Preprints of the 7th IFAC Symposium on Nonlinear Control Systems
21-24 August, 2007, Pretoria, South Africa

AN ONLINE ACTIVE SET STRATEGY FOR FAST ADJOINT BASED NONLINEAR MODEL PREDICTIVE CONTROL

Leonard Wirsching * Hans Joachim Ferreau **
Hans Georg Bock * Moritz Diehl **

* *Interdisciplinary Center for Scientific Computing (IWR), University of Heidelberg, Germany*
** *Optimization in Engineering Center (OPTEC) and Department of Electrical Engineering, K. U. Leuven, Belgium*

Abstract:
Although nonlinear model predictive control (NMPC) has become a well-established control approach its application to fast systems with millisecond timescales is still a challenge. In order to reduce the online computational effort significantly, we propose to combine two recently developed techniques: the adjoint based real-time iteration scheme which allows treatment of nonlinear processes while keeping constraint sensitivities fixed and the online active set strategy for fast solution of the resulting parametric quadratic program. Considerable performance enhancement compared with the standard real-time iteration scheme is demonstrated by controlling a system with fast dynamics. *Copyright © 2007 IFAC*

Keywords: Nonlinear model predictive control, adjoint based real-time iteration scheme, parametric quadratic programming, online active set strategy

1. INTRODUCTION

Nonlinear model predictive control (MPC) has become a well-established approach for closed-loop control of constrained dynamical systems. Theoretical aspects like stability are well understood (Mayne *et al.*, 2000), efficient algorithms exists and have been employed to a number of applications (Qin and Badgwell, 2000). However, controlling systems with fast dynamics in real-time is still a challenge as a nonlinear programming (NLP) problem has to be solved at each sampling instant.

One approach to speed-up the calculation of the optimal control action as well as to shorten the feedback delay is the real-time iteration scheme (RTI) (Diehl, 2001; Diehl *et al.*, 2005*a*) in the context of direct multiple shooting. Wirsching *et al.* (2006) presented a novel adjoint based RTI scheme which reduces the online computational effort considerably and is suited for moderately nonlinear systems (see also Bock *et al.* (2006)). In this paper we propose to combine this adjoint variant of the RTI with the recently developed online active set strategy (Ferreau *et al.*, 2006) for parametric quadratic programming. For optimal control problems where the QP solution time cannot be neglected—which is typically the case for fast systems with many control variables—this concept promises further computational savings compared to the standard or adjoint RTI.

The article is organised as follows: section 2 summarises the main ideas of direct multiple shooting and the standard real-time iteration scheme. Af-

terwards, the concept of adjoint based sensitivity generation is described in section 3. Section 4 gives a brief outline of the online active set strategy. A first test example is given is section 5; section 6 concludes.

2. REAL-TIME ITERATION SCHEME FOR NMPC

For discretising the infinite-dimensional optimal control problem

$$\min_{y(\cdot),u(\cdot)} \int_0^T L(y(t),u(t))\,dt, \tag{1a}$$

$$\text{s. t.} \quad \dot{y}(t) = f(y(t),u(t)) \tag{1b}$$

$$0 = y(0) - y_0 \tag{1c}$$

$$0 \le h(y(t),u(t)) \tag{1d}$$

we apply the *direct multiple shooting* method which is due to Bock and Plitt, see (Bock and Plitt, 1984).

We divide the control horizon $[0,T]$ into subintervals $[t_i,t_{i+1}]$ with

$$0 = t_0 < t_1 < \ldots < t_N = T. \tag{2}$$

On each subinterval $[t_i,t_{i+1}]$ we set

$$u(t) = q_i, \tag{3}$$

i.e. we choose the controls to be piecewise constant. We could also choose the controls from any other function subspace which can be parametrised by finite dimensional parameter vectors q_i, e.g. piecewise linear or piecewise cubic controls. In the case of piecewise constant controls, bounds on the control u transfer immediately to bounds on the parameter vectors q_i and vice versa.

