INTERNATIONAL JOURNAL OF ROBUST AND NONLINEAR CONTROL
Int. J. Robust Nonlinear Control 2015; **25**:3546–3571
Published online 20 November 2014 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/rnc.3279

A convex approach for NMPC based on second order Volterra series models

J. K. Gruber[1,*,†], D. R. Ramirez[2], D. Limon[2] and T. Alamo[2]

[1]*Electrical Systems Unit, IMDEA Energy Institute, Madrid, Spain*
[2]*Department of Automatic Control and Systems Engineering, Escuela Superior de Ingenieros, University of Seville, Seville, Spain*

SUMMARY

In model predictive control (MPC), the input sequence is computed, minimizing a usually quadratic cost function based on the predicted evolution of the system output. In the case of nonlinear MPC (NMPC), the use of nonlinear prediction models frequently leads to non-convex optimization problems with several minimums. This paper proposes a new NMPC strategy based on second order Volterra series models where the original performance index is approximated by quadratic functions, which represent a lower bound of the original performance index. Convexity of the approximating quadratic cost functions can be achieved easily by a suitable choice of the weighting of the control increments in the performance index. The approximating cost functions can be globally minimized by convex optimization techniques in order to compute the input sequence. The minimization of the performance index is carried out by an iterative optimization procedure, which guarantees convergence to the solution. Furthermore, for a nominal prediction model, asymptotic stability for the proposed NMPC strategy can be shown. In the case of considering an estimation error in the prediction model, input-to-state practical stability is assured. The control performance of the NMPC strategy is illustrated by experimental results. Copyright © 2014 John Wiley & Sons, Ltd.

Received 29 September 2013; Revised 14 September 2014; Accepted 30 October 2014

KEY WORDS: nonlinear model predictive control; convex optimization; convergence; stability

1. INTRODUCTION

Model predictive control (MPC) represents nowadays one of the most common advanced control techniques applied to industrial processes [1–4]. In MPC, the input sequence is computed minimizing a usually quadratic cost function based on the predicted evolution of the system. The mentioned prediction of the future evolution of the system is carried out by means of a mathematical model of the system. Consequently, the quality of the mathematical model used to approximate the system dynamics has a decisive influence on the control performance. In nonlinear MPC (NMPC), the combination of a nonlinear prediction model and a quadratic cost function frequently leads to a non-convex optimization problem, resulting in different computational problems. A non-convex character of the cost function to be minimized gives rise to local minimums, making the minimization of the cost function a more difficult task. A modification in the cost function [5] in order to obtain a convex optimization problem can reduce the computational complexity.

Virtually all dynamic processes of practical importance exhibit some degree of nonlinear behavior [6, 7]. However, the use of NMPC in industry applications is very limited due to the higher computational complexity of the oftentimes non-convex optimization problems. While linear MPC requires in every sampling period the solution of a convex problem, usually carried out with quadratic

[*]Correspondence to: J. K. Gruber, Electrical Systems Unit, IMDEA Energy Institute, Avda. Ramón de la Sagra 3, 28935 Móstoles, Madrid, Spain.

[†]E-mail: jorn.gruber@imdea.org

programming (QP), NMPC requires (at least a partial) solution with the help of nonlinear programming [8]. The difficulty of the optimization problem results in an important increase in the computation time, limiting the use of NMPC in many cases to slow processes or the consideration of small horizons. Besides, from a more theoretical point of view, the possible non-convexity of the optimization problem considerably complicates the study of stability and robustness. The mentioned problems account for the seldom use of NMPC in industry where only a few NMPC applications have been reported, including the areas of refining, chemicals, polymers as well as air and gas processing [9, 10].

Nevertheless, when the process is described by a Volterra series model, computationally efficient solutions for the model predictive control problem can be found. As they represent a natural extension of linear convolution models, Volterra series models can be obtained from input–output data without detailed knowledge of the process. Being linear in the parameters, widely used identification methods, for example, least squares method, can be used to determine the model parameters. A computationally efficient unconstrained NMPC strategy based on second order Volterra series models has been proposed in [11, 12]. The unconstrained optimization is carried out by means of an iterative algorithm based on the separation of the linear and the nonlinear terms of a Volterra series model. The NMPC strategy consists of a conventional linear MPC extended by an auxiliary loop containing the nonlinear dynamics in the optimization process. The proposed NMPC for second order Volterra series models was generalized in [13] for its use in combination with higher order Volterra series models. Analogously to the original optimization approach, the optimization problem is solved by an iterative optimization algorithm based on the separation of the linear and nonlinear terms where the auxiliary loop contains also third and higher-order terms. In [14], two suboptimal NMPC strategies for second order Volterra series models have been presented. These strategies are based on the assumption of constant increments in the input signal over the entire control horizon, that is, the resulting input sequence has a constant slope, or a step-like control where the input signal is assumed to be constant along the control horizon. In both NMPC strategies, depending only on one sole decision variable (constant slope or constant input signal), the resulting one-dimensional optimization problem, which can still be non-convex, can be easily minimized with respect to the decision variable. Recently, [15] presented a robust NMPC based on a Volterra series model with a low computational complexity. The use of terminal constraints guarantees convergence to a neighborhood around the setpoint in presence of disturbances.

With respect to the stability of NMPC strategies, it is important to have in mind that Volterra series models represent a generalization of finite impulse response (FIR) models and can be considered as stable fading memory systems. For MPC strategies based on linear FIR models, several authors have proven closed-loop stability, among others [16–18]. The use of terminal constraints to ensure stability of nonlinear controllers has been proposed [19–21]. Other publications, especially in the area of economic NMPC [22–24], point out that general stability can be proven under certain conditions using Lyapunov functions. In spite of the mentioned propositions, stability of NMPC strategies based on Volterra series models is an open field with few results.

In this work, a new NMPC strategy based on second order Volterra series models is presented. The proposed strategy uses approximating quadratic functions that represent lower bounds of the original cost function. For these approximating quadratic functions, convexity can be achieved easily by adding a weighting of the input sequence increments. After determining a first approximating function, global minimization of this function is carried out in order to compute a first candidate input sequence. With the first candidate input sequence, a new approximating function can be computed. An optimization algorithm computes a new candidate input sequence by minimizing the pointwise maximum of all approximating functions. This procedure is repeated until a certain accuracy criterion is satisfied. The convergence of the proposed procedure can be shown because the difference between the approximated and the original cost is strict monotonically decreasing. Furthermore, with the pointwise maximum of the approximated convex cost functions being a convex function, the minimization of the pointwise maximum leads to a global minimization of the original cost function. Besides, the proposed NMPC strategy guarantees asymptotic stability in the case of a nominal prediction model. Also, under consideration of an estimation error as a consequence of a model mismatch, input-to-state practical stability is proven.

DOI: 10.1002/rnc

The paper is organized as follows: Section 2 presents the problem description and the notation used in this work. Section 3 presents the general concept of using an approximating cost function to guarantee convexity and its use in combination with second order Volterra series models. The resulting control law, a procedure for global minimization that exploits the convex character of the approximating functions, is presented in Section 4. Stability of the proposed NMPC strategy is proven in Section 5, and experimental results of the control strategy applied to a nonlinear process are presented in Section 6. Finally, in Section 7, the major conclusions are drawn.

2. PROBLEM DESCRIPTION

Considering a fading memory system [25], a non-autoregressive second order Volterra series model is defined as [26]

$$y(k) = y_0 + \sum_{i=1}^{N_t} a_i u(k-i) + \sum_{i=1}^{N_t} \sum_{j=i}^{N_t} b_{i,j} u(k-i) u(k-j) \tag{1}$$

where $y(k)$ denotes the model output and $u(k)$ is the model input subject to the constraints $u(k) \in U \triangleq [u_{min}, u_{max}]$. The model parameters y_0, a_i, and $b_{i,j}$ represent the offset, the linear term parameters, and nonlinear term parameters[‡], respectively. The truncation orders for the linear and nonlinear part of the model can be different, but for the sake of simplicity and without loss of generality, N_t will be used as a common truncation order for both parts[§]. It can be seen easily from the model (1) that the output $y(k)$ in the sample k depends only on the past input values $u(k-i)$ along the truncation orders (with $1 \leqslant i \leqslant N_t$). For the prediction of the future output values, a known estimation error representing a measurable disturbance will be considered. Hence, the prediction model based on a second order Volterra series model can be written as

$$y(k+j|k) = y_0 + \sum_{i=1}^{N_t} a_i u(k+j-i|k) + \sum_{i=1}^{N_t} \sum_{l=i}^{N_t} b_{i,l} u(k+j-i|k) u(k+j-l) + d(k) \tag{2}$$

where $y(k+j|k)$ represents the output prediction for $k+j$ made at the sampling period k and $d(k)$ is the measurable disturbance in k.

The second order Volterra series model with the measurable disturbance (2) can be used as prediction model in a general quadratic cost function [2]:

$$J(\boldsymbol{u}) = \sum_{j=1}^{N} (y(k+j|k) - r(k))^2 + \lambda \sum_{j=0}^{N_u-1} (u(k+j|k) - u_r(k))^2 \tag{3}$$

where the future input sequence $\boldsymbol{u} \in \mathbb{R}^{N_u}$ computed at k is given by $\boldsymbol{u} = [u(k|k), \ldots, u(k+N_u-1|k)]^T$. The upper limits N and N_u denote the considered prediction and control horizons, respectively. For stability reasons, it is assumed that the chosen horizons and the truncation order satisfy $N \geqslant N_t + N_u$. Besides, the parameter λ is used as a weighting factor for the control effort. The variable $r(k)$ represents the setpoint for the output, and $u_r(k)$ is the corresponding steady-state input defined by

$$u_r(k) = \varphi(r(k) - d(k)) \tag{4}$$

where the function $\varphi : \mathbb{R} \mapsto \mathbb{R}$ returns the nominal steady-state input for a given nominal steady-state output, that is, $u_{ss} = \varphi(y_{ss})$ for an equilibrium of the nominal model given by (u_{ss}, y_{ss}). It

[‡] The general notation for the linear and nonlinear term parameters is $h_1(i)$ and $h_2(i,j)$ [12]. To shorten the syntax, $a_i = h_1(i)$ and $b_{i,j} = h_2(i,j)$ have been used.

[§] If different truncation orders are used, for example, N_1 for the linear and N_2 for the nonlinear part, the common truncation order should be chosen as $N_t = \max(N_1, N_2)$. If $N_1 > N_2$, that is, $N_t = N_1$, the missing second order term parameters are defined as $b_{i,j} = 0 \ \forall\ i > N_2 \ \forall\ j > N_2$. In the opposite case, that is, $N_2 > N_1$ and therefore $N_t = N_2$, the linear term parameters are defined as $a_i = 0 \ \forall\ i > N_1$.

DOI: 10.1002/rnc

is assumed that the setpoint $r(k)$ is chosen so that the resulting steady-state input satisfies $u_r(k) \in U$ for any possible value of $d(k)$. Note that the disturbance $d(k)$ can be easily computed as the difference between the process output and the output of the nominal model (1).

As usual in NMPC, the control action for the system is calculated minimizing the cost function (3):

$$\begin{aligned} \boldsymbol{u}^* = \arg \min_{\boldsymbol{u}} \ & J(\boldsymbol{u}) \\ \text{s.t.} \ & u(k+i|k) \in U, \qquad i = 0, \dots, N_u - 1 \\ & u(k+i|k) = u_r(k), \ i = N_u, \dots, N-1 \end{aligned} \tag{5}$$

considering input constraints along the control horizon. The solution of the optimization problem is applied using a receding horizon strategy, habitual in all predictive control schemes [2].

The combination of a second order model (2) with a quadratic function (3) leads to a fourth order cost function. The main problem of the optimization is the not necessarily convex fourth order function, which gives rise to several minimums. Hence, the aim of this paper is to minimize the cost by means of approximated convex cost functions. Therefore, convexity of the approximated cost functions has to be assured.

3. CONVEXIFICATION

This section shows the concept to assure strict convexity by means of approximation for a quadratic cost function when the model output is defined by a second order function.

3.1. General development

Consider the general quadratic cost function under consideration of the control effort

$$J(\boldsymbol{u}, R) = J_{\boldsymbol{e}}(\boldsymbol{u}) + J_{\boldsymbol{u}}(\boldsymbol{u}, R) \tag{6}$$

with

$$J_{\boldsymbol{e}}(\boldsymbol{u}) = (\boldsymbol{y} - \boldsymbol{r})^T Q (\boldsymbol{y} - \boldsymbol{r}) = \boldsymbol{e}^T Q \boldsymbol{e} \tag{7}$$

$$J_{\boldsymbol{u}}(\boldsymbol{u}, R) = (\boldsymbol{u} - \boldsymbol{u}_r)^T R (\boldsymbol{u} - \boldsymbol{u}_r) \tag{8}$$

where $\boldsymbol{y} = [y(k+1|k), \dots, y(k+N|k)]^T \in \mathbb{R}^N$ represents the vector of the predicted system output based on a quadratic function, $\boldsymbol{r} \in \mathbb{R}^N$ denotes the reference sequence over the prediction horizon, and $\boldsymbol{e} \in \mathbb{R}^N$ is the output error with $\boldsymbol{e} = \boldsymbol{y} - \boldsymbol{r}$. The vector $\boldsymbol{u}_r \in \mathbb{R}^{N_u}$ is given by $\boldsymbol{u}_r = [u_r(k), \dots, u_r(k)]^T$ and contains the steady-state input $u_r(k)$ defined in (4). Q and R are diagonal semi-definite positive matrices weighting the output error and the control effort in the cost function. Note that R has been included in the list of arguments in (6) and (8) as it represents an important parameter in the convexification.

