INTERNATIONAL JOURNAL OF ROBUST AND NONLINEAR CONTROL
Int. J. Robust Nonlinear Control 2008; **18**:816–830
Published online 9 July 2007 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/rnc.1251

An online active set strategy to overcome the limitations of explicit MPC

H. J. Ferreau[1,*,†], H. G. Bock[2] and M. Diehl[1]

[1] *Optimization in Engineering Center (OPTEC) and Department of Electrical Engineering, K.U. Leuven, Kasteelpark Arenberg 10, B-3001 Leuven, Belgium*
[2] *Interdisciplinary Center for Scientific Computing (IWR), University of Heidelberg, Im Neuenheimer Feld 368, D-69120 Heidelberg, Germany*

SUMMARY

Nearly all algorithms for linear model predictive control (MPC) either rely on the solution of convex quadratic programs (QPs) in real time, or on an explicit precalculation of this solution for all possible problem instances. In this paper, we present an online active set strategy for the fast solution of parametric QPs arising in MPC. This strategy exploits solution information of the previous QP under the assumption that the active set does not change much from one QP to the next. Furthermore, we present a modification where the CPU time is limited in order to make it suitable for strict real-time applications. Its performance is demonstrated with a challenging test example comprising 240 variables and 1191 inequalities, which depends on 57 parameters and is prohibitive for explicit MPC approaches. In this example, our strategy allows CPU times of well below 100 ms per QP and was about one order of magnitude faster than a standard active set QP solver. Copyright © 2007 John Wiley & Sons, Ltd.

Received 18 October 2006; Revised 30 April 2007; Accepted 29 May 2007

KEY WORDS: model predictive control; parametric quadratic programming; online active set strategy

1. INTRODUCTION

The fast and reliable solution of convex quadratic programming (QP) problems in real time is a crucial ingredient of most algorithms for both linear and nonlinear model predictive control (MPC). The success of linear MPC—where just one QP needs to be solved in each sampling time—can even be attributed to the fact that highly efficient and reliable methods for QP solution

*Correspondence to: H. J. Ferreau, Optimization in Engineering Center (OPTEC) and Department of Electrical Engineering, K.U. Leuven, Kasteelpark Arenberg 10, B-3001 Leuven, Belgium.
†E-mail: joachim.ferreau@esat.kuleuven.be

Contract/grant sponsor: REGINS-PREDIMOT European project
Contract/grant sponsor: Research Council KUL; contract/grant number: CoE EF/05/006
Contract/grant sponsor: Belgian Federal Science Policy Office; contract/grant number: IUAP P6/04

Copyright © 2007 John Wiley & Sons, Ltd.

have existed for decades, and that their computation times are much smaller than the required sampling times in typical applications. On the other hand, in nonlinear MPC algorithms, QPs often arise as subproblems during the iterative nonlinear solution procedure, so that not only one, but several QPs need to be solved in each sampling time. In most MPC algorithms, the arising QPs are treated by well tested and efficient standard methods for optimization.

When sampling times become so short that the computation times for QP solution can no longer be neglected, specialized algorithms that exploit the structure of the QPs arising in MPC algorithms become an interesting alternative. Basically, two approaches to fast QP solution in MPC can be distinguished: first, the explicit, or *off-line QP solution*, which precomputes the QP solution for all possibly arising problem instances. This can be done quite efficiently, as shown by Bemporad *et al.* [1], but is limited to small-state dimensions that are surely well below 20. On the other hand, the *online QP solution* is the classical way to treat the sequence of QPs in MPC for varying initial values. Several QP solution methods exist, among them the most prominent are *active set* methods, which come in two variants, namely *primal* [2, 3] and *dual* [4, 5] active set methods. Active set methods suffer from the drawback that no polynomial bound on the runtime of the algorithm can be given, as has famously been shown by Klee and Minty [6] in the context of linear programming.

Furthermore, (*primal–dual*) *interior point methods*, cf. [7], have emerged as a strong competitor to active set methods, and have also been proposed for the use in MPC [8]. For interior point methods, a polynomial runtime guarantee can be given and they possess relatively constant computational demands. However, they suffer from the drawback that till date no efficient warm start techniques exist.

In this paper, we propose to use an active set strategy that has been developed independently from the MPC context by Best [9]. It is inspired by some important observations from the field of parametric quadratic programming. Our online active set method comes in two variants: while the first is just an alternative way to exactly solve the QPs arising in MPC efficiently (but without theoretical runtime limit), the second one is able to give a central processing unit (CPU) time guarantee. This guarantee, however, comes at the expense of sometimes not solving exactly the QP that we want to solve at a given sampling instant. In these circumstances—that arise, e.g. after large disturbances—instead, an intermediate QP that is in between the previous and the current problem is solved.

This article—which is based on [10]—is organized as follows: in Section 2 the required theoretical background of MPC and parametric QP is briefly summarized. Our online active set strategy is presented in Section 3, which also contains a short discussion on degeneracy handling and implementation details. The section ends with the description of a variant of the proposed online active set strategy that is suitable for real-time applications. A challenging test problem forms the basis of a comparison between our online active set strategy and a standard QP solver in Section 4. Section 5 is devoted to a conclusion and some ideas for future work, e.g. an extension of the proposed online active set strategy to nonlinear MPC.

2. QUADRATIC PROGRAMMING FOR MODEL PREDICTIVE CONTROL

2.1. Optimal control problem formulation

We consider the following *discrete-time*, *time-invariant*, and *linear* system:

$$x_{k+1} = Ax_k + Bu_k \tag{1}$$

Copyright © 2007 John Wiley & Sons, Ltd.

DOI: 10.1002/rnc

in which, for all $k \in \mathbb{N}$, $x_k \in \mathbb{R}^{n_x}$ is the state, $u_k \in \mathbb{R}^{n_u}$ is the input (or controls), and $A \in \mathbb{R}^{n_x \times n_x}$, $B \in \mathbb{R}^{n_x \times n_u}$ are fixed matrices. For simplicity, let us assume that we want to regulate system (1) from the current state x_0 into the origin, even though our method is equally applicable to trajectory tracking. At the same time, we want to guarantee that certain input and state constraints of the form

$$\underline{c} \leqslant Cx_k \leqslant \bar{c} \tag{2a}$$

$$\underline{d} \leqslant Du_k \leqslant \bar{d} \tag{2b}$$

are fulfilled for all $k \in \mathbb{N}$. Therein, $C \in \mathbb{R}^{n_{oc} \times n_x}$ and $D \in \mathbb{R}^{n_{ic} \times n_u}$ are constant matrices and $\underline{c}, \bar{c} \in \mathbb{R}^{n_{oc}}$ and $\underline{d}, \bar{d} \in \mathbb{R}^{n_{ic}}$ are constant vectors satisfying $\underline{c} \leqslant \bar{c}$ and $\underline{d} \leqslant \bar{d}$.