As additional degrees of freedom we introduce variables s_i $(i = 0,\ldots,N-1)$ that are used as initial values for the state trajectories $y_i(t)$ on each subinterval $[t_i,t_{i+1}]$ and a variable s_N for the final state. We compute the trajectories as the solutions of the corresponding initial value problems:

$$\dot{y}_i(t) = f(y_i(t),q_i), \tag{4a}$$

$$y_i(t_i) = s_i\,. \tag{4b}$$

To ensure continuity of the whole state trajectory $y(t)$ we set up the *matching conditions*

$$s_{i+1} - y_i(t_{i+1};s_i,q_i) = 0, \quad i = 0,\ldots,N-1. \tag{5}$$

By evaluating the objective function on each subinterval independently and imposing the path constraints in the grid points only, the following nonlinear programming problem is obtained:

$$\min_{s_i,q_i} \sum_{i=0}^{N-1} \int_{t_i}^{t_{i+1}} L(y_i(t),q_i)\,dt \tag{6a}$$

$$\text{s. t.} \quad 0 = s_0 - y_0, \tag{6b}$$

$$0 = s_{i+1} - y_i(t_{i+1};s_i,q_i), \tag{6c}$$

$$0 \le h(s_i,q_i). \tag{6d}$$

This NLP is large but structured and its Lagrangian is separable in the sense that the Hessian of the Lagrangian is block diagonal. Note the important fact that the parameter y_0 enters the constraints only linearly and only in the initial condition (6b). Therefore, we can write problem (6) in a more general form as

$$\min_w a(w), \quad \text{s. t.} \begin{cases} 0 = b(w) + Ty_0 \\ 0 \le c(w) \end{cases}$$

with $w = (s_0,\ldots,s_N,q_0,\ldots,q_{N-1})$ and a suitable matrix T. The Lagrangian function of this problem is

$$\mathcal{L}(w,\lambda,\mu) = a(w) - \lambda^T\left(b(w) + Ty_0\right) - \mu^T c(w). \tag{7}$$

For solving the NLP a standard method would be *sequential quadratic programming* (SQP). Starting with an initial guess w_0 for the primal variables and (λ_0,μ_0) for the Lagrange multipliers of the equality and inequality constraints, we solve in each iteration the quadratic subproblem

$$\min_{\Delta w_k} \frac{1}{2}\Delta w_k^T A_k \Delta w_k + \nabla a(w_k)^T \Delta w_k \tag{8a}$$

$$\text{s. t.} \quad \frac{db}{dw}(w_k)\,\Delta w_k + b(w_k) + Ty_0 = 0 \tag{8b}$$

$$\frac{dc}{dw}(w_k)\,\Delta w_k + c(w_k) \ge 0 \tag{8c}$$

with A_k being (an approximation of) the Hessian of the Lagrangian of NLP (6) and then update the problem variables.

Note that the Jacobian $\frac{db}{dw}$ has a special block banded structure and is invertible with respect to $(s_0,\ldots,s_N)$. So we can use (8b) to reduce the degrees of freedom in the quadratic subproblem to the variables $\Delta q = (\Delta q_0,\ldots,\Delta q_{N-1})$. We obtain

$$\Delta w_k = \hat{b}(w_k) + \hat{T}_k y_0 + Q_k \Delta q_k \tag{9}$$

with suitable vector $\hat{b}(w_k)$ and matrices $\hat{T}_k$ and Q_k. Substituting this relation in (8) we obtain the following parametric quadratic program:

$$\min_{\Delta q_k} \frac{1}{2}\Delta q_k^T H_k \Delta q_k + \left(F_k y_0 + \hat{g}_k\right)^T \Delta q_k \tag{10a}$$

$$\text{s. t.} \quad G_k \Delta q_k \ge \hat{c}_k + E_k y_0, \tag{10b}$$

again, with suitable matrices H_k, G_k, F_k, E_k and vector $\hat{g}_k$, $\hat{c}_k$. For detailed information about this *condensing technique* the reader is referred to (Diehl, 2001).

