COMPUTATIONAL DELAY IN NONLINEAR MODEL PREDICTIVE CONTROL

Rolf Findeisen [*] **Frank Allgöwer** [*]

[*] *Institute for Systems Theory in Engineering, University of Stuttgart, 70550 Stuttgart, Germany,*
{findeise,allgower}@ist.uni-stuttgart.de

Abstract: By now a series of NMPC schemes exist that lead to guaranteed stability of the closed-loop. However, in these schemes the computation time to find a solution of the open-loop optimal control problem is often neglected. In practice the necessary computation time is often not negligible, and leads, since not explicitly considered, to a delay between the state information and the input signal implemented on the system. This delay can lead to a drastic performance decrease or even to instability of the closed-loop. In this paper we outline a simple approach how the computational delay can be considered in nonlinear model predictive control schemes and provide conditions under which the stability of the closed-loop can be guaranteed. This allows to employ nonlinear model predictive control even in the case that the necessary numerical solution time is significant. The presented approach is exemplified considering the control of a continuous reactor.

Keywords: nonlinear predictive control, computational delay, stability

1. INTRODUCTION

In many process control problems it is desired to design a stabilizing feedback such that a performance criterion is minimized while satisfying constraints on the controls and the states. From an optimal control point of view one would ideally like to solve the corresponding Hamilton-Jacobi-Bellmann equations to obtain an explicit solution of the corresponding feedback law. However, often the explicit solution of the corresponding partial differential equations can not be obtained. One way to circumvent this problem is the application of model predictive control(MPC) strategies.

The work presented in this paper is concerned with nonlinear model predictive control (NMPC) for continuous time processes and the problems resulting from the often not negligible on-line computation time. While by now a series of NMPC schemes exist that guarantee closed loop stability (see for example (Mayne *et al.* 2000, Rawlings 2000, Allgöwer *et al.* 1999) for an overview), in these schemes the necessary on-line computation time is typically not taken into account.Even though that recent developments in dynamic optimization have lead to efficient numerical solution methods for the open-loop optimal control problem (See e.g. (Bartlett *et al.* 2000, Findeisen *et al.* 2002, Tenny and Rawlings 2001, Diehl *et al.* 2002)), the solution time is often significant. Neglecting the resulting delay is thus of paramount interrest. Otherwise the performance might degrade significantly or even instability of the closed loop can occur.

One of the few works that take the delay into account is the work presented in (Chen *et al.* 2000). In this paper we outline a similar, rather simple method on how the occuring delay can be taken into account in sampled-data NMPC. In comparison to (Chen *et al.* 2000), the derived results allow to stabilize a wider class of systems and to consider more general cost functions. We furthermore exemplify the importance of the consideration of the delay via a small example system.

The paper is structured as follows: In Section 2 we discuss the difference between the so called sampled-data and the instantaneous approach to NMPC. Section 3 contains a description and the proof of stability for the proposed NMPC approach that takes the delay into account. The properties of this approach are discussed in Section 4. Before we conclude in Section 6 we present in Section 5 a small example considering the control of a simple CSTR.

2. SAMPLED-DATA NMPC

We consider the stabilization of continuous time nonlinear systems described by

$$\dot{x}(t) = f(x(t),u(t)), \quad x(0) = x_0 \tag{1}$$

subject to the input and state constraints

$$u(t) \in \mathcal{U}, \quad x(t) \in \mathcal{X}, \quad \forall t \geq 0, \tag{2}$$

where $x(t) \in \mathcal{X} \subseteq \mathbb{R}^n$ and $u(t) \in \mathcal{U} \subset \mathbb{R}^m$ denote the vector of states and inputs, respectively. The set of feasible inputs is denoted by $\mathcal{U}$ and the set of feasible states is denoted by $\mathcal{X}$. We assume that $\mathcal{U} \subset \mathbb{R}^m$ is compact, $\mathcal{X} \subseteq \mathbb{R}^n$ is connected and $(0,0) \in \mathcal{X} \times \mathcal{U}$. With respect to the vector field $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ we assume that it is locally Lipschitz continuous and satisfies $f(0,0) = 0$.

