A Mixed Integer Dual Quadratic Programming Algorithm Tailored for MPC

Daniel Axehill, Anders Hansson

Division of Automatic Control

E-mail: daniel@isy.liu.se, hansson@isy.liu.se

3rd January 2007

Report no.: [LiTH-ISY-R-2761](#)

Accepted for publication in Proceedings the 45th IEEE Conference on Decision and Control

Address:

Department of Electrical Engineering

Linköpings universitet

SE-581 83 Linköping, Sweden

WWW: <http://www.control.isy.liu.se>

AUTOMATIC CONTROL
REGLERTEKNIK
LINKÖPINGS UNIVERSITET

Technical reports from the Automatic Control group in Linköping are available from

<http://www.control.isy.liu.se/publications>.

Abstract

The objective of this work is to derive an MIQP solver tailored for MPC. The MIQP solver is built on the branch and bound method, where QP relaxations of the original problem are solved in the nodes of a binary search tree. The difference between the subproblems is often small and therefore it is interesting to be able to use a previous solution as a starting point in a new subproblem. This is referred to as a warm start of the solver. Because of its good warm start properties, a dual active set QP method was chosen. The method is tailored for MPC by solving a part of the KKT system using a Riccati recursion, which makes the computational complexity of the QP iterations grow linearly with the prediction horizon. Simulation results are presented both for the QP solver itself and when it is incorporated as a part of the MIQP solver. In both cases the computational complexity is significantly reduced compared to if a primal active set solver not utilizing structure is used.

Keywords: Predictive control, Hybrid systems, Integer programming, Quadratic programming

A Mixed Integer Dual Quadratic Programming Algorithm Tailored for MPC[†]

Daniel Axehill and Anders Hansson
Linköpings universitet
SE-581 83 Linköping, Sweden
{daniel,hansson}@isy.liu.se

Abstract—The objective of this work is to derive an MIQP solver tailored for MPC. The MIQP solver is built on the branch and bound method, where QP relaxations of the original problem are solved in the nodes of a binary search tree. The difference between the subproblems is often small and therefore it is interesting to be able to use a previous solution as a starting point in a new subproblem. This is referred to as a warm start of the solver. Because of its good warm start properties, a dual active set QP method was chosen. The method is tailored for MPC by solving a part of the KKT system using a Riccati recursion, which makes the computational complexity of the QP iterations grow linearly with the prediction horizon. Simulation results are presented both for the QP solver itself and when it is incorporated as a part of the MIQP solver. In both cases the computational complexity is significantly reduced compared to if a primal active set solver not utilizing structure is used.

I. INTRODUCTION

From the beginning, Model Predictive Control (MPC) was only applicable to linearly constrained linear systems. Nowadays, MPC is applicable to nonlinear systems, and specifically to hybrid systems in Mixed Logical Dynamical (MLD) form, [1]. In the basic linear setup, the MPC problem can be cast in the form of a Quadratic Programming (QP) problem. When MPC is extended to more advanced systems, also the type of optimization problem to solve in each sample is changed. When a hybrid system is to be controlled, the corresponding optimization problem is changed from a QP problem to a Mixed Integer Quadratic Programming (MIQP) problem, hence the term Mixed Integer Predictive Control (MIPC) is sometimes used. Today there exist tailored optimization routines with the required performance for linear MPC. However, there is still a need for efficient optimization routines for MIPC. The MIQP problem is in general known to be $\mathcal{NP}$ -hard, [2]. Therefore, to be able to use the algorithm in real-time, it is desirable to reduce the computational complexity. One way of doing that is by utilizing problem structure when solving the optimization problem.

A popular method for solving MIQP problems is branch and bound, where the original integer optimization problem is solved as a sequence of smaller QP subproblems. The subproblems are ordered in a tree structure, where one new integer variable is fixed at each level. Depending on the problem, sometimes a large number of QP subproblems have to be solved. Therefore it is important to be able to efficiently solve these subproblems. In this paper, the efficiency of

this process is enhanced in two ways. First, because the difference between different subproblems in the tree is small, the solution from a previously solved subproblem is reused as a starting point in a new subproblem. This procedure is often called a warm start of the solver, and is most easily and effectively implemented if a dual active set QP solver is used, [3]. Early work on dual active set solvers can be found in [4] and [5]. A more recent method is found in [6], which has been refined in [7], [8] and [9]. Second, in this dual QP solver a large part of the Karush-Kuhn-Tucker system is solved using a Riccati recursion. This has previously been used in primal active set QP solvers, *e.g.*, [10], and in interior point solvers, *e.g.*, [11]. A more thorough description of the material presented in this paper can be found in [3].

For what follows, two notations will be introduced. First, a set with integers on an interval, $\mathbb{Z}_{i,j} = \{i, i+1, \dots, j\}$, and second, $\mathbb{S}_{++}^n$ ($\mathbb{S}_+^n$) that denotes the set of symmetric positive (semi) definite matrices with n rows.

II. PROBLEM DEFINITION

When the MPC problem is cast in the form of a QP problem, it can either be written as a QP problem with only control signals as free variables or it can be written as a QP problem where control signals, states and control errors all are free variables. In this paper, it will be shown that the latter approach will result in a KKT system possible to solve using a Riccati recursion, and it will be seen that this is advantageous from a computational point of view.

