K_0, g_0} \sum_{i=0}^{N_1} \left(J(K_0 v_i + g_0, v_i) - V^*(v_i) + \alpha \|K_0 v_i + g_0 - U^*(v_i)\|^2\right) \tag{8.20}$$

$$\textit{subject to } \; G(K_0 v_i + g_0, v_i) \leq 0 \,, \; \forall v_i \in V_0 \tag{8.21}$$

In (8.20), $J(K_0 v_i + g_0, v_i)$ 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 (tuned in an ad-hoc fashion). Note that the computed linear *regressor* feedback $\widehat{U}_0(\tilde{z}) = K_0\tilde{z} + g_0$ satisfies the constraints in Problem 8.2 only at the discrete set of points $V_0 \subset Z_0$. After the feedback $\widehat{U}_0(\tilde{z})$ has been determined, an estimate $\widehat{\varepsilon}_0$ of the maximal cost function approximation error in Z_0 is computed as follows:

$$\widehat{\varepsilon}_0 = \max_{i \in \{0,1,2,\dots,N_1\}} \left(J(K_0 v_i + g_0, v_i) - V^*(v_i)\right) \tag{8.22}$$

If $\widehat{\varepsilon}_0 > \bar{\varepsilon}$, where $\bar{\varepsilon} > 0$ is the specified tolerance of the approximation error, the region Z_0 is divided and the procedure is repeated for the new regions. The approximate PWL *regressor* feedback law is found by applying the approximate mp-NLP algorithm, described in Section 1.1.5.2. The mp-NLP algorithm terminates with a PWL function $\widehat{U}(\tilde{z}) = [\widehat{u}_0(\tilde{z}), \widehat{u}_1(\tilde{z}), \dots, \widehat{u}_{N-1}(\tilde{z})]$ that is defined on an inner approximation Z_Π of the set $Z \cap Z_f$.

8.4 Design of Explicit Dual-Mode Controller

Generally, it will be difficult to guarantee that the local linearization at a nominal equilibrium point of an NN ARX model is accurate. The inaccuracies of the model may result in a steady-state offset of the explicit NN-NMPC controller. In [5, 6], 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 NN-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.

8.4.1 Design of LQR with Integral Action in a Neighborhood of the Equilibrium

Consider a linear ARX model ([13]):

$$\begin{aligned} y(t+1) = A_1 y(t) + A_2 y(t-1) + \ldots + A_{l+1} y(t-l) + B_1(u(t) - u^*_{st}) + \\ B_2(u(t-1) - u^*_{st}) + \ldots + B_{l+1}(u(t-l) - u^*_{st}), \end{aligned} \tag{8.23}$$

that is valid in a neighborhood of the equilibrium $y = 0$, $u = u^*_{st}$ of the considered nonlinear dynamical system (8.4)–(8.5). In (8.23), 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 (8.23), the least squares estimation method or the four-stage instrumental variable method can be applied ([13]). Based on the linear ARX model, an LQR that will regulate the system (8.23) to the origin, is designed. In order to achieve an offset-free performance, the model (8.23) is augmented with the following output $y_{int} \in \mathbb{R}^p$, which takes into account the integral error:

$$y_{int}(t+1) = y_{int}(t) + T_s y(t) \tag{8.24}$$

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:

$$\begin{aligned} y_e(t+1) = A^e_1 y_e(t) + A^e_2 y_e(t-1) + \ldots + A^e_{l+1} y_e(t-l) + \\ B^e_1 u_e(t) + B^e_2 u_e(t-1) + \ldots + B^e_{l+1} u_e(t-l), \end{aligned} \tag{8.25}$$

where $A^e_1 = \begin{bmatrix} A_1 & 0_p \\ T_s I_p & I_p \end{bmatrix}, A^e_i = \begin{bmatrix} A_i & 0_p \\ 0_p & 0_p \end{bmatrix}, i = 2, 3, \ldots, l+1, B^e_i = \begin{bmatrix} B_i \\ 0_{p,m} \end{bmatrix}, i = 1, 2, \ldots, l+1$. 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 [5, 6]:

$$\tilde{z}_e(t) = \begin{cases} [y_e(t), y_e(t-1), \ldots, y_e(t-l), u_e(t-1), u_e(t-2), \ldots, u_e(t-l)], \text{ if } l > 0 \\ y_e(t), \text{ if } l = 0 \end{cases} \tag{8.26}$$

Thus, $\tilde{z}_e(t) \in \mathbb{R}^{q_e}$ with $q_e = (l+1)2p + lm$. This vector can also be represented as $\tilde{z}_e(t) = [z_1(t), z_2(t), \ldots, z_{l+l+1}(t)]$, where $z_1(t), \ldots, z_{l+1}(t)$ are the shifted values of y_e and $z_{l+2}(t), \ldots, z_{l+l+1}(t)$ are the shifted values of u_e. The following relations hold [5, 6]:

$$\begin{aligned} & y_e(t+1) = z_1(t+1) \\ z_1(t) &= y_e(t) = z_2(t+1) \\ z_2(t) &= y_e(t-1) = z_3(t+1) \\ &\vdots \\ z_l(t) &= y_e(t-l+1) = z_{l+1}(t+1) \\ z_{l+1}(t) &= y_e(t-l) \end{aligned} \tag{8.27}$$

$$\begin{aligned} & u_e(t) = z_{l+2}(t+1) \\ z_{l+2}(t) &= u_e(t-1) = z_{l+3}(t+1) \\ z_{l+3}(t) &= u_e(t-2) = z_{l+4}(t+1) \\ &\vdots \\ z_{l+l}(t) &= u_e(t-l+1) = z_{l+l+1}(t+1) \\ z_{l+l+1}(t) &= u_e(t-l) \end{aligned} \tag{8.28}$$

