2011 50th IEEE Conference on Decision and Control and
European Control Conference (CDC-ECC)
Orlando, FL, USA, December 12-15, 2011

Rigorous solution vs. fast update: Acceptable computational delay in NMPC

Inga J. Wolf, Lynn Würth and Wolfgang Marquardt

Abstract— **We present a method to improve the performance of nonlinear model predictive control (NMPC) by compromising between the time delay caused by a computational algorithm and the accuracy of the resulting control law in order to achieve best possible closed-loop performance. The main feature of the method is an a-priori error approximation derived for the neighboring-extremal update (NEU) algorithm, a fast NMPC algorithm presented recently by the authors. The error estimate provides the deviation of the current control trajectory from the (unknown) optimal control trajectory. The a-priori error estimator is incorporated in an on-line decision making process which simultaneously decides on the quality of the computed controls and the computational delay. In particular, the optimal number of QP iterations in an SQP strategy is determined on each horizon prior to the computation of the current control move.**

I. Introduction

Nonlinear model-predictive control (NMPC) is a control technique where nonlinear optimal control problems are repeatedly solved on-line on a moving horizon. If the optimal control problem relies on a large-scale system model or has to deal with very short sampling times, a compromise is inevitable to trade off solution accuracy and computational delay, which refers to the delayed availability of the updated controls. If the solution accuracy of the updated controls is to be improved for better control performance, the solution time of the optimal control problem increases. However, long computing times result in the delayed availability of the updated controls and thus decrease control performance because outdated controls are applied as long as the control update is not available. Thus, a criterion is required which addresses this trade-off.

If the computational delay τ_d is ignored during controller design the closed-loop control performance can decrease substantially and the stability of the system is endangered [1]. Consequently, any computational algorithm applied should explicitly consider computational delay even if it is very efficient and hence diminishes the problem in many cases. To this end, Findeisen and Allgöwer [1] suggested to account for the delay of updated controls by solving the optimal control problem for the current sampling time starting at a predicted initial state at the end of an estimated computing time. Because the control move of the previous horizon is applied during computing time, the predicted initial state can be determined by a forward simulation. In this way, control performance is improved and closed-loop stability is guaranteed under well defined conditions for a wider class of problems than those considered by Chen et. al. [2]. A more involved strategy reported in [3] explicitly accounts for the trade-off between solution accuracy and computing time in a framework monitoring the optimal control updating time.

In order to safely ignore the computational delay τ_d in NMPC design, the ratio $\eta = \tau_d/\Delta t$ needs to be much smaller than 1, i.e. the sampling time Δt, reflecting the system's dominant time constant, has to be much larger than the computational delay τ_d. In recent work, efficient computational schemes have been developed that aim to decrease η to values close to zero such that τ_d can be neglected even for systems with fast dynamics. These efficient schemes reduce computing time by either applying suboptimal solutions without iterating the nonlinear optimization problem to convergence (e.g. [4], [5]) or by calculating fast sensitivity-based updates as a function of parametric uncertainties [6]. For both strategies, good control performance is often achieved, particularly in case of process disturbances of small magnitude and for mildly nonlinear systems. However, since a loss in control performance may be observed for large disturbances (e.g. [7]), different extensions have been developed. For example, Zavala and Biegler [8] determine a rigorous solution for a predicted future plant state in each sampling time and compute a fast sensitivity-based update once current state measurements are available. However, if the rigorous solution cannot be determined during Δt for a very large-scale problem, the method might fail. In this case, extensions are still applicable which do not determine a rigorous solution but improve the control move of the fast schemes by additional QP iterations in an SQP strategy. In the context of linear MPC, Way and Boyd [9] observe good control quality after few iterations and heuristically set an iteration limit (3 - 5 iterations). In our own work on NMPC ([7], [10]), additional QP iterations are conducted until prespecified feasibility and optimality conditions are fulfilled. These extensions do not address the trade-off between the

Inga J. Wolf is with Chair of AVT - Process Systems Engineering, RWTH-Aachen University, D-52056 Aachen, Germany inga.wolf@avt.rwth-aachen.de

Lynn Würth is with Bayer Technology Services, 51373 Leverkusen, Germany lynn.wuerth@bayer.com

Wolfgang Marquardt is with Chair of AVT - Process Systems Engineering, RWTH-Aachen University, D-52056 Aachen, Germany wolfgang.marquardt@avt.rwth-aachen.de

978-1-61284-799-3/11/$26.00 ©2011 IEEE

number of QP iterations and the computational delay though the condition $\eta << 1$ might not hold.

Motivated by the deterioration of control performance due to computational delay and the shortcomings of available methods, we present an extended version of the neighboring-extremal update (NEU) algorithm [7], particularly suited for problems with non-negligible η, which explicitly addresses the trade-off between solution accuracy and computational delay. In this SQP-type algorithm as well as in the original NEU algorithm, fast sensitivity-based updates, the so-called NEU, are computed almost instantaneously and are directly applied to the process. In contrast to the original NEU algorithm, which does not account for this trade-off, the extended NEU algorithm determines the optimal number of additional QP iterations in a fast pre-optimization step.

