Multiparametric Nonlinear Integer Programming and Explicit Quantized Optimal Control

Alberto Bemporad
Dip. Ingegneria dell’Informazione
University of Siena
Via Roma 56, 53100 Siena, Italy
Email: bemporad@unisi.it

August 27, 2003

Technical Report, Dept. Information Engineering, University of Siena

Abstract

This paper deals with multiparametric nonlinear integer problems where the optimization variables belong to a finite set and where the cost function and the constraints depend in an arbitrary nonlinear fashion on the optimization variables and in a linear fashion on the parameters. We examine the main theoretical properties of the optimizer and of the optimum as a function of the parameters, and propose a solution algorithm. The methodology is employed to investigate properties of quantized optimal control laws and optimal performance, and to obtain their explicit representation as a function of the state vector.

1 Introduction

In several control synthesis problems the number of possible control actions is finite, a situation usually referred to as quantization of the input signals. While in most applications the quantization introduced by analog-to-digital converters, finite precision arithmetic units, and digital to analog converters can be safely neglected by treating the control variables as continuous, in some problems this assumption may lead to an unacceptable deterioration of the closed-loop performance. Examples of control problems that must handle quantization range from more traditional mechanical problems (e.g., problems involving stepping motors) and hydraulic problems (e.g., with on/off valves), to new problems in communications, such as the one dealt with in [1], where quantized control is used to coordinate adaptation of multimedia applications and hardware resource, in order to provide user-preferable QoS requirements under resource contention and energy constraints.

It is therefore worthwhile to devise methods that take into account phenomena of quantization, either for the analysis of the effect of quantization of the input signal, or for the synthesis of quantized control laws. Both research topics are currently receiving a growing attention [2–9], especially in the field of hybrid systems because
of the interactions between a continuous dynamical system and a discrete quantized controller.

Among other approaches, receding horizon optimal control ideas were proposed for synthesizing quantized control laws for linear systems with quantized inputs and quadratic optimality criteria. In [10], the authors ensure practical stability properties, by forcing the terminal state to belong to a special invariant set [6], they deal with state constraints, and propose on-line mixed-integer optimization for the implementation of the control law. In the absence of state-constraints, in [11] the authors show that the control law can be equivalently rewritten as a piecewise affine mapping.

Ideas for solving optimal control problems as an explicit function of the state vector were proposed earlier for linear systems [12–15], nonlinear systems [16], hybrid systems [17, 18], and uncertain linear systems [19]. These approaches rely on multiparametric solvers, namely solution algorithms that are able to express the optimizer vector (=the optimal input) as a function of a certain number of parameters (=the current states). The first method for solving multiparametric linear programs dates back to Gal and Nedoma [20]. The book [21] is an excellent reference for properties of generic nonlinear multiparametric problems.

Optimal control problems where all decision variables are quantized and where cost function and constraints depend on a real-valued state vector can be handled by multiparametric integer programming solvers. The first approaches to parametric integer programming were limited to scalar parameters [22], we refer the interested reader to the excellent annotated bibliographic survey [23] for more details.

A multiparametric integer solver for linear objectives and linear constraints was developed in [24, 25]. The algorithm finds the lexicographic minimum of the set of integer points which lie inside a convex polyhedron that depend linearly on one or more integral parameters, and is based on parameterized Gomory’s cuts followed by a parameterized dual simplex method. An alternative method based on a contraction algorithm for multiparametric integer linear programming problems was proposed in [26]. Algorithms for solving a special class of multiparametric nonlinear integer programming problems were investigated in [27].

In this paper we propose a method for solving a quite general class of multiparametric nonlinear integer problems where: (1) the cost function and the constraints depend linearly on a vector of parameters, (2) they depend in an arbitrary nonlinear fashion on the optimization variables, and (3) these are restricted to belong to a finite set. Because of feature (2), the use of relaxation to non-quantized optimization variables and branching, which is the approach of most multiparametric mixed-integer solvers, would be inappropriate here.

The paper is organized as follows. After examining in Section 2 the main theoretical properties of the optimizer and optimum as a function of the parameters, we propose a solver in Section 3. Multiparametric integer programming is used in Section 4 in the context of quantized optimal control. Numerical results are finally reported in Section 5.

---

1 As underlined in [2], the classical concept of stability must be replaced in a quantized context by “practical” stability.
2 Multiparametric Nonlinear Integer Programming

We consider the following multiparametric optimization problem:

\[ V^*(\theta) = \min_{x \in \mathcal{Q}} f_1(x) + f_2(x)\theta \]
\[ \text{s.t.} \quad g_1(x) \leq g_2(x)\theta, \]
\[ \theta \in \Theta \tag{1} \]

where: \( x \in \mathbb{R}^n \) is the optimization vector, which is constrained to belong to the finite set of values \( \mathcal{Q} = \{q_1, \ldots, q_N\} \), \( q_i \in \mathbb{R}^n \), \( \forall i = 1, \ldots, N \); \( \theta \in \mathbb{R}^m \) is a vector of parameters, lying in the polyhedron \( \Theta = \{\theta \in \mathbb{R}^m : T\theta \leq S\} \subseteq \mathbb{R}^m \); \( f_1 : \mathbb{R}^n \mapsto \mathbb{R}, f_2 : \mathbb{R}^n \mapsto \mathbb{R}^m, g_1 : \mathbb{R}^n \mapsto \mathbb{R}^p, g_2 : \mathbb{R}^n \mapsto \mathbb{R}^{m \times p} \) are generic nonlinear functions of the optimization variables.

A typical instance of \( \mathcal{Q} \) is given when each component \( x^{(j)} \) of \( x \) is restricted to a finite set \( \Phi_j = \{\phi_{j1}, \ldots, \phi_{jN_j}\} \), \( j = 1, \ldots, n \), so that \( \mathcal{Q} \) is the Cartesian product \( \Phi_1 \times \ldots \times \Phi_n \), and its cardinality \( N = \prod_{j=1}^n N_j \).

A solution to the multiparametric program (1) is defined as follows:

**Definition 1** The feasible parameter set \( \Theta^* \) is the set of all \( \theta \in \Theta \) for which there is a vector \( x \in \mathcal{Q} \) such that \( g_1(x) \leq g_2(x)\theta \).

**Definition 2** The value function \( V^* : \Theta^* \mapsto \mathbb{R} \) is the function that associates to a parameter vector \( \theta \in \Theta^* \) the corresponding optimum \( V^*(\theta) \) of problem (1).

**Definition 3** The optimizer set function \( X^* : \Theta^* \mapsto 2^\mathcal{Q} \) is the function that associates to a parameter vector \( \theta \in \Theta^* \) the corresponding set of optimizers \( X^*(\theta) = \{x \in \mathcal{Q} : f_1(x) + f_2(x)\theta = V^*(\theta)\} \) of problem (1).

**Definition 4** The optimizer function \( x^* : \Theta^* \mapsto \mathcal{Q} \) is the function that associates to a parameter vector \( \theta \in \Theta^* \) the lexicographic\(^2\) minimum \( x^*(\theta) \) of \( X^*(\theta) \).

