Relaxations Applicable to Mixed Integer Predictive Control – Comparisons and Efficient Computations

Daniel Axehill, Anders Hansson, Lieven Vandenberghe

Division of Automatic Control

E-mail: daniel@isy.liu.se, hansson@isy.liu.se,
vandenbe@ee.ucla.edu

31st January 2008

Report no.: LiTH-ISY-R-2839

Accepted for publication in Proceedings the 46th 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

Abstract

In this work, different relaxations applicable to an MPC problem with a mix of real valued and binary valued control signals are compared. In the problem description considered, there are linear inequality constraints on states and control signals. The relaxations are related theoretically and both the tightness of the bounds and the computational complexities are compared in numerical experiments. The relaxations considered are the Quadratic Programming (QP) relaxation, the standard Semidefinite Programming (SDP) relaxation and an equality constrained SDP relaxation. The result is that the standard SDP relaxation is the one that usually gives the best bound and is most computationally demanding, while the QP relaxation is the one that gives the worst bound and is least computationally demanding. The equality constrained relaxation presented in this paper often gives a better bound than the QP relaxation and is less computationally demanding compared to the standard SDP relaxation. Furthermore, it is also shown how the equality constrained SDP relaxation can be efficiently computed by solving the Newton system in an Interior Point algorithm using a Riccati recursion. This makes it possible to compute the equality constrained relaxation with approximately linear computational complexity in the prediction horizon.

Keywords: Predictive control, Hybrid systems, Binary control, Integer programming, Semidefinite programming

Relaxations Applicable to Mixed Integer Predictive Control – Comparisons and Efficient Computations

Daniel Axehill, Anders Hansson and Lieven Vandenbergh

Abstract—In this work, different relaxations applicable to an MPC problem with a mix of real valued and binary valued control signals are compared. In the problem description considered, there are linear inequality constraints on states and control signals. The relaxations are related theoretically and both the tightness of the bounds and the computational complexities are compared in numerical experiments. The relaxations considered are the Quadratic Programming (QP) relaxation, the standard Semidefinite Programming (SDP) relaxation and an equality constrained SDP relaxation. The result is that the standard SDP relaxation is the one that usually gives the best bound and is most computationally demanding, while the QP relaxation is the one that gives the worst bound and is least computationally demanding. The equality constrained relaxation presented in this paper often gives a better bound than the QP relaxation and is less computationally demanding compared to the standard SDP relaxation. Furthermore, it is also shown how the equality constrained SDP relaxation can be efficiently computed by solving the Newton system in an Interior Point algorithm using a Riccati recursion. This makes it possible to compute the equality constrained relaxation with approximately linear computational complexity in the prediction horizon.

I. INTRODUCTION

In recent years the field of application of the popular control strategy Model Predictive Control (MPC) has been broadened in several steps. From the beginning, MPC was only applicable to linearly constrained linear systems. Today, it is possible to use MPC for control of nonlinear systems and hybrid systems. In this work, the focus is on control of hybrid systems. In the basic linear setup, the MPC problem can be cast in the form of a Quadratic Programming (QP) problem. When a hybrid system is to be controlled, binary variables are introduced and the optimization problem is changed from a QP to a Mixed Integer Quadratic Programming (MIQP) problem, which is in general known to be $\mathcal{NP}$ -hard, [1]. MPC for hybrid systems is sometimes called Mixed Integer Predictive Control (MIPC). Today, there exist tailored optimization routines with low computational complexity for linear MPC. However, there is still a need for efficient optimization routines for MIPC.

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 and the worst case complexity is known to be exponential. The efficiency of the branch and bound method highly relies on the possibility to efficiently compute good bounds on the optimal objective function value. For MIQP-problems, usually QP relaxations (which are often called linear relaxations), where integer constraints are relaxed to interval constraints, are solved in the nodes to produce these bounds. However, recent research has shown that it is possible to use Semidefinite Programming (SDP) in order to compute tighter bounds for the problem. Unfortunately, solving the SDP relaxation is generally much more time consuming than solving the corresponding QP relaxation. Therefore, it is of greatest interest to investigate if it is possible to decrease the computational complexity. In this work this is done by using problem structure. The SDP relaxations have previously been considered in several contexts and they have successfully been applied to, *e.g.*, the Max Cut problem [2] and the Multiuser Detection problem [3], [4].

In a previous work by the authors, *i.e.*, in [5], the different relaxations have been compared for the special case without inequality constraints. In this work, the results are extended to the general case with inequality constraints on states and control signals. Computational results are shown for the general case with mixed real valued and binary valued control signals. Furthermore, it is shown how the proposed equality constrained SDP relaxation can be efficiently solved by an Interior Point (IP) method where the Newton system is solved using a Riccati recursion. The computational performance of the different relaxations are compared and the expected performance gain from the use of Riccati recursions in the SDP solver are illustrated in numerical experiments.

In this paper, $\mathbb{S}_{++}^n$ ($\mathbb{S}_+^n$) denotes symmetric positive (semi) definite matrices with n columns, and $\mathbb{R}_{++}^n$ denotes positive real n -vectors. Superscript $*$ is used to denote values of variables and functions at optimum. The set $\mathcal{T} = \{0, \dots, N-1\}$ is also frequently used.

II. INTRODUCTION TO THE CONTROL PROBLEM

In this paper, an MIPC problem with a quadratic objective function, or performance measure, in the form

$$J_{\text{MPC}} = \frac{1}{2} \sum_{t=0}^{N-1} (\|e(t)\|_{Q_e}^2 + \|u(t)\|_{Q_u}^2) + \frac{1}{2} \|e(N)\|_{Q_e}^2 \quad (1)$$

is considered, where $\|v\|_Q^2 = v^T Q v$, $Q_e \in \mathbb{S}_{++}^p$ and $Q_u \in \mathbb{S}_{++}^m$. The dynamical system to be controlled is in the form

$$\begin{aligned} x(0) &= x_0 \\ x(t+1) &= Ax(t) + Bu(t) \\ e(t) &= Cx(t) - r(t) \end{aligned} \quad (2)$$

D. Axehill and A. Hansson are with the Division of Automatic Control, Linköpings universitet, SE-581 83 Linköping, Sweden, {daniel,hansson}@isy.liu.se.

