SIAM J. CONTROL OPTIM.
Vol. 43, No. 5, pp. 1714–1736

A REAL-TIME ITERATION SCHEME FOR NONLINEAR OPTIMIZATION IN 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 contractivity of this *real-time iteration* scheme is proven, and a bound on the loss of optimality—compared with the theoretical optimal solution—is given. 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.

Key words. direct multiple shooting, Newton-type optimization, optimal feedback control, nonlinear model predictive control, ordinary differential equations, real-time optimization

AMS subject classifications. 34B15, 34H05, 49N35, 49N90, 90C06, 90C30, 90C55, 90C59, 90C90, 93C55

DOI. 10.1137/S0363012902400713

1. 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, particularly in chemical engineering [4, 27, 1, 28]. 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 modeling the process using ordinary or partial differential equations (ODEs or PDEs), the capability of directly handling equality and inequality constraints, and the possibility of treating large disturbances quickly.

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 on how to adapt off-line optimal control algorithms for use in on-line 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 [32] and de Oliveira and Biegler [15] and the "feasibility-perturbed SQP" approach to NMPC by Tenny, Wright, and Rawlings [38]. 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 [5] or the direct multiple shooting method in Bock et al. [7] and in Santos [37], which allow

*Received by the editors July 30, 2002; accepted for publication (in revised form) April 12, 2004; published electronically March 11, 2005. 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."
http://www.siam.org/journals/sicon/43-5/40071.html

†Interdisciplinary Center for Scientific Computing (IWR), University of Heidelberg, Im Neuenheimer Feld 368, 69120 Heidelberg, Germany (m.diehl@iwr.uni-heidelberg.de, scicom@iwr.uni-heidelberg.de, johannes.schloeder@iwr.uni-heidelberg.de).

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 quickly 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 "instantaneous 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. [14, 13]). 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 linearization along a fixed optimal trajectory over the whole time horizon and can, e.g., be found in Krämer-Eis and Bock [29] or Kugelmann and Pesch [30]. 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 toward 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 [32], the real-time iteration scheme is based on the direct multiple shooting method [12], a simultaneous optimization technique, which offers excellent convergence properties, particularly for tracking problems and problems with state constraints.

The scheme was introduced in its present form in Diehl et al. [19] going back to ideas presented in Bock et al. [10]. In its actual implementation it is able to treat *differential algebraic equation (DAE)* models (Leineweber [31]), 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, particularly a binary distillation column [11, 33, 19]. Moreover, it has been applied for the NMPC of a real pilot plant distillation column situated at the *Institut für Systemdynamik und Regelungstechnik (ISR)* of the University of Stuttgart [25, 16, 21].

However, to concentrate on the essential features of the method and on a new proof of contractivity of the scheme, we will restrict the presentation in this paper to 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 us to formulate a discrete-time system from an ODE system, is only introduced later, and very briefly, when the numerical example is presented. Part of the material is also covered in [18]; for technical details about the real-time iteration scheme we refer to [16, 20].

1.1. Real-time optimal feedback control. Throughout this paper, let us consider the simplified nonlinear controlled discrete-time system

$$x_{k+1} = f_k(x_k, u_k), \quad k = 0, \dots, N-1, \tag{1.1}$$

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

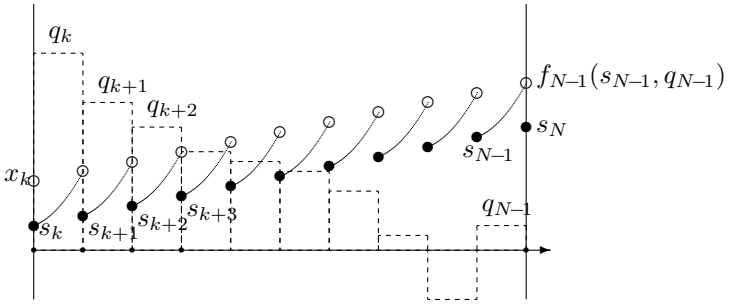

FIG. 1.1. *Problem $P_k(x_k)$: Initial value x_k and problem variables $s_k, \dots, s_N$ and $q_k, \dots, q_{N-1}$.*

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 visualization, see Figure 1.1). Let us define $P_k(x_k)$ to be the problem

$$\min_{\substack{s_k,\dots,s_N,\\ q_k,\dots,q_{N-1}}} \quad \sum_{i=k}^{N-1} L_i(s_i, q_i) + E(s_N) \tag{1.2a}$$

subject to

$$x_k - s_k = 0, \tag{1.2b}$$

$$f_i(s_i, q_i) - s_{i+1} = 0, \quad i = k, \dots, N-1. \tag{1.2c}$$

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

$$u_k := q_k^*.$$

Note that, due to the dynamic programming property, the optimal control trajectory $(q_0^*, \dots, q_{N-1}^*)$ of the first problem $P_0(x_0)$ would already give all later closed-loop controls $u_0, u_1, \dots u_{N-1}$, 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 us 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 us 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.

1.2. Overview. The paper is organized as follows:

- In section 2 we give a review of Newton-type optimization methods for the solution of optimal control problems of type (1.2) and discuss the problem structure.
- The real-time iteration scheme is presented in section 3 building on the previously introduced Newton-type methods. It performs only one Newton-type iteration per optimization problem $P_k(x_k)$, applies the obtained feedback control to the real system, and then proceeds already to the following problem, $P_{k+1}(x_{k+1})$, until the end of the horizon is reached. This allows a particularly fast reaction to disturbances.
- A new contractivity result for the scheme is presented and proven in section 4. The theorem guarantees that the real-time iteration scheme is contracting under mild conditions and delivers approximations to the optimal feedback control with diminishing error.
- Based on the contractivity result, a bound on the loss of optimality of the scheme compared to exact optimal feedback control is established in section 5.
- In section 6 we finally present a numerical example, the real-time control of a kite that shall start to fly loops. A new kite model is developed and a periodic reference orbit is defined. The optimal control problem is to steer the kite into the periodic orbit, starting at an a priori unknown initial value. This example, though of small state dimension, is particularly challenging, because the system is highly nonlinear and unstable.

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

$$\mathcal{L}^k(\cdot) = \sum_{i=k}^{N-1} L_i(s_i, q_i) + E(s_N) + {\lambda_k}^T(x_k - s_k) + \sum_{i=k}^{N-1} {\lambda_{i+1}}^T(f_i(s_i, q_i) - s_{i+1}).$$

Summarizing all variables in a vector $y := (\lambda_k, s_k, q_k, \lambda_{k+1}, s_{k+1}, q_{k+1}, \dots, \lambda_N, s_N) \in \mathbb{R}^{n_k}$,[1] we can formulate necessary optimality conditions of first order (also called *Karush–Kuhn–Tucker* conditions):

$$\nabla_y \mathcal{L}^k(y) = 0. \tag{2.1}$$

To solve this system, the exact full-step Newton–Raphson method would start at an initial guess y_0 and compute a sequence of iterates $y_1, y_2, \dots$ according to

$$y_{i+1} = y_i + \Delta y_i, \tag{2.2}$$

where each Δy_i is the solution of the linearized system

$$\nabla_y \mathcal{L}^k(y_i) + \nabla_y^2 \mathcal{L}^k(y_i)\Delta y_i = 0. \tag{2.3}$$

The Newton-type methods considered in this paper differ from the exact Newton–Raphson method in the way that a part of the exact second derivative $\nabla_y^2 \mathcal{L}^k$, namely, the Hessian $\nabla^2_{(q,s)} \mathcal{L}^k$, is replaced by a (symmetric) approximation. We denote the resulting approximation of $\nabla_y^2 \mathcal{L}^k(y)$ by $J^k(y)$. For our Newton-type method, (2.3) is replaced by the approximation

$$\nabla_y \mathcal{L}^k(y_i) + J^k(y_i)\Delta y_i = 0. \tag{2.4}$$

[1] For simplicity, we omit the index k for the variable y and implicitly assume that $y \in \mathbb{R}^{n_k}$ when not specified otherwise. Note that $n_k = (2n_x + n_u)(N - k) + 2n_x$.

The matrix $\nabla_y^2 \mathcal{L}^k$, respectively, its approximation J^k, is often referred to as the *Karush–Kuhn–Tucker (KKT)* matrix.

