PERGAMON

Annual Reviews in Control 28 (2004) 37–45

Annual Reviews in Control

www.elsevier.com/locate/arcontrol

Efficient NMPC of unstable periodic systems using approximate infinite horizon closed loop costing

Moritz Diehl [a,*], Lalo Magni [b], Giuseppe De Nicolao [b]

[a] *Interdisciplinary Center for Scientific Computing (IWR), University of Heidelberg, Germany*
[b] *Dipartimento di Informatica e Sistemistica, University of Pavia, Italy*

Received 20 October 2003; received in revised form 20 January 2004; accepted 24 January 2004

Abstract

We develop a state-of-the-art nonlinear model predictive controller (NMPC) for periodic unstable systems, and apply the method to a dual line kite that shall fly loops. The kite is described by a nonlinear unstable ODE system (which we freely distribute), and the aim is to let the kite fly a periodic figure. Our NMPC approach is based on the "infinite horizon closed loop costing" scheme to ensure nominal stability. To be able to apply this scheme, we first determine a periodic LQR controller to stabilize the kite locally in the periodic orbit. Then, we formulate a two-stage NMPC optimal control problem penalizing deviations of the system state from the periodic orbit, which also contains a state constraint that avoids that the kite collides with the ground. To solve the optimal control problems reliably and in real-time, we apply the newly developed "real-time iteration scheme" for fast online optimization in NMPC. The optimization based NMPC leads to significantly improved performance compared to the LQR controller, in particular as it respects state constraints. The NMPC closed loop also performs well in the presence of large random disturbances and shows considerable robustness against changes in the wind direction.

Keywords: Periodic control; LQR; Nonlinear control systems; Predictive control; Online optimization; Stability; Numerical methods; Optimal control

1. Introduction

Nonlinear model predictive control (NMPC) is a feedback control technique that is based on the real-time optimization of a nonlinear dynamic process model on a moving horizon that has attracted increasing attention over the past decade (Qin & Badgwell, 2001). Important challenges that need to be addressed for any NMPC application are stability of the closed loop system and the numerical solution of the optimal control problems in real-time. In this paper we show how state-of-the-art NMPC techniques addressing these challenges can be applied to control a strongly unstable periodic system, namely a dual line kite that shall fly loops. The aim of our automatic control is to make the kite fly a figure that may be called a "lying eight". The corresponding orbit is not open loop stable, so that feedback has to be applied. We assume the state is fully accessible for control.

Since the natural setting of the problem is in continuous time, the NMPC implementation proposed here is developed for continuous-time systems. However, it basically differs from the continuous-time NMPC algorithms for nonlinear systems previously published in the literature, see e.g., Mayne and Michalska (1990) and Chen and Allgöwer (1998). Continuous-time methods usually assume that the NMPC law is continuously computed by solving at any time instant a difficult optimization problem. This is impossible in practice, as any implementation is performed in digital form and requires a non-negligible computational time. The NMPC setup proposed here is based on the method proposed in Magni, Scattolini, and Astrom (2002), where a continuous time locally stabilizing control law is first designed. Then, a piecewise constant term computed via NMPC is added to the control signal provided by the stabilizing control law, in order to achieve some specific goals, such as the minimization of a prescribed cost or the enlargment of the output admissible set. In so doing, it is assumed that the signal computed by NMPC is piecewise constant and with a limited number of free moves in the future. Nominal stability of the overall system is preserved using the "infinite horizon closed loop costing" scheme proposed in De Nicolao, Magni, and Scattolini (1998). In the usual setting of this scheme, the optimization problems are solved up to a prespecified accuracy

* Corresponding author.
E-mail address: m.diehl@iwr.uni-heidelberg.de (M. Diehl).
URL: http://www.iwr.uni-heidelberg.de/Moritz.Diehl/.

1367-5788/$ – see front matter © 2004 Elsevier Ltd. All rights reserved.
doi:10.1016/j.arcontrol.2004.01.011

during each sampling time so that a feedback delay of one sampling time is introduced in the closed loop.

In this paper, however, we avoid this feedback delay by using the recently developed "real-time iteration" scheme (Diehl, Bock, Schlöder, Findeisen, Nagy, & Allgöwer, 2002b) for online optimization. The algorithm is based on the direct multiple shooting approach to optimal control problems (Bock & Plitt, 1984; Leineweber, 1999), but is characterized by the following features: first, the scheme efficiently initializes each new problem and performs only one optimization iteration per optimization problem. Thus, it reduces sampling times to a minimum. Second, the computations of each "real-time iteration" are divided into a very short "feedback phase", and a much longer "preparation phase", which uses the sampling time to *prepare* the next feedback. Thus, each NMPC feedback is directly applied to the system, with a negligible delay that is considerably shorter than the sampling time.

1.1. Overview

The paper is organized as follows. In Section 2 we derive the model equations for the kite model. The periodic reference orbit is analysed in Section 3 and we show how to design a stabilizing periodic linear controller based on LQR techniques. In Section 4 we finally describe the NMPC setup, and in Section 5 we briefly present the real-time optimization algorithm. Simulated closed loop experiments are presented and briefly discussed in Section 6.

2. Kite model

The kite is held by two lines which allow to control the lateral angle of the kite, see Fig. 1. By pulling one line the kite will turn in the direction of the line being pulled. In this paper we employ a kite model that was originally developed in Diehl (2002) and Diehl, Bock, and Schlöder (2002a).