L. Vandenbergh is with the Department of Electrical Engineering, University of California Los Angeles, Los Angeles, California 90095-1594, USA, vandenbe@ee.ucla.edu.

where $t \in \mathbb{Z}$ is the discrete time, $x(t) \in \mathbb{R}^n$ is the state, x_0 the initial state, $u(t)^T = [u_c(t)^T \ u_b(t)^T]$ is the control input and $e(t) \in \mathbb{R}^p$ is the control error with reference signal $r(t) \in \mathbb{R}^p$. Furthermore, $u_c(t) \in \mathbb{R}^{m_c}$, $u_b(t) \in \{0, 1\}^{m_b}$ and $m = m_c + m_b$. Note that the choice of partitioning of the control signal into real valued and binary valued components can be made without any loss of generality. The system is also in each time instant subject to c linear inequality constraints in the form

$$\begin{aligned} H_x x(t) + H_u u(t) + h &\leq 0, \ t \in \mathcal{T} \\ H_x x(N) + h &\leq 0 \end{aligned} \quad (3)$$

where $H_x \in \mathbb{R}^{c \times n}$, $H_u \in \mathbb{R}^{c \times m}$ and $h \in \mathbb{R}^c$.

Remark 1: Even though a time-invariant description is chosen in the presentation of this work, all results also hold for the case with a time-varying problem description $A(t)$, $B(t)$, $C(t)$, $H_x(t)$, $H_u(t)$, $h(t)$ and $c(t)$.

It is known from, e.g., [5], that this MIPC problem can be written in at least two different, but equivalent, forms. First, the equality constraints representing the dynamics of the system, can be kept and the result is an MIQP in the form

$$\begin{aligned} \text{minimize}_{x,u,e} \quad & J_{\text{MIQP}_1}(u, e) \\ \text{s.t.} \quad & \begin{bmatrix} A & B & 0 \\ C & 0 & -I \end{bmatrix} \begin{bmatrix} x^T & u^T & e^T \end{bmatrix}^T = \begin{bmatrix} b \\ r \end{bmatrix} \\ & \begin{bmatrix} H_x & H_u & 0 \end{bmatrix} \begin{bmatrix} x^T & u^T & e^T \end{bmatrix}^T + h \leq 0 \\ & u_i \in \{0, 1\}, \ i \in \mathcal{B} \end{aligned} \quad (4)$$

where $J_{\text{MIQP}_1}(u, e) = \frac{1}{2}e^T Q_e e + \frac{1}{2}u^T Q_u u$ and the set $\mathcal{B}$ contains the indices to the binary components in u . Q_e and Q_u are block diagonal matrices with Q_e and Q_u , respectively, along the diagonal. A detailed explanation of the notation can be found in [6, pp. 113–115]. Second, the equality constraints in (2) can be used to eliminate the states and control errors and the resulting optimization problem can be expressed as an MIQP problem equivalent to the problem in (4) in the form

$$\begin{aligned} \text{minimize}_u \quad & J_{\text{MIQP}_2}(u) \\ \text{s.t.} \quad & (H_x S_u + H_u) u + h + H_x S_x x_0 \leq 0 \\ & u_i \in \{0, 1\}, \ i \in \mathcal{B} \end{aligned} \quad (5)$$

where $J_{\text{MIQP}_2}(u) = \frac{1}{2}u^T (E^T Q_e E + Q_u) u + e_0^T Q_e E u + \frac{1}{2}e_0^T Q_e e_0$, $E = -CA^{-1}B$ and $e_0 = CA^{-1}b - r$, and the notation is similar to the one used in [6]. The main idea used in this paper is the fact that $E^T Q_e E + Q_u$ is dense while Q_e and Q_u are block diagonal. The optimal objective function values of the problems in (4) and in (5) coincide and the optimal objective function value is defined as $J^* \triangleq J_{\text{MIQP}_1}^* = J_{\text{MIQP}_2}^* = J_{\text{MPC}}^*$.

III. RELAXATIONS

An optimization problem is said to be a relaxation of another optimization problem if the feasible set is larger than the feasible set of the original problem and the objective functions are equivalent in the two problems. In this work, the integer constraints are relaxed in different ways and the

result is new convex optimization problems that are much easier to solve.

The QP relaxations of the problems in (4) and (5), can easily be derived by replacing the binary constraints with interval constraints. This means that the QP relaxation of the problem in (4) is

$$\begin{aligned} \text{minimize}_{x,u,e} \quad & J_{\text{QP}_1}(u, e) \\ \text{s.t.} \quad & \begin{bmatrix} A & B & 0 \\ C & 0 & -I \end{bmatrix} \begin{bmatrix} x^T & u^T & e^T \end{bmatrix} = \begin{bmatrix} b \\ r \end{bmatrix} \\ & \begin{bmatrix} H_x & H_u & 0 \end{bmatrix} \begin{bmatrix} x^T & u^T & e^T \end{bmatrix} + h \leq 0 \\ & 0 \leq u_i \leq 1, \ i \in \mathcal{B} \end{aligned} \quad (6)$$

where $J_{\text{QP}_1}(u, e) = J_{\text{MIQP}_1}(u, e)$ and the QP relaxation of the problem in (5) is

$$\begin{aligned} \text{minimize}_u \quad & J_{\text{QP}_2}(u) \\ \text{s.t.} \quad & (H_x S_u + H_u) u + h + H_x S_x x_0 \leq 0 \\ & 0 \leq u_i \leq 1, \ i \in \mathcal{B} \end{aligned} \quad (7)$$

where $J_{\text{QP}_2}(u) = J_{\text{MIQP}_2}(u)$.

