07-95

Formulation of Closed-Loop Min–Max MPC as a Quadratically Constrained Quadratic Program

Moritz Diehl

***Abstract*—In this note, we show that min-max model predictive control (MPC) for linearly constrained polytopic systems with quadratic cost can be cast as a quadratically constrained quadratic program (QCQP). We use the rigorous closed loop formulation of min–max MPC, and show that any such min–max MPC problem with convex costs and constraints can be cast as a finite dimensional convex optimization problem, with the QCQP arising from quadratic costs as a special case. At the base of the proof is a lemma showing the convexity of the dynamic programming cost-to-go, which implies that the worst case on an infinite polytopic set is assumed on one of its finitely many vertices. As the approach is based on a scenario tree formulation, the number of variables in this problem grows exponentially with the horizon length. Fortunately, the QCQP is tree structured, and can thus be efficiently solved by specially tailored interior-point methods whose computational costs are linear in the number of variables. The new formulation as a tree sparse QCQP promises to facilitate online solution of the rigorous min–max MPC problem with quadratic costs.**

***Index Terms*—Constraints, convex programming, model predictive control (MPC), receding horizon control (RHC), robustness.**

I. Introduction

Model predictive control (MPC) is a nowadays classical control technique based on the online optimization of an open-loop control sequence on a moving horizon, in order to minimize a nominal cost function subject to constraints on inputs and deterministically predicted states. Only the first step of the optimized control sequence is injected into the real plant, and after one sampling time a new open-loop problem is solved, in order to generate feedback to possible disturbances. The ability of MPC to cope with hard control and state constraints is one of its major advantages and is an important reason for its success in industry [1]. Note that a key assumption in classical MPC is that the system is deterministic and known. However, in many practical applications—in particular when feedback control is most urgently needed—we know that this assumption is not valid. In **robust** MPC, originally proposed by Witsenhausen [2], the system is therefore not assumed to be known exactly, and the optimization is performed against the worst-case system behaviour. This typically leads to min-max optimization problems. In this note, we are concerned with discrete-time systems only. Robust MPC formulations differ in the way in which they model uncertainty and how they incorporate feedback in the optimization horizon. Lee and Yu [3] address this question by distinguishing between an open and a closed loop min-max formulation of robust MPC. In the open-loop formulation the optimizer chooses all controls on the moving horizon such that the worst-case system behaviour for the open-loop control sequence is minimized; this formulation leads to large but tractable optimization problems treated for example in [4]–[7]. However, it may lead to overly conservative controllers. As a possible remedy, parameterized feedback on the optimization horizon has been proposed e.g., in [8] and [9].

Manuscript received May 30, 2006; revised August 26, 2006. Recommended by Associate Editor L. Magni. This work was supported by the DFG under Grant BO864/10-1.

The author is with the Center of Excellence "Optimization in Engineering"/ Electrical Engineering Department, K.U. Leuven, 3001 Leuven-Heverlee, Belgium (e-mail: moritz.diehl@esat.kuleuven.be).

Digital Object Identifier 10.1109/TAC.2006.890372

In the rigorous closed-loop formulation, however, which we treat in this note, feedback is considered by adressing the recurrent nature of the optimization directly. This leads to a multi-stage min-max optimization problem which is in general difficult to treat, but can be attacked by robust dynamic programming [3], [10]. Scokaert and Mayne [11] formulate such a closed-loop min–max problem for linear systems with convex piecewise affine (CPWA) costs and bounded additive disturbances, and also Kerrigan and Maciejowski [12] adopt this approach and formulate the problem as a linear program. The min-max MPC problem with CPWA costs is also addressed in [13]–[15] where the more general case of linear systems with polytopic uncertainty is treated. In this note, we address the same uncertain system class, but allow also a quadratic cost function, which is a crucial feature of classical MPC that is used, for example, in classical proofs for MPC closed-loop stability. This small, but important difference does no longer allow us to use (parametric) linear programming based techniques as [13]–[15] or [11], [12], but leads us directly to quadratically constrained QPs, a class of convex optimization optimization problems.