The paper is structured as follows. In Section II, the receding horizon formulation of the NMPC problem is outlined and the original NEU algorithm is presented which provides an on-line approximate solution to the NMPC problem. Furthermore, an explicit expression is derived for the NEU in order to allow a straightforward evaluation of the error induced in the NEU algorithm. In Section III, an a-priori estimate of the deviation between the NEU and the (unknown) optimal control move is provided based on the explicit expression derived in Section II. The estimate is then used in the extended NEU algorithm to decide whether additional QP iterations improve the control performance. The paper concludes with an illustrative simulated case study to control a non-isothermal continuous stirred tank reactor (CSTR) with a Van de Vusse reaction in Section IV.

II. NEU algorithm for fast NMPC

A. NMPC problem formulation

If we assume that the process model can be described by a DAE system of index one, the continuous-time receding horizon formulation of the control problem can be stated as

$$\min_{u_j(t),x_j(t)} \quad \hat{\Phi}\left(x_j\left(t\right),u_j\left(t\right)\right) \tag{1a}$$

$$s.t. \quad 0 = f(\dot{x}_j(t), x_j(t), u_j(t), P_j), \tag{1b}$$

$$x(t_j^0) = \hat{x}_j^0, \tag{1c}$$

$$0 \geq h(x_j(t), u_j(t), P_j), \tag{1d}$$

$$0 \geq e(x_j(t_j^f), u_j(t_j^f), P_j), \tag{1e}$$

$$t \in \mathcal{I}_j := [t_j^0, t_j^f],\ t_j^f := t_j^0 + K\Delta t,\ K \in \mathbb{N}, \tag{1f}$$

$$t_j^0 := t_{j-1}^0 + \Delta t\ , \tag{1g}$$

where $u_j : \mathcal{I}_j \rightarrow \mathbb{R}^{n_u}$ and $x_j : \mathcal{I}_j \rightarrow \mathbb{R}^{n_x}$ represent the control trajectory and the state trajectory on time horizon $\mathcal{I}_j$, respectively. $P_j \in \mathbb{R}^{n_p}$ is a parameter vector. The mapping $\hat{\Phi}(\cdot)$ represents the objective function of the optimal control problem which is subject to the process model (1b) with consistent initial conditions (1c), input or path constraints (1d) and end point constraints (1e). At each sampling instant t_j^0, the optimal control problem (1) is solved on the horizon $\mathcal{I}_j$ with the number of samples K and the sampling time Δt. We assume that a feasible solution exists for the current parameter vector P_j at each sampling instant t_j^0. For the sake of simplicity, we consider state feedback, though an extension to output feedback via state estimation is straightforward. One of the computationally efficient schemes developed recently to solve (1) is the NEU algorithm which is outlined briefly in the next section since this NMPC algorithm is the basis for the extension in Section III.

B. Neighboring-extremal update algorithm

First, each control $u_j^i(t)$, $i = 1...n_u$, is parameterized via $u_j^i(t) \approx \sum_{l=1}^{K} \left(z_{(i+n_u(l-1))}\right)_j (\Omega_l)_j(t)$ where $z_j \in \mathbb{R}^{n_z}$ is the parameterized control vector of dimension $n_z = n_u \cdot K$ and $(\Omega_l)_j(t)$ represents a piecewise constant B-spline basis function. Since (1) does not comprise any disturbances, and since we consider state feedback and assume that there is no model-plant mismatch within one sampling time, the optimal control problems (1) of successive horizons differ by parameters $p_j = P_j$. In general, p_j is an uncertainty vector which also comprises parameterized process disturbances and the initial conditions resulting from state estimation [7]. In this way, the optimal control problem (1) is transcribed into the nonlinear programming problem

$$\begin{aligned} \min_{z_j} \quad & \Phi(z_j, p_j) \\ s.t. \quad & g_k(z_j, p_j) = 0 \quad \forall\ k \in E, \\ & g_k(z_j, p_j) \leq 0 \quad \forall\ k \in I, \end{aligned} \tag{2}$$

where $g_k, k \in E$, are the equality constraints and $g_k, k \in I$, are the inequality constraints resulting after control parameterization from (1d) and (1e), respectively. For given p_j, (2) is a nonlinear program which has to be solved on every horizon $\mathcal{I}_j$ using, for example, an SQP strategy. The objective function, the constraints and their gradients with respect to z_j and p_j are computed by simultaneous integration of the model (1b) with initial conditions (1c) and the associated sensitivity equation system (see, e.g., [11] for an efficient algorithm).

Assumption 1. $\Phi(\cdot)$ and $g(\cdot) = \{g_i(\cdot) | i \in (E \cup I)\}$ are twice continuously differentiable functions and the linear independence constraint qualification (LICQ) [12, p. 328] as well as the strong second-order sufficient conditions (SSC) of optimality [12, p. 345] are fulfilled.

