Proceeding of the 2004 American Control Conference
Boston, Massachusetts June 30 - July 2, 2004

Low Complexity Control of Piecewise Affine Systems with Stability Guarantee

Pascal Grieder, Michal Kvasnica, Mato Baotić and Manfred Morari

Abstract

Piecewise affine systems are powerful models for describing both non-linear and hybrid systems. One of the key problems in controlling these systems is the inherent computational complexity of controller synthesis and analysis, especially if constraints on states and inputs are present. This paper illustrates how reachability analysis based on multi-parametric programming may serve to obtain controllers of low complexity. Specifically, two different controller computation schemes are presented. In addition, a method to obtain stability guarantees for general receding horizon control of PWA systems is given.

Keywords: Piecewise Affine Systems, Receding Horizon Control, Minimum-Time Control

I. Introduction

Optimal control of piecewise affine (PWA) systems has garnered increasing interest in the research community since they represent a powerful tool for approximating non-linear systems and because of their equivalence to hybrid systems [9]. Optimal control for PWA systems may be obtained by solving mixed-integer optimization problems on-line [4], [11], or as was shown in [1], [6], [10], by solving off-line a number of multi-parametric programs which were introduced for constrained linear systems in [5]. By multi-parametric programming, a linear (mp-LP) or quadratic (mp-QP) optimization problem is solved off-line. The associated solution takes the form of a PWA state feedback law. In particular, the state-space is partitioned into polyhedral sets and for each of those sets the optimal control law is given as one affine function of the state. In the on-line implementation of such controllers, input computation reduces to a simple set-membership test. Even though the approaches in [1], [6], [10] rely on off-line computation of a feedback law, the computation quickly becomes prohibitive for larger problems. This is not only due to the high complexity of the multi-parametric programs involved [8], but mainly because of the exponential number of transitions between regions which can occur when a controller is computed in a dynamic programming fashion [6], [10].

Institut für Automatik, ETH - Swiss Federal Institute of Technology, CH-8092 Zürich
Corresponding Author: E-mail: `grieder@aut.ee.ethz.ch`, Tel. +41 01 632 7313

This paper addresses the clear need for low complexity controllers for hybrid systems and presents two algorithms which tend to achieve this goal. Specifically, the computation of a minimum-time feedback controller is presented as well as a control scheme which aims at obtaining a low (but not necessarily minimal) number of switches in the system dynamics. In addition, a general scheme for obtaining stability guarantees for generic PWA systems subject to receding horizon control will be presented. Unlike the method in [11], we do not require the PWA dynamics to be continuous. This scheme can also be used in connection with other controller computation methods [11], [1], [6], [10] to obtain stability guarantees. Specifically, an invariant target set along with a piecewise linear feedback law and an associated Lyapunov function is computed with semi-definite programming methods and the optimal control problem is subsequently updated according to [12] in order to obtain stability properties.

II. Problem Description and Properties

This section first covers some of the fundamentals of mp-QP for linear systems before restating recent results for PWA systems. Consider a discrete-time linear time-invariant system

$$x(k+1) \quad = \quad Ax(k) + Bu(k) \tag{1}$$

with $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$. Let $x(k)$ denote the measured state at time k and x_k (u_k) denote the predicted state (input) at time k, given $x(0)$. Assume now that the states and the inputs of the system in (1) are subject to the following constraints

$$x(k) \in \mathbb{X} \subset \mathbb{R}^n, \quad u(k) \in \mathbb{U} \subset \mathbb{R}^m, \quad \forall k > 0, \tag{2}$$

where $\mathbb{X}$ and $\mathbb{U}$ are polyhedral sets containing the origin in their interior, and consider the constrained finite-time optimal control problem

$$J_N^*(x(0)) = \min_{u_0,\dots,u_{N-1}} \sum_{k=0}^{N-1} \left(u_k'\mathcal{R}u_k + x_k'\mathcal{Q}x_k\right) + x_N'\mathcal{Q}_f x_N, \quad (3a)$$

$$\text{subj. to.} \quad x_k \in \mathbb{X},\ u_{k-1} \in \mathbb{U}, \quad \forall k \in \{1,\dots,N\}, \quad (3b)$$