2.1. Kite dynamics in polar coordinates

The movement of the kite at the sky can be modelled by Newton's law of motion and a suitable model for the aerodynamic force. Let us introduce polar coordinates θ, ϕ, r so that the position p of the kite relative to the kite pilot (in the origin) is given by: $p = (r\sin(\theta)\cos(\phi), r\sin(\theta)\sin(\phi), r\cos(\theta))^{\mathrm{T}}$ with the last component being the height of the kite over the ground, and θ being the angle that the kite lines form with the vertical. We introduce a local right handed coordinate system with the basis vectors e_θ, e_ϕ, e_r, each pointing in the direction where the corresponding polar coordinate increases:

$$e_\theta = \frac{\partial p/\partial\theta}{\|\partial p/\partial\theta\|} = [(\cos(\theta)\cos(\phi), \cos(\theta)\ \sin(\phi), -\sin(\theta))]^{\mathrm{T}}, \quad \text{etc.}$$

Defining the corresponding components of the total force F acting on the kite, we can write Newton's law of motion for constant r in the form

$$\ddot{\theta} = \frac{F_\theta}{rm} + \sin(\theta)\cos(\theta)\dot{\phi}^2, \tag{1}$$

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

where m denotes the mass of the kite. The force consists of two contributions, gravitational and aerodynamic force, so that we obtain $F_\theta = \sin(\theta)mg + F_\theta^{\mathrm{aer}}$ and $F_\phi = F_\phi^{\mathrm{aer}}$, where $g = 9.81\,\mathrm{m\,s^{-2}}$ is the earth's gravitational acceleration. It remains to determine the aerodynamic forces F_ϕ^{aer} and F_θ^{aer}.

2.2. Kite orientation

To model the aerodynamic force we first determine the kite's orientation. We assume that the kite's trailing edge is strongly pulled by the tail into the direction of the effective wind at the kite. Under this assumption the kite's

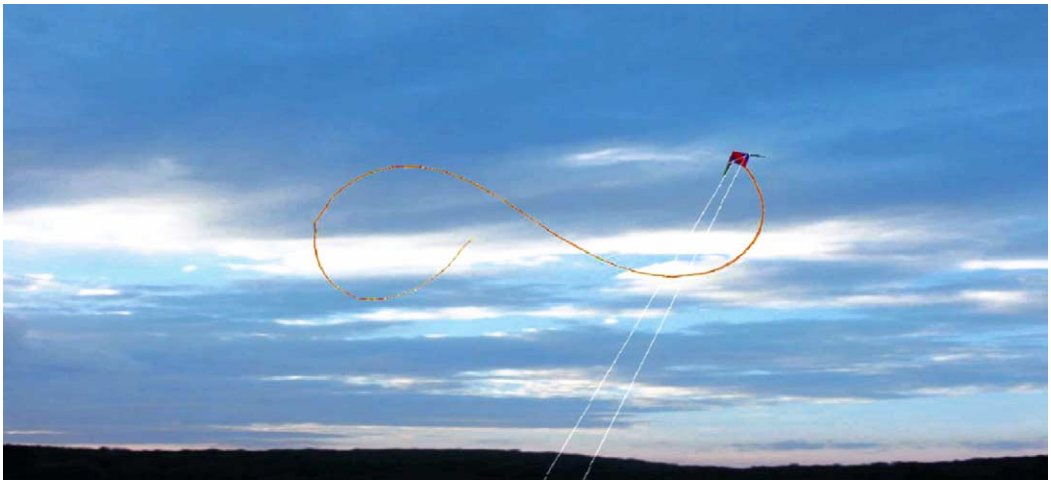

Fig. 1. A picture of the dual line kite with tail.

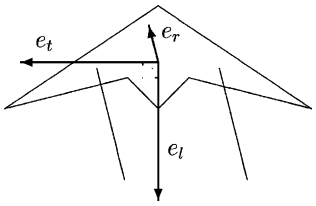

Fig. 2. The kite as seen from the pilot, and unit vectors e_r, e_l, and e_t. Vector e_r need not be perpendicular to e_l, e_t.

longitudinal axis is always in line with the effective wind vector $w_e := w - \dot{p}$, where $w = (v_w, 0, 0)^{\mathrm{T}}$ 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 towards the trailing edge of the kite (cf. Fig. 2), we therefore assume that $e_l = w_e/\|w_e\|$. The transversal axis of the kite can be described by a perpendicular unit vector e_t that is pointing from the right to the left wing tip (as seen from the kite pilot, in upright kite orientation). The orientation of e_t can be controlled, but it has to be orthogonal to e_l (cf. Fig. 2),

$$e_t \cdot e_l = 0. \tag{3}$$

However, the projection of e_t onto the lines' axis (which is given by the vector e_r) is determined from the length difference Δl of the two lines, see Fig. 3. If the distance between the two lines' fixing points on the kite is d, then the vector from the right to the left fixing point is $d \cdot e_t$, and the projection of this vector onto the lines' axis should equal Δl (being positive if the left wing tip is farther away from the pilot), i.e., $\Delta l = d\, e_t \cdot e_r$. Let us define the *lateral angle* ψ to be $\psi = \arcsin(\Delta l/d)$. For simplicity, we assume that we control this angle ψ directly. It determines the orientation of e_t which has to satisfy:

$$e_t \cdot e_r = \frac{\Delta l}{d} = \sin(\psi). \tag{4}$$

A third requirement that e_t must satisfy and which determines its sign is

$$(e_l \times e_t) \cdot e_r > 0. \tag{5}$$

This ensures that the kite is always in the same orientation with respect to the lines. We will now give an explicit form for the unit vector e_t that satisfies (3)–(5). Using the projection $\bar{w}_e$ of the effective wind vector w_e onto the tangent plane spanned by e_θ and e_ϕ, $\bar{w}_e := w_e - e_r(e_r \cdot w_e)$, we can define the orthogonal unit vectors $e_1 := \bar{w}_e/\|\bar{w}_e\|$ and $e_2 := e_r \times e_1$, so that (e_1, e_2, e_r) forms an orthogonal right-handed coordinate basis. In this basis the effective wind w_e has no component in the e_2 direction: $w_e = \|\bar{w}_e\| e_1 + (w_e \cdot e_r) e_r$. The definition