MPC treats such a constrained control problem by repeatedly solving the following optimization problem:

$$\min_{\substack{u_0,\ldots,u_{N-1} \\ x_1,\ldots,x_N}} x_N^{\mathrm{T}} P x_N + \sum_{i=0}^{N-1} (x_i^{\mathrm{T}} Q x_i + u_i^{\mathrm{T}} R u_i) \tag{3a}$$

$$\text{s.t.} \quad x_{k+1} = Ax_k + Bu_k \tag{3b}$$

$$(x_0 \text{ given}) \tag{3c}$$

$$\underline{c} \leqslant Cx_k \leqslant \bar{c} \tag{3d}$$

$$\underline{d} \leqslant Du_k \leqslant \bar{d} \tag{3e}$$

$$c_T \leqslant C_T x_N \tag{3f}$$

where $P \in \mathbb{R}^{n_x \times n_x}$, $Q \in \mathbb{R}^{n_x \times n_x}$ are positive semidefinite matrices, and $R \in \mathbb{R}^{n_u \times n_u}$ is a positive-definite matrix and $N \in \mathbb{N}$ denotes the length of the *prediction horizon* (which is here assumed to equal the length of the control horizon). Matrix $C_T \in \mathbb{R}^{n_T \times n_x}$ and vector $c_T \in \mathbb{R}^{n_T}$ express possible terminal constraints, and P is the *terminal penalty weight matrix*. This matrix could, for example, be obtained by solving a Riccati equation for the unconstrained system.

After the solution of (3) only the first optimal control $(u_0)^*$ is applied as input to the system. After a *sampling time*, say δ, the current state is obtained (measured or estimated) and the optimization problem (3) is solved again with this new initial value for the state. This yields the MPC strategy also known as *receding horizon control*. It is based on a successive solution of the parametric QP (3) for varying parameters x_0.

By eliminating the state variables $x_1, \ldots, x_N$ using Equations (3b), (3c) the optimization problem (3) can be expressed as the following smaller-scale QP:

$$\mathrm{QP}(x_0): \quad \min_{w} \quad \tfrac{1}{2} w^{\mathrm{T}} H w + w^{\mathrm{T}} \underbrace{F^{\mathrm{T}} x_0}_{=:g(x_0)} \tag{4a}$$

$$\text{s.t.} \quad Gw \geqslant \underbrace{\bar{b} + Ex_0}_{=:b(x_0)} \tag{4b}$$

Copyright © 2007 John Wiley & Sons, Ltd.

DOI: 10.1002/rnc

where $w := (u_0^{\mathrm{T}}, \ldots, u_{N-1}^{\mathrm{T}})^{\mathrm{T}} \in \mathbb{R}^n$ denotes the vector of all inputs on the horizon and $H \in \mathbb{R}^{n \times n}$, $F \in \mathbb{R}^{n_x \times n}$, $G \in \mathbb{R}^{m \times n}$, $\bar{b} \in \mathbb{R}^m$, $E \in \mathbb{R}^{m \times n_x}$ (with $n := Nn_u$ and $m := 2n_{\mathrm{ic}} + 2n_{\mathrm{oc}} + n_{c_T}$) are constants and can be easily derived from (3), cf. [1]. We might call the transition from the large structured QP (3) to the smaller, but less structured QP (4) *condensing*, cf. [11].

2.2. *Parametric quadratic programming*

All possible QPs (4) differ only in the gradient $g(x_0)$ in the objective function and the right-hand side vector $b(x_0)$ of the constraints, which are both affine functions of the current state x_0. Thus, the above QP is a *parametric* QP with a particularly favourable structure.

Throughout this paper, we can assume that the QP (4) is *strictly convex*, i.e. the Hessian matrix H is positive definite (this is a straightforward consequence of positive definiteness of the control penalty R). If the *feasible set* of a specific problem QP(x_0), namely the set

$$\mathscr{F}(x_0) := \{w \in \mathbb{R}^n | Gw \geqslant b(x_0)\}$$

is nonempty, we say that QP(x_0) is *feasible*. The *set of feasible parameters* $\mathbb{P}$ is defined as

$$\mathbb{P} := \{x_0 \in \mathbb{R}^{n_x} | \mathscr{F}(x_0) \neq \emptyset\}$$

It is easy to show that $\mathbb{P}$ is a (convex) polyhedron, as it is the projection of the polyhedron $\{(x_0, w) | Gw - Ex_0 \geqslant \bar{b}\}$ onto the x_0 coordinates. For the online active set algorithm presented in this paper, convexity of the set $\mathbb{P}$ is a crucial prerequisite.

We denote the ith row of the constraint matrix G by the vector G_i^{T}, and say that constraint $G_i^{\mathrm{T}} w \geqslant b_i(x_0)$, $1 \leqslant i \leqslant m$, is *active* at point $w_0 \in \mathscr{F}(x_0)$ if $G_i^{\mathrm{T}} w_0 = b_i$ holds. The set of all constraints that are active at point w_0 for problem QP(x_0) is denoted by $\mathbb{A}(x_0, w_0) \subseteq \{1, \ldots, m\}$. Conversely, $\mathbb{I}(x_0, w_0) := \{1, \ldots, m\} \backslash \mathbb{A}(x_0, w_0)$ is called the set of *inactive constraints* at point w_0.

The matrix composed of the rows corresponding to constraints in any index set $\mathbb{A} \subseteq \{1, \ldots, m\}$ (e.g. the active or inactive set) is denoted by $G_{\mathbb{A}}$. The corresponding part of the constraint vector b (or any other vector $y \in \mathbb{R}^m$) is denoted by $b_{\mathbb{A}}$ ($y_{\mathbb{A}}$).

Finding the unique solution of a strictly convex and feasible problem QP(x_0) is equivalent to finding a pair (w^*, y^*) of primal and dual variables (the latter are often called Lagrange multipliers) that satisfies the following well known necessary—and here also sufficient—optimality conditions.