Under Assumption 1, the following quadratic program provides a fast NEU z_{j+1}^{NEU}, which approximates the solution of (2) in the presence of perturbations $\Delta p_j = p_{j+1} - p_j$ [10]:

$$\begin{aligned} \min_{\Delta z_j} \quad & \phi_{j+1} \overset{def}{=} \frac{1}{2}\Delta z_j^T L_{zz,j} \Delta z_j + \Delta p_j^T L_{pz,j} \Delta z_j + \Phi_z^T \Delta z_j \\ s.t. \quad & g_j + g_{z,j}\Delta z_j + g_{p,j}\Delta p_j \leq 0, \end{aligned} \tag{3}$$

where $\Delta z_j = z_{j+1}^{NEU} - z_j$ refers to the control update. $L(z_j, p_j, \lambda_j) = \Phi(z_j, p_j) + \lambda_j^T g(z_j, p_j)$ denotes the Lagrange function and $\lambda_j \in \mathbb{R}^{n_g}$ the Lagrange multipliers. The notation $(\cdot)_v$ and $(\cdot)_{vw}$ represent first and second-order derivatives $\frac{\partial(\cdot)}{\partial v}$ and $\frac{\partial^2(\cdot)}{\partial v \partial w}$. In the NEU algorithm, feasibility and optimality criteria (FOC) are checked for each update Δz_j in order to evaluate the performance of the approximated control move. If the FOC criteria do not hold additional QP iterations are conducted until the FOC are fulfilled.

The basic version of the NEU algorithm is as follows

1) set counter $j := 0$
2) compute the optimal solution z_0^* of (1) on $[t_1^0, t_1^f]$ for nominal parameters p_0 and efficiently computable second-order derivatives $L_{zz,0}$ and $L_{zp,0}$ (cf. [13])
3) for $j = 0, N_j$ do
   a) determine p_{j+1} and compute Δp_j
   b) compute NEU z_{j+1}^{NEU} using (3) with Δp_j and set counter $q_j := 0$
   c) while (FOC false) do
      i) conduct QP iteration
      ii) set $q_j = q_j + 1$
   d) end while;
   e) send updated controls z_j^q to system
   f) shift horizon, i.e. $t_{j+2}^0 = t_{j+1}^0 + \Delta t$ and $t_{j+2}^f = t_{j+1}^f + \Delta t$, and compute $L_{zz,j+1}$ and $L_{zp,j+1}$
   g) set $j = j + 1$
4) end for.

Since (3) approximates the control vector on every horizon $\mathcal{I}_{j+1}$, the error propagation occurs from one horizon to the next. It can be substantial for optimal control problems with high curvature $L_{zz,j+1}$ and $L_{pz,j+1}$ as well as for large perturbations Δp_j. We now derive an explicit formula for the NEU from (3), such that the deviation of z_{j+1}^{NEU} from z_{j+1}^* can be estimated from the a-priori error approximation derived in Section III-A.

C. Neighboring-extremal update

In this work, we pragmatically state

Assumption 2. The active set does not change from $\mathcal{I}_j$ to $\mathcal{I}_{j+1}$ and the strict complementarity conditions hold.

The more general case of changing active sets from horizon to horizon will be treated in future work. The Lagrange function of (3) is then given by $\mathcal{L}_{j+1} = \phi_{j+1} + \lambda_{j+1}^T(g_j^a + g_{z,j}^a \Delta z + g_{p,j}^a \Delta p_j)$, where g_j^a represents all constraints which belong to the active set $\mathcal{G}_j^a = E \cup \{k \in I | g_{k,j} = 0\}$ with $g_j^a = \{g_{k,j} | k \in \mathcal{G}_j^a\}$. The optimal solution of (3) $(\Delta z^T, \Delta\lambda^T)_j^T$ with $\Delta\lambda_j = (\lambda_{j+1} - \lambda_j)$ can be computed from the necessary conditions of optimality (NCO)

$$\begin{aligned} &\mathcal{L}_{\Delta z, j+1} = L_{zz,j}\Delta z_j + L_{zp,j}\Delta p_j + \Phi_{z,j} + g_{z,j}^{aT}\lambda_{j+1} = 0, \\ &g_j^a + g_{z,j}^a \Delta z_j + g_{p,j}^a \Delta p_j = 0. \end{aligned} \tag{4}$$

Since the error is propagated from horizon to horizon, a non-optimal control update is computed and the residuals of the NCO of (2) are not fulfilled. Introducing the residual errors $\mathcal{E}_{L,j}$ and $\mathcal{E}_{G,j}$, we can write

$$\begin{aligned} L_{z,j} = \Phi_{z,j} + g_{z,j}^{aT}\lambda_j &= \mathcal{E}_{L,j}, \\ g_j^a &= \mathcal{E}_{G,j}. \end{aligned} \tag{5}$$