$$e_t := e_1(-\cos\psi \sin\eta) + e_2(\cos\psi \cos\eta) + e_r \sin\psi$$

with $\eta := \arcsin((w_e \cdot e_r/\|\bar{w}_e\|)\tan(\psi))$ indeed satisfies (3)–(5), as can be verified by direct substitution. Therefore, we are now able to determine the orientation of the kite depending on the control ψ and the effective wind w_e only.

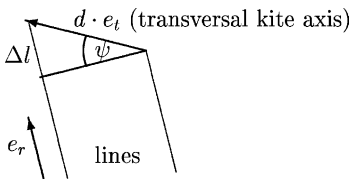

Fig. 3. Kite and lines seen from the top, and visualization of the lateral angle ψ.

Table 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

2.3. Aerodynamic lift and drag

The two vectors $e_l \times e_t$ 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

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

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 $F^{\mathrm{aer}} = F_{\mathrm{L}}(e_l \times e_t) + F_{\mathrm{D}} e_l$ or, in the local coordinate system

$$F^{\mathrm{aer}}_\theta = F_{\mathrm{L}}((e_l \times e_t) \cdot e_\theta) + F_{\mathrm{D}}(e_l \cdot e_\theta),$$
$$F^{\mathrm{aer}}_\phi = F_{\mathrm{L}}((e_l \times e_t) \cdot e_\phi) + F_{\mathrm{D}}(e_l \cdot e_\phi).$$

The system parameters that have been chosen for the simulation model are listed in Table 1. Defining the system state $x := (\theta, \phi, \dot{\theta}, \dot{\phi})^{\mathrm{T}}$ and the control $u := \psi$ we can summarize the four system equations, i.e., (1) and (2) and the trivial equations $\partial\theta/\partial t = \dot{\theta}$, $\partial\phi/\partial t = \dot{\phi}$, in the short form $\dot{x} = f(x, u)$. We mention here that the kite model is freely available on the Internet in form of a MATLAB s-function (Diehl, 2003).

3. Reference orbit and periodic LQR

Using the above system model, a periodic orbit was determined that can be characterized as a "lying eight" and which can be seen in Fig. 4, as a (ϕ, θ)-plot. The wind is assumed to blow in the direction of the p_1-axis ($\theta = 90^\circ$ and $\phi = 0^\circ$). The periodic solution was computed using an off-line variant of the direct multiple shooting method,

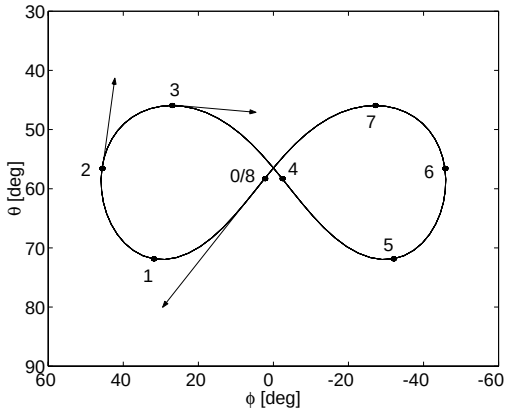

Fig. 4. Periodic reference orbit plotted in the (ϕ, θ)-plane, as seen by the kite pilot. The dots separate intervals of 1 s.

MUSCOD-II (Leineweber, 1999), imposing periodicity conditions with period $T := 8\,\mathrm{s}$ and suitable state bounds and a suitable objective function in order to yield a solution that we considered to be a nice reference orbit. We denote the periodic reference solution by $x_r(t)$ and $u_r(t)$. This solution is defined for all $t \in (-\infty, \infty)$ and satisfies the periodicity condition $x_r(t + T) = x_r(t)$ and $u_r(t + T) = u_r(t)$.

3.1. Open loop stability analysis

Numerical simulations of the kite using the open loop inputs $u_r(t)$ show that the kite crashes onto the ground very quickly after small disturbances, cf. Fig. 5. To analyse the asymptotic stability properties of the open loop system along the periodic reference orbit theoretically, let us consider the linearization of the system along the orbit. An infinitesimal deviation $\delta x(t) := x(t) - x_r(t)$ and $\delta u(t) := u(t) - u_r(t)$ would satisfy the periodically time-varying linear differential equation:

$$\delta\dot{x}(t) = A(t)\delta x(t) + B(t)\delta u(t), \tag{6}$$

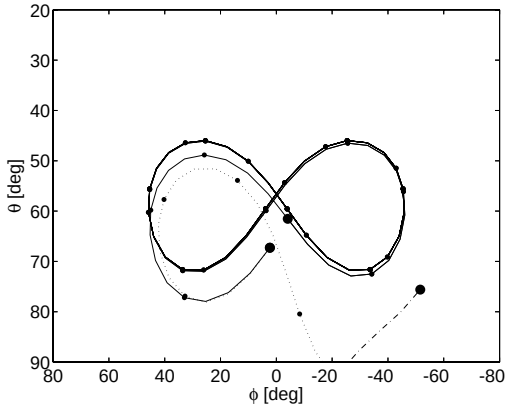

Fig. 5. The linear periodic controller is able to stabilize the system locally from slightly disturbed system states (solid line), in contrast to the open loop system (dotted), but let the kite crash onto the ground for a larger deviation (dash dotted).

with

$$A(t) := \frac{\partial f}{\partial x}(x_r(t), u_r(t)) \quad \text{and} \quad B(t) := \frac{\partial f}{\partial u}(\cdot).$$

Based on the linear time variant periodic system $\dot{x}(t) = A(t)x(t)$ we can compute its fundamental solution and the sensitivity of the final state of each period with respect to its initial value, which is called the "monodromy matrix". A numerical computation of the monodromy matrix for the kite orbit and eigenvalue decomposition yields two eigenvalues ("Floquet multipliers") that are greater than one, confirming the observation that the system is unstable.