In first place, the function (7) is linearly approximated in the neighbourhood $\boldsymbol{e}^{(0)}$ of the predicted output error in the following form:

$$\tilde{J}_{\boldsymbol{e}}(\boldsymbol{u}, \boldsymbol{e}^{(0)}) = -\boldsymbol{e}^{(0)T} Q \boldsymbol{e}^{(0)} + 2\boldsymbol{e}^{(0)T} Q \boldsymbol{e} \tag{9}$$

with the predicted output error $\boldsymbol{e}$ defined generally as

$$\boldsymbol{e} = \mathbf{p}(\boldsymbol{u}) + \boldsymbol{b} \tag{10}$$

where $\mathbf{p}(\boldsymbol{u})$: $\mathbb{R}^N \to \mathbb{R}^N$ represents a nonlinear second order function in $\boldsymbol{u}$ and $\boldsymbol{b} \in \mathbb{R}^N$ denotes a constant vector[¶]. With (10), the neighborhood of the predicted output error can be written as $\boldsymbol{e}^{(0)} = \mathbf{p}(\boldsymbol{u}^{(0)}) + \boldsymbol{b}$ and the approximated function (9) can be rewritten in the following form:

$$\tilde{J}_{\boldsymbol{e}}(\boldsymbol{u}, \boldsymbol{e}^{(0)}) = 2\boldsymbol{e}^{(0)T} Q \mathbf{p}(\boldsymbol{u}) + 2\boldsymbol{e}^{(0)T} Q \boldsymbol{b} - \boldsymbol{e}^{(0)T} Q \boldsymbol{e}^{(0)} \tag{11}$$

[¶] Note that any constant term of $\boldsymbol{e}$ is considered in $\boldsymbol{b}$, that is, the nonlinear second order function $\mathbf{p}(\boldsymbol{u})$ contains only quadratic and linear terms in $\boldsymbol{u}$.

DOI: 10.1002/rnc

In second place, the approximated function (11) is decomposed in its quadratic and linear terms. It can be seen easily that only the first right-hand term, that is, $\mathbf{p}(\boldsymbol{u})$, depends on the control sequence $\boldsymbol{u}$. Therefore the term $2\boldsymbol{e}^{(0)T} Q\, \mathbf{p}(\boldsymbol{u})$ is decomposed in the following way:

$$2\boldsymbol{e}^{(0)T} Q\, \mathbf{p}(\boldsymbol{u}) = \boldsymbol{u}^T S \boldsymbol{u} + 2\boldsymbol{h}^T \boldsymbol{u} \tag{12}$$

Then using the decomposition (12) in (11), the approximated function becomes

$$\tilde{J}_e(\boldsymbol{u}, \boldsymbol{e}^{(0)}) = \boldsymbol{u}^T S \boldsymbol{u} + 2\boldsymbol{h}^T \boldsymbol{u} + 2\boldsymbol{e}^{(0)T} Q\, \boldsymbol{b} - \boldsymbol{e}^{(0)T} Q \boldsymbol{e}^{(0)} \tag{13}$$

Finally, the approximated cost function, including the approximated function $\tilde{J}_{\boldsymbol{e}}(\boldsymbol{u}, \boldsymbol{e}^{(0)})$ (13) and the control effort $J_{\boldsymbol{u}}(\boldsymbol{u}, R^{(0)})$ (8) based on the initial weighting matrix $R^{(0)}$, can be written as

$$\begin{aligned} \tilde{J}(\boldsymbol{u}, \boldsymbol{e}^{(0)}, R^{(0)}) &= \tilde{J}_e(\boldsymbol{u}, \boldsymbol{e}^{(0)}) + J_{\boldsymbol{u}}(\boldsymbol{u}, R^{(0)}) \\ &= \boldsymbol{u}^T(S + R^{(0)})\boldsymbol{u} + 2(\boldsymbol{h}^T - \boldsymbol{u}_r^T R^{(0)})\boldsymbol{u} + \\ &\quad\; 2\boldsymbol{e}^{(0)T} Q \boldsymbol{b} - \boldsymbol{e}^{(0)T} Q \boldsymbol{e}^{(0)} + \boldsymbol{u}_r^T R^{(0)} \boldsymbol{u}_r \end{aligned} \tag{14}$$

having the necessary form to easily check strict convexity ($S + R^{(0)} > 0$). Then, substituting $R^{(0)}$ by a new weighting matrix $R^{(1)}$ can be used to guarantee strict convexity of the approximated cost function (14). The mentioned weighting matrix is computed with $j = 1$ in the following form:

$$R^{(j)} = \begin{cases} R^{(j-1)} + \chi \boldsymbol{I}_{N_u} & \text{if } \lambda_{min}^{(j-1)} \leqslant 0 \\ R^{(j-1)} & \text{if } \lambda_{min}^{(j-1)} > 0 \end{cases} \tag{15}$$

being $\lambda_{\min}^{(j-1)}$ the smallest eigenvalue of $S + R^{(j-1)}$, $\chi > -\lambda_{\min}^{(j-1)}$ and $\boldsymbol{I}_{N_u} \in \mathbb{R}^{N_u \times N_u}$ denotes an identity matrix.

With $\tilde{J}(\boldsymbol{u}, \boldsymbol{e}^{(0)}, R^{(1)})$ being a strict convex function, it can be shown that

$$\tilde{J}(\boldsymbol{u}, \boldsymbol{e}^{(0)}, R^{(1)}) \leqslant J(\boldsymbol{u}, R^{(1)}) \tag{16}$$

where the strict inequality holds for $\boldsymbol{u} \neq \boldsymbol{u}^{(0)}$ and the equality is valid in the point of approximation $\boldsymbol{u} = \boldsymbol{u}^{(0)}$. Hence, $\tilde{J}(\boldsymbol{u}, \boldsymbol{e}^{(0)}, R^{(1)})$ can be considered a lower bound of the original cost function $J(\boldsymbol{u}, R^{(1)})$ given in (6).

Consider several convex cost functions $\tilde{J}(\boldsymbol{u}, \boldsymbol{e}^{(j-1)}, R^{(j)})$ approximated around $\boldsymbol{e}^{(j-1)}$ for $j = 1, \ldots, n$ where $R^{(j)}$ guarantees convexity for the cost function approximated around $\boldsymbol{e}^{(j-1)}$. With the condition

$$R^{(j)} \geqslant R^{(j-1)} \tag{17}$$

the weighting matrix $R^{(n)}$ assures convexity for all cost functions $\tilde{J}(\boldsymbol{u}, \boldsymbol{e}^{(j-1)}, R^{(n)})$ with $j = 1, \ldots, n$. The maximum of several approximated cost functions based on the weighting matrix $R^{(n)}$ is defined by

$$\tilde{J}_{\Sigma}^{(n)}(\boldsymbol{u}, R^{(n)}) = \max_{j=1,\ldots,n} \left\{ \tilde{J}(\boldsymbol{u}, \boldsymbol{e}^{(j-1)}, R^{(n)}) \right\} \tag{18}$$

Using (18), the original cost function (6) with the weighting factor $R^{(n)}$ is approximated around different points $\boldsymbol{e}^{(j-1)}$. It is important to mention that the pointwise maximum of these approximated strict convex cost functions is also a strict convex function [27]. Therefore, $\tilde{J}_{\Sigma}^{(n)}(\boldsymbol{u}, R^{(n)})$ represents a convex hull of the original cost function $J(\boldsymbol{u}, R^{(n)})$ using the weighting matrix $R^{(n)}$ [28].

3.2. Application to a second order Volterra series model

After the general development to guarantee convexity (Section 3.1), the proposed approach is applied to a second order Volterra series model. Therefore, the output prediction of the second order

Volterra series model considering a measurable disturbance (2) is rewritten in a form similar to the one presented in [12]:

$$\boldsymbol{y} = G\boldsymbol{u} + \mathbf{f}_{nl}(\boldsymbol{u}) + \mathbf{f}_{l}(\boldsymbol{u}) + \boldsymbol{c} \tag{19}$$

$$\boldsymbol{c} = H\boldsymbol{u}_p + \mathbf{g}(\boldsymbol{u}_p) + \boldsymbol{y}_0 + \boldsymbol{d} \tag{20}$$

where $G\boldsymbol{u}$ with $G \in \mathbb{R}^{N \times N_u}$ represents the linear part of the output prediction, which depends on the future input sequence $\boldsymbol{u}$. The vectors $\mathbf{f}_{nl}(\boldsymbol{u}) \in \mathbb{R}^N$ and $\mathbf{f}_l(\boldsymbol{u}) \in \mathbb{R}^N$ contain the nonlinear future–future and the linear future–past cross terms, respectively (for an exact definition of $\mathbf{f}_{nl}(\boldsymbol{u})$ and $\mathbf{f}_l(\boldsymbol{u})$, see (25) and (27) as well as Appendix A). The term $H\boldsymbol{u}_p$ with $H \in \mathbb{R}^{N \times N_t}$ represents the linear part of the output depending on the past input values given by $\boldsymbol{u}_p = [u(k-1), \dots, u(k-N_t)]^T \in \mathbb{R}^{N_t}$. The past–past cross terms are considered in $\mathbf{g}(\boldsymbol{u}_p) \in \mathbb{R}^N$. The vector $\boldsymbol{y}_0 \in \mathbb{R}^N$ contains the model offset given by $\boldsymbol{y}_0 = [y_0, \dots, y_0]^T$, and $\boldsymbol{d} \in \mathbb{R}^N$ considers the estimation error at k defined by $\boldsymbol{d} = [d(k), \dots, d(k)]^T$. For further details on the different vectors and matrices used in (19)–(20), see [12, 29].

In a first step, the output prediction error $\boldsymbol{e} = \boldsymbol{y} - \boldsymbol{r}$ can be written for the second order Volterra series model (19)–(20) in the form given in (10):

$$\mathbf{p}(\boldsymbol{u}) = G\boldsymbol{u} + \mathbf{f}_{nl}(\boldsymbol{u}) + \mathbf{f}_{l}(\boldsymbol{u}) \tag{21}$$

$$\boldsymbol{b} = H\boldsymbol{u}_p + \mathbf{g}(\boldsymbol{u}_p) + \boldsymbol{y}_0 + \boldsymbol{d} - \boldsymbol{r} \tag{22}$$

with $G\boldsymbol{u}$, $\mathbf{f}_{nl}(\boldsymbol{u})$, and $\mathbf{f}_l(\boldsymbol{u})$ as the only terms depending on the input sequence. In a second step, the term $2\boldsymbol{e}^{(0)T} Q\, \mathbf{p}(\boldsymbol{u})$ (12) has to be decomposed in a quadratic and a linear

$$\boldsymbol{u}^T S \boldsymbol{u} = 2\boldsymbol{e}^{(0)T} Q \mathbf{f}_{nl}(\boldsymbol{u}) \tag{23}$$

$$2\boldsymbol{h}^T \boldsymbol{u} = 2\boldsymbol{e}^{(0)T} Q \left(G\boldsymbol{u} + \mathbf{f}_l(\boldsymbol{u})\right) \tag{24}$$

where the quadratic term $\mathbf{f}_{nl}(\boldsymbol{u})$ defines the matrix S and the linear terms $G\boldsymbol{u}$ and $\mathbf{f}_l(\boldsymbol{u})$ determine the vector $\boldsymbol{h}$.

With the nonlinear term $\mathbf{f}_{nl}(\boldsymbol{u})$ written in a general form as

$$\mathbf{f}_{nl}(\boldsymbol{u}) = \begin{bmatrix} \mathrm{f}_{nl,1}(\boldsymbol{u}) \\ \mathrm{f}_{nl,2}(\boldsymbol{u}) \\ \vdots \\ \mathrm{f}_{nl,N}(\boldsymbol{u}) \end{bmatrix} = \begin{bmatrix} \boldsymbol{u}^T B_{nl,1} \boldsymbol{u} \\ \boldsymbol{u}^T B_{nl,2} \boldsymbol{u} \\ \vdots \\ \boldsymbol{u}^T B_{nl,N} \boldsymbol{u} \end{bmatrix} \tag{25}$$

and the definition $2\boldsymbol{e}^{(0)T} Q = [\phi_1 \phi_2 \dots \phi_N]$, the matrix S (23) is defined as

$$S = \sum_{i=1}^{N} \phi_i B_{nl,i} \tag{26}$$

Analogously, writing the linear term $\mathbf{f}_l(\boldsymbol{u})$ as

$$\mathbf{f}_{l}(\boldsymbol{u}) = \begin{bmatrix} \mathrm{f}_{l,1} \\ \mathrm{f}_{l,2} \\ \vdots \\ \mathrm{f}_{l,N} \end{bmatrix} = \begin{bmatrix} \boldsymbol{u}_p^T B_{l,1} \\ \boldsymbol{u}_p^T B_{l,2} \\ \vdots \\ \boldsymbol{u}_p^T B_{l,N} \end{bmatrix} \boldsymbol{u} \tag{27}$$

and defining the matrix G in the following way:

$$G = \left[\begin{array}{c} \boldsymbol{\gamma}_1^T \\ \hline \boldsymbol{\gamma}_2^T \\ \hline \vdots \\ \hline \boldsymbol{\gamma}_N^T \end{array}\right] \tag{28}$$

the parameter $\boldsymbol{h}$ has the form:

$$\boldsymbol{h}^T = \frac{1}{2}\sum_{i=1}^{N} \phi_i \left(\boldsymbol{\gamma}_i^T + \boldsymbol{u}_p^T B_{l,i}\right) \tag{29}$$

The matrices $B_{nl,i}$ and $B_{l,i}$ for $i = 1, \ldots, N$ used to define the terms $\mathbf{f}_{nl}(\boldsymbol{u})$ and $\mathbf{f}_l(\boldsymbol{u})$ are given in Appendix A and in [29].

