Computationally efficient nonlinear Min−Max Model Predictive Control based on Volterra series models—Application to a pilot plant

Article *in* Journal of Process Control · April 2013

DOI: 10.1016/j.jprocont.2013.01.007

Daniel R. Ramirez
University of Seville
65 PUBLICATIONS **950** CITATIONS
SEE PROFILE

D. Limon
Universidad de Sevilla
120 PUBLICATIONS **2,922** CITATIONS
SEE PROFILE

Teodoro Alamo
Universidad de Sevilla
254 PUBLICATIONS **5,386** CITATIONS
SEE PROFILE

Computationally efficient nonlinear Min–Max Model Predictive Control based on Volterra series models—Application to a pilot plant

J.K. Gruber [a,*], D.R. Ramirez [b], D. Limon [b], T. Alamo [b]

[a] Electrical Systems Unit, IMDEA Energy Institute, Avda. Ramón de la Sagra, 3, 28935 Móstoles, Madrid, Spain
[b] Departamento de Ingeniería de Sistemas y Automática, Escuela Superior de Ingenieros, Universidad de Sevilla, Camino de los Descubrimientos s/n, 41092 Sevilla, Spain

ABSTRACT

The mathematical model used in Min–Max MPC (MMMPC) to predict the future trajectory of the system explicitly considers disturbances and uncertainties . Based on the future trajectory, the control sequence is computed minimizing the worst case cost with respect to all possible trajectories of the disturbances and uncertainties . This approach leads to a more robust control performance but also complicates the practical implementation of MMMPC due to the high computational burden required to solve the opti- mization problem. This computational burden is even worse if a nonlinear prediction model is used. In fact, to the best of the authors ' knowledge , there have not yet been reported any applications of non - linear MMMPC to real processes . In this paper a nonlinear MMMPC strategy based on a second order Volterra series model is presented. The particular structure of the used prediction model allows to obtain an explicit formulation of the worst case cost and its computation in polynomial time. Real time appli- cations with typical prediction and control horizons are possible because of the reduced complexity of the proposed control strategy . Furthermore , input - to - state practical stability for the proposed control strategy is guaranteed under certain conditions . The MMMPC strategy is implemented and validated in experiments with a continuous stirred tank reactor whose temperature dynamics are approximated by a second order Volterra series model. The control performance of the proposed MMMPC strategy is illustrated by the obtained experimental results.

1. Introduction

In Model Predictive Control (MPC) [1–4], a wide range of different mathematical models can be used to predict the future evolution of the considered process. However, even with complex mathematical models it is difficult to capture the dynamics of a physical process. The possible model mismatch and external disturbances lead to a deficient prediction of the future evolution of the considered system and frequently to an insufficient control performance. In order to obtain a more robust control law, uncertain prediction models can be used within the MPC framework. The use of uncertain models to predict the future system trajectory considerably complicates the solution of the optimization problem and leads both to computational and theoretical difficulties. In the case of bounded uncertainties the resulting family of trajectories is also bounded. This bound represents the worst case with respect to the uncertainty and the minimization of the associated cost by a suitable choice of the input action results in a more robust control. The minimization of the mentioned worst case cost in order to compute the control action is known as Min–Max Model Predictive Control (MMMPC), initially proposed in [5].

In open-loop MMMPC the worst case of the predicted evolution is minimized without taking into account that the control signal is applied in a feedback manner [6,7]. The imposed constraints have to be satisfied for all possible trajectories of the evolution of the system, resulting in a conservative control performance. The complexity of the resulting optimization problem depends exponentially on the used prediction horizon and represents an NP-hard problem [8].

In closed-loop MMMPC, proposed in [9], the problem is minimized under explicit consideration of a feedback of the predicted system evolution [10] leading to a less conservative control law. The feedback approach results in an infinite dimensional optimization problem and is obviously far more complex than the one corresponding to the open-loop MMMPC. As a result of the high computational complexity of the optimization problem resulting both from open- and closed-loop MMMPC, the number of applications of MMMPC is very small, even

[*] Corresponding author. Tel.: +34 917371151; fax: +34 917371140.
E-mail addresses: jorn.gruber@imdea.org (J.K. Gruber), danirr@cartuja.us.es (D.R. Ramirez), limon@cartuja.us.es (D. Limon), alamo@cartuja.us.es (T. Alamo).

when there is evidence that the min–max approach can perform better than standard MPC in processes with uncertain dynamics [11–13]. Furthermore exists the semi-feedback approach based on open-loop predictions which adds some closed-loop behavior to the system. This approach can be used to achieve some desired property such as nominal stability or LQR optimality. It has to be mentioned that the open-loop and semi-feedback approach result in the same optimization problem in the case of non-autoregressive and stable prediction models. In fact, for Volterra prediction models, the uncertainty contribution to the system output does not grow indefinitely over time, i.e. it stabilizes to a certain value, as in a semi-feedback approach. Thus, in this work only open-loop MMMPC has been considered.

Volterra series models are used in a wide range of areas to model nonlinear processes including biomedical applications, acoustics, electronics, nonlinear filter design [14,15] and process control [16–19]. A great variety of different nonlinear dynamics, e.g. non-minimum-phase systems, can be approximated by this model type without the necessity of a deep understanding of the regarded system [16,20]. It has been shown in [21] that Volterra series models can be used to approximate arbitrarily well any stable system with fading memory characteristics. The main drawbacks of Volterra series models are the elevated number of parameters to describe the nonlinear dynamics of a process, requiring large data sets for identification purposes, and the impossibility to use Volterra series models to approximate unstable processes. However, the limitation to open-loop stable systems can be neutralized using a prestabilizing control.

In this work a Nonlinear Min–Max Model Predictive Control based on second order Volterra series models is presented. Using an explicit formulation of the worst case cost, the min–max optimization problem can be converted into a pure minimization problem. The low complexity of the optimization problem of the control strategy allows its use in real time applications with typical prediction and control horizons. For the presented MMMPC, input-to-state practical stability (ISpS) is guaranteed for sufficiently long finite prediction horizons. The MMMPC strategy is validated in experiments with a pilot plant that has been used before as a benchmark system for nonlinear and Min–Max Model Predictive Control strategies [13,18]. Furthermore, the results are compared to the ones obtained with a Nonlinear Model Predictive Control (NMPC) strategy based on the same mathematical model (disregarding the uncertainty of the model).

The paper is organized as follows: Section 2 defines the general problem description resulting from the chosen min–max approach in combination with a second order Volterra series model. The complexity of the min–max problem and how to solve it by means of an evaluation of all possible extreme values of the uncertainty are presented in Section 3. On the other hand, Section 4 shows a much more efficient way of computing the exact worst case cost for the considered MMMPC optimization problem. The input-to-state practical stability of the proposed MMMPC strategy is proven in Section 5. Section 6 describes the process used to validate the MMMPC strategy and Section 7 presents the obtained experimental results. Finally, in Section 8 the mayor conclusions are drawn.

2. Problem description

This work presents an MMMPC strategy for a Single Input Single Output (SISO) nonlinear system subject to input constraints. The nominal process model takes the form of a general second order non-autoregressive Volterra series model:

$$y(k) = h_0 + \sum_{i=1}^{N_t} h_1(i)u(k-i) + \sum_{i=1}^{N_t}\sum_{j=i}^{N_t} h_2(i,j)u(k-i)u(k-j) \tag{1}$$

where h_0 is a static coefficient (offset), $h_1(i)$ are the parameters for the linear terms and $h_2(i,j)$ are the parameters for the nonlinear terms of the model. The variables $y(k)$ and $u(k)$ denote the model output and input, respectively. The latter is subject to constraints, i.e. $u(k) \in U \triangleq [u_{min}, u_{max}]$. 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.[1] This type of models has been used to approximate fading memory systems [21]. Furthermore, a known estimation error (which can represent measurable disturbances) and a bounded additive uncertainty term (which accounts for modeling errors and non measurable disturbances) are considered. Thus, the uncertain prediction model will be:

$$y(k+j|k) = h_0 + \sum_{i=1}^{N_t} h_1(i)u(k+j-i|k) + \sum_{i=1}^{N_t}\sum_{l=i}^{N_t} h_2(i,\,l)u(k+j-i|k)u(k+j-l|k) + d(k) + \theta(k+j|k) \tag{2}$$

The term $d(k)$ represents the measured estimation error and $\theta(k+j|k)$, the additive uncertainty is bounded by the condition $|\theta(k+j|k)| \le \varepsilon\,\forall j$.

The Volterra series model (2) can be used as prediction model in a general cost function $J(\boldsymbol{\theta}, \boldsymbol{u})$ [3]:

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

where $y(k+j|k)$ is the output prediction for $k+j$ made at the sampling period k when the future values of the uncertainty are supposed to be given by the sequence $\boldsymbol{\theta} = [\theta(k+1|k), \ldots, \theta(k+N|k)]^T$ with $\theta(k|k) = 0$ (as any error in the output at time k has been already considered by the estimation error $d(k)$) and $|\theta(k+j|k)| \le \varepsilon\,\forall j$, and the future input sequence computed at k is given by the vector $\boldsymbol{u} \in \mathbb{R}^{N_u}$ defined by $\boldsymbol{u} = [u(k|k), \ldots, u(k+N_u-1|k)]^T$. Note that a control effort weighting factor λ is used. The variable $r(k)$ represents the setpoint for the output and $u_r(k)$ is the corresponding steady-state input for such setpoint value and it is defined by:

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

where $\varphi(\cdot)$ is a function that returns the necessary steady state nominal input (using the nominal model (1)) for a given steady state nominal output (i.e. given an equilibrium of the nominal model (u_{ss}, y_{ss}), then $u_{ss} = \varphi(y_{ss})$). Note that it is assumed that the setpoint $r(k)$ is chosen so that the resulting $u_r(k) \in U$ for any possible value of $d(k)$ and well defined. Also, $d(k)$ can be easily computed as the difference between the process output and the nominal model (1).

[1] If different truncation orders are used (e.g. N_1 and N_2) N_t should be chosen as $N_t = \max(N_1, N_2)$. If $N_1 > N_2$, i.e. $N_t = N_1$, the missing second order term parameters are defined as $h_2(i,j) = 0\,\forall i > N_2\,\forall j > N_2$. In the opposite case, i.e. $N_2 > N_1$ and therefore $N_t = N_2$, the linear term parameters are defined as $h_1(i) = 0\,\forall i > N_1$.

The upper limits N and N_u of the sums denote the considered prediction and control horizons, respectively. For stability reasons it is assumed that $N \geq N_t + N_u$.

The scheme of predictive control considered here is the Min–Max Model Predictive Control [3] in which the optimal sequence $\boldsymbol{u}^*$ is calculated solving a min–max problem. Considering only input constraints,[2] the optimization problem can be written as:

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

with

$$\begin{aligned} J^*(\boldsymbol{u}) = \quad & \max_{\boldsymbol{\theta}} \quad J(\boldsymbol{\theta}, \boldsymbol{u}) \\ & \text{s.t.} \quad |\theta(k+j|k)| \leq \varepsilon, \ i = 1, \ldots, N \end{aligned} \tag{6}$$

The solution of this problem is applied using a receding horizon strategy, as in all predictive control schemes [3].

3. Complexity of the min–max problem

Min–Max problems found in MPC strategies tend to suffer from a very high computational burden derived from its computational complexity, often of the NP hard class. In this section, the complexity of the min–max problem (5) is analyzed. Note that the complexity depends mainly on the worst case cost $J^*(\boldsymbol{u})$, which in turn depends on the properties of $J(\boldsymbol{u}, \boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$. Thus the complexity analysis should start with the main component of $J(\boldsymbol{u}, \boldsymbol{\theta})$, i.e. the predictions of the output of model (2) over the prediction horizon. With the prediction horizon N, the control horizon N_u and the truncation order N_t the output of the model (2) can be written in a similar form as the one presented in [16]:

$$\boldsymbol{y} = G\boldsymbol{u} + \mathbf{f}(\boldsymbol{u}) + \boldsymbol{c} + \boldsymbol{\theta} \tag{7}$$

$$\boldsymbol{c} = H\boldsymbol{u}_{pas} + \mathbf{g}(\boldsymbol{u}_{pas}) + \boldsymbol{d} \tag{8}$$

being $\boldsymbol{y} \in \mathbb{R}^N = [y(k+1|k), \ldots, y(k+N|k)]^T$ the predicted system output along the prediction horizon, and $\boldsymbol{d} \in \mathbb{R}^N$ the future estimation error vector defined as $\boldsymbol{d} = [d(k), \ldots, d(k)]^T$. The term $G\,\boldsymbol{u}$ with $G \in \mathbb{R}^{N \times N_u}$ represents the linear part depending on the future input sequence and the vector $\mathbf{f}(\boldsymbol{u}) \in \mathbb{R}^N$ contains the future–future and future–past cross terms. The term $H\ \boldsymbol{u}_{pas}$ with $H \in \mathbb{R}^{N \times N_t}$ represents the linear influence of the past input signals $\boldsymbol{u}_{pas} \in \mathbb{R}^{N_t}$ and the vector $\mathbf{g}(\boldsymbol{u}_{pas})$ contains the past–past cross terms, i.e. the nonlinear influence of the past inputs. For more details on the used models (7) and (8) see [16,18].