II. Robust MPC With Quadratic Costs

In this note, we consider uncertain discrete-time systems

$$x_{k+1} = f_k(x_k, u_k), \qquad f_k \in \mathsf{F}, \quad k \in \mathbb{N}$$

with states $x_k \in \mathbb{R}^{n_x}$ and controls $u_k \in \mathbb{R}^{n_u}$. Here the transition functions f_k in each step are uncertain, but we assume they are known to be in a certain set F. In particular, we regard *polytopic* systems that are linear and of the form

$$f(x, u) = Ax + Bu + c \tag{1}$$

where the matrices A and B and the vector c are contained in a polytope[1]

$$\mathsf{F} = \mathbf{conv}\left\{(A_1|B_1|c_1), \ldots, (A_{n_f}|B_{n_f}|c_{n_f})\right\}. \tag{2}$$

Each vertex of this polytope is denoted by $f^\nu(x, u) = A_\nu x + B_\nu u + c_\nu$, $\nu = 1, \ldots, n_f$. Given an N-stage *policy* $\pi = (u_0(\cdot), u_1(\cdot), \ldots, u_{N-1}(\cdot))$, and given an *uncertainty realization* $\phi = (f_0, f_1, \ldots, f_{N-1}) \in \mathsf{F}^N$, as well as an initial value x_0, we define the corresponding state sequence x_k^{π,ϕ,x_0}, $k = 0, \ldots, N-1$, by $x_0^{\pi,\phi,x_0} = x_0$ and $x_{k+1}^{\pi,\phi,x_0} = f_k(x_k^{\pi,\phi,x_0}, u_k(x_k^{\pi,\phi,x_0}))$. In the closed-loop formulation of robust model predictive control, feedback is obtained by formulating an N-stage min–max optimization problem and employing the first part $u_0(\cdot)$ of an N-stage policy π that solves, for any x_0, the optimization problem

$$\begin{aligned} \min_{\pi} \max_{\phi \in \mathsf{F}^N} & \sum_{k=0}^{N-1} L\left(x_k^{\pi,\phi,x_0}, u_k\left(x_k^{\pi,\phi,x_0}\right)\right) + V_N\left(x_N^{\pi,\phi,x_0}\right) \\ \text{subject to}: & \left(x_k^{\pi,\phi,x_0}, u_k\left(x_k^{\pi,\phi,x_0}\right)\right) \in \mathsf{L}, x_N^{\pi,\phi,x_0} \in \mathsf{X}_N \\ & \forall \phi \in \mathsf{F}^N, \qquad k \in \{0, \ldots, N-1\}. \end{aligned} \tag{3}$$

[1]We identify each $f \in \mathsf{F}$ with the corresponding matrix $(A\,|\,B\,|\,c)$, and we assume all terms in the convex combination to be nonredundant, i.e., to be vertices of F.

0018-9286/$25.00 © 2007 IEEE

339-343

The sets L and X_N specify constraints on states and controls. In this note—as customary in MPC applications—we treat linear constraints that result in polyhedral feasible sets

$$\mathsf{L} = \left\{ \begin{bmatrix} x \\ u \end{bmatrix} \middle| \, Fx + Gu \leq g \right\} \quad \mathsf{X}_N = \{x \,|\, P_N x \leq p_N\}. \tag{4}$$

We refer to L as the stage cost and to V_N as the terminal cost, which are in this note—like in classical MPC formulations—both assumed to be convex quadratic, namely

$$L(x,u) = x^T Q x + u^T R u \quad V_N(x) = x^T Q_N x \tag{5}$$

with positive semidefinite matrices Q, R, Q_N. We point out, however, that the linear constraints could without difficulty for the methods proposed in this note be augmented with some convex quadratic inequalities, and that the cost function could contain also convex piecewise linear contributions as used in [11]–[15]. We omit both generalizations for notational simplicity.