As we see from (6), we have to solve neighbouring optimisation problems in NMPC, which differ only in the parameter y_0. Hence an important

question is how to exploit efficiently solution information of the previous problem for initialization of the following problem. The *initial value embedding* strategy proposes to use not only the control profile but the full solution of the previous problem as initialisation data. This means that one refuses to use the known new initial value y_0 and accepts a violation of the initial value constraint (6b) at the initial guess. However, one can show that already the first QP solution in a full step exact Hessian SQP method provides a first order approximation of the exact solution if the problem is initialised in the solution of the previous problem, even in the presence of a change in the active set. The initial value constraint (6b) is satisfied after one full step SQP iteration.

This observation motivates the real-time iteration (RTI) scheme that proceeds as follows: instead of solving a sequence of NLPs (6), we regard the SQP method to be iterating infinitely, and change the initial value in each iteration to the most current state observation, $y_0^{(k)}$. The constraint (6b) is then satisfied after each iteration. At the same time, the iterations contain a corrector term, that drives the SQP iterates further towards the exact solution. If $y_0^{(k)}$ would remain constant, the usual full step SQP method would be recovered. For the Hessian we use a Gauss-Newton approximation in the case of least squares cost functions.

The RTI scheme allows to shorten the sampling intervals significantly. In contrast to conventional NMPC schemes, where one has to apply the old controls until the converged solution is available, one can sooner apply corrected controls. Furthermore, the calculations can be divided into a *preparation phase* that can be performed without knowledge of y_0, and a much shorter *feedback phase* that allows to make the delay even shorter than the sampling time. This remaining delay is typically orders of magnitude smaller than the sampling time and thus we can consider the feedback to be instantaneous. So the sampling time is practically only needed to prepare the *following* real-time iteration.

For a more detailed description of the real-time iteration scheme as well as its convergence and nominal stability properties we refer to (Diehl, 2001; Diehl *et al.*, 2005*b*).

3. ADJOINT BASED SENSITIVITY GENERATION

Although the standard RTI scheme as presented above results in a significant reduction of feedback delay, the possible reduction of the sampling intervals is limited by some important issues: during the preparation phase one has to evaluate the constraint Jacobians needed in QP (8). This is particularly expensive for the Jacobian of the matching conditions since its evaluation requires calculating the derivatives of solutions of a dynamical system with respect to initial values and control parameters. Furthermore, the QP has to be condensed to obtain the smaller QP (10). Time consumption for both issues scales significantly with the number of differential states, controls and multiple shooting intervals. During the feedback phase we have to efficiently treat QP (10) which comprises the potential necessity of limiting the maximum number of active set changes while retaining meaningful iterates.

To reduce time consumption for derivative calculation, one would like to use cheap approximations B_k and C_k for the constraint Jacobians $\frac{db}{dw}(w_k)$ and $\frac{dc}{dw}(w_k)$ in (8), respectively. This is indeed possible, but one has to replace the objective gradient $\nabla a(w_k)$ in (8) by the "modified gradient"

$$m(w_k,\lambda_k,\mu_k)^T := \nabla a(w_k)^T + \lambda_k^T \left(B_k - \frac{db}{dw}(w_k)\right) + \mu_k^T \left(C_k - \frac{dc}{dw}(w_k)\right). \quad (11)$$

To calculate $m(w_k,\lambda_k,\mu_k)^T$ we have to compute vector-Jacobian products $-\lambda_k^T \frac{db}{dw}(w_k)$ and $-\mu_k^T \frac{dc}{dw}(w_k)$. This can be done very efficiently, e.g. by applying the backward mode of automatic differentiation (Griewank, 2000) to the integration scheme and the constraint functions. It can be shown that the cost for evaluating these vector-Jacobian products is only three to five times the cost of an evaluation of the corresponding constraint function. Local convergence of the SQP method with constraint Jacobian approximations and details on the calculation of $m(w_k,\lambda_k,\mu_k)^T$ have been investigated in (Wirsching, 2006).