With S and $\boldsymbol{h}^T$, the original cost function based on a second order Volterra series model can be approximated by (14). By means of the design parameter R, strict convexity of the approximated cost function can be guaranteed.

4. CONTROL LAW

With the convexification procedure applied to Volterra series models (Section 3.2), a new NMPC strategy based on an iterative control law has been developed. The approximation of the original cost function by a convex hull generated by the approximating convex cost functions allows to calculate the control sequence and guarantees convergence of the solution.

4.1. Iterative optimization

In a first step, an initial candidate input sequence is used to compute a convex function, which approximates the original cost based on a second order Volterra series model. Minimizing this approximating convex function, a new candidate input sequence can be obtained. The new input sequence is then used to determine a second approximating convex function. The pointwise maximum of several convex functions represents a convex hull for the original cost function[‖]. The global minimization of the convex hull results then in a new input sequence. The difference between the original cost function and the convex hull is reduced with an increasing number of approximating convex functions. This procedure is repeated until the final accuracy tolerance is satisfied, that is, the difference between the original cost and the convex hull falls below a certain level.

The initial candidate sequence at k is given by $\boldsymbol{u}_0$, and the positive semi-definite initial weighting matrix is given as $R^{(0)} = R_{k-1}$ where R_{k-1} is the final weighting matrix at the previous sample $k-1$. Then, the iterative procedure for the optimization based on the convexification can be expressed in the following form:

1. Define $j = 0$, $\boldsymbol{u}^{(0)} = \boldsymbol{u}_0$, $R^{(0)} = R_{k-1}$, and calculate $\boldsymbol{b}$.
2. Set $j = j + 1$ and calculate $\boldsymbol{e}^{(j-1)} = \mathbf{p}(\boldsymbol{u}^{(j-1)}) + \boldsymbol{b}$.
3. Determine $S^{(j-1)}$ and $\boldsymbol{h}^{(j-1)}$ so that
$$2\boldsymbol{e}^{(j-1)T} Q\mathbf{p}(\boldsymbol{u}) = \boldsymbol{u}^T S^{(j-1)}\boldsymbol{u} + 2\boldsymbol{h}^{(j-1)T}\boldsymbol{u}$$
4. Check if $S^{(j-1)} + R^{(j-1)} > 0$. If yes, define $R^{(j)} = R^{(j-1)}$. If not, calculate the smallest eigenvalue $\lambda_{min}^{(j-1)}$ of $S^{(j-1)} + R^{(j-1)}$ and define
$$R^{(j)} = R^{(j-1)} + (-\lambda_{\min}^{(j-1)} + \varepsilon)\boldsymbol{I}_{N_u} \tag{30}$$
where $\varepsilon > 0$ is used to guarantee strict convexity.

[‖] The different approximating functions are used in the jth iteration with the maximum weighting factor $\boldsymbol{R}^{(j)}$ to assure convexity, see (17). Furthermore, the original cost function considering a weighting of the control effort is also used with the maximum weighting factor $\boldsymbol{R}^{(j)}$.

5. The sequence $\tilde{\boldsymbol{u}}$ is the solution to the optimization problem

$$\begin{aligned}\min_{\tilde{\boldsymbol{u}},\vartheta^{(j)}} \; & \vartheta^{(j)} \\ \text{s.t. } & \tilde{\boldsymbol{u}}^T \left(S^{(n)} + R^{(j)}\right) \tilde{\boldsymbol{u}} + 2\left(\boldsymbol{h}^{(n)T} - \boldsymbol{u}_r^T R^{(j)}\right) \tilde{\boldsymbol{u}} + \ldots \\ & 2\boldsymbol{e}^{(n)T} Q \boldsymbol{b} - \boldsymbol{e}^{(n)T} Q \boldsymbol{e}^{(n)} + \boldsymbol{u}_r^T R^{(j)} \boldsymbol{u}_r \leqslant \vartheta^{(j)} \\ & \tilde{u}(k+i|k) \in U, \; i = 0, \ldots, N_u - 1 \\ & \tilde{u}(k+i|k) = u_r(k), \; i = N_u, \ldots, N-1 \\ & n = 0, \ldots, j-1 \end{aligned} \tag{31}$$

6. Define the search direction $\boldsymbol{\Delta}_u = \tilde{\boldsymbol{u}} - \boldsymbol{u}^{(j-1)}$ and the step size $t = 1$ for the backtracking line search [27]. Furthermore, calculate the gradient $\nabla J(\boldsymbol{u}^{(j-1)}, R^{(j)}) = 2\left(S^{(j-1)} + R^{(j)}\right)\boldsymbol{u}^{(j-1)} + 2\left(\boldsymbol{h}^{(j-1)T} - \boldsymbol{u}_r^T R^{(j)}\right)^T$. Then, check if $J(\boldsymbol{u}^{(j-1)} + t\boldsymbol{\Delta}_u, R^{(j)}) \leqslant J(\boldsymbol{u}^{(j-1)}, R^{(j)}) + \alpha t \nabla J(\boldsymbol{u}^{(j-1)}, R^{(j)})^T \boldsymbol{\Delta}_u$ with $\alpha \in (0, 0.5)$. If yes, set $\boldsymbol{u}^{(j)} = \boldsymbol{u}^{(j-1)} + t\boldsymbol{\Delta}_u$. If not, set $t = \beta t$ with $\beta \in (0, 1)$ and repeat the line search until the previous condition is satisfied.
7. Check if the final accuracy tolerance $\|\boldsymbol{u}^{(j)} - \boldsymbol{u}^{(j-1)}\|_1 \leqslant \delta$ with $\delta \geqslant 0$ is satisfied. If the convergence condition is satisfied, set $\boldsymbol{u}^* = \boldsymbol{u}^{(j)}$, $R_k = R^{(j)}$ and apply the input sequence to the system. Otherwise return to Step 2.

Note that the presented algorithm can be considered as a descent method with backtracking line search, which possesses an approximately linear convergence. For an exhaustive study of the influence of the parameters α and β on the convergence of the given method, see [27].

4.2. *Convergence of the iterative algorithm*

Consider the solution $\tilde{\boldsymbol{u}}$ of the optimization problem given in Step 5, that is, the candidate input sequence that minimizes the convex hull defined by the pointwise maximum of the approximating convex functions $\tilde{J}_\Sigma^{(j)}(\boldsymbol{u}, R^{(j)})$ (18). As the approximating convex function $\tilde{J}(\boldsymbol{u}, R^{(j)}, \boldsymbol{e}^{(j-1)})$ represents a Taylor approximation of the original cost function $J(\boldsymbol{u}, R^{(j)})$ around the approximation point $\boldsymbol{u}^{(j-1)}$, the values of these functions satisfy

$$J(\boldsymbol{u}^{(j-1)}, R^{(j)}) = \tilde{J}(\boldsymbol{u}^{(j-1)}, R^{(j)}, \boldsymbol{e}^{(j-1)}) \tag{32}$$

and for the derivatives, holds the statement

$$\left.\frac{\partial J(\boldsymbol{u}, R^{(j)})}{\partial \boldsymbol{u}}\right|_{\boldsymbol{u}=\boldsymbol{u}^{(j-1)}} = \left.\frac{\partial \tilde{J}(\boldsymbol{u}, R^{(j)}, \boldsymbol{e}^{(j-1)})}{\partial \boldsymbol{u}}\right|_{\boldsymbol{u}=\boldsymbol{u}^{(j-1)}} \tag{33}$$

Furthermore, the input sequence of the jth iteration is defined as

$$\boldsymbol{u}^{(j)} = \boldsymbol{u}^{(j-1)} + \beta^m \boldsymbol{\Delta}_u \tag{34}$$

where m denotes the number of repetitions of the scaling carried out in Step 6 of the proposed optimization algorithm.

Consider the closed set $\boldsymbol{U} = U \times U \times \ldots \times U \subseteq \mathbb{R}^{N_u}$ defined by the constraints given in Step 5 of the proposed iterative procedure. Hence, the solution to the optimization problem in Step 5 satisfies $\tilde{\boldsymbol{u}} \in \boldsymbol{U}$. With $u^{(j-1)} \in \boldsymbol{U}$ and $\tilde{\boldsymbol{u}} \in \boldsymbol{U}$, the new candidate input sequence $\boldsymbol{u}^{(j)}$ computed in Step 6 by backtracking line search also satisfies $\boldsymbol{u}^{(j)} \in \boldsymbol{U}$. With $J(\boldsymbol{u}, R^{(j)})$ being a continuous function in $\boldsymbol{U}$, it can be shown with the extreme value theorem that the cost function $J(\boldsymbol{u}, R^{(j)})$ has an upper and lower bound. Hence, for the original cost function exists a weighting matrix R^{max}, which completely convexifies the function $J(\boldsymbol{u}, R^{max})$.

Convergence of the optimization algorithm in a finite number of iterations is then assured as follows:

(a) If the weighting matrix in the jth iteration is equal to the one of the $j-1$th iteration, that is, $R^{(j)} = R^{(j-1)}$, the value of the cost function decreases monotonically[**] and satisfies the inequality relation $J(\boldsymbol{u}^{(j)}, R^{(j)}) \leqslant J(\boldsymbol{u}^{(j-1)}, R^{(j)}) - \alpha\beta^l \nabla J(\boldsymbol{u}^{(j-1)}, R^{(j)})^T \boldsymbol{\Delta}_u$ [27]. As a consequence, the result of the optimization procedure converges in a finite number of iterations.

(b) After a finite number of modifications of the weighting matrix, that is, $R^{(j)} > R^{(j-1)}$, and under consideration of the parameter $\varepsilon > 0$ (see Step 4 of the optimization procedure), the statement $R^{(j)} \geqslant R^{max}$ holds. In this case, the function $J(\boldsymbol{u}, R^{(j)})$ is completely convexified and a further increase in the weighting matrix is not necessary.

(c) As a direct consequence of the finite number of iterations based on the cases (a) and (b), follows that the candidate input sequence $\boldsymbol{u}^{(j)}$ computed in Step 6, converges in a finite number of iterations to the solution $\boldsymbol{u}^*$ of the optimization problem.

Remark 1

In the case that the weighting matrix of the jth iteration is equal to the one of the $j-1$th iteration, that is, $R^{(j)} = R^{(j-1)}$, the cost monotonically decreases and converges to the optimal solution. If the weighting matrix of the jth iteration is not equal to the one of the $j-1$th iteration, the cost can increase with respect to the cost of the previous iteration. However, the mentioned increase in the cost is limited by the upper bound R^{max} of the weighting matrix. If the weighting matrix in the $j-1$th iteration corresponds to the upper bound R^{max}, the computed costs in the following iterations are always monotonically decreasing because a further increase of the weighting in order to convexify the cost function is not necessary.

5. ROBUST STABILITY

Input-to-state practical stability of the NMPC strategy based on the iterative convexification approach (Section 4.1) can be shown. In the case of a nominal model, that is, the estimation error satisfies $d(k) = 0$, the control strategy is asymptotic stable. The stability proof requires a reformulation of the optimization problem in a state-space-like representation.