Theorem 2.1 (Karush–Kuhn–Tucker conditions)
Let QP(x_0) (4) be a strictly convex and feasible QP. Then there exists a unique $w^* \in \mathbb{R}^n$ and at least one working set $\mathbb{A}$ and a vector $y^* \in \mathbb{R}^m$ that satisfy the following conditions:

$$Hw^* - G_{\mathbb{A}}^{\mathrm{T}} y_{\mathbb{A}}^* = -g(x_0) \tag{5a}$$

$$G_{\mathbb{A}} w^* = b_{\mathbb{A}}(x_0) \tag{5b}$$

$$y_{\mathbb{I}}^* = 0 \quad (\mathbb{I} := \{1, \ldots, m\} \backslash \mathbb{A}) \tag{5c}$$

$$G_{\mathbb{I}} w^* \geqslant b_{\mathbb{I}}(x_0) \tag{5d}$$

$$y_{\mathbb{A}}^* \geqslant 0 \tag{5e}$$

Furthermore, w^* is the unique global minimizer of (4).

Copyright © 2007 John Wiley & Sons, Ltd.

DOI: 10.1002/rnc

A proof of this basic result can be found in any textbook on optimization, e.g. [12, p. 244]. Note that neither the set $\mathbb{A} \subseteq \mathbb{A}(x_0, w^*)$ nor the dual variables y^* are necessarily unique. If all rows of the matrix $G_{\mathbb{A}}$ are linearly independent and $\mathbb{A}$ is fixed, however, y^* would be uniquely determined from (5a) and (5c). If $\mathbb{A} = \mathbb{A}(x_0, w^*)$, this condition is called *linear independence constraint qualification* (LICQ). Unfortunately, we cannot make this assumption in general within our algorithm (see Section 3.2 for how we treat cases in which the LICQ does not hold).

2.3. *Explicit, or off-line solution of parametric quadratic program*

A popular and very successful approach to treat parametric QPs (4) with low-dimensional x_0 was introduced to the MPC community by Bemporad *et al.* [1]. In contrast to online QP solution approaches, here an explicit representation of the QP solution is computed off-line, i.e. before the runtime of the process. The major observation is that the set $\mathbb{P}$ can be subdivided into polyhedral *critical regions*, on each of which the optimal active set is constant, and that the explicit solution of the parametric QP on each of these critical regions is an affine function of x_0, as has been observed in the context of MPC by Zafiriou [13]. Thus, only the critical regions and the parameters of the affine function need to be stored. A large community of researchers has investigated, applied, and extended this approach, cf. [14–16]. The major advantage of the explicit approach is its fast online computation time, as only the critical region corresponding to the recent value of x_0 must be found and an affine function be evaluated. Its main drawback is its limitation to low-dimensional parameter spaces due to the exponential growth of the offline computation (and storage) requirements in the parameter size. This makes trajectory tracking impossible as the reference trajectory has to be described by additional parameters (whereas the computational complexity of online QP solution is not affected by the dimension of the parameter space $\mathbb{P}$). Moreover, due to the high offline computational cost, another serious problem in practice is that online tuning is difficult. One variant to reduce the off-line complexity of the explicit approach at the expense of a suboptimal online performance was proposed in [17].

2.4. *Online QP solution*

Due to these limitations of the explicit approach—Pannocchia *et al.* [17] even state: 'Practically speaking, the offline computation [...] may be tractable for SISO problems with relatively short control horizon, but it quickly becomes intractable as the [state and input space] dimensions and control horizon grow'—we strongly advocate the use of *online QP solvers*. However, it is often not appropriate to simply use a general purpose QP solver as the QPs arising in MPC have a special structure, e.g. a positive-definite Hessian matrix. This property allows the usage of specially tailored dual active set methods like QPSchur [5] or an adaptation of the classical method of Goldfarb and Idnani [4] that was recently used in MPC applications with sampling rates below 100 µs [18]. Moreover, interior point methods can exploit the convexity of the objective function and have been used in MPC [8]. However, these methods do not take full advantage of the parametric nature of MPC problems, which is most readily exploited by the explicit approach.

Copyright © 2007 John Wiley & Sons, Ltd.

DOI: 10.1002/rnc

3. ONLINE ACTIVE SET STRATEGY FOR MPC

Inspired by the explicit solution approach, but aiming to avoid its prohibitive off-line computational cost, we propose an online active set strategy for use in MPC. 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.

For transition from the 'old' problem to the 'new' one, we propose to move on a straight line in the parameter space, i.e. in the set $\mathbb{P}$. As this set is convex, we can be sure that all QPs on this line remain feasible and can be solved. As long as we stay in one critical region, the QP solution depends affinely on x_0, but it might happen that we have to cross the boundaries of critical regions during our way on the line. The basic idea of our algorithm, the use of a homotopy for parametric QP, has previously been proposed by Best [9] (Figure 1).

3.1. *Main idea*

Let us assume that we have solved a QP (4) for a certain initial state x_0 and (after one sampling time) we want to solve it again for a new initial state vector x_0^{new} with the unknown solution $(w^*_{\text{new}}, y^*_{\text{new}})$. By setting

$$\Delta x_0 := x_0^{\text{new}} - x_0 \tag{6a}$$

$$\Delta g := g(x_0^{\text{new}}) - g(x_0) = F^{\mathrm{T}} \Delta x_0 \tag{6b}$$

$$\Delta b := b(x_0^{\text{new}}) - b(x_0) = \bar{b} + E\Delta x_0 \tag{6c}$$

we can re-parameterize gradient and right-hand side vector as follows:

$$\tilde{x}_0 : [0, 1] \to \mathbb{R}^{n_x}, \quad \tilde{x}_0(\tau) := x_0 + \tau \Delta x_0 \tag{7a}$$

$$\tilde{g} : [0, 1] \to \mathbb{R}^{n}, \quad \tilde{g}(\tau) := g(x_0) + \tau \Delta g \tag{7b}$$

$$\tilde{b} : [0, 1] \to \mathbb{R}^{m}, \quad \tilde{b}(\tau) := b(x_0) + \tau \Delta b \tag{7c}$$

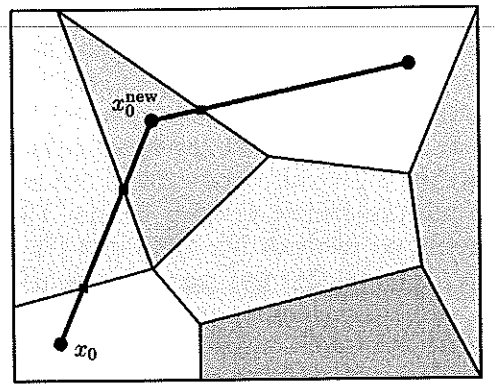

Figure 1. Homotopy paths from one QP to the next across multiple critical regions.