The following Lemma 1 and Theorem 1 establish the main properties of the multiparametric solution to problem (1).

**Lemma 1** Consider problem (1) without inequality constraints. Then \( V^* : \Theta \mapsto \mathbb{R} \) is a concave piecewise affine function, and \( x^* : \Theta \mapsto \mathbb{R}^n \) is a piecewise constant function.

**Proof.** In the absence of inequality constraints, \( V^*(\theta) = \min_{i=1,\ldots,N} \{f_1(q_i) + \theta f_2(q_i)\} \) and by Schechter’s result [28] it follows that \( V^* \) is a piecewise affine concave function over a polyhedral partition of \( \Theta \), where the hyperplanes defining the partition have either the form \( (f_2(q_i) - f_2(q_j))\theta \leq f_1(q_j) - f_1(q_i) \) or \( T^{(h)}\theta \leq S^{(h)} \) (\(^{(h)} \) denotes the \( h \)th row or component).

**Example 2.1** For the parametric integer problem

\[ V^*(\theta) = \min_{x \in \{0,1\}} (1 - x)^3 + x\theta \tag{2} \]

\(^2\)The lexicographic order is referred to the order of the elements of \( \mathcal{Q} \). For example, if \( X^*(\theta) = \{q_i, q_j\} \subseteq \mathcal{Q} \) and \( i < j \), then \( x^*(\theta) = q_i \).
\( (n = m = 1, \mathcal{Q} = \{0, 1\}, \Theta = \mathbb{R}) \) we have

\[
V^*(\theta) = \min\{1, \theta\} = \begin{cases} 
\theta & \text{if } \theta \leq 1 \\
1 & \text{if } \theta > 1,
\end{cases}
\]
\[
x^*(\theta) = \begin{cases}
1 & \text{if } \theta \leq 1 \\
0 & \text{if } \theta > 1.
\end{cases}
\]

The value function \( V^* \) is piecewise affine and concave over \( \Theta^* = \mathbb{R} \), and is depicted in Figure 1(a).

Next Theorem 1 establishes the main properties of the multiparametric solution of problem (1) with inequality constraints.

**Theorem 1** Let \( \Theta^* \subseteq \Theta \subseteq \mathbb{R}^m \) be the feasible parameter set of (1), and let \( V^* : \Theta \mapsto \mathbb{R} \), \( x^* : \Theta \mapsto \mathcal{Q} \) the corresponding value function and optimizer function, respectively. Then \( \Theta^* \) is the (possibly nonconvex) union of at most \( N \) convex polyhedra, and \( V^* \), \( x^* \) are piecewise affine and a piecewise constant function, respectively, of the parameters over a partition of \( \Theta^* \) in at most \( 2^N - 1 \) (possibly nonconvex) polyhedra.

**Proof.** For each \( i \in \{1, \ldots, N\} \) the linear inequality constraints \( g_i(q) \theta \geq q_i(q) \) and \( T \theta \leq S \) define a (possibly empty) polyhedron \( P_i \) in \( \mathbb{R}^m \). Then, \( \Theta^* = \bigcup_{i=1}^N P_i \). Consider now the set \( C \) of all combinations of indices \( I = \{i_1, \ldots, i_K\}, i_1 \geq 1, i_K \leq N, K \leq N, i_j < i_{j+1}, \forall j \in \{1, \ldots, K-1\} \), without permutations and repetitions (e.g. for \( N = 3 \) the combinations \{1, 2\}, \{2, 1\}, \{1, 1, 2\}, \{2, 1, 1\}, \{1, 2, 2\}, \{2, 2, 1\} are only taken once as \{1, 2\}). The number of elements of \( C \) is \( \sum_{k=1}^N \binom{N}{k} = 2^N - 1 \). Then, for \( K = 1, \ldots, N \) consider the (possibly nonconvex) polyhedral sets

\[
R_{i_1 \ldots i_K} = \{\theta \in \mathbb{R}^m : \theta \in P_{i_2}, \forall j \in \{i_1, \ldots, i_{K-1}\} \} \quad \text{and} \quad \theta \notin P_h, \forall h \notin \{i_1, \ldots, i_K\}
\]

(for instance, for \( N = 2 \) we have \( R_1 = P_1 \setminus (P_1 \cap P_2), R_2 = P_2 \setminus (P_1 \cap P_2), R_{12} = P_1 \cap P_2 \); another example is reported in Figure 2, where it can be noticed that \( R_1, R_4, R_{14} \) are nonconvex polyhedral sets, and that \( R_1, R_4 \) are also disconnected).

Define \( \bar{C} \subseteq C \) as the subset of indices \( I \) for which \( R_I \) is nonempty (although \( R_I \) may not be full dimensional). As \( \bigcup_{I \in \bar{C}} R_I = \Theta^* \), the sets \( R_I \) define a partition of \( \Theta^* \) into a finite number of (possibly nonconvex) polyhedra.

On each set \( R_I \), we have

\[
V^*(\theta) = \min_{i \in I} \{f_1(q_i) + f_2(q_i) \theta\}, \forall \theta \in R_I, \tag{3}
\]

and by Lemma 1 we conclude that \( V^* \) is a concave piecewise affine function of \( \theta \) over \( R_I \). Hence, \( V^* \) is piecewise affine over \( \Theta^* \). For each given \( \theta \in R_I \) the corresponding optimizer is defined as \( x^*(\theta) = q_j \), where \( j = \min\{i \in I : f_1(q_i) + f_2(q_i) \theta = V^*(\theta)\} \), and where minimization is necessary to obtain the lexicographic minimum in case of multiple optima. \( ^* \)We use here the following definition of nonconvex polyhedral set: A set \( \Omega \subseteq \mathbb{R}^m \) is a nonconvex polyhedral set if \( \Omega \) is nonconvex and \( \Omega = \bigcup_{i=1}^s \Omega_i \), where each set \( \Omega_i \) is a convex polyhedron and \( \Omega_i \cap \Omega_j \) is not full dimensional, \( \forall i, j = 1, \ldots, s, i \neq j \).
The proof of Theorem 1 is based on the enumeration of all possible subsets of $Q$ that are feasible for problem (1), and provides a worst-case upper-bound to the complexity of the multiparametric solution. Fortunately, in general, the number of regions $R_I$ that are useful for characterizing the solution is much smaller, for two reasons. First, emptiness of $R_I$ for several combinations $I$; second, because of optimality considerations: if for some combination $I$ and $j \notin I$ we have $f_1(q_i) + f_2'(q_i)\theta \leq f_1(q_j) + f_2'(q_j)\theta$ for all $i \in I$ and for all $\theta \in R_{I \cup (j)}$, then $R_{I \cup (j)}$ is not needed to characterize the solution ($R_I$ is sufficient). The above feasibility and optimality considerations will be exploited in Section 3 to derive a solution algorithm for problem (1).