III. Scenario Tree Formulation of the Problem

Let us assume for a moment—and prove later—that the worst case in each stage is always attained in one of the vertices of the polytopic uncertainty set F. We will show that, for a fixed x_0, the optimization problem (3) over control policies can be recast as a standard, but large nonlinear programming problem with a multistage scenario tree structure. Let $\mathcal{V}$ denote the index set of nodes of this tree, where each node $j \in \mathcal{V}$ corresponds to one specific stage and a specific uncertainty realization. That is, if the set F has n_f vertices, at stage $k = 0$ we have only one node, the root node, at stage $k = 1$ we have n_f nodes, and at stage k we have n_f^k nodes. Altogether, the tree will have $|\mathcal{V}| = \sum_{k=0}^{N} n_f^k$ nodes, that we will number in some ascending order, starting with the root node $j = 0$. For a visualization of a tree, see Fig. 1. The stage corresponding to a node will be denoted by $k(j)$, so the ascending order means that the identity $k(0) = 0$ holds, as well as $k(j) = 1$ for $j = 1, \ldots, n_f$ and $k(j) = 2$ for $j = n_f + 1, \ldots, n_f + n_f^2$ etc. We denote the parent of some node j by the symbol $p(j)$, so that the trivial identity $k(p(j)) = k(j) - 1$ holds for any $j \in \mathcal{V} \backslash \{0\}$. The root node has no parent, and the leaf nodes have no children. We denote the set of all nodes that are children of others by $\mathcal{C} = \mathcal{V} \backslash \{0\}$ and the set of all parents by $\mathcal{P} = \mathcal{V} \backslash \{j \in \mathcal{V} | k(j) = N\}$. To each node $j \in \mathcal{C}$ an uncertainty realization $f^{\nu(j)} \in \mathsf{F}$ is attached, where the integer $\nu(j)$ denotes one of the vertices $\{1, \ldots, n_f\}$ of F. We assume the uncertainty realization happens just *before* stage $k(j)$, i.e., we regard the dynamics $x_j = f^{\nu(j)}(x_{p(j)}, u_{p(j)})$. Together with the realizations of its ancestor nodes, the realization $f^{\nu(j)}$ forms the whole scenario corresponding to the node j, namely the sequence $(f^{\nu(p^{(k(j)-1)}(j))}, \ldots, f^{\nu(p(p(j)))}, f^{\nu(p(j))}, f^{\nu(j)}) \in \mathsf{F}^{k(j)}$. The tree structured problem for a fixed x_0 reads as follows:

$$V_0(x_0) = \min_{\substack{u_j, v_j (j \in \mathcal{P}), \\ x_j (j \in \mathcal{C})}} L(x_0, u_0) + v_0 \tag{6a}$$

$$\text{subject to} \quad (x_j, u_j) \in \mathsf{L} \qquad \forall j \in \mathcal{P} \tag{6b}$$

$$f^{\nu(j)}\left(x_{p(j)}, u_{p(j)}\right) = x_j \qquad \forall j \in \mathcal{C} \tag{6c}$$

$$L(x_j, u_j) + v_j \leq v_{p(j)} \qquad \forall j \in \mathcal{P} \cap \mathcal{C} \tag{6d}$$

$$x_j \in \mathsf{X}_N, V_N(x_j) \leq v_{p(j)} \qquad \forall j \in \mathcal{C} \backslash \mathcal{P}. \tag{6e}$$

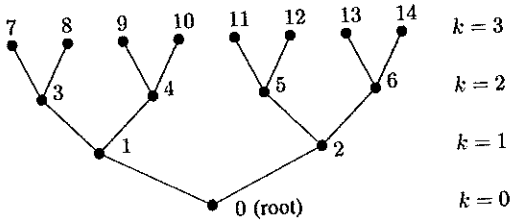


Fig. 1. Scenario tree for $N = 3$ and $n_f = 2$. Here, $\mathcal{C} = \{1, 2, \ldots, 14\}$ and $\mathcal{P} = \{0, 1, \ldots, 6\}$. It can be seen that e.g., the parent of node 2 is $p(2) = 0$, and that its children are $C(2) = \{5, 6\}$, and that its grandchildren are $C^2(2) = \{11, 12, 13, 14\}$.