Model predictive control is based on the repeated solution of an open-loop optimal control problem subject to the system dynamics and the constraints. Based on the system state at time t, the controller predicts the behavior of the system over a prediction time T_p in the future[1] such that an open-loop performance objective functional is minimized. To incorporate feedback that counteracts possible disturbances, the optimal open-loop input is implemented only until the next *recalculation instant*. Based on the new system state information, the whole procedure – prediction and optimization – is repeated, moving the control and prediction horizon forward.

Mathematically the open-loop optimal control problem that is solved at the recalculation instants can be formulated as:

$$\min_{\bar{u}(\cdot)} J(\bar{u}(\cdot), x(t)) \tag{3a}$$

$$\text{s.t. } \dot{\bar{x}} = f(\bar{x},\bar{u}), \quad \bar{x}(t) = x(t) \tag{3b}$$

$$\bar{u}(\tau) \in \mathcal{U}, \ \bar{x}(\tau) \in \mathcal{X}, \ \tau \in [t, t+T_p], \tag{3c}$$

$$\bar{x}(t+T_p) \in \mathcal{E} \tag{3d}$$

where the cost function J is typically given by

$$J(\cdot) = \int_t^{t+T_p} F(\bar{x}(\tau),\bar{u}(\tau))d\tau + E(\bar{x}(t + T_p)). \tag{3e}$$

The bar denotes internal controller variables, $\bar{x}(\cdot)$ is the solution of (3b) driven by the input $\bar{u}(\cdot) : [0,T_p] \to \mathcal{U}$ with initial condition $x(t)$. We assume that the "stage cost" $F : \mathcal{X} \times \mathcal{U} \to \mathbb{R}$ is locally Lipschitz continuous with $F(0,0) = 0$ and $F(x,u) > 0 \ \forall \ \mathcal{X} \times \mathcal{U} \ni (x,u) \neq (0,0)$. The end penalty E and the terminal region constraint $\mathcal{E}$ are often used to enforce stability of the closed-loop (Mayne *et al.* 2000, Allgöwer *et al.* 1999, Fontes 2000).

In the following, optimal solutions of the dynamic optimization problem (3) are marked by $(\cdot)^*$. For example we denote the optimal input for $x(t)$ by $u^*(\cdot;x(t)) : [0,T_p] \to \mathcal{U}$.

The input applied to the system in NMPC is based on the optimal input u^*. Depending on how "often" the open-loop optimal control problem (3) is recalculated, different concepts of NMPC exist. If the open-loop is solved at all time instants, we refer to it as *instantaneous NMPC*. If the dynamic optimization is solved only at disjoint recalculation instants and the resulting optimal control signal is implemented open-loop in between, the resulting scheme is called *sampled-data NMPC*.

Instantaneous NMPC: In instantaneous NMPC the input applied to the system is given by

$$u(x(t)) = u^*(t;x(t)), \tag{4}$$

leading to the *nominal closed-loop system*

$$\dot{x}(t) = f(x(t),u(x(t))). \tag{5}$$

Various instantaneous NMPC schemes exist, see for example (Mayne *et al.* 2000). From a practical point of view instantaneous NMPC schemes are not appealing, since an open-loop optimal control problem must be solved at *all times*, which is certainly not possible in practice.

Sampled-data NMPC: In the remainder of the paper we consider sampled-data NMPC. In difference to instantaneous NMPC, in sampled-data NMPC, the open-loop optimal control problem is only solved at the *discrete* recalculation instants and the resulting optimal input signal is applied open-loop to the system until the next recalculation instant. Thus the applied input is given by

$$u(\tau) = u^*(\tau;x(t_i)) \ , \tau \in [t_i,t_{i+1}) \tag{6}$$

where t_i denotes the discrete recalculation instants. The *nominal closed-loop system* under the feedback (6) is given by

$$\dot{x}(t) = f\left(x(t),u^*(t;x(t_i))\right). \tag{7}$$

For simplicity and clarity we denote the resulting state by $x(\tau;x(t_i),u^*(\cdot;x(t_i)))$, $\tau \in [t_i,t_{i+1})$.

We assume that the recalculation instants t_i are given by a partition π of the time axis.