Example 2.2 If we add the linear constraint

$$x \leq 2\theta$$

(4)

to problem (2), the solution changes to

$$V^*(\theta) = \begin{cases} 
1 & \text{if } \theta < \frac{1}{2} \text{ or } \theta \geq 1 \\
\theta & \text{if } \frac{1}{2} \leq \theta < 1
\end{cases}$$

(5a)

$$x^*(\theta) = \begin{cases} 
0 & \text{if } \theta < \frac{4}{7} \text{ or } \theta \geq 1 \\
1 & \text{if } \frac{1}{2} \leq \theta < 1.
\end{cases}$$

(5b)

In this case the value function is piecewise affine over $\Theta^* = \{\theta \in \mathbb{R} : \theta \geq 0\}$ and has a discontinuity for $\theta = \frac{1}{2}$, as depicted in Figure 1(b). □

Remark 2.1 If equality constraints of the form $h_1(x) + h_2'(x)\theta = 0$ are considered in problem (1), the set of feasible parameters $\Theta^*$ (or subsets of it) may not be full dimensional. In fact, as the optimizer function $x^*(\theta) \in Q$ can only assume a finite number $N$ of values, equality constraints $h_1(x^*(\theta)) + h_2'(x^*(\theta))\theta = 0$ would force $\theta$ to lie on a finite number of hyperplanes. More precisely, if $x^*(\theta) = q_i$ on some subset $\Theta^*_i \subseteq \Theta^*$, the dimension of $\Theta^*_i$ is $m - \text{rank}(h_2(q_i))$. In particular, when $h_2$ is an $n$-by-$m$ full-rank matrix function on $Q$, $\Theta^*$ reduces to a lattice. Note that, instead, in multiparametric mixed-integer problems the continuous components of the optimizer

Figure 1: Value function for Examples 2.1 and 2.2
Figure 2: Example of a partition of $\Theta^*$ into (possibly nonconvex and disconnected) regions $R_I$, where $R_I$ is the set of all $\theta \in \Theta$ such that $g_1(q_i) \leq g_2'(q_i)\theta$ if and only if $i \in I$.

may vary in a nonconstant fashion with the parameter $\theta$, therefore allowing the satisfaction of equality constraints on full dimensional subsets of $\Theta$. The above considerations have important implications when formulating finite-time optimal control problems with equality constraints on the terminal state, as discussed later in Section 4.

Remark 2.2 The piecewise linearity result for the value function of problem (1), in spite of the general nonlinear form of $f_1$, $f_2$, $g_1$, $g_2$, should not be surprising, as problem (1) can be reformulated as

$$V^*(\theta) \triangleq \min_{i \in \{1, \ldots, N\}} f_{1i} + f_{2i}' \theta$$

s.t. $g_{1i} \leq g_{2i}' \theta$, \hfill (6)

where $f_{1i} \triangleq f_1(q_i)$, $f_{2i} \triangleq f_2(q_i)$, $g_{1i} \triangleq g_1(q_i)$, $g_{2i} \triangleq f_2(q_i)$ become constant data of the problem, for all $i = 1, \ldots, N$. \hfill \Box

3 A Multiparametric Nonlinear Integer Programming Solver

Multiparametric programming solvers have been proposed for several classes of problems: linear [20,29,30], quadratic [12,15], mixed-integer linear (see [17] and references therein). A complete theory for general nonlinear multiparametric programming was developed in [21]. Most of the solvers rely upon the fact that the optimizer is a piecewise affine function of the parameters defined over convex polyhedra. On the other hand, Theorem 1 provides a characterization of the solution over a partition of nonconvex (in general) polyhedra. Although nonconvex polyhedra may be split into several convex components, this approach would largely increase the number of partitions. Moreover, mixed-integer solvers rely on the relaxation of integer constraints, an approach that cannot be followed in our context due to the arbitrary nonlinear dependence on the optimization variables.

Parametric programming solvers especially tailored to problems where all the variables are integer were proposed by several authors, as surveyed in [23], although
most of them deal with scalar parameters. A multiparametric integer solver for linear objectives and linear constraints was developed in [24], which finds the lexicographic minimum of the set of integer points which lie inside a convex polyhedron that depend linearly on one or more integral parameters. An alternative method based on a contraction algorithm for multiparametric integer linear programming problems was proposed in [26]. In [27] the authors present algorithms for solving a special class of multiparametric nonlinear integer programs, where \( f_1(x) = -\sum_{i=1}^{n} f_{1i}(x^{(i)}) \), \( g_{1i}^{(j)}(x) = \sum_{i=1}^{n} g_{1ij}^{(j)}(x^{(i)}) + b \), \( f_1^i \) and \( g_{1ij}^{(j)} \) are non-decreasing functions, \( \forall i = 1, \ldots, n \) and \( \forall j = 1, \ldots, p \), \( f_2(x) \equiv 0 \), \( g_2(x) \equiv G_2 \) is a constant diagonal matrix (i.e., the \( j \)-th component of \( \theta \) only perturbs the \( j \)-th constraint, and therefore \( p = m \)), \( Q = \{ x \in \mathbb{Z}^n : x^- \leq x \leq x^+ \} \), and \( \Theta = [0, 1]^m \).

In this paper we deal with a more general class of multiparametric nonlinear integer problems of the form (1), for which the aforementioned methods are not applicable. A direct application of the ideas used to prove Theorem 1 would lead to fully enumerating all \( 2^N - 1 \) possible combinations of indices \( I \subset C \), test for nonemptiness of \( R_I \), and characterize the value function and the optimizer on \( R_I \) according to (3). We provide here a more efficient solution method.

Before proceeding further, for any set of indices \( I = \{i_1, \ldots, i_K\} \subseteq \{1, \ldots, N\} \), where \( N \) is the cardinality of \( Q \), let \( P_I \equiv \bigcap_{i \in I} P_i \), where \( P_i = \{ \theta \in \Theta : g_1(i) \leq g_2(i) \theta \} \). Note that \( R_I \subseteq P_I \). Moreover, denote by \( V_i : \mathbb{R}^m \to \mathbb{R} \) the linear function that maps \( \theta \) to \( V_i(\theta) = f_1(i) + f_2'(i) \theta \), \( i = 1, \ldots, N \).

The method we propose here is based on two simple considerations. Let \( I = \{i_1 \ldots i_K\} \subseteq \{1, \ldots, N\} \) and \( j \) any index such that \( j \in \{i_K + 1, \ldots, N\} \). The first consideration relates to feasibility: if \( P_I \) is empty, then \( P_{I \cup \{j\}} \) is certainly empty. The second relates to optimality: we can avoid considering a polyhedral region \( P_{I \cup \{j\}} \) if \( V_j(\theta) \geq V_i(\theta) \) for all \( i \in I \) and for all \( \theta \in P_{I \cup \{j\}} \), or if \( P_{I \cup \{j\}} \subset P_{I \cup \{h\}} \) and \( V_j(\theta) \geq V_h(\theta) \) for all \( \theta \in P_{I \cup \{h\}} \).