Consider a linear system in state space form

$$\begin{aligned} x(t+1) &= A(t)x(t) + B(t)u(t) \\ y(t) &= C(t)x(t) \end{aligned} \quad (1)$$

and the resulting MPC optimization problem

$$\begin{aligned} \underset{x, u, e}{\text{minimize}} \quad & \frac{1}{2} \sum_{t=0}^{N-1} e^T(t) Q_e(t) e(t) + u^T(t) Q_u(t) u(t) \\ & + \frac{1}{2} e^T(N) Q_e(N) e(N) \end{aligned}$$

$$\begin{aligned} \text{subject to} \quad & x(0) = x_0 \\ & x(t+1) = A(t)x(t) + B(t)u(t), \quad t \in \mathbb{Z}_{0, N-1} \\ & e(t) = M(t)x(t), \quad t \in \mathbb{Z}_{0, N} \\ & h(0) + H_u(0)u(0) \leq 0 \\ & h(t) + H_x(t)x(t) + H_u(t)u(t) \leq 0, \quad t \in \mathbb{Z}_{1, N-1} \end{aligned} \quad (2)$$

[†] Research supported by the Swedish Research Council for Engineering Sciences under contract Nr. 621-2002-3822 and by the VINNOVA competence centre ISIS.

where x , u and e denote vectors in which the states, control signals and control errors respectively from all time instants are stacked. Furthermore, $A(t) \in \mathbb{R}^{n \times n}$, $B(t) \in \mathbb{R}^{n \times m}$, $M(t) \in \mathbb{R}^{p \times n}$, $H_x(t) \in \mathbb{R}^{c(t) \times n}$, $H_u(t) \in \mathbb{R}^{c(t) \times m}$ and $h(t) \in \mathbb{R}^{c(t)}$ where $c(t)$ denotes the number of inequality constraints at time t . Also, the following assumptions are made

- Assumption 1:* $Q_e(t) \in \mathbb{S}_{++}^p$, $t \in \mathbb{Z}_{0,N}$
Assumption 2: $Q_u(t) \in \mathbb{S}_{++}^m$, $t \in \mathbb{Z}_{0,N-1}$

III. OPTIMIZATION PRELIMINARIES

In this paper, an optimization problem such as

$$\begin{aligned} & \underset{x}{\text{minimize}} \quad f_0(x) \\ & \text{subject to} \quad f_i(x) \leq 0, \quad i = 1, \dots, m \\ & \quad \quad \quad h_i(x) = 0, \quad i = 1, \dots, p \end{aligned} \quad (3)$$

where $f_0 : \mathbb{R}^n \rightarrow \mathbb{R}$, $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$ and $h_i : \mathbb{R}^n \rightarrow \mathbb{R}$ is said to be in standard form. The optimal objective function value of (3) is denoted p^* . Associated with each optimization problem there is a Lagrange dual problem in the form

$$\begin{aligned} & \underset{\lambda, \nu}{\text{maximize}} \quad g(\lambda, \nu) \\ & \text{subject to} \quad \lambda \geq 0 \end{aligned} \quad (4)$$

where $g(\lambda, \nu)$ is called the Lagrange dual function. Duality is thoroughly discussed in, e.g., [12]. An important property of the Lagrange dual function is that for any $\lambda \geq 0$, the following inequality holds

$$g(\lambda, \nu) \leq p^* \quad (5)$$

The optimal dual objective function value is denoted d^* . From (5) the following inequality can be derived

$$d^* \leq p^* \quad (6)$$

This inequality is called weak duality. Weak duality holds even if the primal problem is non-convex and it still holds if d^* or p^* are infinite. For some problems, inequality (6) can be shown to hold with equality, i.e.,

$$d^* = p^* \quad (7)$$

This important property is called strong duality.

In this paper the KKT conditions are used as optimality conditions. They are stated in the following theorem, which is based on the discussion in [12, pp. 243–244].

Theorem 1 (KKT): Consider the optimization problem (3). Assume that it is convex, that $f_i(x)$, $i = 0, \dots, m$ are differentiable and that strong duality holds. Then the following so-called Karush-Kuhn-Tucker (KKT) conditions are necessary and sufficient conditions for x^* and (λ^*, ν^*) to be primal respectively dual optimal points

$$f_i(x^*) \leq 0, \quad i = 1, \dots, m \quad (8a)$$

$$h_i(x^*) = 0, \quad i = 1, \dots, p \quad (8b)$$

$$\lambda_i^* \geq 0, \quad i = 1, \dots, m \quad (8c)$$

$$\lambda_i^* f_i(x^*) = 0, \quad i = 1, \dots, m \quad (8d)$$

$$\nabla f_0(x^*) + \sum_{i=1}^m \lambda_i^* \nabla f_i(x^*) + \sum_{i=1}^p \nu_i^* \nabla h_i(x^*) = 0 \quad (8e)$$