Then, the system (8.25) can be represented:

$$\tilde{z}_e(t+1) = \tilde{A}^e \tilde{z}_e(t) + \tilde{B}^e u_e(t) \tag{8.29}$$

For $l > 0$, the matrices $\tilde{A}^e$ and $\tilde{B}^e$ in (8.29) are given by:

$$\tilde{A}^e = \begin{bmatrix} A_1^e & A_2^e & \dots & A_l^e & A_{l+1}^e & B_2^e & \dots & B_l^e & B_{l+1}^e \\ I_{2p} & 0_{2p} & \dots & 0_{2p} & 0_{2p} & 0_{2p,m} & \dots & 0_{2p,m} & 0_{2p,m} \\ 0_{2p} & I_{2p} & \dots & 0_{2p} & 0_{2p} & 0_{2p,m} & \dots & 0_{2p,m} & 0_{2p,m} \\ \vdots & & & & & & & & \\ 0_{2p} & 0_{2p} & \dots & I_{2p} & 0_{2p} & 0_{2p,m} & \dots & 0_{2p,m} & 0_{2p,m} \\ 0_{m,2p} & 0_{m,2p} & \dots & 0_{m,2p} & 0_{m,2p} & 0_m & \dots & 0_m & 0_m \\ 0_{m,2p} & 0_{m,2p} & \dots & 0_{m,2p} & 0_{m,2p} & I_m & \dots & 0_m & 0_m \\ \vdots & & & & & & & & \\ 0_{m,2p} & 0_{m,2p} & \dots & 0_{m,2p} & 0_{m,2p} & 0_m & \dots & I_m & 0_m \end{bmatrix} \tag{8.30}$$

$$\tilde{B}^e = [B_1^e \; 0_{2p,m} \; 0_{2p,m} \dots 0_{2p,m} \; I_m \; 0_m \dots 0_m]^T \tag{8.31}$$

In (8.30), (8.31), 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 (8.29) solves the following optimization problem:

$$\min_{\{u_e(t), u_e(t+1), \dots\}} \sum_{k=0}^{\infty} \left[\|\tilde{z}_e(t+k)\|_{Q_e}^2 + \|u_e(t+k)\|_{R_e}^2 \right] \tag{8.32}$$

where $Q_e, R_e \succ 0$. The solution to (8.32) is the linear feedback control law:

$$u_e(t+k) = -K\tilde{z}_e(t+k)\,,\; k \geq 0\,, \tag{8.33}$$

where the controller gain K is given by [17]:

$$K = \left(\tilde{B}^{eT} P \tilde{B}^e + R_e\right)^{-1} \tilde{B}^{eT} P \tilde{A}^e \tag{8.34}$$

$$P = \tilde{A}^{eT} P \tilde{A}^e + Q_e - \tilde{A}^{eT} P \tilde{B}^e \left(\tilde{B}^{eT} P \tilde{B}^e + R_e\right)^{-1} \left(\tilde{A}^{eT} P \tilde{B}^e\right)^T \tag{8.35}$$

By taking into account that $u_e(t) \equiv u(t) - u^*_{st}$, it follows from (8.33) that the control input applied to the system is [5, 6]:

$$u(t+k) = -K\tilde{z}_e(t+k) + u^*_{st}\,,\; k \geq 0 \tag{8.36}$$

8.4.2 Explicit Dual-Mode Controller

Consider the closed-loop system:

$$\tilde{z}_e(t+k) = (\tilde{A}^e - \tilde{B}^e K)\tilde{z}_e(t+k-1),\; k \geq 0\,, \tag{8.37}$$

where $\tilde{z}_e(t+k)$ is defined by (8.26) if t is replaced by $t+k$. Assume that $A_{cl} = \tilde{A}^e - \tilde{B}^e K$ is strictly Hurwitz. Let $\Gamma_e = \{\tilde{z}_e \in \mathbb{R}^{q_e} \,|\, \tilde{z}_e^T S \tilde{z}_e \leq \sigma\}$ with $S \succ 0$, $\sigma > 0$, be a positively invariant admissible set for the system (8.37). It means that $\forall \tilde{z}_e(t) \in \Gamma_e$, $\tilde{z}_e(t+k) \in \Gamma_e$, $\forall k > 0$ and:

$$y_{\min} < [\Psi\, 0_{2p} ... 0_{2p}\, 0_m ... 0_m]\tilde{z}_e(t+k) < y_{\max},\; k \geq 0 \tag{8.38}$$