2.1. Structure of the Karush–Kuhn–Tucker matrix. The Lagrangian function $\mathcal{L}^k$ of the optimal control problem is *partially separable* [12], and its second derivative has a block diagonal structure:

$$\nabla_y^2 \mathcal{L}^k(y) = \begin{pmatrix}
 & -\mathbb{I} & & & & & & & & \\
-\mathbb{I} & Q_k & M_k & A_k^T & & & & & & \\
 & M_k^T & R_k & B_k^T & & & & & & \\
 & A_k & B_k & & -\mathbb{I} & & & & & \\
 & & & -\mathbb{I} & Q_{k+1} & M_{k+1} & A_{k+1}^T & & & \\
 & & & & M_{k+1}^T & R_{k+1} & B_{k+1}^T & & & \\
 & & & & A_{k+1} & B_{k+1} & & & & \\
 & & & & & & & \ddots & & \\
 & & & & & & \ddots & Q_{N-1} & M_{N-1} & A_{N-1}^T \\
 & & & & & & & M_{N-1}^T & R_{N-1} & B_{N-1}^T \\
 & & & & & & & A_{N-1} & B_{N-1} & \\
 & & & & & & & & & -\mathbb{I} \\
 & & & & & & & & -\mathbb{I} & Q_N
\end{pmatrix},$$

where we have set

$$A_i := \frac{\partial f_i}{\partial s_i}, \quad B_i := \frac{\partial f_i}{\partial q_i}, \quad \begin{pmatrix} Q_i & M_i \\ M_i^T & R_i \end{pmatrix} := \nabla^2_{(s_i,q_i)} \mathcal{L}^k, \quad \text{and} \quad Q_N := \nabla^2_{s_N} \mathcal{L}^k.$$

In the approximation $J^k(y)$ of this second derivative, we replace Q_i, M_i, and R_i by approximations $Q_i^H(s_i, q_i, \lambda_{k+1}), M_i^H(s_i, q_i, \lambda_{k+1})$, and $R_i^H(s_i, q_i, \lambda_{k+1})$.

Remark 1. Note that the KKT matrix of each problem $P_k(x_k)$ is completely independent of the value of x_k.

Remark 2. If we split the variables $y = (\lambda_k, s_k, q_k, \dots) = (\lambda_k, s_k, q_k, \tilde{y})$ into a first and a second part, the second part $\tilde{y} \in \mathbb{R}^{n_{k+1}}$ corresponds directly to the variable space of the next, shrunken problem $P_{k+1}(x_{k+1})$, and we can see that the KKT matrix contains the KKT matrix of the next problem as a submatrix, as

$$\nabla_y^2 \mathcal{L}^k(y) = \left(\begin{array}{ccc|c}
 & -\mathbb{I} & & \\
-\mathbb{I} & Q_k & M_k & A_k^T \\
 & M_k^T & R_k & B_k^T \\
\hline
 & A_k & B_k & \\
 & & & \nabla_{\tilde{y}}^2 \mathcal{L}^{k+1}(\tilde{y}) \\
 & & &
\end{array}\right).$$

Remark 3. The favorable structure of the matrix $\nabla_y^2 \mathcal{L}^k(y)$, respectively, its approximation $J^k(y)$, allows an efficient solution of the linear system $J^k(y)x = b$ by a Riccati recursion proposed independently by Pantoja [34] and Dunn and Bertsekas [26]; cf. also [36, 16].

2.2. The constrained Gauss–Newton method. An important special case of the Newton-type methods considered in this paper is the constrained Gauss–Newton method, which is applicable for problems with a least squares form of the objective function

$$\sum_{i=k}^{N-1} \frac{1}{2} \|l_i(s_i, q_i)\|_2^2 + \frac{1}{2} \|e(s_N)\|_2^2. \tag{2.5}$$

For this case, the Hessian block approximations Q_i^H, M_i^H, and R_i^H are defined to be

$$\begin{pmatrix} Q_i^H & M_i^H \\ (M_i^H)^T & R_i^H \end{pmatrix} := \left(\frac{\partial l_i(s_i,q_i)}{\partial(s_i,q_i)}\right)^T \frac{\partial l_i(s_i,q_i)}{\partial(s_i,q_i)}, \quad Q_N^H := \left(\frac{\partial e(s_N)}{\partial s_N}\right)^T \frac{\partial e(s_N)}{\partial s_N}. \tag{2.6}$$

Note that these Hessian block approximations do not depend on the values of the Lagrange multipliers.

2.3. Local convergence. It is well known that the Newton-type scheme (2.4) for the solution of (2.1) 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.

3. Real-time iterations. Let us now go back to the real-time scenario described in section 1.1, 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 4.

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 (1.2), 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 $\mathbb{R}^{n_0} \supset \cdots \supset \mathbb{R}^{n_k} \supset \mathbb{R}^{n_{k+1}} \supset \cdots \supset \mathbb{R}^{n_{N-1}}$ of different sizes. 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 $\Pi^{k+1} y = \tilde{y} \in \mathbb{R}^{n_{k+1}}$.

3.1. 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 [19] or [16]).
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 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 as an *initial value embedding* of each problem into the manifold of perturbed problems. Though this formulation comes quite naturally in the framework of an *infeasible path* (also *simultaneous*) solution strategy, as presented, where optimality and constraints are treated simultaneously, it deserves strong emphasis as it is a feature that 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.

4. 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 $y^0 \in D_0 \subset \mathbb{R}^{n_0}$. Let us 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 $\|{\Pi^{k+1}}^T \tilde{y}\|_k = \|\tilde{y}\|_{k+1}$.

THEOREM 4.1 (local contractivity of the real-time iterations). *Let us assume that the Lagrangian functions $\mathcal{L}^k : D_k \to \mathbb{R}$ for all $k = 0, \dots, N$ are twice continuously differentiable and that their second derivative approximations $J^k : D_k \to \mathbb{R}^{n_k \times n_k}$ are continuous and have a bounded inverse $(J^k)^{-1} : D_k \to \mathbb{R}^{n_k \times n_k}$.*

Furthermore, let us assume that there exist a $\kappa < 1$ and an $\omega < \infty$ such that for each $k = 0, \dots, N$ and all $y', y \in D_k$, $\Delta y = y' - y$, and all $t \in [0,1]$ it holds that

$$\left\| \left(J^k(y')\right)^{-1} \left(J^k(y + t\Delta y) - \nabla_y^2 \mathcal{L}^k(y + t\Delta y)\right) \Delta y \right\|_k \le \kappa \|\Delta y\|_k \tag{4.1a}$$

and that

$$\left\| \left(J^k(y')\right)^{-1} \left(J^k(y + t\Delta y) - J^k(y)\right) \Delta y \right\|_k \le \omega t \|\Delta y\|_k^2, \tag{4.1b}$$

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

$$\left\| \left(J^{k+1}(\Pi^{k+1} y')\right)^{-1} \Pi^{k+1} \left(J^k(y + t\Delta y) - J^k(y)\right) \Delta y \right\|_{k+1} \le \omega t \|\Delta y\|_k^2. \tag{4.1c}$$

We suppose that the first step $\Delta y^0 := J^0(y^0)^{-1} \nabla_y \mathcal{L}^0(y^0)$ starting at the initial guess y^0 is sufficiently small so that

$$\delta_0 := \kappa + \frac{\omega}{2} \|\Delta y^0\|_0 < 1 \tag{4.1d}$$

and that the ball

$$B_0 := \left\{ y \in \mathbb{R}^{n_0} \,|\, \|y - y^0\|_0 \le \frac{\|\Delta y^0\|_0}{1-\delta_0} \right\} \tag{4.1e}$$

is completely contained in D_0. Under these conditions the real-time iterates $y^0, \dots, y^N$ defined by

$$\Delta y^k := -J^k(y^k)^{-1} \nabla_y \mathcal{L}^k(y^k), \quad y^{k+1} := \Pi^{k+1}(y^k + \Delta y^k)$$

(where $\mathcal{L}^k$ is the Lagrangian function corresponding to problem $P_k(x_k)$ with the system state obtained according to the closed-loop dynamics $x_{k+1} = f_k(x_k, u_k)$, $u_k := q_k^k + \Delta q_k^k$) are well defined and stay in the projections of the ball B_0, i.e.,

$$y^k \in \Pi^k \cdots \Pi^1 B_0 \subset D_k, \tag{4.2}$$

and satisfy the contraction condition

$$\|\Delta y^{k+1}\|_{k+1} \le \left(\kappa + \frac{\omega}{2} \|\Delta y^k\|_k \right) \|\Delta y^k\|_k =: \delta_k \|\Delta y^k\|_k \le \delta_0 \|\Delta y^k\|_k. \tag{4.3}$$

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

$$\|y^k - y_*^k\|_k \le \frac{\|\Delta y^k\|_k}{1-\delta_k} \le \frac{(\delta_0)^k \|\Delta y^0\|_0}{1-\delta_0}. \tag{4.4}$$

Proof. We divide the proof into three parts, corresponding to the properties (4.3), (4.2), and (4.4).