Based on the above considerations, a recursive algorithm for determining the feasible parameter set \( \Theta^* \), its subpartition, the value function \( V^* \), and the optimizer function \( x^* \), is summarized by Algorithm 3.1.

The algorithm builds an optimality tree \( T \), as depicted in Figure 3, where each node is characterized by a sequence \( I = I_0 \cup \{j\} \) and a polyhedron \( W_{I_0,j} = \{ \theta \in \Theta : g_1(i) \leq g_2(i) \theta, \forall i \in I, V_j(\theta) \leq V_i(\theta), \forall i \in I_0 \} \), where \( I_0 \) is the sequence characterizing the father node.

The root node corresponds to \( I = \emptyset, W_\emptyset = \Theta \). The maximum depth of the tree is \( N = |Q| \). The maximum number of nodes is \( 2^N \). Clearly, \( T \) is always unbalanced by construction: a feasible combination \( \{i_1, i_2, i_3\} \) will be always child of \( \{i_1, i_2\} \) rather than \( \{i_2, i_3\} \); in particular \( \{N\} \) will always be a leaf node.

As the number of nodes in \( T \) depends not only on \( f_1, f_2, g_1, g_2 \), and on the number \( N \) of elements of \( Q \), but also on the order of the elements of \( Q \), at Step 2, the elements \( q_j \) that are infeasible for all \( \theta \in \Theta \) (i.e., \( P_j \) is an empty convex polyhedron) are eliminated, and the remaining ones pre-ordered by increasing values of \( f_1(q_j) \). An alternative is to consider the value \( f_1(q_j) + \min \{ f_2'(q_j) \theta | g_1(q_j) \leq g_2(q_j) \theta \} \) as an ordering criterion, which can be easily computed via linear programming for each feasible element \( q_j \in Q \).

At step 5.2.1. the set \( W_{I_0,j} \) represents the set of all vectors \( \theta \) for which \( q_j \) is feasible, \( q_i \) is feasible for all \( i \in I_0 \), and that have a cost smaller than the cost at the father node (and, by induction, than the cost at all parent nodes). At step 5.2.2.
1. $T \leftarrow \{\text{root\_node}\}$;
2. Remove the elements of $Q$ that are infeasible for all $\theta \in \Theta$ and order the remaining elements by increasing cost $f_k$;
3. Execute examine$(T, \text{root\_node}, \emptyset)$;
4. End.

5. Function examine$(T, \text{node}, I_0)$;
   5.1. If $I_0 \neq \emptyset$ then let $i \leftarrow$ largest element of $I_0$, otherwise let $i \leftarrow 0$;
   5.2. For $j \in \{i + 1, \ldots, N\}$:
      5.2.1. Let $W_{I_0,j} = \{\theta \in P_{I_0} : g_1(q_j) \leq g_2(q_j)\theta, V_j(\theta) \leq V_i(\theta), \forall i \in I_0\}$;
      5.2.2. If $W_{I_0,j} \neq \emptyset$ and the set $\{h : i + 1 \leq h < j, W_{I_0,j} \subset W_{I_0,h}, \text{ and } V_j(\theta) \geq V_h(\theta), \forall \theta \in W_{I_0,j}\}$ is empty or if it is included in $W_{I_0,j}$ already considered so far, and if everywhere on $W_{I_0,j}$ the cost $V_j(\theta)$ is larger than $V_h(\theta)$.
         5.2.2.1 Append child node node$\_j$ to node in $T$;
         5.2.2.2 Execute examine$(T, \text{node}, I_0 \cup \{j\})$;
   5.3. End.

Algorithm 3.1: Multiparametric integer programming solver.

the algorithm determines if a child node must be generated. A node is not generated if $W_{I_0,j}$ is empty or if it is included in $W_{I_0,h}$ for some “brother” node labeled by $I_0 \cup \{h\}$ already considered so far, and if everywhere on $W_{I_0,j}$ the cost $V_j(\theta)$ is larger than $V_h(\theta)$.

After the execution of Algorithm 3.1 and the construction of the tree $T$, the multiparametric solution can be simplified by removing branches from $T$ according to a criterion similar to the one in Step 5.2.2.: for each node node$\_j$ characterized by $I_0 \cup \{j\}$, we can check if there exists a “brother” node node$\_h$, $j < h \leq N$, such that $W_{I_0,j} \subset W_{I_0,h}$ and $V_j(\theta) \geq V_h(\theta), \forall \theta \in W_{I_0,j}$. If this happens, node$\_j$ and its whole sub-tree can be safely removed, without affecting the multiparametric solution.

Remark 3.1 Complexity and suboptimality of the multiparametric solution can be traded off with minor modifications to Algorithm 3.1. In fact, given a suboptimality tolerance $\epsilon \geq 0$, we can modify the optimality requirement in Step 5.2.1. by imposing that $V_j(\theta) \leq V_i(\theta) - \epsilon$, so that a child node is added only if the cost improves at least by $\epsilon$, and, similarly, in Step 5.2.2. by asking that $V_j(\theta) \geq V_h(\theta) - \epsilon$.

3.1 Evaluation of the Solution

The tree structure $T$ constructed by Algorithm 3.1 can be immediately used for storing the multiparametric solution in the form (7), and for evaluating the optimal value and the optimizer for a given $\theta \in \mathbb{R}^m$, as detailed in the recursive Algorithm 3.2.

During the execution of Algorithm 3.2, children nodes must be visited in lexicographic order, namely if $j < h$, the node corresponding to the sequence $I = \{i_1, \ldots, i_k, j\}$ must be visited before the node corresponding to the sequence $I = \{i_1, \ldots, i_k, h\}$. This ordering comes naturally by the way Algorithm 3.1 constructs tree $T$. At Step 2.2., one can avoid evaluating the whole inclusion $\theta \in P_I$. Indeed, only checking $\theta \in P_{i_M}$, where $i_M \triangleq \max(I)$, is enough, as the remaining conditions
\( \theta \in P_i \), by recursion, were already checked for all \( i \in I \setminus \{i_M\} \). Moreover, only the inequalities of \( P_{i_M} \) which are not redundant on \( P_{I \setminus \{i_M\}} \) need to be evaluated, which allows one to save memory space and computation time. In view of the above considerations, and since \( q_{i_M} \) is always the optimizer, only \( i_M = \max(I) \) needs to be stored in the node, rather than the whole sequence \( I \). Moreover, all the constraints defining \( P_{i_M} \) belong to a finite constraint store, so that rather than memorizing copies of the same constraints one can memorize pointers to entries of such a constraint store.

