15th Triennial World Congress, Barcelona, Spain

A GEOMETRIC APPROACH TO REACHABILITY COMPUTATIONS FOR CONSTRAINED DISCRETE-TIME SYSTEMS

Eric C Kerrigan, John Lygeros and Jan M Maciejowski

Department of Engineering, University of Cambridge
Trumpington Street, Cambridge CB2 1PZ, United Kingdom
Tel: +44-1223-339706/2770/2732, Fax: +44-1223-332662
email: {eck21,jl290,jmm}@eng.cam.ac.uk
URL: http://www-control.eng.cam.ac.uk

Abstract: The problem of reachability computations for systems with simultaneous constraints on the states, control inputs and disturbances is treated. A problem formulation is provided for the general case and, for the special case where the disturbance constraints are independent of the states and inputs, it is discussed how the solution can be approached using standard geometric concepts. It is also discussed how the procedure can be implemented for a class of piecewise affine systems with polyhedral constraints using computational geometry software.

Keywords: Nonlinear systems, uncertain dynamic systems, controllability, reachability, invariance, constraint satisfaction, piecewise linear, robust control.

1. INTRODUCTION

The problems of reachability, invariance and controlled invariance for discrete time systems have been extensively studied in the literature for over three decades. Most recently these problems have attracted renewed attention, partly because improvements in computational capabilities have made it possible to implement the algorithms for systems of practical interest. Another reason for the renewed interest in these problems is the emergence of new classes of practically important systems, such as hybrid systems. These are systems whose states, inputs and outputs can take on values from both a countable set (e.g. the set of integers) as well as an uncountable set (e.g. the set of real numbers). In recent years, invariance and reachability problems for classes of hybrid systems have been studied by a number of authors (Berardi *et al.*, 1999; Koutsoukos and Antsaklis, 1999; Torrisi and Bemporad, 2001; Habets and van Schuppen, 2001; Vidal *et al.*, 2001).

One class of systems which, to the authors' knowledge, has received relatively little attention are systems with mixed constraints on the inputs and states (or equivalently, systems whose input constraints depend on the state). Often, when this class of systems is treated, it is with an insufficient amount of detail and overly-conservative approximations (and mistakes) are made. Systems with mixed state and input constraints may arise in practice for a number of reasons:

(1) When modelling systems with physical constraints. Here the model must reflect the fact that the constraints will be satisfied by all evolutions of the system, whatever the inputs.
(2) When designing controllers to meet safety or performance specifications, i.e. to ensure that the state of the system remains in a certain region of the state space. Safety and performance specifications may be violated if the inputs are not chosen properly. One can show (see e.g. Vidal *et al.* (2001)) that the least restrictive controller that satisfies such specifications involves state-dependent input constraints.

A simple example illustrates the point. Consider the following model for the longitudinal motion of a car on a highway:

$$\begin{bmatrix} x_{k+1} \\ v_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_k \\ v_k \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_k + \begin{bmatrix} 0 \\ 1 \end{bmatrix} w_k$$

where $x \in \mathbb{R}$ represents the position of the car, $v \in \mathbb{R}$ its velocity, $u \in [\underline{u}, \overline{u}]$ represents the control acceleration applied by the engine or brakes, and $w \in [\underline{w}, \overline{w}]$ a disturbance acceleration due to wind. It is assumed that $\underline{u} < 0 < \overline{u}$ and $\underline{w} < \overline{w}$. For simplicity all other constants have been normalised to 1.

One would like to capture the situation where the vehicle is prevented from going backwards. This is a reasonable requirement in many cases (e.g. on a highway) and is very easy to implement in practice (assuming that the wind is incapable of pushing the car backwards when the brakes are applied one could simply disallow the reverse gear). This can be captured by the hard state constraint $v \geq 0$. To enforce this constraint, the model needs to incorporate the additional state-dependent constraint

$$v + u + w \geq 0 \tag{1}$$

on the inputs (control and disturbance).