5.1. *Optimization problem in state-space representation*

The prediction model based on a second order Volterra series model considering an estimation error (2), can be expressed in a state-space form (see Appendix B for the detailed model transformation) by

$$\begin{aligned} \boldsymbol{x}(k+i+1|k) &= A\boldsymbol{x}(k+i|k) + Bu(k+i|k) \\ y(k+i|k) &= l(\boldsymbol{x}(k+i|k)) + d(k) \end{aligned} \tag{35}$$

where $\boldsymbol{x}(k) \in \mathbb{R}^{N_t}\ \forall k$, $\boldsymbol{x}(k|k) = \boldsymbol{x}(k)$, $d(k) = y(k) - l(\boldsymbol{x}(k))$. Furthermore, all assumptions made for (2) hold for the model in state-space representation, and the state variables are defined as

$$x_i(k) = u(k-i) \quad \text{for} \quad i = 1, \dots, N_t \tag{36}$$

where $x_i(k)$ represents the ith element of $\boldsymbol{x}(k)$. Note that the evolution of the states is given by a linear model and that the nonlinearity affects only the system output by means of the function $l(\cdot)$. With respect to the nominal model, the output prediction considering the current estimation error is defined by

$$y(k+i|k) = \tilde{y}(k+i|k) + d(k) \tag{37}$$

[**] If $\boldsymbol{u}^{(j)} = \boldsymbol{u}^{(j-1)}$, the condition of a monotonic decrease in the cost is not satisfied. However, in this case, the final accuracy tolerance of Step 7 is fulfilled and the optimization algorithm terminates with the input sequence $\boldsymbol{u}^* = \boldsymbol{u}^{(j)}$.

with $\tilde{y}(k+i|k)$ being the nominal model output given by:

$$\tilde{y}(k+i|k) = l(\boldsymbol{x}(k+i|k)) \tag{38}$$

Then, with the Volterra series model (35) in state-space representation considering an estimation error, the optimization problem (5) of the NMPC strategy can be expressed as

$$\begin{aligned} \boldsymbol{u}^* = \arg \min_{\boldsymbol{u}} \ & J(\boldsymbol{u}, R^{(j)}) \\ \text{s.t.} \ & u(k+i|k) \in U, \qquad i = 0, \ldots, N_u - 1 \\ & u(k+i|k) = u_r(k), \ i = N_u, \ldots, N-1 \\ & \boldsymbol{x}(k+i|k) \in \boldsymbol{X}, \qquad i = 1, \ldots, N \end{aligned} \tag{39}$$

where $\boldsymbol{X} = U \times U \times \ldots \times U \subseteq \mathbb{R}^N$ is the set of admissible states. Note that the constraint $\boldsymbol{x}(k+i|k) \in \boldsymbol{X}$ is in fact redundant because of the constraint $u(k+i|k) \in U$ and the relation (36) between states and past input values. However, the explicit constraint in the state vector has been included in the problem (39) to follow the standard formulation for stability of MPC (see [3] for more information on this topic).

Taking into account the future input sequence $\boldsymbol{u}$, the steady-state signal $u_r(k)$, and the estimation error $d(k)$, the cost function used in (39) is defined as

$$J(\boldsymbol{u}, R^{(j)}) = \sum_{i=0}^{N_u-1} L(\boldsymbol{x}(k+i|k), u(k+i|k), d(k), R^{(j)}) + \sum_{i=N_u}^{N} L_h(\boldsymbol{x}(k+i|k) \tag{40}$$

with the quadratic stage costs $L(\cdot,\cdot,\cdot,\cdot)$ and $L_h(\cdot)$ given by

$$\begin{aligned} L(\boldsymbol{x}(k+i|k), u(k+i|k), d(k), R^{(j)}) &= \|l(\boldsymbol{x}(k+i|k)) + d(k) - r(k)\|_Q^2 + \\ & \quad \|u(k+i|k) - u_r(k)\|_{R^{(j)}}^2 \\ L_h(\boldsymbol{x}(k+i|k) &= \|l(\boldsymbol{x}(k+i|k)) + d(k) - r(k)\|_Q^2 \end{aligned} \tag{41}$$

where the function $l(\cdot)$, defined in (60) in the Appendix B, represents the output nonlinearity of the Volterra series model in state-space representation.

5.2. *Feasibility of the shifted solution*

Consider the input sequence

$$\boldsymbol{u}^*(k) = [u^*(k|k), u^*(k+1|k), \ldots, u^*(k+N_u-1|k)]^T \tag{42}$$

being at k, the optimal solution for the NMPC problem (39) with the associated minimum cost $J(\boldsymbol{u}^*(k), R^{(j)})$. Furthermore, consider the shifted sequence $\boldsymbol{u}^f(k+1) \in \mathbb{R}^{N_u}$ for $k+1$:

$$\boldsymbol{u}^f(k+1) = [u^f(k+1|k+1), \ldots, u^f(k+N_u|k+1)]^T \tag{43}$$

with the elements based on the optimal solution in k and the steady-state input for $k+1$:

$$u^f(k+i|k+1) = \begin{cases} u^*(k+i|k) & \text{for } i = 1, \ldots, N_u - 1 \\ u_r(k+1) & \text{for } i = N_u \end{cases} \tag{44}$$

Note that the first $N_u - 1$ elements of the shifted sequence $\boldsymbol{u}^f(k+1)$ are feasible as they have been computed at k with the optimal solution to the problem (39). Then, with the steady-state input signal $u_r(k+1)$ being feasible by definition, the shifted sequence $\boldsymbol{u}^f(k+1)$ is a feasible solution to the optimization problem (39) at $k+1$.

5.3. Convergence

Consider the cost $J^*(\boldsymbol{x}(k), R_{k+1})$ at k based on the control effort weighting R_{k+1} and the optimal solution $\boldsymbol{u}^*(k)$ minimizing the problem (39). Furthermore, consider the cost $J^*(\boldsymbol{x}(k+1), R_{k+1})$ at $k+1$ associated to the weighting matrix R_{k+1}. Convergence of the NMPC using the convexification approach can be guaranteed if the cost for $k+1$ is monotonically decreasing with respect to the cost for k.

Using the general definition of the cost function (40), the cost $J^*(\boldsymbol{x}(k), R_{k+1})$ at k can be written as

$$J^*(\boldsymbol{x}(k), R_{k+1}) = \sum_{i=0}^{N_u-1} L(\boldsymbol{x}^*(k+i|k), u^*(k+i|k), d(k), R_{k+1}) \sum_{i=N_u^N} L_h(\boldsymbol{x}^*(k+i|k) \quad (45)$$

where $\boldsymbol{x}^*(k+i|k)$ with $i = 0, \ldots, N$ is the state variable associated to the optimal solution $\boldsymbol{u}^*(k)$.

The following theorem characterizes the cost difference between the costs $J^*(\boldsymbol{x}(k), R_{k+1})$ at k and $J^*(\boldsymbol{x}(k+1), R_{k+1})$ at $k+1$ based on the optimal input sequences $\boldsymbol{u}^*(k)$ and $\boldsymbol{u}^*(k+1)$, respectively. This theorem shows the convergence of the proposed control law under certain conditions and will be used to prove input-to-state practical stability.

Theorem 1

Consider the cost $J^*(\boldsymbol{x}(k), R_{k+1})$ associated to the control effort weighting R_{k+1} and the optimal solution $\boldsymbol{u}^*(k)$ calculated at k minimizing the optimization problem (39). Furthermore, consider the cost $J^*(\boldsymbol{x}(k+1), R_{k+1})$ at $k+1$. The cost difference $\Delta J^*(k+1) = J^*(\boldsymbol{x}(k+1), R_{k+1}) - J^*(\boldsymbol{x}(k), R_{k+1})$ is bounded by

$$J^*(\boldsymbol{x}(k+1), R_{k+1}) - J^*(\boldsymbol{x}(k), R_{k+1}) \leqslant -L(\boldsymbol{x}^*(k|k), u^*(k|k), d(k), R_{k+1}) + c_d \cdot \|\Delta d\| \quad (46)$$

if there is a $\hat{k}$ such that $R_{\hat{k}} = R_k \ \ \forall \ \ k \geqslant \hat{k}$ and c_d is a positive and constant parameter.

Proof

See Appendix C. □

With the upper bound (46) of the cost difference $\Delta J^*(k+1)$, the terms $-L(\boldsymbol{x}^*(k|k), u^*(k|k), d(k), R_{k+1}) < 0$ and $c_d \cdot \|\Delta d\| > 0$ ensure that the cost based on the solution $\boldsymbol{u}^*(k+1)$ decreases as long as the condition $L(\boldsymbol{x}^*(k|k), u^*(k|k), d(k)), R_{k+1}) > c_d \cdot \|\Delta d\|$ is satisfied. Hence, the system is steered into the set

$$\Psi_d = \{\boldsymbol{x}^*(k|k) \colon \ L(\boldsymbol{x}^*(k|k), u^*(k|k), d(k), R_{k+1}) \leqslant c_d \cdot \|\Delta d\|\} \quad (47)$$

from any arbitrary $\boldsymbol{x}$. However, if the state evolves out of Ψ_d, the system will remain in another set from which it will evolve back to the set Ψ_d. Taking into account the inequality relation $-L(\boldsymbol{x}^*(k|k), u^*(k|k), d(k), R_{k+1}) \leqslant 0$ for any $\boldsymbol{x}^*(k|k)$, (46) can be written as

$$J^*(\boldsymbol{x}(k+1), R_{k+1}) \leqslant \ J^*(\boldsymbol{x}(k), R_{k+1}) + c_d \cdot \|\Delta d\| \quad (48)$$

Then, for any $\boldsymbol{x}(k) \in \Psi_d$, the statement:

$$J^*(\boldsymbol{x}(k), R_{k+1}) + c_d \cdot \|\Delta d\| \leqslant \max_{\boldsymbol{x} \in \Psi_d} J^*(\boldsymbol{x}, R_{k+1}) + c_d \cdot \|\Delta d\| = \beta_d \quad (49)$$

holds. From (48) and (49), follows directly that:

$$J^*(\boldsymbol{x}(k+1), R_{k+1}) \leqslant \beta_d, \qquad \forall\, \boldsymbol{x}(k) \in \Psi_d \quad (50)$$

Whenever the state enters into Ψ_d, it evolves into the set

$$\Psi_\beta = \{\boldsymbol{x} : J^*(\boldsymbol{x}, R_{k+1}) \leqslant \beta_d\} \quad (51)$$

Finally, the system may evolve out of Ψ_d, but will remain in the set Ψ_β. From the set Ψ_β, the system will be steered again into Ψ_d and so on. As a consequence, the state is ultimately bounded, and the system is stabilized using the feasible solution. Hence, the NMPC strategy based on a second order Volterra series model using the proposed convexification approach is input-to-state practical stable and maintains the system inside the set Ψ_β.

6. EXPERIMENTS

The proposed NMPC strategy based on the iterative convexification approach was applied to a continuous stirred tank reactor (CSTR). The following sections give a description of the used process, the identification of the process dynamics by a second order Volterra series, and the implementation of the control strategy. The control performance will be illustrated by means of experimental results.

6.1. Process description

The pilot plant shown in Figure 1 is used to emulate a CSTR and contains several means to cool or to heat the fluid in the reactor. The main elements of the system are the tank reactor, an electric resistance, a cooling jacket, a valve to manipulate the flow rate through the cooling jacket, and a water tank.

The pilot plant is operated with water, both in the reactor and the cooling jacket. The reactor has an electric resistance in its interior with a maximum power of 14.4 kW and can be used to supply caloric energy to the water in the reactor. On the other hand, the cooling jacket is used to reduce the caloric energy of the reactor content. The heat dissipation can be regulated by the aperture v_8 of the valve manipulating the flow rate through the cooling jacket. The cooling fluid, water, circulating

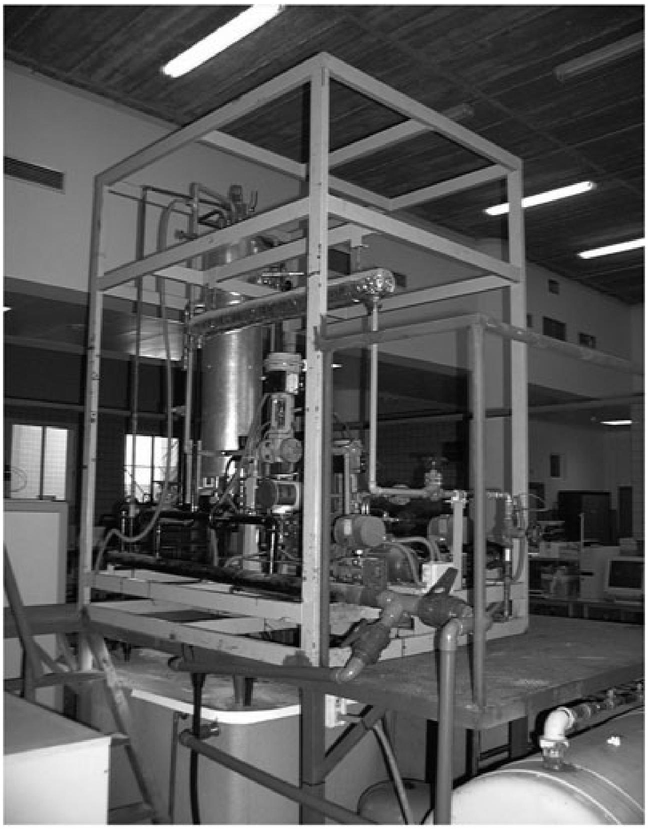

Figure 1. Pilot plant used to apply the proposed nonlinear model predictive control strategy based on the convexification approach.

DOI: 10.1002/rnc

through the cooling jacket is taken from a tank with a capacity of 1 m^3. After circulating through the jacket, the cooling fluid returns to the tank. To maintain the temperature of the cold water in a certain interval, the tank has an auxiliary cooler controlled by a thermostat, which maintains the tank temperature between 18 °C and 19 °C.