In recent years, the moment relaxation [7], [8] of problems with binary variables has been extensively studied. The moment relaxation of the problem formulation in (4) is

$$\begin{aligned} \text{minimize}_{U,x,u,e} \quad & J_{\text{SDP}_1}(U, x, u, e) \\ \text{s.t.} \quad & \begin{bmatrix} A & B & 0 \\ C & 0 & -I \end{bmatrix} \begin{bmatrix} x^T & u^T & e^T \end{bmatrix} = \begin{bmatrix} b \\ r \end{bmatrix} \\ & \begin{bmatrix} H_x & H_u & 0 \end{bmatrix} \begin{bmatrix} x^T & u^T & e^T \end{bmatrix} + h \leq 0 \\ & U_{ii} = u_i, \ i \in \mathcal{B} \\ & \begin{bmatrix} U & u \\ u^T & 1 \end{bmatrix} \succeq 0 \end{aligned} \quad (8)$$

where $J_{\text{SDP}_1}(U, x, u, e) = \frac{1}{2}e^T Q_e e + \frac{1}{2} \text{tr}(Q_u U)$, $U \in \mathbb{S}_+^{Nm}$, $x \in \mathbb{R}^{(N+1)n}$, $u \in \mathbb{R}^{Nm}$, $e \in \mathbb{R}^{(N+1)p}$ and where U_{ii} denotes diagonal element i of the matrix U . The relaxation in (8) is in this work referred to as the equality constrained SDP relaxation.

Similarly, the moment relaxation of the problem in (5) can be found to be

$$\begin{aligned} \text{minimize}_{U,u} \quad & J_{\text{SDP}_2}(U, u) \\ \text{s.t.} \quad & (H_x S_u + H_u) u + h + H_x S_x x_0 \leq 0 \\ & U_{ii} = u_i, \ i \in \mathcal{B} \\ & \begin{bmatrix} U & u \\ u^T & 1 \end{bmatrix} \succeq 0 \end{aligned} \quad (9)$$

where $J_{\text{SDP}_2}(U, u) = \frac{1}{2} \text{tr}((E^T Q_e E + Q_u) U) + e_0^T Q_e E u + \frac{1}{2}e_0^T Q_e e_0$, $u \in \mathbb{R}^{Nm}$ and $U \in \mathbb{S}_+^{Nm}$.

The relation between the optimization problems in (4)–(9) has previously been thoroughly discussed in [5]. The result is

$$J_{\text{QP}_1}^* = J_{\text{QP}_2}^* \leq J_{\text{SDP}_1}^* \leq J_{\text{SDP}_2}^* \leq J^* \quad (10)$$

which holds similarly also in the case with inequality constraints. One important conclusion is that the lower bounds on the optimal objective function value from the SDP relaxation does in general depend on if the dynamics is expressed

as equality constraints or if it is eliminated and included in the objective function.

IV. REDUCING COMPUTATIONAL COMPLEXITY

A. Utilizing block structure

The main advantage with the problem in (8) compared to the one in (9) is that the objective function Hessian for the problem in (8) is block diagonal. Specifically, Q_u is block diagonal. This property can be used as described in [5] to rewrite the problem in the equivalent form

$$\begin{aligned}
& \underset{U(t), x(t), u(t), e(t)}{\text{minimize}} && J_{\text{SDP}_1'}(U(t), x(t), u(t), e(t)) \\
& \text{s.t.} && U_{ii}(t) = u_i(t), \quad i \in \{m_c + 1, \dots, m\} \\
& && 0 \preceq \begin{bmatrix} U(t) & u(t) \\ u(t)^T & 1 \end{bmatrix} \\
& && x(0) = x_0 \\
& && x(t+1) = Ax(t) + Bu(t) \\
& && e(t) = Cx(t) + Du(t) - r(t) \\
& && e(N) = Cx(N) - r(N) \\
& && 0 \geq H_x x(t) + H_u u(t) + h \\
& && 0 \geq H_x x(N) + h
\end{aligned} \tag{11}$$

where $t \in \mathcal{T}$ unless stated differently and

$$\begin{aligned}
& J_{\text{SDP}_1'}(U(t), x(t), u(t), e(t)) \\
& = \frac{1}{2} \sum_{t=0}^{N-1} (e(t)^T Q_e e(t) + \text{tr}(Q_u U(t))) \\
& + \frac{1}{2} e(N)^T Q_e(N) e(N)
\end{aligned}$$

For completeness, a direct term from $u(t)$ to $e(t)$, with coefficient matrix D , has been introduced in the dynamics of the system in (11). With the problem in the form in (11), the dynamics is clearly visible and the large matrix variable U has been replaced by several smaller matrix variables $U(t) \in \mathbb{S}_+^m$. The number of variables (matrix elements) now grows linearly in the prediction horizon N .

Remark 2: If Q_u is diagonal, then the problem in (11) can be formulated as a QP and this makes it tractable to replace the ordinary QP relaxation used in branch and bound with the SDP relaxation in (11) (computed as a QP). This is further described in [5] for the case without inequality constraints.

B. Efficient computation of search directions

In this section, it will be shown how the search directions used in an IP solver applied to the problem in (11) can be computed efficiently using Riccati recursions. For notational simplicity, the derivation is performed for the special case with only binary valued control signals, i.e., $m = m_b$. Using the log-determinant barrier function, the associated centering

problem to the SDP problem in (11) can be written as

$$\begin{aligned}
& \underset{U(t), e(t), x(t), u(t)}{\text{minimize}} && J_c(U(t), e(t), x(t), u(t), s) \\
& \text{s.t.} && U_{ii}(t) = u_i(t), \quad i \in \{1, \dots, m\} \\
& && x(0) = x_0 \\
& && x(t+1) = Ax(t) + Bu(t) \\
& && e(t) = Cx(t) + Du(t) - r(t) \\
& && e(N) = Cx(N) - r(N) \\
& && w(t) = 1
\end{aligned} \tag{12}$$