3.2. Design of a periodic LQR controller

In order to design a locally stabilizing controller (which is needed if we want to apply the infinite horizon closed loop costing NMPC scheme), we use the classical LQR design technique, applied to the periodic linear system (6). We introduce diagonal weighting matrices:

$$Q := \mathrm{diag}(0.4,\ 1,\ s^2,\ s^2) \quad \text{and} \quad R := 33. \tag{7}$$

To determine the optimal periodic LQR controller that minimizes the objective $\int 1/2(x - x_r(t))^{\mathrm{T}} Q(x - x_r(t)) + 1/2(u - u_r(t))^{\mathrm{T}} R(u - u_r(t))\,\mathrm{d}t$, we find the symmetric periodic matrix solution $P(t)$ for the differential Riccati equation

$$-\dot{P} = Q + A(t)^{\mathrm{T}}P(t) + P(t)A(t) - P(t)B(t)R^{-1}B(t)^{\mathrm{T}}P(t),$$

by integrating the equation backwards for a sufficiently long time, starting with the unit matrix as final value. Once the periodic $P(t)$ is determined, the optimal LQR controller for (6) and (7) is given by $\delta u(t) = -K(t)\delta x(t)$ with $K(t) := R^{-1}B(t)^{\mathrm{T}}P(t)$. We finally define the linear periodic feedback for the original system as

$$u_{\mathrm{LQR}}(x, t) := u_r(t) - K(t)(x - x_r(t)). \tag{8}$$

The linearly stabilized system $\dot{x}(t) = f(x(t), u_{\mathrm{LQR}}(x(t), t))$ is locally stable, as illustrated in Fig. 5. The periodic LQR is available together with the kite model on the kite problem homepage (Diehl, 2003).

4. NMPC controller setup

The aim of the NMPC controller is to stabilize the system in a larger region of attraction and to respect certain bounds that the linear controller may violate, and, furthermore, to lead to an improved performance with respect to a user defined criterion. In our case, the bounds arise because we want the closed loop kite to respect a security distance to the ground ($\theta = 90^\circ$). We achieve this by requiring that $h(x, u, t) := 76.5^\circ - \theta \geq 0$. The performance of the controller is measured by the integral of a function $L(x(t), u(t), t)$, which is in our case the squared deviation of the state x at time t from the reference orbit,

$$L(x, u, t) := \tfrac{1}{2}(x - x_r(t))^{\mathrm{T}} Q(x - x_r(t)).$$

We introduce a sampling interval δ and give NMPC feedback to the system only at the times $t_k := k\delta$. At each sampling time t_k the NMPC shall deliver new controls $u(t), t \in [t_k, t_{k+1}]$ that depend on the current system value $x(t_k)$, where the optimization is based on a prediction of the future system behaviour. Many NMPC schemes exist that guarantee nominal stability, see e.g., Allgöwer, Badgwell, Qin, Rawlings, and Wright (1999), De Nicolao, Magni, and Scattolini (2000) and Mayne (2000); they mainly differ in the way the optimal control problems are formulated. Here, we work in the framework of the infinite horizon closed loop costing scheme (De Nicolao et al., 1998).

4.1. Infinite horizon closed loop costing

In the infinite horizon closed loop costing scheme we express the control $u(t)$ that is actually applied to the plant at time $t \in [t_k, t_{k+1}]$ by the sum

$$u(t) = u_{\mathrm{LQR}}(x(t), t) + v_k, \quad \forall t \in [t_k, t_{k+1}], \tag{9}$$

where the constant vector v_k is determined by the NMPC optimizer and implicitly depends on $x(t_k)$, i.e., we will sometimes write $v_k(x(t_k))$ to keep this dependence in mind. Note that $v_k \equiv 0$ yields the linearly controlled closed loop. In the sequel we will use a bar to distinguish the predicted system state $\bar{x}(t)$ and controls $\bar{u}(t)$ from the state and control vector of the real system.

Given the state $x(t_k)$ of the "real" kite at time t_k, we formulate the following optimal control problem, with control horizon $T_c = M\delta$ and prediction horizon $T_p > T_c$ (where T_p shall ideally be infinity):

$$\min_{\bar{v}_i, \bar{u}(\cdot), \bar{x}(\cdot)} \int_{t_k}^{t_k+T_p} L(\bar{x}(t), \bar{u}(t), t)\, \mathrm{d}t, \tag{10}$$

subject to

$$\begin{aligned}
&\dot{\bar{x}}(t) = f(\bar{x}(t), \bar{u}(t)), \quad \forall t \in [t_k, t_k + T_p],\\
&\bar{x}(t_k) = x(t_k),\\
&\bar{u}(t) = u_{\mathrm{LQR}}(\bar{x}(t), t) + \bar{v}_i, \quad \forall t \in [t_i, t_{i+1}],\\
&\qquad\qquad\qquad\qquad (i = k, \ldots, k + M - 1),\\
&\bar{u}(t) = u_{\mathrm{LQR}}(\bar{x}(t), t), \quad \forall t \in [t_k + T_c, t_k + T_p],\\
&0 \le h(\bar{x}(t), \bar{u}(t), t), \quad \forall t \in [t_k, t_k + T_p].
\end{aligned} \tag{11}$$

In the case that $T_p = \infty$ and if the optimal control problem has a solution for $x(t_0)$, stability of the closed loop trajectory can be proved in a rigorous way (De Nicolao et al., 1998; Magni et al., 2002).