The sensors and actuators of the plant are connected to a Schneider M340 programmable automation controller (Schneider Electric, Rueil-Malmaison, France). The M340 is connected by Ethernet to a personal computer that runs the Unity Pro software package. The proposed control strategy can be implemented directly in Matlab/Simulink (MathWorks, Natick, MA, USA), and the communication with Unity Pro is performed using the object linking and embedding for process control protocol. Hence, both the Unity Pro environment and the controller implemented in Matlab/Simulink run on the same personal computer based on a Pentium 4 processor (Intel Corporation, Santa Clara, CA, USA) with 3 GHz using a Windows XP operating system.

6.2. *Emulated chemical reaction*

The pilot plant is used to emulate an exothermic chemical reaction based on temperature changes as performed in [30]. For the emulation, the energy generated by the chemical reaction is calculated with a mathematical model of the reaction and supplied by the electric resistance. The use of a resistance has the advantage that no chemical reaction takes place in the reactor while real industrial instrumentation and equipment are used.

The emulated chemical reaction considered in this document represents a refinement process where a reactant A is transformed to a substance B generating caloric energy. The reaction is generally defined by $A \to B$ and was used previously by several authors [31, 32]. The evolution of the temperature T in the reactor is described by the mathematical model

$$\frac{\mathrm{d}T}{\mathrm{d}t} = -\frac{F_j}{V}(T_{j,in} - T_{j,out}) + \frac{(-\Delta H)V}{MC_p} k_0 \, e^{\frac{-E}{RT}} C_A^2 \tag{52}$$

where the first term considers the heat dissipation by the cooling jacket and the second term denotes the heat generated by the exothermic chemical reaction. The variables F_j, $T_{j,in}$ y $T_{j,out}$ represent the flow rate through the cooling jacket and the temperature of the cooling fluid entering and leaving the cooling jacket, respectively. The concentration of the reactant C_A in the reactor content is defined by

$$\frac{\mathrm{d}C_A}{\mathrm{d}t} = \frac{F_f}{V}(C_{A,in} - C_A) - k_0 \, e^{\frac{-E}{RT}} C_A^2 \tag{53}$$

where the first term represents changes in the reactant concentration because of the feed and the outflow and the second term considers the reduction of the concentration as a result of the reactant consumption by the chemical reaction. The variables F_f and $C_{A,in}$ denote the feed and the reactive concentration in the feed. The caloric energy generated by the emulated reaction is given in the following form:

$$P = C_p M \frac{\mathrm{d}T}{\mathrm{d}t} \tag{54}$$

being C_p the specific heat capacity of the reactor content and M is the corresponding mass. Hence, (52), (53), and (54) are employed for emulation purposes where (52) is only used to compute the temperature gradient necessary to determine the amount of caloric energy generated by the emulated exothermic reaction. Finally, the necessary energy is supplied to the plant by adjusting the duty cycle of the electric resistance in accordance to the power (54). The parameters used in (52)-(54) are given in Table I, for further details on the plant and the emulated reaction see [29].

6.3. *Identification*

A simple input sequence (pseudo random multilevel sequence [33, 34]) with three different levels for the recirculation valve was applied to the pilot plant in order to collect suitable input/output data

DOI: 10.1002/rnc

Table I. Model parameters of the emulated exothermic chemical reaction.

Name	Value
k_0	$1.265 \cdot 10^{17}$ l/mol s
C_p	4.18 kJ/K kg
ΔH	−105.57 kJ/mol
$C_{A,in}$	1.2 mol/l
M	25 kg
F_f	0.05 l/s
$T_{j,in}$	291.15 K
E/R	13 550 K
V	25 l

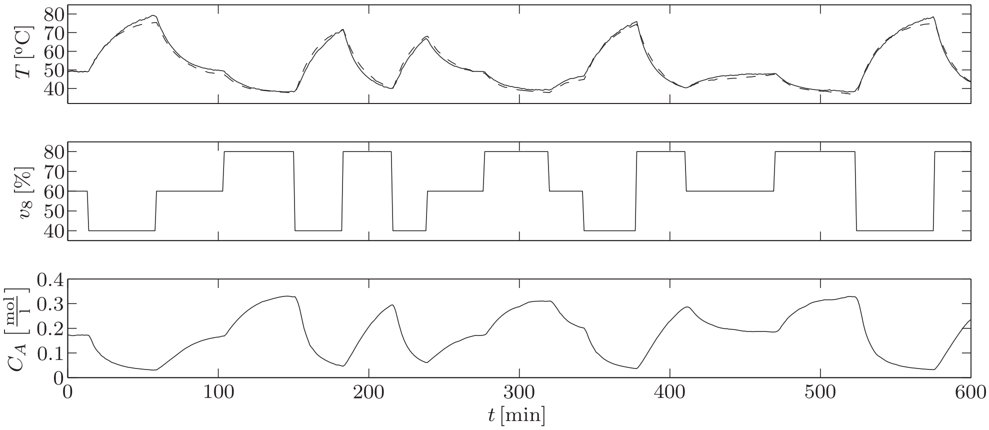

Figure 2. Comparison of the experimental data (solid line) used for the identification and the model output of the Volterra model (dashed line). From top to bottom: tank temperature T, aperture of the valve v_8, and reactant concentration C_A.

for the parameter identification of the second order Volterra model. The periods of the chosen input sequence were long enough to observe the reaction of the pilot plant (Figure 2).

The parameter identification for the second order Volterra model was carried out with the least squares method using the obtained experimental input/output data. During the identification process, it was observed that the election of a sampling time of $t_m = 60$ s represents a good compromise between the goodness of the identification and the number of identified parameters. In order to reduce the number of parameters to be identified, a diagonal model was used, that is, $b_{i,j} = 0$ for $i \neq j$. Finally, the second order diagonal Volterra model was identified with a truncation of $N_1 = 60$ linear and $N_2 = 30$ s order parameters and a delay of 1 sampling period (Figure 3). A comparison of the measured tank temperature and the output of the identified model is given in Figure 2.

6.4. Results

The proposed NMPC control strategy (Section 4) based on the identified Volterra series model (Section 6.3) was implemented in Matlab/Simulink. Sequential quadratic programming (SQP)[††] was used to minimize the approximated cost (31) and to compute the input sequence. A prediction horizon of $N = 80$, a control horizon of $N_u = 15$, and a final accuracy of $\delta = 10^{-3}$ have been used for the proposed control strategy. Furthermore, the input constraints

$$\begin{array}{rcl} 5 \leqslant & u(k+i|k) & \leqslant 100, \; i = 0, \dots, 14 \\ -20 \leqslant & \Delta u(k+i|k) & \leqslant 20, \quad i = 0, \dots, 14 \end{array} \tag{55}$$

[††] Note that the solution to the optimization problem can be found by standard convex optimization algorithms.

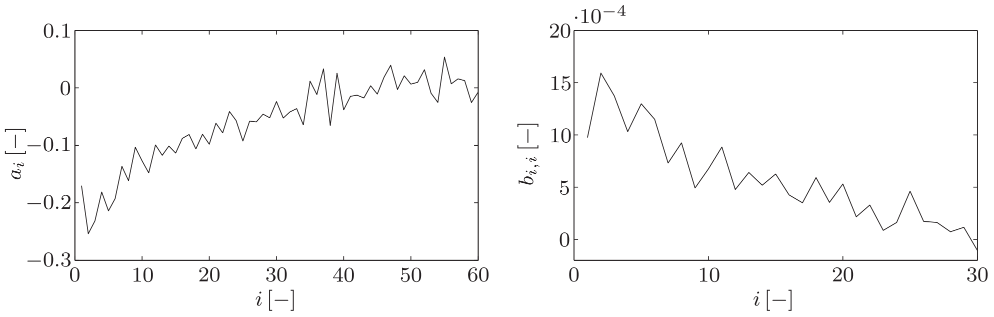

Figure 3. Identified parameters a_i and $b_{i,i}$ for the diagonal second order Volterra series model approximating the pilot plant emulating an exothermic chemical reaction.

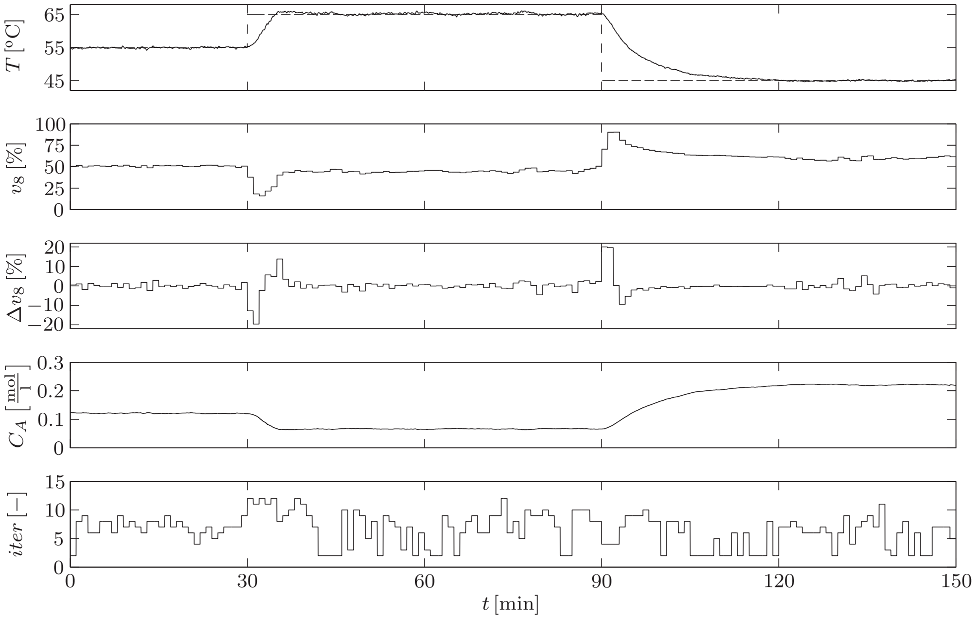

Figure 4. Setpoint tracking experiment. From top to bottom: tank temperature T, aperture of the valve v_8, increment in the valve aperture Δv_8, reactant concentration C_A, and necessary iterations $iter$.

have been considered. Note that weighting matrix $R^{(0)}$ for the control signal has been initialized in each sample with the zero matrix $0_{N_u \times N_u} \in \mathbb{R}^{N_u \times N_u}$ in order to obtain a faster dynamic behavior of the developed control.

In a first step, a setpoint tracking experiment (Figure 4) was carried out with the pilot plant emulating an exothermic chemical reaction and controlled by the NMPC strategy based on the convexification approach with guaranteed stability. During the experiment, the setpoint was changed twice, first from 55 °C to 65 °C and later to 45 °C. Only after the first setpoint change was a small overshoot observed (approximately 1.2 °C). After both setpoint changes, when the system reaches steady state, the control signal shows only insignificant changes, necessary to maintain the temperature in the given setpoint. In the shown results, the computation of the input sequence requires between 2 and 12 iterations.

In the second experiment, the disturbance rejection capabilities of the proposed controller were proven by means of a constant error in the parameter E of the emulated exothermic chemical reaction model and by changes in the given setpoint (Figure 5). As the parameter E has a strong influence on the dynamic behavior of the emulated reaction, the introduced error corresponds only to 3% of the original value of E. After the setpoint changes, the system shows an overshoot of approximately

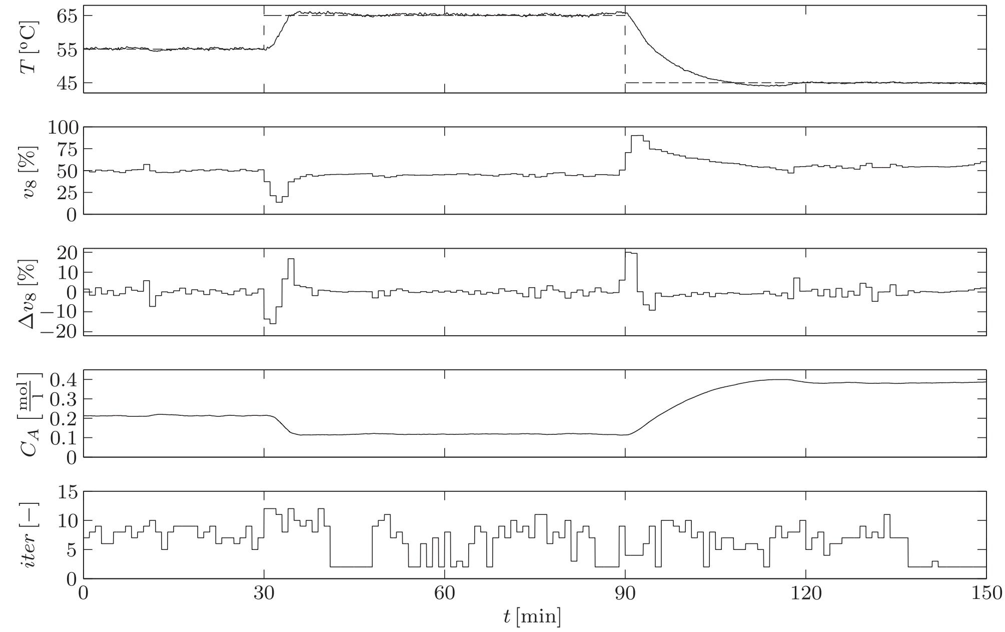

Figure 5. Disturbance rejection experiment (persistent disturbance in the emulated chemical reaction). From top to bottom: tank temperature T, aperture of the valve v_8, increment in the valve aperture Δv_8, reactant concentration C_A, and necessary iterations *iter*.