Contraction property. We will first show that the contraction property (4.3) holds. By adding zero to the defining equation of Δy^{k+1} we get

$$\begin{aligned} -\Delta y^{k+1} &= J^{k+1}(y^{k+1})^{-1} \nabla_y \mathcal{L}^{k+1}(y^{k+1}) \\ &= J^{k+1}(y^{k+1})^{-1} \left(\nabla_y \mathcal{L}^{k+1}(y^{k+1}) - \Pi^{k+1} \left(\nabla_y \mathcal{L}^k(y^k) + J^k(y^k) \Delta y^k \right) \right). \end{aligned} \tag{4.5}$$

Using the notation $y^k = (\lambda_k^k, s_k^k, q_k^k, \lambda_{k+1}^k, s_{k+1}^k, q_{k+1}^k, \dots)$ we observe that

$$\begin{aligned} \nabla_y \mathcal{L}^{k+1}(y^{k+1}) &= \nabla_y \mathcal{L}^{k+1}(\Pi^{k+1}(y^k + \Delta y^k)) = \begin{pmatrix} x_{k+1} - (s_{k+1}^k + \Delta s_{k+1}^k) \\ \vdots \end{pmatrix} \\ &= \begin{pmatrix} f_k(s_k^k + \Delta s_k^k, q_k^k + \Delta q_k^k) - (s_{k+1}^k + \Delta s_{k+1}^k) \\ \vdots \end{pmatrix} \\ &= \Pi^{k+1} \nabla_y \mathcal{L}^k(y^k + \Delta y^k), \end{aligned}$$

because $x_{k+1} = f_k(x_k, u_k) = f_k(s_k^k + \Delta s_k^k, q_k^k + \Delta q_k^k)$ if the system was undisturbed.[2] Therefore, we can continue to transform Δy^{k+1} and write

$$\begin{aligned} -\Delta y^{k+1} &= J^{k+1}(y^{k+1})^{-1} \Pi^{k+1} \left(\nabla_y \mathcal{L}^k(y^k + \Delta y^k) - \nabla_y \mathcal{L}^k(y^k) - J^k(y^k) \Delta y^k \right) \\ &= J^{k+1}(y^{k+1})^{-1} \Pi^{k+1} \int_0^1 (\nabla_y^2 \mathcal{L}^k(y^k + t\Delta y^k) - J^k(y^k)) \Delta y^k \, dt \\ &= J^{k+1}(y^{k+1})^{-1} \Pi^{k+1} \int_0^1 (\nabla_y^2 \mathcal{L}^k(y^k + t\Delta y^k) - J^k(y^k + t\Delta y^k)) \Delta y^k \, dt \\ &\quad + J^{k+1}(y^{k+1})^{-1} \Pi^{k+1} \int_0^1 (J^k(y^k + t\Delta y^k) - J^k(y^k)) \Delta y^k \, dt. \end{aligned} \tag{4.6}$$

[2] Note that $s_k^k + \Delta s_k^k = x_k$ due to the linearity of the constraint $x_k - s_k = 0$ and that $u_k = q_k^k + \Delta q_k^k$ by definition.

Noting that, with $\tilde{y} := \Pi^{k+1} y$,

$$\begin{aligned}\Pi^{k+1}\left(\nabla_y^2 \mathcal{L}^k(y) - J^k(y)\right) &= \Pi^{k+1}\left(\begin{array}{ccc|c} & 0 & & \\ 0 & \Delta Q_k & \Delta M_k & 0 \\ & \Delta M_k^T & \Delta R_k & 0 \\ \hline & 0 & 0 & \nabla_{\tilde{y}}^2 \mathcal{L}^{k+1}(\tilde{y}) - J^{k+1}(\tilde{y}) \end{array}\right) \\ &= \left(\begin{array}{ccc|c} 0 & 0 & 0 & \\ \vdots & \vdots & \vdots & \nabla_{\tilde{y}}^2 \mathcal{L}^{k+1}(\tilde{y}) - J^{k+1}(\tilde{y}) \\ 0 & 0 & 0 & \end{array}\right)\end{aligned}$$

and abbreviating $\tilde{y}^k := \Pi^{k+1} y^k$, $\Delta\tilde{y}^k := \Pi^{k+1}\Delta y^k$, we can exploit assumption (4.1a) to obtain

$$\begin{aligned}&\|J^{k+1}(y^{k+1})^{-1}\,\Pi^{k+1}(\nabla_y^2 \mathcal{L}^k(y^k + t\Delta y^k) - J^k(y^k + t\Delta y^k))\Delta y^k\|_{k+1} \\ &\quad = \|J^{k+1}(y^{k+1})^{-1}(\nabla_y^2 \mathcal{L}^{k+1}(\tilde{y}^k + t\Delta\tilde{y}^k) - J^{k+1}(\tilde{y}^k + t\Delta\tilde{y}^k))\Delta\tilde{y}^k\|_{k+1} \\ &\quad \le \kappa\|\Delta\tilde{y}^k\|_{k+1} = \kappa\|\Pi^{k+1}\Delta y^k\|_{k+1} \le \kappa\|\Delta y^k\|_k.\end{aligned}$$

Making also use of assumption (4.1c), we can, building on (4.6), prove the left inequality of the contraction property (4.3):

$$\|\Delta y^{k+1}\|_{k+1} \le \kappa\|\Delta y^k\|_k + \int_0^1 \omega t\|\Delta y^k\|_k^2\, dt. = \kappa\|\Delta y^k\|_k + \frac{1}{2}\omega\|\Delta y^k\|_k^2 =: \delta_k\|\Delta y^k\|_k.$$

With the help of condition (4.1d) ($\delta_0 < 1$) it is straightforward to deduce inductively that

$$\delta_{k+1} = \kappa + \frac{\omega}{2}\|\Delta y^{k+1}\|_{k+1} \le \kappa + \frac{\omega}{2}\delta_k\|\Delta y^k\|_k \le \delta_k \le \delta_0,$$

which proves the remaining part of (4.3).

Well definedness. To show that the iterates remain inside the domains of definition as stated in (4.2) we first observe that

$$\|\Delta y^k\|_k \le \delta_{k-1}\delta_{k-2}\dots\delta_0\|\Delta y^0\|_0 \le (\delta_0)^k\|\Delta y^0\|_0.$$

Using the representation

$$\begin{aligned} y^k &= \Pi^k(y^{k-1} + \Delta y^{k-1}) = \Pi^k(\Pi^{k-1}(y^{k-2} + \Delta y^{k-2}) + \Delta y^{k-1}) \\ &= \Pi^k(\Pi^{k-1}(\dots\Pi^1(y^0 + \Delta y^0)\dots) + \Delta y^{k-1}) \\ &= \Pi^k\cdots\Pi^1 y^0 + \Pi^k\cdots\Pi^1\Delta y^0 + \cdots + \Pi^k\Delta y^{k-1},\end{aligned}$$

we can find $y' := y^0 + (\Pi^k\cdots\Pi^1)^T(y^k - \Pi^k\cdots\Pi^1 y^0)$ such that

$$\begin{aligned}\|y' - y^0\|_0 &= \|(\Pi^k\cdots\Pi^1)^T(y^k - \Pi^k\cdots\Pi^1 y^0)\|_0 = \|y^k - \Pi^k\cdots\Pi^1 y^0\|_k \\ &\le \textstyle\sum_{i=0}^{k-1}\|\Delta y^i\|_i \le \|\Delta y^0\|_0\sum_{i=0}^{k-1}(\delta_0)^i \le \frac{\|\Delta y^0\|_0}{1-\delta_0},\end{aligned}$$

i.e., $y' \in B_0$ and $y^k = \Pi^k\dots\Pi^1 y'$, i.e., $y^k \in \Pi^k\dots\Pi^1 B_0$, as desired.