$$u_{\min} < -K\tilde{z}_e(t+k) + u^*_{st} < u_{\max},\; k \geq 0 \tag{8.39}$$

where $\Psi = [I_p\; 0_p]$ and I_p, 0_p, 0_{2p}, 0_m are defined above. Γ_e can be determined in a way similar to Lemma 1 in [1]. If S satisfies the Lyapunov equation:

$$A_{cl}^T S A_{cl} - S = -\mu S - Q_e - K^T R_e K \tag{8.40}$$

for some $\mu > 0$, then there exists a constant $\sigma > 0$ such that the set Γ_e is a positively invariant admissible set for the system (8.37). For $l \leq L$, let $\widetilde{\Gamma}_e^1 = \{\xi \in \mathbb{R}^{\tilde{q}_e} \,|\, \xi^T \widetilde{S}_1 \xi \leq \tilde{\sigma}_1\}$ with $\widetilde{S}_1 \succ 0$, $\tilde{\sigma}_1 > 0$ be the orthogonal projection of Γ_e onto $\mathbb{R}^{\tilde{q}_e}$, $\tilde{q}_e = (l+1)p + lm$, by omitting all integrator elements from the *regressor* vector $\tilde{z}_e$. Let $\widetilde{\Omega}^1 = \{\zeta \in \mathbb{R}^{\tilde{q}_e} \,|\, \zeta^T \widetilde{S}_1 \zeta \leq \|\widetilde{S}_1\| \tilde{\delta}_1\}$ be the orthogonal projection of the terminal set Ω onto $\mathbb{R}^{\tilde{q}_e}$, where $\|\widetilde{S}_1\|$ is the induced norm of matrix $\widetilde{S}_1$. Then, it is required $\|\widetilde{S}_1\| \tilde{\delta}_1 < \tilde{\sigma}_1$, so that $\widetilde{\Omega}^1 \subset \widetilde{\Gamma}_e^1$. For $l > L$, let $\widetilde{\Gamma}_e^2 = \{\xi \in \mathbb{R}^q \,|\, \xi^T \widetilde{S}_2 \xi \leq \tilde{\sigma}_2\}$ with $\widetilde{S}_2 \succ 0$, $\tilde{\sigma}_2 > 0$ be the orthogonal projection of Γ_e onto $\mathbb{R}^q$, $q = (L+1)p + Lm$, by omitting all integrator elements and the elements $y(t-L-1), ..., y(t-l)$ from $\tilde{z}_e$. Let $\widetilde{\Omega}^2 = \{\zeta \in \mathbb{R}^q \,|\, \zeta^T \widetilde{S}_2 \zeta \leq \|\widetilde{S}_2\| \tilde{\delta}_2\}$, with $\tilde{\delta}_2 > 0$ be a set such that $\Omega \subseteq \widetilde{\Omega}^2$. Similar to above it is required $\|\widetilde{S}_2\| \tilde{\delta}_2 < \tilde{\sigma}_2$, so that $\widetilde{\Omega}^2 \subset \widetilde{\Gamma}_e^2$.

In order to define the dual-mode controller, the *regressor* vector, associated to the system (8.37), is introduced:

$$\tilde{z}_r(t) = \begin{cases} [\Psi y_e(t), \Psi y_e(t-1), ..., \Psi y_e(t-l), \\ u_e(t-1)+u^*_{st}, ..., u_e(t-l)+u^*_{st}]\,, \text{ if } l>0 \\ \Psi y_e(t)\,, \text{ if } l=0 \end{cases} \tag{8.41}$$

where Ψ is defined above. Thus, $\tilde{z}_r(t) \in \mathbb{R}^{q_r}$ with $q_r = (l+1)p + lm$. Let $\Gamma_r \in \mathbb{R}^{q_r}$ be the orthogonal projection of Γ_e onto $\mathbb{R}^{q_r}$, specified by (8.41) (note that $q_r < q_e$). Further, for $l = L$, it is required $\Gamma_r \subset Z_\Pi \subset \mathbb{R}^q$. For $l < L$, $\Gamma_r \subset \widetilde{Z}_\Pi \subset \mathbb{R}^{q_r}$, where $\widetilde{Z}_\Pi$ is the orthogonal projection of Z_Π onto $\mathbb{R}^{q_r}$, obtained by omitting the regressors with lag larger than l. For $l > L$, $\widetilde{\Gamma}_r \subset Z_\Pi \subset \mathbb{R}^q$, where $\widetilde{\Gamma}_r$ is the orthogonal projection of Γ_r onto $\mathbb{R}^q$, obtained by omitting the regressors with lag larger than L.

Let $\tilde{z}$, $\tilde{z}_e$, and $\tilde{z}_r$ be the values of the *regressor* vectors (8.7), (8.26), and (8.41) at the current time t. Then, the explicit dual-mode controller is defined as follows:

$$u_d \triangleq \begin{cases} \widehat{u}_0(\tilde{z})\,, \text{ if } \tilde{z}_r \notin \Gamma_r \\ -K\tilde{z}_e + u^*_{st}\,, \text{ if } \tilde{z}_r \in \Gamma_r \end{cases} \tag{8.42}$$

The expression in the first row of (8.42) means that the control is performed by the explicit NN-NMPC controller when the system is far from equilibrium. The expression in the second row implies that the control will be switched to the LQR when $\tilde{z}_r$ enters the set Γ_r and the LQR will continue controlling the system until $\tilde{z}_r$ leaves this set due to a large disturbance, for example. The integrator output y_{int} is used only when $\tilde{z}_r \in \Gamma_r$. In the case when $\tilde{z}_r \notin \Gamma_r$, y_{int} is set to zero and not used.