Copyright © 2007 John Wiley & Sons, Ltd.

DOI: 10.1002/rnc

According to our assumption, we know the solution w^* and y^* (and a corresponding working set $\mathbb{A}$) of QP(x_0) and want to solve QP(x_0^{new}). The basic idea of our online active set strategy is to move from x_0 towards x_0^{new}, and thus from (w^*, y^*) towards $(w^*_{\text{new}}, y^*_{\text{new}})$, while keeping primal and dual feasibility (i.e. optimality) for all intermediate points. This means that we are looking for homotopies ($\mathbb{M} := \{1, \ldots, m\}$)

$$\tilde{w}^* : [0,1] \to \mathbb{R}^n, \quad \tilde{w}^*(0) = w^*, \quad \tilde{w}^*(1) = w^*_{\text{new}} \tag{8a}$$

$$\tilde{y}^* : [0,1] \to \mathbb{R}^m, \quad \tilde{y}^*(0) = y^*, \quad \tilde{y}^*(1) = y^*_{\text{new}} \tag{8b}$$

$$\tilde{\mathbb{A}} : [0,1] \to 2^{\mathbb{M}}, \quad \tilde{\mathbb{A}}(0) = \mathbb{A}, \quad \tilde{\mathbb{A}}(\tau) \subseteq \mathbb{M} \tag{8c}$$

$$\tilde{\mathbb{I}} : [0,1] \to 2^{\mathbb{M}}, \quad \tilde{\mathbb{I}}(\tau) := \mathbb{M} \backslash \tilde{\mathbb{A}}(\tau) \tag{8d}$$

which satisfy the conditions of Theorem 2.1 at every point $\tau \in [0,1]$:

$$\begin{pmatrix} H & G^{\mathrm{T}}_{\tilde{\mathbb{A}}(\tau)} \\ G_{\tilde{\mathbb{A}}(\tau)} & 0 \end{pmatrix} \begin{pmatrix} \tilde{w}^*(\tau) \\ -\tilde{y}^*_{\tilde{\mathbb{A}}(\tau)}(\tau) \end{pmatrix} = \begin{pmatrix} -\tilde{g}(\tau) \\ \tilde{b}_{\tilde{\mathbb{A}}(\tau)}(\tau) \end{pmatrix} \tag{9a}$$

$$\tilde{y}^*_{\tilde{\mathbb{I}}(\tau)}(\tau) = 0 \tag{9b}$$

$$G_{\tilde{\mathbb{I}}(\tau)} \tilde{w}^*(\tau) \geqslant b_{\tilde{\mathbb{I}}(\tau)}(\tau) \tag{9c}$$

$$\tilde{y}^*_{\tilde{\mathbb{A}}(\tau)}(\tau) \geqslant 0 \tag{9d}$$

If we assume that the LICQ holds (the degenerate case is discussed in Section 3.2), conditions (9) imply that $\tilde{w}^*(\tau)$ and $\tilde{y}^*(\tau)$ are well defined, piecewise linear and continuous functions, as shown in [19].

Locally, we must have a relation of the form

$$\tilde{w}^*(\tau) := w^* + \tau \Delta w^* \tag{10a}$$

$$\tilde{y}^*_{\mathbb{A}}(\tau) := y^*_{\mathbb{A}} + \tau \Delta y^*_{\mathbb{A}} \tag{10b}$$

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

Since we start from an optimal solution we know that conditions (9) are satisfied at $\tau = 0$. Therefore, equality (9a) is satisfied for all $\tau \in [0, \tau_{\max}]$ if and only if

$$\begin{pmatrix} H & G^{\mathrm{T}}_{\mathbb{A}} \\ G_{\mathbb{A}} & 0 \end{pmatrix} \begin{pmatrix} \Delta w^* \\ -\Delta y^*_{\mathbb{A}} \end{pmatrix} = \begin{pmatrix} -\Delta g \\ \Delta b_{\mathbb{A}} \end{pmatrix} \tag{11}$$

holds. Since it is ensured that all rows of $G_{\mathbb{A}}$ are linearly independent, Equation (11) has a unique solution.

The active set stays constant as long as no previously inactive constraint becomes active, i.e.

$$G_i^{\mathrm{T}}(w^* + \tau \Delta w^*) = b_i(x_0) + \tau \Delta b_i$$

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

$$y_i^* + \tau \Delta y_i = 0$$

Copyright © 2007 John Wiley & Sons, Ltd.

DOI: 10.1002/rnc

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

$$\tau_{\max}^{\text{prim}} := \min_{\substack{i \in \mathbb{I} \\ G_i^T \Delta w^* < \Delta b_i}} \frac{b_i(x_0) - G_i^T w^*}{G_i^T \Delta w^* - \Delta b_i} \tag{12a}$$

$$\tau_{\max}^{\text{dual}} := \min_{\substack{i \in \mathbb{A} \\ \Delta y_i < 0}} -\frac{y_i^*}{\Delta y_i} \tag{12b}$$

$$\tau_{\max} := \min\{1, \tau_{\max}^{\text{prim}}, \tau_{\max}^{\text{dual}}\} \tag{12c}$$

This choice of $\tau_{\max}$ ensures that conditions (9c),(9d) remain fulfilled. Moreover, if we define $\Delta y_{\mathbb{I}}^* := 0$ then also equality (9b) holds for all $\tau \in [0, \tau_{\max}]$. Our online active set strategy is summarized in Algorithm 1.

Algorithm 1. Online active set strategy.