The prediction Eqs. (7) and (8) are used in the previously defined cost function (3), that can be rewritten as:

$$J(\boldsymbol{\theta}, \boldsymbol{u}) = \boldsymbol{u}^T M_{uu} \boldsymbol{u} + 2\boldsymbol{u}^T M_{uf} \mathbf{f}(\boldsymbol{u}) + \mathbf{f}(\boldsymbol{u})^T \mathbf{f}(\boldsymbol{u}) + \boldsymbol{\theta}^T \boldsymbol{\theta} + 2\boldsymbol{\theta}^T M_{\theta u} \boldsymbol{u} + 2\boldsymbol{\theta}^T \mathbf{f}(\boldsymbol{u}) + 2\boldsymbol{\theta}^T M_{\theta c} + 2\boldsymbol{u}^T M_{uc} + 2\mathbf{f}(\boldsymbol{u})^T M_{fc} + M_{cc} \tag{9}$$

with the matrices defined by:

$$M_{uf} = M_{\theta u}^T = G^T, \quad M_{uc} = G^T(\boldsymbol{c} - \boldsymbol{r}) - \lambda \boldsymbol{u}_r, \quad M_{\theta c} = \boldsymbol{c} - \boldsymbol{r}, \quad M_{fc} = \boldsymbol{c} - \boldsymbol{r}, \quad M_{uu} = G^T G + \lambda I_N, \quad M_{cc} = \boldsymbol{c}^T \boldsymbol{c} + \boldsymbol{r}^T \boldsymbol{r} - 2\boldsymbol{c}^T \boldsymbol{r} + \lambda \boldsymbol{u}_r^T \boldsymbol{u}_r \tag{10}$$

where $\boldsymbol{u}_r \in \mathbb{R}^N = [u_r(k), \ldots, u_r(k)]^T$ and $\boldsymbol{r} \in \mathbb{R}^N = [r(k), \ldots, r(k)]^T$. It can be seen that the cost function (9) is a fourth order function with respect to the input signal $\boldsymbol{u}$ due to the quadratic function $\mathbf{f}(\boldsymbol{u})$. Moreover, note that the prediction Eqs. (7) and (8) are affine in $\boldsymbol{\theta}$. This implies that the quadratic cost function is convex in $\boldsymbol{\theta}$, thus the solution of problem (6) is the maximization of a convex function over a convex set. This leads to the following property.

Property 1. *Due to the convexity of the cost function in $\boldsymbol{\theta}$, the solution of the maximization problem (6) is attained at least at one of the vertices of the hypercube $\boldsymbol{\Theta}$ defined as $\boldsymbol{\Theta} = \{\boldsymbol{\theta} \in \mathbb{R}^N : \|\boldsymbol{\theta}\|_\infty \leq \varepsilon\}$ (note that a similar result is also valid when the prediction model is linear [3]). Therefore the maximization problem (6) is equivalent to*

$$J^*(\boldsymbol{u}) = \max_{\boldsymbol{\theta} \in \text{vert}\{\boldsymbol{\Theta}\}} J(\boldsymbol{\theta}, \boldsymbol{u}) \tag{11}$$

where vert$\{\boldsymbol{\Theta}\}$ *is the set of vertices of* $\boldsymbol{\Theta}$.

Thus, in order to solve problem (11), all the vertices of $\boldsymbol{\Theta}$ must be explored. The mandatory evaluation of each one of the 2^N vertices of $\boldsymbol{\Theta}$ leads to an exponential complexity. Therefore the maximization problem (11), and as a consequence the min–max problem (5), are of the NP-hard class, and can only be solved in real time for small prediction horizons.

The strategy presented in this paper is directed to reduce the computational cost of problem (5). The special character of the model, without autoregressive term in (2), is exploited to reduce the computational burden of the maximization. With the new algorithm the exact worst case cost can be calculated with a complexity of $O(N^2)$ instead of $O(2^N)$.

[2] Output constraints can be easily considered but they result in quadratic constraints in the minimization problem due to the nonlinearity of the Volterra series model. These nonlinear constraints can be handled with advanced optimization algorithms such as SQP techniques. It is noteworthy that the results of Section 4 would be still valid, thus even in this case the problem will be computationally tractable.

4. Efficient calculation of the worst case cost

This section presents how to compute the exact worst case cost for the maximization problem given in (11) without evaluating all the possible extreme values of the uncertainty. Thus, the exact worst case cost can be determined with simple mathematical operations and allows an easy and computationally efficient implementation.

Using the cost function (9) based on a second order Volterra series prediction model in (11), the maximization problem for a given input sequence $\boldsymbol{u}$ can be expressed as:

$$J^*(\boldsymbol{u}) = J(\mathbf{0}, \boldsymbol{u}) + \max_{\boldsymbol{\theta} \in \text{vert}\{\boldsymbol{\Theta}\}} \boldsymbol{\theta}^T \boldsymbol{\theta} + 2\,\boldsymbol{\theta}^T \boldsymbol{q}(\boldsymbol{u}) \tag{12}$$

with

$$\begin{aligned} \boldsymbol{q}(\boldsymbol{u}) &= M_{\theta u}\boldsymbol{u} + \mathbf{f}(\boldsymbol{u}) + M_{\theta c} \\ J(\mathbf{0}, \boldsymbol{u}) &= \boldsymbol{u}^T M_{uu}\boldsymbol{u} + 2\boldsymbol{u}^T M_{uf}\mathbf{f}(\boldsymbol{u}) + 2\boldsymbol{u}^T M_{uc} + \mathbf{f}(\boldsymbol{u})^T\mathbf{f}(\boldsymbol{u}) + 2\mathbf{f}(\boldsymbol{u})^T M_{fc} + M_{cc} \end{aligned} \tag{13}$$

with $\boldsymbol{q}(\boldsymbol{u}) \in \mathbb{R}^N$ and $J(\mathbf{0},\boldsymbol{u})$ represent the nominal cost, i.e. the cost in the absence of uncertainties and disturbances. Then, the calculation of $J^*(\boldsymbol{u})$ requires the maximization of the sum of the terms $\boldsymbol{\theta}^T\boldsymbol{\theta}$ and $\boldsymbol{\theta}^T\,\boldsymbol{q}(\boldsymbol{u})$ with respect to the disturbance vector $\boldsymbol{\theta}$. Obviously, with $J(\mathbf{0},\boldsymbol{u})$ not being a function of $\boldsymbol{\theta}$, the nominal cost cannot be maximized (nor minimized) with respect to the uncertainty.

The term $\boldsymbol{\theta}^T\boldsymbol{\theta}$ can be written as:

$$\boldsymbol{\theta}^T\boldsymbol{\theta} = N\varepsilon^2 \quad \forall \boldsymbol{\theta} \in \text{vert}\{\boldsymbol{\Theta}\} \tag{14}$$

and represents a constant term for all possible vertices, i.e. $\boldsymbol{\theta}^T\boldsymbol{\theta}$ adopts its maximum value independently of the chosen vertex $\boldsymbol{\theta}$. Thus, the maximization problem (12) can be expressed as:

$$J^*(\boldsymbol{u}) = J(\mathbf{0}, \boldsymbol{u}) + N\varepsilon^2 + \max_{\boldsymbol{\theta} \in \text{vert}\{\boldsymbol{\Theta}\}} 2\boldsymbol{\theta}^T \boldsymbol{q}(\boldsymbol{u}) \tag{15}$$

Now, the optimization problem (15) is solved by maximizing $\boldsymbol{\theta}^T\,\boldsymbol{q}(\boldsymbol{u})$. This can be easily achieved when the elements of $\boldsymbol{\theta}$ have the same leading signs as the elements of $\boldsymbol{q}(\boldsymbol{u})$, i.e. the disturbance vector $\boldsymbol{\theta}$ that maximizes the term $\boldsymbol{\theta}^T\,\boldsymbol{q}(\boldsymbol{u})$ is $\boldsymbol{\theta} = \text{sgn}(\boldsymbol{q}(\boldsymbol{u}))\varepsilon$. The resulting maximum corresponds to the sum of absolute values of $\boldsymbol{q}(\boldsymbol{u})$:

$$\max_{\boldsymbol{\theta} \in \text{vert}\{\boldsymbol{\Theta}\}} \boldsymbol{\theta}^T \boldsymbol{q}(\boldsymbol{u}) = \text{sgn}(\boldsymbol{q}(\boldsymbol{u}))^T \varepsilon \boldsymbol{q}(\boldsymbol{u}) = \varepsilon\|\boldsymbol{q}(\boldsymbol{u})\|_1 \tag{16}$$

Then, being $J(\mathbf{0},\boldsymbol{u})$ a constant function with respect to the disturbance vector $\boldsymbol{\theta}$ and using (16) in (15), the initial maximization problem (12) becomes:

$$J^*(\boldsymbol{u}) = J(\mathbf{0}, \boldsymbol{u}) + N\varepsilon^2 + 2\varepsilon\|\boldsymbol{q}(\boldsymbol{u})\|_1 \tag{17}$$

Furthermore, since in the maximization problem (12) only the term (16) depends directly on the chosen $\boldsymbol{\theta}$, the vertex leading to the maximum value of $\boldsymbol{\theta}^T\,\boldsymbol{q}(\boldsymbol{u})$ is also the vertex which represents the worst case. Hence, the vertex $\boldsymbol{\theta}^*$ which results in the worst case is defined as:

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta} \in \text{vert}\{\boldsymbol{\Theta}\}} J(\boldsymbol{\theta}, \boldsymbol{u}) = \arg\max_{\boldsymbol{\theta} \in \text{vert}\{\boldsymbol{\Theta}\}} \boldsymbol{\theta}^T \boldsymbol{q}(\boldsymbol{u}) = \text{sgn}(\boldsymbol{q}(\boldsymbol{u}))\varepsilon \tag{18}$$

Note that (17) is the exact solution to the maximization problem and can be calculated easily by simple mathematical operations, i.e. the computation of the worst case cost does not require an evaluation of the 2^N vertices in $\boldsymbol{\Theta}$. The reduction of the complexity of the maximization to $O(N^2)$ is especially relevant for the implementation when using longer prediction horizons. It has to be mentioned that the exact and easy computation of the worst case cost $J^*(\boldsymbol{u})$ is a result of the non-autoregressive character of the used second order Volterra series model with additive uncertainty (2). The non-autoregressive character of the used model leads to the particular form of the quadratic uncertainty term in (12), which is constant and can be computed beforehand. This does not hold in the case of an autoregressive second order Volterra series model with additive uncertainty, and an approximation of the worst case cost like in [22] should be used instead of the exact solution of the maximization problem.

Remark 1. The solution to the maximization problem in (12) has been found by separately computing and later adding the individual maxima of two terms with respect to $\boldsymbol{\theta} \in \text{vert}\{\boldsymbol{\Theta}\}$, i.e. $\boldsymbol{\theta}^T\boldsymbol{\theta}$ and $\boldsymbol{\theta}^T\,\boldsymbol{q}(\boldsymbol{u})$. But in general the sum of the individual maxima is not equal to the maximum of the sum of the terms. In fact, the sum of the maxima of these two terms and the maximum of the sum of them satisfies the following inequality:

$$\max_{\boldsymbol{\theta} \in \text{vert}\{\boldsymbol{\Theta}\}} \boldsymbol{\theta}^T\boldsymbol{\theta} + 2\boldsymbol{\theta}^T\boldsymbol{q}(\boldsymbol{u}) \leq \max_{\boldsymbol{\theta} \in \text{vert}\{\boldsymbol{\Theta}\}} \boldsymbol{\theta}^T\boldsymbol{\theta} + \max_{\boldsymbol{\theta} \in \text{vert}\{\boldsymbol{\Theta}\}} 2\boldsymbol{\theta}^T\boldsymbol{q}(\boldsymbol{u}) \tag{19}$$

for all possible trajectories $\boldsymbol{\theta} \in \text{vert}\{\boldsymbol{\Theta}\}$. Note that in (19) the equality holds if and only if both right-side terms reach their maximum for the same vector $\boldsymbol{\theta}$. This is the case in problem (12) because the constant term $\boldsymbol{\theta}^T\boldsymbol{\theta} = N\varepsilon^2$ adopts its maximum value for every $\boldsymbol{\theta} \in \text{vert}\{\boldsymbol{\Theta}\}$. Hence, the disturbance vector $\boldsymbol{\theta}^*$ which maximizes the term $\boldsymbol{\theta}^T\,\boldsymbol{q}(\boldsymbol{u})$ also maximizes the term $\boldsymbol{\theta}^T\boldsymbol{\theta}$. As a consequence, the equality in (19) holds for the optimization problem (12), and the approach followed here to find the solution of the maximization in (12) is correct.