If the NN ARX model describes exactly the system dynamics far from the origin (outside the set Γ_r) and the problem (8.18) is convex, then the closed-loop system stability can be ensured by conditions similar to those in [10]. In presence of model errors far from the origin, it would be necessary to apply approaches to explicit *robust* NMPC ([4]). If the problem (8.18) is non-convex, then the closed-loop stability can not be guaranteed, but it can be verified by off-line simulations.

8.5 Application: Regulation of a pH Maintaining System

In [5, 6], 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 modeled 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.

8.5.1 The pH Maintaining System

A simplified schematic diagram of the pH maintaining system taken from [9] is given in Fig. 8.1. 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.

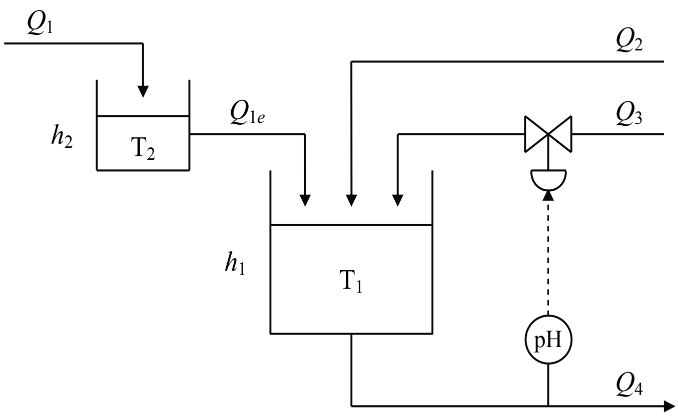


Fig. 8.1 Scheme of the pH maintaining system.

In [9], a dynamic model of the pH maintaining system is derived using conservation equations and equilibrium relations. 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 according to [9].

The chemical reactions for the system are:

$$H_2CO_3 \longleftrightarrow HCO_3^- + H^+ \tag{8.43}$$

$$HCO_3^- \longleftrightarrow CO_3^= + H^+ \tag{8.44}$$

$$H_2O \longleftrightarrow OH^- + H^+ \tag{8.45}$$

The corresponding equilibrium constants are:

$$K_{a1} = \frac{[HCO_3^-][H^+]}{[H_2CO_3]}\ ,\ K_{a2} = \frac{[CO_3^=][H^+]}{[HCO_3^-]}\ ,\ K_w = [H^+][OH^-] \tag{8.46}$$

The chemical equilibria is modeled by defining two reaction invariants for each of the streams Q_i, $i \in \{1,2,3,4\}$ [9]:

$$W_{ai} = [H^+]_i - [OH^-]_i - [HCO_3^-]_i - 2[CO_3^=]_i \tag{8.47}$$

$$W_{bi} = [H_2CO_3]_i + [HCO_3^-]_i + [CO_3^=]_i \tag{8.48}$$

The invariant W_a is a charge related quantity, while W_b represents the concentration of the $CO_3^=$ ion. The pH can be determined from W_a and W_b using the following relations [9]:

$$W_b \frac{\frac{K_{a1}}{[H^+]} + \frac{2K_{a1}K_{a2}}{[H^+]^2}}{1 + \frac{K_{a1}}{[H^+]} + \frac{K_{a1}K_{a2}}{[H^+]^2}} + W_a + \frac{K_w}{[H^+]} - [H^+] = 0 \tag{8.49}$$

$$\mathrm{pH} = -\log([H^+]) \tag{8.50}$$

In [9], 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:

$$A_1 \frac{dh_1}{dt} = Q_{1e} + Q_2 + Q_3 - Q_4 \,, \tag{8.51}$$

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 modeled as:

$$Q_4 = C_v (h_1 + l)^s \,, \tag{8.52}$$

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 [9]:

$$A_1 h_1 \frac{dW_{a4}}{dt} = Q_{1e}(W_{a1} - W_{a4}) + Q_2(W_{a2} - W_{a4}) + Q_3(W_{a3} - W_{a4}) \tag{8.53}$$

$$A_1 h_1 \frac{dW_{b4}}{dt} = Q_{1e}(W_{b1} - W_{b4}) + Q_2(W_{b2} - W_{b4}) + Q_3(W_{b3} - W_{b4}) \tag{8.54}$$

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:

$$x = [W_{a4}\; W_{b4}\; h_1]^T \,,\; \tilde{u} = Q_3 \,,\; \tilde{y} = \mathrm{pH} \tag{8.55}$$