We propose to use the same fixed approximations for all iterates k, e.g. $A_k := \frac{d^2}{dw^2}\mathcal{L}(w^*,\lambda^*,\mu^*)$, $B_k := \frac{db}{dw}(w^*)$ and $C_k := \frac{dc}{dw}(w^*)$ at some reference solution (w^*,λ^*,μ^*). This has the strong effect that the matrices $\hat{L}_k$ and Q_k and therefore the matrices H_k, G_k, F_k and E_k, are also fixed for all iterations, i.e. we only need to condense QP (8) once which can be done offline. One only has to compute the gradient vector $g_k(y_0) := \hat{g}_k + F^T y_0$ and the constraint vector $d_k(y_0) := \hat{c}_k + E y_0$ in each iteration [1] and then solve the quadratic program

[1] Note that gradient as well as constraint vector also implicitly depend on the current iterate (w_k,λ_k,μ_k) via the (condensed) modified gradient.

$$\mathrm{QP}(y_0): \min_{x_k} \frac{1}{2} x_k^T H x_k + g_k(y_0)^T x_k \quad (12a)$$
$$\text{s.t.} \quad G x_k \geq d_k(y_0), \quad (12b)$$

where $x_k := \Delta q_k$ for ease of notation.

4. AN ONLINE ACTIVE SET STRATEGY FOR PARAMETRIC QUADRATIC PROGRAMMING

For solving QP (12) we propose the recently developed online active set strategy for applications in model predictive control (Ferreau *et al.*, 2006). It builds on the expectation that the active set does not change much from one QP to the next, but is different from conventional warm starting techniques. We give a concise outline of the main ideas and refer the reader to (Ferreau, 2006) for an extensive description including implementation details and degeneracy handling.

For transition from the solved kth quadratic program $\mathrm{QP}(y_0^{(k)})$ to the next one $\mathrm{QP}(y_0^{(k+1)})$, the online active set strategy moves on a straight line in the parameter space, i.e., in the set

$$\mathbb{P} := \{y_0 \in \mathbb{R}^{n_{y_0}} \mid \mathrm{QP}(y_0) \text{ is feasible}\}. \quad (13)$$

Using the definitions

$$\Delta y_0 := y_0^{(k+1)} - y_0^{(k)}, \quad (14a)$$
$$\Delta g := g_k(y_0^{(k+1)}) - g_k(y_0^{(k)}) \quad (14b)$$
$$= (\hat{g}_{k+1} - \hat{g}_k) + F^T \Delta y_0,$$
$$\Delta d := d_k(y_0^{(k+1)}) - d_k(y_0^{(k)}) \quad (14c)$$
$$= (\hat{c}_{k+1} - \hat{c}_k) + \hat{E} \Delta y_0,$$

gradient and constraint vector are re-parametrised as follows:

$$\tilde{y}_0 : [0,1] \to \mathbb{R}^{n_x}, \ \tilde{y}_0(\tau) := y_0^{(k)} + \tau \Delta y_0, \quad (15a)$$
$$\tilde{g} : [0,1] \to \mathbb{R}^n, \ \tilde{g}_k(\tau) := g(y_0^{(k)}) + \tau \Delta g, \quad (15b)$$
$$\tilde{d} : [0,1] \to \mathbb{R}^m, \ \tilde{d}_k(\tau) := d(y_0^{(k)}) + \tau \Delta d. \quad (15c)$$

