An Efficient Algorithm for Nonlinear Model Predictive Control of Large-Scale Systems
Part I: Description of the Method

Ein effizienter Algorithmus für die nichtlineare prädiktive Regelung großer Systeme
Teil I: Methodenbeschreibung

Moritz Diehl, Rolf Findeisen, Stefan Schwarzkopf, Ilknur Uslu, Frank Allgöwer, Hans Georg Bock, Ernst Dieter Gilles, and Johannes P. Schlöder

The work at hand consists of two parts: In Part I appearing in this issue, we present an efficient state-of-the-art algorithm for online optimization in nonlinear model predictive control (NMPC), the so called real-time iteration scheme; in Part II, which will appear in the next issue, we give an experimental proof-of-concept of the method, by presenting results obtained at a pilot-scale distillation column, where reoptimized controls are delivered every 20 seconds, employing a stiff differential-algebraic optimization model with more than 200 system states. The efficiency of the approach that is based on the direct multiple shooting method is due to a special initialization technique, the so called initial value embedding, which enables an optimal transition from one optimization problem to the next one. This allows to intertwine the optimization iterations with the process development in a way that only one iteration needs to be performed per sampling time, and the iterates nevertheless stay close to the respective optimal nonlinear solutions. A main advantage of the algorithm lies in the fact that nonlinear first principle system models – which are often developed anyway for process analysis and design – can be reused in a straightforward way for control purposes, even if they are large-scale.

Die vorliegende Arbeit besteht aus zwei Teilen: Im ersten Teil wird ein effizienter Algorithmus für die Echtzeit-Optimierung in der nichtlinearen prädiktiven Regelung (NMPC) vorgestellt, das so genannte Echtzeit-Iterations-Schema; in Teil II, der in der nächsten Ausgabe erscheint, wird anhand experimenteller Ergebnisse für eine Destillationskolonne im Pilotmaßstab die Praxistauglichkeit der beschriebenen Methode nachgewiesen, wobei alle 20 Sekunden neuoptimierte Steuergrößen bei einem steifen differentiell-algebraischen Optimierungsmodell mit über 200 Zuständen berechnet werden. Die Effizienz des Ansatzes, der auf dem direkten Mehrzielverfahren basiert, beruht auf einer speziellen Initialisierungstechnik, der Anfangswerteinbettung, die einen optimalen Übergang von einem Optimierungsproblem zum nächsten ermöglicht. Dies erlaubt, die Optimierungsiterationen mit der Prozessentwicklung zu verzahnen, sodass nur eine Iteration pro Abtastzeit durchgeführt werden muss, und die Iterationen trotzdem dicht an den jeweils optimalen Lösungen bleiben. Ein Hauptvorteil des Algorithmus liegt in der Tatsache, dass nichtlineare physikalische Systemmodelle – die häufig bereits für Prozessanalyse und -design entwickelt wurden – direkt für Regelungszwecke wiederverwendet werden können, selbst wenn sie große Systemdimensionen aufweisen.

Schlagwörter: Nichtlineare Prädiktive Regelung, Echtzeit-Optimierung, Differentiell-Algebraische Gleichungen, Direktes Mehrzielverfahren, Destillationskolonnen

Keywords: Nonlinear model predictive control, online-optimization, differential-algebraic equations, direct multiple shooting, distillation columns

1 Introduction

Online optimization of dynamic process models and nonlinear model predictive control (NMPC) have attracted increasing attention over the past decade [1; 8; 24; 30; 36]. Among the advantages of this approach are the flexibility provided in formulating the objective and the process model, the capability to directly handle equality and inequality constraints, and the possibility to treat unforeseen disturbances fast. It is in particular the availability of detailed nonlinear process models – that are increasingly being used for the *design* of industrial processes – which promises to make NMPC an appealing alternative to conventional control. One important precondition for successful NMPC applications, however, is the availability of reliable and efficient numerical optimal control algorithms, one of which will be presented in this paper.

While *linear* model predictive control is popular since the 70s of the past century and by now widely used in the process industry [25; 31; 35], the 90s have witnessed a steadily increasing attention from control theoreticians as well as control practitioners in the area of nonlinear model predictive control (NMPC). The practical interest is motivated by the fact that many systems are inherently nonlinear, and that higher product quality specifications, increasing productivity demands, tighter environmental regulations and demanding economical considerations in the process industry require to operate the systems closer to the boundary of the admissible operating region. Linear models are often inadequate to describe the process dynamics and thus nonlinear models must be used. Furthermore, new modeling techniques and the desire to use first principles models from the design of the plant up to the operation increases the interest in control techniques that can directly utilize the already available first principle models.

By now a couple of NMPC schemes that guarantee closed-loop stability and/or reduced computational demand have been developed, see for example [1; 14; 20; 30]. However, for a practical application of NMPC not only closed-loop stability is of interest. Questions such as performance, computational implementability, reliability and simple tuning are often much more important and thus must be carefully examined before NMPC can be successfully applied in practice. One essential problem is the high on-line computational load that is often associated with NMPC, since at each sampling instance a nonlinear optimal control problem must be solved. The optimization algorithms shall ideally be able to treat large-scale nonlinear first principle models as they are, without further need of modelling or model reduction. Several works have addressed this problem recently [2; 6; 21; 37; 40].

In this paper, we will present the recently developed *real-time iteration* scheme [15; 17] for optimization in NMPC. The scheme is based on the direct multiple shooting method for systems described by *differential algebraic equations* (DAE) [10; 26], as they often arise in practical applications. The efficiency of the direct multiple shooting method for dynamic optimization, which has been observed in many practical applications, has several reasons. One of the most important points is the possibility to incorporate information about the behavior of the state trajectory into the initial guess for the iterative solution procedure; this can damp the influence of poor initial guesses for the control inputs (which are usually much less known). In the context of NMPC, where a sequence of neighboring optimization problems is treated, solution information of the previous problem can be exploited efficiently by an *initial value embedding* strategy, as explained in Sect. 4. The efficiency of direct multiple shooting using the initial value embedding for NMPC has been observed and compared for various NMPC schemes in [17; 23; 32].

While the initial value embedding alone leads already to a significant decrease in the computation time, to iterate the optimization until final convergence at every sampling time might still be too long for an on-line implementation, if the model is large-scale or the process is fast. The solution we propose only performs one optimization iteration per sampling instant. The calculated approximated solutions can be shown to stay close to the exact optimal solutions, while feedback can be obtained much more frequently and the delay between the state measurements and the resulting optimal input response is considerably decreased. Furthermore, under suitable conditions it is possible to establish stability of this scheme.

In its actual implementation, the real-time iteration scheme is realized as part of the optimal control package MUSCOD-II [18; 26], which offers several advantages in the context of practical online optimization. Among these are the possibility to provide the DAE model equations as generic C or Fortran-Code or in the gPROMS modelling language [28; 34], to make use of efficient state-of-the-art DAE solvers (e. g. DDASAC [12], DAESOL [4]), or to employ an existing parallelization in the portable MPI standard, in time critical cases.

1.1 Outline

The paper consists of two parts: in this first part the numerical optimization method is explained allowing to capture the main underlying ideas. We start by introducing the considered moving horizon optimal control problem formulation in Sect. 2. The direct multiple shooting method for the solution of dynamic optimization problems is reviewed in Sect. 3. The crucial ingredient of the real-time iteration scheme, namely the initial value embedding for initialization of subsequent optimization problems, is introduced, visualized and discussed in Sect. 4. In Sect. 5 the *real-time iteration scheme* is presented.