If $\Phi_{z,j}$ and g_j^a in (4) are eliminated by (5), the optimal solution of (3) can be computed from the linear system

$$\begin{pmatrix} \Delta z \\ \Delta\lambda \end{pmatrix}_j = \underbrace{-\mathcal{M}_j^{-1} \begin{pmatrix} L_{zp} \\ g_p^a \end{pmatrix}_j}_{\frac{d(z_j^T, \lambda_j^T)^T}{dp_j}} \Delta p_j \quad \underbrace{-\mathcal{M}_j^{-1} \begin{pmatrix} \mathcal{E}_L \\ \mathcal{E}_G \end{pmatrix}_j}_{(e_j^{shift\,T}, \Delta\lambda_j^{shift\,T})^T}, \tag{6}$$

where $\mathcal{M} = [L_{zz}, g_z^{aT}; g_z^a, 0]$ is the KKT matrix. The resulting NEU $z_{j+1}^{NEU} = \Delta z_j + z_j$ is composed of a first-order term of a truncated Taylor series based on the first-order sensitivity theorem [14] and an error-induced shift e_j^{shift} which is independent of p_{j+1} and equal to zero if the NCO of (2) hold exactly (cf. [7]). Diehl et. al. [15] refer to (6) as the approximate tangential predictor.

The NEU algorithm is extended next by a decision criterion which addresses the trade-off between solution accuracy and computational delay in order to achieve the best possible control performance.

III. Computational delay adaptation for NEU algorithm

In the NEU algorithm, the computational delay is largely due to the computation of second-order derivatives in step (3f), while the delay for retrieving the NEU in step (3a) is small because $\mathcal{M}^{-1}$, $(L_{zp}^T, g_p^{aT})^T$ and e_j^{shift} can be computed on the previous horizon. We therefore state:

Assumption 3. The NEU in step (3a) is instantaneous.

In order to assess the solution accuracy of the successive NEU for horizon $\mathcal{I}_{j+1}$, the main feature of the method, an a-priori error approximation, is derived. To this end, the deviation between the unknown optimal control trajectory z_{j+1}^* and the control trajectory computed by the NEU z_{j+1}^{NEU} is estimated with respect to the perturbation Δp_j in Section III-A. In Section III-B, the approximated solution accuracy is used to decide whether more computational delay is acceptable, i.e. whether the control performance is improved by additional QP iterations.

A. Error approximation for the extended NEU algorithm

According to Bellman's principle of optimality [18], the optimal trajectory does not change for the transition from one horizon to the next for a shrinking horizon setting, when neither disturbances nor model-plant mismatch exist. We pragmatically state

Assumption 4. Though Bellman's principle of optimality is not valid for a finite moving horizon setting, we neglect the error induced.

Thus, the total approximation error accumulates from one horizon to the next due to consecutive approximations by NEU. We derive this error, $e_{j+1}^{prop} = z_{j+1}^* - z_{j+1}^{NEU}$, by an error propagation starting from the known optimal

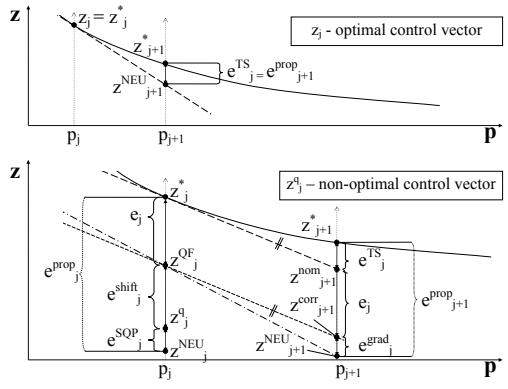

Fig. 1. Error contributions for one z and one p. Solid line - optimal solution trajectory. Dashed line - $\frac{dz_j^*}{dp_j}$. Dash-dotted line - $\frac{dz_j}{dp_j}$.

solution z_0^*, where $e_0^{prop} = 0$. The error propagation is initialized every time a new setpoint z_0^* is determined by an upper level dynamic economic optimization (cf. [17]).

Fig. 1 illustrates all contributions to e_{j+1}^{prop} for a single control parameter z with respect to a single parameter p.

For an optimal control vector $z_j = z_j^*$, such as the initial optimal solution z_0^*, the NEU (6) comprises only the first-order term of a truncated Taylor series, $(\Delta z^T, \Delta\lambda^T)_j^T = \frac{d(z_j^T, \lambda_j^T)^T}{dp_j}\Delta p_j$ (cf. top of Fig. 1), while the second term equals zero because the NCO are fulfilled (cf. (6)). Thus, the only error contribution for the successive horizon $\mathcal{I}_{j+1}$ results from the truncation of second and higher-order terms (h.o.t.)

$$e_j^{TS} = e_j^{2nd} + h.o.t.\,, \tag{7}$$

where $e_j^{2nd} = \frac{1}{2}\Delta p_j^T \frac{d^2 z_j}{dp_j^2}\Delta p_j$. Note that $\frac{d^2 z_j}{dp_j^2}$ corresponds to the optimal second-order derivative, since $z_j = z_j^*$ for p_j. Once Δp_j is available on horizon $\mathcal{I}_{j+1}$, e_j^{2nd} is determined instantaneously because the time-consuming second-order derivative $\frac{d^2 z_j}{dp_j^2}$ can be computed in advance on horizon $\mathcal{I}_j$. The total error is given by $e_{j+1}^{prop} = e_j^{TS} \approx e_j^{2nd}$. The optimal control vector can be approximated by $z_{j+1}^* \approx z_{j+1}^{NEU} + e_j^{2nd}$, where z_{j+1}^{NEU} is the NEU on horizon $\mathcal{I}_{j+1}$.