$$x_N \in \mathcal{X}_{set}, \quad (3c)$$

$$x_{k+1} = Ax_k + Bu_k,\ x_0 = x(0), \quad (3d)$$

$$\mathcal{Q} = \mathcal{Q}' \succeq 0,\quad \mathcal{Q}_f = \mathcal{Q}_f' \succeq 0,\quad \mathcal{R} = \mathcal{R}' \succ 0, \quad (3e)$$

where (3c) is a user defined set constraint on the final state which may be chosen along with $\mathcal{Q}_f$ such that stability of the closed-loop system is guaranteed [12].

Definition 1: We define the N-step feasible set $\mathcal{X}^N \subseteq \mathbb{R}^n$ as the set of initial states $x(0)$ for which the optimal control problem (3) is feasible, i.e.

$$\mathcal{X}^N = \{x(0) \in \mathbb{R}^n \mid \exists U_N \in \mathbb{R}^{Nm}, x_k \in \mathbb{X},\ u_{k-1} \in \mathbb{U},\ \forall k \in \{1,\dots,N\}\}.$$

where $U_N = [u_0',\dots,u_{N-1}']'$ is the optimization vector. By considering $x(0)$ as a parameter, problem (3) can be stated as an mp-QP [5] which can be solved to obtain a feedback solution with the following properties,

Theorem 1: [5] Consider the finite time constrained regulation problem (3). Then, the set of feasible parameters $\mathcal{X}^N$ is convex, the optimizer $U_N^* : \mathcal{X}^N \to \mathbb{R}^{Nm}$ is continuous and piecewise affine (PWA), i.e.

$$U_N^*(x(0)) = F_r x(0) + G_r, \quad \text{if} \quad x(0) \in \mathcal{P}_r$$
$$\mathcal{P}_r = \{x \in \mathbb{R}^n \mid H_r x \le K_r\},\ r = 1,\dots,R$$

and the optimal cost $J_N^* : \mathcal{X}^N \to \mathbb{R}$ is continuous, convex and piecewise quadratic.

According to Theorem 1, the feasible state space $\mathcal{X}^N$ is partitioned into R polytopic regions, i.e., $\mathcal{X}^N = \{\mathcal{P}_r\}_{r=1}^R$. The results in [5] were extended in [6] to compute the optimal explicit feedback controller for PWA systems of the form

$$x(k+1) = A_i x(k) + B_i u(k) + f_i, \quad (4a)$$

$$L_i x(k) + E_i u(k) \le W_i,\ i \in \mathcal{I} \quad (4b)$$

$$\text{if} \quad x(k) \in \mathcal{D}_i \quad (4c)$$

whereby the dynamics (4a) with the associated constraints (4b) are valid in the polyhedral set $\mathcal{D}_i$ defined in (4c). The set $\mathcal{I} \subset \mathbb{N}$ is defined as $\mathcal{I} = \{1,\dots,I_f\}$ where I_f denotes the number of different dynamics and $\mathbb{N} = \{0,1,\dots\}$ denotes the set of integers greater equal zero (and $\mathbb{N}^+ = \{1,2,\dots\}$). Henceforth, we will abbreviate (4a) and (4c) with $x(k+1) = f_{\text{PWA}}(x(k),u(k))$. Note that we do not require $x(k+1) = f_{\text{PWA}}(x(k),u(k))$ to be continuous. The optimization problem considered here is given by

$$J_N^*(x(0)) = \min_{u_0,\dots,u_{N-1}} \sum_{k=0}^{N-1} \left(u_k'\mathcal{R}u_k + x_k'\mathcal{Q}x_k\right) + x_N'\mathcal{Q}_f x_N, \quad (5a)$$

$$\text{subj. to} \quad x_N \in \mathcal{X}_{set}, \quad (5b)$$

$$L_i x_k + E_i u_k \le W_i, \quad \text{if } x_k \in \mathcal{D}_i, \quad (5c)$$

$$x_{k+1} = A_i x_k + B_i u_k + f_i,\ x_0 = x(0), \quad (5d)$$

$$\mathcal{Q} = \mathcal{Q}' \succeq 0,\quad \mathcal{Q}_f = \mathcal{Q}_f' \succeq 0,\quad \mathcal{R} = \mathcal{R}' \succ 0. \quad (5e)$$

In the following section, an algorithm is needed that can detect if a convex polyhedron $\mathcal{P}_0$ is covered by a finite set of non-empty convex polyhedra $\{\mathcal{P}_r\}_{r=1}^R$. Due to space constraints, we refer the reader to [2], [7], where an efficient algorithm is given to perform this task.