The state space model has the form [9]:

$$\dot{x} = \tilde{f}(x) + \tilde{g}(x)\tilde{u} \tag{8.56}$$

$$c(x, \tilde{y}) = 0 \,, \tag{8.57}$$

where:

$$\tilde{f}(x) = \begin{bmatrix} \frac{Q_1(W_{a1}-x_1)+Q_2(W_{a2}-x_1)}{A_1 x_3} \\ \frac{Q_1(W_{b1}-x_2)+Q_2(W_{b2}-x_2)}{A_1 x_3} \\ \frac{Q_1 - C_v(x_3+l)^s + Q_2}{A_1} \end{bmatrix} \,,\; \tilde{g}(x) = \begin{bmatrix} \frac{W_{a3}-x_1}{A_1 x_3} \\ \frac{W_{b3}-x_2}{A_1 x_3} \\ \frac{1}{A_1} \end{bmatrix} \tag{8.58}$$

$$c(x,\tilde{y}) = x_1 + 10^{\tilde{y}-14} - 10^{-\tilde{y}} + \frac{x_2(1+2\times 10^{\tilde{y}-pK_2})}{1+10^{pK_1-\tilde{y}}+10^{\tilde{y}-pK_2}} \tag{8.59}$$

The relation between the constants K_{a1}, K_{a2} in (8.49) and the constants K_1, K_2 in (8.59) is:

$$K_{a1} = 10^{-pK_1}\ ,\ K_{a2} = 10^{-pK_2}\ ,\ p > 0\ . \tag{8.60}$$

The parameters of the model (8.56)–(8.60) are given in [9].

8.5.2 ARX Model Identification

8.5.2.1 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 (8.56)–(8.57), where the liquid level h_1 in tank T_1 is assumed to be constant [5, 6]. 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 (8.56)–(8.57) 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 (8.7)–(8.8). 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 modeling approaches. The number of regressors (delayed inputs and outputs) was determined by the method described in [8]. A trade-off between modeling error and complexity was taken into the account. The final selection was that the system model has the form:

$$y(t+1) = f_{NN}(\tilde{z}(t), u(t), \theta) \tag{8.61}$$

$$\tilde{z}(t) = [y(t), y(t-1), y(t-2), u(t-1), u(t-2)] \tag{8.62}$$

It should be noted that in difference to the state space model (8.56)–(8.57) where $\tilde{y} = \text{pH}$, in the NN model (8.61)–(8.62) 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 [9] 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 (8.61)–(8.62) 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. 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 (8.3), 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 (8.61)–(8.62) were obtained, with thirteen neurons in the hidden layer. More about systematic network structure selection, pruning and other issues regarding neural networks modeling can be found in various literature describing this topic and its applications (e.g. [16], [2], [8], [7], [19], [21]).

Fig. 8.2 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.

8.5.2.2 Linear ARX Model Identification

The equilibrium point of the pH maintaining system (8.56)–(8.57) is $\tilde{y} = 4.8$, $\tilde{u}^*_{st} = 10.94\,[\mathrm{ml/s}]$ (respectively $y = 0$, $u^*_{st} = 0.1732$ after scaling). A validation of the obtained NN ARX model near this point clearly shows that it is not accurate (see Fig. 8.3).

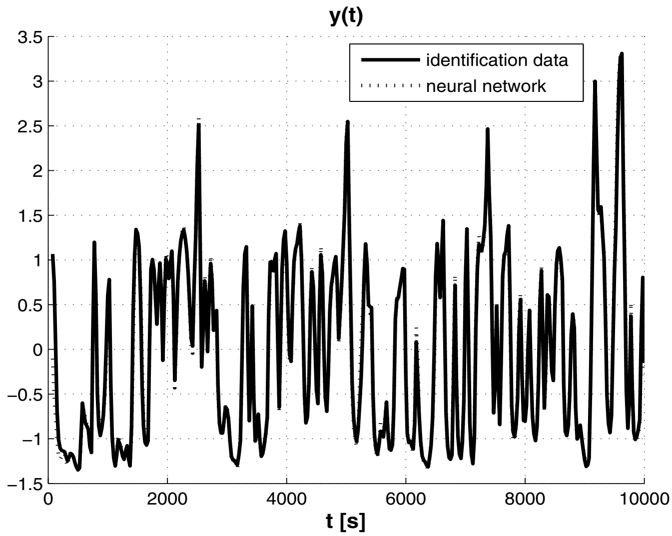
In order to obtain accurate predictions when the output variable is close to zero, the following 1-st order linear ARX model is identified [5, 6]:

$$y(t+1) = 0.7704y(t) + 0.0539(u(t) - u^*_{st}) \qquad (8.63)$$

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 (8.63). The simulated response of the ARX model (8.63) is depicted in Fig. 8.3.

8.5.3 Design of Explicit Dual-Mode Controller

The approach described in Sections 8.3 and 8.4 is applied to design an explicit dual-mode controller for the pH maintaining system based on its NN model (8.61)–(8.62) and linear ARX model (8.63) [5, 6]. Recall that due to scaling, the variables u and y can take both positive and negative values.