Distance to optimal solutions. It remains to be shown that the iterates y^k approach the exact solutions of the corresponding problems $P_k(x_k)$ as stated in (4.4). For this aim we devise a hypothetical standard Newton-type algorithm as introduced in (2.4) that allows us to compute the exact solution y_*^k of $P_k(x_k)$. As a by-product, we obtain a bound on the distance of y_*^k from y^k.

The hypothetical algorithm would proceed by starting at $y_0^k := y^k$ and iterating with iterates $y_1^k, y_2^k, \dots$ according to

$$y_{i+1}^k := y_i^k + \Delta y_i^k, \quad \Delta y_i^k := -J^k(y_i^k)^{-1} \nabla_y \mathcal{L}^k(y_i^k).$$

Note that the first step Δy_0^k is identical to Δy^k. It is for this hypothetical algorithm only that we need assumption (4.1b). Due to this condition and (4.1a), we have the contraction property

$$\|\Delta y_{i+1}^k\|_k \le \left(\kappa + \frac{\omega}{2}\|\Delta y_i^k\|_k\right) \|\Delta y_i^k\|_k$$

as can be shown by a well-known technique for Newton-type methods (see, e.g., [9]):

(4.7)

$$\begin{aligned}
\|\Delta y_{i+1}^k\|_k &= \|J^k(y_{i+1}^k)^{-1} \cdot \nabla_y \mathcal{L}^k(y_{i+1}^k)\|_k \\
&= \|J^k(y_{i+1}^k)^{-1} \cdot (\nabla_y \mathcal{L}^k(y_{i+1}^k) - \nabla_y \mathcal{L}^k(y_i^k) - J^k(y_i^k) \cdot \Delta y_i^k)\|_k \\
&= \|J^k(y_{i+1}^k)^{-1} \cdot \textstyle\int_0^1 (\nabla_y^2 \mathcal{L}^k(y_i^k + t\Delta y_i^k) - J^k(y_i^k)) \cdot \Delta y_i^k \, dt\|_k \\
&= \|J^k(y_{i+1}^k)^{-1} \cdot \textstyle\int_0^1 (\nabla_y^2 \mathcal{L}^k(y_i^k + t\Delta y_i^k) - J^k(y_i^k + t\Delta y_i^k))\Delta y_i^k \, dt \\
&\quad + J^k(y_{i+1}^k)^{-1} \cdot \textstyle\int_0^1 (J^k(y_i^k + t\Delta y_i^k) - J^k(y_i^k))\Delta y_i^k \, dt\|_k \\
&\le \textstyle\int_0^1 \|J^k(y_{i+1}^k)^{-1} (\nabla_y^2 \mathcal{L}^k(y_i^k + t\Delta y_i^k) - J^k(y_i^k + t\Delta y_i^k))\Delta y_i^k\|_k \, dt \\
&\quad + \textstyle\int_0^1 \|J^k(y_{i+1}^k)^{-1} (J^k(y_i^k + t\Delta y_i^k) - J^k(y_i^k))\Delta y_i^k\|_k \, dt \\
&\le \kappa\|\Delta y_i^k\|_k + \textstyle\int_0^1 \omega t \|\Delta y_i^k\|_k^2 \, dt \\
&= \left(\kappa + \tfrac{\omega}{2}\|\Delta y_i^k\|_k\right) \|\Delta y_i^k\|_k.
\end{aligned}$$

Together with the property

$$\kappa + \frac{\omega}{2}\|\Delta y_0^k\|_k = \kappa + \frac{\omega}{2}\|\Delta y^k\|_k = \delta_k < 1,$$

this leads again to the conclusion that $\|\Delta y_i^k\|_k \le (\delta_k)^i \|\Delta y^k\|_k$, so that $y_0^k, y_1^k, y_2^k, \dots$ is a Cauchy sequence and remains in the ball

$$B_k := \left\{ y \in \mathbb{R}^{n_k} \,|\, \|y - y^k\|_k \le \frac{\|\Delta y^k\|_k}{1 - \delta_k} \right\}$$

and thus converges toward a point $y_*^k \in B_k$, which satisfies $\nabla_y \mathcal{L}^k(y_*^k) = 0$ due to the boundedness of J^k on the (compact) ball B_k, as $\nabla_y \mathcal{L}^k(y_i^k) = -J^k(y_i^k)\Delta y_i^k \to 0$ for $i \to \infty$. ∎

5. Comparison with optimal feedback control. To assess the performance of the proposed real-time iteration scheme, we will compare the resulting system trajectory with the one which would have been obtained by exact optimal feedback control. For this aim, we denote by $u_0, \dots, u_{N-1}$ and $x_1, \dots, x_N$ the control and system state trajectories obtained by an application of the real-time iteration scheme to the system starting at the state x_0, when the iteration scheme was initialized with an initial guess $y^0 = (\lambda_0^0, s_0^0, q_0^0, \dots, \lambda_N^0, s_N^0)$, as in Theorem 4.1. On the other hand,

we denote by $q_0^*, \dots, q_{N-1}^*$ and $s_1^*, \dots, s_N^*$ the corresponding trajectories which would have been obtained by an application of exact optimal feedback control, starting at the same initial state x_0. Note that this trajectory is contained in the exact primal-dual solution vector $y_*^0 = (\lambda_0^*, s_0^*, q_0^*, \lambda_1^*, s_1^*, q_1^*, \dots, \lambda_N^*, s_N^*)$ of problem $P_0(x_0)$, as already pointed out in section 1.1.

THEOREM 5.1 (loss of optimality). *Let us in addition to the assumptions of Theorem* 4.1 *suppose that the Hessian of the Lagrangian $\mathcal{L}^0(\cdot)$ of problem $P_0(\cdot)$ is bounded on B_0, i.e.,*

$$\|\nabla_y^2 \mathcal{L}^0(y)\|_0 \le C \quad \forall y \in B_0. \tag{5.1}$$

Then the objective values, on the one hand evaluated at the closed-loop trajectory resulting from the real-time iteration scheme and on the other hand at the trajectory resulting from optimal feedback control

$$F_{\text{real}} := \sum_{i=k}^{N-1} L_i(x_i, u_i) + E(x_N) \quad \textit{and} \quad F_{\text{opt}} := \sum_{i=k}^{N-1} L_i(s_i^*, q_i^*) + E(s_N^*),$$

can be compared by

$$F_{\text{real}} \le F_{\text{opt}} + 2C \left(\frac{\delta_0}{1-\delta_0} \right)^2 \|\Delta y^0\|_0^2. \tag{5.2}$$

In particular, if $\kappa = 0$ (as for the exact Newton method), the loss of optimality is of fourth order in the size of the first step Δy^0:

$$F_{\text{real}} \le F_{\text{opt}} + \frac{C}{2} \left(\frac{\omega}{1 - \frac{\omega}{2}\|\Delta y^0\|_0} \right)^2 \|\Delta y^0\|_0^4. \tag{5.3}$$

Proof. First note that both the real-time iteration trajectory $(x_0, u_0, x_1, \dots, x_N)$ and the optimal feedback control trajectory $(s_0^*, q_0^*, s_1^*, \dots) = (x_0, q_0^*, s_1^*, \dots)$ are feasible "points" for the optimization problem $P_0(x_0)$. Let us augment the real-time iteration trajectory to a primal-dual point $y_{\text{real}} := (\lambda_0, x_0, u_0, \dots, \lambda_N, x_N)$, which is obtained by

$$y_{\text{real}} := y^0 + \Delta y^0 + {\Pi^1}^T \Delta y^1 + (\Pi^2 \Pi^1)^T \Delta y^2 + \cdots + (\Pi^N \cdots \Pi^1)^T \Delta y^N.$$

From the contractivity condition (4.3), it can easily be verified that $\|y_{\text{real}} - y^0\|_0 \le \frac{\|\Delta y^0\|_0}{1-\delta_0}$, i.e., that $y_{\text{real}} \in B_0$. We similarly see that

$$\|y_{\text{real}} - (y^0 + \Delta y^0)\|_0 \le \frac{\delta_0 \|\Delta y^0\|_0}{1-\delta_0} \quad \text{and that} \quad \|y_*^0 - (y^0 + \Delta y^0)\|_0 \le \frac{\delta_0 \|\Delta y^0\|_0}{1-\delta_0},$$

where the latter bound is due to contraction property (4.7) for the hypothetical Newton-type iterations toward the solution of $P_0(x_0)$, and the fact that the first step Δy_0^0 of these iterations coincides with the step vector Δy^0 of the real-time iterations. We can conclude that

$$\|y_{\text{real}} - y_*^0\|_0 \le \|y_{\text{real}} - (y^0 + \Delta y^0)\|_0 + \|y_*^0 - (y^0 + \Delta y^0)\|_0 \le 2 \frac{\delta_0 \|\Delta y^0\|_0}{1-\delta_0}. \tag{5.4}$$

