Proceedings of the
44th IEEE Conference on Decision and Control, and
the European Control Conference 2005
Seville, Spain, December 12-15, 2005

Real-Time Iterations for Nonlinear Optimal Feedback Control

Moritz Diehl, Hans Georg Bock, and Johannes P. Schlöder

Abstract— **An efficient Newton-type scheme for the approximate on-line solution of optimization problems as they occur in optimal feedback control is presented. The scheme allows a fast reaction to disturbances by delivering approximations of the exact optimal feedback control which are iteratively refined *during the runtime* of the controlled process. The approximation errors of this *real-time iteration scheme* can be bounded and the solution contracts towards the optimal feedback control. The robustness and excellent real-time performance of the method is demonstrated in a numerical experiment, the control of an unstable system, namely an airborne kite that shall fly loops.**

This paper is a short version of an article that appeared recently in the SIAM Journal on Control and Optimization [1].

I. INTRODUCTION

Feedback control based on the real-time optimization of nonlinear dynamic process models, also referred to as *nonlinear model predictive control (NMPC)*, has attracted increasing attention over the past decade, in particular in chemical engineering [2], [3], [4], [5]. Based on the current system state, feedback is provided by an online optimization of the predicted system behavior, using the mathematical model. The first part of the optimized control trajectory is implemented at the real system, and a sampling time later the optimization procedure is repeated. Among the advantages of this approach are the flexibility provided in formulating the objective and in modelling the process using ordinary or partial differential equations (ODE or PDE), the capability to directly handle equality and inequality constraints, and the possibility to treat large disturbances fast.

One important precondition, however, is the availability of reliable and efficient numerical optimal control algorithms. One particularly successful algorithm that is designed to achieve this aim, the recently developed *real-time iteration* scheme, will be the focus of this paper. In the literature, several suggestions have been made how to adapt off-line optimal control algorithms for use in online optimization. For an overview and comparison of important approaches see e.g. Binder et al. [6]. We particularly mention here the "Newton-type control algorithm" proposed by Li and Biegler [7] and de Oliveira and Biegler [8] and the "feasibility-perturbed SQP" approach to NMPC by Tenny et al. [9]. Both approaches keep even intermediate optimization iterates feasible. This is in contrast to the *simultaneous* dynamic optimization methods, as the collocation method proposed in Biegler [10] or the direct multiple shooting method in Bock et al. [11] and in Santos [12], which allow infeasible state trajectories and are more suitable for trajectory following problems and problems with final state constraints. The real-time iteration scheme belongs to this latter class.

Most approaches in the literature try to solve fast but exactly an optimal control problem. However, if the time scale for feedback is too short for exact computation, some approximations must be made: for this aim an "instantanous control" technique has been proposed in the context of PDE models that approximates the optimal feedback control problem by regarding one future time step only (Choi et al. [13], [14]). By construction, this "greedy" approach to optimal control is based on immediate gains only and neglects future costs; thus it may result in poor performance when future costs matter. A somewhat opposed approach to derive a feedback approximation (formulated for ODE models) is based on a system linearisation along a fixed optimal trajectory over the whole time horizon and can e.g. be found in Krämer-Eis and Bock [15] or Kugelmann and Pesch [16]. The approach works well when the nonlinear system is not too largely disturbed and stays close to the nominal trajectory.

The real-time iteration scheme presented in this paper is a different approximation technique for optimal feedback control. It regards the complete time horizon and performs successive linearizations along (approximately) optimal trajectories to provide feedback approximations. Using these linearizations, it iterates towards the rigorous optimal solutions *during the runtime of the process*. In this way a truly nonlinear optimal feedback control is provided whose accuracy is limited, however, by the time needed to converge to the current optimal solutions. In contrast to a somewhat similar idea mentioned in [7] the real-time iteration scheme is based on the direct multiple shooting method [17], a simultaneous optimization technique, which offers excellent convergence properties in particular for tracking problems and problems with state constraints.

The scheme was introduced in its present form in Diehl et al. [18] going back to ideas presented in Bock et al. [19]. In its actual implementation it is able to treat *differential algebraic equation (DAE)* models (Leineweber [20]), as they often arise in practical applications. It has already been successfully tested for the feedback control of large-scale DAE models with inequality constraints, in particular a binary distillation column [21], [22], [18]. Moreover, it has been applied for the nonlinear model predictive control of a real pilot plant distillation column situated at the *Institut für Systemdynamik und Regelungstechnik (ISR)* of the University of Stuttgart [23], [24], [25].