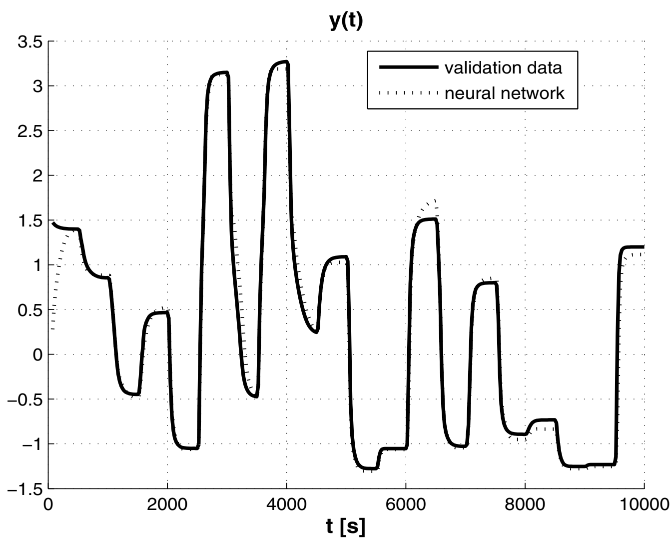


Fig. 8.2 Response of the NN model to the excitation signal used for identification (top) and to the excitation signal used for validation (bottom).

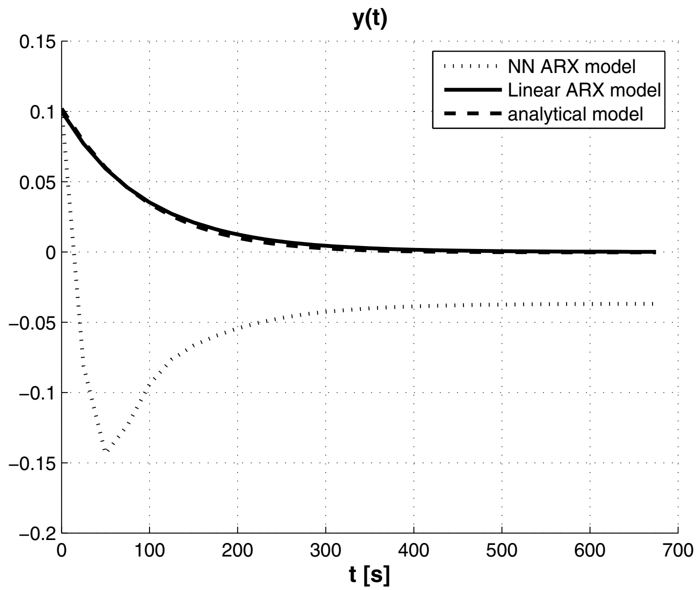


Fig. 8.3 Validation of the NN ARX and the linear ARX models. The dotted curve is with the NN model (8.61)–(8.62), the solid curve is with the linear ARX model (8.63), and the dashed curve is with the first-principles model (8.56)–(8.57). Constant control input $u = u^*_{st}$ is used as an excitation signal.

First, the approach in Section 8.3 is applied to design an explicit approximate NN-NMPC controller. The following control input constraint is imposed on the system:

$$-0.4 \leq u \leq 0.4 \tag{8.64}$$

The horizon is $N = 8$ and the terminal constraint in Problem 8.1 is:

$$\tilde{z}^c_{t+N|t} \in \Omega \ , \tag{8.65}$$

where $\Omega = \{\tilde{z}^c \in \mathbb{R}^5 \,|\, \|\tilde{z}^c\|^2 \leq 0.05\}$. The weighting matrices in the cost function (8.17) are $Q = 10$, $R = 1$, $F = 10$. The NN-NMPC minimizes the cost function (8.17) subject to the model (8.61)–(8.62) and the constraints (8.64)–(8.65). In (8.20), 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 in a binary search tree representation. Totally, 33 arithmetic operations are needed in real-time to compute the control input by traversing the binary search tree (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 (8.63):

$$y(t+1) = 0.7704y(t) + 0.0539u_e(t) \tag{8.66}$$

$$y_{int}(t+1) = y_{int}(t) + T_s y(t) \tag{8.67}$$

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

$$\tilde{z}_e(t+1) = \tilde{A}^e \tilde{z}_e(t) + \tilde{B}^e u_e(t) \ , \tag{8.68}$$

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 (8.68) is:

$$u_e = -K\tilde{z}_e = -k_1 y - k_2 y_{int} \ , \ \text{where } K = [0.7994, 0.0069] \tag{8.69}$$

This control law solves the optimization problem (8.32) with weighting matrices $Q_e = \mathrm{diag}\{10, 0.0005\}, R_e = 10$.

Then, the explicit dual-mode controller for the pH maintaining system is defined according to (8.42) with $\Gamma_r = \{\tilde{z}_r \in \mathbb{R} \,|\, \tilde{z}_r^2 \le 0.09\}$, where $\tilde{z}_r(t) = y(t)$.

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 (8.56)–(8.57). Further, it is assumed that there are persistent disturbances in the acid and the buffer flow rates, which have the following values $\tilde{Q}_1 = 16.8\,[\mathrm{ml/s}]$, $\tilde{Q}_2 = 0.53\,[\mathrm{ml/s}]$ (different from the nominal values $Q_1^* = 16.6\,[\mathrm{ml/s}]$, $Q_2^* = 0.55\,[\mathrm{ml/s}]$). 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 Fig. 8.4, while the trajectories of the exit flow rate Q_4 and the liquid level h_1 are depicted in Fig. 8.5.