Aim of this note is to prove that the min-max MPC problem can indeed be formulated as this (convex) nonlinear programming problem, and to discuss its specific structure and numerical solution.

IV. Equivalence of Min–Max MPC and Scenario Tree Formulation

In order to prove that the min–max MPC problem (3) over control policies is equivalent, for fixed x_0, to the scenario tree formulation (6), we will use a dynamic programming based viewpoint. Dynamic programming [16] is a well established technique for solution of the original min–max MPC problem (3), as investigated e.g., in [3]. Dynamic programming with constraints, as in this note, was employed for (robust) MPC problems in [13], [17], [14], [15]. We make use of the so called dynamic programming *cost-to-go functions* $V_k(\cdot)$ with corresponding *feasible sets* $\mathsf{X}_k \subset \mathsf{R}^{n_x}$, i.e., $V_k : \mathsf{X}_k \to \mathsf{R}$. Starting with V_N and X_N, we can compute V_k and X_k recursively, for $k = N-1, \ldots, 1$, by the robust Bellman equation with constraints (cf. [17], [14], [10], and [15])

$$V_k(x) := \min_{u \in \mathsf{R}^{n_u}} L(x,u) + \max_{f \in \mathsf{F}} V_{k+1}(f(x,u)) \tag{7a}$$

$$\text{subject to} \quad (x,u) \in \mathsf{L} \tag{7b}$$

$$f(x,u) \in \mathsf{X}_{k+1} \qquad \forall f \in \mathsf{F}. \tag{7c}$$

The set X_k is defined to be the set of all x for which the above problem is feasible.

For polytopic systems the worst case in the objective (7a) and in the constraint (7c) is taken, for each (x,u), over a polytopic set, namely the set $\mathbf{conv}(f^1(x,u), \ldots, f^{n_f}(x,u))$. Therefore, if both X_{k+1} and V_{k+1} are convex, it is well known [11] that it is sufficient to maximize in (7a) only over the vertices of this polytopic set, as well as to check feasibility of (7c) only in the vertices. Therefore, after introduction of some slack variables, problem (7) is equivalent to the finite-dimensional optimization problem:

$$V_k(x) = \min_{u \in \mathsf{R}^{n_u}, x_\nu \in \mathsf{R}^{n_x}, v \in \mathsf{R}} L(x,u) + v \tag{8a}$$

$$\text{subject to} \quad (x,u) \in \mathsf{L} \tag{8b}$$

$$A_\nu x + B_\nu u + c_\nu = x_\nu, \qquad \nu = 1, \ldots, n_f \tag{8c}$$

$$x_\nu \in \mathsf{X}_{k+1}, \qquad \nu = 1, \ldots, n_f \tag{8d}$$

$$V_{k+1}(x_\nu) \leq v, \qquad \nu = 1, \ldots, n_f. \tag{8e}$$

This problem is convex, if in addition to X_{k+1} and V_{k+1} also L and L are convex. It is easily shown that under these conditions also X_k

and V_k are convex, e.g., by observing that the epigraph of V_k on X_k is nothing else than the projection of the convex set

$$\left\{ \begin{bmatrix} x \\ u \\ x_1 \\ \vdots \\ x_{n_f} \\ v \\ s \end{bmatrix} \middle| \begin{array}{c} L(x,u)+v \le s \\ (x,u) \in \mathsf{L} \\ A_\nu x + B_\nu u + c_\nu = x_\nu, \quad \nu = 1,\dots,n_f \\ x_\nu \in \mathsf{X}_{k+1}, \quad \nu = 1,\dots,n_f \\ V_{k+1}(x_\nu) \le v, \quad \nu = 1,\dots,n_f. \end{array} \right\}$$

onto the (x,s)-dimension. Thus, the following lemma holds.