However, to concentrate on the essential features of the method and on a new proof of contractivity of the scheme,

Interdisciplinary Center for Scientific Computing (IWR), University of Heidelberg, Im Neuenheimer Feld 368, 69120 Heidelberg, Germany.
Corresponding author: M. Diehl m.diehl@iwr.uni-heidelberg.de

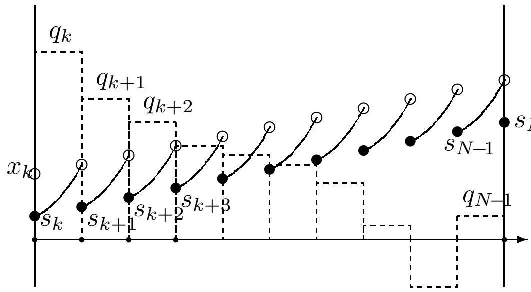

Fig. 1. Problem $P_k(x_k)$: initial value x_k and NLP variables $s_k, \ldots, s_N$ and $q_k, \ldots, q_{N-1}$.

we will restrict the presentation in this paper to ordinary differential equation (ODE) models and optimization problems of a simplified type. Moreover, we will start the paper by regarding (nonlinear) *discrete*-time systems first; the multiple shooting technique, which allows to formulate a discrete-time system from an ODE system, is only introduced later, and very briefly, when the numerical example is presented. All the material is covered in more detail in in [1]; for more information about the real-time iteration scheme we refer to [24], [26], [27], [28].

A. Real-Time Optimal Feedback Control

Throughout this paper, let us consider the following simplified nonlinear controlled discrete-time system:

$$x_{k+1} = f_k(x_k, u_k), \quad k = 0, \ldots, N-1, \tag{1}$$

with system states $x_k \in \mathbb{R}^{n_x}$ and controls $u_k \in \mathbb{R}^{n_u}$. The aim of optimal feedback control is to find controls u_k that depend on the current system state x_k and that are optimal with respect to a specified objective. As time advances, we proceed by solving a sequence of nonlinear programming problems $P_k(x_k)$ on shrinking horizons, each with the current system state x_k as initial value (for a visualisation, see Fig. 1). Let us define $P_k(x_k)$ to be the problem:

$$\min_{\substack{s_k, \ldots, s_N, \\ q_k, \ldots, q_{N-1}}} \quad \sum_{i=k}^{N-1} L_i(s_i, q_i) + E(s_N) \tag{2a}$$

$$\text{subject to} \qquad x_k - s_k = 0, \tag{2b}$$

$$f_i(s_i, q_i) - s_{i+1} = 0, \quad i = k, \ldots, N-1. \tag{2c}$$

The control part $(q_k^*, \ldots, q_{N-1}^*)$ of the solution of problem $P_k(x_k)$ allows to define the optimal feedback control

$$u_k := q_k^*.$$

Note that, due to the dynamic programming property, the optimal control trajectory $(q_0^*, \ldots, q_{N-1}^*)$ of the first problem $P_0(x_0)$ would already give all later closed-loop controls $u_0, u_1, \ldots u_{N-1}$, if the system behaves as predicted by the model. The practical reason to introduce the closed-loop optimal feedback control is, of course, that it allows to optimally respond to disturbances.

We will now assume that we know each initial value x_k only at the time when the corresponding control u_k is already needed for application to the real process, and that the solution time for each problem $P_k(x_k)$ is not negligible compared with the runtime of the process. This is a typical situation in realistic applications: ideally, we would like to have the solution of each problem $P_k(x_k)$ instantaneously, but due to finite computing power this usually cannot be accomplished in practice. In this paper we propose and investigate an efficient Newton-type scheme that allows to approximately solve the optimization problems $P_k(x_k)$ *during the runtime of the real process.*

Remark: In practical applications, inequality path constraints of the form $h(s_i, q_i) \geq 0$, like bounds on controls or states, are of major interest, and are usually present in the formulation of the optimization problems $P_k(x_k)$. For the purpose of this paper we leave such constraints unconsidered, since general convergence results for Newton type methods with changing active sets are difficult to establish. However, we note that in the practical implementation of the real-time iteration scheme they are included and pose no difficulty for the performance of the algorithm.

B. Newton-Type Optimization Methods

In order to solve an optimization problem $P_k(x_k)$, let us first introduce the Lagrange multipliers $\lambda_k, \ldots, \lambda_N$, and define the Lagrangian function $\mathcal{L}^k(\lambda_k, s_k, q_k, \ldots)$ of problem $P_k(x_k)$ to be

$$\begin{aligned} \mathcal{L}^k(\cdot) \quad = \quad & \textstyle\sum_{i=k}^{N-1} L_i(s_i, q_i) + E(s_N) + {\lambda_k}^T(x_k - s_k) \\ & + \textstyle\sum_{i=k}^{N-1} {\lambda_{i+1}}^T(f_i(s_i, q_i) - s_{i+1}) \end{aligned} \tag{3}$$