In Part II, which appears in the next issue, we demonstrate for a real world problem, namely the control of a pilot-plant distillation column, that the method makes NMPC based on first principle models feasible.

2 Nonlinear Model Predictive Control

Throughout this paper, we consider time invariant DAE models of index one in the following form

$$\begin{aligned} B(\bar{x}(t), \bar{z}(t), \bar{u}(t))\dot{\bar{x}}(t) &= f(\bar{x}(t), \bar{z}(t), \bar{u}(t)) \\ 0 &= g(\bar{x}(t), \bar{z}(t), \bar{u}(t)) \,. \end{aligned}$$

Here, $\bar{x}$ and $\bar{z}$ denote the differential and the algebraic state vectors, respectively, and $\bar{u}$ is the vector of control inputs. The bar is introduced to distinguish the states and the implemented controls of the *real* system from the *simulated* states x, z and controls u in the optimizer, introduced in the following subsection. Note that the algorithm is equally applicable to DAE depending on time and parameters, which are here omitted for clarity of presentation.

2.1 Open Loop Optimal Control Problem

Given an observed system state $\bar{x}(t)$ at a time t, a typical NMPC control scheme (see [1; 29] for more details) obtains a feedback from the repeated solution of an open-loop optimal control problem on a prediction horizon $\left[0, T_p\right]$ with horizon length T_p:

$$\min_{u(\cdot), x(\cdot), z(\cdot)} \int_0^{T_p} L(x(\tau), z(\tau), u(\tau))d\tau + E\big(x\big(T_p\big)\big) \tag{1a}$$

subject to the initial value constraint

$$x(0) = \bar{x}(t) \tag{1b}$$

and the DAE system

$$B(\cdot)\dot{x}(\tau) = f(x(\tau), z(t), u(\tau)) \,, \tag{1c}$$

$$0 = g(x(\tau), z(\tau), u(\tau)) \,. \tag{1d}$$

as well as possible constraints on states and controls

$$h(x(\tau), z(\tau), u(\tau)) \geq 0 \,, \quad \forall \tau \in \left[0, T_p\right] \tag{1e}$$

and a final terminal constraint of the form

$$r\big(x\big(T_p\big)\big) \geq 0 \,. \tag{1f}$$

The function L appearing in the cost (1a) is often positive definite and can arise for example from economical considerations. The terminal penalty term in (1a) penalizes the final predicted state and is often introduced together with the terminal constraint (1f) for stability reasons [20; 29].

Solving this problem for a given initial value $\bar{x}(t)$, we obtain an open-loop optimal control input $u^*(\tau; \bar{x}(t))$ and a corresponding state trajectory $(x^*(\tau; \bar{x}(t)), z^*(\tau; \bar{x}(t)))$, $\tau \in \left[0, T_p\right]$.

2.2 Implemented Feedback

If the real system state $(\bar{x}(t), \bar{z}(t))$ would obey the nominal DAE the optimal input $u^*(\cdot; \bar{x}(t))$ could be applied open-loop until $t_0 + T_p$. However, due to occurring disturbances and model uncertainties the predicted open-loop trajectory and the resulting closed-loop trajectory will differ in general. To introduce feedback, the open-loop optimal control problem (1) is thus solved repeatedly and the feedback applied is given by the repeated solution of the NMPC open-loop optimal control problem. Ideally, this would be realized by an instantaneous feedback, similar to optimal feedback control, $u(\bar{x}) := u^*(0; \bar{x})$, which would either require the instantaneous solution of the optimal control problems (1) for each $\bar{x}$, or an extensive precalculation, both of which are not feasible in most practical applications. Therefore, in NMPC, the optimal control problems are typically only solved at discrete sampling times $t_k, t_{k+1}, \ldots,$ and the obtained open-loop controls are applied in between, by defining

$$\bar{u}(t) := u^*(t - t_k; \bar{x}(t_k)) \,, \quad t \in [t_k, t_{k+1}) \,, \tag{2}$$

where t_k is the previous closest sampling instant to the time t. The times t_k are typically assumed to be a fixed time $\delta_r = t_k - t_{k-1}$ apart. While this delivers sufficient time for reoptimization, note that most numerical schemes used so far would also result in a feedback delay of δ_r between the incoming state observation $\bar{x}(t_k)$, and the availablility of the feedback controls, even though this topic is rarely addressed in the literature. It is the objective of this paper to derive an efficient solution method for the online solution of (1), which not only allows to decrease the recalculation time δ_r considerably, but also to minimize the remaining feedback delay to a fraction of δ_r.

Due to its origin from optimal control, the NMPC feedback law has several appealing properties: among them are the possibility to base the feedback on economic criteria, to make use of important process knowledge in the form of nonlinear first principle models, and to include constraints (1e) in a straightforward way.

Given suitable choices of the objective functional defined via L and E and the final state constraint (1f), stability of the nominal NMPC dynamics can be proven even for nonlinear systems [13; 14; 20; 30].

Note that similar methods as outlined in the following for NMPC can also be used for state estimation using moving horizon state estimators (MHE) [11; 22], since the moving horizon state estimation involves the solution of a related dynamic optimization problem.

3 Direct Multiple Shooting

Different solution strategies for the open-loop optimal control problem (1) exist, see for example [7; 8]. The real-time iteration scheme proposed in this paper is based on the direct multiple shooting method, which is reviewed briefly in this section. The review given here lays the basis for the presentation of the real-time iteration scheme introduced in Sects. 4 and 5. For a more detailed description of direct multiple shooting and its implementation in the package MUSCOD-II we refer to [26–28], or [18].

3.1 State and Input Parametrization

The parametrization of the infinite optimization problem consists of two steps. For a suitable partition of the prediction time horizon $[0, T_p]$ into N subintervals $[\tau_i, \tau_{i+1}]$ with

$$0 = \tau_0 < \tau_1 < \ldots < \tau_N = T_p$$

we first discretize the control function $u(\cdot)$. For simplicity, we assume here that $u(\cdot)$ is parametrized as a piecewise constant vector function

$$u(\tau) = u_i\,, \quad \text{for}\, \tau \in [\tau_i, \tau_{i+1}]\,.$$

Note, however, that every parametrization with local support could be used without changing the basic structure of the resulting problem.

In a second step the DAE is parametrized by *multiple shooting*. We decouple the DAE solution on the N intervals $[\tau_i, \tau_{i+1}]$ by introducing the initial values s_i^x and s_i^z (for simplicity they are combined in the vector s_i in the following) of differential and algebraic states at the multiple shooting nodes τ_i as additional optimization variables.