Proof: See [12, p. 244]. ■

IV. QUADRATIC PROGRAMMING

In this paper the following special case of a general QP will be of interest

$$\begin{aligned} & \underset{x_1, x_2}{\text{minimize}} \quad \frac{1}{2} \begin{bmatrix} x_1^T & x_2^T \end{bmatrix} \underbrace{\begin{bmatrix} \tilde{H} & 0 \\ 0 & 0 \end{bmatrix}}_{\triangleq H} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} \tilde{f}^T & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \\ & \text{subject to} \quad \begin{bmatrix} A_{\mathcal{E},1} & A_{\mathcal{E},2} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = b_{\mathcal{E}} \\ & \quad \quad \quad \begin{bmatrix} A_{\mathcal{I},1} & A_{\mathcal{I},2} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \leq b_{\mathcal{I}} \end{aligned} \quad (9)$$

where $x_1 \in \mathbb{R}^{n_1}$, $x_2 \in \mathbb{R}^{n_2}$, $n = n_1 + n_2$, $\tilde{H} \in \mathbb{S}_{++}^{n_1}$, $\tilde{f} \in \mathbb{R}^{n_1}$. This QP problem differs from what is normally considered in the literature in the context of QP duality, where the Hessian often is assumed positive definite. The dual problem is found by forming the Lagrange dual function and performing maximization over λ and ν . This is thoroughly described in [3]. A problem equivalent to the dual problem of (9) can then be found to be

$$\begin{aligned} & \underset{\lambda, \nu}{\text{minimize}} \quad \frac{1}{2} \begin{bmatrix} \lambda^T & \nu^T \end{bmatrix} \begin{bmatrix} A_{\mathcal{I},1} \\ A_{\mathcal{E},1} \end{bmatrix} \tilde{H}^{-1} \begin{bmatrix} A_{\mathcal{I},1}^T & A_{\mathcal{E},1}^T \end{bmatrix} \begin{bmatrix} \lambda \\ \nu \end{bmatrix} + \\ & \quad \quad \quad + \left(\tilde{f}^T \tilde{H}^{-1} \begin{bmatrix} A_{\mathcal{I},1}^T & A_{\mathcal{E},1}^T \end{bmatrix} + \begin{bmatrix} b_{\mathcal{I}}^T & b_{\mathcal{E}}^T \end{bmatrix} \right) \begin{bmatrix} \lambda \\ \nu \end{bmatrix} \\ & \text{subject to} \quad A_{\mathcal{I},2}^T \lambda + A_{\mathcal{E},2}^T \nu = 0 \\ & \quad \quad \quad \lambda \geq 0 \end{aligned} \quad (10)$$

By saying that (10) is an equivalent problem to the dual problem of (9) it is meant that given the solution to one of the equivalent problems the other one can easily be derived from the first one. According to [3], some important strong duality results can be stated for this problem. First, if the primal problem is feasible, then the primal and dual optimal objective function values coincide. Second, the primal is feasible if and only if the dual optimal objective function value is bounded from above. At large, these results also hold for the problem (10) equivalent to the dual, but some extra care should be taken since the sign of the objective function has been changed and the objective function value has in general been off-set. For an extensive bibliography on QP, see [13].

A. Active Set Methods

A QP can either be solved by an active set method or an Interior Point (IP) method. Compared to active set methods, IP methods are often preferred for large problems since their computational performance is less sensitive to the number of constraints, [14]. On the other hand, active set methods can gain more from warm starts [15], which makes them preferable for solving a sequence of slightly modified problems. Because of their good warm start properties, an active set method was chosen in this paper. For a thorough description of an active set method applicable to QP, see [16]. Briefly, an active set method solves an inequality constrained optimization problem by solving a sequence of equality constrained optimization problems. The set containing the equality constraints is called the working set, and it contains

all original equality constraints and usually some of the original inequality constraints treated as equality constraints. Solving an equality constrained problem involves the solution of a KKT system in the form

$$\begin{bmatrix} H & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} x \\ \nu \end{bmatrix} = \begin{bmatrix} -f \\ b \end{bmatrix} \quad (11)$$

Since one such system is solved in each QP iteration, it is very important to be able to solve systems of this type efficiently. For the MPC application, this has previously been done for primal active set methods in [10]. In this paper it will be shown how this can be performed for a dual active set algorithm as well.

A primal active set method starts in a primal feasible point and primal feasibility is maintained during all subsequent QP iterations. In a dual active set method, dual feasibility is instead maintained. One of the main advantages with a dual QP algorithm is that it is in many cases much easier to find an initial dual feasible point than an initial primal feasible point. Usually, an initial feasible point is found by a so-called Phase I algorithm. According to [6], on average between one-third to one-half of the total effort needed to solve a QP with “typical primal algorithms” is spent in Phase I. Consider the dual problem (10). It is easy to see that the origin, *i.e.*, $\lambda = 0$ and $\nu = 0$, is always a feasible point both with respect to the equality and the inequality constraints. Hence, an initial dual feasible point can easily be found.

The solution time for an active set algorithm is much dependent on how many QP iterations that have to be performed. If the optimal active set was known in advance, the process would terminate in a single QP iteration. This is the idea behind warm starts, where a slightly modified problem can be solved efficiently by utilizing information from a previously solved problem. Due to the simple structure of the inequality constraints in the dual problem, a dual active set QP algorithm is very easy to warm start. This property makes it suitable for, *e.g.*, Sequential Quadratic Programming (SQP), where several similar inequality constrained QPs are solved sequentially, [7].

The structure of the dual problem also makes it possible to use the gradient projection method efficiently, which allows for rapid changes to the working set, [16]. If this method is employed, the number of QP iterations can be reduced.

V. MIXED INTEGER QUADRATIC PROGRAMMING

An MIQP problem is similar to the ordinary QP problem in (9), but the optimization is to be performed for $x \in \mathbb{R}^{n_c} \times \{0, 1\}^{n_b}$. This means that the optimization variables are not only allowed to be real valued, but also integer valued. This “slight” modification turns the easily solved QP problem, into an $\mathcal{NP}$ -hard problem, [2]. In this paper, a common special case of MIQP, *i.e.*, when the integer variables are constrained to be 0 or 1, is considered.