4.2. Practical NMPC setup

We choose a sampling interval $\delta = 1\,\mathrm{s}$ and $M = 8$ sampling intervals as control horizon, $T_c = 8\,\mathrm{s}$, cf. Fig. 8. As the simulation of the periodic system over an infinite horizon is impossible, we employ here a finite $T_p = 24\,\mathrm{s}$, that we believe to be sufficiently long to deliver a fair approximation to the infinite horizon cost. For a theoretical discussion on how to truncate the series expressing the infinite horizon cost associated with the auxiliary linear control law without losing stability see Magni, De Nicolao, Magnani, and Scattolini (2001). Furthermore, to avoid a semi-infinite optimization problem, the problem is changed by imposing the inequality path constraints (11) only at prespecified points in time, here chosen to be the sampling times t_i on the control horizon, as well as start, center and end point of the prediction horizon.

5. Real-time optimization algorithm

The numerical solution of the sequence of optimization problems is achieved by the recently developed *real-time iteration* scheme which we will briefly describe in this section. For details, we refer to Diehl (2002), Diehl et al. (2002b) and Diehl et al. (2002c).

The scheme is based on the *direct multiple shooting* method (Bock & Plitt, 1984) that reformulates the optimization problem into a finite dimensional nonlinear programming problem (NLP) with a special structure, and solves this NLP with an iterative optimization algorithm. In particular, the direct multiple shooting technique keeps both, controls and states, as constrained optimization variables, and does *not* eliminate the state trajectory (which many other NMPC optimization schemes do, e.g., in Li & Biegler, 1989; Oliveira & Biegler, 1995). This temporarily allows infeasible state trajectories, cf. Fig. 8, which reduces nonlinearity of the NLP problem and results in excellent convergence behaviour of the optimization procedure, in particular for tracking problems as in NMPC.

5.1. Software environment

In its actual implementation, the real-time iteration scheme is realized as part of the optimal control package MUSCOD-II (Diehl, Leineweber, & Schäfer, 2001; Leineweber, 1999) which offers several advantages in the context of real-time optimization for NMPC:

- The model equations can either be provided as generic C or Fortran-Code, as a MATLAB s-function or in the gPROMS (PSE, 2000) modelling language (Leineweber, Schäfer, Bock, & Schlöder, 2003).
- Efficient state-of-the-art DAE solvers—e.g., DDASAC (Caracotsios & Stewart, 1985), DAESOL (Bauer, 1999)—are employed to calculate the system trajectories and derivatives quickly and accurately.
- The direct multiple shooting method allows to efficiently treat highly nonlinear and unstable systems (Baake, Baake, Bock, & Briggs, 1992; Diehl et al., 2002a; Kallrath, Bock, & Schlöder, 1993).

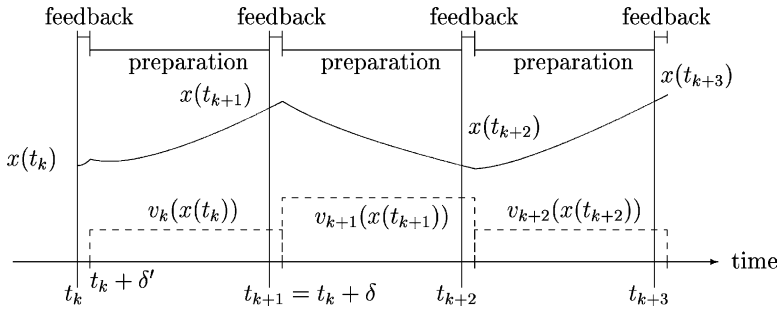

Fig. 6. Use of computation time in the real-time iteration scheme; real system state and controls, with sampling time δ and feedback delay $\delta' \ll \delta$.

- The approach allows a robust treatment of control and path constraints as well as terminal constraints.
- The method can efficiently be parallelized with considerable speedups (Leineweber et al., 2003), if necessary.

5.2. Real-time iteration scheme and initial value embedding

The real-time iteration scheme exploits the fact that in NMPC optimization a sequence of neighboring optimization problems has to be solved. Here, solution information of the previous problem can be exploited for initialization of the following problem by a so called *initial value embedding* strategy. This initialization procedure is so efficient that it allows to perform only one single optimization iteration per optimization problem, without sacrificing much solution accuracy. The sampling time δ can therefore be chosen very small, just as long as needed to perform one optimization iteration. For solution of the kite optimization problem (10), one iteration took never more than 0.5 s, thus safely allowing $\delta = 1$ s.

5.3. Feedback delay for real-time iteration scheme

One further advantage of the scheme is that it avoids the feedback delay of one sampling time δ present in most other NMPC optimization schemes. Instead, each sampling time is divided into a very short "feedback phase", and a much longer "preparation phase", which is used to *prepare* the next feedback, as much as possible without knowledge of the next state $x(t_{k+1})$. Due to this, the real-time iteration scheme has only a very short feedback delay $\delta' \ll \delta$, which is for the kite optimization problem $\delta' \approx 0.05\,\text{s} = (1/20)\delta$. In the presented simulations, we have completely neglected

Fig. 7. The simulated "real" kite just after a random jump of 45° in ϕ.