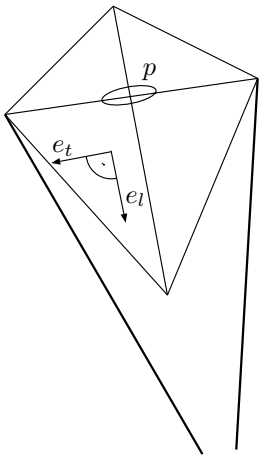

FIG. 6.1. *A picture of the kite from the pilot's point of view.*

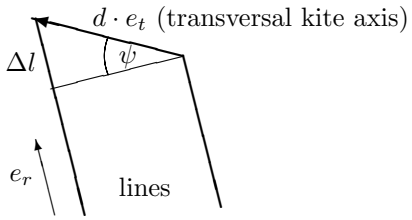

FIG. 6.2. *The kite seen from the top and visualization of the roll angle ψ.*

Because of feasibility of the primal-dual points y_{real} and y_*^0 the values of the Lagrangian function coincide with those of the objective, so that we can deduce

$$\begin{aligned} F_{\text{real}} - F_{\text{opt}} &= \mathcal{L}^0(y_{\text{real}}) - \mathcal{L}^0(y_*^0) = \int_0^1 \nabla_y \mathcal{L}^0(y_*^0 + t_1(y_{\text{real}} - y_*^0))^T \, (y_{\text{real}} - y_*^0) \, dt_1 \\ &= \int_0^1 \Big(\int_0^{t_1} \nabla_y^2 \mathcal{L}^0(y_*^0 + t_2(y_{\text{real}} - y_*^0))(y_{\text{real}} - y_*^0) \, dt_2 \Big)^T (y_{\text{real}} - y_*^0) \, dt_1 \\ &= (y_{\text{real}} - y_*^0)^T \Big(\int_0^1 \int_0^{t_1} \nabla_y^2 \mathcal{L}^0(y_*^0 + t_2(y_{\text{real}} - y_*^0)) \, dt_2 \, dt_1 \Big)^T (y_{\text{real}} - y_*^0), \end{aligned}$$

where we have used the fact that $\nabla_y \mathcal{L}^0(y_*^0) = 0$. We conclude with (5.1) and (5.4) that

$$F_{\text{real}} - F_{\text{opt}} \le \frac{1}{2} C \| y_{\text{real}} - y_*^0 \|_0^2 \le \frac{1}{2} C \left(2 \frac{\delta_0 \| \Delta y^0 \|_0}{1 - \delta_0} \right)^2 . \qquad \square$$

6. Numerical example: 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 control of the roll angle of the kite; see Figures 6.1 and 6.2. 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 Figure 6.3). 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.

6.1. The dual line kite model. The movement of the kite in the sky can be modeled by Newton's laws of motion and a suitable model for the aerodynamic force. The most difficulty lies in the determination of suitable coordinate systems; we will first describe the kite's motion in polar coordinates, and second we will determine the direction of the aerodynamic forces.

6.1.1. Newton's laws of motion in polar coordinates. The position $p \in \mathbb{R}^3$ of the kite can be modeled in three-dimensional Euclidean space, choosing the position of the kite pilot as the origin, and the third component p_3 to be the height of the kite above the ground. With m denoting the mass of the kite and $F \in \mathbb{R}^3$ the total force acting on the kite, Newton's law of motion reads

$$\ddot{p} = \frac{d^2 p}{dt^2} = \frac{F}{m}.$$

Let us introduce polar coordinates θ, ϕ, r:

$$p = \begin{pmatrix} p_1 \\ p_2 \\ p_3 \end{pmatrix} = \begin{pmatrix} r\sin(\theta)\cos(\phi) \\ r\sin(\theta)\sin(\phi) \\ r\cos(\theta) \end{pmatrix}.$$

Note that the distance r between pilot and kite is usually constant during flight, and θ is the angle that the lines form with the vertical. Let us introduce a local right-handed coordinate system with the three basis vectors

$$e_\theta = \begin{pmatrix} \cos(\theta)\cos(\phi) \\ \cos(\theta)\sin(\phi) \\ -\sin(\theta) \end{pmatrix}, \quad e_\phi = \begin{pmatrix} -\sin(\phi) \\ \cos(\phi) \\ 0 \end{pmatrix}, \quad \text{and} \quad e_r = \begin{pmatrix} \sin(\theta)\cos(\phi) \\ \sin(\theta)\sin(\phi) \\ \cos(\theta) \end{pmatrix}.$$

Defining $F_\theta := F \cdot e_\theta, F_\phi := F \cdot e_\phi$, and $F_r := F \cdot e_r$, we can write Newton's laws of motion in the form

$$\begin{aligned} r\ddot{\theta} - r\sin(\theta)\cos(\theta)\dot{\phi}^2 + 2\dot{r}\dot{\theta} &= \frac{F_\theta}{m}, \\ r\sin(\theta)\ddot{\phi} + 2r\cos(\theta)\dot{\phi}\dot{\theta} + 2\sin(\theta)\dot{r}\dot{\phi} &= \frac{F_\phi}{m}, \\ \ddot{r} - r\dot{\theta}^2 - r\sin^2(\theta)\dot{\phi}^2 &= \frac{F_r}{m}. \end{aligned} \tag{6.1}$$

If the length of the lines, denoted by r, is kept constant, all terms involving time derivatives of r will drop out. Furthermore, the last equation (6.1) will become redundant, as any acting force F_r' in the radial direction will automatically be augmented by a constraint force contribution $F_c := F_r + mr\dot{\theta}^2 + mr\sin^2(\theta)\dot{\phi}^2$ so that (6.1) is satisfied with $F_r := F_r' - F_c$. In this case we can regard only the components F_θ and F_ϕ which are not changed by the constraint force. The equations of motion[3] simplify to

$$\ddot{\theta} = \frac{F_\theta}{rm} + \sin(\theta)\cos(\theta)\dot{\phi}^2, \tag{6.2}$$

$$\ddot{\phi} = \frac{F_\phi}{rm\sin(\theta)} - 2\cot(\theta)\dot{\phi}\dot{\theta}. \tag{6.3}$$

In our model, the force F acting on the kite consists of three contributions, constraint force $-F_c e_r$, gravitational force F^{gra}, and aerodynamic force F^{aer}. In Cartesian coordinates, $F^{\text{gra}} = (0, 0, -mg)^T$ with $g = 9.81$ m s^{-2} being the earth's gravitational acceleration. In local coordinates we therefore have

$$F_\theta = F_\theta^{\text{gra}} + F_\theta^{\text{aer}} = \sin(\theta)mg + F_\theta^{\text{aer}} \quad \text{and} \quad F_\phi = F_\phi^{\text{aer}}.$$

It remains to derive an expression for the aerodynamic force F^{aer}.

[3]Note that the validity of these equations requires that $F_c \geq 0$, as a line can only pull and not push.

6.1.2. Kite orientation and the aerodynamic force. To model the aerodynamic force that is acting on the kite, we first assume that the kite's trailing edge is always pulled by the tail into the direction of the effective wind, as seen from the kite's body-fixed frame. This assumption can be regarded as the limiting case of very large tail force. It crucially simplifies the model by allowing us to disregard angular momentum and the moments acting on the kite. Under this assumption we are able to determine the kite orientation as an explicit function of position and velocity only, as shown in the following.