4.1. Control strategy using the exact worst case cost

The considered MMMPC problem (5) can be solved easily with the explicit expression of the worst case cost $J^*(\boldsymbol{u})$ (17). In fact, using the explicit solution, the original min–max optimization problem has been reduced to a minimization problem similar to the ones of NMPC strategies.

The variable $\boldsymbol{q}(\boldsymbol{u})$ in the exact solution (17) can be rewritten with the definitions (10) and (13) as $G\,\boldsymbol{u}+\mathbf{f}(\boldsymbol{u})+\boldsymbol{c}-\boldsymbol{r}$. With the exact solution of the worst case cost the optimization problem (5) used to compute an input sequence can be rewritten as:

$$\begin{aligned}\boldsymbol{u}^* = arg \quad & \min_{\boldsymbol{u}} \quad J(\mathbf{0},\boldsymbol{u})+N\varepsilon^2+2\varepsilon\|G\boldsymbol{u}+\mathbf{f}(\boldsymbol{u})+\boldsymbol{c}-\boldsymbol{r}\|_1 \\ & \text{s.t.} \quad u(k+i|k)\in U \\ & \qquad\;\; u(k+i|k)=u_r(k),\ i=N_u,\ldots,N-1\end{aligned} \tag{20}$$

Note that the optimization is subject to linear constraints. The calculated input sequence $\boldsymbol{u}^*$ can then be used in a receding control strategy where in every sampling period only the first element of $\boldsymbol{u}^*$ is applied to the controlled system.

Note that the use of an explicit solution of the maximization problem reduces considerably the computational complexity of the control strategy with respect to strategies using a vertex search approach or an upper bound of the worst case cost. Furthermore, the term $N\varepsilon^2$ in (20) does not depend on $\boldsymbol{u}$ and can be removed from the minimization problem. The last term in (20) consists only of a summation of absolute values and of simple matrix and vector operations. Therefore, the optimization problem considering an uncertainty or disturbance has a computational complexity similar to the optimization of the nominal model. Finally, the calculation of the input sequence can be carried out solving the optimization problem with nonlinear programming methods such as sequential quadratic programming (SQP).

5. Robust stability

The proposed MMMPC controller ensures the robust stability of the controlled plant. This section proves the robust stability, defined as Input-to-State practical Stability (ISpS) with regard to the uncertainty [23], for the proposed MMMPC strategy.

5.1. Optimization problem in state-space representation

The prediction model based on a second order Volterra series model considering an estimation error and a future uncertainty can be expressed in a state-space form by:

$$\begin{aligned}\boldsymbol{x}(k+i+1|k) &= A\boldsymbol{x}(k+i|k)+Bu(k+i|k) \\ y(k+i|k) &= l(\boldsymbol{x}(k+i|k))+d(k)+\theta(k+i|k)\end{aligned} \tag{21}$$

where $\boldsymbol{x}(k)\in\mathbb{R}^{N_t}\ \forall k$, $\boldsymbol{x}(k|k)=\boldsymbol{x}(k)$ and all the assumptions made for model (2) also hold. With $d(k)=y(k)-l(\boldsymbol{x}(k))$ the state-space model includes output feedback and considers with $\theta(k+i|k)$ the influence of disturbances. Note that the evolution of the system state is described by a linear model, and that the nonlinearity affects only the system output by means of the function $l(\cdot)$. The state variables (see Appendix A for the detailed model transformation) are defined as:

$$x_i(k)=u(k-i) \quad \text{for} \quad i=1,\ldots,N_t \tag{22}$$

The mapping of the past input values into the state vector is a common approach in many mathematical models [3]. Note that the mapping is applied only to the past input values whereas the current control signal is maintained as the input variable $u(k)$. It has to be underlined that the change from the original second order Volterra series model (2) to the model in state-space representation (21) was done in order to use the theory of Lyapunov in the proof of robust stability.

Furthermore, with respect to the nominal model, the output prediction considering the current estimation error and future disturbance is defined by:

$$y(k+i|k)=\tilde{y}(k+i|k)+d(k)+\theta(k+i|k) \tag{23}$$

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

$$\tilde{y}(k+i|k)=l(\boldsymbol{x}(k+i|k)) \tag{24}$$

Then, with the model considering an estimation error and an uncertainty, the optimization problem of the MMMPC strategy (5) can be rewritten in a general manner in state-space as:

$$\begin{aligned}\boldsymbol{u}^* = arg \quad & \min_{\boldsymbol{u}} \quad J^*(\boldsymbol{u}) \\ & \text{s.t.} \quad u(k+i|k)\in U, \qquad i=0,\ldots,N_u-1 \\ & \qquad\;\; u(k+i|k)=u_r(k), \quad i=N_u,\ldots,N-1 \\ & \qquad\;\; \boldsymbol{x}(k+i|k)\in\boldsymbol{X}, \qquad i=1,\ldots,N\end{aligned} \tag{25}$$

where $\boldsymbol{X}=U\times U\times\cdots\times U\subseteq\mathbb{R}^N$ is the set of admissible states and

$$\begin{aligned}J^*(\boldsymbol{u}) = \quad & \max_{\boldsymbol{\theta}} \quad J(\boldsymbol{\theta},\boldsymbol{u}) \\ & \text{s.t.} \quad \theta(k+i|k)\in\Theta,\ i=1,\ldots,N\end{aligned} \tag{26}$$

Note that constraint $\boldsymbol{x}(k+i|k)\in\boldsymbol{X}$ is in fact redundant because of the constraints on $u(k+i|k)$, but it has been included in problem (25) to follow the standard formulations for stability in MPC (see [4] for a general treatment on this topic).

Taking into account the future input sequence $\boldsymbol{u}$, the steady-state input signal $u_r(k)$ and the sequence of the uncertainty $\boldsymbol{\theta}(k)$ as well as the current estimation error $d(k)$, the cost function $J(\cdot,\cdot)$ used in (26) is defined as:

$$J(\boldsymbol{\theta},\boldsymbol{u})=\sum_{i=0}^{N_u-1}L(\boldsymbol{x}(k+i|k),u(k+i|k),d(k),\theta(k+i|k))+\sum_{i=N_u}^{N}L_h(\boldsymbol{x}(k+i|k)) \tag{27}$$

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

$$\begin{aligned} &L(\boldsymbol{x}(k+i|k), u(k+i|k), d(k), \theta(k+i|k)) = \|l(\boldsymbol{x}(k+i|k)) + d(k) + \theta(k+i|k) - r(k)\|_Q^2 + \|u(k+i|k) - u_r(k)\|_R^2 \\ &L_h(\boldsymbol{x}(k+i|k)) = \|l(\boldsymbol{x}(k+i|k)) + d(k) + \theta(k+i|k) - r(k)\|_Q^2 \end{aligned} \tag{28}$$

where l is a nonlinear function defined in (A.3) in Appendix A and $r(k)$ denotes the desired reference for the system output.

5.2. Feasibility of the shifted solution

Consider the sequence:

$$\boldsymbol{u}^*(k) = [u^*(k|k), u^*(k+1|k), \ldots, u^*(k+N_u-1|k)]^T \tag{29}$$

being at k the optimal solution for the MMMPC problem (25) with the associated worst case optimal cost $J^*(\boldsymbol{u}^*(k))$. Furthermore, consider the shifted solution $\boldsymbol{u}^f(k+1)$ for $k+1$:

$$\boldsymbol{u}^f(k+1) = [u^f(k+1|k+1), u^f(k+2|k+1), \ldots, u^f(k+N_u|k+1)]^T \tag{30}$$

where the elements can be defined by means of the optimal solution in k and the steady-state input signal for $k+1$:

$$u^f(k+i|k+1) = \begin{cases} u^*(k+i|k) & \text{for} \quad i = 1, \ldots, N_u - 1 \\ u_r(k+1) & \text{for} \quad i = N_u \end{cases} \tag{31}$$

Note that the first $N_u - 1$ components of $u^f(k+i|k+1)$ are feasible as they were computed at k as the optimal solution of problem (25). On the other hand, $u_r(k+1)$ is by definition feasible, thus it can be concluded that the shifted sequence $\boldsymbol{u}^f(k+1)$ is a feasible solution to the optimization problem (25) at time $k+1$.

5.3. Convergence

Consider the cost $J^*(\boldsymbol{x}(k))$ at k based on the optimal solution $\boldsymbol{u}^*(k)$ minimizing the problem (25). Furthermore consider the cost $J^*(\boldsymbol{x}(k+1))$ at $k+1$. Convergence can be guaranteed if the calculated cost for $k+1$ is monotonically decreasing with respect to the cost for k.

With the general definition of the cost function (27), the optimal cost $J^*(\boldsymbol{x}(k))$ at k is given by:

$$J^*(\boldsymbol{x}(k)) = \sum_{i=0}^{N_u-1} L(\boldsymbol{x}^*(k+i|k), u^*(k+i|k), d(k), \theta^*(k+i|k)) + \sum_{i=N_u}^{N} L_h(\boldsymbol{x}^*(k+i|k)) \tag{32}$$

where $\theta^*(k+i|k)$ is the ith component of $\boldsymbol{\theta}^*$ defined as in (18) for $\boldsymbol{u}^*(k)$.

The following theorem characterizes the cost difference between the optimal cost $J^*(\boldsymbol{x}(k))$ at k and that cost at $k+1$. This theorem will be used to show the convergence of the proposed control law under certain conditions.

Theorem 1. *Consider the optimal solution $\boldsymbol{u}^*(k)$ at k which minimizes the optimization problem (25) and leads to the cost $J^*(\boldsymbol{x}(k))$. Furthermore, consider the cost $J^*(\boldsymbol{x}(k+1))$ at $k+1$. Taking into account that, by definition, $\theta^*(k|k) = 0$, the cost difference $\Delta J^*(k+1) = J^*(\boldsymbol{x}(k+1)) - J^*(\boldsymbol{x}(k))$ is bounded by:*

$$J^*(\boldsymbol{x}(k+1)) - J^*(\boldsymbol{x}(k)) \leq -L(\boldsymbol{x}^*(k|k), u^*(k|k), d(k), 0) + c_d \cdot \|\Delta d\| + c_\varepsilon \cdot \varepsilon \tag{33}$$

where c_d, and c_ε are positive and constant parameters.

Proof. See Appendix B. □

Then, with the bound (33) of the cost difference $\Delta J^*(k+1)$, the terms $c_d \cdot \| \Delta d \| + c_\varepsilon \cdot \varepsilon > 0$ and $-L(\boldsymbol{x}^*(k|k), u^*(k|k), d(k), 0) \leq 0$ ensure that the cost based on the feasible solution cost will decrease as long as the stage cost satisfies $L(\boldsymbol{x}^*(k|k), u^*(k|k), d(k), 0) > c_d \cdot \| \Delta d \| + c_\varepsilon \cdot \varepsilon$. Hence, the system is steered into the set:

$$\Psi_d = \{\boldsymbol{x}^*(k|k) : L(\boldsymbol{x}^*(k|k), u^*(k|k), d(k), 0) \leq c_d \cdot \|\Delta d\| + c_\varepsilon \cdot \varepsilon\} \tag{34}$$

from any arbitrary $\boldsymbol{x}^*(k|k)$. Nevertheless, as shown in the following paragraphs, if the system state evolves out of Ψ_d, the system will be kept in another set from which it will evolve back to the set Ψ_d. For any $\boldsymbol{x}^*(k|k)$, the stage cost always satisfies $-L(\boldsymbol{x}^*(k|k), u^*(k|k), d(k), \theta(k|k)) \leq 0$, hence (33) can be written in the form:

$$J^*(\boldsymbol{x}(k+1)) \leq J^*(\boldsymbol{x}(k)) + c_d \cdot \|\Delta d\| + c_\varepsilon \cdot \varepsilon \tag{35}$$

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

$$J^*(\boldsymbol{x}(k)) + c_d \cdot \|\Delta d\| + c_\varepsilon \cdot \varepsilon \leq \max_{\boldsymbol{x} \in \Psi_d} J^*(\boldsymbol{x}) + c_d \cdot \|\Delta d\| + c_\varepsilon \cdot \varepsilon = \beta_d \tag{36}$$

holds. Now, from (35) and (36) follows that:

$$J^*(\boldsymbol{x}(k+1)) \leq \beta_d, \quad \forall \boldsymbol{x}(k) \in \Psi_d \tag{37}$$

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

$$\Psi_\beta = \{\boldsymbol{x} : J^*(\boldsymbol{x}) \leq \beta_d\} \tag{38}$$

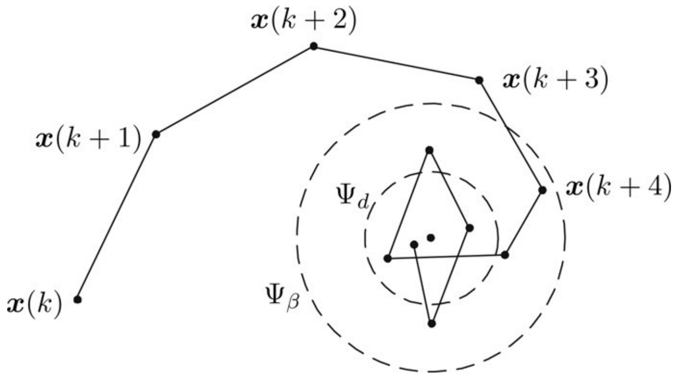

Fig. 1. Evolution of the system state $\boldsymbol{x}(\cdot)$ and the sets Ψ_d and Ψ_β.