For a non-optimal control vector z_j^q, several other contributions to the accumulated error have to be considered (cf. bottom of Fig. 1). To this end, a recursive function for e_{j+1}^{prop} will be developed.

Current error. The propagated error e_j^{prop} on the current horizon $\mathcal{I}_j$ depends on all previous Δp_l, $l = 0...j-1$,

$$e_j^{prop} = z_j^*(p_j) - z_j^{NEU}(p_j). \tag{8}$$

This error contribution starts from the initial optimal solution z_0^*, $e_0^{prop} = 0$.

Additional QP iterations on horizon $\mathcal{I}_j$. We also have to consider that additional QP iterations may have been performed on horizon $\mathcal{I}_j$ to reduce the total error:

$$e_j^{SQP} = z_j^q - z_j^{NEU}, \tag{9}$$

where z_j^q is the current control update retrieved after $q_j \geq 0$ additional QP iterations on horizon $\mathcal{I}_j$. This error contribution can be computed exactly on horizon $\mathcal{I}_j$ since z_j^q and $z_j^{NEU} = z_j$ are given.

Error-induced shift. The error-induced shift $e_j^{shift} = z_j^{QF} - z_j^q$ (cf. (6)) is independent of the perturbation p_{j+1} and can exactly be determined in advance because $(\mathcal{E}_L^T, \mathcal{E}_G^T)_j^T$ can be computed from (5) on horizon $\mathcal{I}_j$.

In the bottom of Fig. 1, e_j denotes the remaining deviation from the optimal control vector z_j^* after the contributions e_j^{SQP} and e_j^{shift} are considered. If $p_{j+1} = p_j$, $z_{j+1}^{NEU} = z_j^{QF}$ and $e_{j+1}^{prop} = e_j$. If $p_{j+1} \neq p_j$, we have to consider additional error contributions.

Non-optimal sensitivities. If the sensitivity $\frac{dz_j}{dp_j}$ equals $\frac{dz_j^*}{dp_j}$, the NEU will be z_{j+1}^{corr} and the deviation from the NEU z_{j+1}^{nom} computed around the optimal solution z_j^* will be $z_{j+1}^{nom} - z_{j+1}^{corr} = e_j$ (cf. bottom of Fig. 1). However, $\frac{dz_j}{dp_j} \neq \frac{dz_j^*}{dp_j}$ for a general non-optimal z_j. Thus, the bigger Δp_j, the more the control vectors $z_{j+1}^{corr}(\frac{dz_j^*}{dp_j})$ and $z_{j+1}^{NEU}(\frac{dz_j}{dp_j})$ differ. The induced error is

$$\begin{aligned} e_j^{grad} &= \left(\frac{dz_j^*}{dp_j} - \frac{dz_j}{dp_j}\right)\Delta p_j = z_{j+1}^{corr} - z_{j+1}^{NEU} \\ &\approx \left(\frac{d(z_j + e_j^{prop})}{dp_j} - \frac{dz_j}{dp_j}\right)\Delta p_j = e_j^{1st}. \end{aligned} \tag{10}$$

The optimal first-order sensitivity $\frac{dz_j^*}{dp_j}$ is approximated by $\frac{d(z_j + e_j^{prop})}{dp_j}$ and can already be computed on horizon $\mathcal{I}_j$.

Truncation of Taylor series. The error e_j^{TS} caused by the truncation of the Taylor series after the first-order term is given by (7) and estimated by e_j^{2nd}. In Fig. 1, $z_{j+1}^{nom} = z_{j+1}^* - e_j^{TS}$.

Error propagation for successive horizon $\mathcal{I}_{j+1}$. Having derived the five error contributions, the total error e_{j+1}^{prop} on the successive horizon $\mathcal{I}_{j+1}$ can be now expressed by the following recursive formulation

$$\begin{aligned} e_{j+1}^{prop} &= e_j^{prop} - e_j^{SQP} - e_j^{shift} + e_j^{TS} + e_j^{grad} \\ &= e_j + z_{j+1}^* - z_{j+1}^{nom} + z_{j+1}^{corr} - z_{j+1}^{NEU} = z_{j+1}^* - z_{j+1}^{NEU}, \end{aligned} \tag{11}$$

which can be approximated by

$$e_{j+1}^{est} = e_j^{est} - e_j^{SQP} - e_j^{shift} + e_j^{2nd} + e_j^{1st} \approx e_{j+1}^{prop}, \tag{12}$$

where $e_0^{est} = 0$.