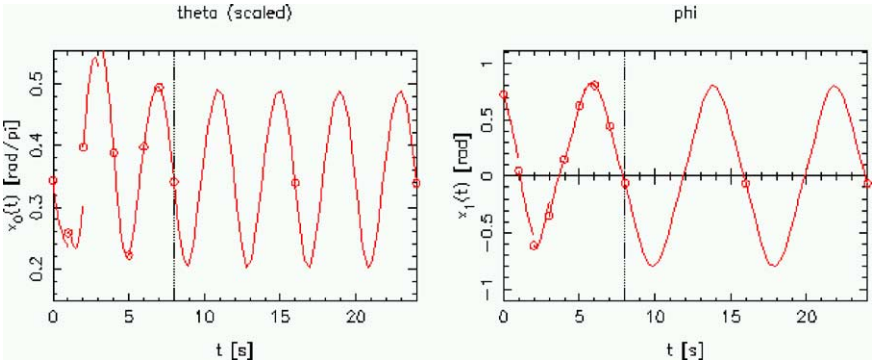

Fig. 8. The prediction horizon in the real-time optimization algorithm just after the random jump of Fig. 7. The state trajectory shows gaps that will be closed after a few iterations if no further disturbance occurs.

this delay, but in a real world application it would be present, as visualized in Fig. 6, such that the closed loop control would instead of Eq. (9) be given by

$$u(t) = u_{\mathrm{LQR}}(x(t), t) + v_k(x(t_k)), \quad \forall t \in [t_k + \delta', t_{k+1} + \delta']. \tag{12}$$

The interplay between "real" kite and the online optimizer using the real-time iteration scheme is illustrated in Figs. 7 and 8, after a large disturbance to the kite.

5.4. Nominal stability in the presence of numerical errors

As mentioned in Section 4, a central question in NMPC is nominal stability of the closed loop, which is theoretically established for the infinite horizon closed loop costing (De Nicolao et al., 1998) employed in this paper. However, in the simulations presented here, the nonlinear optimization problems are only approximately solved by the real-time iteration scheme, so that the presence of numerical errors needs to be addressed.

For the real-time iteration scheme, the state vector of the closed loop consists of the real system state and the content of the prediction horizon in the optimizer (cf. Figs. 7 and 8). Due to the close connection of system and optimizer, stability of the closed loop system can only be addressed by combining concepts from both, NMPC stability theory and convergence analysis of Newton-type optimization methods. This analysis has been carried out for the real-time iteration scheme in conjunction with another NMPC setup, the zero terminal constraint (Mayne & Michalska, 1990), in Diehl, Findeisen, Allgöwer, Bock, and Schlöder (2004), where a proof of nominal stability of the closed loop is given under reasonable assumptions. The line of proof could easily be extended to treat the case of the infinite horizon closed loop costing scheme.

We mention here that the real-time iteration scheme was already applied to a looping kite control problem without LQR prestabilization in Diehl (2002) and Diehl et al. (2002a); there, however, a different optimal control setup was treated that does not address the question of nominal stability.

In another series of tests the real-time iteration scheme was applied to control a pilot scale distillation column described by a stiff differential-algebraic equation model with more than 200 states, making a sampling time of $\delta = 20\,\mathrm{s}$ possible, with feedback delay $\delta' = 0.5\,\mathrm{s}$ (Diehl et al., 2003).

6. Closed loop experiments

In order to test the NMPC closed loop we have performed several numerical experiments. Here, the "real" kite is simulated by a model that coincides with the optimization model, but is subject to disturbances of different type.

6.1. Comparison with LQR

First, let us in Fig. 9 compare the NMPC with the periodic LQR. It can be seen that the NMPC is able to respect the state constraint $\theta \leq 76.5^\circ$ even for the scenario with the

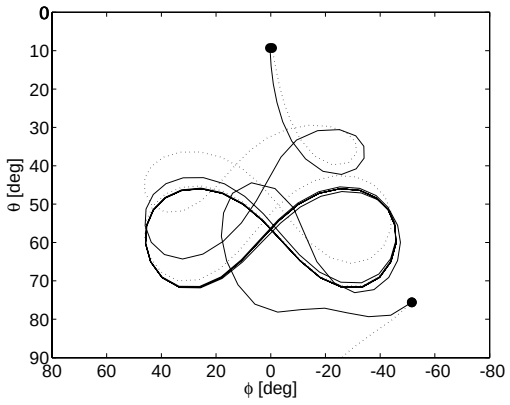

Fig. 9. The NMPC controller is able to control the kite even for the largely disturbed states at the bottom (solid line), in contrast to the LQR controller (dotted, cf. Fig. 5). For the disturbed state at the top, the performance for the NMPC (solid line, integrated costs 1.51) is better that that of the LQR (dotted, costs 1.75), as expected.

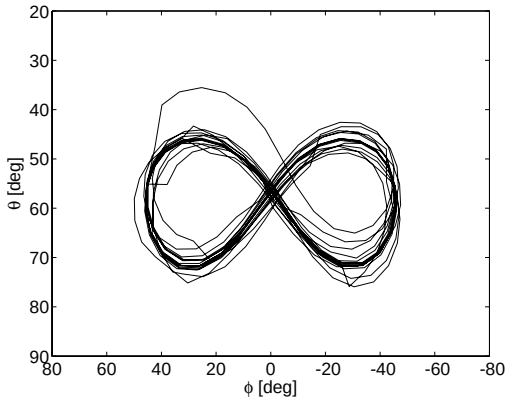

Fig. 10. NMPC closed loop response under moderate random disturbances, for 150 s.