By the large tail force assumption, the kite's longitudinal axis is always in line with the effective wind vector $w_e := w - \dot{p}$, where $w = (v_w, 0, 0)^T$ is the wind as seen from the earth system, and $\dot{p}$ is the kite velocity. If we introduce a unit vector e_l pointing from the front toward the trailing edge of the kite (cf. Figure 6.1), we therefore assume that

$$e_l = \frac{w_e}{\|w_e\|}.$$

The transversal axis of the kite can be described by a perpendicular unit vector e_t that is pointing from the left to the right wing tip. Clearly, it is orthogonal to the longitudinal axis, i.e.,

$$e_t \cdot e_l = \frac{e_t \cdot w_e}{\|w_e\|} = 0. \tag{6.4}$$

The orientation of the transversal axis e_t against the lines' axis (which is given by the vector e_r) can be influenced by the length difference Δl of the two lines. If the distance between the two lines' fixing points on the kite is d, then the vector from the left to the right fixing point is de_t, and the projection of this vector onto the lines' axis should equal $\Delta l = d\, e_t \cdot e_r$, being positive if the left hand's lines wingtip is farther away from the pilot; cf. Figure 6.2. Let us define the *roll angle* ψ to be

$$\psi = \arcsin\left(\frac{\Delta l}{d}\right).$$

We will assume that we control this angle ψ directly. It determines the orientation of e_t which has to satisfy

$$e_t \cdot e_r = \frac{\Delta l}{d} = \sin(\psi). \tag{6.5}$$

A third requirement that e_t should satisfy is that

$$(e_l \times e_t) \cdot e_r = \frac{w_e \times e_t}{\|w_e\|} \cdot e_r > 0, \tag{6.6}$$

which takes account of the fact that the kite is always in the same orientation with respect to the lines.

How does one find a unit vector e_t that satisfies these requirements (6.4)–(6.6)? Using the projection w_e^p of the effective wind vector w_e onto the tangent plane spanned by e_θ and e_ϕ,

$$w_e^p := e_\theta(e_\theta \cdot w_e) + e_\phi(e_\phi \cdot w_e) = w_e - e_r(e_r \cdot w_e),$$

we can define the orthogonal unit vectors

$$e_w := \frac{w_e^p}{\|w_e^p\|} \quad \text{and} \quad e_o := e_r \times e_w,$$

so that (e_w, e_o, e_r) forms an orthogonal right-handed coordinate basis. Note that in this basis the effective wind w_e has no component in the e_o direction, as $w_e = \|w_e^p\| e_w + (w_e \cdot e_r) e_r$. We will show that the definition

$$e_t := e_w(-\cos(\psi)\sin(\eta)) + e_o(\cos(\psi)\cos(\eta)) + e_r \sin(\psi) \tag{6.7}$$

with

$$\eta := \arcsin\left(\frac{w_e \cdot e_r}{\|w_e^p\|} \tan(\psi)\right)$$

satisfies the requirements (6.4)–(6.6).[4] Equation (6.4) can be verified by substitution of the definition of η into

$$e_t \cdot w_e = -\cos(\psi)\sin(\eta)\|w_e^p\| + \sin(\psi)(w_e \cdot e_r) = 0.$$

Equation (6.5) is trivially satisfied, and (6.6) can be verified by calculation of

$$\begin{aligned}(w_e \times e_t) \cdot e_r &= (w_e \cdot e_w)\cos(\psi)\cos(\eta) - (w_e \cdot e_o)(-\cos(\psi)\sin(\eta)) \\ &= \|w_e^p\| \cos(\psi)\cos(\eta)\end{aligned}$$

(where we used the fact that $w_e \cdot e_o = 0$). For angles ψ and η in the range from $-\pi/2$ to $\pi/2$ this expression is always positive. The above considerations allow us to determine the orientation of the kite depending on the control ψ and the effective wind w_e only. Note that the considerations would break down if the projection of the effective wind w_e^p would be equal to zero, if $|\psi| \geq \frac{\pi}{2}$, or if

$$\left|\frac{w_e \cdot e_r}{\|w_e^p\|} \tan(\psi)\right| > 1.$$

The two vectors $e_n := e_l \times e_t$ and e_l are the directions of aerodynamic lift and drag, respectively. To compute the magnitudes F_{L} and F_{D} of lift and drag we assume that the lift and drag coefficients C_{L} and C_{D} are constant, so that we have

$$F_{\mathrm{L}} = \frac{1}{2}\rho\|w_e\|^2 A C_{\mathrm{L}} \quad \text{and} \quad F_{\mathrm{D}} = \frac{1}{2}\rho\|w_e\|^2 A C_{\mathrm{D}},$$

with ρ being the density of air and A being the characteristic area of the kite. Given the directions and magnitudes of lift and drag, we can compute F^{aer} as their sum, yielding $F^{\mathrm{aer}} = F_{\mathrm{L}} e_n + F_{\mathrm{D}} e_l$, or, in the local coordinate system,

$$F_\theta^{\mathrm{aer}} = F_{\mathrm{L}}(e_n \cdot e_\theta) + F_{\mathrm{D}}(e_l \cdot e_\theta) \text{ and } F_\phi^{\mathrm{aer}} = F_{\mathrm{L}}(e_n \cdot e_\phi) + F_{\mathrm{D}}(e_l \cdot e_\phi).$$

The system parameters that have been chosen for the simulation model are listed in Table 6.1. Defining the system state $\xi := (\theta, \phi, \dot\theta, \dot\phi)^T$ and the control $u := \psi$, we can summarize the four system equations, i.e., (6.2)–(6.3) and the trivial equations $\frac{\partial\theta}{\partial t} = \dot\theta$, $\frac{\partial\phi}{\partial t} = \dot\phi$, in the short form

$$\dot\xi = \hat f(\xi, u).$$

[4] It is interesting to note that the assignment of e_t can be made more transparent by considering a rotation from the (e_w, e_o, e_r) tangential frame to the body frame (e_l, e_t, e_n), with $e_n := e_l \times e_t$. Specifically, starting from (e_w, e_o, e_r), we rotate about the e_r-axis through the *yaw* angle $-\eta$ and then *roll* through the angle ψ about the e_l-axis. From this we find that e_t as the second basis vector of the body frame is represented by (6.7) in the tangential frame (e_w, e_o, e_r).

TABLE 6.1
The kite parameters.

Name	Symbol	Value
Line length	r	50 m
Kite mass	m	1 kg
Wind velocity	v_w	6 m/s
Density of air	ρ	1.2 kg/m^3
Characteristic area	A	0.5 m^2
Lift coefficient	C_{L}	1.5
Drag coefficient	C_{D}	0.29

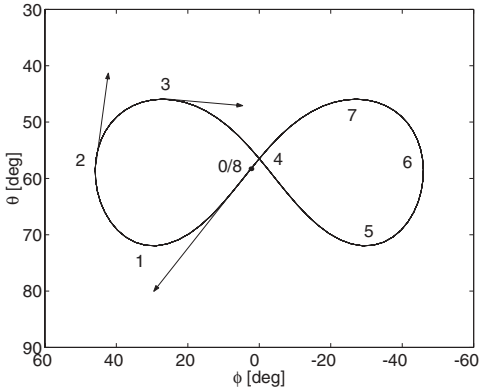

FIG. 6.3. *Periodic orbit plotted in the (ϕ, θ)-plane, as seen by the kite pilot. The dots separate intervals of one second.*

6.2. A periodic orbit. Using the above system model, a periodic orbit was determined that can be characterized as a "lying eight" and which is depicted as a (ϕ, θ)-plot in Figure 6.3. The wind is assumed to blow in the direction of the p_1-axis ($\theta = 90^o$ and $\phi = 0^o$). The periodic solution was computed using an off-line variant of the direct multiple shooting method (MUSCOD-II, due to Leineweber [31, 22, 23]), imposing periodicity conditions with period $T = 8$ seconds and suitable state bounds and a suitable objective function in order to yield a solution that was considered to be a meaningful reference orbit. We will denote the periodic reference solution by $\xi_r(t)$ and $u_r(t)$. This solution is defined for all $t \in (-\infty, \infty)$ and satisfies the periodicity condition $\xi_r(t+T) = \xi_r(t)$ and $u_r(t+T) = u_r(t)$. It is interesting to note that small errors accumulate very quickly so that the uncontrolled system will not stay in the periodic orbit very long during a numerical simulation; this observation can be confirmed by investigating the asymptotic stability properties of the periodic orbit [16], which shows that local errors are amplified by a factor of more than 5 during each period. Thus, the open-loop system is highly unstable in the periodic orbit.

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 [17].