**Remark 3.2** Multiparametric linear, quadratic, and mixed-integer linear solvers partition the set of feasible parameters \( \Theta^* \) into the union of convex polyhedral sets with nonoverlapping interiors \([12,13,15,17,20]\). A potential drawback of this type of representation is that when the number of polyhedra is large the evaluation of the solution may be computationally expensive, because to determine which polyhedron contains a given parameter vector \( \theta \) may require (in the worst case) evaluating the defining inequalities of all polyhedra. A useful approach to overcome this problem was taken in \([31]\), where the authors suggested to organize a given polyhedral partition on a search tree, so that the amount of computation for evaluating the solution is on average logarithmic in the number of polyhedra.

Here we have taken a different approach by expressing the solution as a multi-level conditional expression (i.e., as a tree of nested conditionals), similarly to what is done in \([25]\) for multiparametric integer linear programming\(^4\). In fact, the multi-

---

\(^4\)In [25] the authors denominate a multi-level conditional expression a *quast*. 

---

Figure 3: Optimality tree \( \mathcal{F} \), related to the partition depicted in Figure 2.
1. \([V^*(\theta), x^*(\theta)] \leftarrow \text{eval}(T, \text{root node}, \theta)\);

2. Function \([V^*, x^*] \leftarrow \text{eval}(T, \text{node}, \theta)\);
   2.1. Let \(V^* \leftarrow +\infty, x^* \leftarrow \emptyset\);
   2.2. If \(\theta \in P\):
      2.2.1. Let \(I \leftarrow \text{combination associated with node}\);
      2.2.2. Let \(c \leftarrow \text{number of children of node} \); Let \(i \leftarrow 0\);
      2.2.3. While \(i < c \) and \(V^* = +\infty\):
         2.2.3.1 \(i \leftarrow i + 1\);
         2.2.3.2 Let \(\text{node}_i \leftarrow i\text{-th child of node}\);
         2.2.3.3 \([V^*, x^*] \leftarrow \text{eval}(T, \text{node}_i, \theta)\);
      2.2.4. If \(V^* = +\infty\) and \(I \neq \emptyset\):
         2.2.4.1 Let \(i^* \leftarrow \text{largest element of I}\);
         2.2.4.2 Let \(x^* \leftarrow q_{i^*}, V^* \leftarrow f_1(q_{i^*}) + f_2(q_{i^*})\theta\);
   2.3. Return \([V^*, x^*]\);
   2.4. End.

Algorithm 3.2: Evaluation of the optimal value \(V^*(\theta)\) and of the lexicographic minimum \(x^*(\theta)\)

parametric solution can be written as:

\[
\text{if } \theta \in \Theta \text{ then} \\
\quad \text{if } H_1\theta \leq K_1 \text{ then} \\
\quad \quad \ldots \\
\quad \quad \text{if } H_1\theta \leq K_1 \text{ then} \\
\quad \quad \quad \ldots \\
\quad \quad \text{elseif } H_k\theta \leq K_k \text{ then} \\
\quad \quad \quad \ldots \\
\quad \quad \quad \text{else} \\
\quad \quad \quad \quad \text{problem is infeasible} \\
\quad \quad \text{else} \\
\quad \quad \quad \quad \text{solution is undefined (} \theta \notin \Theta \text{)} \\
\quad \text{end} \\
\text{end}
\]

where \(H_i, K_i\), are (possibly empty) matrices/vectors of suitable dimensions. \(\Box\)

**Example 3.1** The solution reported in (5) can be obtained by running Algorithm 3.1 for \(\Theta = \{\theta : \|\theta\|_{\infty} \leq 10\}\), with \(Q = \{1, 0\}\) (the elements \(q_1, q_2\) of \(Q\) are ordered by
increasing cost \( f_1(q_i) \), and arranged as follows:

\[
\begin{align*}
\text{if } \left[ \frac{1}{10} \right] \leq \theta \leq \left[ \frac{10}{10} \right] & \text{ then return} \\
\text{if } -\theta \leq -1 & \text{ then return} \\
& \quad \text{if } -2\theta \leq -1 \\
& \quad \quad \text{then } x^*(\theta) = q_2, V^*(\theta) = (1 - q_2)^3 + q_2\theta, \text{ where } q_2 = 0 \\
& \quad \quad \text{else } x^*(\theta) = q_1, V^*(\theta) = (1 - q_1)^3 + q_1\theta, \text{ where } q_1 = 1 \\
& \quad \text{end} \\
& \quad \text{elseif } -2\theta \leq 0 \text{ then return} \\
& \quad \quad \text{then } x^*(\theta) = q_2, V^*(\theta) = (1 - q_2)^3 + q_2\theta, \text{ where } q_2 = 0 \\
& \quad \quad \text{else } \text{ problem is infeasible } \\
& \quad \text{end} \\
& \quad \text{else } \text{ solution is undefined (because } \|\theta\|_\infty > 10) \\
& \text{end}
\end{align*}
\]

Note that (8) is not a minimal multi-level conditional expression. Determining ways for ensuring the minimality of the multilevel conditional solution is a topic that remains to be investigated.

\[\square\]

4 Explicit Quantized Optimal Control

Consider the following linear discrete time invariant system

\[
x(t+1) = Ax(t) + Bu(t)
\]

where \( x \in \mathbb{R}^n, u \in \mathcal{U} \triangleq \{ \bar{u}_1, \bar{u}_2, \ldots, \bar{u}_L \}, \bar{u}_i \in \mathbb{R}^n \) are the levels of quantization, and \((A, B)\) is a stabilizable pair. Starting from the initial state \( x(0) \), we wish to control the final state \( x(T) \) to a target set \( \Omega \) while satisfying the constraints

\[
\bar{A}x(t) + \bar{B}u(t) \leq \bar{C}, \ t=0, \ldots, T-1.
\]

Constraints (10) are generic linear constraints on input and state variables. A typical instance are box constraints of the form \( x_{\min} \leq x_k \leq x_{\max} \) (constraints of the form \( u_{\min} \leq u_k \leq u_{\max} \) can be immediately taken into account by simply excluding from \( \mathcal{U} \) those values \( \bar{u}_i \) outside the bounds). We assume that the set \( \Omega \) is a full-dimensional polyhedral terminal set for the state vector, as in case of non full-dimensional sets \( \Omega \), the set \( \Theta \) of initial states \( x(0) \) for which (10) are feasible may be lower-dimensional, for instance if \( \Omega = \{0\} \), corresponding to the constraint \( x(T) = 0 \), \( \Theta \) would be a lattice. In the following subsections we show how the multiparametric integer solver developed earlier can be used to derive explicit optimal control laws based on the minimization of a quadratic or linear performance index.
4.1 Quadratic Quantized Optimal Control

Consider the following optimal control problem:

\[
\min_U \left \{ J(U, \theta) = x_T^P x_T + \sum_{k=0}^{T-1} (x_k^T Q x_k + u_k^T R u_k) \right \} \tag{11a}
\]

subject to

\[
\begin{align*}
  x_0 &= \theta \\
  x_{k+1} &= A x_k + B u_k, \quad k = 0, \ldots, T - 1, \\
  A x_k + B u_k &\leq C, \quad k = 0, \ldots, T - 1, \\
  x_T &\in \Omega \\
  u_k &\in \mathcal{U} \triangleq \{ \bar{u}_1, \ldots, \bar{u}_L \}
\end{align*} \tag{11b}
\]

where \( R = R' > 0, \ Q = Q' \geq 0, \ P \geq 0 \) are matrices of suitable dimensions, \( \theta \) represents a generic initial condition, \( U \triangleq [u_0', u_1', \ldots, u_{T-1}'] \) is the set of free control moves, \( U \in \mathcal{Q}, \) where \( \mathcal{Q} \triangleq \mathcal{U}^T = U \times \cdots \times U \), and \( U^*(\theta) \triangleq [u_0^*, u_1^*, \ldots, u_{T-1}^*](\theta)' \) is the optimizer (or, in case of multiple optima, the lexicographic minimum of the set of optimizers).