Definition 1 (Partition) *Every series $\pi = (t_i)$, $i \in \mathbb{N}$ of positive real numbers such that $t_0 = 0$, $t_i < t_{i+1}$ and $t_i \to \infty$ for $i \to \infty$ is called a partition. Furthermore,*

- $\bar{\pi} := \sup_{i\in\mathbb{N}}(t_{i+1} - t_i)$ *is the upper diameter of* π *(longest recalculation time).*
- $\underline{\pi} := \inf_{i\in\mathbb{N}}(t_{i+1} - t_i)$ *is the lower diameter of* π *(shortest recalculation time).* ♢

For a given t, t_i should be taken as the nearest recalculation instant with $t_i < t$. We denote the time between

[1] For simplicity we assume that the prediction and control horizon coincide.

two consecutive recalculation instants t_i and t_{i+1} as recalculation time $\delta_i^r = t_{i+1} - t_i$. Allowing for varying recalculation times, allows to re-optimize the input more frequently if the system dynamics changes rapidly. Sampled-data NMPC schemes leading to stability of the closed-loop are for example given in (Fontes 2000, Michalska and Mayne 1993, Chen and Allgöwer 1998, Jadbabaie *et al.* 2001, de Oliveira Kothare and Morari 2000, Magni and Scattolini 2002, Chen *et al.* 2000, Findeisen *et al.* 2003).

Even so that in sampled-data NMPC in principle the recalculation time $\delta_i^r = t_{i+1} - t_i$ is available for the solution of the open-loop optimal control problem, most of the existing standard NMPC schemes that guarantee stability do not take the necessary solution time for (3) and the resulting delay into account. One of the few exceptions is the work presented in (Chen *et al.* 2000), in which the computational delay is taken into account by optimizing at every recalculation instant, based on a prediction of the state at the next recalculation instant, the open-loop optimal control problem for the next recalculation instant. The purpose of this work is to expand the results in (Chen *et al.* 2000) and to outline rather general conditions that guarantee that the closed loop is stable. Furthermore we underpin by a simple example the importance of a correct consideration of the occuring computational delay.

We assume in the following that the *maximum time for finding the solution* to the open-loop optimal control problem is known and denoted by $\bar{\delta}^c$. Furthermore, we assume that the lower diameter of the recalculation instant partition $\underline{\pi}$ satisfies $\underline{\pi} \geq \bar{\delta}^c$ and that $T_p \geq \bar{\pi}$.

Remark 1 *Note, that we do not necessarily sample and hold the input in between recalculation instants. The reason for this is twofold: First of all the use of a fixed input does not allow to achieve asymptotic convergence to the origin if it is not considered during the optimization without* decreasing the recalculation time to zero. *Secondly, in practice the recalculation time is either predetermined by the time needed to solve the open-loop optimal control problem or by an external scheduling mechanism. It is typically significantly larger than the sampling time of the process control system. As sampling time* δ^s *we refer to the time the process control system operates, i.e. the A/D and D/A converter operate. Typically, the sampling time is in the order of seconds, whereas the time needed for solving the open-loop optimal control problem (which often also defines the recalculation time) is typically in the order of tenth of seconds, minutes or even tenth of minutes. Thus the open-loop optimal input signal that is applied to the system during* $[t_i, t_{i+1}]$ *can be sufficiently well approximated by a sample and hold staircase related to the sampling time* δ^s *of the D/A converters see Figure 1. Since the sampling time is often significantly faster then the recalculation time, the remaining approximation error can be seen as an (small) input disturbance, which NMPC under certain conditions is able to handle (Findeisen* et al. *2003).*$\diamondsuit$

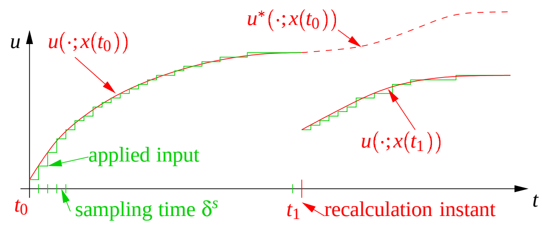

Fig. 1. Recalculation time, sampling time and sample and hold.

The next section outlines a simple approach to consider the necessary solution time in the NMPC problem and gives conditions under which stability of the closed loop can be guaranteed.