There exist several methods for solving MIQP problems. Among them several authors consider the branch and bound method as the best choice for MIQP problems, [1], [17]. An integer program can in principle be solved by explicit enumeration, where the objective function value is evaluated for all integer combinations and the best combinations are

chosen as the optimizers. Branch and bound cuts down the number of combinations necessary to evaluate. A thorough description of the branch and bound algorithm can be found in, *e.g.*, [2] and [18].

In branch and bound for MIQP, QP subproblems are solved in the tree. According to [17], solving the subproblems using a dual active set method offers the most straightforward way to exploit the structure introduced in the branching procedure. According to [2], active set methods (the reference considers the linear programming case) is preferable for solving the relaxed problems in branch and bound. However, for very large problems, IP algorithms can be used to solve the first subproblem, but in the subsequent subproblems an active set method should be used. After a branch, the solution to the parent primal problem is in general infeasible in the child primal problems. But, as later will become apparent, a dual feasible starting point for the child problems is directly available from the dual solution of the parent problem. Consequently, it is possible to warm start a dual active set solver using information from the solution of the parent problem. Also, since a dual active set method is an ascend method generating dual feasible points, it can use an upper bound as a cut-off value for terminating the QP solver in the subproblems prematurely, [17].

An important step in an MIQP solver is the preprocessing step, where the problem is reduced if possible and the formulation is made as strong as possible. A preprocessing algorithm for unconstrained MIPC is presented in [19].

VI. A DUAL QUADRATIC PROGRAMMING ALGORITHM

The idea in this paper is to solve a primal problem in the form in (9) by solving the associated Lagrange dual problem in the form in (10). In the first part of the derivation the following assumption is made

Assumption 3: $H_u(t)$ has full row rank for $t \in \mathbb{Z}_{0,N-1}$. Following the general procedure in Section III, a problem equivalent to the dual problem can be found to be

$$\begin{aligned} \underset{\tilde{x}, \tilde{u}}{\text{minimize}} \quad & \frac{1}{2} \tilde{u}^T(-1) \tilde{Q}_{\tilde{u}}(-1) \tilde{u}(-1) \\ & + \frac{1}{2} \sum_{\tau=0}^{N-1} (\tilde{x}^T(\tau) \tilde{Q}_{\tilde{x}}(\tau) \tilde{x}(\tau) + \tilde{u}^T(\tau) \tilde{Q}_{\tilde{u}}(\tau) \tilde{u}(\tau) \\ & + 2\tilde{x}^T(\tau) \tilde{Q}_{\tilde{x}\tilde{u}}(\tau) \tilde{u}(\tau) + 2\tilde{q}_{\tilde{u}}^T(\tau) \tilde{u}(\tau)) + \tilde{q}_{\tilde{x}}^T(N) \tilde{x}(N) \\ \text{subject to} \quad & \tilde{x}(0) = \tilde{B}(-1) \tilde{u}(-1) \\ & \tilde{x}(\tau+1) = \tilde{A}(\tau) \tilde{x}(\tau) + \tilde{B}(\tau) \tilde{u}(\tau), \quad \tau \in \mathbb{Z}_{0,N-1} \\ & [0 \quad -I_{c(N-\tau-1)}] \tilde{u}(\tau) \leq 0, \quad \tau \in \mathbb{Z}_{0,N-1} \end{aligned} \quad (12)$$

where the parameters and variables are defined as in [3, p. 75]. More on duality in control problems can be found in, *e.g.*, [20] and [21].

If the dual problem is feasible it follows from Slater’s refined condition (see [3, p. 14]) that strong duality holds. Then the KKT conditions in Theorem 1 constitutes the following necessary and sufficient conditions for optimality

$$[0 \quad -I_{c(N-\tau-1)}] \tilde{u}(\tau) \leq 0, \quad \tau \in \mathbb{Z}_{0,N-1} \quad (13a)$$

$$\tilde{x}(0) = \tilde{B}(-1) \tilde{u}(-1) \quad (13b)$$

$$\tilde{x}(\tau+1) = \tilde{A}(\tau) \tilde{x}(\tau) + \tilde{B}(\tau) \tilde{u}(\tau), \quad \tau \in \mathbb{Z}_{0,N-1} \quad (13c)$$

From the dual feasibility condition (8c), it holds that

$$\mu(\tau) \geq 0, \quad \tau \in \mathbb{Z}_{0,N-1} \quad (14)$$

The complementary slackness condition (8d) gives the equation

$$\begin{aligned} \mu_i(\tau) \tilde{u}_{\tilde{m}(\tau)-c(N-\tau-1)+i}(\tau) &= 0, \\ i \in \mathbb{Z}_{1,c(N-\tau-1)}, \tau \in \mathbb{Z}_{0,N-1} \end{aligned} \quad (15)$$