Summarizing all variables in a vector

$$y = (\lambda_k, s_k, q_k, \lambda_{k+1}, s_{k+1}, q_{k+1}, \ldots, \lambda_N, s_N), \tag{4}$$

we can formulate necessary optimality conditions of first order (also called *Karush-Kuhn-Tucker* conditions):

$$\nabla_y \mathcal{L}^k(y) = 0. \tag{5}$$

To solve this system, a Newton-type method would start at an initial guess y_0 and compute a sequence of iterates $y_1, y_2, \ldots$ according to $y_{i+1} = y_i + \Delta y_i$, where each Δy_i is the solution of the linear system

$$\nabla_y \mathcal{L}^k(y_i) + J^k(y_i)\Delta y_i = 0. \tag{6}$$

Here, $J^k(y)$ is an approximation of $\nabla_y^2 \mathcal{L}^k(y)$ (in this paper we use a Gauss-Newton approximation, for details see [24], [1]). Note that this second derivative matrix (resp. its approximation) is independent of x_k, and that it has a a block sparse structure [17].

It is well known that the Newton-type scheme (6) for the solution of (5) converges in a neighborhood $D_k \subset \mathbb{R}^{n_k}$ of a solution y_*^k that satisfies the second order sufficient conditions for optimality of problem $P_k(x_k)$, if $J^k(y)$ approximates $\nabla_y^2 \mathcal{L}^k(y)$ sufficiently well on D_k.

II. Real-Time Iterations

Let us now consider the real-time scenario described in Section I-A, where we want to solve the sequence of optimization problems $P_k(x_k)$, but where we do not have the time to iterate each problem to convergence. Let us more specifically assume that each Newton-type iteration needs exactly as much computation time as corresponds to the time that the real process needs for the transition from one system state to the next. Thus, we can only perform *one single Newton-type iteration* for each problem $P_k(x_k)$ and then we have to proceed already to the next problem $P_{k+1}(x_{k+1})$. The *real-time iteration* scheme that we will investigate here is based on a carefully designed transition between subsequent problems. After an initial disturbance, it subsequently delivers approximations u_k for the optimal feedback control that become better and better, if no further disturbance occurs, as will be shown in Section III.

It turns out that the computations of the real-time iteration belonging to problem $P_k(x_k)$ can largely be prepared *without knowledge of the value of* x_k, so that we can assume that the approximation u_k of the optimal feedback control is instantly available at the time that x_k is known. However, after this feedback has been delivered, we need to *prepare* the next real-time iteration (belonging to problem $P_{k+1}(x_{k+1})$) which needs the full computing time.

In the framework for optimal feedback control on shrinking horizons (2), we reduce the number of remaining intervals from one problem $P_k(x_k)$ to the next $P_{k+1}(x_{k+1})$, in order to keep pace with the process development. Therefore we have to perform real-time iterates in primal-dual variable spaces $\mathbb{R}^{n_0} \supset \ldots \supset \mathbb{R}^{n_k} \supset \mathbb{R}^{n_{k+1}} \supset \ldots \supset \mathbb{R}^{n_{N-1}}$ of different size. Let us denote by Π^{k+1} the projection from $\mathbb{R}^{n_k}$ onto $\mathbb{R}^{n_{k+1}}$, i.e., if $y = (\lambda_k, s_k, q_k, \tilde{y}) \in \mathbb{R}^{n_k}$ then $\Pi^{k+1} y = \tilde{y} \in \mathbb{R}^{n_{k+1}}$.

A. The Real-Time Iteration Algorithm

Let us assume that we have an initial guess $y^0 \in \mathbb{R}^{n_0}$ for the primal-dual variables of problem $P_0(\cdot)$. We set the iteration index k to zero and perform the following steps:

1) Preparation: Based on the initial guess $y^k \in \mathbb{R}^{n_k}$, compute the vector $\nabla_y \mathcal{L}^k(y^k)$ and the matrix $J^k(y^k)$: Note that $J^k(y^k)$ is completely independent of the value of x_k, and that of the vector $\nabla_y \mathcal{L}^k(y^k)$ only the first component ($\nabla_{\lambda_k} \mathcal{L}^k = x_k - s_k$) depends on x_k. This component will only be needed in the second step. Therefore, prepare the linear algebra computation of $J^k(y^k)^{-1} \nabla_y \mathcal{L}^k(y^k)$ as much as possible without knowledge of the value of x_k (a detailed description how this can be achieved is given in [18] or [24]).
2) Feedback Response: At the time when x_k is exactly known, finish the computation of the step vector $\Delta y^k = -J^k(y)^{-1} \nabla_y \mathcal{L}^k(y^k)$ and give the control $u_k := q_k + \Delta q_k$ immediately to the real system.
3) Transition: If $k = N-1$ stop. Otherwise, compute the next initial guess y^{k+1} by adding the step vector to y^k and "shrinking" the resulting variable vector onto $\mathbb{R}^{n_{k+1}}$, i.e., $y^{k+1} := \Pi^{k+1}(y^k + \Delta y^k)$. Set $k = k+1$ and go to 1.