It can be seen from Fig. 8.4 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 Figs. 8.4 and 8.5 are obtained. The exact NMPC response is computed by solving at each time instant an open-loop NMPC problem formulated for the first-principles model (8.56)–(8.57). In contrast, the approximate explicit NMPC solution is first computed off-line as an approximation to Problem 8.1, in which the NN ARX model by itself represents another approximation. Then, its performance is simulated in closed-loop with the first-principles model (8.56)–(8.57). Thus, the performance degradation far from the origin is due to the approximations in the model and in the NMPC

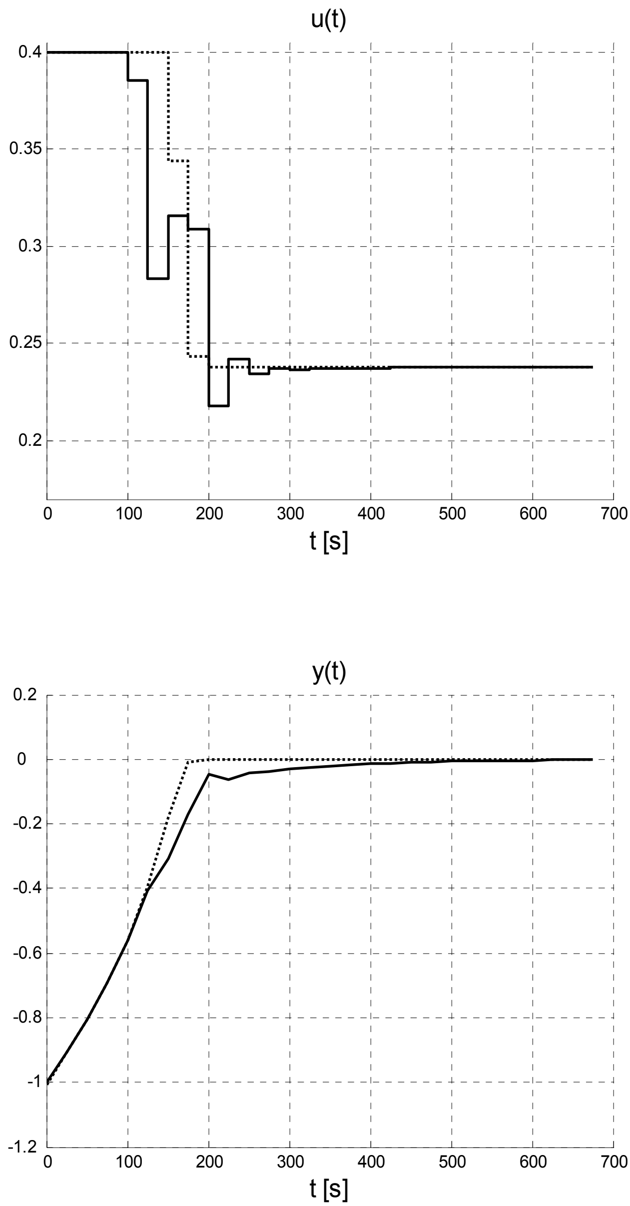
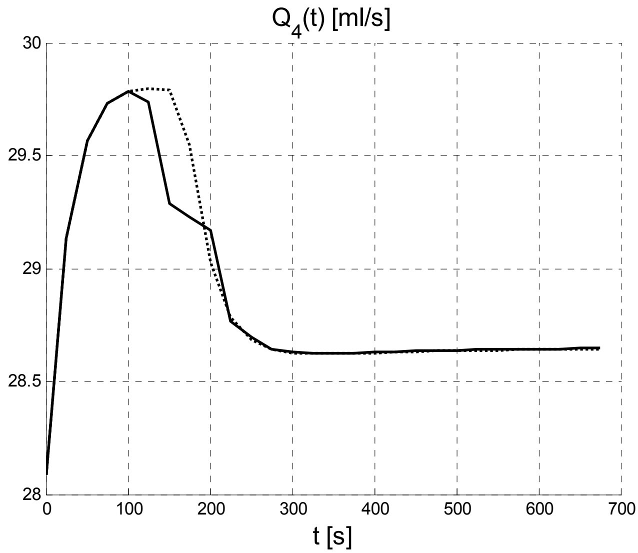


Fig. 8.4 Control input u (top) and output variable y (bottom) obtained with the explicit dual-mode controller in closed-loop with the first-principles model (8.56)–(8.57). The solid curves are with the approximate explicit NN-NMPC and the dotted curves are with the exact NN-NMPC.