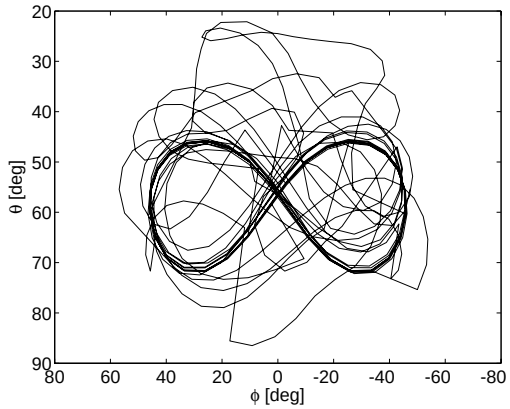

Fig. 11. NMPC closed loop response under large random disturbances for 150 s. Some system states violate the upper bound for θ so that the optimization problems are infeasible.

largely disturbed initial state at the bottom (cf. Fig. 5), in contrast to the LQR. For another scenario, where the system kite starts much too high at the sky, both controllers are able to stabilize the system. However, NMPC leads to a reduced objective with the cost integral $\int_0^\infty L(x(t), u(t), t)\,\mathrm{d}t$ being 1.51 in contrast to 1.75 for the LQR. This difference can be seen in form of a considerably faster convergence towards the periodic orbit.

6.2. Random kicks

Let us now investigate the response of the NMPC controller in the presence of random kicks. The kicks occur at random points in time, but in average once per period. At each kick the system state x is disturbed by independent random disturbances Δx of moderate size: in a first scenario, the disturbances are bounded as $|\Delta\theta| \leq 5^\circ$, $|\Delta\phi| \leq 5^\circ$, $|\Delta\dot{\theta}| \leq 5^\circ/\mathrm{s}$, and $|\Delta\dot{\phi}| \leq 5^\circ/\mathrm{s}$. The corresponding NMPC closed loop response can be seen in Fig. 10, for a simulation of 150 s.

In a second test, we augmented the size of the disturbances by a factor of 4, to $|\Delta\theta| \leq 20^\circ$, etc. The closed loop response for 150 s can be seen in Fig. 11. Some of the occurring optimization problems are infeasible, due to the state constraint requiring $\theta \geq 76.5^\circ$ that is already violated by the initial state $x(t_k)$. However, it can be seen that the online optimizer is sufficiently robust even in the presence of such optimization problems that should never occur in practice.

6.3. Robustness test with strong sidewind

In another scenario, the closed loop is tested against model uncertainty: we consider a continuing disturbance resulting from a change in the wind direction. The wind component in p_2-direction, that is assumed by the optimizer to be zero, is for the "real" kite set to a value 3 m/s that is 50% of the nominal wind v_w in p_1 direction. The NMPC closed loop results in a considerably disturbed but stable periodic orbit, as can be seen in Fig. 12; the disturbed periodic orbit is reached after a very short transient. This contrasts sharply with the open loop and the LQR closed loop response which both result in a crash after a short time.

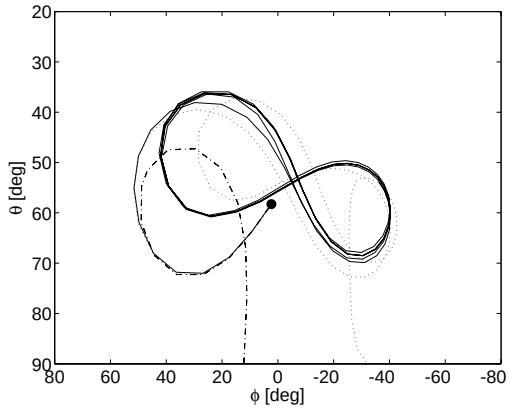

Fig. 12. Effect of model uncertainty in form of a strong side wind with 50% the nominal wind speed. After a very short transient, the NMPC controlled kite loops in a considerably disturbed periodic orbit, but remains stable (solid line). The LQR closed loop response results in a crash after two periods (dotted), and the open loop crashes after 4 s (dash dotted).

7. Conclusions

We have presented a method to design a nonlinear model predictive controller for periodic unstable systems, and have applied the method to a kite that shall fly loops. The method is based on the "infinite horizon closed loop costing" which requires a locally prestabilizing feedback. This prestabilization is achieved by a periodic LQR controller based on a system linearization along the periodic orbit. The NMPC controller uses an objective which only penalizes state deviations and a state constraint is formulated to ensure that the kite does not crash onto the ground. The resulting optimal control problems are solved in real-time, once a second, by a state-of-the-art online optimization algorithm, the

"real-time iteration scheme". This numerical scheme avoids the large feedback delay present in most optimization approaches to NMPC and allows to reduce sampling times to a minimum. The NMPC closed loop gives an excellent response to strong disturbances and the optimizer performs well even when large random disturbances are applied to the kite. Furthermore, the NMPC closed loop shows good robustness against model plant mismatch: in the presence of additional sidewind of 50% the nominal wind velocity the periodic orbit changes shape, but remains stable. As the kite model may serve as a benchmark problem for periodic NMPC it can be freely downloaded (Diehl, 2003).

We want to mention here that the real-time iteration NMPC scheme used for the computations in this paper has also been successfully applied to a real pilot scale distillation column described by a stiff differential-algebraic equation model with over 200 states, making a sampling time of 20 s possible (Diehl et al., 2003).