Finally, from (8e) the following equations are obtained

$$\begin{aligned} \frac{\partial L_D}{\partial \tilde{x}(\tau)} &= \tilde{Q}_{\tilde{x}}(\tau) \tilde{x}(\tau) + \tilde{Q}_{\tilde{x}\tilde{u}}(\tau) \tilde{u}(\tau) + \tilde{A}^T(\tau) \lambda(\tau+1) \\ &\quad - \lambda(\tau) = 0, \tau \in \mathbb{Z}_{0,N-1} \end{aligned} \quad (16a)$$

$$\frac{\partial L_D}{\partial \tilde{x}(N)} = \tilde{q}_{\tilde{x}}(N) - \lambda(N) = 0 \quad (16b)$$

$$\begin{aligned} \frac{\partial L_D}{\partial \tilde{u}(\tau)} &= \tilde{Q}_{\tilde{u}}(\tau) \tilde{u}(\tau) + \tilde{Q}_{\tilde{x}\tilde{u}}^T(\tau) \tilde{x}(\tau) + \tilde{q}_{\tilde{u}}(\tau) \\ &\quad + \tilde{B}^T(\tau) \lambda(\tau+1) - \begin{bmatrix} 0 \\ I \end{bmatrix} \mu(\tau) = 0, \\ \tau \in \mathbb{Z}_{0,N-1} \end{aligned} \quad (16c)$$

$$\frac{\partial L_D}{\partial \tilde{u}(-1)} = \tilde{Q}_{\tilde{u}}(-1) \tilde{u}(-1) + \tilde{B}^T(-1) \lambda(0) = 0 \quad (16d)$$

Using a general notation, for a specific QP iteration the KKT conditions for the dual problem, excluding the non-linear complementary slackness condition (15), is given by a linear system in the form

$$\begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix} \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \end{bmatrix} = \begin{bmatrix} H & A_{\mathcal{E}}^T & A_{\mathcal{I} \cap \mathcal{W}}^T \\ A_{\mathcal{E}} & 0 & 0 \\ A_{\mathcal{I} \cap \mathcal{W}} & 0 & 0 \end{bmatrix} \begin{bmatrix} \hat{x} \\ \hat{v} \\ \hat{\lambda}_{\mathcal{I} \cap \mathcal{W}} \end{bmatrix} = \begin{bmatrix} -f \\ b_{\mathcal{E}} \\ b_{\mathcal{I} \cap \mathcal{W}} \end{bmatrix} \quad (17)$$

where $A_{\mathcal{I} \cap \mathcal{W}}$ contains the rows in $A_{\mathcal{I}}$ corresponding to the inequality constraints contained in the working set and $b_{\mathcal{I} \cap \mathcal{W}}$ is defined analogously. Furthermore, $\hat{\lambda}_{\mathcal{I} \cap \mathcal{W}}$ contains the dual variables corresponding to constraints in the working set. The matrix $A_{\mathcal{E}}$ and the vector $b_{\mathcal{E}}$ define the equality constraints in the dual problem. Dual variables corresponding to inequality constraints not in the working set, that is $\{\lambda_i \mid i \notin \mathcal{W}\}$, are set to zero by the active set algorithm. Since the upper left block in (17) is unchanged during the QP iterations it first seems possible to proceed as in [10], where it is proposed to use block elimination and Riccati recursions in order to solve (17). Trying to follow the same approach in the dual might cause problems. By simple examples, it can be shown that K_{11} in the dual problem cannot, in general, be expected to be nonsingular, and therefore, block elimination cannot be used similarly as in [10] when solving the dual problem.

Since it is generally not possible to compute K_{11}^{-1} , an alternative efficient algorithm has to be developed. For what follows, it is necessary to split $\tilde{u}$ into w and v , where w contains the stacked unconstrained dual control signals and v contains the stacked constrained dual control signals. Relating back to (12), let $\bar{x}_1 = [\bar{x}^T \ w^T \ \lambda^T \ v_{\mathcal{W}_v}^T]^T$ and $\bar{x}_2 = [v_{\mathcal{W}_v}^T \ \mu_{\mathcal{W}_v}^T]^T$, where the subindex indicates whether the corresponding variable contains elements related to the constraints in the working set or not. The result after reordering rows and columns in K is a system in the form

$$\begin{bmatrix} \bar{K}_{11} & \bar{K}_{12} \\ \bar{K}_{21} & \bar{K}_{22} \end{bmatrix} \begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \end{bmatrix} = \begin{bmatrix} \bar{R}_{11} & \bar{R}_{12} & 0 \\ \bar{R}_{21} & \bar{R}_{22} & -I \\ 0 & -I & 0 \end{bmatrix} \begin{bmatrix} \bar{x}_1 \\ \bar{x}_{2,1} \\ \bar{x}_{2,2} \end{bmatrix} = \begin{bmatrix} \bar{b}_1 \\ \bar{b}_{2,1} \\ \bar{b}_{2,2} \end{bmatrix} \quad (18)$$

Note that $\bar{K}_{22}$ is non-singular. In (18), it has been used that the inequality constraints are in the form $-v(\tau) \leq 0$, which implies that the matrix containing the coefficient for $v_{\mathcal{W}_v}$ is $-I$. Dual variables corresponding to constraints not in the working set, $\mu_{\mathcal{W}_v^c}$, are set to zero. Also, note that $\bar{b}_{2,2} = 0$. This follows from the fact that the right hand side of the inequality in (13a) is zero. By using a block inversion formula for the case $\bar{K}_{22}$ is non-singular, $\bar{x}_1$ and $\bar{x}_2$ can be calculated as

$$\begin{aligned} \bar{x}_1 &= (\bar{K}_{11} - \bar{K}_{12} \bar{K}_{22}^{-1} \bar{K}_{21})^{-1} (\bar{b}_1 - \bar{K}_{12} \bar{K}_{22}^{-1} \bar{b}_2) \\ \bar{x}_2 &= \bar{K}_{22}^{-1} (\bar{b}_2 - \bar{K}_{21} \bar{x}_1) \end{aligned} \quad (19)$$