Lemma 4.1 (Finite representation of worst case): For polytopic systems with convex constraint sets L, X_N and convex cost functions L, V_N, the dynamic programming recursion (7) leads to feasible sets X_k and cost-to-go functions V_k that are both convex for all $k = N,\dots,0$. Thus, the solution of the semi-infinite problem (7) can be determined as the solution of the finite dimensional problem (8).

Let us now regard again the scenario tree with node set $\mathcal{V}$, and let us pick one arbitrary node $i \in \mathcal{P}$ at stage $k(i)$ that is parent to its set of children nodes, $C(i) := \{j \in \mathcal{V} | p(j) = i\}$. From construction of the tree, we have that $\{\nu(j) | j \in C(i)\} = \{1,\dots,n_f\}$. That is, we can represent the cost-to-go $V_{k(i)}(x_i)$ by the problem

$$\begin{aligned} V_{k(i)}(x_i) = \min_{u_i \in \mathbb{R}^{n_u}, x_j, v_i} \; & L(x_i,u_i) + v_i && \text{(9a)} \\ \text{subject to} \quad & (x_i,u_i) \in \mathsf{L} && \text{(9b)} \\ & A_{\nu(j)} x_{p(j)} + B_{\nu(j)} u_{p(j)} + c_{\nu(j)} = x_j \quad \forall j \in C(i) && \text{(9c)} \\ & x_j \in \mathsf{X}_{k(i)+1} \quad \forall j \in C(i) && \text{(9d)} \\ & V_{k(i)+1}(x_j) \le v_i \quad \forall j \in C(i). && \text{(9e)} \end{aligned}$$

Note that in (9c) we have used $p(j)$ as an equivalent expression for i, in order to prepare already the notation for the next step. We now observe that the last two conditions, (9d) and (9e), can jointly be reformulated by using, recursively, the same definition of X_{k+1} and V_{k+1}, i.e., by regarding the node set $C^2(i) := \{j \in \mathcal{V} | p(p(j)) = i\}$ and by adding, again, slack variables corresponding to all grand-children nodes $j \in C^2(i)$. This would lead us to the following convex optimization problem:

$$\begin{aligned} V_{k(i)}(x_i) = \min_{u_j, v_j, x_j} \; & L(x_i,u_i) + v_i && \text{(10a)} \\ \text{subject to} \quad & (x_j,u_j) \in \mathsf{L}, \quad \forall j \in \{i\} \cup C(i) && \text{(10b)} \\ & A_{\nu(j)} x_{p(j)} + B_{\nu(j)} u_{p(j)} + c_{\nu(j)} = x_j \quad \forall j \in C(i) \cup C^2(i) && \text{(10c)} \\ & L(x_j,u_j) + v_j \le v_{p(j)} \quad \forall j \in C(i) && \text{(10d)} \\ & x_j \in \mathsf{X}_{k(i)+2}, V_{k(i)+2}(x_j) \le v_{p(j)} \quad \forall j \in C^2(i). && \text{(10e)} \end{aligned}$$

Note that the free variables in this problem are u_j, v_j for $j \in \{i\} \cup C(i)$, and x_j for $j \in C(i) \cup C^2(i)$. Note also that constraints (10b) apply to all nodes j that are parents of other nodes, and that constraints (10c) apply to all nodes j that are children of others, while constraints (10d) apply to all nodes j that are both parents and children of others. We now recursively apply the procedure again and again, i.e., we always reformulate the final constraints (10e) by use of extra slack variables. It is easily seen that the problem for $i = 0$ becomes, after N steps of this procedure, a problem in the variables u_j, v_j, $j \in \mathcal{P}$ and $x_j \in \mathcal{C}$, namely the scenario tree formulation (6). Therefore, we have shown the central result of this short note.[2]

Theorem 4.2 (Min–Max MPC as a Convex Program): The min–max MPC problem (3) for polytopic systems with convex constraint sets L, X_N and convex cost functions L, V_N can be reformulated as a tree structured convex programming problem of the form (6).