We start from the known optimal solution x_k^* and μ_k^* (and a corresponding working set $\mathbb{A}$) of the kth $\mathrm{QP}(y_0^{(k)})$ and want to solve $\mathrm{QP}(y_0^{(k+1)})$. The basic idea of the online active set strategy is to move from $y_0^{(k)}$ towards $y_0^{(k+1)}$, and thus from (x_k^*, μ_k^*) towards (x_{k+1}^*, μ_{k+1}^*), while keeping primal and dual feasibility (i.e. optimality) for all intermediate points. This means that we are looking for homotopies ($\mathbb{M} := \{1, \ldots, m\}$)

$$\tilde{x}^* : [0,1] \to \mathbb{R}^n, \ \tilde{x}^*(0) = x_k^*, \ \tilde{x}^*(1) = x_{k+1}^*, \quad (16a)$$
$$\tilde{\mu}^* : [0,1] \to \mathbb{R}^m, \ \tilde{\mu}^*(0) = \mu_k^*, \ \tilde{\mu}^*(1) = \mu_{k+1}^*, \quad (16b)$$
$$\tilde{\mathbb{A}} : [0,1] \to 2^{\mathbb{M}}, \ \tilde{\mathbb{A}}(0) = \mathbb{A}, \ \tilde{\mathbb{A}}(\tau) \subseteq \mathbb{M}, \quad (16c)$$
$$\tilde{\mathbb{I}} : [0,1] \to 2^{\mathbb{M}}, \ \tilde{\mathbb{I}}(\tau) := \mathbb{M} \setminus \tilde{\mathbb{A}}(\tau), \quad (16d)$$

which satisfy the well-known KKT conditions (see e.g. (Boyd and Vandenberghe, 2004)) at every point $\tau \in [0,1]$:

$$\begin{pmatrix} H & G_{\tilde{\mathbb{A}}(\tau)}^T \\ G_{\tilde{\mathbb{A}}(\tau)} & 0 \end{pmatrix} \begin{pmatrix} \tilde{x}^*(\tau) \\ -\tilde{\mu}_{\tilde{\mathbb{A}}(\tau)}^*(\tau) \end{pmatrix} = \begin{pmatrix} -\tilde{g}(\tau) \\ \tilde{d}_{\tilde{\mathbb{A}}(\tau)}(\tau) \end{pmatrix}, \quad (17a)$$
$$\tilde{\mu}_{\tilde{\mathbb{I}}(\tau)}^*(\tau) = 0, \quad (17b)$$
$$G_{\tilde{\mathbb{I}}(\tau)} \tilde{x}^*(\tau) \geq d_{\tilde{\mathbb{I}}(\tau)}(\tau), \quad (17c)$$
$$\tilde{\mu}_{\tilde{\mathbb{A}}(\tau)}^*(\tau) \geq 0. \quad (17d)$$

Since $\tilde{x}^*(\tau)$ and $\tilde{\mu}^*(\tau)$ are piecewise linear functions ($\tilde{x}^*(\tau)$ is even continuous), as already shown in (Fiacco, 1983), locally we must have a relation of the form

$$\tilde{x}^*(\tau) := x_k^* + \tau \Delta x^*, \quad (18a)$$
$$\tilde{\mu}_{\mathbb{A}}^*(\tau) := (\mu_k^*)_{\mathbb{A}} + \tau \Delta \mu_{\mathbb{A}}^*, \quad (18b)$$

which holds for sufficiently small $\tau \in [0, \tau_{\max}]$, $\tau_{\max} \geq 0$.

We know that conditions (17) are met at $\tau = 0$, as we start from an optimal solution. Therefore equality (17a) is satisfied for all $\tau \in [0, \tau_{\max}]$ if and only if

$$\begin{pmatrix} H & G_{\mathbb{A}}^T \\ G_{\mathbb{A}} & 0 \end{pmatrix} \begin{pmatrix} \Delta x^* \\ -\Delta \mu_{\mathbb{A}}^* \end{pmatrix} = \begin{pmatrix} -\Delta g \\ \Delta d_{\mathbb{A}} \end{pmatrix} \quad (19)$$

holds. Because linear independence of the rows of $G_{\mathbb{A}}$ can easily be ensured (Best, 1996), Eq. (19) has a unique solution.