Notice that

$$\bar{K}_{22}^{-1} = \begin{bmatrix} 0 & -I \\ -I & -\bar{R}_{22} \end{bmatrix} \quad (20)$$

Furthermore,

$$\bar{K}_{12} \bar{K}_{22}^{-1} = [\bar{R}_{12} \ 0] \begin{bmatrix} 0 & -I \\ -I & -\bar{R}_{22} \end{bmatrix} = [0 \ -\bar{R}_{12}] \quad (21)$$

From this it immediately follows that

$$\bar{K}_{12} \bar{K}_{22}^{-1} \bar{K}_{21} = [0 \ -\bar{R}_{12}] \begin{bmatrix} \bar{R}_{21} \\ 0 \end{bmatrix} = 0 \quad (22)$$

Using (21) and (22) in the expression for $\bar{x}_1$ in (19), the following simplified equation for $\bar{x}_1$ is obtained

$$\bar{R}_{11} \bar{x}_1 = \bar{b}_1 + \bar{R}_{12} \bar{b}_{2,2} = \bar{b}_1 \quad (23)$$

where the last equality follows from $\bar{b}_{2,2} = 0$. By using (20) and that $\bar{b}_{2,2} = 0$, the expression for $\bar{x}_2$ can be simplified to

$$\bar{x}_2 = \begin{bmatrix} 0 \\ \bar{R}_{21} \bar{x}_1 - \bar{b}_{2,1} \end{bmatrix} \quad (24)$$

where $\bar{R}_{21}$ is block diagonal. As a consequence, when $\bar{x}_1$ has been computed, $v_{\mathcal{W}_v}$ and $\mu_{\mathcal{W}_v}$ can easily be computed as

$$v_{\mathcal{W}_v} = 0, \mu_{\mathcal{W}_v} = \bar{R}_{21} \bar{x}_1 - \bar{b}_{2,1} \quad (25)$$

Left to solve is the equation $\bar{R}_{11} \bar{x}_1 = \bar{b}_1$. This equation corresponds to solving a modified version of the equations in (13) and (16). From (18), it follows that (16) should be modified by setting components in $v_{\mathcal{W}_v}$ and $\mu_{\mathcal{W}_v^c}$ to zero. After dropping time indices and a suitable reordering of the variables and equations, (23) can be written as

$$\begin{bmatrix} \bar{Q}_{\tilde{u}} \bar{B}^T & 0 & 0 & 0 & 0 & 0 & \dots & \dots & \dots & 0 \\ \bar{B}^T & 0 & -I & 0 & 0 & 0 & \dots & \dots & \dots & \vdots \\ 0 & -I & \bar{Q}_{\tilde{x}} & \bar{Q}_{\tilde{x}\tilde{u}} & \bar{A}^T & 0 & 0 & \dots & \dots & \vdots \\ 0 & 0 & \bar{Q}_{\tilde{x}\tilde{u}}^T & \bar{Q}_{\tilde{u}} & \bar{B}^T & 0 & 0 & \dots & \dots & \vdots \\ 0 & 0 & \bar{A} & \bar{B} & 0 & -I & 0 & \dots & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & \dots & \dots & \dots & \dots & \dots & \bar{A} & \bar{B} & 0 & -I \end{bmatrix} \bar{x}'_1 = \bar{b}'_1 \quad (26)$$

where $\bar{x}'_1$ and $\bar{b}'_1$ are defined as

$$\bar{x}'_1 = \begin{bmatrix} \bar{u}(-1) \\ \lambda(0) \\ \bar{x}(0) \\ \bar{u}(0) \\ \vdots \\ \bar{x}(N-1) \\ \bar{u}(N-1) \\ \lambda(N) \\ \bar{x}(N) \end{bmatrix}, \quad \bar{b}'_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -\bar{q}_{\bar{u}}(0) \\ 0 \\ \vdots \\ -\bar{q}_{\bar{u}}(N-1) \\ -\bar{q}_{\bar{x}}(N) \end{bmatrix} \quad (27)$$

where the system in (26) includes all components of $w(\tau)$ but only the components in $v(\tau)$ not included in a constraint in the working set since these are directly set to zero.

Under the assumption $\tilde{Q}_u(\tau) \in \mathbb{S}_{++}^{\tilde{m}(\tau)}$, a system of equations in this form can be solved using Riccati recursions. This is shown in detail in, e.g., [3], [22] and [23]. Since all computations necessary to solve the KKT system, including the Riccati recursion [22], [23], can be performed with linear computational complexity in the time horizon N , the entire operation can be concluded to have linear computational complexity in N .

As already concluded in Section IV-A, the origin can always be used as a feasible initial point. Hence, the need for a Phase I algorithm has been eliminated. Also, it can be realized that the origin is always feasible independently of the working set chosen, i.e., reuse of old working sets during warm starts can be easily performed.

Unfortunately, Assumption 3 makes it impossible to have upper and lower bound constraints on the optimization variables since then the constraint gradients become linearly dependent. To be able to use the QP algorithm in a branch and bound framework where binary constraints are relaxed to interval constraints, Assumption 3 has to be relaxed. Consider constraints in the form

$$\begin{bmatrix} \bar{H}_x(t) \\ \hat{H}_x(t) \\ -\hat{H}_x(t) \end{bmatrix} x(t) + \begin{bmatrix} \bar{H}_u(t) \\ \hat{H}_u(t) \\ -\hat{H}_u(t) \end{bmatrix} u(t) \leq \begin{bmatrix} \bar{h}(t) \\ \hat{h}_+(t) \\ -\hat{h}_-(t) \end{bmatrix} \quad (28)$$

where the top block contains general constraints, and the second and third block together contain upper and lower bound constraints defining a set with strictly feasible points, i.e., $\hat{h}_+(t) > \hat{h}_-(t)$. A relaxed assumption can be formulated as

Assumption 4: $[\bar{H}_u^T(t) \ \hat{H}_u^T(t)]^T$ in (28) has full row rank and $\hat{h}_+(t) > \hat{h}_-(t)$ for all $t \in \mathbb{Z}_{0,N-1}$.