1.6 °C and −2 °C. The higher overshoot in comparison to the results given in Figure 4 is a result of the introduced model mismatch. In spite of the error in the parameter E, the proposed control strategy stabilizes the system around the given reference and shows only negligible changes in the control action for the system in steady-state. The necessary iterations to compute the input sequence are similar to the ones obtained in the first experiment (compare Figure 4) and vary between 2 and 12 iterations.

The third experiment was carried out with an additive disturbance in the system input (Figure 6). Therefore, an error of $\Delta v_8 = -15\%$ was applied to the valve opening v_8 in $t = 70$ min and removed in $t = 110$ min. During the application of the disturbance, the effective valve opening is given by $v_8 = u + \Delta v_8$, whereas in absence of the disturbance, the valve opening is defined by $v_8 = u$. After the application of the disturbance, the temperature increases rapidly because of the reduced effective opening of the valve v_8. Due to this error, the proposed controller increases the valve opening and reduces the divergence between the measured temperature and the setpoint and, as a consequence, rejects the applied disturbance successfully. After the removal of the disturbance, the controller compensates the error in the output and stabilizes the system in the setpoint. In spite of the amplitude of the chosen disturbance, the proposed predictive control strategy rejects the disturbance efficiently and leads the system to the desired setpoint.

It can be observed in the presented experimental results that the NMPC satisfied the constraints (55) imposed to the input sequence and its increments. Furthermore, the proposed control strategy always solved the optimization problem within the used sampling time of $t_s = 60$ s. During the experiments, the average computation time to compute a new input sequence was $t_c^{avg} = 1.115$ s, with a minimum of $t_c^{min} = 0.2$ s and a maximum of $t_c^{max} = 7.945$ s. The average number of necessary iterations to find the solution to the optimization problem was 6.35, with a maximum of 12 and a minimum of 2 iterations.

Finally, the solution computed by the proposed convexification approach was compared in simulations to the solution obtained solving the original optimization problem with an SQP algorithm. The simulations were carried out by means of a Matlab/Simulink model of the exothermic chemical reaction and the pilot plant using a personal computer based on an Intel i3-2330M processor running Windows 7. The results of the simulated setpoint tracking are given in Figure 7.

DOI: 10.1002/rnc

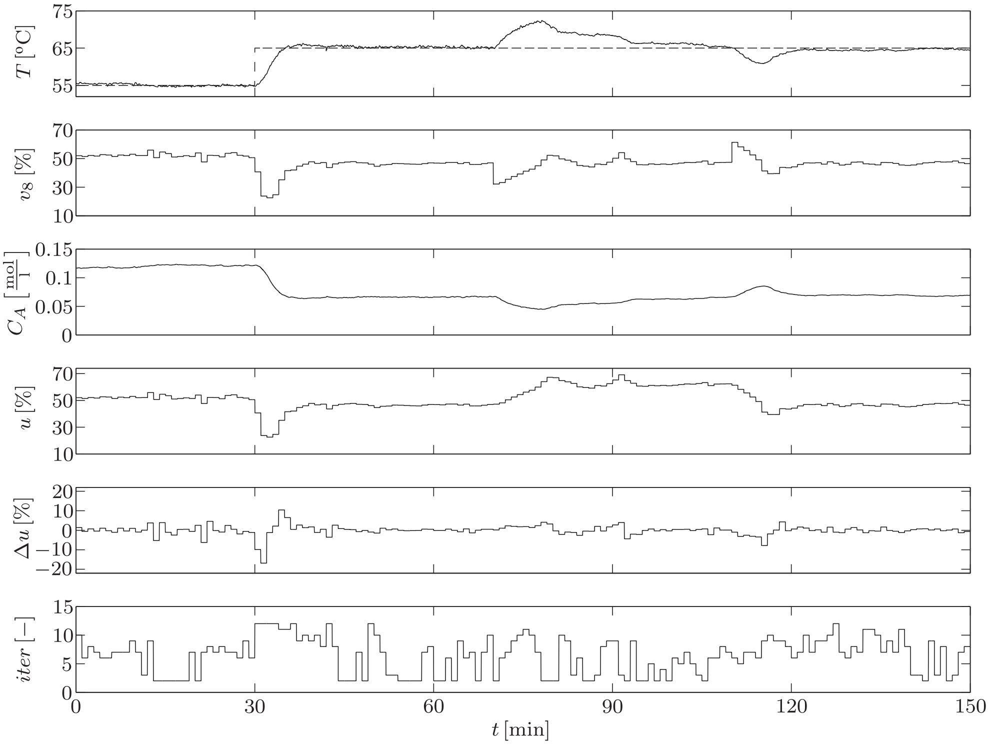

Figure 6. Disturbance rejection experiment (disturbance in the valve opening v_8). From top to bottom: tank temperature T, aperture of the valve v_8, reactant concentration C_A, input value u, and increment Δu calculated by the controller and necessary iterations *iter*.

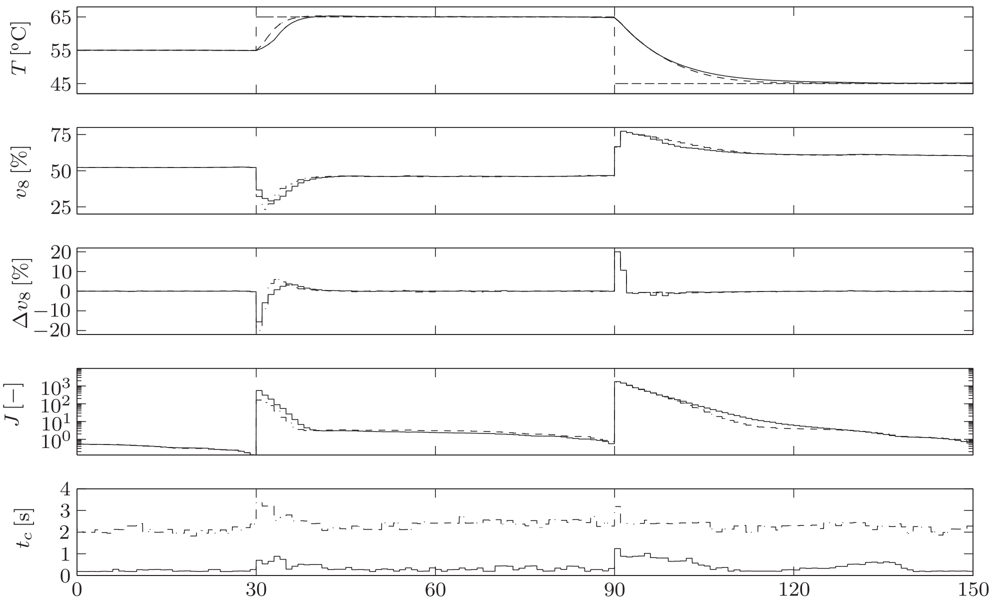

Figure 7. Simulation results of the convexification approach (solid line) and the sequential quadratic programming algorithm (dash-dotted line). From top to bottom: tank temperature T, aperture of the valve v_8, increment in the valve aperture Δv_8, cost J, and computation time t_c.

DOI: 10.1002/rnc

The solution computed directly with the SQP algorithm shows a slightly faster reaction to setpoint changes and a lower cost (3) during transitions. The use of the proposed optimization approach and the SQP algorithm led in the given simulation results to mean square errors in the tank temperature of 3.47 and 3.33, respectively. In contrast, the convexification required on average only 0.351 s to obtain the solution, whereas the SQP algorithm needed about 2.306 s to solve the optimization problem. An alternative is the use of first principles nonlinear models in a predictive control framework. However, this kind of model is very difficult to obtain for real processes, and the possible model mismatch adds some extra difficulty to the control problem, requiring in many cases the use of computationally more complex robust control strategies such as min-max MPC.

7. CONCLUSIONS

This work presented a new NMPC strategy based on the convexification of the performance index to be minimized. The control strategy has been developed for quadratic cost functions considering second order Volterra series prediction models.

The proposed control strategy is based on the approximation of the original cost function by convex quadratic functions. With a weighting of the control effort, convexity of the approximated functions can be assured by a suitable choice of the weighting matrix R. It can be shown that the original cost function with a weighting matrix R can be approximated by means of a convex hull based on the pointwise maximum of several convex quadratic functions using the same weighting matrix R. Then, globally minimizing the convex hull, a new input sequence can be calculated. With the optimization routine based on a decent method with backtracking line search, convergence of the procedure can be shown. Furthermore, input-to-state practical stability can be proven for the proposed NMPC strategy (in the case of a nominal model asymptotic stability can be ensured).

The proposed NMPC strategy was applied to a CSTR, and several setpoint tracking and disturbance rejection experiments were carried out. The control strategy showed a good performance and stabilized the system in the given setpoint, even in presence of strong disturbances. The necessary computational effort allowed to compute the control sequence within the used sampling time. Finally, the control performance was illustrated by experimental results.

APPENDIX A: MATRIX DEFINITIONS

In order to apply the proposed NMPC strategy using second order Volterra series models, the matrices B_l and B_{nl} for the linear and nonlinear terms $\mathbf{f}_l(\boldsymbol{u})$ and $\mathbf{f}_{nl}(\boldsymbol{u})$ have to be defined (Section 3.2). The parameters N_t, N, and N_u represent the truncation order, the prediction horizon, and the control horizon, respectively. The matrix $B_l \in \mathbb{R}^{N_t \times N_u \times N}$ or the submatrices $B_{l,i} \in \mathbb{R}^{N_t \times N_u}$ with $i = 1, \ldots, N$ are defined as

$$B_{l,i} = \begin{bmatrix} b_{i,i+1} & \dots & b_{i-(N_u-2),i+1} & \sum\limits_{j=1}^{i-N_u+1} b_{j,i+1} \\ b_{i,i+2} & \dots & b_{i-(N_u-2),i+2} & \sum\limits_{j=1}^{i-N_u+1} b_{j,i+2} \\ \vdots & \vdots & \vdots & \vdots \\ b_{i,N_t} & \dots & b_{i-(N_u-2),N_t} & \sum\limits_{j=1}^{i-N_u+1} b_{j,N_t} \\ 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{56}$$

DOI: 10.1002/rnc

Analogously, the matrix $B_{nl} \in \mathbb{R}^{N_u \times N_u \times N}$ or the submatrices $B_{nl,i} \in \mathbb{R}^{N_u \times N_u}$ with $i = 1, \dots, N_t$ are defined as

$$B_{nl,i} = \begin{bmatrix} b_{i,i} & \dots & b_{i-(N_u-2),i} & \sum\limits_{j=1}^{i-N_u+1} b_{j,i} \\ 0 & \dots & b_{i-(N_u-2),i-1} & \sum\limits_{j=1}^{i-N_u+1} b_{j,i-1} \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & b_{i-(N_u-2),i-(N_u-2)} & \sum\limits_{j=1}^{i-N_u+1} b_{j,i-(N_u-2)} \\ 0 & \dots & 0 & \sum\limits_{k=1}^{i-N_u+1} \sum\limits_{j=1}^{k} b_{j,k} \end{bmatrix} \tag{57}$$

APPENDIX B: VOLTERRA SERIES MODELS IN STATE-SPACE-LIKE REPRESENTATION

Generally, a non-autoregressive second order Volterra series model as the one given in (1) can be described as a discrete state-space-like model. The state-space-like representation has a special importance for the stability proof (Section 5) for the proposed NMPC strategy.

In a first step, the past input values $u(k-i)$ with $i = 1, \cdots, N_t$ of the non-autoregressive second order Volterra series model (1) can be considered as system states, for example, the model states are defined by

$$x_i(k) = u(k-i) \quad \text{for} \quad i = 1, \dots, N_t \tag{58}$$

where $x_i(k)$ is the ith element of the state vector $\boldsymbol{x}(k) \in \mathbb{R}^{N_t}$. It can be seen easily that the definition of the states (58) can be rewritten in the form

$$\begin{aligned} x_1(k) &= u(k-1) \\ x_2(k) &= x_1(k-1) \\ x_3(k) &= x_2(k-1) \\ \vdots\ &\vdots \quad\ \vdots \\ x_{N_t}(k) &= x_{N_t-1}(k-1) \end{aligned} \tag{59}$$

where the first state $x_i(k)$ represents the last applied input signal and the remaining states depend on the states from the previous instants. With the states $x_i(k)$ for $i = 1, \dots, N_t$ defined in (59), the model output of the second order Volterra series model (1) can be expressed by

$$y(k) = l(\boldsymbol{x}(k)) = \sum_{i=1}^{N_t} a_i x_i(k) + \sum_{i=1}^{N_t} \sum_{j=i}^{N_t} b_{i,j} x_i(k) x_j(k) \tag{60}$$

Note the similarity of (60) and (1) where the past input values $u(k-i)$ for $i = 1, \dots, N_t$ have been substituted by the previously defined states $x_i(k)$ for $i = 1, \dots, N_t$. Now, it can be seen easily that (1) can be expressed as a nonlinear state-space-like model defined by

$$\begin{aligned} \boldsymbol{x}(k+1) &= A\boldsymbol{x}(k) + Bu(k) \\ y(k) &= l(\boldsymbol{x}(k)) \end{aligned} \tag{61}$$

Taking into account (59), the state matrix $A \in \mathbb{R}^{N_t \times N_t}$ and the input matrix $B \in \mathbb{R}^{N_t}$ are given by

$$A = \begin{bmatrix} 0\;0 & \dots & 0\;0\;0 \\ 1\;0 & \dots & 0\;0\;0 \\ 0\;1 & \dots & 0\;0\;0 \\ \vdots\;\vdots & \ddots & \vdots\;\vdots\;\vdots \\ 0\;0 & \dots & 1\;0\;0 \\ 0\;0 & \dots & 0\;1\;0 \end{bmatrix}, \qquad B = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix} \tag{62}$$

The model in state-space representation (61) can be used easily as a prediction model in an NMPC framework. Under consideration of the current estimation error $d(k)$, (61) can be rewritten in the following form:

$$\begin{aligned} \boldsymbol{x}(k+i+1|k) &= A\boldsymbol{x}(k+i|k) + Bu(k+i|k) \\ y(k+i|k) &= l(\boldsymbol{x}(k+i|k)) + d(k) \end{aligned} \tag{63}$$

APPENDIX C: PROOF OF THEOREM 1

This section presents the detailed proof of Theorem 1 (Section 5.3) used to demonstrate input-to-state practical stability of the NMPC strategy based on the convexification approach.