`input`: Data and solution (w^*, y^*) of QP(x_0), corresponding working set $\mathbb{A}$, new parameter $x_0^{\text{new}} \in \mathbb{P}$

`output`: Solution pair $(w_{\text{new}}^*, y_{\text{new}}^*)$ of QP(x_0^{new}), corresponding working set $\mathbb{A}^{\text{new}}$

(1) Calculate Δx_0, Δg and Δb via Equations (6).
(2) Calculate primal and dual step directions Δw^* and Δy^* via Equation (11).
(3) Determine maximum homotopy step length $\tau_{\max}$ from Equations (12).
(4) Obtain optimal solution of QP$(\tilde{x}_0)$:
  (a) $\tilde{x}_0 \leftarrow x_0 + \tau_{\max} \Delta x_0$,
  (b) $\tilde{w}^* \leftarrow w^* + \tau_{\max} \Delta w^*$,
  (c) $\tilde{y}^* \leftarrow y^* + \tau_{\max} \Delta y^*$.
(5) `if` $\tau_{\max} = 1$:
  Optimal solution of QP(x_0^{new}) found. Set $w_{\text{new}}^* \leftarrow \tilde{w}^*$, $y_{\text{new}}^* \leftarrow \tilde{y}^*$ and $\mathbb{A}^{\text{new}} \leftarrow \mathbb{A}$. `stop!`
(6) `if` $\tau_{\max} = \tau_{\max}^{\text{dual}}$:
  Remove a dual blocking constraint j ($\tau_{\max}^{\text{dual}} = -\frac{y_j^*}{\Delta y_j}$) from working set, i.e. $\mathbb{A} \leftarrow \mathbb{A} \backslash \{j\}$.
  `elseif` $\tau_{\max} = \tau_{\max}^{\text{prim}}$:
  Add a primal blocking constraint j ($\tau_{\max}^{\text{prim}} = \frac{b_j(x_0) - G_j^T w^*}{G_j^T \Delta w^* - \Delta b_j}$) to working set, i.e. $\mathbb{A} \leftarrow \mathbb{A} \cup \{j\}$, while ensuring linear independence (see Section 3.2).
(7) Set $x_0 \leftarrow \tilde{x}_0$, $w^* \leftarrow \tilde{w}^*$, $y^* \leftarrow \tilde{y}^*$ and continue with step (1).

3.2. *Degeneracy handling*

Our algorithm requires that the *KKT matrix* in Equation (9a) is nonsingular. Due to the assumed positive definiteness of H this property holds if and only if matrix $G_{\mathbb{A}}$ has full-row rank. Since deletion of a row cannot lead to rank deficiency, linear independence only needs to be ensured if a row is added to $G_{\mathbb{A}}$, i.e. if a constraint is added to the working set. In the case that constraint j shall be added to the working set, Best [9] proposed the solution of the following

Copyright © 2007 John Wiley & Sons, Ltd.

DOI: 10.1002/rnc

auxiliary system as a test if G_j and $G_{\mathbb{A}}$ are linearly independent:

$$\begin{pmatrix} H & G_{\mathbb{A}}^{\mathrm{T}} \\ G_{\mathbb{A}} & 0 \end{pmatrix} \begin{pmatrix} s \\ \xi \end{pmatrix} = \begin{pmatrix} G_j \\ 0 \end{pmatrix} \tag{13}$$

It can be shown that $G_{\mathbb{A}\cup\{j\}}$ still has full-row rank if and only if $s \neq 0$. If $s = 0$, the coefficients ξ and the current dual vector $y_{\mathbb{A}}^*$ can be used to determine a currently active constraint, which must be removed before adding constraint j to the working set. If such a constraint does not exist infeasibility of the QP is detected, and we know that we are at the boundary of the set $\mathbb{P}$ of admissible initial values. In this case, our online active set strategy just stops the homotopy at the boundary of $\mathbb{P}$ and waits for the next QP which may be feasible again. However, a more elaborate infeasibility handling seems conceivable.

Also note that the computational effort to solve Equation (13) is much lower than a primal-dual step determination because of its special structure. However, in the case of degeneracy more computations than in a standard iteration are necessary to perform the additional change of the working set.

3.3. *Factorization of the KKT system*

For determination of the next primal and dual step direction the KKT system (9a) has to be solved before each homotopy step. Our implementation is based on the so-called *null space* approach for the solution of QPs, cf. [20]. Therein, an orthonormal basis Z of the null space spanned by the rows of $G_{\mathbb{A}}$ as well as a Cholesky decomposition of the *projected Hessian matrix* $Z^{\mathrm{T}}HZ$ is computed:

$$G_{\mathbb{A}} = (0 \;\; T) \begin{pmatrix} Z^{\mathrm{T}} \\ Y^{\mathrm{T}} \end{pmatrix} \tag{14a}$$

$$Z^{\mathrm{T}}HZ = R^{\mathrm{T}}R \tag{14b}$$

where $Z \in \mathbb{R}^{n\times(n-|\mathbb{A}|)}$, $Y \in \mathbb{R}^{n\times|\mathbb{A}|}$ are such that $(Z|Y)$ is orthonormal, $T \in \mathbb{R}^{|\mathbb{A}|\times|\mathbb{A}|}$ is a reverse lower triangular matrix and $R \in \mathbb{R}^{(n-|\mathbb{A}|)\times(n-|\mathbb{A}|)}$ is an upper triangular matrix. These decompositions of the matrices $G_{\mathbb{A}}$ and $Z^{\mathrm{T}}HZ$ are stored and updated efficiently after each change of the active set. Our implementation uses the update procedures for matrix factorizations as presented in [3] leading to a complexity of $\mathcal{O}(n^2)$ per active set change (provided that the number of constraints grows linearly in the number of variables). Note that the matrix decompositions are in particular also kept during the transition from one sampling time's QP to the next.

3.4. *Implementation details*

Instead of the general formulation (4), our online active set strategy was implemented for QPs of the following form:

$$\min_{w} \quad \tfrac{1}{2}w^{\mathrm{T}}Hw + w^{\mathrm{T}}g(x_0) \tag{15a}$$

$$\text{s.t.} \quad \underline{b_{\mathrm{C}}}(x_0) \leqslant Gw \leqslant \overline{b_{\mathrm{C}}}(x_0) \tag{15b}$$

$$\underline{b_{\mathrm{B}}}(x_0) \leqslant w \leqslant \overline{b_{\mathrm{B}}}(x_0) \tag{15c}$$

Copyright © 2007 John Wiley & Sons, Ltd.

DOI: 10.1002/rnc

where $G \in \mathbb{R}^{m \times n}$, $\underline{b_C}(x_0), \overline{b_C}(x_0) \in \mathbb{R}^m$ and $\underline{b_B}(x_0), \overline{b_B}(x_0) \in \mathbb{R}^n$ for all $x_0 \in \mathbb{P}$. The distinction between *constraints* and *bounds* seems adequate because bounds arise naturally in the context of MPC and special treatment of them can lead to substantial computational savings as described in [3].

In order to initialize the proposed online active set strategy the optimal solution pair (w^*, y^*) of the initial QP and a corresponding working set $\mathbb{A}$ must be available. So the question that naturally arises as how to obtain this information. One possibility would be to solve the initial QP by means of a standard active set QP solver. However, this would be rather inconvenient since all the efforts needed to implement and set up such a solver would be necessary just for the solution of the very first QP. Instead, our online active set strategy allows for an easy workaround: one simply has to set up a QP whose solution is known. A straightforward idea is to set the gradient to zero, all upper bounds to some positive constant value, say ϱ, and all lower bounds to $-\varrho$. Then $(0, 0)$ is obviously a primal–dual solution pair, the corresponding working set is empty and we can use our usual homotopy to go to the solution of the initial QP.