It is shown in [3], that if the initial working set is properly chosen, the QP algorithm still works under Assumption 4.

VII. A MIXED INTEGER QUADRATIC PROGRAMMING ALGORITHM

In this section it is described how the dual active set solver developed in Section VI can be used for solving the subproblems in branch and bound. By using a dual solver for the subproblems, a straightforward reuse of an old working set is enabled and the problem of choosing a feasible initial point is solved. This is now motivated by considering the primal problem (9) and the problem equivalent to the dual (10). The subproblems to be solved in the nodes of the branch and bound tree will be of the type (10). A branch can be interpreted as if a primal inequality constraint is converted to a primal equality constraint. Note that ν and λ are the Lagrange multiplier vectors corresponding to the equality constraints and the inequality constraints respectively. During a branch, the number of elements in ν is increased by one and the number of elements in λ is decreased by one. Hence, a new initial working set can be found from the working set of the parent problem by simply removing any inequality constraint regarding the removed element in λ from the working set. The interpretation in the dual MPC problem

Fig. 1. This plot shows the computational times for two different QP solvers. The QP algorithm described in this paper is implemented in the function `drqp`. The solid line shows the computational time when this algorithm solves the problem from scratch. The dash-dotted line shows the computational time when a warm start is simulated when using `drqp`. The dashed line shows the computational time for the standard QP solver `quadprog`.

(12) is that a sign constrained input signal is replaced by an unconstrained input signal. The conclusion is that during a branch, the feasible set of the dual problem is enlarged and the old feasible set is a subset of the new one. Hence, a feasible dual solution is also a feasible dual solution after a branch.

A possible enhancement, not yet implemented, enabled by the use of a dual solver is to utilize the inequality (5) to prematurely abort the solution process of a subproblem.

VIII. EXAMPLES

In this section, the performance of the presented algorithms is evaluated. A more thorough description of the setup and the results can be found in [3]. The tests have been performed on an Intel Pentium 4 2.66 GHz with 512 Mb RAM running Microsoft Windows XP Professional Version 2002 Service Pack 2 and MATLAB 7.0.1 Service Pack 1. The computational times are obtained using the MATLAB command `cputime`. In the tests, the main objective has been to compare the performance in terms of computational complexity instead of absolute computational time. The absolute values are highly dependent on the level of code optimization, which has been left as future work.

A. Dual Active Set QP

In the computations presented in this section, the dual active set QP algorithm is applied to the MPC problem of controlling a mass in one dimension with a single real control signal $u(t)$. The magnitude of $u(t)$ is limited according to $|u(t)| \leq 9$. By limiting the magnitude of $u(t)$ to 9, a reasonable amount of constraints are active at the optimum.

The problem has been solved for different prediction horizons and the computational times have been measured. The result is shown in Figure 1. The QP algorithm presented in this paper is implemented in the function `drqp`. In the tests, `drqp` is used to solve the problem first from scratch and, second, given the optimal active set. In the latter test, the algorithm starts in the origin and it has to solve one QP subproblem before the optimal solution is found. This test is supposed to, at least roughly, simulate a warm start. Considering prediction horizons from $N = 100$ to $N = 1500$, the computational complexity for `drqp` when cold started is in this test found to be approximately $\mathcal{O}(N^2)$. If the same algorithm is warm started, the computational complexity is reduced to approximately $\mathcal{O}(N)$. The latter result was

Fig. 2. This plot shows the computational times for two different MIQP solvers. The MIQP algorithm described in this paper is implemented in the function `drmiqp`. The solid line shows the computational time when this algorithm is used. The dashed line shows the computational time for the standard MIQP solver `miqp`.

expected since in the warm start case considered, only one QP iteration in the form described in Section VI had to be performed and those are since previously concluded to have computational complexity $\mathcal{O}(N)$. The MATLAB function `quadprog` is found to have an approximate computational complexity of $\mathcal{O}(N^{3.2})$. Therefore, the conclusion from this test is that the algorithm presented in this paper has a significantly lower computational complexity compared to the generic algorithm used in `quadprog`. Also, warm starts are found to be very efficient.

B. MIQP

In the current section, the MIQP algorithm presented in Section VII is applied to the MPC problem to control a satellite with three actuators; two oppositely directed external thrusters and one internal reaction wheel. The thrusters are assumed to be controlled binary and the magnitude of the control signal to the reaction wheel is limited to less than or equal to one. The system contains three states, *i.e.*, the satellite attitude, the angular velocity and the internal wheel velocity. The problem has been solved for several different prediction horizons in the range $N = 10$ to $N = 220$ and the corresponding computational times are presented in Figure 2. The MIQP algorithm presented in this paper is implemented in the function `drmiqp`. In this example, for prediction horizons longer than 20 time steps, `drmiqp` has an approximate computational complexity of $\mathcal{O}(N^{2.7})$, while the standard function `miqp`, [24], using the QP solver `quadprog`, has an approximate computational complexity of $\mathcal{O}(N^{3.4})$. The implementation of the branch and bound algorithm in `drmiqp` has been based on the code in `miqp`, which has been modified in order to be able to use `drqp` and to enable the use of warm starts. Hence, it is exactly the same branch and bound code used in the results for `drmiqp` and `miqp`, and therefore, the branch and bound tree has been explored in exactly the same way. The conclusion from the test is that the MIQP algorithm presented in this paper has a significantly lower computational complexity compared to the generic MIQP algorithm used in `miqp`.