We have derived an error estimator e_{j+1}^{est} for NEU algorithms with and without additional QP iterations which can be computed on horizon $\mathcal{I}_j$ and which also allows us to derive the extended NEU algorithm in Section III-B.

B. The extended NEU algorithm

In the extended NEU algorithm, a decision criterion $\mathcal{K}_j$ is introduced which addresses the trade-off between solution accuracy and computational delay. $\mathcal{K}_j(q_j) : D \subset \mathbb{N}_0 \rightarrow \mathbb{R}$ represents the deviation between the approximated optimal control vector for q_j additional QP iterations on horizon $\mathcal{I}_j$. Since the additional QP iterations and the second-order derivatives of the successive NEU have to be computed within Δt, $D = \{q_j | 0 \leq q_j \leq q_{max,j}\}$ can be determined in advance by calculating the maximal number of iterations, $q_{max,j} = \left\lfloor (\Delta t - \tau_{d,j}^{sens})(\tau_{d,j}^{QP})^{-1} \right\rfloor$, where $\tau_{d,j}^{QP}$ represents the computing time for one QP iteration and $\tau_{d,j}^{sens}$ the computing time for the second-order derivatives of the NEU. The best control performance is achieved if $\mathcal{K}_j$ is minimized with respect to q_j in a fast integer pre-optimization:

$$\min_{q_j \in D} \quad \mathcal{K}_j(q_j) \stackrel{def}{=} \left\| z_j^* - z_j^q \right\|_1, \tag{13}$$

where z_j^* denotes the optimal but usually unknown control vector and z_j^q the control update retrieved after q_j QP iterations. In (13), the 1-norm is used, since equal weighting of all elements of $z_j^* - z_j^q$ is desired. Though the deviation from the (unknown) optimal objective function value is often used as a criterion to assess non-optimality (e.g. [3]), the decision vector of the extended NEU algorithm is based on the deviation from the (unknown) optimal control trajectory, since the optimal control trajectory can be tracked more reliably, in case the deviation from the optimal objective function value is small but the control structure differs clearly from the optimal one.

In order to include the computational delay $\tau_{d,j} = q_j \cdot \tau_{d,j}^{QP} < \Delta t$ in the decision criterion, the control vector z_j is split into two subvectors, $z_j = (\sigma^T, \kappa^T)_j^T$. Here, σ_j is the subvector containing exactly the first parameterized control of each control variable u_j^i applied on $[t_j^0, t_j^0 + \Delta t)$ (cf. control parameterization in Section II-B), whereas κ_j contains the remaining adjustable elements $n_z - n_u$. Thus, $(\sigma^T, \kappa^T)_j^T = ((z_1, ..., z_{n_u}), (z_{n_u+1}, ..., z_{n_u \cdot K}))_j^T$. During $[t_j^0, t_j^0 + \tau_{d,j})$, the subvector of the fast NEU σ_j^{NEU} is applied to the process until σ_j^q is available at $t_j^0 + \tau_{d,j}$. Thus, $\mathcal{K}_j(q_j) = v_1 \left\| \sigma_j^* - \sigma_j^{NEU} \right\|_1 + v_2 \left\| \sigma_j^* - \sigma_j^q \right\|_1 + \left\| \kappa_j^* - \kappa_j^q \right\|_1$, where $v_1 = \frac{\tau_{d,j}}{\Delta t}$ and $v_2 = 1 - v_1$. A solver efficiency map E_j is identified on-line in order to describe the change in the deviation of the adjustable controls from the optimal trajectory with respect to q_j, i.e. $v_2 \left\| \sigma_j^* - \sigma_j^q \right\|_1 + \left\| \kappa_j^* - \kappa_j^q \right\|_1 = E_j \left(v_2 \left\| \sigma_j^* - \sigma_j^{NEU} \right\|_1 + \left\| \kappa_j^* - \kappa_j^{NEU} \right\|_1 \right)$. Since the solver efficiency cannot be determined before the actual optimization routine is performed, E_j is approximated by fitting data from optimizations on previous horizons to the structure

$$E_j = (\alpha_{j-1}^f \max(0, q_{j-1} - q_{j-1}^f) + 1)^{-1} \tag{14}$$

as suggested by Alamir [3], where α_{j-1}^f monitors the speed of convergence and q_{j-1}^f describes the number of iterations required before the deviation from the optimal control vector decreases. By definition, E_j is influenced by the system's nonlinearity and the number of degrees of freedom of the optimal control problem. If the parameter structure changes severely, no relation between the current and the previous E_j might exist. However, even if the optimal number of QP iterations is not determined correctly, the extended NEU algorithm still improves the performance in comparison to other schemes, since computational delay is incorporated into the algorithm. Including the efficiency map and the error estimator of Section III-A, the decision criterion is given by

$$\mathcal{K}_j(q_j) = v_1 \left\| e_{\sigma\, j}^{est} \right\|_1 + \left(v_2 \left\| e_{\sigma\, j}^{est} \right\|_1 + \left\| e_{\kappa\, j}^{est} \right\|_1 \right) E_j \tag{15}$$

with $e_{\sigma\, j}^{est} = (e_1^{est}, ..., e_{n_u}^{est})_j^T$ and $e_{\kappa\, j}^{est} = (e_{n_u+1}^{est}, ..., e_{n_u \cdot K}^{est})_j^T$.