Note that after one iteration belonging to system state x_k we expect the next system state to be $x_{k+1} = f_k(x_k, u_k)$, but that this may not be true due to disturbances. The scheme allows an immediate feedback to such disturbances, due to the separation of steps 1 and 2. This separation is only possible because we do *not* require the guess of initial value, s_k, to be equal to the real initial value, x_k. This formulation may be regarded an *initial value embedding* of each problem into the manifold of perturbed problems and is crucial for the success of the method in practice.

We will in the following investigate the contraction properties of the real-time iteration scheme. Though a principal advantage of the scheme lies in this immediate response to disturbances, we will investigate contractivity only under the assumption that after an initial disturbance the system behaves according to the model. This is analogous to the notion of "nominal stability" for an infinite horizon steady state tracking problem.

III. Contractivity of the Real-Time Iterations

In this subsection we investigate the contraction properties of the real-time iteration scheme. The system starts at an initial state x_0, and the real-time algorithm is initialized with an initial guess $y^0 \in D_0 \subset \mathbb{R}^{n_0}$. If we assume that after an initial disturbance the system behaves according to its deterministic dynamics (1), the real-time iteration scheme obeys to the following *system-optimizer-dynamics*:

$$y^{k+1} = \Pi^{k+1}(y^k - J^k(y^k)^{-1} \nabla_y \mathcal{L}^k(y^k)) \qquad (7)$$

$$x_{k+1} = f_k(x_k, q_k^{k+1}). \qquad (8)$$

The difficulty in investigating the contractivity of the real-time iterations stems from the fact that the system and the optimizer mutually interact: (7) depends via $\nabla_y \mathcal{L}^k$ on the current system state x_k, cf. (3). Conversely, (8) depends on y^{k+1} by using its third component as feedback control, cf. (4).

Let define the projections D_k of the neighborhood D_0 onto the primal-dual subspaces $\mathbb{R}^{n_k}$, i.e., $D_{k+1} := \Pi^{k+1} D_k$. We will in the following make use of vector and corresponding matrix norms $\|\cdot\|_k$ defined on the subspaces $\mathbb{R}^{n_k}$. These norms are assumed to be compatible in the sense that $\|\Pi^{k+1} y\|_{k+1} \le \|y\|_k$ and that $\|{\Pi^{k+1}}^T \tilde{y}\|_k = \|\tilde{y}\|_{k+1}$. We will use the following technical assumption.

Assumption 1: For all $k = 0, \ldots, N$ it we have $\mathcal{L}^k \in C^2(D_k, \mathbb{R})$ and $J^k \in C^0(D_k, \mathbb{R}^{n_k \times n_k})$ and J^k have a bounded inverse $(J^k)^{-1}$. Furthermore, there exists a $\kappa < 1$ and an $\omega < \infty$ such that for each $k = 0, \ldots, N$, and all $y', y \in D_k$, $\Delta y = y' - y$ and all $t \in [0,1]$ it holds that

$$\left\| \left(J^k(y')\right)^{-1} \left(J^k(y + t\Delta y) - \nabla_y^2 \mathcal{L}^k(y + t\Delta y)\right) \Delta y \right\|_k \le \kappa \|\Delta y\|_k$$

and that

$$\left\| \left(J^k(y')\right)^{-1} \left(J^k(y + t\Delta y) - J^k(y)\right) \Delta y \right\|_k \le \omega t \|\Delta y\|_k^2,$$

and such that for each $k = 0, \dots, N-1$ it holds that

$$\left\| \left(J^{k+1}(\Pi^{k+1} y')\right)^{-1} \Pi^{k+1} \left(J^k(y + t\Delta y) - J^k(y)\right) \Delta y \right\|_{k+1} \leq \omega t \|\Delta y\|_k^2.$$

In the folowing, we will use the abbreviations $\Delta y^k := J^k(y^k)^{-1} \nabla_y \mathcal{L}^k(y^k)$ and $\delta_k := \kappa + \frac{\omega}{2} \|\Delta y^k\|_k$.

Theorem 3.1 (Local Contractivity): Suppose Assumption 1, $\delta_0 < 1$, and that the ball with radius $r = \frac{\|\Delta y^0\|_0}{1-\delta_0}$ around y_0 is completely contained in D_0. Then the real-time iterates $y^0, \dots, y^N$ generated by (7) – with x_k generated by (8) – are well defined and satisfy the contraction condition

$$\|\Delta y^{k+1}\|_{k+1} \leq \delta_k \|\Delta y^k\|_k \leq \delta_0 \|\Delta y^k\|_k. \qquad (10)$$

Furthermore, the iterates y^k approach the exact stationary points y_*^k of the corresponding problems $P_k(x_k)$:

$$\|y^k - y_*^k\|_k \leq \frac{\|\Delta y^k\|_k}{1-\delta_k} \leq \frac{(\delta_0)^k \|\Delta y^0\|_0}{1-\delta_0}. \qquad (11)$$

The proof of the Theorem and a bound on the loss of optimality can be found in [1].