3. NMPC AND COMPUTATIONAL DELAY

The approach we propose is based on the idea to continue applying the input from the last recalculation instant t_i also during the (maximum) time $\bar{\delta}^c$ needed for solving the open-loop optimal control problem. In comparison to (6), the open-loop input that is applied to the system is thus given by:

$$u(\tau) = u^*(\tau; x(t_i)), \ \tau \in [t_i + \bar{\delta}^c, t_{i+1} + \bar{\delta}^c).$$

Since the input for the time $[t_i, t_i + \bar{\delta}^c)$ is now given by the previous recalculation, it is not any longer available as degree of freedom in the open-loop optimal control problem (3). Thus problem (3) must be adapted to account for this new situation. In principle one can add the additional constraint

$$\bar{u}(\tau) = u^*(\tau; x(t_{i-1})) \ \tau \in [t_i, t_i + \bar{\delta}^c) \qquad (8)$$

to (3) or one can use $u^*(\tau; x(t_{i-1})) \ \tau \in [t_i, t_i + \bar{\delta}^c)$ to predict $x(t_i + \bar{\delta}^c)$ and solve the open-loop optimal control problem for this "initial" state. For simplicity of notation we follow the first approach. For this reason we require additionally that $T_p \geq \bar{\pi} + \bar{\delta}^c$, i.e. the prediction horizon is long enough to at least span to $t_{i+1} + \bar{\delta}^c$. The resulting open-loop optimal control problem that is solved at every recalculation instant t_i is give by

$$\min_{\bar{u}(\cdot)} J(\bar{u}(\cdot), x(t_i)) \qquad (9a)$$

$$\text{s.t. } \dot{\bar{x}} = f(\bar{x}, \bar{u}), \quad \bar{x}(t) = x(t_i) \qquad (9b)$$

$$\bar{u}(\tau) = u^*(\tau; x(t_{i-1})), \tau \in (t_i, t_i + \bar{\delta}^c] \qquad (9c)$$

$$\bar{u}(\tau) \in \mathcal{U}, \ \bar{x}(\tau) \in \mathcal{X}, \ \tau \in [t_i, t_i + T_p], \qquad (9d)$$

$$\bar{x}(t_i + T_p) \in \mathcal{E}, \qquad (9e)$$

with J given by (3e). Note that the notation $u^*(\tau; x(t_i), t_i, t_i + T_p)$ is not totally correct. The optimal input u^* now also depends on the input at t_{i-1}.

In the following we state a theorem establishing conditions for stability of the closed-loop. The theorem is along the lines of the results in (Fontes 2000, Chen and Allgöwer 1998), which do not consider the computational delay.

Theorem 3.1 (Stability of sampled-data NMPC considering computational delay) *Suppose there exists a set* $\mathcal{E}$ *and a terminal penalty* E *such that*

(a) $E \in C^1$ *and* $E(0) = 0$,
(b) $\mathcal{E} \subseteq \mathcal{X}$ *is closed and connected with the origin contained in* $\mathcal{E}$,
(c) $\forall x \in \mathcal{E}$ *there exists a input* $u_{\mathcal{E}} : [0, \bar{\pi}] \to \mathcal{U}$ *such that* $x(\tau) \in \mathcal{E}$, $\forall \tau \in [0, \bar{\pi}]$ *and*

$$\frac{\partial E}{\partial x} f(x(\tau), u_{\mathcal{E}}(\tau)) + F(x(\tau), u_{\mathcal{E}}(\tau)) \le 0 \qquad (10)$$

(d) the NMPC open-loop optimal control problem has a feasible solution for t_0.

Then the state of the nominal closed-loop system defined by (9), (8), *and* (3e) *converges to the origin for all partitions* π *that satisfy* $\underline{\pi} \ge \bar{\delta}^c$, $T_p \ge \bar{\pi} + \bar{\delta}^c$. *Furthermore, the region of attraction* $\mathcal{R}$ *is given by the set of states for which the open-loop optimal control problem* (9) *has a solution.*

Note that we achieve stability in the sense of convergence to the origin (=steady state).

Proof.

As usual in predictive control the proof consists of two parts: a feasibility part and a convergence part.