where $t \in \mathcal{T}$ unless stated differently and

$$\begin{aligned}
J_c(U(t), e(t), x(t), u(t), s) &= s J_{\text{SDP}_1'}(U(t), x(t), u(t), e(t)) \\
&+ \sum_{t=0}^{N-1} \left(-\log \det(\tilde{U}(t)) \right) \\
&- \sum_{i=1}^c \log(-h_i - H_{x,i}x(t) - H_{u,i}u(t)) \\
&- \sum_{i=1}^c \log(-h_i - H_{x,i}x(N))
\end{aligned}$$

and where s is the barrier parameter, $U(t) \in \mathbb{S}_+^m$, $u(t) \in \mathbb{R}^m$ and $\tilde{U}(t) = \begin{bmatrix} U(t) & u(t) \\ u(t)^T & w(t) \end{bmatrix}$. After introducing the Lagrangian multipliers $y(t)$, $p(t)$, $\lambda(t)$ and $\nu(t)$, and after forming the Lagrangian, the KKT conditions are found to be

$$\frac{\partial L}{\partial e(t)} = s Q_e e(t) - p(t) = 0, \quad t = 0, \dots, N \tag{13a}$$

$$\begin{aligned}
\frac{\partial L}{\partial x(t)} &= -y(t) + A^T y(t+1) + C^T p(t) \\
&- \sum_{i=1}^c V_{x,i}(x(t), u(t)) = 0
\end{aligned} \tag{13b}$$

$$\begin{aligned}
\frac{\partial L}{\partial x(N)} &= -y(N) + C^T p(N) \\
&- \sum_{i=1}^c V_{x,i}(x(N), u(N)) = 0
\end{aligned} \tag{13c}$$

$$\frac{\partial L}{\partial U(t)} = \frac{s}{2} Q_u - \left(\tilde{U}(t)^{-1} \right)_{11} + \text{diag}(\lambda(t)) = 0 \tag{13d}$$

$$\begin{aligned}
\frac{\partial L}{\partial u(t)} &= -2 \left(\tilde{U}(t)^{-1} \right)_{12} + B^T y(t+1) + D^T p(t) \\
&- \lambda(t) - \sum_{i=1}^c V_{u,i}(x(t), u(t)) = 0
\end{aligned} \tag{13e}$$

$$\frac{\partial L}{\partial w(t)} = - \left(\tilde{U}(t)^{-1} \right)_{22} + \nu(t) = 0 \tag{13f}$$

$$x(0) = x_0 \tag{13g}$$

$$x(t+1) = Ax(t) + Bu(t) \tag{13h}$$

$$e(t) = Cx(t) + Du(t) - r(t) \tag{13i}$$

$$e(N) = Cx(N) - r(N) \tag{13j}$$

$$U_{ii}(t) = u_i(t), \quad i \in \{1, \dots, m\} \tag{13k}$$

$$w(t) = 1 \tag{13l}$$

where $t \in \mathcal{T}$ unless stated differently and the functions $V_{x,i}$ and $V_{u,i}$ are defined in (30). $(\cdot)_{ij}$ denotes sub block (i, j) . In the KKT system in (13), the equations in (13b)–(13f) are nonlinear. The Newton system is found by linearizing the KKT system with respect to $U(t)$, $x(t)$ and $u(t)$ around $U^0(t)$, $x^0(t)$ and $u^0(t)$ respectively. After linearization according to (28), (13d) can approximately be written as

$$\begin{aligned} & \frac{s}{2} Q_u - Z^0(t) + Z^0(t) \Delta U(t) Z^0(t) + z^0(t) \Delta u(t)^T Z^0(t) \\ & + Z^0(t) \Delta u(t) z^0(t)^T + \text{diag}(\lambda(t)) = 0, \quad t \in \mathcal{T} \end{aligned} \quad (14)$$

where $Z^0(t) \in \mathbb{S}_{++}^m$ and $z^0(t) \in \mathbb{R}^m$ are defined in (29). By multiplying the equations in (14) from left and right by $Z^0(t)^{-1}$, $\Delta U(t)$ can be found as

$$\begin{aligned} \Delta U(t) &= Z^0(t)^{-1} - \frac{s}{2} Z^0(t)^{-1} Q_u Z^0(t)^{-1} \\ &- Z^0(t)^{-1} z^0(t) \Delta u(t)^T - \Delta u(t) z^0(t)^T Z^0(t)^{-1} \\ &- Z^0(t)^{-1} \text{diag}(\lambda(t)) Z^0(t)^{-1}, \quad t \in \mathcal{T} \end{aligned} \quad (15)$$

By considering the diagonal of (15), and using the equations in (13k), (31) and (32), the following result is obtained

$$\begin{aligned} \Delta u(t) &= -2 \text{diag}(Z^0(t)^{-1} z^0(t)) \Delta u(t) - \bar{Z}^0(t) \lambda(t) \\ &+ \text{diag}(Z^0(t)^{-1} - \frac{s}{2} Z^0(t)^{-1} Q_u Z^0(t)^{-1}) \end{aligned} \quad (16)$$

By using (16) and that $\bar{Z}^0(t) \succ 0$ (see Appendix), $\lambda(t)$ can be found as

$$\begin{aligned} \lambda(t) &= \bar{Z}^0(t)^{-1} \left\{ - \left(I + 2 \text{diag}(Z^0(t)^{-1} z^0(t)) \right) \Delta u(t) \right. \\ &\left. + \text{diag}(Z^0(t)^{-1} - \frac{s}{2} Z^0(t)^{-1} Q_u Z^0(t)^{-1}) \right\} \end{aligned} \quad (17)$$