C.1. General statements

The following lemmas and theorems have been applied to define an upper bound for the cost difference (46):

Lemma 1
A quadratic function $g(a) = a^2$ is locally Lipschitz continuous in $a \in [b_1, b_2]$ with $-\infty < b_1 \leqslant b_2 < \infty$. With this condition, a Lipschitz constant L_q can be found such that $\|g(a_1+a_2)-g(a_1)\| \leqslant L_q\|a_2\|$.

Lemma 2
The steady-state input function $\varphi(\cdot)$ (4) of the Volterra model in state-space representation is Lipschitz continuous and, as a consequence, the inequality condition $\|\varphi(a_1+a_2)-\varphi(a_1)\| \leqslant L_\varphi\|a_2\|$ is satisfied.

Lemma 3
The output nonlinearity $l(\cdot)$ (60) of the Volterra model in state-space representation is Lipschitz continuous and can be bounded by $\|l(\boldsymbol{a}_1+\boldsymbol{a}_2)-l(\boldsymbol{a}_1)\| \leqslant L_l\|\boldsymbol{a}_2\|$.

Theorem 2
Consider the steady-state input difference $\Delta u_r = u_r(k+1) - u_r(k)$ where $u_r(k)$ and $u_r(k+1)$ denote the steady-state inputs at k and $k+1$, respectively. Furthermore, consider the state values $\boldsymbol{x}^f(k+i|k+1)$ predicted at $k+1$ using the feasible solution $\boldsymbol{u}^f(k+1)$ given in (43). Being c_x, a positive and constant parameter, the difference between the states $\Delta\boldsymbol{x}(k+i) = \boldsymbol{x}^f(k+i|k+1) - \boldsymbol{x}^*(k+i|k)$ for $i = N_u+1, \dots, N$ is bounded by

$$\|\Delta\boldsymbol{x}(k+i)\| \leqslant c_x\|\Delta u_r\| \tag{64}$$

Proof
The predicted states satisfy $\boldsymbol{x}^f(k+i|k+1) = \boldsymbol{x}^*(k+i|k)$ for $i = 1, \dots, N_u$ as $u^f(k+i|k+1) = u^*(k+i|k)$ for $i = 1, \dots, N_u - 1$. Furthermore, the prediction of the states based on the optimal and the feasible solution are defined for $i = N_u+1, \dots, N$ by

$$\begin{aligned} \boldsymbol{x}^*(k+i|k) &= A\boldsymbol{x}^*(k+i-1|k) + Bu_r(k) \\ \boldsymbol{x}^f(k+i|k+1) &= A\boldsymbol{x}^f(k+i-1|k+1) + Bu_r(k+1) \end{aligned} \tag{65}$$

DOI: 10.1002/rnc

Hence, the difference between the states $\Delta \boldsymbol{x}(k+i) = \boldsymbol{x}^f(k+i|k+1) - \boldsymbol{x}^*(k+i|k)$ for $i = N_u + 1, \dots, N$ can be written as

$$\Delta \boldsymbol{x}(k+i) = A\Delta \boldsymbol{x}(k+i-1) + B\Delta u_r \tag{66}$$

where $\Delta u_r = u_r(k+1) - u_r(k)$ represents the difference of the steady-state inputs at $k+1$ and k. Taking into account that $\Delta \boldsymbol{x}(k+N_u) = 0$, the recursion (66) ensures the existence of $c_x = \|A^{N-(N_u+1)}B + \cdots + B\|$ with $c_x > 0$ such that $\|\Delta \boldsymbol{x}(k+i)\| \leqslant c_x \|\Delta u_r\|$ for $i = N_u + 1, \dots, N$. □

C.2. Definition of the cost difference

Consider the cost $J^f(\boldsymbol{x}(k+1), R_{k+1})$ at $k+1$ depending on the feasible solution $\boldsymbol{u}^f(k+1)$ (43) and the control weighting effort R_{k+1}, which can be expressed as

$$\begin{aligned} J^f(\boldsymbol{x}(k+1), R_{k+1}) = & \sum_{i=1}^{N_u-1} L(\boldsymbol{x}^f(k+i|k+1), u^f(k+i|k+1), d(k+1), R_{k+1}) + \\ & \sum_{i=N_u}^{N+1} L_h(\boldsymbol{x}^f(k+i|k+1) \end{aligned} \tag{67}$$

Based on the costs given in (67) and (45), the cost difference $\Delta J(k+1) = J^f(\boldsymbol{x}(k+1), R_{k+1}) - J^*(\boldsymbol{x}(k), R_{k+1})$ becomes

$$\begin{aligned} \Delta J(k+1) = & L_h(\boldsymbol{x}^f(k+N+1|k+1) - L(\boldsymbol{x}^*(k|k), u^*(k|k), d(k), R_{k+1}) + \\ & \sum_{i=1}^{N_u-1} (L(\boldsymbol{x}^f(k+i|k+1), u^f(k+i|k+1), d(k+1), R_{k+1}) - \\ & \qquad L(\boldsymbol{x}^*(k+i|k), u^*(k+i|k), d(k), R_{k+1})) + \\ & \sum_{i=N_u}^{N} (L_h(\boldsymbol{x}^f(k+i|k+1) - L_h(\boldsymbol{x}^*(k+i|k)) \end{aligned} \tag{68}$$

Then, the difference of the two cost functions (68) can be written in the following form:

$$\Delta J(k+1) = -L(\boldsymbol{x}^*(k|k), u^*(k|k), d(k), R_{k+1}) + \alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 \tag{69}$$

with the terms $\alpha_1, \alpha_2, \alpha_3$ and α_4 given by:

$$\begin{aligned} \alpha_1 &= \sum_{i=1}^{N_u-1} \Big(\|l(\boldsymbol{x}^f(k+i|k+1)) + d(k+1) - r(k+1)\|_Q^2 \\ & \qquad - \|l(\boldsymbol{x}^*(k+i|k)) + d(k) - r(k)\|_Q^2 \Big) \\ \alpha_2 &= \sum_{i=1}^{N_u-1} \Big(\|u^f(k+i|k+1) - u_r(k+1)\|_{R_{k+1}}^2 - \|u^*(k+i|k) - u_r(k)\|_{R_{k+1}}^2 \Big) \\ \alpha_3 &= \sum_{i=N_u}^{N} \Big(\|l(\boldsymbol{x}^f(k+i|k+1)) + d(k+1) - r(k+1)\|_Q^2 \\ & \qquad - \|l(\boldsymbol{x}^*(k+i|k)) + d(k) - r(k)\|_Q^2 \Big) \\ \alpha_4 &= \|l(\boldsymbol{x}^f(k+N+1|k+1)) + d(k+1) - r(k+1)\|_Q^2 \end{aligned}$$

The following theorems define upper bounds of the terms α_1, α_2, α_3, and α_4 used in the cost difference (69):

Theorem 3
Consider the estimation error increment given by $\Delta d = d(k+1) - d(k)$ where $d(k+1)$ and $d(k)$ denote the estimation errors at $k+1$ and k, respectively. Then, the term α_1 used in (69) is bounded by

$$\alpha_1 \leqslant c_1(Q, N_u) \cdot \|\Delta d\| \tag{70}$$

where c_1 denotes a positive Lipschitz constant.

Proof
With $\boldsymbol{x}^*(k+1|k) = \boldsymbol{x}(k+1)$ and the solutions $u^f(k+i|k+1) = u^*(k+i|k)$ for $i = 1, \ldots, N_u-1$, the predicted states satisfy $\boldsymbol{x}^f(k+i|k+1) = \boldsymbol{x}^*(k+i|k)$ for $i = 1, \ldots, N_u - 1$. For the reference applies $r(k) = r(k+1)$, and the increment in the estimation error is given generally as $\Delta d = d(k+1) - d(k)$. Defining the auxiliary variable $z(k+i|k) = l(\boldsymbol{x}^*(k+i|k)) + d(k) - r(k)$, the term α_1 can be written as

$$\alpha_1 = \sum_{i=1}^{N_u-1} \left(\|z(k+i|k) + \Delta d\|_Q^2 - \|z(k+i|k)\|_Q^2\right) \tag{71}$$

Taking into account Lemma 1, the term α_1 (71) can be bounded by

$$\alpha_1 \leqslant c_1(Q, N_u) \cdot \|\Delta d\| \tag{72}$$

being $c_1(\cdot,\cdot)$ a positive Lipschitz constant which depends on the weighting factor Q and the control horizon N_u. □

Theorem 4
Consider the estimation error increment given by $\Delta d = d(k+1) - d(k)$ where $d(k+1)$ and $d(k)$ denote the estimation errors at $k+1$ and k, respectively. Then, the term α_2 in (69) can be bounded by

$$\alpha_2 \leqslant c_2(R_{k+1}, L_\varphi, N_u) \cdot \|\Delta d\| \tag{73}$$

being c_2 a positive Lipschitz constant.

Proof
Consider the optimal and the feasible solution satisfying $u^f(k+i|k+1) = u^*(k+i|k)$ for $i = 1, \ldots, N_u - 1$. Defining the increment in the steady-state input as $\Delta u_r = u_r(k+1) - u_r(k)$, and using the auxiliary variable $z_1(k+i|k) = u^*(k+i|k) - u_r(k)$, the term α_2 can be expressed as

$$\alpha_2 = \sum_{i=1}^{N_u-1} \|z_1(k+i|k) - \Delta u_r\|_{R_{k+1}}^2 - \|z_1(k+i|k)\|_{R_{k+1}}^2 \tag{74}$$

Then, applying Lemma 1 to (74), the term α_2 is bounded by

$$\alpha_2 \leqslant c_2(R_{k+1}, N_u) \cdot \|\Delta u_r\| \tag{75}$$

where $c_2(\cdot,\cdot)$ is a positive parameter. Based on the definition (4), the increment in the steady-state input can be written as

$$\Delta u_r = \varphi(r(k+1) - d(k+1)) - \varphi(r(k) - d(k)) \tag{76}$$

The increment in the estimation error is defined as $\Delta d = d(k+1) - d(k)$, and for the reference applies $r(k) = r(k+1)$. Defining the auxiliary variable $z_2 = r(k) - d(k)$, the increment in the steady-state input can be expressed as

$$\Delta u_r = \varphi(z_2 - \Delta d) - \varphi(z_2) \tag{77}$$

DOI: 10.1002/rnc

Taking into account Lemma 2, the norm of Δu_r is bounded by

$$\|\Delta u_r\| \leqslant L_\varphi \|\Delta d\| \tag{78}$$

Finally, using the bound for $\|\Delta u_r\|$ in (75), the term α_2 is limited to

$$\alpha_2 \leqslant c_2(R_{k+1}, N_u, L_\varphi) \cdot \|\Delta d\| \tag{79}$$

where $c_2(\cdot,\cdot,\cdot)$ is a positive Lipschitz constant, which depends on the weighting factor R_{k+1}, the control horizon N_u, and the parameter L_φ. □

Theorem 5
Consider the estimation error increment given by $\Delta d = d(k+1) - d(k)$ based on the estimation errors $d(k+1)$ and $d(k)$ at $k+1$ and k, respectively. Then, the term α_3 used in (69) is bounded by

$$\alpha_3 \leqslant c_3(Q, L_\varphi, L_l, c_x, N, N_u) \cdot \|\Delta d\| \tag{80}$$

with c_3 being a positive Lipschitz constant.