Feasibility: Take any time t_i for which a solution exists (e.g. t_0). After solving the open-loop optimal control problem, the optimal input $u^*(\tau; x(t_i))$ corresponding to $x(t_i)$ is implemented for $\tau \in (t_i + \bar{\delta}^c, t_{i+1} + \bar{\delta}^c]$. Since we assume no model plant mismatch and since the open-loop input from the previous recalculation, which is applied during the solution of (9) is taken into account, the predicted open-loop state $\bar{x}(t_{i+1})$ at t_{i+1} coincides with $x(t_{i+1})$. Thus, the remaining piece of the optimal input $u^*(\tau; x(t_i))$, $\tau \in [t_{i+1}, t_i + T_p]$ satisfies the state and input constraints if "applied" to (9b), and $\bar{x}(t_i + T_p; x(t_i), u^*(\tau; x(t_i))) \in \mathcal{E}$. According to Theorem 3.1 (c) $\mathcal{E}$ and E are chosen such that for every $x(t) \in \mathcal{E}$ there exists at least one input $u_{\mathcal{E}}(\cdot)$ that renders $\mathcal{E}$ invariant over $\bar{\pi}$. Consider the following input candidate for t_{i+1},

$$\tilde{u}(\tau) = \begin{cases} u^*(\tau; x(t_i)), \ \tau \in [t_{i+1}, t_i + T_p] \\ u_{\mathcal{E}}(\tau), \tau \in (t_i + T_p, t_{i+1} + T_p] \end{cases} \qquad (11)$$

which is a concatenation of the remaining old input and $u_{\mathcal{E}}(\cdot)$. This input satisfies all constraints and leads to $x(t_{i+1} + T_p; x(t_{i+1}), \tilde{u}(\cdot)) \in \mathcal{E}$. Thus, feasibility at time t_i implies feasibility at t_{i+1}, i.e. if the open-loop optimal control problem has a solution for t_0 it also has a solution afterwards. Furthermore, if one can show that the states for which (9) has a (initial) solution converge to the origin, it is clear that the region of attraction $\mathcal{R}$ consists of the points for which (9) posses a solution. This is established in the next part of the proof.

Convergence: We denote the optimal cost at every recalculation instant t_i as value function $V(x(t_i)) = J^*(u^*(\cdot, x(t_i))$. We show that the value function is strictly decreasing. This allows to establish convergence of the state to the origin. Remember that the value of V at the recalculation instant t_i is given by:

$$\begin{aligned} V(x(t_i)) = &\int_{t_i}^{t_i+T_p} F(\bar{x}(\tau; x(t_i), u^*(\cdot; x(t_i))), u^*(\tau; x(t_i)) d\tau \\ &+ E(\bar{x}(t_i + T_p; x(t_i), u^*(\cdot; x(t_i))). \end{aligned}$$

Consider now the cost resulting from the application of $\tilde{u}$ a starting from $x(t_{i+1})$:

$$\begin{aligned} J(\tilde{u}(\cdot), x(t_{i+1})) = &\int_{t_{i+1}}^{t_{i+1}+T_p} F(\bar{x}(\tau; x(t_{i+1}), \tilde{u}(\cdot)), \tilde{u}(\tau)) d\tau \\ &+ E(\bar{x}(t_{i+1} + T_p; x(t_{i+1}), \tilde{u}(\cdot))). \end{aligned}$$

Reformulating yields

$$\begin{aligned} J(\tilde{u}(\cdot), x(t_{i+1})) = &V(x(t_i)) \\ &- \int_{t_i}^{t_{i+1}} F(\bar{x}(\tau; x(t_i), u^*(\cdot; x(t_i))), u^*(\tau; x(t_i))) d\tau \\ &- E(\bar{x}(t_i + T_p; x(t_i), u^*(\cdot; x(t_i)))) \\ &+ \int_{t_i+T_p}^{t_{i+1}+T_p} F(\bar{x}(\tau; x(t_{i+1}), \tilde{u}(\cdot)), \tilde{u}(\tau)) d\tau \\ &+ E(\bar{x}(t_{i+1} + T_p; x(t_{i+1}), \tilde{u}(\cdot))) \end{aligned}$$

Integrating inequality (10) over $\tau \in [t_i + T_p, t_{i+1} + T_p]$ we can upper bound the last three terms by zero. Thus, we obtain

$$\begin{aligned} &V(x(t_i)) - J(\tilde{u}(\cdot), x(t_{i+1})) \\ &\le -\int_{t_i}^{t_{i+1}} F(\bar{x}(\tau; x(t_i), u^*(\cdot; x(t_i))), u^*(\tau; x(t_i))) d\tau \end{aligned}$$