After linearization according to (28) of the equation in (13e), the result is

$$\begin{aligned} & 2(z^0(t) z^0(t)^T + v^0(t) Z^0(t)) \Delta u(t) + B^T y(t+1) \\ & + D^T p(t) - \lambda(t) + 2 Z^0(t) \Delta U(t) z^0(t) - 2 z^0(t) \\ & + \sum_{i=1}^c V_{u,i}(x^0(t), u^0(t)) V_{u,i}(x^0(t), u^0(t))^T \Delta u(t) \\ & = \sum_{i=1}^c V_{u,i}(x^0(t), u^0(t)), \quad t \in \mathcal{T} \end{aligned} \quad (18)$$

where the equations in (29), (33) and (34) have been used, and where $v^0(t) \in \mathbb{R}_{++}$ is defined in (29). By using the expression for $\Delta U(t)$ in (15), the equation in (18) can be written as

$$\begin{aligned} & 2(v^0(t) - z^0(t)^T Z^0(t)^{-1} z^0(t)) Z^0(t) \Delta u(t) + B^T y(t+1) \\ & + D^T p(t) - \lambda(t) - 2 \text{diag}(\lambda(t)) Z^0(t)^{-1} z^0(t) \\ & + \sum_{i=1}^c V_{u,i}(x^0(t), u^0(t)) V_{u,i}(x^0(t), u^0(t))^T \Delta u(t) \\ & = s Q_u Z^0(t)^{-1} z^0(t) + \sum_{i=1}^c V_{u,i}(x^0(t), u^0(t)) \end{aligned} \quad (19)$$

By inserting (17) into (19), an expression in the following form will be obtained

$$R(t) \Delta u(t) + B^T y(t+1) + D^T p(t) = l(t), \quad t \in \mathcal{T} \quad (20)$$

where $R(t)$ and $l(t)$ are defined as

$$\begin{aligned} R(t) &= 2(v^0(t) - z^0(t)^T Z^0(t)^{-1} z^0(t)) Z^0(t) \\ &+ \hat{Z}^0(t) \bar{Z}^0(t)^{-1} \hat{Z}^0(t) \\ &+ \sum_{i=1}^c V_{u,i}(x^0(t), u^0(t)) V_{u,i}(x^0(t), u^0(t))^T \\ l(t) &= s Q_u Z^0(t)^{-1} z^0(t) + \hat{Z}^0(t) \cdot \bar{Z}^0(t)^{-1} \\ &\cdot \text{diag}\left(Z^0(t)^{-1} - \frac{s}{2} Z^0(t)^{-1} Q_u Z^0(t)^{-1}\right) \\ &+ \sum_{i=1}^c V_{u,i}(x^0(t), u^0(t)) \end{aligned} \quad (21)$$

where $\hat{Z}^0(t) = (I + 2 \text{diag}(Z^0(t)^{-1} z^0(t)))$. For what follows, it is important to show that $R(t) \succ 0$. Since $U^0(t)$ and $u^0(t)$ are interior points that satisfy the positive semidefinite constraints in the problem strictly, it follows from the definition in (29) that $Z^0(t) \succ 0$ and $\bar{Z}^0(t) \succ 0$. Therefore, it follows from the Schur complement formula that

$$\bar{Z}^0(t) \succ 0 \Leftrightarrow v^0(t) - z^0(t)^T Z^0(t)^{-1} z^0(t) \succ 0 \quad (22)$$

which implies that the first term in $R(t)$ in (21) is positive definite. Since the other terms are quadratic and $\bar{Z}^0(t) \succ 0$, the sum is positive definite. Hence, $R(t)$ is positive definite.

Now, by using $x(t) = x^0(t) + \Delta x(t)$ and $u(t) = u^0(t) + \Delta u(t)$, the equations in (13h) and (13i) can be expressed in $\Delta x(t)$ and $\Delta u(t)$ as follows

$$\begin{aligned} \Delta x(t+1) &= A \Delta x(t) + B \Delta u(t) \\ &- x^0(t+1) + A x^0(t) + B u^0(t) \\ e(t) &= C \Delta x(t) + D \Delta u(t) - r(t) \\ &+ C x^0(t) + D u^0(t) \end{aligned} \quad (23)$$

If the equations in (13a), (13b), (20) and (23) are collected blockwise, the result is

$$\begin{aligned} & \begin{bmatrix} 0 & -I & D \\ -I & s Q_e & 0 \\ D^T & 0 & R(t) \\ C^T & 0 & 0 \\ 0 & 0 & B \end{bmatrix} \begin{bmatrix} C \\ 0 \\ 0 \\ \sum_i V_{x,i}(x^0(t), u^0(t)) V_{x,i}(x^0(t), u^0(t))^T \\ A \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ -I \\ 0 \end{bmatrix} \begin{bmatrix} p(t) \\ e(t) \\ \Delta u(t) \\ \Delta x(t) \\ y(t) \end{bmatrix} \\ & + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ B^T & 0 \\ A^T & 0 \\ 0 & -I \end{bmatrix} \begin{bmatrix} y(t+1) \\ \Delta x(t+1) \end{bmatrix} = \begin{bmatrix} r(t) - C x^0(t) - D u^0(t) \\ 0 \\ l(t) \\ \sum_i V_{x,i}(x^0(t), u^0(t)) \\ x^0(t+1) - A x^0(t) - B u^0(t) \end{bmatrix}, \quad t \in \mathcal{T} \end{aligned} \quad (24)$$

where the sums over i range from 1 to c . After $e(t)$ and $p(t)$ have been eliminated the result can be written in the form

$$\begin{aligned} & \begin{bmatrix} -I & Q_1(t) & K(t) \\ 0 & K(t)^T & Q_2(t) \\ 0 & A & B \end{bmatrix} \begin{bmatrix} y(t) \\ \Delta x(t) \\ \Delta u(t) \end{bmatrix} + \begin{bmatrix} A^T & 0 \\ B^T & 0 \\ 0 & -I \end{bmatrix} \begin{bmatrix} y(t+1) \\ \Delta x(t+1) \end{bmatrix} \\ & = \begin{bmatrix} \bar{q}(t) \\ \bar{r}(t) \\ \bar{b}(t+1) \end{bmatrix}, \quad t \in \mathcal{T} \end{aligned} \quad (25)$$