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 (see Fig. 1). As a consequence, the state is ultimately bounded and the system is stabilized using the feasible solution. Hence, the MMMPC strategy based on a second order Volterra series model is input-to-state stable and maintains the system inside the set Ψ_β. It is clear from (36) that Δd and ε determine the value of β_d and consequently the size of the set Ψ_β. The value of β_d can be considered as a conservative estimation used to define the set Ψ_β to which the system will evolve in closed loop operation.

6. Process description

A real process represented by a pilot plant has been chosen for the application of the proposed MMMPC strategy. The process has been studied previously by several authors [18,13,24] and has been used as a benchmark for control purposes [25].

6.1. Laboratory process

The pilot plant (see Fig. 2) is used to emulate exothermic chemical reactions based on temperature changes as done in [26]. The main elements of the pilot plant are the reactor, the heat exchanger, the cooling jacket, the valve to manipulate the flow rate through the cooling jacket and the electrical resistance. The plant structure with the mentioned main elements is given in the schematic diagram in Fig. 3.

The cooling jacket is used to reduce the temperature of the reactor content. The heat dissipation can be regulated by the valve v_8 manipulating the flow rate F_j through the cooling jacket. The cooling fluid, water, circulating 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 constant the tank has an auxiliary cooler controlled by a thermostat which maintains the temperature T_{T2} near to a desired value in an interval of approximately 1°.

The reactant is supplied to the reactor by the feed $F_{f,in}$ to keep the chemical reaction active. Before entering the reactor, the feed passes through a heat exchanger in order to reduce the temperature difference between the feed and the reactor content. The outflow $F_{f,out}$ is used to keep the volume of the reactor content constant. As a consequence, as feed and outflow have the same flow rate and nearly the same temperature, the two flows hardly provoke temperature changes in the interior of the reactor.

To emulate exothermic reactions, the pilot plant possesses an electrical resistance in order to supply thermal energy to the reactor content. The energy to be supplied by the 14.4 kW electrical resistance is calculated with a nonlinear mathematical model of the reaction. The use of a resistance has the advantage that no chemical reaction takes place in the reactor, instead the reaction is emulated on basis of temperature changes, as done in [26].

6.2. Mathematical model

Although it is not necessary to have a mathematical model for the design of the proposed MMMPC, this section shows the process model to emphasize its nonlinear character. The mathematical model also justifies the way to emulate the heat generated by the chemical reaction with the aid of the resistance.

The emulated chemical reaction, representing a refinement process, was used previously in [13,18,27]. Considering identical flow rates for the feed and the outflow, i.e. $F_f = F_{f,in} = F_{f,out}$, the reactor volume V and the mass M are constant. The temperature changes of the reactor content can be defined as:

$$\frac{\mathrm{d}T}{\mathrm{d}t} = -\frac{F_j}{V}(T_{j,out} - T_{j,in}) + \frac{(-\Delta H)\cdot V}{MC_p} k_0 e^{-E/(RT)} C_A^2 \tag{39}$$

where the first term considers the heat dissipation by the cooling jacket and the second term denotes the generated heat by the exothermic chemical reaction. The variables F_j, $T_{j,in}$ and $T_{j,out}$ represent the flow rate through the cooling jacket and the temperature of the cooling fluid entering and leaving the cooling jacket, respectively. C_A is the concentration of the reactant in the reactor content. It has been assumed that the feed neither supplies nor removes caloric energy from the reactor as the feed passes through a heat exchanger and enters the reactor nearly with the temperature of the reactor content. For the heat exchange in the cooling jacket the empirical model:

$$F_j \cdot (T_{j,out} - T_{j,in}) = \frac{T-\alpha}{\beta}(1 - e^{-\gamma F_j}) \tag{40}$$

with $\alpha = 292.19$ K, $\beta = 14.94$ s/l and $\gamma = 13.18$ s/l was used.

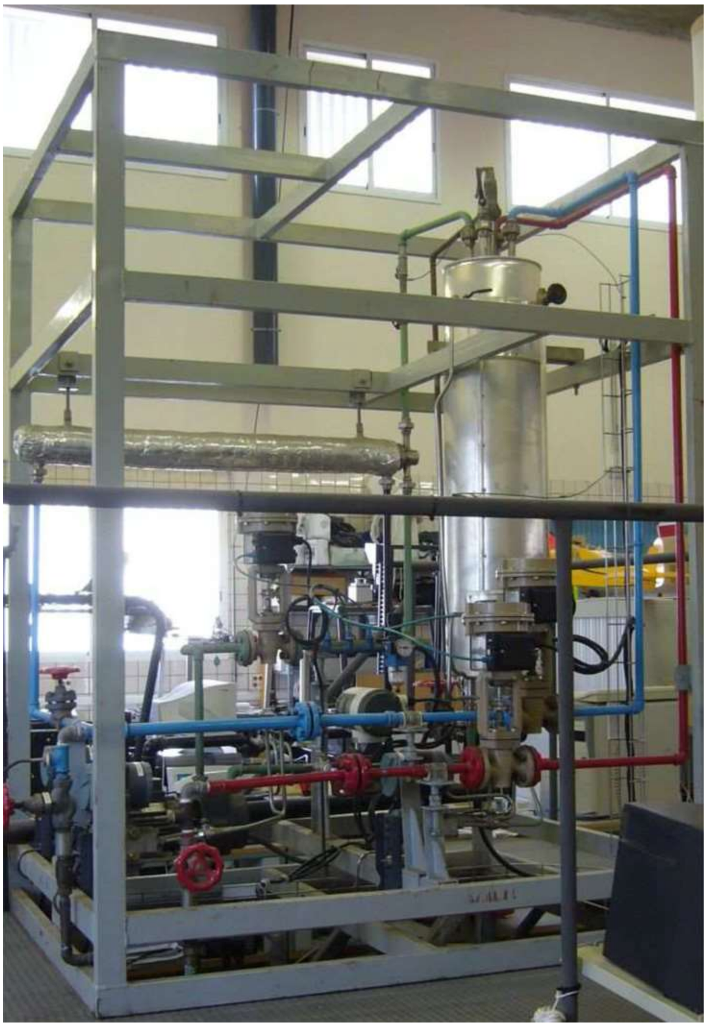

Fig. 2. Pilot plant used to apply the proposed MMMPC based on a Volterra series model.

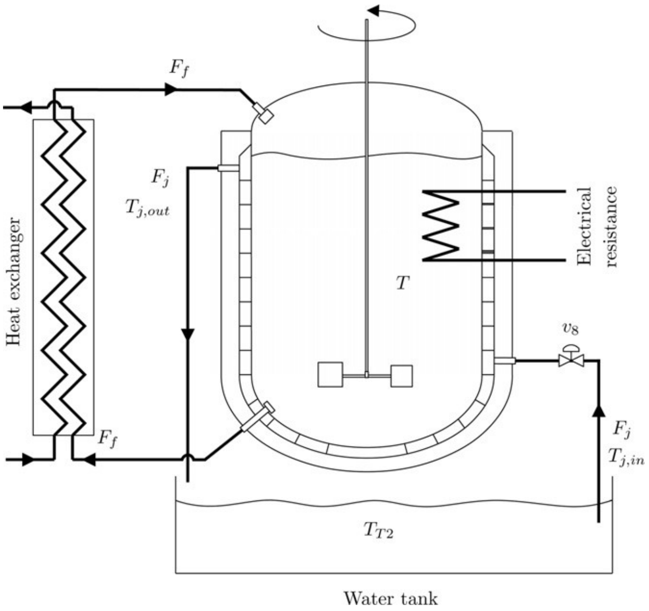

Fig. 3. Diagram of the pilot plant with its main elements.

Table 1
Model parameters and constant variables of the chemical reaction.

Parameter	Value	Unit
k_0	1.2650×10^{17}	l/(mol s)
C_p	4.18	kJ/(K kg)
ΔH	-105.57	kJ/mol
E/R	13,550	K

Variable	Value	Unit
V	25	l
M	25	kg
$C_{A,in}$	1.2	mol/l
F_f	0.0	l/s
$T_{j,in}$	291.15	K

The reactant concentration C_A in the plant reactor is calculated by:

$$\frac{\mathrm{d}C_A}{\mathrm{d}t} = \frac{F_f}{V}(C_{A,in} - C_A) - k_0 e^{-E/(RT)} C_A^2 \tag{41}$$

where the first term represents changes in the reactant concentration due to the feed and the outflow. The second term considers the reduction of the concentration as a result of the reactant consumption by the chemical reaction. $C_{A,in}$ denotes the reactant concentration in the feed. The model parameters and the variables used with constant values are shown in Table 1.

The electric resistance of the pilot plant is used during the emulation to supply the thermal energy which would be generated by the exothermic reaction if the chemical process would take place in the reactor. The temperature gradient based only on the heat generated by the exothermic process corresponds to the second right-hand term in (39). Then, the necessary power to emulate the chemical reaction, i.e. the temperature gradient, in the reactor is defined by:

$$P = C_p M \frac{(-\Delta H) \cdot V}{M C_p} k_0 e^{-E/(RT)} C_A^2 \tag{42}$$

where C_p denotes the specific heat capacity of the reactor content and M is the corresponding mass. Hence, the power P can be computed with the concentration C_A calculated in (41) and the measured temperature T. Finally, the necessary energy is supplied to the plant by adjusting the duty cycle of the electric resistance in accordance with the power (42).

As can be easily seen from the model Eqs. (39) and (41) the chemical reaction possesses nonlinearities in the dynamics of the temperature and the concentration. For further details on the model parameters see [19].

6.3. Description of the control system

The sensors and actuators of the used pilot plant are connected to a Schneider M340 programmable automation controller (PAC). The configuration, programming and debugging of the M340 programmable automation controller has been carried out with the Unity Pro software package. In the pilot plant, the M340 PAC is used to perform basic data acquisition tasks, to supervise the plant conditions (temperature and levels within safe limits), and also to implement the start-up, shut-down and emergency sequences. The M340 communicates via Ethernet with a personal computer that runs a Vijeo Citect SCADA (supervisory control and data acquisition) and Matlab. The proposed MMMPC strategy has been implemented directly in Matlab/Simulink and the communication with the SCADA is done using the OPC protocol (OLE for Process Control). Hence, both the SCADA and the controller implemented in Matlab/Simulink run on the same personal computer based on a Pentium 4 processor with 3GHz using Windows XP as operating system.

7. Experimental results

In this section the proposed nonlinear MMMPC strategy based on Volterra series models (see Section 4) is applied to the refinement process described in Section 6.

7.1. Identification of the prediction model

A simple input sequence (pseudo random multilevel sequence [14,28]) with three different levels for the recirculation valve has been applied to the pilot plant in order to collect suitable input–output data for the parameter identification of the nonlinear Volterra series model. The periods of the sequence were 100 min, long enough to observe the reaction of the pilot plant. With the input–output data a second order Volterra series model has been identified. In order to reduce the number of parameters to be identified, a diagonal Volterra series model ($h_2(i,j) \neq 0 \, \forall \, i = j$ and $h_2(i,j) = 0 \, \forall \, i \neq j$) has been used. Being linear in the parameters, the model has been identified with the least squares method. With a sampling time of $t_s = 60$ s and a delay of 1 sampling period the truncation orders were $N_1 = 60$ for the linear part and $N_2 = 30$ for the nonlinear part of the model.[3] The chosen sampling time represents a trade-off between the number of model parameters to be identified and the quality of fit of the resulting model. A shorter sampling time would lead to a considerably higher number of parameters and the necessity of larger data sets for the identification. In the case of a higher sampling time, the mathematical model would capture the dynamic behavior of the considered system with less precision.

[3] In order to use one unique truncation order $N_t = 60$, the parameters $h_2(i,j)$ have been defined as $h_2(i,j) = 0 \, \forall \, i,j = N_2 + 1, \ldots, N_t$.