Since $\tilde{u}$ is only a feasible, but not the optimal input for $x(t_{i+1})$ it follows that

$$\begin{aligned} &V(x(t_i)) - V(x(t_{i+1})) \\ &\le -\int_{t_i}^{t_{i+1}} F(\bar{x}(\tau; x(t_i), u^*(\cdot; x(t_i))), u^*(\tau; x(t_i))) d\tau \quad (12) \end{aligned}$$

This establishes that for any partition with $\underline{\pi} \ge \bar{\delta}^c$ (the time between two recalculations is sufficiently long to allow the solution of the open-loop optimal control problem) and with $T_p \ge \bar{\pi} + \bar{\delta}^c$ (the prediction horizon spans sufficiently long into the future) the value function is decreasing. Since the decrease in (12) is strictly positive for $(x, u) \neq (0, 0)$ it is possible, similar to (Fontes 2000, Chen and Allgöwer 1998), to employ a variant of Barbalat's lemma to establish that the states converge to the origin for $t \to \infty$. ■

4. DISCUSSION

The conditions for stability in Theorem 3.1 are, similar to the results in (Fontes 2000), rather general. We do not give specific details on how to obtain a suitable terminal region or terminal penalty term, since most NMPC approaches with guaranteed stability and do not take the computational delay into account can be simply adapted. Examples for suitable approaches are the zero terminal constraint approach (Mayne and Michalska 1990), quasi-infinite horizon NMPC (Chen and Allgöwer 1998), control Lyapunov function based approaches (Jadbabaie *et al.* 2001), and the so called simulation approximated infinite horizon NMPC approach (De Nicolao *et al.* 1998).

In comparison to the scheme presented in (Chen *et al.* 2000) that takes the computational delay into account, the outlined approach is applicable to a wider class of systems and does not require to consider a quadratic cost function. Furthermore, the presented conditions even allow to design NMPC controllers that can stabilize systems which can not be stabilized by a feedback that is continuous in the state, compare (Fontes 2000).
The key reason for including the computation time into the open-loop optimal control problem is that if it is neglected, it is strictly not possible to establish stability, as also shown in the example in the next section. Only if the delay due to the numerical solution is sufficiently small, it can be consider it as an disturbance which NMPC under certain conditions is able to handle (Findeisen *et al.* 2003).

5. EXAMPLE

To illustrate the outlined method and the general influence of a neglected computational delay, we consider the control of classical continuous stirred tank reactor (CSTR), for the exothermic, irreversible reaction $A \rightarrow B$ as outlined in. The model under the assumption of constant liquid volume takes the following form (Henson and Seborg 1997):

$$\dot{c}_A = \frac{q}{V}(c_{Af} - c_A) - k_0 e^{\frac{-E}{RT}} c_A$$
$$\dot{T} = \frac{q}{V}(T_f - T) + \frac{-\Delta H}{\rho C_p} k_0 e^{\frac{-E}{RT}} c_A + \frac{UA}{V \rho C_p}(T_c - T),$$

where UA, q, V, c_{Af}, E, RT, ρ, k_0, $-\Delta H$, C_p are constants, c_A is the concentration of substance A, T is the reactor temperature, and T_c is the manipulate variable – the coolant stream. The objective is to stabilize the operating point $T_s = 375K$, $c_{As} = 0.159\text{mol/L}$ via the coolant stream T_c ($T_{cs} = 302.84K$), where T_c is limited to the interval $[220\text{K}, 330\text{K}]$. As NMPC method quasi-infinite horizon NMPC is applied. The terminal penalty term E and the terminal region $\mathcal{E}$ are obtained considering the quadratic "stage cost" $F = x^{\mathrm{T}} \left[\begin{smallmatrix} 4 & 0 \\ 0 & 4 \end{smallmatrix}\right] x + 2(T - T_{cs})^2$, where $x = \left[\begin{smallmatrix} c_a - c_{As} \\ T - T_s \end{smallmatrix}\right]$, using the direct semi-infinite optimization approach as outlined in (Chen and Allgöwer 1998). For simplicity we assume that the recalculation instants are equally apart, i.e. $t_i = i\delta^r$, where $\delta^r = 0.15\text{min}$. Furthermore we assume, that the maximum required solution time $\bar{\delta}^c$ coincides with the recalculation time, i.e. $\bar{\delta}^c = \delta^r$. The prediction horizon horizon is set to $T_p = 3\text{min}$. The open-loop optimal control problem is solved using a direct optimization method (see e.g. (Biegler and Rawlings 1991)) that is implemented in Matlab. For this purpose the input signal is parametrized as piecewise constant with a sampling time that also coincides with the time between the recalculation instants, i.e. $\delta^s = \delta^r$. Figure 2 and 3 show the simulation result for the initial condition $c_A(0) = 0.5\text{mol/L}$ and $T(0) = 350\text{K}$. Shown are