On each subinterval $[\tau_i, \tau_{i+1}]$ we compute the trajectories $x_i(t)$ and $z_i(t)$ as the solution of the corresponding initial value problem:

$$B(\cdot)\dot{x}_i(\tau) = f(x_i(\tau), z_i(\tau), u_i) \tag{3a}$$

$$0 = g(x_i(\tau), z_i(\tau), u_i) - \alpha_i(\tau) g\big(s_i^x, s_i^z, u_i\big) \tag{3b}$$

$$x_i(\tau_i) = s_i^x \tag{3c}$$

The subtrahend in (3b) is deliberately introduced to allow an efficient DAE solution for initial values and controls s_i^x, s_i^z, u_i that may violate temporarily the consistency conditions (1d). Therefore we require for the scalar damping factor α that $\alpha_i(t_i) = 1$. For more details on the relaxation of the DAE the reader is referred, e. g. to [9; 26; 39]. Note that the trajectories $x_i(t)$ and $z_i(t)$ on interval $[\tau_i, \tau_{i+1}]$ are functions of the initial values and controls s_i^x, s_i^z, u_i only. Thus, we will refer to them as $x_i(t; s_i, u_i)$ and $z_i(t; s_i, u_i)$ in the following. The integral part of the cost function is evaluated on each interval independently:

$$L_i(s_i, u_i) := \int_{\tau_i}^{\tau_{i+1}} L(x_i(\tau), z_i(\tau), u_i) d\tau\,. \tag{4}$$

3.2 Resulting Structured Nonlinear Program

The parametrization of problem (1a)–(1f) using multiple shooting and a piecewise constant control representation leads to the following structured nonlinear programming (NLP) problem:

$$\min_{u_i, s_i} \sum_{i=0}^{N-1} L_i(s_i, u_i) + E\left(s_N^x\right) \tag{5a}$$

subject to the initial value constraint

$$s_0^x = \bar{x}(t_k)\,, \tag{5b}$$

the continuity conditions

$$s_{i+1}^x = x_i(\tau_{i+1}; s_i, u_i) \qquad i = 0, 1, \ldots N-1\,, \tag{5c}$$

and the consistency conditions

$$0 = g\left(s_i^x, s_i^z, u_i\right) \qquad i = 0, 1, \ldots N-1\,. \tag{5d}$$

Note that the control and path constraints are imposed only pointwise at the multiple shooting nodes

$$h\left(s_i^x, s_i^z, u_i\right) \geq 0 \qquad i = 0, 1, \ldots N-1 \tag{5e}$$

as well as for the final node

$$r\left(s_N^x\right) \geq 0\,. \tag{5f}$$

The NLP (5a))–(5f) can be summarized as

$$\min_w F(w) \quad \text{subject to} \quad \begin{cases} G(w) = 0 \\ H(w) \geq 0\,, \end{cases} \tag{6}$$

where w contains all the multiple shooting state variables and controls:

$$w^T = \left[{s_0^x}^T, {s_0^z}^T, {u_0}^T, {s_1^x}^T, {s_1^z}^T, {u_1}^T, \ldots, {u_{N-1}}^T, {s_N^x}^T\right]\,.$$

The discretized initial value problem is included in the equality constraints, as

$$G(w) = \begin{bmatrix} s_0^x - \bar{x}\,(t_k) \\ s_1^x - x_0\left(\tau_1; s_0^x, s_0^z, u_0\right) \\ g\left(s_0^x, s_0^z, u_0\right) \\ \vdots \end{bmatrix} \tag{7}$$

3.3 Sequential Quadratic Programming (SQP)

The NLP problem (6) can be solved efficiently by a *sequential quadratic programming (SQP)* method tailored to the multiple shooting structure. Starting from an initial guess w^0, a simple SQP method for the solution of the NLP iterates

$$w^{k+1} = w^k + \alpha^k \Delta w^k, \quad k = 0, 1, \ldots\,, \tag{8}$$

where $\alpha^k \in [0, 1]$ is a relaxation factor to be determined by a line search, and the search direction Δw^k is the solution of a quadratic programming (QP) subproblem

$$\min_{\Delta w \in \mathbb{R}^n} \nabla F\left(w^k\right)^T \Delta w + \frac{1}{2} \Delta w^T A^k \Delta w \tag{9}$$

subject to

$$G\big(w^k\big) + \nabla G\big(w^k\big)^T \Delta w = 0$$
$$H\big(w^k\big) + \nabla H\big(w^k\big)^T \Delta w \geq 0\,.$$

Here, A^k denotes an approximation of the Hessian $\nabla_w^2 \mathcal{L}$ of the *Lagrangian function* $\mathcal{L}$,

$$\mathcal{L}(w, \lambda, \mu) = F(w) - \lambda^T G(w) - \mu^T H(w)\,,$$

where λ and μ are the Lagrange multipliers. Several possibilities for the Hessian approximations A^k exist:

- In the exact Hessian SQP method, $A^k := \nabla_w^2 \mathcal{L}\left(w^k, \lambda^k, \mu^k\right)$ where multiplier guesses λ^k, μ^k can be obtained from the solution of the QP (9) at the previous iterate w^{k-1}.
- A popular method is to compute the Hessian approximation by an update formula based on the Lagrange gradient differences, which are cheaply available.
- A third approach to obtain the Hessian approximation – the constrained Gauß–Newton method – is recommended in the special case of a least squares type cost function $F(w) = \|C(w)\|_2^2$. The cheaply available matrix
$$A^k := 2\nabla_w C(w^k) \nabla_w C\left(w^k\right)^T \tag{10}$$
provides an excellent approximation of the Hessian, if the residual $\|C(w)\|_2$ of the cost function is small.

The optimal values of s_i, u_i and the complete vector w resulting from the (iterative) solution of (6) in dependence of $\bar{x}(t_k)$ are shortly denoted by $s_i^*(\bar{x}(t_k))$, $u^*(\bar{x}(t_k))$ and $w^*(\bar{x}(t_k))$.

3.4 Tailoring SQP for Multiple Shooting

In the following we give some remarks on how to exploit the multiple shooting structure in the construction of a specially tailored SQP method. Due to the deliberate choice of state and control parameterizations the NLP problem (6) and the resulting QP problems (9) have a particular structure: the Lagrangian function $\mathcal{L}$ is *partially separable* and its Hessian $\nabla_w^2 \mathcal{L}$ is *block diagonal*, a fact which can for example be exploited by the use of high rank block updates [10]. Similarly, the multiple shooting parametrization introduces a characteristic *block sparse structure* of the Jacobian matrices $\nabla G(w)^T$ and $\nabla H(w)^T$.

It is of crucial importance for performance and numerical stability of the direct multiple shooting method for NMPC that these structures of (5a)–(5f) are fully exploited. We shortly mention three important points that deserve special attention:

- *Internal Numerical Differentiation:* The solution of the DAE initial value problems (3a)–(3c) and the corresponding derivatives necessary to compute (7) can be computed simultaneously by specially designed integrators which use the principle of *internal numerical differentiation*. In particular, the integrator DAESOL [3; 5], which is based on the backward-differentiation-formulae (BDF), is used in the algorithm outlined in this paper.
- *Partial Reduction Technique for DAE:* In Leinweber [26] a reduction technique for DAE systems with a large share of algebraic variables is developed, which is also employed for the computations in this paper. The technique exploits the linearized algebraic consistency conditions (5d) for a reduction in variable space, so that only reduced gradients and Hessian blocks need to be calculated, see also [9; 38].
- *Gauß–Newton Hessian for Least Squares Integrals:* As said above, the Gauß–Newton Hessian approximation (10) is especially recommended for least squares cost functions, which often occur in NMPC. However, the involved least squares terms in the objective (4) typically arise in integral form: $L_i(s_i, u_i) = \int_{\tau_i}^{\tau_{i+1}} \|l(x_i(t), z_i(t), u_i)\|_2^2 d\tau$. An efficient method for computation of the corresponding Gauß–Newton Hessian blocks *simultaneously* with the DAE solution, which delivers the Hessian approximation at virtually no additional costs during each SQP iteration, has been implemented in the current version of the optimal control package MUSCOD-II, in conjunction with the implicit DAE solver DAESOL [15; 19].