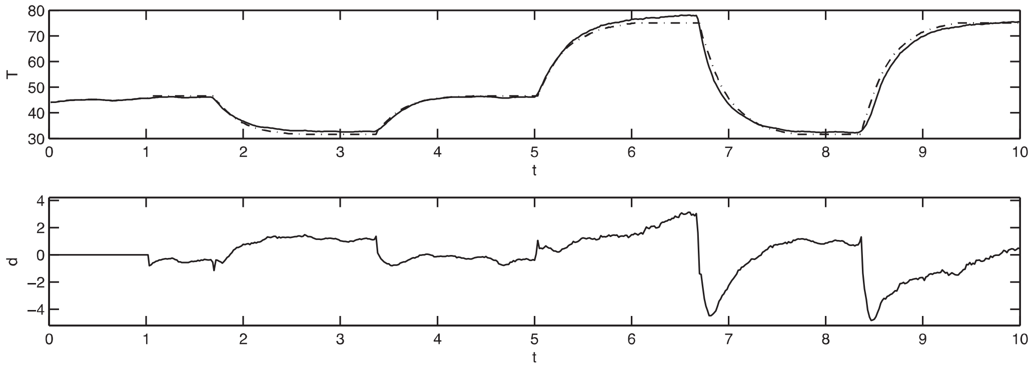

Fig. 4. Results of the model identification with a comparison between model output and measured values (top) and the associated estimation error $d(k)$ (bottom).

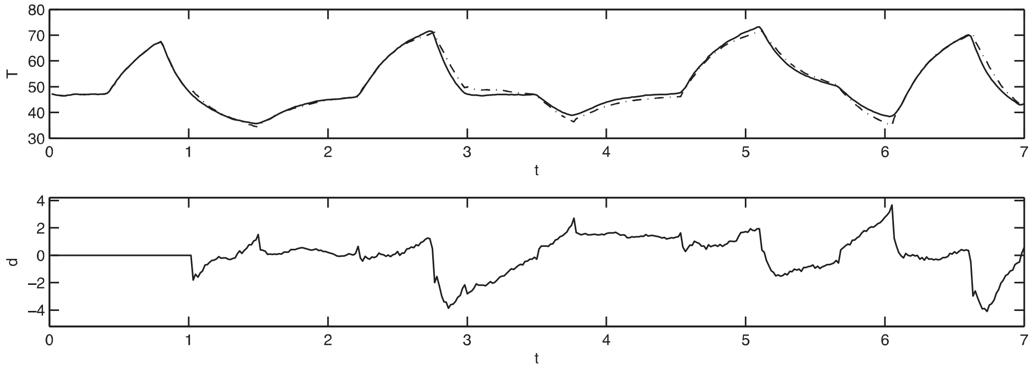

Fig. 5. Results of the model validation with a comparison between model output and measured values (top) and the associated estimation error $d(k)$ (bottom).

A comparison between the data set used for identification purposes and the output of the second order Volterra series model as well as the obtained estimation error, i.e. the difference between the measured temperature and the model output, are shown in Fig. 4. It can be observed that the model output shows an acceptable fit with the measured temperature and that the estimation error adopts its maximum values during the large temperature changes. A validation of the identified second order Volterra series model has been carried out with a second experimental data set. The results of the validation (see Fig. 5) are similar to the ones obtained during the identification and confirm the suitability of the mathematical model. Finally, the identified parameters for the linear and nonlinear terms of the second order Volterra series are given in Fig. 6. The slightly irregular shape of the curves is a result of the used least squares parameter estimation from experimental data. It can be observed that both parameter curves tend to zero and indicate the fading memory behavior of the system.

Note that diagonal Volterra series models are a special case of the Volterra series models. Therefore, the calculation of the worst case cost (see Section 4) is the same for diagonal and non diagonal Volterra series models.

7.2. Experimental results of the controller

The identified Volterra series model has been used as a prediction model in the proposed MMMPC strategy. The Matlab implementation of the control strategy uses the algorithm from Section 4 for the maximization and the minimization is carried out with a Matlab built-in function for sequential quadratic programming (`fmincon`). For the implementation of the control strategy a sampling time of $t_s = 60$ s has been used. This sampling time is 5–10 times faster than the time constant of the fastest dynamics of the considered system and corresponds

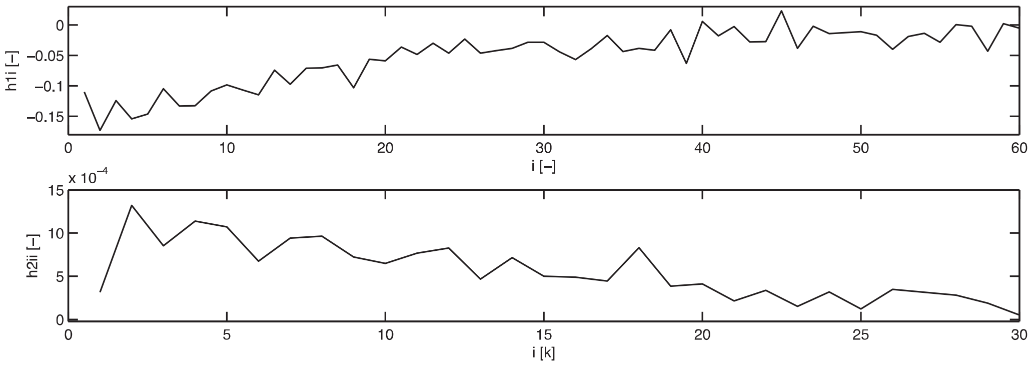

Fig. 6. Volterra series model parameters identified from input–output data by means of the least squares estimation method.

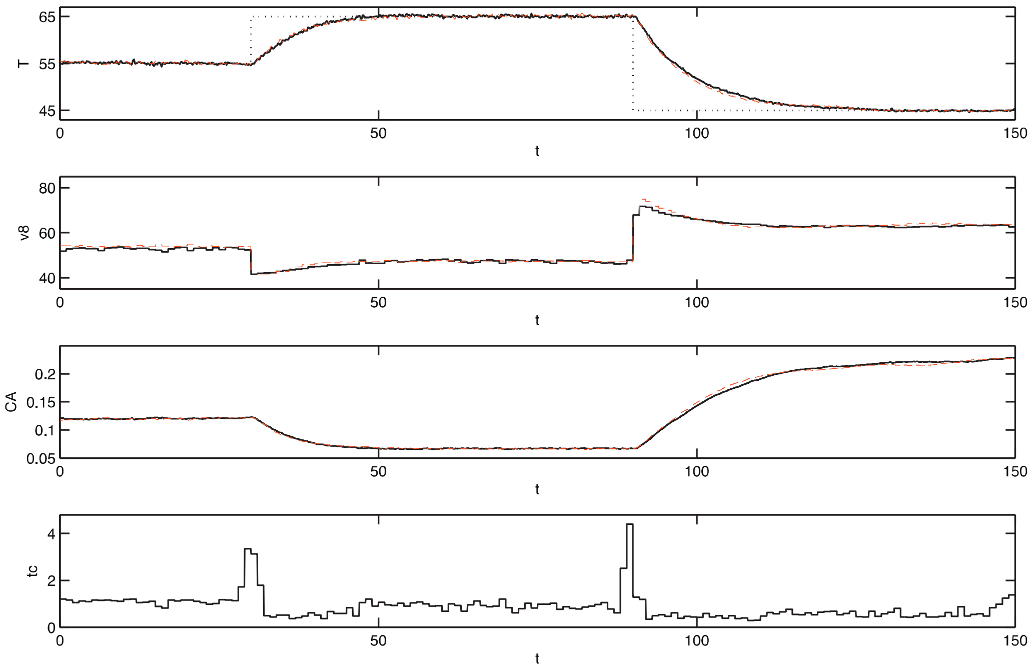

Fig. 7. Reference tracking experiment using the proposed MMMPC strategy (solid line) and the iterative NMPC approach (dashed line). From top to bottom: tank temperature (T), valve opening (v_8), reagent concentration (C_A) and computation time (t_c).

to the sampling used during the identification of the Volterra series model (see Section 7.1). In order to satisfy the necessary condition $N \geq N_t + N_u$ for input-to-state practical stability, a prediction horizon of $N = 80$ and a control horizon of $N_u = 15$ were used. The control effort will be weighted by the factor $\lambda = 5$. The parameter ε which bounds the additive uncertainty considered in the Volterra series prediction model has been identified in experiments. A value of $\varepsilon = 5$ was used to include the errors shown in Figs. 4 and 5. Furthermore, the constraints:

$$\begin{aligned} &5 \leq u(k+i|k) \leq 100, \quad i = 0, \ldots, 14 \\ &u(k+i|k) = u_r(k), \quad i = 15, \ldots, 79 \end{aligned} \tag{43}$$

were considered in the computation of the input sequence. The constraints $\theta(k+i|k) \leq \varepsilon$ for $i = 1, \ldots, 80$ have been considered implicitly in the computation of the worst case cost.

Finally, to compare the results of the proposed MMMPC strategy, the experiments were also carried out with an iterative NMPC strategy [16] based on the same second order Volterra series model. To allow a direct comparison of the results, the NMPC was adjusted with the same parameters (prediction and control horizons, weighting factor, constraints) as the MMMPC.

In the first experiment carried out with the pilot plant, the setpoint tracking quality of the proposed MMMPC strategy with guaranteed stability was validated. During the experiment (see Fig. 7) the setpoint was changed in $t = 30$ min from 55 °C to 65 °C and set in $t = 90$ min to 45 °C. The setpoint changes lead to a very fast reaction of the control strategy which stabilizes the system in the given reference without any overshoot. After the stabilization of the system, neither the temperature nor the input signal shows oscillations. In steady state, only small modifications in the input signal can be observed, necessary to maintain the system output in the setpoint.

In the second experiment (see Fig. 8) the disturbance rejection capability of the proposed control strategy was validated by means of an error in the activation energy E of the underlying exothermic chemical reaction. The introduced error, increasing the parameter E by 3% of the nominal value, was held constant during the entire experiment while two setpoint changes were applied to the system. After the application of the setpoint changes, the control strategy reacts rapidly and stabilizes the system in the given reference. In spite of the model mismatch (due to the error in the underlying chemical reaction model) only a small overshoot can be observed, approximately 0.6 °C and −0.7 °C after the first and the second change, respectively. The influence of the modified parameter can be seen comparing the results with the ones shown in Fig. 7. As a result of the error in the parameter E, the values of the valve v_8 and the concentration C_A have changed considerably, especially at the end of the experiment. Due to these fundamental changes in the chemical reaction, the obtained results represent a good control performance and underline the robustness of the proposed MMMPC strategy.

In the third experiment (see Fig. 9) with the proposed MMMPC strategy an additive disturbance in the system input was applied to the system. The disturbance had a value of $\Delta v_8 = -15\%$ and was active in the interval from $t = 70$ min to $t = 110$ min. Without the disturbance, the effective opening of the valve corresponds to the input signal computed by the control strategy, i.e. $v_8 = u$, whereas during the application of the disturbance the valve opening is given by $v_8 = u + \Delta v_8$. The application of the disturbance leads to a lower valve opening v_8 and results in an increasing temperature. With an increasing error in the system output, the control strategy gradually opens the valve and compensates the divergence. When the system reaches steady state in $t = 100$ min, the effective valve opening corresponds to the one before the application of the disturbance. After the disappearance of the disturbance, the proposed control strategy reduces the input signal and stabilizes the system output in the given reference. The complete disturbance rejection underlines the robustness of the proposed control strategy.

In the last experiment (see Fig. 10) with the pilot plant emulating an exothermic chemical reaction, the proposed MMMPC strategy with guaranteed stability was validated by means of a disturbance in the feed F_f. The disturbance $\Delta F_f = -0.02$ l/s, which corresponds to an error of −40% with respect to the nominal feed, was applied to the system in $t = 60$ min and held constant until the end of the experiment.

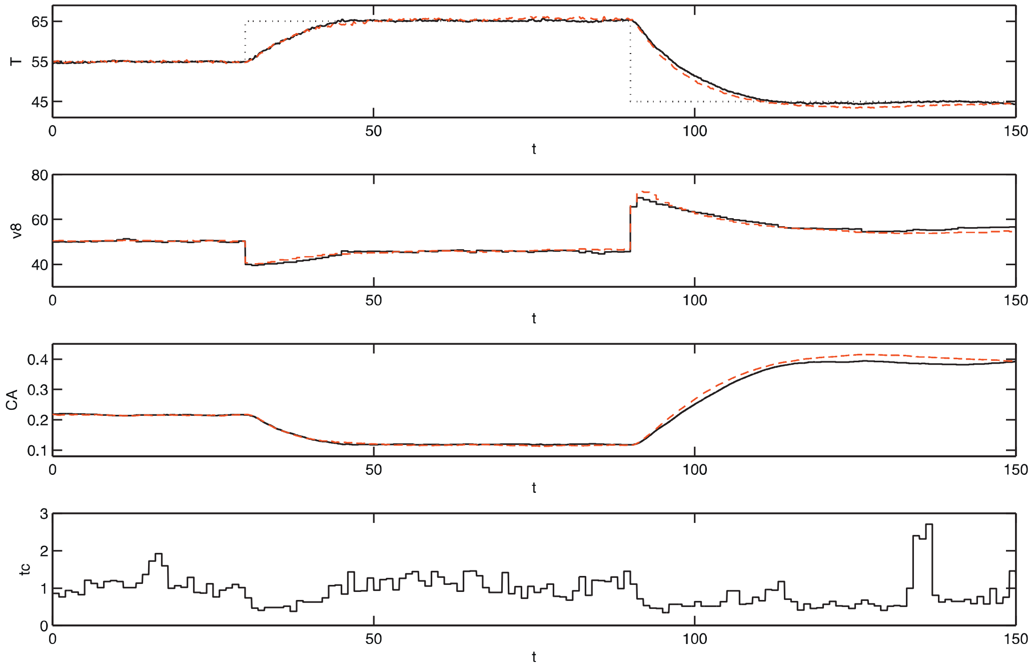

Fig. 8. Reference tracking experiment with a constant error in the underlying model of the exothermic chemical reaction using the proposed MMMPC strategy (solid line) and the iterative NMPC approach (dashed line). From top to bottom: tank temperature (T), valve opening (v_8), reagent concentration (C_A) and computation time (t_c).