More generally, consider state variables x, control variables u and disturbance variables w, taking values in the sets X, U and W respectively (not necessarily vector spaces). Consider dynamic constraints on these variables of the form

$$x_{k+1} = f(x_k, u_k, w_k) \text{ and } (x_k, u_k, w_k) \in P \ , \tag{2}$$

where $P \subseteq X \times U \times W$ and $f : P \to X$. Here P is assumed to capture the physical, state-dependent constraints on the control and disturbance inputs. The goal is to develop methods for designing controllers for this class of dynamical systems. Even though this class has been considered by some authors (e.g. Torrisi and Bemporad (2001)) to our knowledge none of the control and analysis algorithms in the literature are capable of explicitly dealing with it.

In this paper, a reachability controller synthesis problem is abstractly formulated for this class of systems. It is then shown how standard geometric tools can address the problem for the special case where the disturbance constraints are independent of the state and control. Finally, it is discussed how computational tools for manipulating polyhedra can be used to implement the solution for a class of piecewise affine systems with polyhedral constraints.

2. THE REACHABILITY PROBLEM

A *run* of system (2) is defined as an infinite sequence $\{(x_k, u_k, w_k)\}_{k=0}^{\infty}$ such that for all $k \in \mathbb{N}$, $x_{k+1} = f(x_k, u_k, w_k)$ and $(x_k, u_k, w_k) \in P$. Let $\Pi_X(\cdot)$ denote the projection of a set onto X, i.e.

$$\Pi_X(P) := \{x \in X \,|\, \exists (u, w) : (x, u, w) \in P\} \, .$$

In addition, it is assumed that for all $x \in \Pi_X(P)$ there exists a pair (u, w) satisfying $(x, u, w) \in P$ such that $f(x, u, w) \in \Pi_X(P)$. Under this assumption it is easy to show that runs of the system exist for all $x_0 \in \Pi_X(P)$. A system which satisfies this assumption is called *non-blocking*.

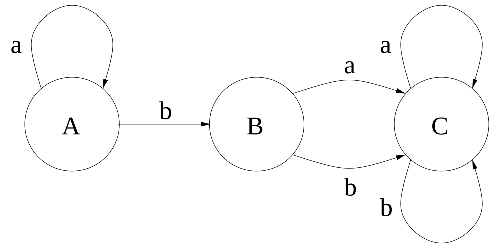

Fig. 1. A discrete state reachability example.

Time-varying, set-valued state feedback controllers will be considered in this section, i.e. maps of the form

$$\mu : X \times \mathbb{N} \to 2^U \ ,$$

where 2^U denotes the set of subsets of U. A run of the closed-loop system obtained by connecting such a controller to the system is again an infinite sequence $\{(x_k, u_k, w_k)\}_{k=0}^{\infty}$ that for all $k \in \mathbb{N}$ satisfies

(1) $x_{k+1} = f(x_k, u_k, w_k)$ and $(x_k, u_k, w_k) \in P$, i.e. is a run of the open-loop system; and,
(2) $u_k \in \mu(x_k, k)$, i.e. satisfies the controller constraints.

Notice that the closed-loop system is another system of the form (2) (possibly after appending k to the state).

The reachability problem discussed here involves a given constraint set $D \subseteq X \times U \times W$, an integer $N \in \mathbb{N}$ and a target set $T_N \subseteq X$. The goal is to determine the set of states T_0 for which there exists a controller μ such that

- the closed-loop system is non-blocking; and
- for all runs, $\{(x_k, u_k, w_k)\}_{k=0}^{\infty}$, of the closed-loop system with $x_0 \in T_0$, $(x_k, u_k, w_k) \in D$ for $k \in \{0, 1, \ldots, N-1\}$ and $x_N \in T_N$.