4 Initial Value Embedding

So far the direct multiple shooting method was only considered for the solution of *one single* optimal control problem (1a)–(1f). In the context of NMPC, several optimization problems at the recalculation times t_k have to be solved consecutively, for different initial values $\bar{x}(t_k)$. For simplicity we denote in the following the problem (1a)–(1f) resp. its multiple shooting parameterization (5a)–(5f) by $P(\bar{x}(t_k))$. The question arises of how to find an initial guess w^0 for initialization of the SQP iterations towards the solution of each problem $P(\bar{x}(t_k))$.

Fortunately, we know already, from a previous optimization, the solution of a neighboring optimization problem $P(\bar{x}(t_k))$. As shown this solution is a good initial guess for the current problem $P(\bar{x}(t_k))$, if the transition from one problem to the next fully exploits the advantages of the direct multiple shooting method. Before we present the initialization proposed here we review a rather straightforward initialization approach which we refer to as the „conventional approach“.

4.1 Conventional Initialization

Given the optimal control values $u_0^*(\bar{x}(t_{k-1}))$, $u_1^*(\bar{x}(t_{k-1}))$, $\ldots, u_{N-1}^*(\bar{x}(t_{k-1}))$ from the solution of the *previous* optimization problem $P(\bar{x}(t_{k-1}))$, and given also a new initial value $\bar{x}(t_k)$, one can open-loop simulate the system over the interval $[0, T_p]$ starting from $\bar{x}(t_k)$. This leads to a physical solution trajectory and the resulting states at the multiple shooting nodes can be used for the initialization of the *current* optimization problem $P(\bar{x}(t_k))$. Note that the controls may also be shifted in time before the open-loop simulation, taking account of the horizon movement, as proposed e. g. in [33]; when the new system state $\bar{x}(t_k)$ is perturbed by unmodelled disturbances, however, this would not make a big difference in the resulting initialization.

The left column of Fig. 1 shows the initialization based on this open-loop simulation, considering a distillation column control problem described in Part II of this paper. In this problem the cost function is of least squares type, penalizing the deviations of the two shown temperature trajectories

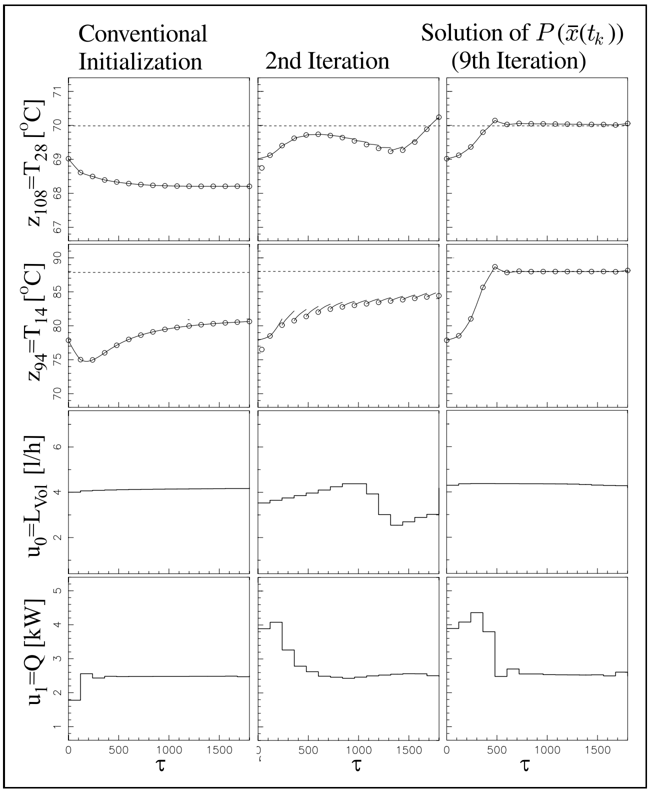

Figure 1: Solution iterations for distillation model, after conventional initialization: shown are two temperatures and two controls, for the initialization (left column), the second iterate (middle column), and the optimal solution (right column).

(first two rows of graphs) from their setpoints 70 °C and 88 °C. The controls used for the initialization (last two rows of the first column of the graphs) are taken from the previous solution of problem $P(x(t_{k-1}))$.

As can be seen, the simulated temperature trajectories using the conventional initialization are far from the setpoint values due to the mismatch between the new, perturbed initial value $\bar{x}(t_k)$ and the previous optimal controls $u_0^*(\bar{x}(t_{k-1})), u_1^*(\bar{x}(t_{k-1})), \ldots, u_{N-1}^*(\bar{x}(t_{k-1}))$. This would equally be true if a shift in the controls was performed.

To obtain the given plots, the Gauß–Newton approach outlined in Sect. 3.4, in conjunction with a full step SQP method is used. After two iterations (middle column), the temperature profile has approached the setpoints 70 °C and 88 °C, but due to nonlinearities of the system this iterate is still far from the optimal solution (right column), which is only attained after 9 SQP iterations. If instead of the Gauß–Newton method the widely used BFGS-update is employed, even 32 SQP iterations are necessary.

4.2 Initialization via Initial Value Embedding

The initialization approach we present here is simple but efficient. It directly builds on the advantages of the direct multiple shooting method. Instead of using the knowledge of the new state $\bar{x}(t_k)$ for the initialization, one can ignore this information and use the complete solution of the previous problem $P(\bar{x}(t_{k-1}))$ *without any modification*. Figure 2, left column shows the corresponding initialization using this approach for the same example problem as for the conventional initialization shown in Fig. 1. Already the *first* iteration delivers a fairly good approximation for the exact solution of $P(\bar{x}(t_k))$ (compare with the second iteration of the conventional approach in Fig. 1). Note that the graphs of the first iteration show discontinuous (inconsistent) trajectories, as they are typical in the multiple shooting method. From a numerical point of view, this can be regarded an equal distribution of the linearization errors which avoids an error accumulation at the end of the horizon and allows the optimizer to stay close to the desired optimal trajectories during the iterations.

The simple initialization for the current problem $P(\bar{x}(t_k))$ using the full information of $P(\bar{x}(t_{k-1}))$ (but neglecting $\bar{x}(t_{k-1})$ in the initialization) results, of course, in a violation of the initial value constraint (5b) in the NLP (5a)–(5f), since $s_0^x = \bar{x}(t_{k-1}) \neq \bar{x}(t_k)$. However, this constraint is immediately satisfied after the first full step SQP iteration, due to its linearity.

In the shown example, the trajectory of this first iteration (middle column in Fig. 2) shows already the main characteristics of the solution (right column), which is attained after 3 iterations.