6.3. The optimal control problem. Given an arbitrary initial state x_0 (that we do not know in advance) we want the kite to fly three times the figure of Figure 6.3, 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 of bringing the system close to the reference orbit. For this aim we define a Lagrange term of least squares type

$$L(\xi, u, t) := \frac{1}{2}(\xi - \xi_r(t))^T Q(\xi - \xi_r(t)) + \frac{1}{2}(u - u_r(t))^T R(u - u_r(t))$$

with diagonal weighting matrices

$$Q := \mathrm{diag}(0.4,\ 1,\ \mathrm{s}^2,\ \mathrm{s}^2)\frac{1}{\mathrm{s}} \quad \text{and} \quad R := 1.0 \cdot 10^{-2}\mathrm{deg}^{-2}\mathrm{s}^{-1}.$$

Using these definitions, we formulate the following optimal control problem on the time horizon of interest $[0, 3T]$:

$$\begin{aligned} &\min_{u(\cdot),\xi(\cdot)} \int_0^{3T} L(\xi(t), u(t), t)\, dt \\ &\text{subject to} \\ &\qquad \dot{\xi}(t) = \hat{f}(\xi(t), u(t)) \qquad \forall t \in [0, 3T], \\ &\qquad \xi(0) = x_0. \end{aligned} \tag{6.8}$$

6.4. Direct multiple shooting formulation. 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 [35, 12]: 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 $q_0, q_1, \ldots, q_{N-1}$, as well as artificial initial values $s_0, \ldots, s_N$, as depicted in Figure 1.1. On each of these intervals we solve the following initial value problem:

$$\begin{aligned} \dot{\xi}_i(t; s_i, q_i) &= \hat{f}\left(\xi_i(t; s_i, q_i), q_i\right), \quad t \in [t_i, t_{i+1}], \\ \xi_i(t_i; s_i, q_i) &= s_i, \end{aligned} \tag{6.9}$$

yielding a trajectory piece $\xi_i(t; s_i, q_i)$ for $t \in [t_i, t_{i+1}]$. This allows us to conveniently define a discrete-time system as in (1.2c) with transition function

$$f_i(s_i, q_i) := \xi_i(t_{i+1}; s_i, q_i).$$

Analogously, we define the objective contributions in (1.2a) by

$$L_i(s_i, q_i) := \int_{t_i}^{t_{i+1}} L(\xi_i(t; s_i, q_i), q_i, t)dt.$$

The main difficulty of the direct multiple shooting method lies in the efficient solution of the initial value problems (6.9) and in the sensitivity computation. For this aim we use the advanced backward differentiation formula (BDF) code DAESOL (Bauer, Bock, and Schlöder [3], Bauer [2]), which is especially suited for stiff problems, as the above kite model. It uses the principle of internal numerical differentiation (IND) as introduced by Bock [8].

Using the multiple shooting formulation, we have transformed the continuous-time optimization problem (6.8) into a nonlinear programming problem $P_0(x_0)$ of exactly the type (1.2).

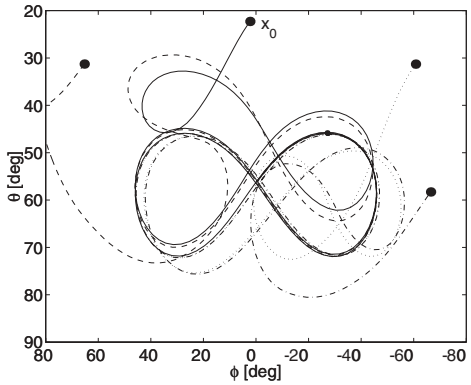

FIG. 6.4. *Closed-loop trajectories resulting from the real-time iteration scheme for different initial values x_0. The kite never crashes onto the ground ($\theta = 90$ degrees).*

6.4.1. Generation of the Gauss–Newton Hessian blocks. The efficient generation of a Gauss–Newton approximation for continuous least squares terms deserves some attention: the Hessian approximations (2.6) are determined according to

$$\begin{pmatrix} Q_i^H & M_i^H \\ (M_i^H)^T & R_i^H \end{pmatrix} := \int_{t_i}^{t_{i+1}} \left(\frac{\partial \xi_i(t; s_i, q_i)}{\partial (s_i, q_i)} \right)^T Q \, \frac{\partial \xi_i(t; s_i, q_i)}{\partial (s_i, q_i)} + \begin{pmatrix} 0 & 0 \\ 0 & R \end{pmatrix} dt.$$

These integrals are efficiently computed during the sensitivity computation using a specially adapted version of the integrator DAESOL [16, 25].

6.5. 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., $y^0 := (\lambda_0^0, s_0^0, q_0^0, \ldots, \lambda_N^0, s_N^0)$, 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 3.1 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 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 Figure 6.4 as (ϕ, θ)-plots. For all scenarios, the third loop is already close to the reference trajectory.

In Figure 6.5 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 seven seconds onto the ground.

Note that the computation of the hypothetical optimal feedback control needs about eight seconds on a Compaq Alpha XP1000 workstation. This delay means that

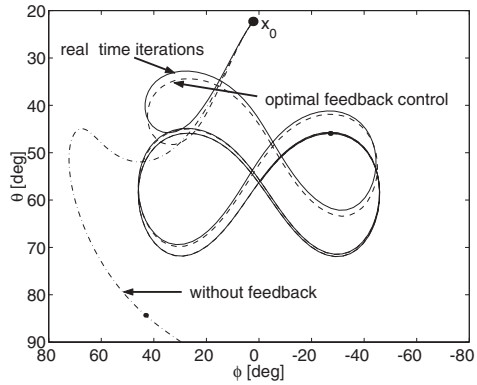

FIG. 6.5. *Comparison of trajectories resulting from the real-time iteration scheme (solid line), the open-loop controls without feedback (dash-dotted line, crashing onto the ground), and a hypothetical optimal feedback control (dashed line).*

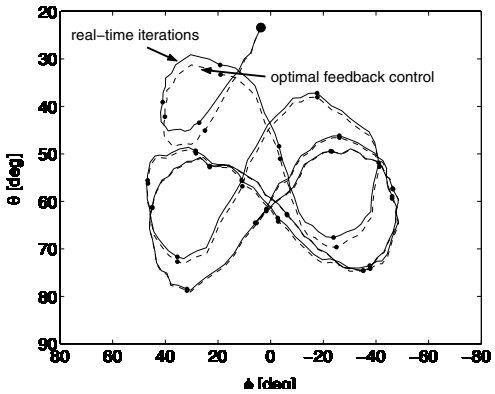

FIG. 6.6. *Disturbance scenario: Closed-loop response resulting from real-time iteration scheme (solid line) and a hypothetical optimal feedback control (dashed line). After a large initial disturbance, the state is additionally disturbed each* 0.1 *second by independent Gaussian noise. The disturbance sequence for both trajectories is identical.*

no feedback can be applied in the meantime, so the kite would have crashed onto the ground before the first response would have been 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 always needed under 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.

6.5.1. High frequency disturbances. In a third feedback simulation scenario shown in Figure 6.6 we test the performance of the real-time iteration scheme in the

presence of random disturbances with a frequency higher than the sampling time: each tenth of a second the state $(\theta, \phi, \dot{\theta}, \dot{\phi})$ is randomly disturbed by independent Gaussian noise of standard deviation $0.01 \cdot (1, 1, \mathrm{s}^{-1}, \mathrm{s}^{-1})$. Because feedback is provided only once a second, the kite flies open-loop during one second before feedback can be provided to the accumulated result of the disturbances. The initial state was much more strongly disturbed, in the same way as in the scenario of Figure 6.5. Despite these combined disturbances the scheme is able to lead the kite efficiently into the reference orbit. Again, it compares well with optimal feedback control. Note, however, that the results of Theorems 4.1 and 5.1 are not directly applicable to this third scenario as these theorems assume undisturbed system behavior after one initial disturbance.

For feedback control simulations of the kite using a moving horizon framework including also state constraints, we refer to [16, 24].

7. Conclusions. We have presented a recently developed Newton-type method for the real-time optimization of nonlinear processes and have given a new contractivity result and a bound on the loss of optimality when 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 us 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 us to obtain an immediate feedback that takes all linearized constraints into account.

The contractivity of the scheme is proven under mild conditions that are nearly identical to the sufficient conditions for convergence of off-line Newton-type methods. Iterates on different horizon lengths have to be compared. The result is that the real-time iterates, after an initial disturbance, geometrically approach the exact optimal solutions during the runtime of the process. When the resulting closed-loop trajectory is compared to optimal feedback control, a bound on the loss of optimality has been established, which is of fourth order in the initial disturbance if an exact Newton–Raphson method is used.