Notice that this problem contains the N-step reachability and the N-step invariance problems as special cases (Kerrigan, 2000). In principle, T_N could be a single point, though for systems with states, inputs and disturbances taking values from an uncountable set, it is unlikely that a controller exists that ensures such a terminal constraint is reached for all disturbance sequences.

It should be noted that a time-varying feedback controller is in general necessary to provide a meaningful solution to this problem. As usual in the presence of disturbances, restricting attention to open-loop controllers would lead to extremely conservative results (Maciejowski, 2001, Sect. 8.5.2). Moreover, a time-invariant feedback controller may be incapable of ensuring that states that can reach T_N not only in N steps but also in fewer than N steps, are included in T_0. The example in Figure 1 illustrates this point. The system has three discrete states, $\{A, B, C\}$, two discrete inputs, $\{a, b\}$, and no disturbance. If we set

$D = \{A, B, C\} \times \{a, b\}$ and $T_N = B$, then it is easy to see that for all $N \geq 1$, $T_0 = \{A\}$. However, there does not exist a time-invariant, set-valued feedback controller capable of ensuring that starting from A the state will reach B in exactly $N \geq 2$ steps. This requirement is easy to meet with a time-varying controller of the form considered here.

The set T_0 can be computed by a backwards reachability computation. This involves iterating the so-called predecessor operator (Vidal *et al.*, 2001)

$$\begin{aligned} T_{k-1} = \{x \mid \exists u : [\exists w : (x, u, w) \in P]\wedge \\ [(x, u, w) \in P \Rightarrow \\ (x, u, w) \in D \wedge f(x, u, w) \in T_k]\} \end{aligned}$$

for $k \in \{1, 2, \ldots, N\}$, starting with T_N. In this paper $\wedge$ will be used to denote conjunction.

Notice that, by design, $T_k \subseteq \Pi_X(P)$ for $k < N$ (though not necessarily for $k = N$). In the process, one can also establish a set-valued controller which achieves the above objective. For $k \in \{0, 1, \ldots, N-1\}$,

$$\begin{aligned} \mu(x, k) := \{u \mid [\exists w : (x, u, w) \in P]\wedge \\ [(x, u, w) \in P \Rightarrow \\ (x, u, w) \in D \wedge f(x, u, w) \in T_{k+1}]\}. \end{aligned}$$

The rest of this paper concentrates on methods for characterising the set T_0 and does not treat the problem of designing the controller μ. Conceptually, obtaining a formula for the controller is straightforward once the set T_0 has been characterised. The computational implementation of the controller for piecewise affine and hybrid systems is a topic of ongoing work.

3. A GEOMETRIC APPROACH

The computation of T_{k-1}, given T_k, can be quite involved in general. If the states and inputs can only take on values from a finite set the computation of T_{k-1} can be performed by enumeration [1]. This section presents some geometric insights into reachability computations for the subclass of systems with additive disturbances and independent disturbance constraints.

With a slight abuse of notation, consider the class of systems

$$x_{k+1} = f(x_k, u_k) + g(w_k), (x_k, u_k, w_k) \in P \times W \tag{3}$$

with $x_k \in X \subset \mathbb{R}^n$, $u_k \in U \subset \mathbb{R}^m$ and $w_k \in W \subset \mathbb{R}^d$. Let $P \subseteq X \times U$ and $f : P \to \mathbb{R}^n$, $g : W \to \mathbb{R}^n$. Assume that the system is non-blocking, i.e. for all $x \in \Pi_X(P)$ there exists a $u \in U$ satisfying $(x, u) \in P$ such that $f(x, u) \in \Pi_X(P)$. Additional mixed constraints $D \subseteq X \times U$ and a target set $T_N \subseteq X$ are also given. It is assumed that P represents the physical constraints of the system and D represents desired performance, safety or other specifications imposed on the system.