The formulation of the initial value constraint (5b) in the NLP (5a)–(5f) can be considered a linear embedding of each optimization problem into the manifold of perturbed problems. It is this particular feature of the direct multiple

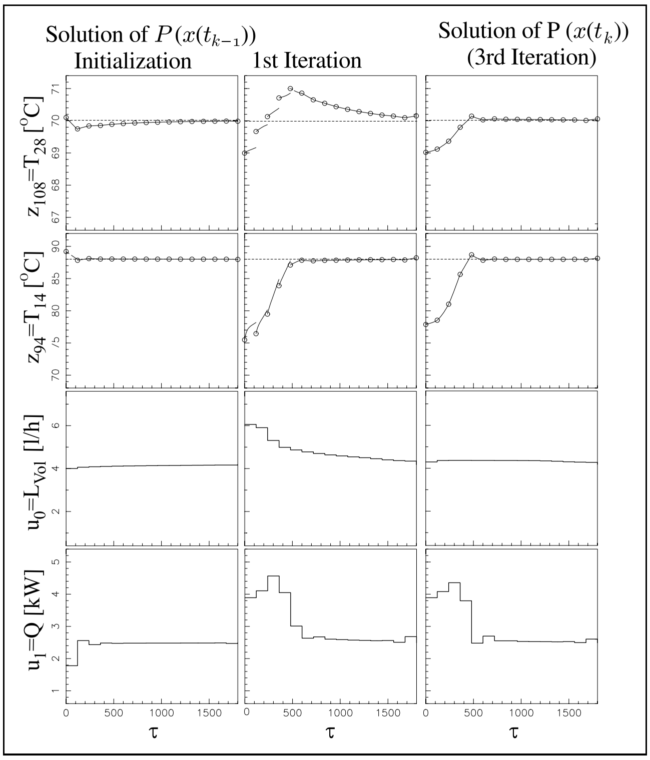

Figure 2: Initial value embedding: initialization at the solution of $P(\bar{x}(t_{k-1}))$ (left column), the first iteration (middle column) and the exact solution $P(\bar{x}(t_k))$.

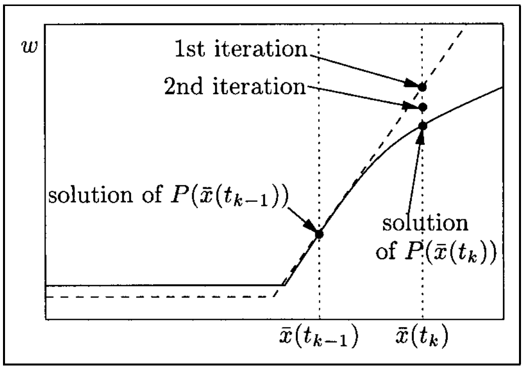

Figure 3: Exact solution manifold (solid line) and tangential predictor after initial value embedding (dashed line), when initialized with the solution of $P(\bar{x}(t_{k-1}))$. The first iteration delivers already a good predictor for the exact solution of $P(\bar{x}(t_k))$.

shooting method that allows an efficient transition from one optimization problem to the next.

A visualization of the principle behind the initial value embedding is given in Fig. 3. The solid line represents for each initial value $\bar{x}(t)$ the corresponding solution vector w in the NLP variable space. This is the exact solution manifold, which is in practice never computed. Note that usually also some nondifferentiable points occur whenever the set of active inequality constraints changes. The dashed line shows the tangential predictor that is delivered from the first (full step) SQP iteration, when the variables are initialized with the solution of $P(\bar{x}(t_{k-1}))$. This predictor is in particular valid for the current value $\bar{x}(t_k)$ and thus delivers a first iterate w^1 that is close to the exact solution $w^*(\bar{x}(t_k))$ of $P(\bar{x}(t_k))$. This property, namely that

$$\left\| w^1 - w^*(\bar{x}(t_k)) \right\| = O\left(\left\| (\bar{x}(t_{k-1}) - \bar{x}(t_k) \right\|^2 \right) ,$$

even holds under mild conditions *at points where the active set changes* [15].

Note that the predictor is strictly tangential to the solution manifold only for the exact Hessian SQP method. However, for the Gauß–Newton approach the predictor is still approximately tangential.

Remark 1: Note that the initialization by the initial value embedding does require *no* computational effort at all, as the previous solution is already available – this is in contrast to the conventional initialization, where an additional simulation of the DAE system had to be performed. Furthermore, most computations that are necessary for the solution of the first QP (9) can already be executed *before* the current value of $\bar{x}(t_k)$ is actually available as outlined in the next section. This fact plays a crucial role in the context of the real-time iteration scheme, because it allows to deliver a fast feedback response with negligible computational delay.

Remark 2: For *linear* systems with least squares objective the first iterate is already the solution, as in this case the multiple shooting NLP (5a)–(5f) is itself nothing else than a QP, and therefore the first QP (9) is already identical to the full NLP (6) and delivers the same solution.

Remark 3: Note that the initial value embedding is fully compatible with a shift strategy (cf. [33]), where all variables from the previous optimization problem are shifted to take account of the horizon movement in time, supplemented at the end, for example, by a simulated trajectory piece with steady state input. The shift strategy for real-time iterations is especially advantageous for periodic or time dependent processes [15].

Even though the initial value embedding strategy for direct multiple shooting does already lead to a decrease in the required solution time in NMPC [17; 21; 23] the required solution time might still be too high for a real-time application, especially for fast processes or large-scale systems. Furthermore, we would still have a feedback delay of one sampling time, if we solve each problem until convergence and apply the resulting control only afterwards. To overcome these two problems, we propose in the following an approximated solution strategy using only one SQP iteration per recalculation instant, which solves the dynamic optimization problems simultaneously with the process development.

5 Real-Time Iterations

We consider now the full real-time scenario, where we want to solve a sequence of optimization problems $P(\bar{x}(t_k))$.

5.1 The Online Dilemma

Given that the computational time for one SQP iteration is more or less constant, we have to confront the following dilemma: If we want to obtain a sufficiently exact solution for a problem $P(\bar{x}(t_k))$, we have to perform several SQP iterations until a prespecified convergence criterion is satisfied. Assume that for achieving this we have to perform n iterations, and that each iteration takes a time δ. This means that we obtain the optimal feedback control $u_0^*(\bar{x}(t_k))$ only at a time $t_k + n\delta$, i. e., with a considerable delay. However at time $t_k + n\delta$ the system state has already moved to some system state $\bar{x}(t_k + n\delta) \neq \bar{x}(t_k)$, and $u_0^*(x(t_k))$ is *not* the exact NMPC feedback, $u_0^*(\bar{x}(t_k + n\delta))$. In the best case the system state has not changed much in the meantime and it is a good approximation of the exact NMPC feedback. The problem of which controls have to be applied in the meantime is still unsolved: a possible choice would be to use previously optimized controls in an open-loop manner. Note that with this approach we can realize an NMPC recalculation rate with intervals of length $\delta_r = n\delta$ (under the assumption that *each* problem needs n iterations and that each SQP iteration requires less or equal than δ), and that each feedback comes with a delay of $\delta_d = \delta_r = n\delta$.

We may initialize, of course, subsequent problems with the help of the initial value embedding. However, note that due to long recalculation times the system state may change considerably from one problem to the next, and the SQP procedure may therefore still require a relatively high number of iterates.