It is immediate to cast problem (11) as an integer quadratic program (IQP). Indeed, by substituting \( x_k = A^k x(t) + \sum_{j=0}^{k-1} A^j B u_{k-j}, \) Eq. (11a) can be rewritten as

\[
\min_U \left \{ \frac{1}{2} U' H U + U' F' \theta + \frac{1}{2} \theta' Y \theta \right \} \tag{12}
\]

subject to \( GU \leq W + E \theta \)

where the column vector \( U \triangleq [u_0', u_1', \ldots, u_{T-1}'] \) is the optimization vector, \( H = H' > 0, \) and \( H, \ F, \ Y, \ G, \ W, \ E \) are easily obtained from \( Q, \ R, \) and (11a).

The optimization problem (11) is an IQP which depends on the initial state \( \theta. \) The multiparametric nonlinear integer programming algorithm developed earlier can be conveniently used to compute the piecewise constant solution \( U^*(x_0) \) to the optimal control problem (11). In fact, after taking apart the quadratic term \( \frac{1}{2} \theta' Y \theta \) that does not affect the optimizer \( U^*(\theta) \), problem (12) can be recast in the form (1) by setting \( f_1(U) = \frac{1}{2} U' H U, \ f_2(U) = F U, \ g_1(U) = GU - W, \ g_2(U) = E'. \)

The following result immediately follows by Theorem 1.

**Corollary 1** Consider the optimal control problem (11), parameterized by the initial condition \( x_0 = \theta. \) Then

(i) The set of parameters \( \Theta^* \subseteq \mathbb{R}^{n*} \) for which a solution to (11) exists is the union of at most \( L^T \) convex polyhedra.

(ii) The value function \( V^* : \mathbb{R}^{n*} \subseteq \mathbb{R} \) is a piecewise quadratic function of \( x_0 \) (more exactly, the sum of a convex quadratic and a piecewise affine function) over a partition of \( \Theta^* \) in at most \( 2^{L^T} - 1 \) (possibly nonconvex) polyhedra.

(iii) The optimizer function \( U^* : \mathbb{R}^{n*} \subseteq \mathcal{U}^T \) is a piecewise constant function of \( x_0 \) defined over the same partition of \( \Theta^*. \)

Moreover, in the absence of inequality constraints \( Ax_k + Bu_k \leq C, k = 0, \ldots, T - 1, \) and \( x_T \in \Omega, \) \( V^* \) is the sum of a convex quadratic and a piecewise affine concave function of \( x_0. \)
4.2 Linear Quantized Optimal Control

Consider the following optimal control problem:

\[
\min_U \left\{ J(U, \theta) = \| Px_T \|_\infty + \sum_{k=0}^{T-1} (\| Q x_k \|_\infty + \| R u_k \|_\infty) \right\} \tag{13a}
\]

s.t. \( (11b) \)  \( (13b) \)

where \( R, Q, P \) are full-rank matrices with \( n_R, n_Q, n_P \) rows, respectively, and a suitable number of columns, and \( \theta, U, Q, \) and \( U^*(\theta) \) are defined as above.

Similarly to [17], problem (13) can be cast as an integer linear program (ILP). To this end, we introduce the following constraints

\[
\begin{align*}
\epsilon^u_t & \geq \pm R^{(j)} u_t, \quad \forall j = 1, \ldots, n_R, \forall t = 0, \ldots, T - 1 \\
\epsilon^x_t & \geq \pm Q^{(j)} x_t, \quad \forall j = 1, \ldots, n_Q, \forall t = 0, \ldots, T - 1 \\
\epsilon^x_T & \geq \pm P^{(j)} x_T, \quad \forall j = 1, \ldots, n_P,
\end{align*}
\]

where \{j\} denotes the jth row. By substituting again \( x_k = A^k x(t) + \sum_{j=0}^{k-1} A^j B u_{k-1-j} \), and by letting \( E = [\epsilon^u_0 \epsilon^u_1 \ldots \epsilon^u_{T-1-1} \epsilon^x_T] \), problem (13) can be rewritten as

\[
\min_{U, E} \left\{ \epsilon^x_T + \sum_{t=1}^{T-1} \epsilon^x_t + \epsilon^u_t \right\} \tag{14}
\]

\[
\text{subj. to } G [\frac{U}{E}] \leq W + E \theta \\
E \geq 0 \\
[\frac{U}{E}] \in Q \times Q_E,
\]

where \( Q_E \subset \mathbb{R}^{2T+1} \) is also a finite set, as, at optimality, for each optimal component of \( E \) at least one of the constraints (14) is active, so that also the components of \( E \) are indeed quantized. Therefore, the optimization problem (13) is an ILP which depends on the initial state \( \theta \), and the multiparametric integer programming algorithm developed earlier may be used to compute the explicit piecewise constant solution \( [\frac{U}{E}]^* (x_0) \). A corollary of Theorem 1 similar to Corollary 1 may be easily stated.

On the other hand, the approaches of [26] and of [24], that are specialized for multiparametric integer linear problems may be more suitable here. Alternatively, since continuous relaxations of (15) are multiparametric linear programs, it may be more convenient to treat \( E \) as continuous variables and solve (15) by using multiparametric mixed-integer linear solvers, as done in [17] for the generic case of hybrid systems with continuous and discrete inputs.

4.3 Explicit Quantized Receding Horizon Control

A useful way for transforming the \( U^*(\theta) \) into a closed-loop control law is to adopt the so called receding horizon philosophy. The receding horizon controller is defined as

\[
u(t) = u^*_0(x(t)), \tag{16}
\]

where \( u^*_0(x(t)) \) is the first element of the minimizer \( U^*(x(t)) \) of the finite-time quantized optimal control problem, initialized at the current state \( \theta = x(t) \).

An immediate corollary of Corollary 1 is that the control law (16) is a piecewise constant law defined over a polyhedral partition. Criteria for selecting the terminal
set \( \Omega \) in order to guarantee practical stability properties of the quantized control law (16) were analyzed in [10].