As long as we stay in one critical region, the QP solution depends affinely on y_0, but it might happen that we have to cross the boundaries of critical regions during our way along the straight line. The active set stays constant as long as no previously inactive constraint becomes active, i.e.

$$G_i^T (x_k^* + \tau \Delta x^*) = d_i(y_0^{(k)}) + \tau \Delta d_i$$

for some $i \in \mathbb{I}$, and no previously active constraint becomes inactive, i.e.

$$(\mu_k^*)_i + \tau \Delta \mu_i = 0$$

for some $i \in \mathbb{A}$. Thus, we determine the maximum possible homotopy step length $\tau_{\max}$ as follows:

$$\tau_{\max}^{\text{prim}} := \min_{\substack{i \in \mathbb{I} \\ G_i^T \Delta x^* < \Delta d_i}} \frac{d_i(y_0^{(k)}) - G_i^T x_k^*}{G_i^T \Delta x^* - \Delta d_i}, \quad (20a)$$

$$\tau_{\max}^{\text{dual}} := \min_{\substack{i \in \mathbb{A} \\ \Delta \mu_i < 0}} -\frac{(\mu_k^*)_i}{\Delta \mu_i}, \quad (20b)$$

$$\tau_{\max} := \min\left\{1, \tau_{\max}^{\text{prim}}, \tau_{\max}^{\text{dual}}\right\}. \quad (20c)$$

This choice of $\tau_{\max}$ ensures that conditions (17c), (17d) remain fulfilled. Moreover, if we define $\Delta \mu_{\mathbb{I}}^* := 0$ then also equality (17b) holds for all $\tau \in [0, \tau_{\max}]$.

If $\tau_{\max}$ equals one, a full step along the homotopy path can be taken and the solution of the new quadratic program QP($y_0^{(k+1)}$) is found. Otherwise, a primal or dual *blocking constraint*—which causes the restriction of $\tau_{\max}$ to below one—indicates a constraint to be added or removed from the working set $\mathbb{A}$. After updating the working set and the KKT matrix in Eq. (19), a new step direction is calculated and the whole procedure repeats until the solution of QP($y_0^{(k+1)}$) is found. This idea is illustrated in Fig. 1.

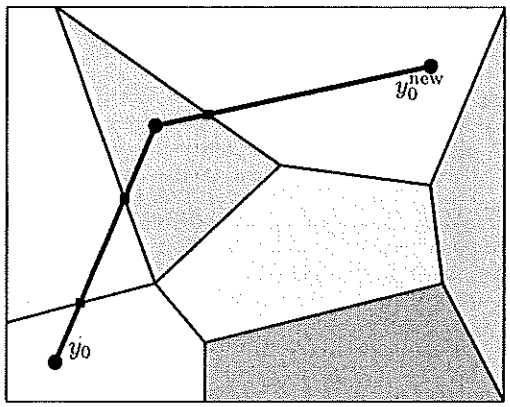

Fig. 1. Homotopy paths from one QP to the next across multiple critical regions.

Real-Time Variant

One big advantage of the online active set strategy is that it produces a *sequence of optimal solutions* for QPs on the homotopy path. Therefore, it is possible to interrupt this sequence after every partial step and start a new homotopy from the current iterate towards the next QP. In particular, no Phase I as in standard (primal) active set methods is necessary because every iterate is optimal and hence feasible. Of course, if the homotopy is interrupted prematurely one might stop at a suboptimal or infeasible point with respect to the QP we want to solve.

Within a real-time setup one can try to find the optimal solution of the current QP within a given sampling time. But if too many active set changes are nessesary to get from the solution of the old QP to that of the current QP one can just stop the solution of the current QP and start a new homotopy towards the solution of the next QP. If the solution of the next QP requires fewer active set changes than computable within the given sampling time the online active set strategy may make up for some unperformed changes from the last QP.