5.2 Real-Time Iteration Idea

We will now present our answer to the online dilemma. The approach is based on three ideas:

- Due to the above observations, we will never be able to compute the *exact* NMPC feedback control $u_0^*(\bar{x}(t_k))$ without delay. Therefore, it may be better to compute only an *approximation* $\tilde{u}_0(\bar{x}(t_k))$ of $u_0^*(\bar{x}(t_k))$, if this approximation can be computed much faster. Fortunately, such an approximation is delivered by the initial value embedding, as the first iteration is a tangential predictor for the exact solution. By using just the first iteration, we could reduce the feedback delay δ_d already considerably, to the time $\delta_d = \delta$. But this can still be improved:
- As mentioned in Remark 1 in Sect. 4.2, the computations for the first iteration can be largely performed *before* the initial value $\bar{x}(t_k)$ is known. Therefore, we can reduce the delay time further, if we perform all these computations before time t_k, and at time t_k we can quickly compute the feedback response $\tilde{u}_0(\bar{x}(t_k))$ to the current state, with a delay δ_d that is even considerably smaller than δ: $\delta_d \ll \delta$.
- Assuming that we already use an approximate solution of the optimal control problem we can ask if it is really necessary to iterate the SQP until convergence requiring a time $\delta_r = n\delta$. Instead, we can furthermore reduce the recalculation time by performing just one iteration per recalculation interval allowing considerably shorter sampling intervals. A positive side-effect is, that this shorter recalculation time ($\delta_r = \delta$ instead of $n\delta$) most probably leads to smaller differences in subsequent initial states $\bar{x}(t_k)$ and $\bar{x}(t_{k+1})$ for consecutive optimization problems.

These ideas set the basis for the real-time iteration scheme. The focus is now shifted from the sequence of optimization problems towards the sequence of iterates: we may regard the SQP procedure iterating uninterrupted, with the only particularity that the initial value $\bar{x}(t_k)$ is modified *during* the iterations. The generation of the feedback controls can then be regarded as a by-product of the SQP iterations. Due to the initial value embedding property, it can be expected that the iterates remain close to the exact solution manifold for each new problem. In Fig. 4 four consecutive real-time iterates are sketched, where the dashed lines show the respective tangential predictors.

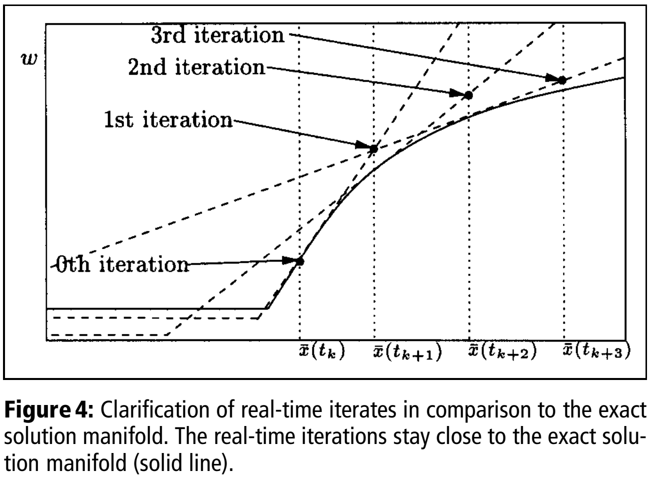

Figure 4: Clarification of real-time iterates in comparison to the exact solution manifold. The real-time iterations stay close to the exact solution manifold (solid line).

5.3 Real-Time Iteration Algorithm

Let us assume that we start with an initial guess w^0 for all multiple shooting variables. We set the iteration index k to zero and perform the following steps:

1. Preparation: Based on the initial guess $w^k = \left(s_0^{x\,k}, s_0^{z\,k}, u_0{}^k, \dots\right)$, compute nearly all components of the QP (9):

$$\min_{\Delta w \in \mathbb{R}^n} \nabla F\left(w^k\right)^T \Delta w + \frac{1}{2} \Delta w^T, A\left(w^k\right) \Delta w$$
$$\text{s.t.} \quad \begin{cases} G\left(w^k\right) + \nabla G\left(w^k\right)^T \Delta w = 0 \\ H\left(w^k\right) + \nabla H\left(w^k\right)^T \Delta w \geq 0 \,. \end{cases}$$

   Note that it is only in the first component of $G(w^k)$, where the initial value $\bar{x}(t_k)$ actually enters, cf. (7); all other components, in particular all derivatives and the Gauß–Newton Hessian $A(w^k)$, can already be computed, which takes already the lion's share of the computations for each SQP iteration. Furthermore, the above QP will be presolved to a large extent, as much as possible without knowledge of $\bar{x}(t_k)$, to prepare the next step.
2. Feedback Response: at time t_k, after the preparation phase has been finished, the real system is observed, and the missing initial value $\bar{x}(t_k)$ is obtained. Using $\bar{x}(t_k)$, the solution of the QP subproblem is calculated fast to obtain the step vector $\Delta w^k = \left(\Delta s_0^{xk}, \Delta s_0^{zk}, \Delta u_0^k, \dots\right)$ and give the tangential approximation of the NMPC feedback control,

$$\tilde{u}(\bar{x}(t_k)) := u_0{}^k + \Delta u_0{}^k \,,$$

   immediately to the real system.
3. Transition: Set the next initial guess as

$$w^{k+1} := w^k + \Delta w^k \,, \tag{11}$$

   set $k = k+1$, and go to 1 .

We only sketched the main ideas and refer to [15; 17] for more details. The time for each recalculation cycle corresponds to the time of one SQP iteration, δ, i. e., a recalculation rate with intervals of length $\delta_r = \delta$ can be realized, and at every recalculation instant a new feedback is delivered.

Furthermore, note that nearly all computations of a recalculation cycle are concentrated in the preparation phase 1; therefore, the feedback delay between the obtained meas-

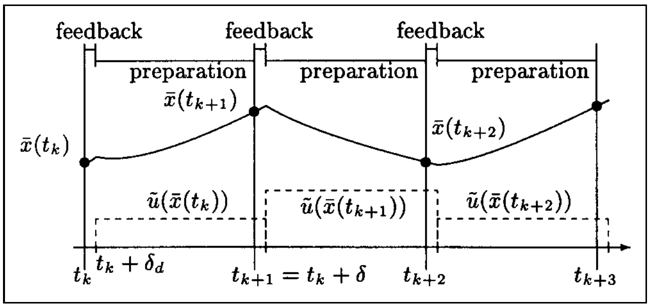

Figure 5: Segmentation of computation time in the real-time iteration scheme; real system state and control trajectory, for sampling time δ and feedback delay $\delta_d \ll \delta$.

urement and resulting input in step 2 is minimized to a value $\delta_d \ll \delta$. An overview on the employment of computation time in the real-time iteration scheme is given in Fig. 5, together with a sketch of state and control trajectories.

The delay δ_d between the measurement and the resulting input is in practice typically two orders of magnitude smaller than the complete recalculation time $\delta_r = \delta$, as will be shown in the experimental case study in Part II of the paper.

5.4 Theoretical Contraction Properties

The contraction properties of the scheme have so far only been investigated theoretically for the related class of shrinking horizon problems for the nominal system. Specifically it has been established that the scheme delivers approximations w^k that come closer and closer to the solutions of the corresponding optimization problems, and that the loss of optimality – compared to the exact optimal feedback control – can be bounded [15; 16]. For the moving horizon case, as treated here, the theoretical convergence properties are currently under investigation, and a nominal stability result will be published soon.

6 Conclusions (Part I)

We have presented a new numerical approach to NMPC, the *real-time iteration* scheme. The scheme is based on the direct multiple shooting method and employs an *initial value embedding* for optimal transition from one optimization problem to the next. The algorithm is implemented within the dynamic optimization package MUSCOD-II and offers the following practical advantages in the context of online process optimization and in particular NMPC:

- The DAE model equations can either be provided in the gPROMS modelling language [28; 34] or as generic C or Fortran-Code.
- Efficient state-of-the-art DAE solvers (e. g. DDASAC [12], DAESOL [3]) are employed to calculate the system trajectories and derivatives quickly and accurately.
- The direct multiple shooting method, as a simultaneous strategy, allows to efficiently treat highly nonlinear and unstable systems [16].
- The approach allows a robust treatment of control and path constraints as well as boundary conditions.
- The real-time iteration scheme needs only short sampling times to perform a fully nonlinear optimization.
- The feedback delay is much shorter than a sampling time.
- The method is well suited for parallel computation and allows considerable speedups, since the integrations are decoupled on different multiple shooting intervals [28]. An existing parallel implementation is based on the portable MPI standard.