**Remark 4.1** As only the first part \( u_0^*(x(t)) \) of the minimizer \( U^*(x(t)) \) is of interest, after the execution of Algorithm 3.1 the multiparametric solution can be simplified by removing subtrees of \( T \) where the first optimal move \( u_1^* \) is the same in all nodes (in depth search of such subtrees would just serve to determine \( u_1^*, \ldots, u_{N-1}^* \)). □

### 4.4 Logic-Based Constraints

A *Boolean expression* is inductively defined by the grammar

\[
\phi ::= X | \neg \phi_1 \lor \phi_2 | \neg \phi_1 \lor \phi_2 | \phi_1 \land \phi_2 | \phi_1 \leftarrow \phi_2 | \phi_1 \rightarrow \phi_2 | \phi_1 \leftrightarrow \phi_2 | (\phi_1),
\]

where \( X \in \{0, 1\} \) is a Boolean variable, and the logic operators \( \neg \) (not), \( \lor \) (or), \( \land \) (and), \( \leftarrow \) (implied by), \( \rightarrow \) (implies), \( \leftrightarrow \) (iff) have the usual semantics. Every Boolean expression can be rewritten in *conjunctive normal form* (CNF), which is defined by the following grammar:

\[
\phi ::= \psi | \phi \land \psi, \quad \psi ::= \psi_1 \lor \psi_2 | \neg X | X.
\]

By generalizing results of [32–34], in [35] the authors illustrated techniques for equivalently expressing arbitrary Boolean functions by a set of linear inequalities over 0-1 variables. In particular, by first converting a Boolean formula into CNF (a task that can be performed automatically by using one of the several available techniques), by letting the CNF have the form

\[
\bigwedge_{j=1}^{m} \left( \bigvee_{i \in P_j} X_i \bigvee_{i \in N_j} \neg X_i \right),
\]

\( N_j, P_j \subseteq \{1, \ldots, \ell\}, \forall j = 1, \ldots, m \), the corresponding set of integer linear inequalities is

\[
\begin{align*}
1 & \leq \sum_{i \in P_1} X_i + \sum_{i \in N_1} (1 - X_i), \\
& \vdots \\
1 & \leq \sum_{i \in P_m} X_i + \sum_{i \in N_m} (1 - X_i),
\end{align*}
\]

that define a polyhedron \( P_{\text{CNF}} \subset \mathbb{R}^{\ell} \).

As a consequence, Boolean constraints involving 0-1 variables \( x = [X_1 \ldots X_n] \) and 0-1 parameters \( \theta = [X_{n+1} \ldots X_{n+m}], \ell = n + m \), having the form

\[
g_B(x, \theta) = 0
\]

where \( g_B \) is an arbitrary Boolean function, can be translated into linear inequalities

\[
A_g x \leq c_g + B_g \theta.
\]

Clearly, (22) fits the general multiparametric integer programming framework (1) with \( Q = \{0, 1\}^n \), and \( \Theta = \{0, 1\}^m \subset [0, 1]^m \).
Table 1: Computational experience for Problem (23): number N of elements of \( Q \), number of nodes in \( T \) generated by Algorithm 3.1 and required CPU time, CPU time for evaluating the solution using Algorithm 3.2 and using enumeration (CPU times are averaged over 1089 values of \( \theta \) uniformly distributed over \( \Theta \)).

<table>
<thead>
<tr>
<th>( N )</th>
<th>25</th>
<th>36</th>
<th>49</th>
<th>64</th>
<th>81</th>
<th>100</th>
<th>121</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nodes in ( T )</td>
<td>14</td>
<td>14</td>
<td>20</td>
<td>23</td>
<td>30</td>
<td>32</td>
<td>38</td>
</tr>
<tr>
<td>Algorithm 3.1</td>
<td>1.42 s</td>
<td>2.06 s</td>
<td>3.15 s</td>
<td>5.11 s</td>
<td>7.93 s</td>
<td>10.22 s</td>
<td>13.79 s</td>
</tr>
<tr>
<td>Algorithm 3.2</td>
<td>2.13 ms</td>
<td>2.37 ms</td>
<td>2.45 ms</td>
<td>2.89 ms</td>
<td>3.40 ms</td>
<td>3.78 ms</td>
<td>4.23 ms</td>
</tr>
<tr>
<td>Enumeration</td>
<td>14.27 ms</td>
<td>20.05 ms</td>
<td>26.98 ms</td>
<td>35.17 ms</td>
<td>44.63 ms</td>
<td>59.94 ms</td>
<td>65.65 ms</td>
</tr>
</tbody>
</table>

5 Examples

Example 5.1 Consider the multiparametric integer problem

\[
V^*(\theta) \triangleq \min_{x \in Q} x_1^3 + |x_2|
\]

\[
\begin{bmatrix}
1 & 0 \\
0.3090|x_2| & 0.9511|x_1| \\
-0.8090 & 0.5878 \\
-0.8090|\sin(x_1/5)| & -0.5878|x_2| \\
0.3090 & -0.9511 \\
6.7506 & 2.6210 \\
3.1557 & -0.4353 \\
5.0342 & -4.8150 \\
4.4299 & 3.2141 \\
6.4565 & -0.5530 \\
0 & 0
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2
\end{bmatrix}
\leq
\begin{bmatrix}
2.6210 & 1.1543 \\
-0.4353 & 2.9194 \\
-4.8150 & 4.2181 \\
3.2141 & 2.3821 \\
-0.5530 & -3.2373 \\
0 & 0
\end{bmatrix}
\begin{bmatrix}
1 - \frac{1}{2}\sqrt{x_1^2 + x_2^2} \\
\theta_1 \\
\theta_2
\end{bmatrix}
\]

where \( Q \) is obtained by gridding the square \([0, 2] \times [0, 2]\) with a grid-step of \( \frac{2}{m} \), \( m = 4, 5, \ldots, 10 \), leading to \( N = (m + 1)^2 \) elements in \( Q \). In Table 1 we report the computation time required by Algorithms 3.1 and 3.2, and compare the latter with the time required for computing the optimal solution by enumeration. In Figure 4 we show the value function when the grid step is 0.2 (\( Q \) contains \( N = 121 \) elements).

Example 5.2 Consider the following optimal reliability design problem proposed in [27, Example 4]:

\[
\begin{align*}
\max & \quad z = (1 - (1 - 0.6)x^{(1)})(1 - (1 - 0.9)x^{(2)})(1 - (1 - 0.55)x^{(3)})(1 - (1 - 0.75)x^{(4)}) \\
\text{s.t.} & \quad \begin{bmatrix} 6.2 & 3.8 & 6.5 & 5.3 \\ 9.5 & 5.5 & 3.8 & 4.0 \end{bmatrix} x \leq \begin{bmatrix} 50 + 15\theta^{(1)} \\ 50 + 20\theta^{(2)} \end{bmatrix} \\
& \quad x \in Q = \{x \in \mathbb{Z}^4 : 1 \leq x \leq \bar{x}\} \\
& \quad \Theta = \{\theta \in \mathbb{R}^2 : 0 \leq \theta \leq 1\}.
\end{align*}
\]

In (24) we have added the upper bound \( \bar{x} \), obtained via linear programming by maximizing \( x^{(j)} \) with respect to \((x, \theta)\) subject to the linear inequality constraints
in (24) and to $\theta \in \Theta$, for $j = 1, 2, 3, 4$, and by rounding off the result to the nearest smaller or equal integer, which provides $\bar{x} = [5 9 7 9]^T$.