5. NUMERICAL EXAMPLE

We consider the following ODE model for a chain of M spring connnected masses subject to controls applied at both ends

$$\ddot{y}_i = \frac{1}{m}\left(F_{i,i+1} - F_{i-1,i}\right) + g \qquad \forall i = 1, \ldots, M, \tag{21}$$

$$\dot{y}_0 = u_0(t), \tag{22}$$

$$\dot{y}_{M+1} = u_1(t), \tag{23}$$

with spring forces

$$F_{i,i+1} = D\left(1 - \frac{L}{\|y_{i+1} - y_i\|}\right)(y_{i+1} - y_i). \tag{24}$$

All y_i, u_i and $F_{i,i+1}$ are in $\mathbb{R}^3$. A more detailed description of the chain model can be found in (Wirsching, 2006). We choose the following system parameters:

M	m	D	L	g
6	0.03	1.0	0.1	$(0, 0, -9.81)$

The second space coordinate of each mass is subject to bounds depending on the current global time θ, in contrast to the local time parametrisations of the subproblems which always start at zero. We define lower bounds l^j $(j = 1, 2, 3)$

$$y_{i,2}(\theta) \geq \begin{cases} l_i^1 \text{ if } \theta \leq \theta_1 \\ l_i^2 \text{ if } \theta_1 < \theta \leq \theta_2 \\ l_i^3 \text{ if } \theta > \theta_2 \end{cases}, \; i = 1, \ldots, M. \tag{25}$$

where θ_1, θ_2 are fixed global time instants. Upper bounds for $y_{i,2}(\theta)$ are defined by adding a constant offset to l_i^j $(j = 1, 2, 3)$.

The subproblems are formulated with a cost function (1a) using

$$L(y, u) = \alpha \sum_{j=1}^{M} \|\dot{y}_j\|_2^2 + \beta \|u\|_2^2, + \gamma \sum_{j=1}^{M} \left(y_{j,2} - r_j(\theta)\right)^2, \tag{26}$$

where $r_j(\theta)$ is a suitable reference trajectory to prevent the controller from driving the chain too close to the boundaries. The control horizon is 6 seconds, divided into $N = 30$ control intervals of equal length. The sampling time is chosen to be 200 milliseconds.

In Figure 2, CPU time[2] per sampling instant is shown for the standard RTI scheme, the adjoint based RTI scheme and the adjoint based RTI scheme with an open-source implementation of the online active set strategy (qpOASES). All three

[2] All simulations were performed on an Intel Pentium 4 processor with 2.53 GHz (single core), 512 kB L2 cache and 1 GB main memory using gcc 3.3.4 with compiler flag -O3.

schemes generate almost identical control profiles for the test scenario. The standard RTI scheme is obviously not capable for real-time control of the chain. Applying the adjoint based RTI scheme reduces CPU time significantly below 200ms for many sampling instants because of savings in condensing and derivative computations. Violations of the real-time requirement at the remaining sampling instants are caused by the standard QP solver (QPSOL), although warm starts are performed. The online active set strategy (fully converged) turns out to be a suitable remedy: it avoids so-called phase I iterations for finding an initial feasible point and reduces the number of working set changes necessary for finding the optimal solution (by about a factor of two on average). The single violation at $t = 2.6\,\mathrm{s}$ could easily be avoided by slightly limiting the maximum number of working set changes.

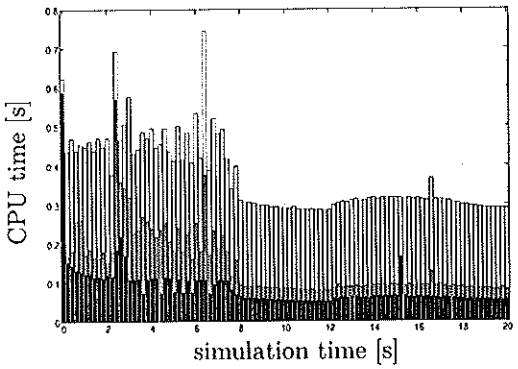

Fig. 2. CPU times per sampling instant for standard RTI scheme (white bars), adjoint based RTI scheme with QPSOL (grey bars) and adjoint based RTI scheme with qpOASES (black bars). Only the adjoint based RTI scheme with qpOASES satisfies the real-time demands of the test scenario.