In Part II of the paper, the algorithm is experimentally applied for the control of a binary distillation column, using a DAE model with 82 differential and 122 algebraic state variables for the optimization, with sampling times of 20 seconds. The presented results show that nonlinear model predictive control can even be applied nowadays for the control of large-scale systems and does lead to satisfying performance if the outlined solution approach is used.

Acknowledgements

Financial support by the *Deutsche Forschungsgemeinschaft (DFG)* within the *DFG-Schwerpunktprogramm* „Echtzeitoptimierung großer Systeme“ is gratefully acknowledged.

References

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

[2] R.A. Bartlett, A. Wächter, L.T. Biegler. Active set vs. inerior point strategies for model predicitve control. In *Proc. Amer. Contr. Conf.*, pages 4229–4233, Chicago, Il, 2000.

[3] I. Bauer. *Numerische Verfahren zur Lösung von Anfangswertaufgaben und zur Generierung von ersten und zweiten Ableitungen mit Anwendungen bei Optimierungsaufgaben in Chemie und Verfahrenstechnik*. PhD thesis, University of Heidelberg, 2000.

[4] I. Bauer, H.G. Bock, J.P. Schlöder. DAESOL – a BDF-code for the numerical solution of differential algebraic equations. Internal report, IWR, SFB 359, University of Heidelberg, 1999.

[5] I. Bauer, F. Finocchi, W.J. Duschl, H.-P. Gail, J.P. Schlöder. Simulation of chemical reactions and dust destruction in protoplanetary accretion discs. *Astron. Astrophys.*, 317:273–289, 1997.

[6] L.T. Biegler. Efficient solution of dynamic optimization and NMPC problems. In F. Allgöwer, A. Zheng, editors, *Nonlinear Predictive Control*, volume 26 of *Progress in Systems Theory*, pages 219–244, Basel, 2000. Birkhäuser.

[7] L.T. Biegler, J.B. Rawlings. Optimization approaches to nonlinear model predictive control. In W.H. Ray, Y. Arkun, editors, *Proc. 4th International Conference on Chemical Process Control – CPC IV*, pages 543–571. AIChE, CACHE, 1991.

[8] T. Binder, L. Blank, H.G. Bock, R. Burlisch, W. Dahmen, M. Diehl, T. Kronseder, W. Marquardt, J.P. Schlöder, O. von

Stryk. Introduction to model based optimization of chemical processes on moving horizons. In M. Grötschel, S.O. Krumke, J. Rambau, editors, *Online Optimization of Large Scale Systems: State of the Art*, pages 295–339. Springer, 2001.

[9] H.G. Bock, E. Eich, J.P. Schlöder. Numerical solution of constrained least squares boundary value problems in differential-algebraic equations. In K. Strehmel, editor, *Numerical Treatment of Differential Equations*. Teubner, Leipzig, 1988.

[10] H.G. Bock, K.J. Plitt. A multiple shooting algorithm for direct solution of optimal control problems. In *Proc. 9th IFAC World Congress Budapest*. Pergamon Press, 1984.

[11] T. Bürner. Algorithmen zur Lösung von Optimalsteuerungsproblemen bei Systemschätzung in Echtzeit. Master’s thesis, University of Heidelberg, 2002.

[12] M. Caracotsios, W.E. Stewart. Sensitivity analysis of initial value problems with mixed odes and algebraic equations. *Comput. Chem. Engng.*, 9:359–365, 1985.

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

[14] G. de Nicolao, L. Magni, R. Scattolini. Stability and robustness of nonlinear receding horizon control. In F. Allgöwer, A. Zheng, editors, *Nonlinear Predictive Control*, pages 3–23. Birkhäuser, 2000.

[15] M. Diehl. *Real-Time Optimization for Large Scale Nonlinear Processes*, volume 920 of *Fortschr.-Ber. VDI Reihe 8, Meß-, Steuerungs- und Regelungstechnik*. VDI Verlag, Düsseldorf, 2002. Download also at: http://www.ub.uni-heidelberg.de/archiv/1659/.

[16] M. Diehl, H.G. Bock, J.P. Schlöder. Newton-type methods for the approximate solution of nonlinear programming problems in real-time. In G. Di Pillo, A. Murli, editors, *High Performance Algorithms and Software for Nonlinear Optimization*. Kluwer Academic Publishers B.V., 2002.

[17] M. Diehl, H.G. Bock, J.P. Schlöder, R. Findeisen, Z. Nagy, F. Allgöwer. Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations. *J. Proc. Contr.*, 12:577–585, 2002.

[18] M. Diehl, D.B. Leineweber, A.S. Schäfer. MUSCOD-II Users’ Manual. IWR-Preprint 2001-25, University of Heidelberg, 2001.

[19] M. Diehl, I. Uslu, R. Findeisen, S. Schwarzkopf, F. Allgöwer, H.G. Bock, T. Bürner, E.D. Gilles, A. Kienle, J.P. Schlöder, E. Stein. Real-time optimization of large scale process models: Nonlinear model predictive control of a high purity distillation column. In M. Grötschel, S.O. Krumke, J. Rambau, editors, *Online Optimization of Large Scale Systems: State of the Art*, pages 363–384. Springer, 2001.

[20] R. Findeisen, F. Allgöwer. The quasi-infinite horizon approach to nonlinear model predictive control. In A. Zinober, D. Owens, editors, *Nonlinear and Adaptive Control*, Lecture Notes in Control and Information Sciences, Berlin, 2001. Springer-Verlag.

[21] R. Findeisen, F. Allgöwer, M. Diehl, H.G. Bock, J.P. Schlöder, Z. Nagy. Efficient nonlinear model predictive control. In *6th International Conference on Chemical Process Control – CPC VI*, pages 454–460, 2000.

[22] R. Findeisen, M. Diehl, T. Bürner, F. Allgöwer, H.G. Bock, J.P. Schlöder. Efficient output feedback nonlinear model predictive control. In *Proc. Amer. Contr. Conf.*, pages 4752–4757, Anchorage, AK, 2002.

[23] R. Findeisen, Z. Nagy, M. Diehl, F. Allgöwer, H.G. Bock, J.P. Schlöder. Computational feasibility and performance of nonlinear model predicitve control. In *Proc. 6st European Control Conference ECC’01*, pages 957–961, Porto, Portugal, 2001.

[24] A. Helbig, O. Abel, W. Marquardt. Model predictive control for on-line optimization of semi-batch reactors. In *Proc. Amer. Contr. Conf.*, pages 1695–1699, Philadelphia, 1998.

[25] J.H. Lee, B. Cooley. Recent advances in model predictive control and other related areas. In J.C. Kantor, C.E. Garcia, B. Carnahan, editors, *Fifth International Conference on Chemical Process Control – CPC V*, pages 201–216. American Institute of Chemical Engineers, 1996.

[26] D.B. Leineweber. *Efficient reduced SQP methods for the optimization of chemical processes described by large sparse DAE models*, volume 613 of *Fortschr.-Ber. VDI Reihe 3, Verfahrenstechnik*. VDI Verlag, Düsseldorf, 1999.