IV. Control of a Looping Kite

In order to demonstrate the versatility of the proposed real-time iteration scheme we present here the control of an airborne kite as a periodic control example. The kite is held by two lines which allow to control the roll angle ψ of the kite. By pulling one line the kite will turn in the direction of the line being pulled. This allows an experienced kite pilot to fly loops or similar figures. The aim of our automatic control is to make the kite fly a figure that may be called a "lying eight", with a cycle time of 8 seconds (see Fig. 2). The corresponding orbit is not open-loop stable, so that feedback has to be applied during the flight – we will show simulation results where our proposed real-time iteration scheme is used to control the kite, starting at a largely disturbed initial state x_0, over three periods, with a sampling time of one second. The kite model can be found in [24], [27], [1]. The kite position is described by polar coordinates θ and ϕ, where θ is the angle that the lines form with the vertical. Defining the system state $\xi := (\theta, \phi, \dot{\theta}, \dot{\phi})^T$ and the control $u := \psi$ we can summarize the four system equations in the short form $\dot{\xi} = \hat{f}(\xi, u)$.

A periodic orbit was determined that can be characterized as a "lying eight" and which is depicted as a (ϕ, θ)-plot in Fig. 2. The open-loop system is highly unstable in the periodic orbit [24].

We want to mention that the kite model and the periodic orbit may serve as a challenging benchmark problem for nonlinear periodic control and are available in MATLAB/SIMULINK format [29].

A. The Optimal Control Problem

Given an arbitrary initial state x_0 (that we do not know in advance) we want the kite to fly 3 times the figure of Fig. 2, on a time horizon of $3T = 24$ seconds. By using the figure as a reference orbit, we formulate an optimal control problem which has the objective to bring

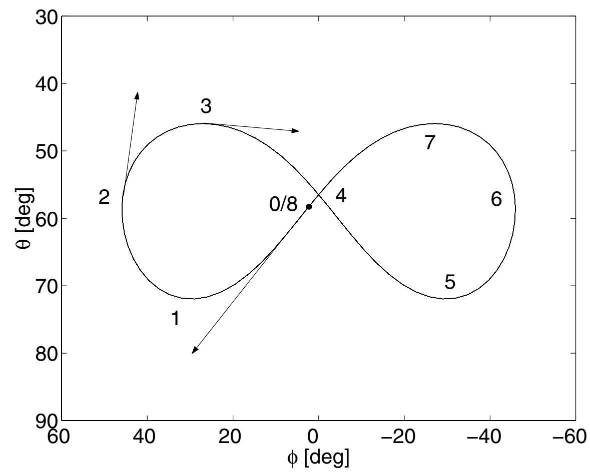

Fig. 2. Periodic orbit plotted in the (ϕ, θ)-plane, as seen by the kite pilot. The dots separate intervals of one second.

the system close to the reference orbit. For this aim we define a Lagrange term of least squares type $L(\xi, u, t) := \|\xi - \xi_r(t)\|_Q^2 + \|u - u_r(t)\|_R^2$ with diagonal weighting matrices $Q := \text{diag}(0.4,\ 1,\ \text{s}^2,\ \text{s}^2)\frac{1}{\text{s}}$ and $R := 1.0 \cdot 10^{-2} \text{deg}^{-2} \text{s}^{-1}$. Using these definitions, we formulate the following optimal control problem on the time horizon of interest $[0, 3T]$:

$$\min_{u(\cdot),\xi(\cdot)} \int_0^{3T} L(\xi(t), u(t), t)\, dt \qquad (12)$$

$$\text{subject to} \quad \dot{\xi}(t) = \hat{f}(\xi(t), u(t)), \qquad \forall t \in [0, 3T],$$
$$\xi(0) = x_0.$$

In order to reformulate the above continuous optimal control problem into a discrete-time optimal control problem, we use the *direct multiple shooting* technique, originally due to Plitt and Bock [30], [17]: We divide the time horizon into $N = 24$ invervals $[t_i, t_{i+1}]$, each of one second length, and introduce a locally constant control representation $q_0, q_1, \dots, q_{N-1}$, as well as artificial initial values $s_0, \dots, s_N$, as depicted in Fig. 1. On each of these intervals we solve an following initial value problem yielding a trajectory piece $\xi_i(t; s_i, q_i)$. This leads naturally to a discrete-time system as in (2c) with transition function $f_i(s_i, q_i) := \xi_i(t_{i+1}; s_i, q_i)$. Using the multiple shooting formulation, we can thus transform the continuous time optimization problem (12) into a nonlinear programming problem of the type (2).