A newly developed kite model is presented. The control aim is, starting from an arbitrary initial state, to steer the kite into a periodic orbit, a "lying eight" with a period duration of eight seconds. We consider a time horizon of 24 seconds. 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 is always 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 NMPC 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 [21].

Acknowledgments. The first author wants to thank F. Bonnans and M. Wright for encouraging publication of this work and also thanks the anonymous referees whose valuable comments helped to improve the paper.

REFERENCES

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

[2] 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. thesis, University of Heidelberg, Heidelberg, Germany, 1999, http://www.ub.uni-heidelberg.de/archiv/1513.

[3] I. Bauer, H. G. Bock, and J. P. Schlöder, *DAESOL: A BDF-Code for the Numerical Solution of Differential Algebraic Equations*, Internal report, IWR, SFB 359, University of Heidelberg, Heidelberg, Germany, 1999.

[4] L. Biegler and J. Rawlings, *Optimization approaches to nonlinear model predictive control*, in Proceedings of the 4th International Conference on Chemical Process Control, W. Ray and Y. Arkun, eds., AIChE, CACHE, Padre Island, TX, 1991, pp. 543–571.

[5] L. T. Biegler, *Efficient solution of dynamic optimization and NMPC problems*, in Nonlinear Model Predictive Control, F. Allgöwer and A. Zheng, eds., Progr. Systems Control Theory 26, Birkhäuser, Basel, 2000, pp. 219–244.

[6] T. Binder, L. Blank, H. G. Bock, R. Bulirsch, W. Dahmen, M. Diehl, T. Kronseder, W. Marquardt, J. P. Schlöder, and O. v. 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. O. Krumke, and J. Rambau, eds., Springer-Verlag, New York, 2001, pp. 295–340. also available online from http://www.zib.de/dfg-echtzeit/Publikationen/Preprints/Preprint-01-15.html.

[7] H. Bock, M. Diehl, D. B. 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., Springer-Verlag, Berlin, 1999, pp. 218–227.

[8] H. G. Bock, *Numerical treatment of inverse problems in chemical reaction kinetics*, in Modelling of Chemical Reaction Systems, K. H. Ebert, P. Deuflhard, and W. Jäger, eds., Springer Ser. Chem. Phys. 18, Springer-Verlag, Heidelberg, 1981, pp. 102–125.

[9] H. G. Bock, *Randwertproblemmethoden zur Parameteridentifizierung in Systemen nichtlinearer Differentialgleichungen*, Bonner Math. Schriften 183, University of Bonn, Bonn, 1987.

[10] H. G. Bock, M. Diehl, D. B. Leineweber, and J. P. Schlöder, *A direct multiple shooting method for real-time optimization of nonlinear DAE processes*, in Nonlinear Model Predictive Control, F. Allgöwer and A. Zheng, eds., Progr. Systems Control Theory 26, Birkhäuser, Basel, 2000, pp. 246–267.

[11] H. G. Bock, M. Diehl, J. P. 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, Proceedings of the International Symposium on Advanced Control of Chemical Processes, Vol. 2, Pisa, Italy, 2000, pp. 695–703.

[12] H. G. Bock and K. J. Plitt, *A multiple shooting algorithm for direct solution of optimal control problems*, in Proceedings of the 9th IFAC World Congress Budapest, Pergamon Press, Oxford, UK, 1984, pp. 243–247.

[13] H. Choi, M. Hinze, and K. Kunisch, *Instantaneous control of backward-facing-step flows*, Appl. Numer. Math., 31 (1999), pp. 133–158.

[14] 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., 253 (1993), pp. 509–543.

[15] N. de Oliveira and L. Biegler, *An extension of Newton-type algorithms for nonlinear process control*, Automatica J. IFAC, 31 (1995), pp. 281–286.

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

[17] M. Diehl, *The Kite Benchmark Problem Homepage*, http://www.iwr.uni-heidelberg.de/~Moritz.Diehl/KITE/kite.html, 2003.

[18] M. Diehl, H. G. Bock, and J. P. Schlöder, *Newton-type methods for the approximate solution of nonlinear programming problems in real time*, in High Performance Algorithms and Software for Nonlinear Optimization, G. D. Pillo and A. Murli, eds., Kluwer Academic Publishers B.V., Dordrecht, The Netherlands, 2002, pp. 177–200.

[19] M. Diehl, H. G. Bock, J. P. 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., 12 (2002), pp. 577–585.

[20] M. Diehl, R. Findeisen, S. Schwarzkopf, I. Uslu, F. Allgöwer, H. G. Bock, E. D. Gilles, and J. P. Schlöder, *An efficient algorithm for nonlinear model predictive control of large-scale systems. Part* I: *Description of the method*, Automatisierungstechnik, 50 (2002), pp. 557–567.

[21] M. Diehl, R. Findeisen, S. Schwarzkopf, I. Uslu, F. Allgöwer, H. G. Bock, E. D. Gilles, and J. P. Schlöder, *An efficient algorithm for nonlinear model predictive control of large-scale systems. Part* II: *Application to a distillation column*, Automatisierungstechnik, 51 (2003), pp. 22–29.

[22] D. B. Leineweber, I. Bauer, H. G. Bock, and J. P. Schlöder, *An effficient multiple shooting based reduced SQP strategy for large-scale dyamic process optimization. Part* I: *Theoretical aspects*, Comp. Chem. Engrg., 27 (2003), pp. 157–166.

[23] D. B. Leineweber, A. Schäfer, H. G. Bock, and J. P. Schlöder, *An effficient multiple shooting based reduced SQP strategy for large-scale dyamic process optimization. Part* II: *Software aspects and applications*, Comp. Chem. Engrg., 27 (2003), pp. 167–174.

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

[25] M. Diehl, I. Uslu, R. Findeisen, S. Schwarzkopf, F. Allgöwer, H. G. Bock, T. Bürner, E. D. Gilles, A. Kienle, J. P. 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-Verlag, New York, 2001, pp. 363–384; also available online from http://www.zib.de/dfg-echtzeit/Publikationen/Preprints/Preprint-01-16.html.

[26] J. C. Dunn and D. P. Bertsekas, *Efficient dynamic programming implementations of Newton's method for unconstrained optimal control problems*, J. Optim. Theory Appl., 63 (1989), pp. 23–38.

[27] A. Helbig, O. Abel, and W. Marquardt, *Model predictive control for on-line optimization of semi-batch reactors*, in Proceedings of the American Control Conference, Philadelphia, PA, 1998, pp. 1695–1699.

[28] B. Kouvaritakis and M. Cannon, eds., *Non-Linear Predictive Control: Theory and Practice*, IEE Publishing, London,, 2001.

[29] 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, Basel, 1987, pp. 287–306.

[30] B. Kugelmann and H. Pesch, *New general guidance method in constrained optimal control, part* 1: *Numerical method*, J. Optim. Theory Appl., 67 (1990), pp. 421–435.

[31] D. B. Leineweber, *Efficient Reduced SQP Methods for the Optimization of Chemical Processes Described by Large Sparse DAE Models*, Fortschr.-Ber. VDI Reihe 3, Verfahrenstechnik 613, VDI Verlag, Düsseldorf, 1999.

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

[33] Z. Nagy, R. Findeisen, M. Diehl, F. Allgöwer, H. G. Bock, S. Agachi, J. P. Schlöder, and D. B. Leineweber, *Real-time feasibility of nonlinear predictive control for large scale processes: A case study*, in Proceedings of the American Control Conference, Chicago, IL, 2000, pp. 4249–4254.

[34] J. O. Pantoja, *Differential dynamic programming and Newton's method*, Internat. J. Control, 47 (1988), pp. 1539–1553.

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

[36] C. Rao, S. Wright, and J. Rawlings, *Application of interior-point methods to model predictive control*, J. Optim Theory Appl., 99 (1998), pp. 723–757.

[37] L. Santos, *Multivariable Predictive Control of Nonlinear Chemical Processes*, Ph.D. thesis, Universidade de Coimbra, Coimbra, Portugal, 2000.

[38] M. J. Tenny, S. J. Wright, and J. B. Rawlings, *Nonlinear model predictive control via feasibility-perturbed sequential quadratic programming*, Comput. Optim. Appl., 28 (2004), pp. 87–121.