Algorithm 1 Riccati recursion

```
P(N) = Q1(N)
Ψ(N) = q̄(N)
for t = 1 to N - 1 do
  G(t+1) = Q2(t) + BTP(t+1)B
  H(t+1) = K(t) + ATP(t+1)B
  L(t+1) = G(t+1)-1H(t+1)T
  P(t) = Q1(t) + ATP(t+1)A - H(t+1)L(t+1)
  Ψ(t) = q̄(t) - L(t+1)T(B(t)TΨ(t+1) + r̄(t)) +
    ATΨ(t+1)
end for
x̃(0) = -b̃(0)
for t = 1 to N - 1 do
  ù(t) = -L(t+1)x̃(t)
  + G(t+1)-1(BTΨ(t+1) + r̄(t))
  x̃(t+1) = A x̃(t) + B ù(t) - b̃(t+1)
  y(t) = -Ψ(t) + P(t)x̃(t)
end for
y(N) = -Ψ(N) + Q1(N)x̃(N)
```

where $Q_1(t)$, $Q_2(t)$, $K(t)$, $\bar{q}(t)$, $\bar{r}(t)$ and $\bar{b}(t)$ are defined in (35). Furthermore, at time instants $t = 0$ and $t = N$ the following equations hold

$$x(0) = x_0, -y(N) + Q_1(N)\Delta x(N) = \bar{q}(N) \quad (26)$$

Since $R(t) \succ 0$, the system defined by the equations in (25) and (26) are in a form which can be solved very efficiently using Riccati recursions. A derivation of the Riccati recursions can be found in, *e.g.*, [9], [10], and the resulting recursions can be found in Algorithm 1.

Finally, by using (28), the linearized version of the equation in (13f) is found to be

$$\begin{aligned} & -v^0(t) + z^0(t)^T \Delta U(t) z^0(t) + v^0(t) \Delta u(t)^T z^0(t) \\ & + z^0(t)^T \Delta u(t) v^0(t) + \nu(t) = 0 \end{aligned} \quad (27)$$

As a summary, all computations necessary to solve the Newton system efficiently are presented as Algorithm 2. All computations, including the Riccati recursion, can be performed with linear computational complexity in the prediction horizon. In each iteration in an IP method, usually one or two Newton (like) systems of equations have to be solved. Since in practice the number of iterations is roughly independent of problem size, the overall cost of solving the SDP is roughly proportional to the cost of solving the Newton equations, [11]. It should be mentioned that even though the proposed Riccati method for simplicity is illustrated on a basic barrier method, it is also applicable in a primal-dual IP method as the one described in [11]. Consequently, by using the method presented in this section, the optimization problem in (11) can be solved with approximately linear computational complexity in the prediction horizon.

V. NUMERICAL EXPERIMENTS

In this section, the relaxations in (6), (9) and (11) are compared in numerical experiments. Furthermore, it is shown how large performance gain, compared to a generic solver, that can be expected when the problem in (11) is solved

Algorithm 2 Efficient solution of the Newton system

- 1: Precompute Z^0 , $(Z^0)^{-1}$, $\bar{Z}^0$, $(\bar{Z}^0)^{-1}$, z^0 , v^0 , V_x , V_u , Q_1 , Q_2 , K , $\bar{q}$, $\bar{r}$ and $\bar{b}$ for all time instants necessary.
 - 2: Compute $\Delta x(t)$, $\Delta u(t)$ and $y(t)$ using Algorithm 1 with the definitions in (35).
 - 3: Compute $e(t)$ according to (23) and $p(t)$ according to (13a).
 - 4: Compute $\lambda(t)$ according to (17).
 - 5: Compute $\Delta U(t)$ according to (15).
 - 6: Compute $w(t)$ according to (13l).
 - 7: Compute $\nu(t)$ according to (27).
-

efficiently with an IP solver that solves the Newton system using Riccati recursions. All tests of the computational times were performed on a computer with two processors of the type Dual Core AMD Opteron 270 sharing 4 GB RAM (the code was not written to utilize multiple cores) running CentOS release 4.4 Kernel 2.6.9-42.0.3.ELsmp and MATLAB 7.2. In all experiments with SDP problems, the dual SDP problems were derived manually and given to YALMIP, [12], and they were finally solved by SDPT3 version 4.0, [13]. The optimal solution (MIQP) and the QP relaxation were computed by using CPLEX version 10.010.

In the first experiment, the relative gaps of the different relaxations are compared for different prediction horizons. The relative gap is here defined as $\frac{J^* - J_R^*}{J^*}$, where J_R^* is replaced by the optimal objective function value of the relaxation of interest. The results are presented in Figure 1a and are for each prediction horizon found as the average of 5 problems generated by the MATLAB function `drss` with 4 states, 2 real valued control signals, 2 binary valued control signals and full cost matrices Q_x and Q_u . The two real valued control signals were constrained by a random upper and random lower bound, chosen such that the problems are feasible and such that the constraints are “reasonable active” along the prediction horizon. The result from the experiment clearly confirms the theoretical result in (10). It should be mentioned that the tightness is problem dependent, and the quality of the bounds may vary. The result shows that there are practical control problems where the equality constrained relaxation is useful, *i.e.*, the bound is tighter compared to the QP relaxation and the computational time is less than for the ordinary SDP relaxation.

Note that all approaches actually seem to produce rather good bounds, and that the practical difference seems fairly small in the examples considered. In [5], it has been shown in simulations that rather small improvements in the bounds can actually cut down the branch and bound tree significantly and that this, at least in the case with diagonal Q_u , can decrease the computational time.

In the second experiment, the respective computational times for the different relaxations are compared. The result is presented in Figure 1b and it was found by using the MATLAB command `cputime` and it does not include the time spent in YALMIP. The conclusion is that the standard SDP relaxation is rather slow to compute. The equality

(a) This figure shows the average relative gaps between the optimal objective function value and the optimal objective function values from the relaxations. (b) In this figure, the average computational times for the relaxations are shown.

Fig. 1: The figures present the numerical results illustrating the relative tightness of the relaxations in (6), (9) and (11) and the corresponding computational time.

constrained SDP relaxation is much less computationally demanding for large values of N and the complexity grows significantly slower as N grows. The QP relaxation is the least computationally demanding relaxation to compute. Note that, if Q_u is diagonal, the result in Remark 2 applies and it is then possible to compute the equality constrained SDP relaxation to a computational cost similar to the one of the QP relaxation.