[27] D.B. Leineweber, I. Bauer, H.G. Bock, J.P. Schlöder. An efficient multiple shooting based reduced SQP strategy for large-scale dynamic process optimization. Part I: Theoretical aspects. *Comput. Chem. Engng.*, 2002. (in print).

[28] D.B. Leineweber, A. Schäfer, H.G. Bock, J.P. Schlöder. An efficient multiple shooting based reduced sqp strategy for large-scale dynamic process optimization. Part II: Software aspects and applications. *Comput. Chem. Engng.*, 2002. (in print).

[29] D.Q. Mayne. Nonlinear model predictive control: Challenges and opportunities. In F. Allgöwer, A. Zheng, editors, *Nonlinear Predictive Control*, volume 26 of *Progress in Systems Theory*, pages 23–44, Basel, 2000. Birkhäuser.

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

[31] M. Morari, J.H. Lee. Model predicitve control: Past, present and future. *Comput. Chem. Engng.*, 23(4/5):667–682, 1999.

[32] Z. Nagy, R. Findeisen, M. Diehl, F. Allgöwer, H.G. Bock, S. Agachi, J.P. Schlöder, D. Leineweber. Real-time feasibility of nonlinear predictive control for large scale processes – a case study. In *Proc. Amer. Contr. Conf.*, pages 4249–4254, Chicago, 2000.

[33] N.M.C. de Oliveira, L.T. Biegler. Newton-type algorithms for nonlinear process control. Algorithm and stability results. *Automatica*, 31(2):281–286, 1995.

[34] Process Systems Enterprise Ltd., London. *gPROMS Advanced User’s Manual*, 2000.

[35] S.J. Qin, T.A. Badgwell. An overview of nonlinear model predictive control applications. In F. Allgöwer, A. Zheng, editors, *Nonlinear Predictive Control*, pages 369–393. Birkhäuser, 2000.

[36] S.J. Qin, T.A. Badgwell. Review of nonlinear model predictive control applications. In B. Kouvaritakis, M. Cannon, editors, *Nonlinear model predictive control: theory and application*, pages 3–32, London, 2001. The Institute of Electrical Engineers.

[37] L.O. Santos, N.M.C. de Oliveira, L.T. Biegler. Reliable and efficient optimization strategies for nonlinear model predictive control. In *Proc. Fourth IFAC Symposium DYCORD+’95*, pages 33–38, Oxford, 1995. Elsevier Science.

[38] J.P. Schlöder. *Numerische Methoden zur Behandlung hochdimensionaler Aufgaben der Parameteridentifizierung*, volume 187 of *Bonner Mathematische Schriften*. University of Bonn, 1988.

[39] V.H. Schulz, H.G. Bock, M.C. Steinbach. Exploiting invariants in the numerical solution of multipoint boundary value problems for DAEs. *SIAM J. Sci. Comp.*, 19:440–467, 1998.

[40] M.J. Tenny, J.B. Rawlings, R. Bindlish. Feasible real-time nonlinear model predictive control. In *6th International Conference on Chemical Process Control – CPC VI*, AIChE Symposium Series, 2001.

Manuscript received: 11th February 2002.

Dr. Moritz Diehl is mathematics lecturer at the Interdisciplinary Center for Scientific Computing (IWR) of the University of Heidelberg. Main research areas: algorithms for dynamic optimization, nonlinear model predictive control, parameter- and state estimation; applications e. g. in chemical engineering, medicine, robotics, power engineering.

Address: IWR, Universität Heidelberg, INF 368, D-69120 Heidelberg,
Tel.: + 49-(0)6221-54-8831, Fax: + 49-(0)6221-54-5444,
E-Mail: moritz.diehl@iwr.uni-heidelberg.de

Dipl.-Ing. Rolf Findeisen, M.S., is scientific employee at the Institute for Systems Theory in Engineering at the University of Stuttgart. Main research areas: nonlinear model predictive control, optimization based control and state estimation, output feedback control of nonlinear systems, and the application of these methods to chemical and mechanical systems

Address: IST, Universität Stuttgart, Pfaffenwaldring 9, D-70550 Stuttgart,
Tel: +49-(0)711-685-7748, Fax: +49-(0)711-685-7735,
E-Mail: findeise@ist.uni-stuttgart.de

Dipl.-Ing. Stefan Schwarzkopf is scientific employee at the Institute for System Dynamics and Control Engineering at the University of Stuttgart. Main research areas: nonlinear wave propagation phenomena in chemical engineering and their applications for process control.

Address: ISR, Universität Stuttgart, Pfaffenwaldring 9, D-70550 Stuttgart,
Tel: +49-(0)711-685-6299, Fax: +49-(0)711-685-6371,
E-Mail: sts@isr.uni-stuttgart.de

Dr.-Ing. Ilknur Uslu is scientific employee at the Institute for System Dynamics and Control Engineering at the University of Stuttgart. Main research areas: modeling, analysis, and predictive control of chemical processes including the experimental implementation.

Address: ISR, Universität Stuttgart, Pfaffenwaldring 9, D-70550 Stuttgart,
Tel: +49-(0)711-685-6292, Fax: +49-(0)711-685-6371,
E-Mail: ilknur@isr.uni-stuttgart.de

Prof. Dr.-Ing. Frank Allgöwer is director of the Institute for Systems Theory in Engineering at the University of Stuttgart. Main research areas: nonlinear and robust control, predictive control, identification of nonlinear systems, application of modern system and control theoretical methods to various fields including chemical process control, meachatronics, biomedicine and nano technology.

Address: IST, Universität Stuttgart, Pfaffenwaldring 9, D-70550 Stuttgart,
Tel: +49-(0)711-685-7733, Fax: +49-(0)711-685-7735,
E-Mail: allgower@ist.uni-stuttgart.de

Prof. Dr. Dr. h.c. Hans Georg Bock ist director at the IWR of the university of Heidelberg and has the chair for computer science. Main research areas: algorithms for high dimensional nonlinear optimization problems, process optimization, applications e. g. in robotics, fluid dynamics, reaction kinetics and biotechnology.

Address: IWR, Universität Heidelberg, INF 368, D-69120 Heidelberg, Fax:
+49-(0)6221-54-5444, E-Mail: scicom@iwr.uni-heidelberg.de

Prof. Dr.-Ing. Dr. h.c. mult. Ernst Dieter Gilles is director of the Institute for System Dynamics and Control Engineering at the University of Stuttgart and founding director of the Max-Planck-Institute Dynamics of Complex Technical Systems in Magdeburg. Main research areas: modeling, simulation and control of chemical and biochemical engineering processes, network theory of chemical and biological processes, integrated navigation systems for inland vessels.

Address: Max-Planck-Institut für Dynamik komplexer technischer Systeme,
Sandtorstrasse 1, D-39106 Magdeburg,
E-Mail: gilles@mpi-magdeburg.mpg.de

Dr. Johannes P. Schlöder is senior researcher at the IWR of the University of Heidelberg. Main research areas: numerical methods for parameter estimation, optimal experimental design, optimization of nonlinear processes, applications in chemistry, chemical engineering and mechanical engineering.

Address: IWR, Universität Heidelberg, INF 368, D-69120 Heidelberg, Fax:
+49-(0)6221-54-5444, E-Mail: Schloeder@iwr.uni-heidelberg.de