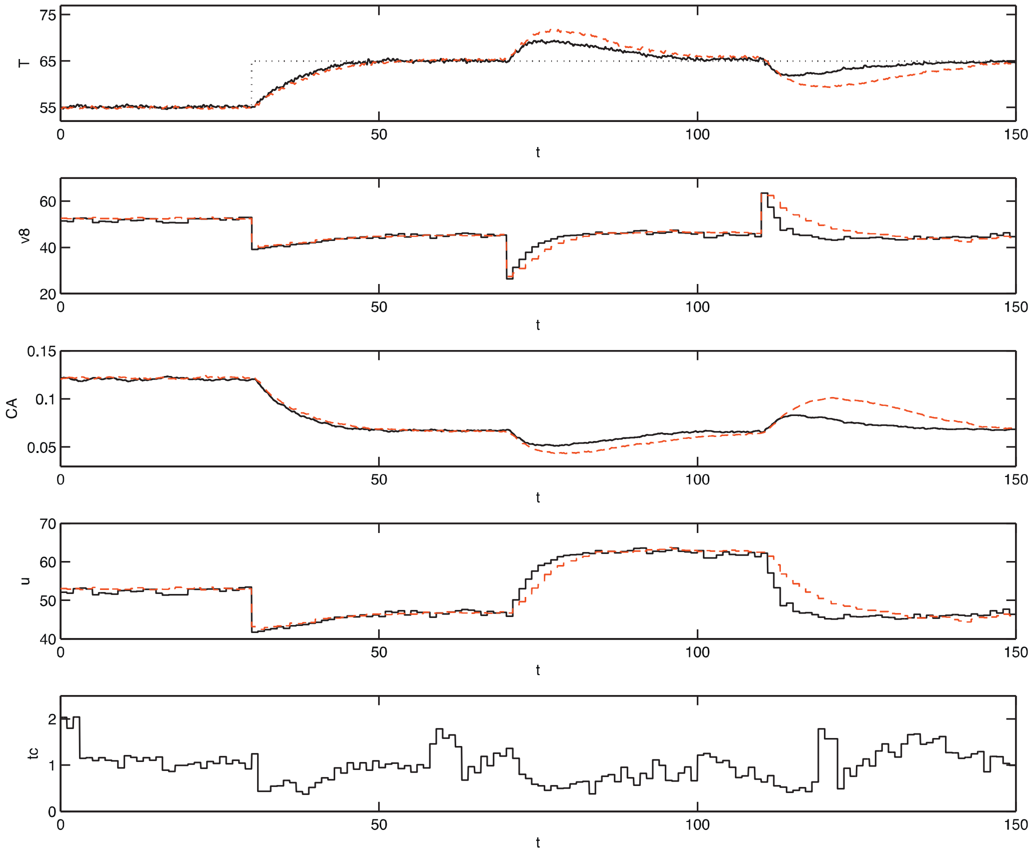

Fig. 9. Disturbance rejection experiment with an error in the valve v_8 using the proposed MMMPC strategy (solid line) and the iterative NMPC approach (dashed line). From top to bottom: tank temperature (T), valve opening (v_8), reagent concentration (C_A), input value calculated by the controller (u) and computation time (t_c).

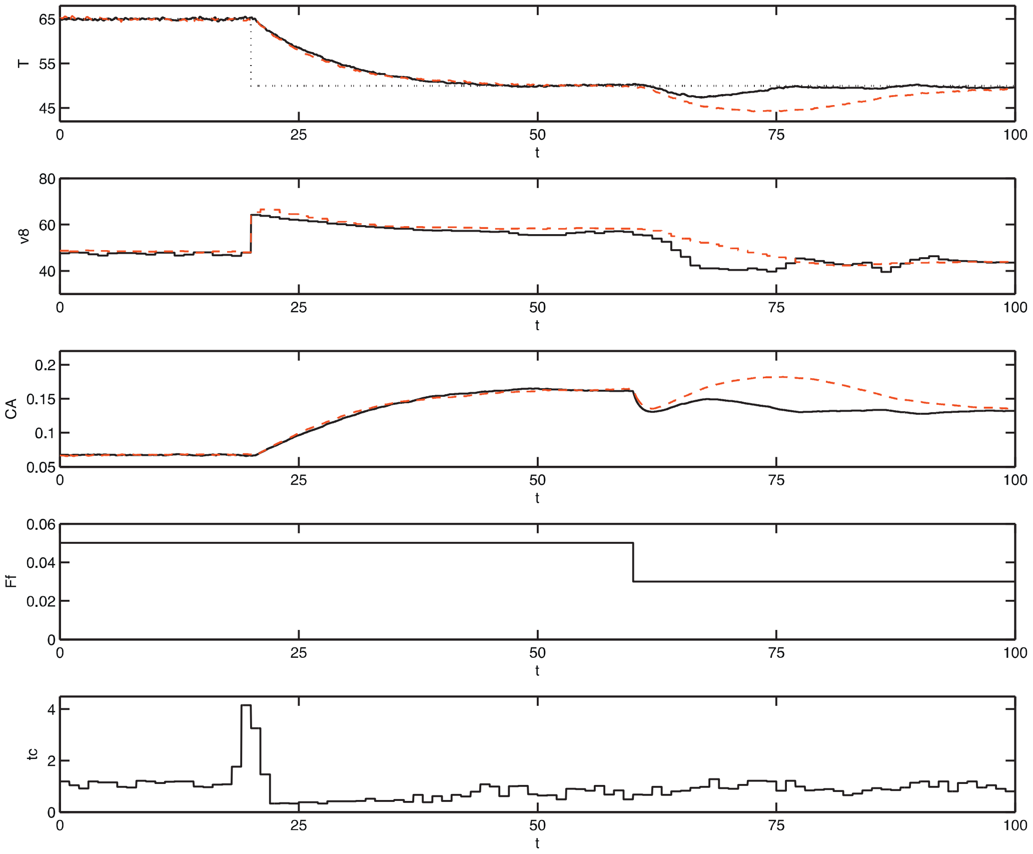

Fig. 10. Disturbance rejection experiment with an error in the feed F_f using the proposed MMMPC strategy (solid line) and the iterative NMPC approach (dashed line). From top to bottom: tank temperature (T), valve opening (v_8), reagent concentration (C_A), feed (F_f) and computation time (t_c).

The disturbance in the feed reduces the supply of the reactive and leads to a decreasing concentration C_A. As a consequence, the chemical reaction slows down and the temperature falls below the given setpoint. With an increasing error in the system output (with a maximum error of −2.6 °C), the control strategy reduces the valve opening and rejects the disturbance. The approximately 15 min required for the disturbance rejection and the observable small oscillations after the stabilization of the system are justified by the magnitude of the disturbance and the resulting mismatch between the prediction model and the controlled system.

The results of the iterative NMPC strategy can be seen in Fig. 7 (setpoint tracking), Fig. 8 (setpoint tracking with error in the exothermic chemical reaction), Fig. 9 (rejection of a disturbance in the valve v_8) and Fig. 10 (rejection of a disturbance in the feed F_f). The controller based on the iterative approach shows similar results in the setpoint tracking experiment, but shows a worse control performance in the disturbance rejection experiments. The proposed MMMPC stabilizes the system more efficiently with less overshoots than the iterative NMPC. During the disturbance rejection and the experiment with a modeling error the min–max controller benefits from its robustness and gains better results.

The proposed MMMPC strategy has been implemented as a Matlab function where t_c corresponds to the required time to compute a new control signal. The computation time t_c is the time to execute the entire function, i.e. from supplying the last temperature measurement until obtaining the new control signal. The optimization is based on Matlab's built-in function `fmincon` for sequential quadratic programming (SQP) which uses an iterative algorithm for the minimization. The necessary time to find the optimal solution depends on the initial guess supplied to the algorithm leading to a faster optimization in the case of an initial guess close to the optimal solution and a slower minimization for an initial solution far away from the optimum. The proposed MMMPC strategy uses the solution obtained in the previous sample as an initial guess, i.e. in steady-state a fast optimization can be expected whereas minimization will require more time during transients. The experimental results given in Figs. 7–10 show important variations in the computation time t_c especially after setpoint changes or the appearance of disturbances. In these moments, the initial guess supplied to the algorithm is rather far away from the optimal solution and leads to an increased computation time t_c. It can be observed that the computation time t_c decreases after the compensation of the output error. It is noteworthy that the maximum computation time obtained in the experiments was $t_c^{max} = 4.408$ s and therefore clearly below the used sampling time of $t_s = 60$ s. The average and minimum computation times observed in the experiments were $t_c^{avg} = 0.925$ s and $t_c^{min} = 0.285$ s, respectively. It has to be emphasized that the required computational effort is quite low considering the used horizons and allows the use of the proposed MMMPC strategy in moderately fast real time applications.

8. Conclusions

In this paper a nonlinear MMMPC strategy based on second order Volterra series models was presented. The used Volterra series model is extended by an additive term to consider the effect of uncertainties and disturbances. The non-autoregressive structure of the mathematical model is exploited and an explicit formulation of the exact worst case cost is obtained. The explicit formulation reduces the original min–max optimization problem to a pure minimization problem with a considerably reduced computational complexity. Furthermore, input-to-state practical stability is proven for sufficiently long prediction horizons.

The low computational burden allowed the application of the proposed MMMPC to a laboratory process. The considered process is a pilot plant emulating an exothermic chemical reaction by means of an electric heater. Realistic values of the prediction and control horizons were used together with a relatively fast sampling time. The results obtained in setpoint tracking and disturbance rejection experiments showed a good control performance of the proposed MMMPC and underlined the low complexity of the optimization problem. Additionally, experiments with an NMPC (iterative control strategy based on the same Volterra series model) have been carried out to compare the results obtained with the MMMPC. The higher robustness of the MMMPC leads to a better control performance in presence of disturbances or model mismatch. Hence, the use of the min–max control is justified for uncertain models or in the case of disturbances. The presented application to a real time process joins the small number of MMMPC applications reported in specialized literature.

Future research will be focused on the inclusion of output constraints in the min–max optimization problem. Besides, the effect of different uncertainty terms could be analyzed under consideration of the stabilizing behavior of the resulting control strategy.

Acknowledgement

Financial support by the Spanish Ministry of Economy and Competitiveness under grant DPI2010-21589-C05-01 is gratefully appreciated.

Appendix A. Transformation to a state-space representation

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

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

$$x_i(k) = u(k-i) \quad \text{for} \quad i = 1, \ldots, N_t \tag{A.1}$$

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 (A.1) can be rewritten in the form:

$$\begin{aligned} x_1(k) &= u(k-1) \\ x_2(k) &= x_1(k-1) \\ x_3(k) &= x_2(k-1) \\ \vdots \quad & \vdots \ \vdots \\ x_{N_t}(k) &= x_{N_t-1}(k-1) \end{aligned} \tag{A.2}$$

where the first state $x_1(k)$ represents the last applied input signal and the remaining states depend on the states from the previous instant. With the states $x_i(k)$ for $i=1, \ldots, N_t$ defined in (A.2), the model output of the second order Volterra series model (1) can be expressed by:

$$y(k) = l(\boldsymbol{x}(k)) = \sum_{i=1}^{N_t} h_1(i)x_i(k) + \sum_{i=1}^{N_t}\sum_{j=i}^{N_t} h_2(i,j)x_i(k)x_j(k) \tag{A.3}$$

Note the similarity of (A.3) and (1) where the past input values $u(k-i)$ for $i=1, \ldots, N_t$ have been substituted by the previously defined states $x_i(k)$ for $i=1, \ldots, N_t$. Now it can be seen easily that (1) can be expressed as a nonlinear state-space model defined by:

$$\begin{aligned} &\boldsymbol{x}(k+1) = A\boldsymbol{x}(k) + Bu(k) \\ &y(k) = l(\boldsymbol{x}(k)) \end{aligned} \tag{A.4}$$

where 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.2) as:

$$A = \begin{bmatrix} 0 & 0 & \ldots & 0 & 0 & 0 \\ 1 & 0 & \ldots & 0 & 0 & 0 \\ 0 & 1 & \ldots & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \ldots & 1 & 0 & 0 \\ 0 & 0 & \ldots & 0 & 1 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix} \tag{A.5}$$

The state-space-like model (A.4) can be used easily as a prediction model in an MMMPC framework. Under consideration of the current estimation error $d(k)$ and the future uncertainty $\theta(k+i|k)$, (A.4) can be rewritten in the following form:

$$\begin{aligned} &\boldsymbol{x}(k+i+1|k) = A\boldsymbol{x}(k+i|k) + Bu(k+i|k) \\ &y(k+i|k) = l(\boldsymbol{x}(k+i|k)) + d(k) + \theta(k+i|k) \end{aligned} \tag{A.6}$$

Appendix B. Proof of Theorem 1

This section presents the detailed proof of Theorem 1 (see Section 5.3) used to demonstrate input-to-state stability of the proposed MMMPC strategy.