B. A Real-Time Scenario

In the following real-time scenario we assume that the Newton-type optimizer is initialized with the reference trajectory itself, i.e. $y^0 := (\lambda_0^0, s_0^0, q_0^0, \dots, \lambda_N^0, s_N^0)$, where $\lambda_i^0 := 0$, and $s_i^0 := \xi_r(t_i)$ and $q_i^0 := \frac{1}{t_{i+1}-t_i} \int_{t_i}^{t_{i+1}} u_r(t)\, dt$ are the corresponding values of the periodic reference solution. This y^0 is (nearly) identical to the solution of the problem $P_0(\xi_r(t_0))$. At the time $t_0 = 0$, when the actual value of x_0 is known, we start the iterations as described in Section II-A by solving the first prepared linear system $\Delta y^0 = -J^0(y^0)^{-1} \nabla_y \mathcal{L}^0(y^0)$ (step 2), and give the first

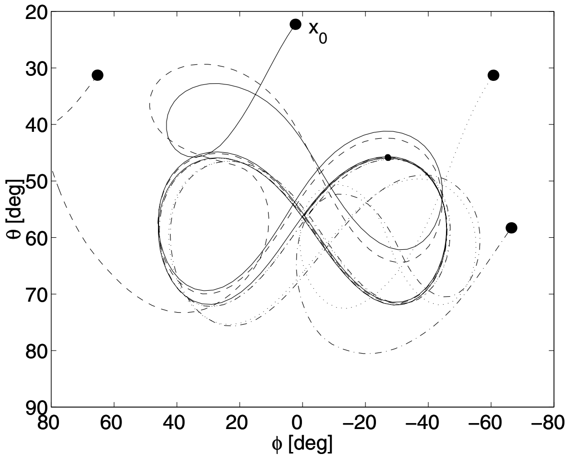

Fig. 3. Closed-loop trajectories resulting from the real-time iteration scheme, for different initial values x_0. The kite never crashes onto the ground (θ =90 degree).

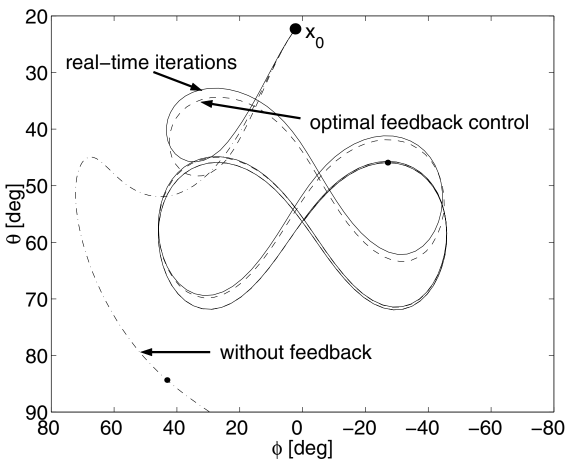

Fig. 4. Comparison of trajectories resulting from the real-time iteration scheme (solid line), the open-loop controls without feedback (dash-dotted line, crashing onto the ground), and a hypothetical optimal feedback control (dashed line).

control $u_0 := q_0^0 + \Delta q_0^0$ immediately to the system. Then we shrink the problem (step 3) and prepare the iteration for the following one (step 1). As we assume no further disturbances, the new initial value is $x_1 = f_0(x_0, u_0) = \xi_0(t_1; x_0, u_0)$ resulting from the (continuous) system dynamics. This cycle is repeated until the $N = 24$ intervals are over.

The corresponding trajectory resulting from the real-time iteration scheme for different initial values x_0 is shown in Fig. 3, as (ϕ, θ)-plots. For all scenarios, the third loop is already close to the reference trajectory.

In Fig. 4 we compare the result of the real-time iteration scheme with the open-loop system dynamics without feedback (dash-dotted line) and with a hypothetical optimal feedback control (dashed line), for one initial value x_0. The open-loop system, where the controls are simply taken from the reference (and initialization) trajectory ($u_k := q_k^0$), crashes after 7 seconds onto the ground.

Note that the computation of the hypothetical optimal feedback control needs about 8 seconds on a Compaq Alpha XP1000 workstation. This delay means that *no* feedback can be applied in the meantime, so the kite would have crashed onto the ground before the first response is computed.

In contrast to this, the first feedback control u_0 of the real-time iteration scheme was available within only 0.05 seconds delay after time t_0 (for the computations of step 2). The sampling time of one second until the next feedback can be applied was necessary to prepare the following real-time iteration (to be exact, step 1 needed always below 0.8 seconds). The comparison with the hypothetical optimal feedback control shows that the real-time iteration scheme delivers a quite good approximation even for this challenging nonlinear and unstable test example with largely disturbed initial values.

V. Conclusions

We have presented a recently developed Newton-type method for the real-time optimization of nonlinear processes, and have given a contractivity result that bounds the errors compared to optimal feedback control. In a numerical case study, the real-time control of an airborne kite, we have demonstrated the practical applicability of the method for a challenging nonlinear control example.