IV. Case study

The performance of the method is demonstrated on a simulated Van de Vusse CSTR based on [16]:

$$\begin{aligned}
\frac{dx_1}{dt} &= -k_1 x_1 - k_2 x_1^2 + \left(x^{in} - x_1\right) \nu_1, \\
\frac{dx_2}{dt} &= k_1 \left(x_1 - x_2\right) + x_2 \nu_1, \\
\frac{dx_3}{dt} &= -\frac{\Delta H_1 k_1 x_1 + \Delta H_2 k_1 x_2 + \Delta H_3 k_2 \; {x_1}^2}{\rho C_p} \ldots \\
&\quad + \nu_2 + (p - x_3)\, \nu_1.
\end{aligned}$$

x_1 and x_2 are the species concentrations and x_3 is the reactor temperature. ν_1 represents the dilution rate and ν_2 the cooling rate. The rate coefficients are given by $k_i = k_{0,i} \exp(E_i/(R x_3))$. The parameters are $p_0 = 403.15\,\mathrm{K}$, $x^{in} = 5\,\mathrm{mol\,l^{-1}}$, $\Delta H_1 = 4.2\,\mathrm{kJ\,mol^{-1}}$, $\Delta H_2 = -11\,\mathrm{kJ\,mol^{-1}}$, $\Delta H_3 = -41.85\,\mathrm{kJ\ mol^{-1}}$, $\rho C_p = 2.812\,\mathrm{kJ(l\,K)^{-1}}$, $k_{0,1} = 1.287 \cdot 10^{12}\ \mathrm{h^{-1}}$, $E_1/R = -9758.3\,\mathrm{K}$, $k_{0,2} = 9.043 \cdot 10^9\ \mathrm{l(mol\ h)^{-1}}$ and $E_2/R = -8560\,\mathrm{K}$. The objective function is $\hat{\Phi} = \int w(x_2 - 1\mathrm{mol/l})^2 + u^T Q u \; dt$ with $w = 10000\,\mathrm{l^2/mol^2}$, $Q = diag(5 \cdot 10^{-4}\mathrm{h^4}, 5 \cdot 10^{-3}\mathrm{h^2/K})$ and $u = (\dot{\nu}_1, \dot{\nu}_2)^T$. The initial values are $x_{1,0} = 1\,\mathrm{mol\,l^{-1}}$, $x_{2,0} = 0.8\,\mathrm{mol\,l^{-1}}$, $x_{3,0} = 405\mathrm{K}$, $\hat{\Phi}_0 = 0$, $\nu_{1,0} = 19.5218\,\mathrm{h^{-1}}$ and $\nu_{2,0} = -160.5684\ \mathrm{K\ h^{-1}}$. No constraints are active throughout the simulation. The horizon length is 30 min and $\Delta t = 3\,\mathrm{min}$. We assume that $\tau_d^{QP} = \tau_d^{NEU} = 36\,\mathrm{sec}$.

In the simulation, the uncertain parameter p drops to 98 % on the first horizon, increases to 100 % on the second horizon and is then held constant. The response of the closed-loop system is examined for four different methods: a rigorous SQP method assuming no computational delay (ND), NEU without additional QP iterations (NEU) similar to [4] and [6], the original NEU algorithm (ONEU) and the extended NEU algorithm considering computational delay (ENEU). Fig. 2 illustrates that the control performance significantly decreases for NEU. In

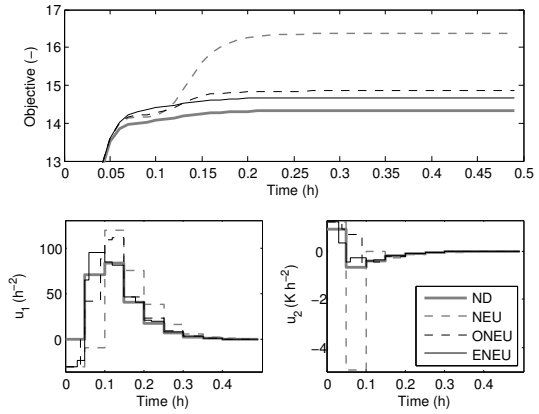

Fig. 2. Control performance of algorithms.

TABLE I

Number of QP iterations on horizons 1 to 10.

	1	2	3	4	5	6	7	8	9	10
ONEU	4	4	2	1	1	1	1	0	0	0
ENEU	3	1	2	3	2	2	2	2	2	2

comparison to the ONEU, the ENEU further improves the control performance since computational delay is considered. Table I shows the number of QP iterations.