B.1. General statements

The following lemmas and theorems have been applied to define upper bounds for the terms $\alpha_1, \alpha_2, \alpha_3$ and α_4 used in the cost difference (B.6):

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

Lemma 2. *The inverse of the static output nonlinearity of the Volterra series model in state space representation (i.e. φ) is Lipschitz continuous and, as a consequence the condition $\|\varphi(a_1 + a_2) - \varphi(a_1)\| \leq L_\chi \| a_2 \|$ is satisfied.*

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

Theorem 2. *Consider the steady-state input difference $\Delta u_r = u_r(k+1) - u_r(k)$ where $u_r(k+1)$ and $u_r(k)$ denote the steady-state inputs at k and $k+1$, respectively. Also, consider the state values $\boldsymbol{x}^f(k+i|k+1)$ predicted at time $k+1$ using the feasible solution $\boldsymbol{u}^f(k+1)$ (see Section 5.2). Being $c_{\boldsymbol{x}}$ 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 $i = N_u+1, \ldots, N$ is bounded by:*

$$\|\Delta\boldsymbol{x}(k+i)\| \leq c_{\boldsymbol{x}} \|\Delta u_r\| \tag{B.1}$$

Proof. The predicted states satisfy $\boldsymbol{x}^f(k+i|k+1) = \boldsymbol{x}^*(k+i|k)$ for $i = 1, \ldots, N_u$ as $u^f(k+i|k+1) = u^*(k+i|k)$ for $i = 1, \ldots, N_u - 1$. Thus, $\Delta\boldsymbol{x}(k+i) = 0$ for $i = 1, \ldots, N_u$. Furthermore, the prediction of the states based on the optimal and the feasible solution are defined for $i = N_u+1, \ldots, N$ by:

$$\begin{aligned} \boldsymbol{x}^*(k+i|k) &= A\boldsymbol{x}^*(k+i-1|k) + Bu_r(k) \\ \boldsymbol{x}^f(k+i|k+1) &= A\boldsymbol{x}^f(k+i-1|k+1) + Bu_r(k+1) \end{aligned} \tag{B.2}$$

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

$$\Delta\boldsymbol{x}(k+i) = A\Delta\boldsymbol{x}(k+i-1) + B\Delta u_r \tag{B.3}$$

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 (B.3) ensures that exists $c_{\boldsymbol{x}} > 0$, $c_{\boldsymbol{x}} = \| A^{N-1}B + \cdots + AB + B \|$ such that $\|\Delta\boldsymbol{x}(k+i)\| \leq c_{\boldsymbol{x}} \| \Delta u_r \|$, for $i = N_u+1, \ldots, N$. Then, (B.1) holds for all i. □

B.2. Definition of the cost difference

Consider the cost $J^f(\boldsymbol{x}(k+1))$ at $k+1$ depending on the feasible solution $\boldsymbol{u}^f(k+1)$ (see Section 5.2), which can be expressed as:

$$J^f(\boldsymbol{x}(k+1)) = \sum_{i=1}^{N_u-1} L(\boldsymbol{x}^f(k+i|k+1), u^f(k+i|k+1), d(k+1), \theta^f(k+i|k+1)) + \sum_{i=N_u}^{N+1} L_h(\boldsymbol{x}^f(k+i|k+1)) \tag{B.4}$$

where $\theta^f(k+i|k+1)$ is the ith component of $\boldsymbol{\theta}^f$ defined as in (18) for $\boldsymbol{u}^f(k+1)$. The cost difference $\Delta J(k+1) = J^f(\boldsymbol{x}(k+1)) - J^*(\boldsymbol{x}(k))$ based on the costs given in (32) and (B.4) becomes:

$$\begin{aligned} \Delta J(k+1) = {} & L_h(\boldsymbol{x}^f(k+N+1|k+1)) - L(\boldsymbol{x}^*(k|k), u^*(k|k), d(k), \theta^*(k|k)) \\ & + \sum_{i=1}^{N_u-1} (L(\boldsymbol{x}^f(k+i|k+1), u^f(k+i|k+1), d(k+1), \theta^f(k+i|k+1)) - L(\boldsymbol{x}^*(k+i|k), u^*(k+i|k), d(k), \theta^*(k+i|k))) \\ & + \sum_{i=N_u}^{N} L_h(\boldsymbol{x}^f(k+i|k+1)) - L_h(\boldsymbol{x}^*(k+i|k)) \end{aligned} \tag{B.5}$$

Hence, taking into account that $\theta^*(k|k) = 0$, the difference of the two cost functions (B.5) can be written in the following form:

$$\Delta J(k+1) = -L(\boldsymbol{x}^*(k|k), u^*(k|k), d(k), 0) + \alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 \tag{B.6}$$

with the terms given by:

$$\alpha_1 = \sum_{i=1}^{N_u-1} (\|l(\boldsymbol{x}^f(k+i|k+1)) + \theta^f(k+i|k+1) + d(k+1) - r(k+1)\|_Q^2 - \|l(\boldsymbol{x}^*(k+i|k)) + \theta^*(k+i|k) + d(k) - r(k)\|_Q^2) \tag{B.7}$$

$$\alpha_2 = \sum_{i=1}^{N_u-1} (\|u^f(k+i|k+1) - u_r(k+1)\|_R^2 - \|u^*(k+i|k) - u_r(k)\|_R^2) \tag{B.8}$$

$$\alpha_3 = \sum_{i=N_u}^{N} (\|l(\boldsymbol{x}^f(k+i|k+1)) + \theta^f(k+i|k+1) + d(k+1) - r(k+1)\|_Q^2 - \|l(\boldsymbol{x}^*(k+i|k)) + \theta^*(k+i|k) + d(k) - r(k)\|_Q^2) \tag{B.9}$$

$$\alpha_4 = L_h(\boldsymbol{x}^f(k+N+1|k+1)) \tag{B.10}$$

The following theorems define upper bounds of the terms $\alpha_1, \alpha_2, \alpha_3$ and α_4 used in the difference of costs (B.6):

Theorem 3. *Consider the estimation error increment $\Delta d = d(k+1) - d(k)$ where $d(k+1)$ and $d(k)$ denote the estimation errors at $k+1$ and k, respectively. Furthermore, consider the constant ε which limits the uncertainty to $|\theta(k+i|k)| \leq \varepsilon$ for $i = 1, \ldots, N$. Then, the term α_1(B.7) is bounded by:*

$$\alpha_1 \leq c_1(Q, N_u) \cdot \|\Delta d + 2\varepsilon\| \tag{B.11}$$

where c_1 denotes a positive Lipschitz constant.

Proof. With $\boldsymbol{x}^*(k+1|k) = \boldsymbol{x}(k+1)$ and $u^f(k+i|k+1) = u^*(k+i|k)$ for $i = 1, \ldots, N_u - 1$, the predicted states computed with the optimal and the feasible solutions 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 increments in the estimation error and the uncertainty are given generally as $\Delta d = d(k+1) - d(k)$ and $\Delta\theta(k+i) = \theta^f(k+i|k+1) - \theta^*(k+i|k)$, respectively. Defining the auxiliary variable $z(k+i|k) = l(\boldsymbol{x}^*(k+i|k)) + d(k) + \theta(k+i|k) - r(k)$, the term α_1 can be expressed as:

$$\alpha_1 = \sum_{i=1}^{N_u-1} \|z(k+i|k) + \Delta d + \Delta\theta(k+i)\|_Q^2 - \|z(k+i|k)\|_Q^2 \tag{B.12}$$

and bounded under consideration of Lemma 1 by:

$$\alpha_1 \leq L_q \sum_{i=1}^{N_u-1} \|\Delta d + \Delta\theta(k+i)\| \tag{B.13}$$

With the uncertainty limited by $\|\theta(\cdot)\| \leq \varepsilon$ the increment in the uncertainty is bounded by $\|\Delta\theta(\cdot)\| \leq 2\varepsilon$. With this boundary the term α_1 is limited to:

$$\alpha_1 \leq c_1(Q, N_u) \cdot \|\Delta d + 2\varepsilon\| \tag{B.14}$$

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

Theorem 4. *Consider the estimation error increment $\Delta d = d(k+1) - d(k)$ where $d(k+1)$ and $d(k)$ denote the estimation errors at $k+1$ and k, respectively. Furthermore, consider the constant ε which limits the uncertainty to $|\theta(k+i|k)| \leq \varepsilon$ for $i = 1, \ldots, N$. Then, the term α_2(B.8) is bounded by:*

$$\alpha_2 \leq c_2(R, L_\chi, N_u) \cdot \|\Delta d\| \tag{B.15}$$

where c_2 denotes a positive Lipschitz constant.

Proof. Consider the optimal solution and the feasible solution satisfying $u^f(k+i|k+1) = u^*(k+i|k)$ for $i = 1, \ldots, N_u - 1$ (31). Defining the increment in the steady-state input as $\Delta u_r = u_r(k+1) - u_r(k)$, the term α_2 can be expressed with the auxiliary variable $z_1(k+i|k) = u^*(k+i|k) - u_r(k)$ in the form:

$$\alpha_2 = \sum_{i=1}^{N_u-1} \|z_1(k+i|k) - \Delta u_r\|_R^2 - \|z_1(k+i|k)\|_R^2 \tag{B.16}$$

Then, applying Lemma 1 to (B.16), the term α_2 can be bounded by:

$$\alpha_2 \leq c_2(R, N_u) \cdot \|\Delta u_r\| \tag{B.17}$$

where $c_2(\cdot, \cdot)$ is a positive parameter. Taking in account the definition (4), the increment in the steady-state input can be written in the following form:

$$\Delta u_r = \varphi(r(k+1) - d(k+1)) - \varphi(r(k) - d(k)) \tag{B.18}$$

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

$$\Delta u_r = \varphi(z_2 - \Delta d) - \varphi(z_2) \tag{B.19}$$

Under consideration of Lemma 2 the norm of Δu_r can be bounded by $\|\Delta u_r\| \leq L_\chi \|\Delta d\|$. Hence, the increment in the steady-state input is bounded by:

$$\|\Delta u_r\| \leq L_\chi \|\Delta d\| \tag{B.20}$$

Using (B.20) in (B.17), the term α_2 can be finally bounded by:

$$\alpha_2 \leq c_2(R, L_\chi, N_u) \cdot \|\Delta d\| \tag{B.21}$$

being $c_2(\cdot, \cdot, \cdot)$ a positive Lipschitz constant which depends on the weighting factor R, the parameter L_χ and the control horizon N_u. □

Theorem 5. *Consider the estimation error increment $\Delta d = d(k+1) - d(k)$ where $d(k+1)$ and $d(k)$ denote the estimation errors at $k+1$ and k, respectively. Furthermore, consider the constant ε which limits the uncertainty to $|\theta(k+i|k)| \leq \varepsilon$ for $i = 1, \ldots, N$. Then, the term α_3(B.9) is bounded by:*

$$\alpha_3 \leq c_3(Q, L_\chi, L_l, c_{\boldsymbol{x}}, N, N_u) \cdot \|\Delta d + 2\varepsilon\| \tag{B.22}$$

where c_3 denotes a positive Lipschitz constant.