For this class of systems the iteration discussed above simplifies somewhat, because the disturbance dynamics $g(\cdot)$ and the disturbance set W are not dependent on the current state or input. It has been suggested by Witsenhausen (1968, p. 13) that T_0 be computed using dynamic programming. The solution to the associated dynamic programming problem has a geometric interpretation. The set T_0 can be computed via the geometric recursion

$$\begin{aligned} T_{k-1} = \{x \mid \exists u : [(x, u) \in P \cap D]\wedge \\ [\forall w \in W : \ f(x, u) + g(w) \in T_k]\} \end{aligned} \tag{4}$$

for $k \in \{1, 2, \ldots, N\}$, starting with T_N. An argument similar to those made by Delfour and Mitter (1969), Glover and Schweppe (1971) and Bertsekas and Rhodes (1971) suggests that T_0 can equivalently be computed via the two-step recursion

$$T_k^* = \{x \,|\forall w \in W : x + g(w) \in T_k\},$$
$$T_{k-1} = \left\{x \,\middle|\, \exists u : (x, u) \in P \cap D \wedge f(x, u) \in T_k^*\right\}.$$

The next few sections will give further insight into how each stage of this recursion can be implemented.

3.1 *Computing T_k^* via the Pontryagin Difference*

The set T_k^* can be interpreted geometrically as the *Pontryagin (Minkowski) difference* between T_k and $g(W)$ (Mayne and Schroeder, 1997; Kolmanovsky and Gilbert, 1998; Kerrigan, 2000). The Pontryagin difference of two subsets of the Euclidean space $\mathbb{R}^n$, A and B, is defined as

$$A \sim B := \left\{a \in \mathbb{R}^n \,|\forall b \in B : a + b \in A\right\}.$$

It follows directly from the definition that

$$T_k^* = T_k \sim g(W).$$

Note that T_k^* is not dependent on the dynamics f. If T_k and $g(W)$ are compact, convex polyhedra, then T_k^* can be computed very efficiently by solving a finite number of linear programming problems (Mayne and Schroeder, 1997; Kolmanovsky and Gilbert, 1998). For non-convex sets a different approach, described next, is more appropriate for computing T_k^*.

3.2 *Computing T_k^* via Minkowski Summation*

The *Minkowski (vector) sum* of two subsets of the Euclidean space, A and B, is defined as

$$A \oplus B := \left\{c \in \mathbb{R}^n \,|\exists a \in A, b \in B : a + b = c\right\}.$$

It can also be shown that

$$A \sim B = \left[A^{\mathrm{c}} \oplus (-B)\right]^{\mathrm{c}},$$

where the superscript c denotes the complement of a set. This is because

$$A \sim B = \left\{a \,\middle|\, \nexists b \in B : a + b \in A^{\mathrm{c}}\right\}$$

[1] The discussion in Vidal *et al.* (2001) suggests that the computation is also possible using quantifier elimination if the states and inputs are all real-valued and all the relevant sets (including the graph of f) can be encoded in a decidable theory of the reals.

$$
\begin{aligned}
\Leftrightarrow (A \sim B)^{\mathrm{c}} &= \left\{ a \,\middle|\, \exists b \in B : a + b \in A^{\mathrm{c}} \right\} \\
&= \left\{ a \,\middle|\, \exists c \in A^{\mathrm{c}}, b \in B : a + b = c \right\} \\
&= \left\{ c \,\middle|\, \exists a \in A^{\mathrm{c}}, b \in B : c + b = a \right\} \\
&= \left\{ c \,\middle|\, \exists a \in A^{\mathrm{c}}, b \in (-B) : a + b = c \right\} \\
&= A^{\mathrm{c}} \oplus (-B).
\end{aligned}
$$

This result implies that the set T_k^* can be computed via Minkowski summation:

$$T_k^* = \left\{ T_k^{\mathrm{c}} \oplus [-g(W)] \right\}^{\mathrm{c}}.$$

3.3 *Computing T_{k-1} via Projection*

As noted in Keerthi and Gilbert (1987) and Blanchini (1994), once T_k^* has been computed, T_{k-1} is given by the projection of the set

$$\left\{ (x,u) \in P \cap D \,\middle|\, f(x,u) \in T_k^* \right\}$$

onto the state space X.