The "real-time iteration" scheme is based on the direct multiple shooting method, which offers several advantages in the context of real-time optimal control, among them the ability to efficiently initialize subsequent optimization problems, to treat highly nonlinear and unstable systems, and to deal efficiently with path constraints. The most important feature of the real-time iteration scheme is a dovetailing of the solution iterations with the process development which allows to reduce sampling times to a minimum, but maintains all advantages of a fully nonlinear treatment of the optimization problems. A separation of the computations in each real-time iteration into a *preparation phase* and a *feedback response phase* can be realized. The feedback phase is typically orders of magnitude shorter than the preparation phase, and allows to deliver an immediate feedback that takes all linearized constraints into account.

The contractivity of the scheme is valid under mild conditions that are nearly identical to the sufficient conditions for convergence of off-line Newton-type methods. The real-time iterates geometrically approach the exact optimal solutions during the runtime of the process.

The control aim for the kite example is to steer the kite into a periodic orbit, a "lying eight", starting from any arbitrary initial state. The initial state is only known at the moment that the first control needs already to be applied – the real-time iteration scheme delivers linearized feedback nearly without delay, and provides a newly linearized feedback after each sampling time of one second, leading to a fully nonlinear optimization, and always being prepared to react to further disturbances. The scheme shows an excellent closed-loop performance for this highly nonlinear and unstable system, and compares well to a hypothetical exact optimal feedback control.

The real-time iteration scheme has also been applied for nonlinear model predictive control of a real pilot plant distillation column described by a stiff DAE model with over 200 system states, allowing feedback sampling times of only 20 seconds [25].

ACKNOWLEDGEMENTS

This work was supported by the *Deutsche Forschungsgemeinschaft (DFG)* within the priority program 469 "Online-Optimization of Large Scale Systems" and within the research project "Optimization based control of chemical processes" under grant Bo 864/10.

REFERENCES

[1] M. Diehl, H. Bock, and J. Schlöder, "A real-time iteration scheme for nonlinear optimization in optimal feedback control," *SIAM J. Control Optim.*, vol. 43, no. 5, pp. 1714–1736, 2005.

[2] L. Biegler and J. Rawlings, "Optimization approaches to nonlinear model predictive control," in *Proc. 4th International Conference on Chemical Process Control - CPC IV*, W. Ray and Y. Arkun, Eds. AIChE, CACHE, 1991, pp. 543–571.

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

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

[5] B. Kouvaritakis and M. Cannon, Eds., *Non-Linear Predictive Control: Theory and Practice*, 1st ed. IEE Publishing, 2001, iSBN: 0852969848.

[6] T. Binder, L. Blank, H. Bock, R. Bulirsch, W. Dahmen, M. Diehl, T. Kronseder, W. Marquardt, J. Schlöder, and O. Stryk, "Introduction to model based optimization of chemical processes on moving horizons," in *Online Optimization of Large Scale Systems: State of the Art*, M. Grötschel, S. Krumke, and J. Rambau, Eds. Springer, 2001, pp. 295–340, http://www.zib.de/dfg-echtzeit/Publikationen/Preprints/Preprint-01-15.html.

[7] W. Li and L. Biegler, "Multistep, Newton-type control strategies for constrained nonlinear processes," *Chem. Eng. Res. Des.*, vol. 67, pp. 562–577, 1989.

[8] N. Oliveira and L. Biegler, "An extension of Newton-type algorithms for nonlinear process control," *Automatica*, vol. 31, no. 2, pp. 281–286, 1995.

[9] M. J. Tenny, S. J. Wright, and J. B. Rawlings, "Nonlinear model predictive control via feasibility-perturbed sequential quadratic programming," *Comp. Optim. Appl.*, vol. 28, no. 1, pp. 87–121, April 2004.

[10] L. Biegler, "Efficient solution of dynamic optimization and NMPC problems," in *Nonlinear Predictive Control*, ser. Progress in Systems Theory, F. Allgöwer and A. Zheng, Eds., vol. 26. Basel: Birkhäuser, 2000, pp. 219–244.

[11] H. Bock, M. Diehl, D. Leineweber, and J. Schlöder, "Efficient direct multiple shooting in nonlinear model predictive control," in *Scientific Computing in Chemical Engineering II*, F. Keil, W. Mackens, H. Voß, and J. Werther, Eds., vol. 2. Berlin: Springer, 1999, pp. 218–227.

[12] L. Santos, "Multivariable predictive control of nonlinear chemical processes," Ph.D. dissertation, Universidade de Coimbra, 2000.

[13] H. Choi, R. Temam, P. Moin, and J. Kim, "Feedback control for unsteady flow and its application to the stochastic burgers equation," *J. Fluid Mech.*, vol. 253, pp. 509–543, 1993.

[14] H. Choi, M. Hinze, and K. Kunisch, "Instantaneous control of backward-facing-step flows," *Applied Numerical Mathematics*, vol. 31, pp. 133–158, 1999.

[15] P. Krämer-Eis and H. Bock, "Numerical treatment of state and control constraints in the computation of feedback laws for nonlinear control problems," in *Large Scale Scientific Computing*, P. D. et al., Ed. Birkhäuser, 1987, pp. 287–306.