3.5. *Real-time variant*

One advantage of our online active set strategy is that it produces a *sequence of optimal solutions* for QPs on the homotopy path. Thus, 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 active set methods is necessary because every iterate is optimal and therefore feasible. Of course, if we interrupt the homotopy before the solution is reached we may stop at a point which is suboptimal and infeasible (as the constraint vector also depends on the homotopy parameter) with respect to the QP we want to solve.

In a real-time scenario one can try to find the optimal solution of the current QP within a given sampling time. However, if too many active set changes are necessary 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 during the last homotopy. This situation is illustrated in Figure 2, wherein only two active set changes are allowed per QP.

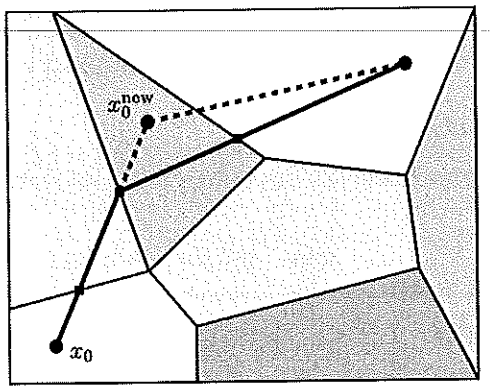

Figure 2. Homotopy paths (solid) from one QP to the next with limited number of active set changes.

Copyright © 2007 John Wiley & Sons, Ltd.

DOI: 10.1002/rnc

The computational effort per active set change is known rather exactly. So, if one obtains an estimate for the number of active set changes from one QP to the next, e.g. from closed-loop simulations, it is easy to estimate the possible sampling time length.

We believe that this real-time variant is a useful heuristic for practical set-ups, mainly for two reasons: first, the parameter vector x_0 is usually obtained by noisy measurements and in cases of strongly disturbed measurements it might not even be reasonable to solve the corresponding QP exactly. Instead, the true value of x_0 lies in a confidence region around its measured value and using the proposed real-time variant we aim at reaching this confidence region at least. Second, reducing the number of working set recalculations per QP solution also allows to decrease the sampling time; and the lower the sampling time the more likely it is that the active set does not change much (and thus few working set recalculations are required). However, if it is very crucial to guarantee optimality and constraint satisfaction at each sampling instant, a fully converged QP solution—possibly within a robust optimal control problem formulation—is more appropriate.

Note that our online active set strategy shares some features with a dual active set method [4]: both allow QP warm starting without phase I. When iterations are terminated prematurely, however, our method solves a QP that is exactly known to lie on the straight line between QP(x_0) and QP(x_0^{new}), while the dual active set method delivers in each iteration the solution to an unknown primal QP.

4. NUMERICAL TESTS

In this section, we compare the performance of our online active set strategy with that of a conventional primal active set QP solver, namely `qpsol` [21]. Since it is not intended to provide a detailed benchmarking we restrict ourselves to a single challenging test problem presented, and described in more detail, in a similar form in [22]: we aim at regulating a chain of nine masses connected by springs to a certain steady state. One end of the chain is fixed on a wall while the three velocity components of the other end are used as control input. Besides fixed lower and upper input bounds, state constraints are also included in the optimization problem in order to ensure that the chain does not hit a vertical wall close to the steady state (illustrated in Figure 3(d)). Furthermore, we use a prediction horizon of 16 s divided into 80 control intervals. The model equations are derived from linearization of the nonlinear ordinary differential equation (ODE) model (with 57 states) at the steady state. Deviation from the steady state, the velocities of all masses and the control action are penalized via the objective function. At each sampling time, this formulation leads to a medium scale, strictly convex QP with the following dimensions: 57 initial value parameters, 240 variables, 480 bounds, and 711 constraints.

We simulate in a closed-loop manner integrating the nonlinear ODE system in order to obtain the movements of the chain. Starting at the steady-state, a strong perturbation is exerted to the chain by moving the free end with a given constant velocity for 3 s. Then the MPC controller takes over and tries to return the chain into its original steady state while not hitting against the wall. (Note that during the initial perturbation phase the optimizer is already running but the calculated optimal control action is not given back to the chain.) This scenario is tested with five different methods: first, we solve every QP exactly using three alternative methods:

M1: `qpsol` with cold start, i.e. initialization with an empty working set and the origin as an initial guess for the solution;

Copyright © 2007 John Wiley & Sons, Ltd.