As such, it follows directly from the definition of the projection operator $\Pi_X(\cdot)$ that

$$T_{k-1} = \Pi_X \left\{ (x,u) \in P \cap D \,\middle|\, f(x,u) \in T_k^* \right\}.$$

3.4 *Computing T_{k-1} via Minkowski Summation*

As a special case of (3), let

$$x_{k+1} = f_x(x_k) + f_u(u_k) + g(w_k) \tag{5}$$

where the physical constraints P are not mixed, i.e. $P := P_x \times P_u$, $f_x : P_x \to \mathbb{R}^n$ and $f_u : P_u \to \mathbb{R}^n$. Assume also that the constraint set $D := D_x \times D_u$ does not define mixed constraints on the states and inputs. In order to simplify notation, it is assumed that D is chosen such that $D \subseteq P$, hence $P \cap D = D$.

This additional structure can be exploited (Bertsekas and Rhodes, 1971; Glover and Schweppe, 1971; Gutman and Cwikel, 1987; Mayne and Schroeder, 1997). Once T_k^* has been computed T_{k-1} can be computed via Minkowski summation, rather than via projection.

Note that one can now write

$$T_{k-1} = \left\{ x \in D_x \,\middle|\, \exists u \in D_u : f_x(x) + f_u(u) \in T_k^* \right\}.$$

By rearranging terms, as done in Section 3.2, it can be shown that

$$T_{k-1} = \left\{ x \in D_x \,\middle|\, f_x(x) \in T_k^* \oplus [-f_u(D_u)] \right\}.$$

Alternatively, one can also write

$$T_{k-1} = D_x \cap f_x^{-1} \left\{ T_k^* \oplus [-f_u(D_u)] \right\}.$$

3.5 *Summary*

Summarising, the sets required to compute T_{k-1}, given T_k, are:

(1) Complement of T_k;
(2) Set $-g(W)$;
(3) Minkowski sum $T_k^{\mathrm{c}} \oplus [-g(W)]$;
(4) Intermediate set $T_k^* = \left\{ T_k^{\mathrm{c}} \oplus [-g(W)] \right\}^{\mathrm{c}}$;
(5) Projection of $\left\{ (x,u) \in P \cap D \,\middle|\, f(x,u) \in T_k^* \right\}$ onto the state space.

In the special case of Section 3.4, the fifth step can be replaced by a Minkowski summation.

3.6 *A Word of Caution*

It is worth pointing out that the algorithms commonly used in the literature for the computation of reachable and invariant sets for linear and piecewise affine systems rely (at least in part) on the fact that there are no state-dependent input constraints (Berardi *et al.*, 1999; Koutsoukos and Antsaklis, 1999; Torrisi and Bemporad, 2001; Vidal *et al.*, 2001). If these algorithms were directly used with mixed state and input constraints they could produce incorrect results. In some cases it is easy to make the modification, but care has to be taken. For example, for the class of piecewise affine systems discussed in Section 4, Minkowski summation could produce incorrect results since, strictly speaking, within each region the dynamics are a special case of (3) and not of (5). Projection should be used instead. A special case where Minkowski summation can indeed be used for piecewise affine systems of the form (3), which are not of the form (5), will be discussed in Section 4.3.