For the special case of a min–max MPC problem with quadratic costs and linear constraints, the problem becomes a specific form of convex problem, namely a quadratically constrained quadratic program (QCQP), as e.g., defined in [18]. As before, we define the vertex of $\mathsf{F} = \text{conv}\{(A_1|B_1|c_1),\dots,(A_{n_f}|B_{n_f}|c_{n_f})\}$ that corresponds to the uncertainty realization of node j by $\nu(j) \in \{1,\dots,n_f\}$. The QCQP reads as follows:

$$\begin{aligned} \min_{u_j, v_j (j\in\mathcal{P}), x_j (j\in\mathcal{C})} \; & x_0^T Q x_0 + u_0^T R u_0 + v_0 && \text{(11a)} \\ \text{subject to} \quad & F x_j + G u_j \le g \quad \forall j \in \mathcal{P} && \text{(11b)} \\ & A_{\nu(j)} x_{p(j)} + B_{\nu(j)} u_{p(j)} + c_{\nu(j)} = x_j \quad \forall j \in \mathcal{C} && \text{(11c)} \\ & x_j^T Q x_j + u_j^T R u_j + v_j \le v_{p(j)} \quad \forall j \in \mathcal{C} \cap \mathcal{P} && \text{(11d)} \\ & x_j^T Q_N x_j - v_{p(j)} \le 0 \quad \forall j \in \mathcal{C} \backslash \mathcal{P}, && \text{(11e)} \\ & P_N x_j \le p_N \quad \forall j \in \mathcal{C} \backslash \mathcal{P}. && \text{(11f)} \end{aligned}$$

V. Solution of the Tree Sparse QCQP by Interior-Point Methods

In order to solve the tree sparse QCQP (11) efficiently, we propose to use a tree structured interior-point method, as proposed for a very similar class of tree sparse convex programs by Steinbach [19]. For excellent introductions into interior-point methods we refer, e.g., to [20] or to [18]. In its simplest form, an interior-point method proceeds by solving, for a shrinking *barrier parameter* $\mu \to 0$, a sequence of problems where all inequality constraints are omitted, but instead represented by a logarithmic penalty in the objective. In the case of the tree sparse QCQP (11) of this note, the barrier optimization problem for fixed $\mu > 0$ would be given by

$$\begin{aligned} \min_{u_j, v_j (j\in\mathcal{P}) \atop x_j (j\in\mathcal{C})} \; & \mu^{-1} \left(x_0^T Q x_0 + u_0^T R u_0 + v_0 \right) \\ & + \sum_{j\in\mathcal{P}} \log(g - F x_j - G u_j) \\ & + \sum_{j\in\mathcal{C}\cap\mathcal{P}} \log\left(v_{p(j)} - x_j^T Q x_j - u_j^T R u_j - v_j \right) \\ & + \sum_{j\in\mathcal{C}\backslash\mathcal{P}} \log\left(v_{p(j)} - x_j^T Q_N x_j \right) \\ & + \sum_{j\in\mathcal{C}\backslash\mathcal{P}} \log(p_N - P_N x_j) \\ \text{subject to} \quad & \forall j \in \mathcal{C} : x_j = A_{\nu(j)} x_{p(j)} \\ & + B_{\nu(j)} u_{p(j)} + c_{\nu(j)}. \end{aligned} \tag{12}$$

The objective function of this problem is self-concordant as, e.g., shown in [18] and thus guarantees efficiency of an interior-point

[2]It is interesting to note that in contrast to [11] no "causality constraint" is present in the formulation (6). This is due to the fact that in our formulation, the number of decision variables is growing with increasing stage index k, while in [11] all possible scenarios (corresponding to the leaves of our tree) carry a complete control sequence with them.