DOI: 10.1002/rnc

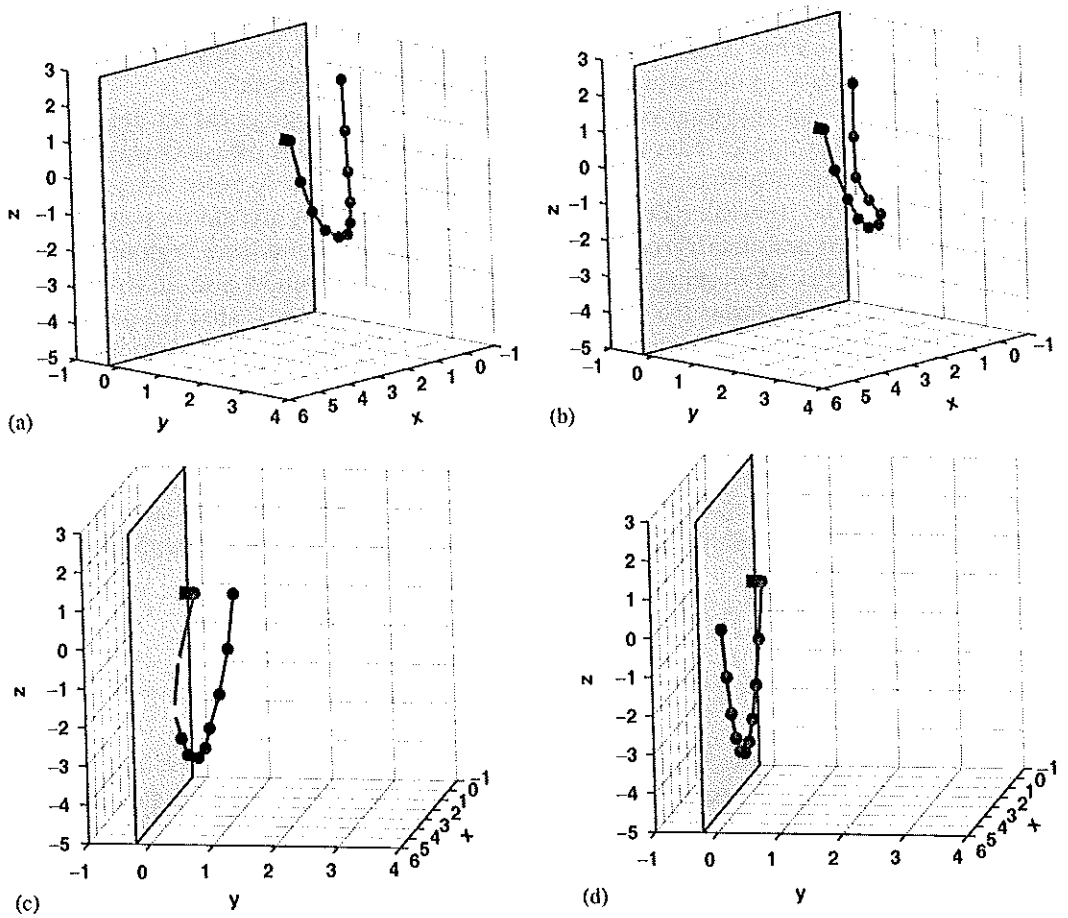

Figure 3. Optimally controlled closed-loop trajectory of the chain with exact QP solution: (a) at the beginning of control phase ($t = 3$ s); (b) early moment in control phase ($t = 4$ s); (c) softly touching the wall ($t = 5.6$ s); and (d) almost at steady state ($t = 15$ s).

M2: `qpsol` with warm start, i.e. the solver is initialized with the solution and corresponding working set of the previous QP (but without providing any matrix factorizations);
M3: Online active set strategy[‡] as presented in Section 3 where we follow every homotopy path until the exact solution is reached.

Second, we allow for an inexact QP solution by limiting the maximum number of active set changes (as described in Section 3.5) to

M4: 10 iterations;
M5: five iterations.

[‡]An open-source implementation of the online active set strategy is available at the qpOASES homepage [23].

Copyright © 2007 John Wiley & Sons, Ltd.

DOI: 10.1002/rnc

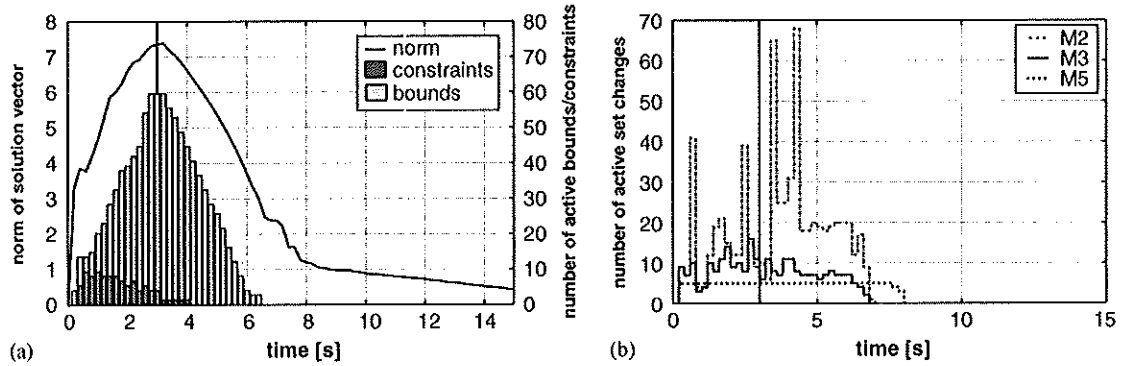

Figure 4. Closed-loop results for the chain example: (a) exact QP solution: Norm of solution vector w^* and number of active bounds/constraints and (b) number of iterations required by standard QP solver with warmstart (M2), and online active set strategies M3 and M5.

Table I. Comparison of standard QP solver and online active set strategies with respect to runtimes, maximum infeasibility (constraint violation), and least-squares performance.

Method	Maximum runtime (ms)	Average runtime (ms)	Maximum infeasibility	Performance
M1: `qpsol` (cold start)	864	237	0.00	1.0000
M2: `qpsol` (warm start)	924	177	0.00	1.0000
M3: online active set strategy (fully converged)	78	18	0.00	1.0000
M4: online active set strategy (at most 10 iterations)	48	17	0.00	1.0002
M5: online active set strategy (at most five iterations)	25	12	0.15	1.0078

The number of bounds and constraints active at the solution as well as the norm of the solution vector is depicted in Figure 4(a) for the case of exact QP solution; Figure 3 illustrates the optimally controlled chain at four particular time instants. Runtimes,[§] least-squares performance over the whole simulation horizon of 15 s and maximum infeasibilities are reported in Table I. Since all QPs remain feasible, infeasibilities can only occur using methods M4 and M5. Furthermore, they are restricted to constraint violations because all bounds are equally fixed for all sampling times and their fulfilment is thus not affected by the current position on the homotopy path. Figure 4(b) depicts the number of active set recalculations (corresponding to Phase II iterations of `qpsol`) for methods M2, M3 and M5.

Although the runtime of `qpsol` can be reduced using the warm start option it needs significantly more CPU time than the proposed online active set strategy. This is mainly due to three reasons: first, in most cases the provided initial guess (i.e. optimal solution of previous QP) is infeasible according to the shifted constraints of the new QP making a time-consuming Phase I calculation necessary; second, as shown in Figure 4(b), a much higher (up to six times) number of active set changes are required afterwards. The latter effect is also observed when using `qpsol` with cold start because many bounds and constraints have to be added to the initially empty

[§]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. The runtimes are obtained from multiple measurements with the linux-specific function `gettimeofday()` and should be accurate within some hundreds of microseconds.

Copyright © 2007 John Wiley & Sons, Ltd.

DOI: 10.1002/rnc

working set. In contrast, the proposed online active set strategy fully exploits the knowledge of the last solution and corresponding working set and performs only few active set changes towards the new solution. Finally, our online active set strategy can use both matrix factorizations from the previous QP, whereas `qpsol` has to calculate them from scratch even if an initial guess for the active set is provided via the warm start feature.