4. A CLASS OF PIECEWISE AFFINE SYSTEMS

Consider the class of piecewise affine systems of the form

$$x_{k+1} = A_q x_k + B_q u_k + c_q + E w_k \text{ if } (x_k, u_k) \in P_q,$$

where $q \in Q \subset \mathbb{N}$ and Q is a finite set, $E \in \mathbb{R}^{n \times d}$ and $(A_q, B_q, c_q) \in \mathbb{R}^{n\times n} \times \mathbb{R}^{n \times m} \times \mathbb{R}^n$ for all $q \in Q$. The sets $D \subseteq X \times U$ and $W \subset \mathbb{R}^d$ are assumed to be compact polyhedra. To keep the notation simple, it is assumed that these sets are convex (it is possible to extend the results in this section if this is not the case). Assume further that the regions P_q in which the dynamics (A_q, B_q, c_q, E) are valid is a closed, convex polyhedron described by a set of r_q linear inequalities:

$$P_q := \left\{ (x,u) \in \mathbb{R}^n \times \mathbb{R}^m \,\middle|\, S_q \begin{bmatrix} x \\ u \end{bmatrix} \le s_q \right\},$$

where $S_q \in \mathbb{R}^{r_q \times (n+m)}$ and $s_q \in \mathbb{R}^{r_q}$. Finally, assume that the sets P_q have pairwise-disjoint interiors. i.e. $p \neq q \Rightarrow P_p^\circ \cap P_q^\circ = \emptyset$, where the superscript $\circ$ denotes the interior of a set.

Clearly this class of systems has the same structure as (3) with

$$f(x,u) := A_q x + B_q u + c_q, \text{ if } (x,u) \in P_q, q \in Q.$$

The union

$$P := \bigcup_{q \in Q} P_q$$

defines the domain of f.

This modelling framework can be used to describe a piecewise affine system which is continuous over its domain. However, it cannot accurately be used to describe a piecewise affine system which is discontinuous over its domain, but only to approximate its behaviour. As discussed by Sontag (1981), in order to accurately describe discontinuous piecewise affine systems, a mixture of non-strict and strict inequalities are needed to describe each of the regions P_q. However, because of finite precision arithmetic it is not possible to represent strict inequalities in standard computers. For the purposes of this paper, the above class of models will be deemed acceptable for approximating discontinuous piecewise affine systems – it is assumed that it is acceptable that strict inequalities of the form $a'x < b$ can be replaced by $a'x <= b - \varepsilon$, where ε is a small, positive number related to the machine precision. If a discontinuous piecewise affine system is modelled, then care has to be taken to ensure that the model is still a single-valued map.

4.1 *Computing the Intermediate Set T_k^**

The set $g(W)$ can easily be computed using linear algebra and linear programming techniques (Kerrigan, 2000, Sect 3.4.2) to give the compact, convex polyhedron $-g(W) = -EW$.

In general, even if T_k is a compact, convex polyhedron, T_{k-1} is not necessarily convex or connected. However, T_{k-1} can be described by the union of a set of compact, convex polyhedra if T_k is given by the union of a set of compact, convex polyhedra (Kerrigan, 2000, Chap. 4). As such, it is assumed that T_k is given by the union of a set of compact, convex polyhedra.

Furthermore, if T_k is given by the union of compact, convex polyhedra, then T_k^c is given by the union of open, convex polyhedra. Given $T_k := \bigcup_{i=1}^{J_k} \Sigma_{i|k}$, where each $\Sigma_{i|k}$ is a compact, convex polyhedron, the complement

$$T_k^c := \bigcup_{i=1}^{L_k} \Phi_{i|k},$$

where each $\Phi_{i|k}$ is an open, convex polyhedron, can be computed as in Kerrigan (2000, App. D). Note that it is not possible to compute T_k^c exactly using finite precision arithmetic. However, the closure of each $\Phi_{i|k}$ can be computed and used in subsequent calculations without introducing a significant error [2].

[2] Most computational geometry software only work with compact polyhedra. If the polyhedron $\Phi_{i|k}$ is unbounded, it can be intersected with a sufficiently large compact, convex polyhedron for computational purposes.