method when applied to a sequence of problems with shrinking μ. If all variables of this problem together are summarized by w, it can be summarized as

$$\min_{w} \Phi_\mu(w) \quad \text{s.t.} \quad \Psi(w) = 0$$

with Lagrangian $\mathcal{L}_\mu(w,\lambda) = \Phi_\mu(w) + \lambda^T \Psi(w)$. If the components of the vector w are sorted in the order $w = (w_0, w_1, \ldots, w_{|\mathcal{V}|-1})$ with $w_0 = (u_0, v_0)$, $w_1 = (x_1, u_1, v_1), \ldots, w_M = (x_M, u_M, v_M)$, where $M = |\mathcal{C} \cap \mathcal{P}|$, and $w_{M+1} = x_{M+1}, \ldots, w_{|\mathcal{V}|-1} = x_{|\mathcal{V}|-1}$, then the Lagrangian $\mathcal{L}_\mu$ can be expressed as

$$\mathcal{L}_\mu(w,\lambda) = \phi_\mu^0(w_0,\lambda) + \sum_{j \in \mathcal{V}+} \phi_\mu^j\left(w_j, w_{p(j)}, \lambda\right).$$

The necessary and sufficient conditions for optimality of a pair (x^*, λ^*), namely

$$\nabla_w \mathcal{L}_\mu(w^*, \lambda^*) = 0, \Psi(w) = 0. \tag{13}$$

Therefore, have a tree structured jacobian. Within any interior-point algorithm, a Newton type method is applied to solve (13) for a given μ, and the tree sparse structure of this jacobian can efficiently be exploited by an approach using the generalized Riccati recursion that was proposed in [19]. The numerical effort per Newton type iteration of this approach grows linearly in the number of nodes in the tree. Because the total number of Newton type iterations within an interior-point method is bounded to be below a few tens in both theory and practice, the QCQP is solvable with a numerical effort linear in the number of nodes of the tree $|\mathcal{V}|$.

It is interesting to observe that in the special case of deterministic MPC the tree degenerates to a sequence. In this case the resulting block sparse quadratic programming problem (QP) is banded, and can also be solved by a Riccati based interior-point QP approach. This idea was proposed in [21] and [22], [23]. It was already successfully applied to QPs in deterministic MPC by Rao *et al.* [24].

VI. Conclusion

In this note, we have shown that the closed loop min-max MPC problem for polytopic systems with quadratic costs and linear constraints can be formulated as a QCQP. While the possibility to formulate the min–max MPC problem *with piecewise linear costs* as a large structured linear programming problem was previously known [11], [12], in this note we have shown that the min–max MPC problem for polytopic systems with arbitrary convex costs and constraints can always be cast as a finite dimensional convex optimization problem (Theorem 4.2). A special case is min-max MPC with quadratic costs that leads to a (convex) QCQP. Finally, we discussed how the tree sparse structure of the QCQP can be exploited within an efficient interior-point method, employing a recently developed generalized Riccati equation based approach [19]. Future work will focus on an efficient implementation of the method, and its application to nontrivial examples of min–max MPC with quadratic cost, as well as its application to more general min–max MPC problems with convex costs and constraints.

Acknowledgment

The author would like to thank S. Sager and an anonymous referee for helpful comments.