As said, the maximum runtime of the online active set strategy can be reduced further by limiting the number of allowed active set changes per sampling time. As this may lead to suboptimal or even infeasible solutions a careful choice of the iteration limit is required. Since significant infeasibility is not acceptable in this test case, the choice of at most five active set changes is too low although almost no suboptimality of the objective is observed (a possible crash against the wall was not modelled, of course). Instead, method M4 provided feasible and de facto optimal solutions throughout the whole simulation while reducing maximum runtime significantly. The average runtime does not differ much because methods M4 and M5 postpone some of the necessary active set changes to oncoming sampling times.

5. CONCLUSIONS AND FUTURE WORK

We have presented an online active set strategy for the fast solution of QP problems arising in MPC that builds on ideas from parametric optimization. This strategy fully exploits the knowledge of the solution of the previous QP and the reasonable assumption that the active set does not change much from one QP to the next. Furthermore, we showed how this strategy can be modified to make it suitable for real-time applications. Its performance was illustrated with a challenging, medium-scale test example including input as well as state bounds. In this example, our strategy turned out to be significantly faster than a standard active set QP solver (even when warm starts were used).

Future work will be focussed on several issues: we will incorporate some implementational refinements which allow, e.g. a faster computation of the maximum homotopy step length; afterwards, a more extensive benchmarking will show if our strategy is also superior to other QP solvers written with MPC applications in mind. Moreover, further theoretical extensions like infeasibility handling or treatment of not strictly convex QPs and linear programs are also conceivable. Moreover, a generalization of the proposed online active set strategy for usage within SQP algorithms (i.e. with varying QP matrices) will be implemented. This generalization is based on the following observation: having solved a QP, it is easy to construct a gradient and a constraint vector such that the previous optimal solution is also an optimal solution for an auxiliary QP with new matrices and these modified gradient and constraint vector. Therefore, one can start a usual homotopy from the solution of this auxiliary QP to that of the new QP (i.e. the one with new matrices *and* new gradient and constraint vectors).

Finally, we will further develop our open-source implementation `qpOASES` [23] and also set up a collection of benchmarking problems for online QP strategies [24].

ACKNOWLEDGEMENTS

We thank Leonard Wirsching for inspiring discussions about the formulation of the nonlinear chain MPC problem, as well as Adrian Wills, William Heath, Lorenz T. Biegler, and the anonymous reviewers for valuable comments that helped to improve the final manuscript.

Copyright © 2007 John Wiley & Sons, Ltd.

This research has partly been performed in the framework of the REGINS-PREDIMOT European project, whose financial support is gratefully acknowledged. The research has also been 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

1. Bemporad A, Morari M, Dua V, Pistikopoulos EN. The explicit linear quadratic regulator for constrained systems. *Automatica* 2002; **38**:3–20.
2. Gill PE, Gould NIM, Murray W, Saunders MA, Wright MH. A weighted Gram–Schmidt method for convex quadratic programming. *Mathematical Programming* 1984; **30**:176–195.
3. Gill PE, Murray W, Saunders MA, Wright MH. Procedures for optimization problems with a mixture of bounds and general linear constraints. *ACM Transactions on Mathematical Software* 1984; **10**(3):282–298.
4. Goldfarb D, Idnani A. A numerically stable dual method for solving strictly convex quadratic programs. *Mathematical Programming* 1983; **27**:1–33.
5. Bartlett RA, Biegler LT. QPSchur: a dual, active set, Schur complement method for large-scale and structured convex quadratic programming algorithm. *Optimization and Engineering* 2006; **7**:5–32.
6. Klee V, Minty GJ. How good is the simplex algorithm? In *Inequalities*, vol. III, Shisha O (ed.). Academic Press: New York, 1972; 159–175.
7. Wright SJ. *Primal–dual Interior-point Methods*. SIAM: Philadelphia, PA, 1997.
8. Rao CV, Wright SJ, Rawlings JB. Application of interior-point methods to model predictive control. *Journal of Optimization Theory and Applications* 1998; **99**:723–757.
9. Best MJ. An algorithm for the solution of the parametric quadratic programming problem. *Applied Mathematics and Parallel Computing*. Physica-Verlag: Heidelberg, 1996; 57–76.
10. Ferreau HJ, Bock HG, Diehl M. An online active set strategy for fast parametric quadratic programming in MPC applications. *Proceedings of the IFAC Workshop on Nonlinear Model Predictive Control for Fast Systems*, Grenoble, 2006.
11. Bock HG, Plitt KJ. A multiple shooting algorithm for direct solution of optimal control problems. *Proceedings of the 9th IFAC World Congress*, Budapest. Pergamon Press: Oxford, 1984; 243–247.
12. Boyd S, Vandenberghe L. *Convex Optimization*. University Press: Cambridge, 2004.
13. Zafiriou E. Robust model predictive control of processes with hard constraints. *Computers & Chemical Engineering* 1990; **14**(4–5):359–371.
14. Borrelli F, Bemporad A, Morari M. A geometric algorithm for multi-parametric linear programming. *Journal of Optimization Theory and Applications* 2003; **118**(3):515–540.
15. Tøndel P, Johansen TA, Bemporad A. An algorithm for multi-parametric quadratic programming and explicit mpc solutions. *Automatica* 2003; **39**:489–497.
16. Tøndel P, Johansen TA, Bemporad A. Computation and approximation of piecewise affine control laws via binary search trees. *Automatica* 2003; **39**:945–950.
17. Pannocchia G, Rawlings JB, Wright SJ. Fast, large-scale model predictive control by partial enumeration. *Automatica* 2007; **43**:852–860.
18. Wills AG, Bates D, Fleming AJ, Ninness B, Moheimani SOR. Application of MPC to an active structure using sampling rates up to 25 kHz. *Forty-fourth IEEE Conference on Decision and Control and European Control Conference ECC'05*, Seville, 2005.
19. Fiacco AV. *Introduction to Sensitivity and Stability Analysis in Nonlinear Programming*. Academic Press: New York, 1983.
20. Gill PE, Murray W. Numerically stable methods for quadratic programming. *Mathematical Programming* 1978; **14**:349–372.
21. NAG. *Fortran Library Routine Document E04NAF*, 1991.
22. Wirsching L. An SQP algorithm with inexact derivatives for a direct multiple shooting method for optimal control problems. *Master's Thesis*, University of Heidelberg, 2006.
23. qpOASES Homepage. http://homes.esat.kuleuven.be/~optec/software/qpOASES/
24. Online QP Benchmark Collection. http://nmpcgroup.uni-hd.de/onlineQP

Copyright © 2007 John Wiley & Sons, Ltd.

DOI: 10.1002/rnc