It follows from the definition of the Minkowski sum that

$$T_k^c \oplus (-EW) = \bigcup_{i=1}^{L_k} \left[\Phi_{i|k} \oplus (-EW)\right].$$

Since $-EW$ and the closure of $\Phi_{i|k}$ are closed, convex polyhedra, the set $\Phi_{i|k} \oplus (-EW)$ can be computed efficiently using standard computational geometry software packages (Veres and Mayne, 2001). Finally, since each of the $\Phi_{i|k} \oplus (-EW)$ is an open, convex polyhedron and T_k is compact, it follows that

$$T_k^* = \left[T_k^c \oplus (-EW)\right]^c := \bigcup_{i=1}^{M_k} \Omega_{i|k},$$

where each $\Omega_{i|k}$ is a compact, convex polyhedron, can be computed as in Kerrigan (2000, App. D).

4.2 *Computing T_{k-1} via Projection*

The fact that P and D have been treated as representing two different aspects of the reachability problem allows one to note that

$$\begin{aligned}
&\left\{(x,u) \in P \cap D \,\middle|\, f(x,u) \in T_k^*\right\} \\
&= \bigcup_{q \in Q} \left\{(x,u) \in P_q \cap D \,\middle|\, f(x,u) \in T_k^*\right\} \\
&= \bigcup_{q \in Q} \bigcup_{i=1}^{M_k} \left\{(x,u) \in P_q \cap D \,\middle|\, f(x,u) \in \Omega_{i|k}\right\} \\
&= \bigcup_{q \in Q} \bigcup_{i=1}^{M_k} \{(x,u) \in P_q \cap D \mid \\
&\qquad\qquad A_q x + B_q u + c_q \in \Omega_{i|k}\}.
\end{aligned}$$

Since $\Omega_{i|k}$ is a compact, convex polyhedron, the set

$$\left\{(x,u) \in P_q \cap D \,\middle|\, A_q x + B_q u + c_q \in \Omega_{i|k}\right\}$$

is also a compact, convex polyhedron. Moreover, the projection of a convex polyhedron is a convex polyhedron and the projection of a union of sets is equal to the union of the projections of these sets. Following on from Section 3.3 this implies that

$$\begin{aligned}
T_{k-1} = \bigcup_{q \in Q} \bigcup_{i=1}^{M_k} \Pi_X \{(x,u) \in P_q \cap D \mid \\
A_q x + B_q u + c_q \in \Omega_{i|k}\},
\end{aligned}$$

where the projection

$$\Pi_X \left\{(x,u) \in P_q \cap D \,\middle|\, A_q x + B_q u + c_q \in \Omega_{i|k}\right\}$$

is a compact, convex polyhedron which can be computed efficiently using standard computational geometry software (Veres and Mayne, 2001).

4.3 *Computing T_{k-1} via Minkowski Summation*

Let $D := D_x \times D_u \subseteq P$ and each of the regions $P_q := P_{x,q} \times P_u$. If the P_q and D are compact, convex polyhedra then projection can be avoided and Minkowski summation can be used to compute T_{k-1}. Similar to the discussion in Section 3.4, it can be shown that

$$T_{k-1} = \bigcup_{q\in Q} \bigcup_{i=1}^{M_k} \{x \in P_{x,q} \cap D_x \mid A_q x + c_q \in \Omega_{i|k} \oplus (-B_q D_u)\}$$

despite the fact that the dynamics are of the form (3) and not (5). The reason why Minkowski summation can be used, is that within each P_q the input constraints are not state-dependent and hence the same ideas as in Section 3.4 are applicable. However, care still has to be taken to include $P_{x,q}$ in the computation. A computational example of this special case is presented in Kerrigan (2000, Sect. 4.6).

5. CONCLUDING REMARKS

A small step towards addressing systems with mixed constraints on the states and inputs has been presented. The approach developed in this paper is capable of dealing with systems with mixed constraints on the state and controls, but relies on the fact that disturbance constraints are independent. Current research is aimed at alleviating this restriction; it is likely that geometrical concepts and tools will also prove useful in this respect.