References

[1] S. Qin and T. Badgwell, "A survey of industrial model predictive control technology," *Control Eng. Pract.*, vol. 11, pp. 733–764, 2003.
[2] H. Witsenhausen, "A minimax control problem for sampled linear systems," *IEEE Trans. Autom. Control*, vol. AC-13, no. 1, pp. 5–21, Jan. 1968.
[3] J. H. Lee and Z. Yu, "Worst-case formulations of model predictive control for systems with bounded parameters," *Automatica*, vol. 33, no. 5, pp. 763–781, 1997.
[4] P. J. Campo and M. Morari, "Robust model predictive control," in *Proc. Amer. Control Conf.*, 1987, pp. 1021–1026.
[5] H. Genceli and M. Nikolaou, "Design of robust constrained model-predictive controllers with volterra series," *AIChE J.*, vol. 41, no. 9, pp. 2098–2107, 1995.
[6] S. Oliveira and M. Morari, "Robust model predictive control for nonlinear systems," in *Proc. 33rd IEEE Conf. Decision Control*, Lake Buena Vista, FL, Dec. 1994, pp. 3561–3566, IEEE.
[7] H. Chen, *Stability and Robustness Considerations in Nonlinear Model Predictive Control*, ser. Fortschr.-Ber. Düsseldorf, Germany: VDI Verlag, 1997, VDI Reihe 8 Nr. 674.
[8] M. Kothare, V. Balakrishnan, and M. Morari, "Robust constrained model predictive control using linear matrix inequalities," *Automatica*, vol. 32, no. 10, pp. 1361–1379, Nov. 1996.
[9] F. Fontes and L. Magni, "Min-max model predictive control of nonlinear systems using discontinuous feedbacks," *IEEE Trans. Autom. Control* **PLEASE PROVIDE ISSUE AND MONTH**, vol. 48, pp. 1750–1755, 2003.
[10] L. Grüne and O. Junge, Global optimal control of perturbed systems Univ. Bayreuth, Bayreuth, Germany, 2005, Tech. Rep. [Online]. Available: www.math.uni-bayreuth.de/lgruene/publ/robsetstab.html
[11] P. O. M. Scokaert and D. Q. Mayne, "Min-max feedback model predictive control for constrained linear systems," *IEEE Trans. Autom. Control* **PLEASE PROVIDE PROVIDE ISSUE AND MONTH**, vol. 43, pp. 1136–1142, 1998.
[12] E. Kerrigan and J. Maciejowski, "Feedback min-max model predictive control using a single linear program: Robust stability and the explicit solution," *Int. J. Robust Nonlinear Control*, vol. 14, pp. 395–413, 2004.
[13] A. Bemporad, F. Borrelli, and M. Morari, "Min-max control of constrained uncertain discrete-time linear systems," *IEEE Trans. Autom. Control* **PLEASE UPDATE AS NEEDED**, 2003, in press.
[14] M. Diehl and J. Björnberg, "Robust dynamic programming for min-max model predictive control of constrained uncertain systems," *IEEE Trans. Autom. Control*, vol. 49, no. 12, pp. 2253–2257, Dec. 2004.
[15] J. Björnberg and M. Diehl, "Approximate robust dynamic programming and robustly stable MPC," *Automatica*, vol. 42, no. 5, pp. 777–782, May 2007.
[16] D. Bertsekas, *Dynamic Programming and Optimal Control*. Belmont, MA: Athena Scientific, 1995, vol. 1 and 2.
[17] D. Q. Mayne, "Control of constrained dynamic systems," *Eur. J. Control*, vol. 7, pp. 87–99, 2001.
[18] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[19] M. C. Steinbach, "Tree-sparse convex programs," *Math. Meth. Oper. Res.*, vol. 56, no. 3, pp. 347–376, 2002.
[20] S. Wright, *Primal-Dual Interior-Point Methods*. Philadelphia, PA: SIAM, 1997.
[21] M. Steinbach, "Fast recursive SQP methods for large-scale optimal control problems," Ph.D. dissertation, Univ. Heidelberg, Heidelberg, Germany, 1995, IWR.
[22] J. Pantoja, "Differential dynamic programming and Newton's method," *Int. J. Control*, vol. 47, no. 5, pp. 1539–1553, 1988.
[23] J. Dunn and D. Bertsekas, "Efficient dynamic programming implementations of Newton's method for unconstrained optimal control problems," *J. Optim. Theory Appl.*, vol. 63, pp. 23–38, 1989.
[24] C. Rao, S. Wright, and J. Rawlings, "Application of interior-point methods to model predictive control," *J. Optim. Theory Appl.*, vol. 99, pp. 723–757, 1998.