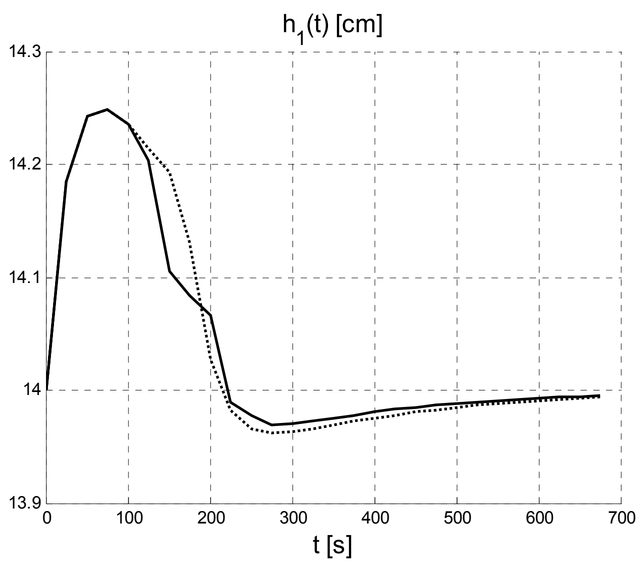


Fig. 8.5 The exit flow rate Q_4 (top) and liquid level h_1 (bottom). The solid curves are with the approximate explicit NN-NMPC and the dotted curves are with the exact NN-NMPC.

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 Figs. 8.4 and 8.5 has a typical amount of performance degradation being representative for other initial conditions and scenarios.

References

1. Chen, H., Allgöwer, F.: A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability. Automatica 34, 1205–1217 (1998)
2. Chen, S., Billings, S.A., Grant, P.M.: Non-linear system identification using neural networks. International Journal of Control 51, 1191–1214 (1990)
3. Fiacco, A.V.: Introduction to sensitivity and stability analysis in nonlinear programming. Academic Press, Orlando (1983)
4. Grancharova, A., Johansen, T.A.: Computation, approximation and stability of explicit feedback min-max nonlinear model predictive control. Automatica 45, 1134–1143 (2009)
5. Grancharova, A., Kocijan, J., Johansen, T.A.: Dual-mode explicit output-feedback predictive control based on neural network models. In: Proceedings of the 8th IFAC Symposium on Nonlinear Control Systems (NOLCOS 2010), Bologna, Italy, pp. 545–550 (2010), `http://www.IFAC-PapersOnLine.net`
6. Grancharova, A., Kocijan, J., Johansen, T.A.: Explicit output-feedback nonlinear predictive control based on black-box models. Engineering Applications of Artificial Intelligence 24, 388–397 (2011)
7. Han, J., Moraga, C., Sinne, S.: Optimization of feedforward neural networks. Engineering Applications of Artificial Intelligence 9, 109–119 (1996)
8. He, X., Asada, H.: A new method for identifying orders of input-output models for nonlinear dynamic systems. In: Proceedings of the American Control Conference, San Francisco, CA, pp. 2520–2523 (1993)
9. Henson, M.A., Seborg, D.E.: Adaptive nonlinear control of a pH neutralization process. IEEE Transactions on Control System Technology 2, 169–183 (1994)
10. Johansen, T.A.: Approximate explicit receding horizon control of constrained nonlinear systems. Automatica 40, 293–300 (2004)
11. Kocijan, J., Murray-Smith, R.: Nonlinear Predictive Control with a Gaussian Process Model. In: Murray-Smith, R., Shorten, R. (eds.) Switching and Learning in Feedback Systems. LNCS, vol. 3355, pp. 185–200. Springer, Heidelberg (2005)
12. Lepetič, M., Škrjanc, I., Chiacchiarini, H.G., Matko, D.: Predictive functional control based on fuzzy model: Magnetic suspension system case study. Engineering Applications of Artificial Intelligence 16, 425–430 (2003)
13. Ljung, L., Glad, T.: Modeling of Dynamic Systems. Prentice Hall, Englewood Cliffs (1994)
14. Mayne, D.Q., Rawlings, J.B., Rao, C.V., Scokaert, P.O.M.: Constrained model predictive control: Stability and optimality. Automatica 36, 789–814 (2000)
15. Michalska, H., Mayne, D.Q.: Robust receding horizon control of constrained nonlinear systems. IEEE Transactions on Automatic Control 38, 1623–1633 (1993)
16. Nørgaard, M., Ravn, O., Poulsen, N.K., Hansen, L.K.: Neural Networks for Modelling and Control of Dynamic Systems. Springer, London (2000)
17. Ogata, K.: Discrete-time Control Systems. Prentice Hall, Englewood Cliffs (1995)

18. Peng, H., Yang, Z.J., Gui, W., Wu, M., Shioya, H., Nakano, K.: Nonlinear system modelling and robust predictive control based on RBF-ARX model. Engineering Applications of Artificial Intelligence 20, 1–9 (2007)
19. Principe, J.C., Euliano, N.R., Lefebvre, W.C.: Neural and Adaptive systems. John Wiley & Sons, New York (2000)
20. Sjöberg, J., Zhang, Q., Ljung, L., Benveniste, A., Delyon, B., Glorennec, P.Y., Hjalmarsson, H., Juditsky, A.: Non-linear black-box modeling in system identification: A unified overview. Automatica 31, 1691–1724 (1995)
21. Venkatasubramanian, V., McAvoy, T.J. (eds.): Neural network applications in chemical engineering. Special Issue of Computers & Chemical Engineering 16(4) (1992)
22. Zeng, G.M., Qin, X.S., He, L., Huang, G.H., Liu, H.L., Lin, Y.P.: A neural network predictive control system for paper mill wastewater treatment. Engineering Applications of Artificial Intelligence 16, 121–129 (2003)