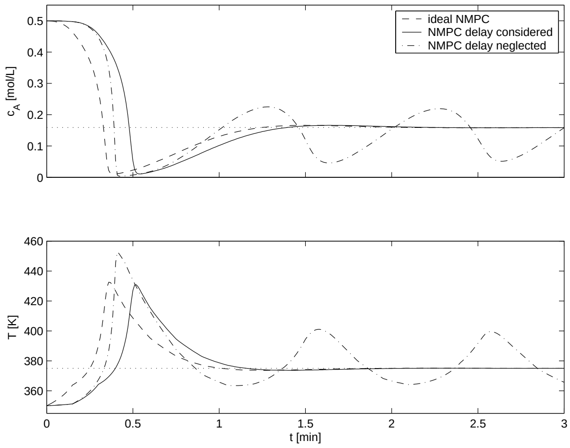

Fig. 2. Resulting states considering an ideal NMPC controller, an NMPC controller that does neglect the delay, and an NMPC controller that accounts for the delay.

the results for an ideal (theoretical) NMPC controller (ideal NMPC), i.e. assuming that the optimal control problem can be solved immediately, an NMPC controller in which the computational delay is not taken into account (NMPC delay neglected), and the scheme outlined in Section 3 (NMPC delay considered). As expected the best performance is achieved for the ideal NMPC controller (which can not be implemented in practice). The more realistic setups, in which a delay occurs, show degraded performance. Clearly it can be seen, that if the delay is not taken into account, that the performance degrades dramatically (curve NMPC delay neglected), i.e. no convergence to the desired steady state is achieved, even so that the delay of 0.15min is rather small. This is also clearly visible in implemented input as shown in Figure 3. Notably,

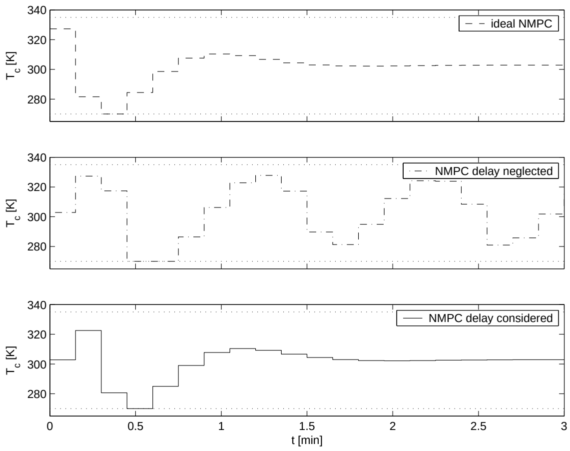

Fig. 3. Resulting input signals.

the NMPC controller that takes the delay into account, achieves very similar performance if compared to the ideal NMPC controller. The remaining difference is mainly due to the initial delay time up to $t = 0.15\text{min}$, in which the old steady state input is applied to the system. Overall it becomes clear, how important the correct consideration of the never avoidable computational delay for stability and good performance.

6. CONCLUSIONS

In this paper we considered the sampled-data NMPC of continuous time systems taking the necessary solution time of the open-loop optimal control problem directly into account. As shown, if the computational delay is not taken into account, the performance of the closed-loop can degrade or even instability can occur. In the approach we outlined, the open-loop input from the previous recalculation is applied until the solution of the optimal control problem is available. Since the "old" input is also taken into account in the open-loop optimal control problem the predicted open-loop trajectory and the closed-loop trajectory coincide in the nominal case. Based on this and suitable assumptions on the terminal region constraint and terminal penalty term, we outlined conditions under which the closed-loop is stable. The assumption on the terminal region constraint and the terminal penalty term are rather general and allow to obtain suitable candidates using various methods such as quasi-infinite horizon NMPC. Overall, the outlined method allows to employ NMPC even in the case that the solution time of the optimal control problem is not negligible.