In the third experiment, the computational times for solving the Newton system are investigated. As previously mentioned, a system of equations of this type has to be solved in each iteration in an IP method and the solution of this system is the major cost in the solution process. In the results in Figure 1b, the generic state-of-the-art SDP solver SDPT3 is used to solve the equality constrained SDP relaxation. The purpose with this third experiment is to show that it would be possible to improve the computational times presented in Figure 1b for the equality constrained SDP relaxation by solving the Newton like equation systems in the IP solver using the Riccati approach in Algorithm 2. The expected performance is evaluated by comparing the computational time of certain parts of the code in the primal-dual IP solver SDPT3 with the computational time to perform a similar operation using the Riccati approach in Algorithm 2. The time presented for SDPT3 is the time it takes to form the Schur complement system, [13], and to solve it (similar to solving a Newton system). This is compared to the time it takes to (implicitly) form and solve a similar Newton system using the Riccati approach for a problem of the same size. Each Newton system is solved 10 times and the presented computational time is an average from these 10 experiments. From Figure 2, it can be found that for large values of N , SDPT3 and the Riccati approach solve the system with computational complexities of about $\mathcal{O}(N^{1.9})$ and $\mathcal{O}(N)$ respectively. The absolute computational time is not of major interest here since the Riccati approach is implemented in m-code, while the corresponding operations in SDPT3 are implemented in fast C-code. Our numerical experiments have shown that the number of iterations used in an IP solver applied to the problem in (11) is, as expected, roughly

Fig. 2: The result in this figure compares the computational time for solving the Newton (like) system in the predictor step in SDPT3 for a problem of the type in (11) and the computational time for solving a similar Newton system using the Riccati approach presented in this paper for a problem with similar properties and size. The computational complexity for large values of N grows as $\mathcal{O}(N^{1.9})$ and $\mathcal{O}(N)$ respectively.

independent of the prediction horizon. In our tests, around 15 iterations were needed to solve the problem. Hence, the overall computational complexity is approximately $\mathcal{O}(N^{1.9})$ for the state-of-the-art generic solver SDPT3 but can be reduced to approximately $\mathcal{O}(N)$ if the Riccati approach presented in this paper is applied.

ACKNOWLEDGMENTS

Johan Löfberg is acknowledged for help with YALMIP. The research has been supported by the Swedish Research Council for Engineering Sciences under contract Nr. 621-2002-3822.

VI. CONCLUSIONS

In this paper, the QP relaxation, the standard moment relaxation and an equality constrained moment relaxation have been applied to an MPC problem with mixed real valued and binary valued control signals subject to linear inequality constraints on states and control signals. The respective tightness and computational complexity of the relaxations have been compared. The conclusion is that the best lower bound is achieved by the standard moment relaxation, which is also the most computationally demanding relaxation. Furthermore, the QP relaxation gives the worst lower bound, but is also the least computationally demanding relaxation. The equality constrained moment relaxation presented in this paper gives a bound at least as good as the bound from the QP relaxation and it is less computationally demanding compared to the standard moment relaxation. This is mainly because of the fact that the number of variables are less and they grow linearly in the prediction horizon. In the special case that the cost matrix for the control signal Q_u is diagonal, the equality constrained SDP relaxation can be formulated and solved as a QP. Furthermore, the computational complexity for computing the solution to the equality constrained SDP relaxation is expected to be possible to reduce further by solving the Newton equations in an IP solver by using

Riccati recursions. This approach has been described in detail and promising results from numerical experiments have been presented. A complete implementation of a solver using the proposed Riccati approach and efficient computation of the solution to the standard SDP relaxation are left as future work.

APPENDIX

In this appendix some constants are defined and some help equations are presented. Recall that $\tilde{U}(t) = \begin{bmatrix} U(t) & u(t) \\ u(t)^T & w(t) \end{bmatrix}$ and let $U(t) = U^0(t) + \Delta U(t)$, $u(t) = u^0(t) + \Delta u(t)$ and $w(t) = 1$. For small deviations $\Delta U(t)$ and $\Delta u(t)$ from $U^0(t)$ and $u^0(t)$, and if time indices are neglected, the following result can be found from the Taylor expansion

$$\begin{aligned} (\tilde{U}^{-1})_{11} &\approx Z^0 - Z^0 \Delta U Z^0 - z^0 \Delta u^T Z^0 - Z^0 \Delta u z^{0T} \\ (\tilde{U}^{-1})_{12} &\approx z^0 - Z^0 \Delta U z^0 - z^0 \Delta u^T z^0 - Z^0 \Delta u v^0 \\ (\tilde{U}^{-1})_{22} &\approx v^0 - z^{0T} \Delta U z^0 - v^0 \Delta u^T z^0 - z^{0T} \Delta u v^0 \end{aligned} \quad (28)$$

where the constants $Z^0(t)$, $z^0(t)$ and $v^0(t)$ are defined as

$$\tilde{Z}^0(t) = \begin{bmatrix} Z^0(t) & z^0(t) \\ z^0(t)^T & v^0(t) \end{bmatrix} = \begin{bmatrix} U^0(t) & u^0(t) \\ u^0(t)^T & 1 \end{bmatrix}^{-1} \quad (29)$$

Note that the inverse in (29) always exists since the IP solver generates a sequence of $U^0(t)$ and $u^0(t)$ that always satisfies the positive semidefinite constraint in the problem strictly. Component i of the vector valued functions V_x and V_u respectively are defined as

$$\begin{aligned} V_{x,i}(x(t), u(t)) &= \frac{H_{x,i}^T}{h_i + H_{x,i}x(t) + H_{u,i}u(t)}, \quad t \in \mathcal{T} \\ V_{x,i}(x(N), u(N)) &= \frac{H_{x,i}^T}{h_i + H_{x,i}x(N)} \\ V_{u,i}(x(t), u(t)) &= \frac{H_{u,i}^T}{h_i + H_{x,i}x(t) + H_{u,i}u(t)}, \quad t \in \mathcal{T} \end{aligned} \quad (30)$$