IX. CONCLUSIONS

In this paper a dual active set QP solver tailored for MPC has been derived. By utilizing problem structure, the QP iterations are performed with computational complexity $\mathcal{O}(N)$. The tailored solver has also been used in an MIQP solver where the warm start properties of the dual active set

type of solver have been utilized. Simulation results indicate that both the QP solver itself and the MIQP solver get a significantly lower computational complexity compared to if a standard primal active set QP solver is used. Some suggestions to future work are to, first, try to apply the gradient projection method and, second, relax Assumption 4, in order to allow for pure state constraints and, third, to abort the QP solver in the MIQP solver prematurely.

REFERENCES

- [1] A. Bemporad and M. Morari, "Control of systems integrating logic, dynamics, and constraints," *Automatica*, vol. 35, pp. 407 – 427, 1999.
- [2] L. A. Wolsey, *Integer Programming*. John Wiley & Sons, Inc., 1998.
- [3] D. Axehill, "Applications of integer quadratic programming in control and communication," Licentiate's Thesis, Linköping's universitet, 2005, URL: <http://www.diva-portal.org/liu/theses/abstract.xsql?dbid=5263>.
- [4] C. E. Lemke, "A method of solution for quadratic programs," *Manage. Sci.*, vol. 8, no. 4, pp. 442 – 453, July 1962.
- [5] C. van de Panne and A. Whinston, "The simplex and the dual method for quadratic programming," *Oper. Res. Quart.*, vol. 15, no. 4, pp. 355 – 388, 1964.
- [6] D. Goldfarb and A. Idnani, "A numerically stable dual method for solving strictly convex quadratic programs," *Math. Program.*, vol. 27, pp. 1 – 33, 1983.
- [7] M. J. D. Powell, "On the quadratic programming algorithm of Goldfarb and Idnani," *Math. Program. Stud.*, vol. 25, pp. 46 – 61, 1985.
- [8] N. L. Boland, "A dual-active-set algorithm for positive semi-definite quadratic programming," *Math. Program.*, vol. 78, pp. 1 – 27, 1997.
- [9] R. A. Bartlett and L. T. Biegler, "A dual, active-set, Schur-complement method for large-scale and structured convex quadratic programming," Department of Chemical Engineering, Carnegie Mellon University, Tech. Rep., 2004.
- [10] H. Jonson, "A Newton method for solving non-linear optimal control problems with general constraints," Ph.D. dissertation, Linköping's Tekniska Högskola, 1983.
- [11] C. V. Rao, S. J. Wright, and J. B. Rawlings, "Application of interior-point methods to model predictive control," Mathematics and Computer Science Division, Argonne National Laboratory, Preprint ANL/MCS-P664-0597, May 1997.
- [12] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.
- [13] N. I. M. Gould and P. L. Toint, "A quadratic programming bibliography," Numerical Analysis Group, Rutherford Appleton Laboratory, Tech. Rep., Feb. 2001.
- [14] R. A. Bartlett, A. Wächter, and L. T. Biegler, "Active set vs. interior point strategies for model predictive control," in *Proceedings of the American Control Conference*, June 2000.
- [15] S. J. Wright, "Applying new optimization algorithms to model predictive control," in *Chemical Process Control – V*, J. C. Kantor, C. E. García, and B. Carnahan, Eds., 1997, pp. 147 – 155.
- [16] J. Nocedal and S. J. Wright, *Numerical Optimization*. Springer Verlag, 1999.
- [17] R. Fletcher and S. Leyffer, "Numerical experience with lower bounds for MIQP branch-and-bound," *SIAM J. Optimiz.*, vol. 8, no. 2, pp. 604 – 616, May 1998.
- [18] C. A. Floudas, *Nonlinear and Mixed-Integer Optimization*. Oxford University Press, 1995.
- [19] D. Axehill and A. Hansson, "A preprocessing algorithm for MIQP solvers with applications to MPC," in *Proceedings of the 43th IEEE Conference on Decision and Control*, Atlantis, Paradise Island, Bahamas, Dec. 2004, pp. 2497–2502.
- [20] C. V. Rao, "Moving horizon strategies for the constrained monitoring and control of nonlinear discrete-time systems," Ph.D. dissertation, University of Wisconsin-Madison, 2000.
- [21] G. C. Goodwin, M. M. Seron, and J. A. De Doná, *Constrained Control and Estimation – An Optimisation Approach*. Springer Verlag, 2005.
- [22] M. Åkerblad, "A second order cone programming algorithm for model predictive control," Licentiate's Thesis, Royal Institute of Technology, 2002.
- [23] L. Vandenberghe, S. Boyd, and M. Nouralishahi, "Robust linear programming and optimal control," Department of Electrical Engineering, University of California Los Angeles, Tech. Rep., 2002.
- [24] A. Bemporad and D. Mignone, "A Matlab function for solving mixed integer quadratic programs version 1.02 user guide," Institut für Automatik, ETH, Tech. Rep., 2000.