The geometric approach discussed here is also currently being applied to a class of hybrid systems where the continuous dynamics are piecewise affine. It is expected that, for the class of piecewise affine hybrid systems, the geometric approach will be orders of magnitude faster than the quantifier elimination approach used by Koutsoukos and Antsaklis (1999) and Vidal *et al.* (2001). The worst-case time complexity of quantifier elimination methods is doubly exponential. Quantifier elimination also does not exploit the structure of the problem. A geometric approach exploits the structure of the problem as well as making use of efficient computational geometry software.

6. ACKNOWLEDGEMENTS

The authors would like to thank Alberto Bemporad and Fabio Torrisi for valuable discussions related to reachability computations for piecewise affine systems.

7. REFERENCES

Berardi, L., E. De Santis and M.D. Di Benedetto (1999). A structural approach to the control of switching systems with an application to automotive engines. In: *Proc. 38th IEEE Conference on Decision and Control*. pp. 3526–3531.

Bertsekas, D.P. and I.B. Rhodes (1971). On the minimax reachability of target sets and target tubes. *Automatica* **7**, 233–247.

Blanchini, F. (1994). Ultimate boundedness control for uncertain discrete-time systems via set-induced Lyapunov functions. *IEEE Trans. Autom. Control* **39**(2), 428–433.

Delfour, M.C. and S.K. Mitter (1969). Reachability of perturbed systems and min sup problems. *SIAM J. Control* **7**(4), 521–533.

Glover, J.D. and F.C. Schweppe (1971). Control of linear dynamic systems with set constrained disturbances. *IEEE Trans. Autom. Control* **AC-16**(5), 411–423.

Gutman, P. and M. Cwikel (1987). An algorithm to find maximal state constraint sets for discrete-time linear dynamical systems with bounded controls and states. *IEEE Trans. Autom. Control* **AC-32**(3), 251–254.

Habets, L.C.G.J.M. and J.H. van Schuppen (2001). A controllability result for piecewise linear hybrid systems. In: *Proc. 6th European Control Conference*. pp. 3870–3873.

Keerthi, S.S. and E.G. Gilbert (1987). Computation of minimum-time feedback control laws for discrete-time systems with state-control constraints. *IEEE Trans. Autom. Control* **AC-32**(5), 432–435.

Kerrigan, E.C. (2000). Robust Constraint Satisfaction: Invariant Sets and Predictive Control. PhD thesis. University of Cambridge, UK. Thesis and Invariant Set Toolbox for MATLAB downloadable from author's website.

Kolmanovsky, I. and E.G. Gilbert (1998). Theory and computation of disturbance invariant sets for discrete-time linear systems. *Math. Probl. Eng.* **4**, 317–367.

Koutsoukos, X.D. and P.J. Antsaklis (1999). Design of hybrid system regulators. In: *Proc. 38th IEEE Conference on Decision and Control*.

Maciejowski, J.M. (2001). *Predictive Control with Constraints*. Prentice Hall, UK.

Mayne, D.Q. and W.R. Schroeder (1997). Robust time-optimal control of constrained linear systems. *Automatica* **33**(12), 2103–2118.

Sontag, E.D. (1981). Nonlinear regulation: The piecewise linear approach. *IEEE Trans. Autom. Control* **AC-26**(2), 346–358.

Torrisi, F.D. and A. Bemporad (2001). Discrete-time hybrid modeling and verification. In: *Proc. 40th IEEE Conference on Decision and Control*.

Veres, S.M. and D.Q. Mayne (2001). *Geometric Bounding Toolbox for* MATLAB. SysBrain. `http://www.sysbrain.com`.

Vidal, R., S. Schaffert, O. Shakernia, J. Lygeros and S. Sastry (2001). Decidable and semi-decidable controller synthesis for classes of discrete time hybrid systems. In: *Proc. 40th IEEE Conference on Decision and Control*.

Witsenhausen, H.S. (1968). A minimax control problem for sampled linear systems. *IEEE Trans. Autom. Control* **AC-13**(1), 5–21.