The following equalities hold

$$\begin{aligned} \text{diag} (Z^0(t)^{-1} z^0(t) \Delta u(t)^T + \Delta u(t) z^0(t)^T Z^0(t)^{-1}) \\ = 2 \text{diag} (Z^0(t)^{-1} z^0(t)) \Delta u(t) \end{aligned} \quad (31)$$

$$\begin{aligned} \text{diag} (Z^0(t)^{-1} \text{diag}(\lambda(t)) Z^0(t)^{-1}) \\ = Z^0(t)^{-1} \circ Z^0(t)^{-1} \lambda(t) \triangleq \bar{Z}^0(t) \lambda(t) \end{aligned} \quad (32)$$

where the constant $\bar{Z}^0(t)$ has been defined and $\circ$ is the Hadamard (elementwise) product. Note that $\bar{Z}^0(t) \succ 0$ since $Z^0(t)^{-1} \succ 0$, [14, p. 458]. Furthermore,

$$z^0(t) \Delta u(t)^T z^0(t) = z^0(t) z^0(t)^T \Delta u(t) \quad (33)$$

since $\Delta u(t)^T z^0(t)$ is a scalar.

$$Z^0(t) \Delta u(t) v^0(t) = v^0(t) Z^0(t) \Delta u(t) \quad (34)$$

since $v^0(t)$ is a scalar.

The following definitions are used in order to use Algorithm 2 to solve the Newton system.

$$\bar{b}(0) = x_0, \quad \bar{b}(t) = x^0(t+1) - Ax^0(t) - Bu^0(t)$$

$$\tilde{b}(0) = \bar{b}(0), \quad \tilde{b}(t) = \bar{b}(t)$$

$$\bar{q}(t) = sC^T Q_e (r(t) - Cx^0(t) - Du^0(t))$$

$$+ \sum_{i=1}^c V_{x,i}(x^0(t), u^0(t))$$

$$\bar{q}(t) = \bar{q}(t) + A^T(t)P(t+1)\bar{b}(t+1), \quad \bar{q}(N) = \bar{q}(N)$$

$$\bar{r}(t) = l(t) + sD^T Q_e (r(t) - Cx^0(t) - Du^0(t))$$

$$\bar{r}(t) = \bar{r}(t) + B^T(t)P(t+1)\bar{b}(t+1)$$

$$\tilde{x}(t) = \Delta x(t), \quad \tilde{u}(t) = \Delta u(t)$$

$$Q_1(t) = sC^T Q_e C$$

$$+ \sum_{i=1}^c V_{x,i}(x^0(t), u^0(t)) V_{x,i}(x^0(t), u^0(t))^T$$

$$Q_2(t) = R(t) + sD^T Q_e D, \quad K(t) = sC^T Q_e D \quad (35)$$

REFERENCES

- [1] L. A. Wolsey, *Integer Programming*. John Wiley & Sons, Inc., 1998.
- [2] M. X. Goemans and D. P. Williamson, "878-approximation algorithms for MAX CUT and MAX 2SAT," in *Proceedings of the 26th Annual ACM Symposium on Theory of Computing, STOC'94*, Montréal, Québec, Canada, May 1994, pp. 422–431.
- [3] W. K. Ma, T. N. Davidson, K. Wong, Z. Q. Luo, and P. Ching, "Quasi-maximum-likelihood multiuser detection using semi-definite relaxation," *IEEE Trans. Signal Process.*, vol. 50, no. 4, pp. 912–922, 2002.
- [4] J. Dahl, B. H. Fleury, and L. Vandenberghe, "Approximate maximum-likelihood estimation using semidefinite programming," in *IEEE International Conference on Acoustics, Speech, and Signal Processing 2003*, vol. 6, Apr. 2003, pp. VI – 721–724.
- [5] D. Axehill, L. Vandenberghe, and A. Hansson, "On relaxations applicable to model predictive control for systems with binary control signals," *Automatic Control, ISY*, Tech. Rep. 2771, Jan. 2007. [Online]. Available: <http://www.rti.isy.liu.se/publications/doc?id=1928>
- [6] D. Axehill, "Applications of integer quadratic programming in control and communication," Licentiate's Thesis, Linköping universitet, 2005. [Online]. Available: <http://www.diva-portal.org/liu/theses/abstract.xsql?dbid=5263>
- [7] J. B. Lasserre, "Global optimization with polynomials and the problem of moments," *SIAM J. Optimiz.*, vol. 11, no. 3, pp. 796 – 817, 2001.
- [8] H. Wolkowicz, R. Saigal, and L. Vandenberghe, Eds., *Handbook of Semidefinite Programming – Theory, Algorithms and Applications*. Kluwer, 2000.
- [9] 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.
- [10] M. Åkerblad, "A second order cone programming algorithm for model predictive control," Licentiate's Thesis, Royal Institute of Technology, 2002.
- [11] L. Vandenberghe, V. R. Balakrishnan, R. Wallin, A. Hansson, and T. Roh, *Interior-point algorithms for semidefinite programming problems derived from the KYP lemma*, ser. Lecture notes in control and information sciences. Springer, Feb. 2005, vol. 312, pp. 195 – 238.
- [12] J. Löfberg, "Yalmip: A toolbox for modeling and optimization in MATLAB," in *Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004. [Online]. Available: <http://control.ee.ethz.ch/~joloef/yalmip.php>
- [13] K. C. Toh, R. Tütüncü, and M. J. Todd, "On the implementation and usage of SDPT3 – a MATLAB software package for semidefinite-quadratic-linear programming, version 4.0," Department of Mathematics, National University of Singapore, Tech. Rep., Jul. 2006.
- [14] R. A. Horn and C. R. Johnson, *Matrix analysis*. Cambridge University Press, 1985.