[16] B. Kugelmann and H. Pesch, "New general guidance method in constrained optimal control, part 1: Numerical method," *J. Optimization Theory and Application*, vol. 67, no. 3, pp. 421–435, 1990.

[17] H. Bock and K. Plitt, "A multiple shooting algorithm for direct solution of optimal control problems," in *Proc. 9th IFAC World Congress Budapest*. Pergamon Press, 1984, pp. 243–247.

[18] M. Diehl, H. Bock, J. Schlöder, R. Findeisen, Z. Nagy, and F. Allgöwer, "Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations," *J. Proc. Contr.*, vol. 12, no. 4, pp. 577–585, 2002.

[19] H. Bock, M. Diehl, D. Leineweber, and J. Schlöder, "A direct multiple shooting method for real-time optimization of nonlinear DAE processes," in *Nonlinear Predictive Control*, ser. Progress in Systems Theory, F. Allgöwer and A. Zheng, Eds., vol. 26. Basel: Birkhäuser, 2000, pp. 246–267.

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

[21] H. Bock, M. Diehl, J. Schlöder, F. Allgöwer, R. Findeisen, and Z. Nagy, "Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations," in *ADCHEM2000 - International Symposium on Advanced Control of Chemical Processes*, vol. 2, Pisa, 2000, pp. 695–703.

[22] Z. Nagy, R. Findeisen, M. Diehl, F. Allgöwer, H. Bock, S. Agachi, J. Schlöder, and D. Leineweber, "Real-time feasibility of nonlinear predictive control for large scale processes – a case study," in *Proc. of ACC 2000*, Chicago, 2000, pp. 4249–4254.

[23] M. Diehl, I. Uslu, R. Findeisen, S. Schwarzkopf, F. Allgöwer, H. Bock, T. Bürner, E. Gilles, A. Kienle, J. Schlöder, and E. Stein, "Real-time optimization for large scale processes: Nonlinear model predictive control of a high purity distillation column," in *Online Optimization of Large Scale Systems: State of the Art*, M. Grötschel, S. O. Krumke, and J. Rambau, Eds. Springer, 2001, pp. 363–384, iSBN 3-540-42459-8, download at: http://www.zib.de/dfg-echtzeit/Publikationen/Preprints/Preprint-01-16.html.

[24] M. Diehl, *Real-Time Optimization for Large Scale Nonlinear Processes*, ser. Fortschr.-Ber. VDI Reihe 8, Mess-, Steuerungs- und Regelungstechnik. Düsseldorf: VDI Verlag, 2002, vol. 920, http://www.ub.uni-heidelberg.de/archiv/1659/.

[25] M. Diehl, R. Findeisen, S. Schwarzkopf, I. Uslu, F. Allgöwer, H. Bock, E. Gilles, and J. Schlöder, "An efficient algorithm for nonlinear model predictive control of large-scale systems. Part II: Application to a distillation column," *Automatisierungstechnik*, vol. 51, no. 1, pp. 22–29, 2003.

[26] ——, "An efficient algorithm for nonlinear model predictive control of large-scale systems. Part I: Description of the method," *Automatisierungstechnik*, vol. 50, no. 12, pp. 557–567, 2002.

[27] M. Diehl, L. Magni, and G. D. Nicolao, "Online NMPC of unstable periodic systems using approximate infinite horizon closed loop costing," *Annual Reviews in Control*, vol. 28, pp. 37–45, 2004.

[28] M. Diehl, R. Findeisen, F. Allgöwer, H. Bock, and J. Schlöder, "Nominal stability of the real-time iteration scheme for nonlinear model predictive control," *IEE Proc.-Control Theory Appl.*, vol. 152, no. 3, pp. 296–308, May 2005.

[29] M. Diehl, "The kite benchmark problem homepage," http://www.iwr.uni-heidelberg.de/˜Moritz.Diehl/KITE/kite.html, September 2003.

[30] K. Plitt, "Ein superlinear konvergentes Mehrzielverfahren zur direkten Berechnung beschränkter optimaler Steuerungen," Master's thesis, University of Bonn, 1981.

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

[32] I. Bauer, "Numerische Verfahren zur Lösung von Anfangswertaufgaben und zur Generierung von ersten und zweiten Ableitungen mit Anwendungen bei Optimierungsaufgaben in Chemie und Verfahrenstechnik," Ph.D. dissertation, University of Heidelberg, 1999, download at: http://www.ub.uni-heidelberg.de/archiv/1513.

[33] H. Bock, "Numerical treatment of inverse problems in chemical reaction kinetics," in *Modelling of Chemical Reaction Systems*, ser. Springer Series in Chemical Physics, K. Ebert, P. Deuflhard, and W. Jäger, Eds. Heidelberg: Springer, 1981, vol. 18, pp. 102–125.