Note that the initial condensing step as well as the QP matrix factorisations could be performed offline which renders also the first sampling instant real-time feasible.

6. CONCLUSIONS

We clarified that employing the real-time iteration scheme with adjoint based sensitivity generation leads to underlying optimisation problems which can efficiently be solved using the online active set strategy. This observation enables a considerably accelerated online solution of moderately nonlinear NMPC problems comprising many control variables. Thus, optimal control problems that are not tractable with linear MPC but computationally too demanding for fully nonlinear NMPC schemes become feasible, as shown in the preliminary test example.

ACKNOWLEDGEMENTS

Research supported by Research Council KUL: CoE EF/05/006 Optimization in Engineering Center (OPTEC), and by Belgian Federal Science Policy Office: IUAP P6/04 (Dynamical systems, control and optimization, 2007-2011).

REFERENCES

Best, M.J. (1996). *Applied Mathematics and Parallel Computing.* Chap. An Algorithm for the Solution of the Parametric Quadratic Programming Problem, pp. 57–76. Physica-Verlag. Heidelberg.

Bock, H.G. and K.J. Plitt (1984). A multiple shooting algorithm for direct solution of optimal control problems. In: *Proceedings 9th IFAC World Congress Budapest.* Pergamon Press. pp. 243–247.

Bock, H.G., M. Diehl, E.A. Kostina and J.P. Schlöder (2006). Constrained optimal feedback control of systems governed by large differential algebraic equations. In: *Real-Time and Online PDE-Constrained Optimization* (L. Biegler, O. Ghattas, M. Heinkenschloss, D. Keyes and B. van Bloemen Waanders, Eds.). SIAM. (in print).

Boyd, S. and L. Vandenberghe (2004). *Convex Optimization.* University Press. Cambridge.

Diehl, M. (2001). Real-Time Optimization for Large Scale Nonlinear Processes. PhD thesis. Universität Heidelberg.

Diehl, M., H.G. Bock and J.P. Schlöder (2005*a*). A real-time iteration scheme for nonlinear optimization in optimal feedback control. *SIAM Journal on Control and Optimization* **43**(5), 1714–1736.

Diehl, M., R. Findeisen, F. Allgöwer, H.G. Bock and J.P. Schlöder (2005*b*). Nominal stability of the real-time iteration scheme for nonlinear model predictive control. *IEE Proc.-Control Theory Appl.* **152**(3), 296–308.

Ferreau, H.J. (2006). An online active set strategy for fast solution of parametric quadratic programs with applications to predictive engine control. Master's thesis. University of Heidelberg.

Ferreau, H.J., H.G. Bock and M. Diehl (2006). An online active set strategy for fast parametric quadratic programming in MPC applications. In: *Proceedings of the IFAC Workshop on Nonlinear Model Predictive Control for Fast Systems, Grenoble.*

Fiacco, A.V. (1983). *Introduction to sensitivity and stability analysis in nonlinear programming.* Academic Press. New York.

Griewank, A. (2000). *Evaluating Derivatives, Principles and Techniques of Algorithmic Differentiation.* number 19 In: *Frontiers in Appl. Math.*. SIAM. Philadelphia.

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

Qin, S.J. and T.A. Badgwell (2000). An overview of nonlinear model predictive control applications. In: *Nonlinear Predictive Control* (F. Allgöwer and A. Zheng, Eds.). Vol. 26 of *Progress in Systems Theory.* Birkhäuser. Basel Boston Berlin. pp. 370–392.

Wirsching, L. (2006). An SQP algorithm with inexact derivatives for a direct multiple shooting method for optimal control problems. Master's thesis. University of Heidelberg.

Wirsching, L., H. G. Bock and M. Diehl (2006). Fast NMPC of a chain of masses connected by springs. In: *Proceedings of the IEEE International Conference on Control Applications, Munich.* pp. 591–596.