7. REFERENCES

Allgöwer, F., T.A. Badgwell, J.S. Qin, J.B. Rawlings and S.J. Wright (1999). Nonlinear predictive control and moving horizon estimation – An introductory overview. In: *Advances in Control, Highlights of ECC'99* (P.M. Frank, Ed.). pp. 391–449. Springer.

Bartlett, R.A., A. Wächter and L.T. Biegler (2000). Active set vs. inerior point strategies for model predicitve control. In: *Proc. Amer. Contr. Conf.*. Chicago, Il. pp. 4229–4233.

Biegler, L.T. and J.B Rawlings (1991). Optimization approaches to nonlinear model predictive control. In: *Proc. 4th International Conference on Chemical Process Control - CPC IV* (W.H. Ray and Y. Arkun, Eds.). pp. 543–571. AIChE, CACHE.

Chen, H. and F. Allgöwer (1998). A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability. *Automatica* **34**(10), 1205–1218.

Chen, W., D.J. Ballance and J. O'Reilly (2000). Model predictive control of nonlinear systems: Computational delay and stability. *IEE Proceedings, Part D* **147**(4), 387–394.

De Nicolao, G., L. Magni and R. Scattolini (1998). Stabilizing receding-horizon control of nonlinear time varyng systems. *IEEE Trans. Automat. Contr.* **43**(5), 1030–1036.

de Oliveira Kothare, S. and M. Morari (2000). Contractive model predictive control for constrained nonlinear systems. *IEEE Trans. Automat. Contr.* **45**(6), 1053–1071.

Diehl, M., R. Findeisen, S. Schwarzkopf, I. Uslu, F. Allgöwer and J.P. Bock, H.G. Schlöder (2002). An efficient approach for nonlinear model predictive control of large-scale systems. Part I: Description of the methodology. *Automatisierungstechnik* **12**, 557–567.

Findeisen, R., L. Imsland, F. Allgöwer and B.A. Foss (2003). Towards a sampled-data theory for nonlinear model predictive control. In: *New Trends in Nonlinear Dynamics and Control* (W. Kang, C. Borges and M. Xiao, Eds.). Lecture Notes in Control and Information Sciences. Springer-Verlag. New York.

Findeisen, R., M. Diehl, I. Uslu, S. Schwarzkopf, F. Allgöwer, H.G. Bock, J.P. Schlöder and Gilles (2002). Computation and performance assesment of nonlinear model predictive control. In: *Proc. 42th IEEE Conf. Decision Contr.*. Las Vegas, USA.

Fontes, F.A. (2000). A general framework to design stabilizing nonlinear model predictive controllers. *Syst. Contr. Lett.* **42**(2), 127–143.

Henson, M.A. and D.E. Seborg (1997). *Nonlinear Process Control*. Prentice Hall. Upper Saddle River, NJ.

Jadbabaie, A., J. Yu and J. Hauser (2001). Unconstrained receding horizon control of nonlinear systems. *IEEE Trans. Automat. Contr.* **46**(5), 776 –783.

Magni, L. and R. Scattolini (2002). State-feedback MPC with piecewise constant control for continuous-time systems. In: *Proc. 42th IEEE Conf. Decision Contr.*. Las Vegas, USA.

Mayne, D.Q. and H. Michalska (1990). Receding horizon control of nonlinear systems. *IEEE Trans. Automat. Contr.* **35**(7), 814–824.

Mayne, D.Q., J.B. Rawlings, C.V. Rao and P.O.M. Scokaert (2000). Constrained model predictive control: stability and optimality. *Automatica* **26**(6), 789–814.

Michalska, H. and D.Q. Mayne (1993). Robust receding horizon control of constrained nonlinear systems. *IEEE Trans. Automat. Contr.* **AC-38**(11), 1623–1633.

Rawlings, J. B. (2000). Tutorial overview of model predictive control. *IEEE Contr. Syst. Magazine* **20**(3), 38–52.

Tenny, M.J. and J.B. Rawlings (2001). Feasible real-time nonlinear model predictive control. In: *6th International Conference on Chemical Process Control – CPC VI*. AIChE Symposium Series.