Proof. Consider the predictions $\boldsymbol{x}^*(k+i|k)$ and $\boldsymbol{x}^f(k+i|k+1)$ for $i = N_u, \ldots, N$ 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 $i = N_u, \ldots, N$ and the initial condition is $\boldsymbol{x}^f(k+N_u-1|k+1) = \boldsymbol{x}^*(k+N_u-1|k)$. Furthermore, consider the increment in the estimation error $\Delta d = d(k+1) - d(k)$ and the difference in the uncertainty $\Delta\theta(k+i) = \theta^f(k+i|k+1) - \theta^*(k+i|k)$. With the auxiliary variables:

$$\begin{aligned} z_1(k+i) &= l(\boldsymbol{x}^*(k+i|k) + \Delta\boldsymbol{x}(k+i)) - l(\boldsymbol{x}^*(k+i|k)) + \Delta d + \Delta\theta(k+i) \\ z_2(k+i) &= l(\boldsymbol{x}^*(k+i|k)) + d(k) + \theta^*(k+i|k) - r(k) \end{aligned} \tag{B.23}$$

and a constant reference, i.e. $r(k+1) = r(k)$, the term a_3 can be expressed as:

$$\alpha_3 = \sum_{i=N_u}^{N} \|z_1(k+i) + z_2(k+i)\|_Q^2 - \|z_2(k+i)\|_Q^2 \tag{B.24}$$

Applying Lemma 1 to (B.24) the term a_3 can be bounded in the following form:

$$\alpha_3 \leq c_3(Q, N, N_u) \cdot \sum_{i=N_u}^{N} \|z_1(k+i)\| \tag{B.25}$$

Furthermore, with the function l being Lipschitz continuous, the term $z_1(k+i)$ can be bounded by Lemma 3:

$$\|z_1(k+i)\| \leq \|l(\boldsymbol{x}^*(k+i|k) + \Delta\boldsymbol{x}(k+i)) - l(\boldsymbol{x}^*(k+i|k))\| + \|\Delta d + \Delta\theta(k+i)\| \leq L_l\|\Delta\boldsymbol{x}(k+i)\| + \|\Delta d + \Delta\theta(k+i)\| \tag{B.26}$$

Hence, using (B.26) in (B.25), the upper bound of α_3 can be expressed as:

$$\alpha_3 \leq \hat{c}_3(Q, N, N_u) \cdot \sum_{i=N_u}^{N} (L_l\|\Delta\boldsymbol{x}(k+i)\| + \|\Delta d + \Delta\theta(k+i)\|) \tag{B.27}$$

With the help of Theorem 2 the difference of the predicted states based on the optimal and the feasible solution can be bounded with $\|\Delta\ \boldsymbol{x}(k+i)\| \leq c_{\boldsymbol{x}} \|\ \Delta u_r\|$. Using this upper bound, (B.27) can be rewritten as:

$$\alpha_3 \leq \hat{c}_3(Q, N, N_u) \cdot \sum_{i=N_u}^{N} (L_l c_{\boldsymbol{x}}\|\Delta u_r\| + \|\Delta d + \Delta\theta(k+i)\|) \tag{B.28}$$

Finally, with Lemma 2 the increment in the steady-state input can be bounded by $\|\Delta u_r\| \leq L_\chi \|\ \Delta d + 2\varepsilon\ \|$ (see explanation for α_2). Furthermore, with the uncertainty limited by $\|\theta(\cdot)\| \leq \varepsilon$ the difference in the uncertainty is bounded by $\|\Delta\theta(\cdot)\| \leq 2\,\varepsilon$. Hence, the upper bound of α_3 (B.28) is defined by:

$$\alpha_3 \leq c_3(Q, L_\chi, L_l, c_{\boldsymbol{x}}, N, N_u) \cdot \|\Delta d + 2\varepsilon\| \tag{B.29}$$

being $c_3(\cdot, \cdot)$ a positive Lipschitz constant which depends on the weighting factor Q, the parameters L_χ, L_l and $c_{\boldsymbol{x}}$, the prediction horizon N and the control horizon N_u. □

Theorem 6. *Consider $\alpha_4 = L_h(\boldsymbol{x}^f(k+N+1|k+1))$, then, α_4 it is bounded by:*

$$\alpha_4 \leq c_4(Q)\varepsilon \tag{B.30}$$

being c_4 a positive constant that depends on Q.

Proof. By definition α_4 is equal to:

$$\alpha_4 = L_h(\boldsymbol{x}^f(k+N+1|k+1)) = \|l(\boldsymbol{x}^f(k+N+1|k+1)) + d(k+1) + \theta^f(k+N+1|k+1) - r(k+1)\|_Q^2 \tag{B.31}$$

Now consider the nilpotent character of the prediction model (21), that is, $A^N = 0$ for $N \geq N_t$. For a prediction horizon of $N \geq N_u + N_t$, the local control law, defined by 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 $\boldsymbol{x}^f(k+N+1|k+1)$ 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 (24), it follows that:

$$l(\boldsymbol{x}^f(k+N+1|k+1)) = r(k+1) - d(k+1) \tag{B.32}$$

Taking into account (B.32) in (B.31) it follows that:

$$\alpha_4 = \|\theta^f(k+N+1|k+1)\|_Q^2 \tag{B.33}$$

which in turn can be bounded by $c_4(Q)\varepsilon$ as in (B.30). □

B.3. Upper bound of the cost difference

Under consideration of the upper bounds of the terms $\alpha_1, \alpha_2, \alpha_3$ and α_4 defined in the previous section, Theorem 1 can be proven by:

Proof *(Theorem 1).* Consider the cost difference $\Delta J(k+1) = J^f(\boldsymbol{x}(k+1)) - J^*(\boldsymbol{x}(k))$ given in (B.6). Furthermore, consider Theorems 3–6 with the definitions of upper bounds of the terms α_1 (B.11), α_2 (B.15), α_3 (B.22) and α_4 (B.30), respectively. Using these upper bounds, and the cost difference $\Delta J(k+1)$ is bounded by:

$$J^f(\boldsymbol{x}(k+1)) - J^*(\boldsymbol{x}(k)) \le -L(\boldsymbol{x}^*(k|k), u^*(k|k), d(k), 0) + c_V \cdot \|\Delta d + 2\,\varepsilon\| + c_2(R, L_\chi, N_u) \cdot \|\Delta d\| + c_4(Q)\varepsilon \tag{B.34}$$

The positive constant c_V is defined by (B.11) and (B.22) as:

$$c_V = c_1(Q, N_u) + c_3(Q, L_\chi, L_l, c_{\boldsymbol{x}}, N, N_u) \tag{B.35}$$

and depends on the weighting factors Q and R, the parameters L_χ, L_l and $c_{\boldsymbol{x}}$, the prediction horizon N and the control horizon N_u. Furthermore, note that $\| \Delta d + 2\,\varepsilon \| \le \| \Delta d \| + 2\,\varepsilon$, thus:

$$J^f(\boldsymbol{x}(k+1)) - J^*(\boldsymbol{x}(k)) \le -L(\boldsymbol{x}^*(k|k), u^*(k|k), d(k), 0) + c_d \cdot \|\Delta d\| + c_\varepsilon \cdot \varepsilon \tag{B.36}$$

with $c_d = c_V + c_2(R, L_\chi, N_u)$ and $c_\varepsilon = 2\,c_V + c_4(Q)$.

On the other hand, for the cost $J^*(\boldsymbol{x}(k+1))$ based on the optimal solution $\boldsymbol{u}^*(k+1)$ at $k+1$ the statement:

$$J^*(\boldsymbol{x}(k+1)) \le J^f(\boldsymbol{x}(k+1)) \tag{B.37}$$

holds. Thus, we can use this into (B.34) to obtain the bound on the difference of the optimal costs:

$$J^*(\boldsymbol{x}(k+1)) - J^*(\boldsymbol{x}(k)) \le -L(\boldsymbol{x}^*(k|k), u^*(k|k), d(k), 0) + c_d \cdot \|\Delta d\| + c_\varepsilon \cdot \varepsilon \tag{B.38}$$

□

References

[1] J.M. Maciejowski, Predictive Control with Constraints, Prentice Hall, Harlow, UK, 2000.

[2] J.A. Rossiter, Model-based Predictive Control: A Practical Approach, CRC Press, Boca Raton, FL, 2003.

[3] E.F. Camacho, C. Bordons, Model Predictive Control, 2nd ed., Springer, London, UK, 2004.

[4] J.B. Rawlings, D.Q. Mayne, Model Predictive Control Theory and Design, Nob Hill Publishing, Madison, WI, 2009.

[5] H.S. Witsenhausen, A min–max control problem for sampled linear systems, IEEE Transactions on Automatic Control 13 (1) (1968) 5–21.

[6] P.J. Campo, M. Morari, Robust model predictive control, in: Proceedings of the 1987 American Control Conference, Minneapolis, MN, 1987, pp. 1021–1026.

[7] J.C. Allwright, G.C. Papavasiliou, On linear programming and robust model-predictive control using impulse-responses, Systems & Control Letters 18 (2) (1992) 159–164.

[8] J. Löfberg, Minimax approaches to robust model predictive control, Ph.D. thesis, Department of Electrical Engineering, Linköping University, 2003.

[9] P.O.M. Scokaert, D.Q. Mayne, J.B. Rawlings, Suboptimal model predictive control (feasibility implies stability), IEEE Transactions on Automatic Control 44 (3) (1999) 648–654.

[10] D.Q. Mayne, J.B. Rawlings, C.V. Rao, P.O.M. Scokaert, Constrained model predictive control: stability and optimality, Automatica 36 (6) (2000) 789–814.

[11] Y.H. Kim, W.H. Kwon, An application of min–max generalized predictive control to sintering processes, Control Engineering Practice 6 (8) (1998) 999–1007.

[12] D. Mu noz de la Pe na, D.R. Ramirez, E.F. Camacho, T. Alamo, Application of an explicit min–max MPC to a scaled laboratory process, Control Engineering Practice 13 (12) (2005) 1463–1471.

[13] J.K. Gruber, D.R. Ramirez, T. Alamo, C. Bordons, E.F. Camacho, Control of a pilot plant using QP based min–max predictive control, Control Engineering Practice 17 (11) (2009) 1358–1366.

[14] R.D. Nowak, B.D. Van Veen, Nonlinear system identification with pseudorandom multilevel excitation sequences, in: Proceedings of the 1993 IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 4, Minneapolis, MN, 1993, pp. 456–459.

[15] A.Y. Kibangou, G. Favier, M.M. Hassani, Selection of generalized orthonormal bases for second-order Volterra filters, Signal Processing 85 (12) (2005) 2371–2385.

[16] F.J. Doyle, R.K. Pearson, B.A. Ogunnaike, Identification and Control Using Volterra Models, Springer, London, 2001.

[17] F. Dorado, Control predictivo no lineal basado en modelos de Volterra (in Spanish), Ph.D. thesis, Departamento de Ingeniería de Sistemas y Automática. Universidad de Sevilla, 2006.

[18] J.K. Gruber, C. Bordons, R. Bars, R. Haber, Nonlinear predictive control of smooth nonlinear systems based on Volterra models. Application to a pilot plant, International Journal of Robust and Nonlinear Control 20 (16) (2009) 1817–1835.

[19] J.K. Gruber, Efficient and robust techniques for predictive control of nonlinear processes, Ph.D. thesis, Departamento de Ingeniería de Sistemas y Automática. Universidad de Sevilla, http://fondosdigitales.us.es/media/thesis/1243/D_T.PROV9.pdf, 2010.

[20] W.J. Rugh, Nonlinear System Theory: The Volterra/Wiener Approach, Johns Hopkins University Press, Baltimore, MD, 1981.

[21] S. Boyd, L.O. Chua, Fading memory and the problem of approximating nonlinear operators with Volterra series, IEEE Transactions on Circuits and Systems 32 (11) (1985) 1150–1161.

[22] D.R. Ramirez, T. Alamo, E.F. Camacho, D. Mu noz de la Pe na, Min–Max MPC based on a computationally efficient upper-bound of the worst case cost, Journal of Process Control 16 (5) (2006) 511–519.

[23] D. Limon, T. Alamo, D.M. Raimondo, D. Mu noz de la Pe na, J.M. Bravo, A. Ferramosca, E.F. Camacho, Input-to-state stability: a unifying framework for robust model predictive control, in: L. Magni, D.M. Raimondo, F. Allgöwer (Eds.), Nonlinear Model Predictive Control, Lecture Notes in Control and Information Sciences, Springer, 2009, pp. 1–26.

[24] F. Szeifert, T. Chovan, L. Nagy, Process dynamics and temperature control of fed-batch reactors, Computers & Chemical Engineering 19 (1) (1995) 447–452.

[25] D.R. Ramirez, D. Limón, J.G. Ortega, E.F. Camacho, Model Based Predictive Control Using Genetic Algorithms Application to a Pilot Plant, in: Proceedings of the 1999 European Control Conference, Karlsruhe, Germany, 1999, pp. 81–85.

[26] L.O. Santos, P.A.F.N.A. Afonso, J.A.A.M. Castro, N.M.C. Oliveira, L.T. Biegler, On-line implementation of nonlinear MPC: an experimental case study, Control Engineering Practice 9 (8) (2001) 847–857.

[27] J.H. Lee, K.S. Lee, W.C. Kim, Model-based iterative learning control with a quadratic criterion for time-varying linear systems, Automatica 36 (5) (2000) 641–657.

[28] R.D. Nowak, B.D. Van Veen, Efficient methods for identification of Volterra filter models, Signal Processing 38 (3) (1994) 417–428.