Acknowledgements

Financial support by the DFG within priority program 469 is gratefully acknowledged, as well as support by the "Institute of Mathematics and its Applications", University of Minnesota, which hosted the first author while parts of the paper have been developed. The first author also wants to thank C. Geider and S. Kredel for setting up the kite visualization. The second and the third authors acknowledge the partial financial support by MURST Project "New techniques for identification and adaptive control of industrial systems".

References

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

Baake, E., Baake, M., Bock, H., & Briggs, K. (1992). Fitting ordinary differential equations to chaotic data. *Physical Review A, 45*, 5524–5529.

Bauer, I. (1999). *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, download at: http://www.ub.uni-heidelberg.de/archiv/1513.

Bock, H., & Plitt, K. (1984). A multiple shooting algorithm for direct solution of optimal control problems. *Proceedings of the 9th IFAC World Congress Budapest*. Pergamon Press.

Caracotsios, M., & Stewart, W. (1985). Sensitivity analysis of initial value problems with mixed odes and algebraic equations. *Computers and Chemical Engineering, 9*, 359–365.

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

De Nicolao, G., Magni, L., & Scattolini, R. (1998). Stabilizing receding-horizon control of nonlinear time varying systems. *IEEE Transactions on Automatic Control, AC-43*, 1030–1036.

De Nicolao, G., Magni, L., & Scattolini, R. (2000). Stability and robustness of nonlinear receding horizon control. In F. Allgöwer & A. Zheng (Eds.), *Nonlinear predictive control. Progress in systems theory* (Vol. 26, pp. 3–23). Basel: Birkhäuser.

Diehl, M. (2002). Real-time optimization for large scale nonlinear processes. *Fortschr.-Ber. VDI Reihe 8, Meß-, Steuerungs- und Regelungstechnik* (Vol. 920). Düsseldorf: VDI Verlag, download also at: http://www.ub.uni-heidelberg.de/archiv/1659/.

Diehl, M. (2003, September). *The kite benchmark problem homepage*. http://www.iwr.uni-heidelberg.de/∼Moritz.Diehl/KITE/kite.html.

Diehl, M., Bock, H., & Schlöder, J. (2002a). Newton-type methods for the approximate solution of nonlinear programming problems in real-time. In G. D. Pillo & A. Murli (Eds.), *High performance algorithms and software for nonlinear optimization*. Kluwer Academic Publishers.

Diehl, M., Bock, H., Schlöder, J., Findeisen, R., Nagy, Z., & Allgöwer, F. (2002b). Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations. *Journal of Processing and Control, 12*(4), 577–585.

Diehl, M., Findeisen, R., Allgöwer, F., Bock, H., & Schlöder, J. (2004). Nominal stability of the real-time iteration scheme for nonlinear model predictive control. *IEE Proceedings on Control Theory and Applications*, in press.

Diehl, M., Findeisen, R., Schwarzkopf, S., Uslu, I., Allgöwer, F., Bock, H., Gilles, E., & Schlöder, J. (2002c). An efficient algorithm for nonlinear model predictive control of large-scale systems. Part I: Description of the method. *Automatisierungstechnik, 50*(12), 557–567.

Diehl, M., Findeisen, R., Schwarzkopf, S., Uslu, I., Allgöwer, F., Bock, H., Gilles, E., & Schlöder, J. (2003). An efficient algorithm for nonlinear model predictive control of large-scale systems. Part II: Application to a distillation column. *Automatisierungstechnik, 51*(1), 22–29.

Diehl, M., Leineweber, D. B., & Schäfer, A. A. S. (2001). *MUSCOD-II Users' Manual*. IWR-Pre-print 2001-25, University of Heidelberg.

Kallrath, J., Bock, H., & Schlöder, J. (1993). Least squares parameter estimation in chaotic differential equations. *Celestial Mechanics and Dynamical Astronomy, 56*, 353–371.

Leineweber, D. (1999). Efficient reduced SQP methods for the optimization of chemical processes described by large sparse DAE models. *Fortschr.-Ber. VDI Reihe 3, Verfahrens-technik* (Vol. 613). Düsseldorf: VDI Verlag.

Leineweber, D., Schäfer, A., Bock, H., & Schlöder, J. (2003). An efficient multiple shooting based reduced SQP strategy for large-scale dynamic process optimization. Part II: Software aspects and applications. *Computers and Chemical Engineering, 27*, 167–174.

Li, W., & Biegler, L. (1989). Multistep, Newton-type control strategies for constrained nonlinear processes. *Chemical Engineering Research and Design, 67*, 562–577.

Magni, L., De Nicolao, G., Magnani, L., & Scattolini, R. (2001). A stabilizing model-based predictive control for nonlinear systems. *Automatica, 37*(9), 1351–1362.

Magni, L., Scattolini, R., & Astrom, K. J. (2002). Global stabilization of the inverted pendulum using model predictive control. *15th IFAC World Congress*. Barcelona.

Mayne, D. (2000). Nonlinear model predictive control: Challenges and opportunities. In F. Allgöwer & A. Zheng (Eds.), *Nonlinear predictive control. Progress in systems theory* (Vol. 26, pp. 23–44). Basel: Birkhäuser.

Mayne, D., & Michalska, H. (1990). Receding horizon control of nonlinear systems. *IEEE Transactions on Automatic Control, 35*(7), 814–824.

Oliveira, N., & Biegler, L. (1995). An extension of Newton-type algorithms for nonlinear process control. *Automatica, 31*(2), 281–286.

PSE. (2000). *gPROMS Advanced User's Manual*. London.

Qin, S., & Badgwell, T. (2001). Review of nonlinear model predictive control applications. In B. Kouvaritakis & M. Cannon (Eds.), *Nonlinear model predictive control: Theory and application* (pp. 3–32). The Institute of Electrical Engineers, London.