V. Conclusions and future work

In this work, the extended NEU algorithm has been presented in order to optimize the control performance by a pre-optimization of the control updating period based on an a-priori error approximation and the consideration of computational delay. In this way, the extended NEU algorithm replaces the feasibility and optimality criteria of [7] and [10], which trigger a re-optimization, by a rigorous determination of the optimal number of additional QP iterations such that best possible control performance is guaranteed for all ratios η computational delay to sampling time. The application of the extended NEU algorithm to a case study has shown that the method presented improves the control performance in comparison to NEU algorithms without pre-optimization.

In a next step, the performance of the extended NEU algorithm will be further examined for a large-scale model of an industrial chemical process. For this purpose, the method presented must be extended because changes in the active set occur frequently in practice.

VI. Acknowledgement

This research has been partially supported by the European 7th framework STREP project HD-MPC, contract number INFSO-ICT-223854.

References

[1] R. Findeisen and F. Allgöwer, Computational delay in nonlinear model predictive control, in *Proc. Int. Symp. Adv. Control of Chemical Processes, IFAC ADCHEM*, Paper ID LTD 4.1.5 on CD-ROM, 2004.

[2] W.H. Chen, D.J. Ballance and J. O'Reilly, Model predictive control of nonlinear systems: computational burden and stability, *IEE Proceedings - Control Theory and Applications*, vol. 147(4), 2000, pp. 387-394.

[3] M. Alamir, A framework for monitoring control updating period in real-time NMPC schemes, in *Nonlinear Model Predicitve Control*, L. Magni, D. Raimondo and F. Allgöwer (Ed), Lecture Notes in Control and Information Sciences, Springer, 2009, pp. 433-445.

[4] M. Diehl, H.G. Bock and J.P. Schlöder, A real-time iteration scheme for nonlinear optimization in optimal feedback control,*SIAM Journal on Control and Optimization*, vol. 43, 2005, pp. 1714-1736.

[5] D. DeHaan and M. Guay, A real-time framework for model-predictive control of continuous-time nonlinear systems,*IEEE Transactions on Automatic Control*, vol. 52(11), 2007, pp. 2047-2057.

[6] C. Büskens and H. Maurer, Real-time control of an industrial robot under control and state constraints, in *Online Optimization of Large Scale Systems*, M. Grötschel, S.O. Krumke and J. Rambau (Ed), Springer, 2001, pp. 83-92.

[7] L. Würth, R. Hannemann and W. Marquardt, Neighboring-extremal updates for nonlinear model-predictive control and dynamic real-time optimization, *Journal of Process Control*, vol. 19(8), 2009, pp. 1277-1288.

[8] V.M. Zavala and L.T. Biegler, The advanced-step NMPC controller: optimality, stability and robustness, *Automatica*, vol. 45(1), 2009, pp. 86-93.

[9] Y. Wang and S. Boyd, Fast model predictive control using online optimization, *IEEE Transactions on Control Systems Technology*, vol. 18(2), 2010, pp. 267-278.

[10] J.V. Kadam and W. Marquardt, Sensitivity-based solution updates in closed-loop dynamic optimization, in *Proceedings of the IFAC DYCOPS*, vol. 7, Cambridge, USA, 2004.

[11] M. Schlegel, K. Stockmann, T. Binder, and W. Marquardt, Dynamic optimization using adaptive control vector parameterization, *Computers & Chemical Engineering*, vol. 29(8), 2005, pp. 1731-1751.

[12] J. Nocedal and S.J. Wright, *Numerical optimization*, Springer, 1999.

[13] R. Hannemann and W. Marquardt, Continuous and discrete composite adjoints for the Hessian of the Lagrangian in shooting algorithms for dynamic optimization, *SIAM Journal on Scientific Computing*, vol. 31(6), 2010, pp. 4675-4695.

[14] A. V. Fiacco, *Introduction to sensitivity and stability analysis in nonlinear programming*, Mathematics in science and engineering, Academic Press, vol. 165, 1983.

[15] M. Diehl, H.J. Ferreau and N. Haverbeke, Efficient numerical methods for nonlinear MPC and moving horizon estimation, in *Nonlinear Model Predictive Control*, L. Magni, D. Raimondo and F. Allgöwer (Ed), Lecture Notes in Control and Information Sciences, Springer, 2009, pp. 391-417.

[16] C. Kravaris, M. Niemiec, R. Berber and C.B. Brosilow, Nonlinear model-based control of nonminimum-phase processes, in *Nonlinear Model Based Process Control*, R. Berber and C. Kravaris (Ed), Kluwer Academic Publishers. 1998, pp. 115-141.

[17] L. Würth, R. Hannemann and W. Marquardt, A two-layer architecture for economically optimal process control and operation, *Journal of Process Control*, vol. 21(3), 2011, pp. 311-321.

[18] R. Bellman, *Dynamic Programming*, Adaptive control processes: a guided tour, Princeton University Press, 1957.