The set $Q$ contains 2835 elements, of which 2484 are infeasible. Algorithm 3.1 is executed in 24.95 s, and provides the solution depicted in Figure 5. The solution coincides with the one reported in [27, Figure 3]. The corresponding optimality tree consists of 18 nodes, where all nodes are leaf nodes, except the root node. □

**Example 5.3** Consider an extremely simplified version of the problem of landing a spacecraft on a planet, where we consider only the vertical motion described by the equations

$$\begin{align*}
\dot{h} &= v, \\
\dot{v} &= -\beta v + u
\end{align*}$$

(25)

where $h$ is the height from ground, $v$ the vertical velocity, and the overall force $u$ acting on the spacecraft is given by

$$u = \begin{cases} 
-mg & \text{thruster off} \\
0 & \text{thruster on (gravity compensation)} \\
mg & \text{double thruster on}
\end{cases}$$

(26)

By choosing the parameters $\beta = 1$, $m = 1$, $g = 1$ (units are omitted here, as the parameters have no particular meaning in this example), and by discretizing the dynamics with a sampling time $T_s = 1$, we obtain the discrete-time linear model

$$x(t+1) = \begin{bmatrix} 1 & 0.6321 \\ 0 & 0.3679 \end{bmatrix} x(t) + \begin{bmatrix} 0.7358 \\ 1.2642 \end{bmatrix} u(t),$$

(27)

where $u(t) \in \mathcal{U} \triangleq \{-1, 0, 1\}$, and $x = \begin{bmatrix} h \\ v \end{bmatrix}$. We wish to design a controller that brings the height of the spacecraft and its velocity to zero while satisfying the constraints

$$\begin{align*}
h &\geq 0 \\
v &\geq -\bar{v},
\end{align*}$$

(28)
where $\bar{v} = 1.5$. To this end, we consider the finite-time optimal control problem

$$\min_{u_0, u_1} \quad x_2'Px_2 + \sum_{k=0}^{1} (x_k'Qx_k + u_k'Ru_k)$$

s.t.

$$x_1 \geq \begin{bmatrix} 0 \\ -\bar{v} \end{bmatrix},$$

$$u_0, u_1 \in \{-1, 0, 1\},$$

where $R = 10$, $Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, and $P \approx \begin{bmatrix} 3.1240 & 1.5677 \\ 1.5677 & 3.3241 \end{bmatrix}$ solves the Riccati equation $P = (A + BK_{LQ})'P(A + BK_{LQ}) + K_{LQ}RK_{LQ} + Q$, $K_{LQ} = -(R + B'P B)^{-1}B'PA$.

The mp-IQP problem associated with the optimal control law (29) has the form (12) with $\theta = x_0$ and

$$H = \begin{bmatrix} 0.7675 & 0.2924 \\ 0.2924 & 0.6323 \end{bmatrix}, \quad F = \begin{bmatrix} 0.2160 & 0.2132 \\ 0.1477 & 0.1468 \end{bmatrix},$$

$$G = \begin{bmatrix} -0.7358 & 0 \\ -1.2642 & 0 \\ 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \end{bmatrix}, \quad W = \begin{bmatrix} 0 \\ 1.5 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \quad E = \begin{bmatrix} 1 & 0 & 0.6321 \\ 0 & 0 & 0.3679 \end{bmatrix},$$

where we have neglected the constant term $\frac{1}{2} \theta'Y \theta$. By running Algorithm 3.1 on $\Theta = \{\theta : \|\theta\|_{\infty} \leq 15\}$, the multiparametric solution is computed in 0.85 s on a Pentium III 800 Mhz running Matlab 5.3, and the associated tree $T$ consists of 24 nodes and has a depth of 5 levels, as depicted in Figure 6. The number of inequalities associated with each node varies between one and four. An evaluation of the value function $V^*$ takes an average of 1.36 ms (this value is obtained by averaging over...
Figure 6: Optimality tree associated with the optimal control problem (29). For each node is reported the number of linear inequalities that must be checked at that node during the on-line evaluation of the solution for a given $x_0$.

a grid of 4225 samples of $\Theta$), against about 6.01 ms needed to compute $V^*$ by enumeration. Even from this simple problem where the number of elements of $Q$ is only $N = 9$, it is clear the advantage of having an explicit representation of $V^*$.

We compare now the solution $U^*(\theta)$ of the integer quadratic problem with the quantization $\hat{U}(\theta)$ to the nearest (in Euclidean norm) feasible point in $Q$ of the solution $U^*_{QP}(\theta)$. The partition associated with $U^*_{QP}(\theta)$, obtained in 0.22 s using the algorithm reported in [15], is depicted in Figure 7(b), while the partition associated with $U^*(\theta)$ is depicted in Figure 7(a). In Figure 8, we report the difference $\hat{V}(\theta) - V^*(\theta)$, where $\hat{V}(\theta) = \frac{1}{2} \hat{U}'(\theta) H \hat{U}(\theta) + \theta' F' \hat{U}(\theta)$, and $V^*(\theta)$ is the optimal value function for the integer quadratic program; clearly $V^*(\theta) \leq \hat{V}(\theta)$, for all $\theta \in \Theta^*$.

By implementing the multiparametric solution in a receding horizon fashion, we obtain the trajectories plotted in Figure 9, that show the closed-loop behavior of the system for the initial condition $x(0) = [10^T]$.

6 Conclusions

For multiparametric nonlinear integer problems where the cost function and the constraints depend in an arbitrary nonlinear fashion on the optimization variables and in a linear fashion on the parameters, and where the optimization variables only belong to a finite set, we have characterized the main theoretical properties of the solution and proposed a solution algorithm. The methodology was employed to investigate properties of quantized optimal control laws and to obtain their explicit
representation as a function of the state vector. A potential benefit to the presented methodology may Techniques based on the integration of multiparametric and dynamic programming are currently under investigation for solving quantized optimal control problems.

An interesting topic for further research is the problem of obtaining minimal representations of the multiparametric solution, and the application of the multiparametric nonlinear integer programming algorithm to other classes of quantized optimal control problems.

7 Acknowledgements

This work was partially supported by the European Project “Computation and Control”. We thank Eric Triki for pointing out a few typos in the original manuscript.

References


Figure 8: Difference $\hat{V}(\theta) - V^*(\theta)$, where $\hat{V}(\theta) = \frac{1}{2} \hat{U}'(\theta)H\hat{U}(\theta) + \theta'F'\hat{U}(\theta)$, and $\hat{U}(\theta)$ is obtained by quantizing the solution of the continuous quadratic program to the nearest feasible point in $Q$.


Figure 9: Receding horizon optimal control with quantized inputs