Proof
Consider the predictions $\boldsymbol{x}^*(k+i|k)$ and $\boldsymbol{x}^f(k+i|k+1)$ for $i = N_u, \dots, N$ made at k and $k+1$ with the optimal and the feasible solution, respectively. The difference between these predictions is defined as $\Delta\boldsymbol{x}(k+i) = \boldsymbol{x}^f(k+i|k+1) - \boldsymbol{x}^*(k+i|k)$ for $i = N_u, \dots, N$ and the initial condition $\boldsymbol{x}^f(k+N_u-1|k+1) = \boldsymbol{x}^*(k+N_u-1|k)$. Furthermore, consider the estimation error increment $\Delta d = d(k+1) - d(k)$ and the auxiliary variables

$$\begin{aligned} z_1(k+i|k) &= l(\boldsymbol{x}^*(k+i|k) + \Delta\boldsymbol{x}(k+i)) - l(\boldsymbol{x}^*(k+i|k)) + \Delta d \\ z_2(k+i|k) &= l(\boldsymbol{x}^*(k+i|k)) + d(k) - r(k) \end{aligned}$$

Then, assuming a constant reference, that is, $r(k+1) = r(k)$, the term α_3 can be expressed as

$$\alpha_3 = \sum_{i=N_u}^{N} \|z_1(k+i|k) + z_2(k+i|k)\|_Q^2 - \|z_2(k+i|k)\|_Q^2$$

Taking into account Lemma 1, the term α_3 can be bounded by

$$\alpha_3 \leqslant c_3(Q) \cdot \sum_{i=N_u}^{N} \|z_1(k+i|k)\| \tag{81}$$

Now, with the function $l(\cdot)$ being Lipschitz continuous, the term $z_1(k+i|k)$ can be bounded using Lemma 3:

$$\begin{aligned} \|z_1(k+i|k)\| &\leqslant \|l(\boldsymbol{x}^*(k+i|k)) + \Delta\boldsymbol{x}(k+i) - l(\boldsymbol{x}^*(k+i|k))\| + \|\Delta d\| \\ &\leqslant L_l\|\Delta\boldsymbol{x}(k+i)\| + \|\Delta d\| \end{aligned} \tag{82}$$

Then, the upper bound (81) of the term α_3 can be rewritten as

$$\alpha_3 \leqslant c_3(Q) \cdot \sum_{i=N_u}^{N} (L_l\|\Delta\boldsymbol{x}(k+i)\| + \|\Delta d\|) \tag{83}$$

With Theorem 2, the difference of the predicted states based on the optimal and the feasible solution can be bounded with $\|\Delta\boldsymbol{x}(k+i)\| \leqslant c_x\|\Delta u_r\|$. Using this bound in (83) and removing the sum, the term α_3 can be expressed as

$$\alpha_3 \leqslant c_3(Q, N, N_u) \cdot (L_l c_x\|\Delta u_r\| + \|\Delta d\|) \tag{84}$$

Taking into account Lemma 2, the increment in the necessary steady-state input can be bounded by $\|\Delta u_r\| \leqslant L_\varphi\|\Delta d\|$ (Proof of Theorem 4). Hence, (84) can be modified and the upper bound of α_3 is defined by

$$\alpha_3 \leqslant c_3(Q, N, N_u, L_\varphi, L_l, c_x) \cdot \|\Delta d\| \tag{85}$$

DOI: 10.1002/rnc

where $c_3(\cdot,\cdot,\cdot,\cdot,\cdot,\cdot)$ denotes a positive Lipschitz constant, which depends on the weighting factor Q, the prediction horizon N, the control horizon N_u, and the parameters L_φ, L_l, and c_x. □

Theorem 6
Consider the truncation order, the prediction horizon, and the control horizon satisfying $N \geqslant N_t + N_u$. Then, the term α_4 used in (69) is given by

$$\alpha_4 = 0 \tag{86}$$

Proof
Consider the nilpotent character of the used prediction model (35) with $A^{N_t} = 0$. For a prediction horizon of $N \geqslant N_u + N_t$, the steady-state input (4) is used (at least) in the last N_t sampling periods. As a consequence of the property $A^{N_t} = 0$, the state $\boldsymbol{x}^f(k+N+1|k+1)$ calculated at $k+1$ reaches steady state. Taking into account the definition of the steady-state input (4), it is clear that the nominal output in $k+N+1$ is given by $\tilde{y}(k+N+1|k+1) = r(k+1) - d(k+1)$. From the definition of the nominal output (38), it follows that

$$l(\boldsymbol{x}^f(k+N+1|k+1)) = r(k+1) - d(k+1) \tag{87}$$

Taking into account (87), the term α_4 is given by

$$\alpha_4 = 0 \tag{88}$$

for a prediction horizon of $N \geqslant N_u + N_t$. □

C.3. Upper bound of the cost difference

Under consideration of the upper bounds of the terms α_1, α_2 and α_3 as well as $\alpha_4 = 0$ defined in the previous section, Theorem 1 can be proven by the following proof:

Proof of Theorem 1
Consider the cost difference $\Delta J(k+1) = J^f(\boldsymbol{x}(k+1), R_{k+1}) - J^*(\boldsymbol{x}(k), R_{k+1})$ given in (69). Furthermore, consider Theorems 3, 4, and 5 with the definitions of upper bounds of the terms α_1 (70), α_2 (73), and α_3 (80), respectively. Then, with $\alpha_4 = 0$ (Theorem 6), the cost difference $\Delta J(k+1)$ based on the weighting factor R_{k+1} is limited by

$$J^f(\boldsymbol{x}(k+1), R_{k+1}) - J^*(\boldsymbol{x}(k), R_{k+1}) \leqslant -L(\boldsymbol{x}^*(k|k), u^*(k|k), d(k), R_{k+1}) + c_d \cdot \|\Delta d\| \tag{89}$$

The positive constant parameter c_d is given by

$$c_d = c_1(Q, N_u) + c_2(R_{k+1}, N_u, L_\varphi) + c_3(Q, N, N_u, L_\varphi, L_l, c_x) \tag{90}$$

and depends on the weighting factors Q and R_{k+1}, the parameters L_φ, L_l and c_x, the prediction horizon N, and the control horizon N_u.

In addition, for the cost $J^*(\boldsymbol{x}(k+1), R_{k+1})$ based on the optimal solution $\boldsymbol{u}^*(k+1)$ at $k+1$, the statement

$$J^*(\boldsymbol{x}(k+1), R_{k+1}) \leqslant J^f(\boldsymbol{x}(k+1), R_{k+1}) \tag{91}$$

holds if there is a $\hat{k}$ such that $R_{\hat{k}} = R_k \;\; \forall \;\; k \geqslant \hat{k}$. Thus, using (91) in (89), the difference of the optimal costs based on the weighting factor R_{k+1} is bounded by

$$J^*(\boldsymbol{x}(k+1), R_{k+1}) - J^*(\boldsymbol{x}(k), R_{k+1}) \leqslant -L(\boldsymbol{x}^*(k|k), u^*(k|k), d(k), R_{k+1}) + c_d \cdot \|\Delta d\| \tag{92}$$

under the condition that there is some $\hat{k}$ such that $R_{\hat{k}} = R_k \;\; \forall \;\; k \geqslant \hat{k}$. Hence, the bound of the difference between the optimal costs depends directly on the estimation error increment $\Delta d = d(k+1) - d(k)$. □

DOI: 10.1002/rnc

ACKNOWLEDGEMENT

This work was supported by the Spanish Ministry of Economy and Competitiveness (grant no. DPI2013-48243-C2-2-R).

REFERENCES

1. Rossiter JA. *Model-Based Predictive Control: A Practical Approach*. CRC Press: Boca Raton, 2003.
2. Camacho EF, Bordons C. *Model Predictive Control*. Springer: London, 2004.
3. Rawlings JB, Mayne DQ. *Model Predictive Control: Theory and Design*. Nob Hill Publishing: Madison, 2009.
4. Grüne L, Pannek J. *Nonlinear Model Predictive Control*. Springer: London, 2011.
5. Alamo T, Muñoz de la Peña D, Limon DM, Camacho EF. Constrained min-max predictive control: modifications of the objective function leading to polynomial complexity. *IEEE Transactions on Automatic Control* 2005; **50**(5): 710–714.
6. Ogunnaike BA. Nonlinear control of industrial processes. In *Nonlinear Predictive Control: Theory and Practice*, Kouvaritakis B, Cannon M (eds)., IEE Control Series, The Institution of Engineering and Technology: London, 2001; 205–221.
7. Seborg DE, Mellichamp DA, Edgar TF, Doyle FJ. *Process Dynamics and Control* (3rd edn). John Wiley & Sons: New York, 2011.
8. Biegler LT. Efficient Solution of Dynamic Optimization and NMPC Problems. In *Nonlinear Model Predictive Control*, Allgöwer F, Zheng A (eds)., Progress in Systems and Control Theory. Birkhäuser Verlag: Basel, 2000; 219–243.
9. Qin SJ, Badgwell TA. A survey of industrial model predictive control technology. *Control Engineering Practice* 2003; **11**(7):733–764.
10. Bauer M, Craig IK. Economic assessment of advanced process control—a survey and framework. *Journal of Process Control* 2008; **18**(1):2–18.
11. Maner BR, Doyle FJ, Ogunnaike BA, Pearson RK. Nonlinear model predictive control of a simulated multivariable polymerization reactor using second-order Volterra models. *Automatica* 1996; **32**(9):1285–1301.
12. Doyle FJ, Pearson RK, Ogunnaike BA. *Identification and Control Using Volterra Models*. Springer: London, 2001.
13. Li Y, Kashiwagi H. High-order Volterra model predictive control and its application to a nonlinear polymerisation process. *International Journal of Automation and Computing* 2005; **2**(2):208–214.
14. Haber R, Bars R, Schmitz U. *Predictive Control in Process Engineering—From the Basics to the Applications*. John Wiley & Sons: Weinheim, 2011.
15. Kumar D, Budman H. Robust nonlinear MPC based on Volterra series and polynomial chaos expansions. *Journal of Process Control* 2014; **24**(1):304–317.
16. Genceli H, Nikolaou M. Robust stability analysis of constrained l_1-norm model predictive control. *AIChE Journal* 1993; **39**(12):1954–1965.
17. Zheng A, Morari M. Stability of model predictive control with mixed constraints. *IEEE Transactions on Automatic Control* 1995; **40**(10):1818–1823.
18. de Nicolao G, Magni L, Scattolini R. Robustness of receding horizon control for nonlinear discrete-time systems. In *Robustness in Identification and Control*, Garulli A, Tesi A, Vicino A (eds)., Lecture Notes in Control and Information Sciences. Springer: London, 1999; 408–421.
19. Mayne DQ, Rawlings JB, Rao CV, Scokaert POM. Constrained model predictive control: stability and optimality. *Automatica* 2000; **36**(6):789–814.
20. Michalska H, Mayne DQ. Robust receding horizon control of constrained nonlinear systems. *IEEE Transactions on Automatic Control* 1993; **38**(11):1623–1633.
21. de Nicolao G, Magni L, Scattolini R. On the robustness of receding-horizon control with terminal constraints. *IEEE Transactions on Automatic Control* 1996; **41**(3):451–453.
22. Huang R, Harinath E, Biegler LT. Lyapunov stability of economically oriented NMPC for cyclic processes. *Journal of Process Control* 2011; **21**(4):501–509.
23. Diehl M, Amrit R, Rawlings JB. A Lyapunov function for economic optimizing model predictive control. *IEEE Transactions on Automatic Control* 2011; **56**(3):703–707.
24. Heidarinejad M, Liu J, Christofides PD. Economic model predictive control of nonlinear process systems using Lyapunov techniques. *AIChE Journal* 2012; **58**(3):855–870.
25. Boyd S, Chua LO. Fading memory and the problem of approximating nonlinear operators with Volterra series. *IEEE Transactions on Circuits and Systems* 1985; **32**(11):1150–1161.
26. Volterra V. *Theory of Functionals and of Integral and Integro-Differential Equations*. Dover Publications: New York, 1959.
27. Boyd S, Vandenberghe L. *Convex Optimization*. Cambridge University Press: Cambridge, 2006.
28. Mäkelä MM. Survey of bundle methods for nonsmooth optimization. *Optimization Methods and Software* 2002; **17**(1):1–29.

DOI: 10.1002/rnc

29. Gruber JK. Efficient and robust techniques for predictive control of nonlinear processes. *PhD Thesis*, Departamento de Ingeniería de Sistemas y Automática. Universidad de Sevilla, 2010. (Available from: https://www.educacion.es/teseo/imprimirFicheroTesis.do?fichero=17446).
30. Santos LO, Afonso P A FNA, Castro J AAM, Oliveira NMC, Biegler LT. On-line implementation of nonlinear MPC: an experimental case study. *Control Engineering Practice* 2001; **9**(8):847–857.
31. Lee JH, Lee KS, Kim WC. Model-based iterative learning control with a quadratic criterion for time-varying linear systems. *Automatica* 2000; **36**(5):641–657.
32. Gruber JK, Ramirez DR, Limon D, Alamo T. Computationally efficient nonlinear min-max model predictive control based on Volterra series models—application to a pilot plant. *Journal of Process Control* 2013; **23**(4):543–560.
33. Nowak RD, Veen BDV. Nonlinear system identification with pseudorandom multilevel excitation sequences. *Proceedings of the 1993 IEEE International Conference on Acoustics, Speech, and Signal Processing*, Vol. 4, Minneapolis, MN, 1993; 456–459.
34. Nowak RD, Veen BDV. Efficient methods for identification of Volterra filter models. *Signal Processing* 1994; **38**(3):417–428.

DOI: 10.1002/rnc