III. Computation of Stabilizing Controllers for Piecewise Affine Systems

One of the main drawbacks of the methods described in [6] is the lack of an a priori stability guarantee for the closed-loop system. Other methods [11] only provide stability guarantees if the origin is contained in the interior of one of the sets $\mathcal{D}_i$. We propose a method for obtaining stabilizing controllers for generic PWA systems. For general dynamic systems, stability is guaranteed if an invariant set is imposed as a terminal state constraint (see (3c)) and the terminal cost in (3) corresponds to a Lyapunov function for that set [12]. Analogous to controllers for linear systems, we here compute a control invariant set $\mathcal{X}_{inv}$ with an associated Lyapunov function. In a first step a stabilizing piecewise linear feedback controller is computed. This can be achieved by searching for feedback controllers K_i and a matrix P such that

$$\begin{aligned} P &\succ 0, \\ (A_i + B_iK_i)'P(A_i + B_iK_i) - P &\preceq 0, \qquad \forall i \in \mathcal{I}. \end{aligned}$$

This can be rewritten as an LMI by using Schur complements and introducing the new variables $Y_i = K_iQ$ and $Q = P^{-1}$,

$$\max_{Y_i,Q} \det(Q), \quad \text{subj. to}$$

$$Q \succ 0,$$

$$\begin{bmatrix} Q & (A_iQ + B_iY_i) \\ (A_iQ + B_iY_i)' & Q \end{bmatrix} \succeq 0, \quad \forall i \in \mathcal{I}.$$

The maximization of $\det(Q)$ serves to maximize the region of stability. Large target sets generally make the subsequent controller computations simpler. In a second step, the maximal output admissible set $\mathcal{X}_{inv}$ of the PWA system subject to the feedback controllers K_i can be computed with the algorithm in [13], which is guaranteed to terminate in a finite number of steps for the problem at hand.

If we add the terminal set constraint $\mathcal{X}_{set} = \mathcal{X}_{inv}$ along with the terminal Lyapunov cost $\mathcal{Q}_f = P$ in (5), stability is guaranteed according to [12]. Note that we only need to consider a single convex terminal set for linear systems. For PWA systems, the terminal set $\mathcal{X}_{inv}$ is given as a union of several convex sets $\mathcal{X}_{inv} = \bigcup_i \mathcal{X}_i^0$ which may be non-convex. However, if the union $\bigcup_i \mathcal{X}_i^0$ is convex, the regions can be merged with the method in [3] which results in reduced algorithm run-time and solution complexity.

Note that the proposed method for guaranteeing closed-loop stability can easily be combined with other control algorithms, e.g., [6], [11], [10], [1]. However, the procedure here is merely sufficient for stability. We cannot guarantee that an invariant set and a Lyapunov function will be found in the suggested manner.

IV. Computation of a Minimum Time Controller for Hybrid Systems

The goal is the design of a feedback controller, such that the system constraints (2) are satisfied for all time and stability is guaranteed. Without loss of generality, we restrict ourselves to the regulation problem, i.e. how the state x can be steered to the origin without violating any of the system constraints along the closed loop trajectory.

One of the key problems in control of PWA systems is the lack of convexity in the controlled sets, which produces a significant computational overhead. Furthermore, the complexity of the cost-to-go function in the dynamic programming approach in [6], [10] makes it necessary to explore an exponentially growing number of possible target sets during the iterations. The algorithms presented here avoid these issues to some extent by forfeiting the optimality of the control law. Specifically, we compute a minimal time and a 'reduced-switching' controller. Unlike the approaches in [6], [10], the cost-to-go here will only assume discrete values. Due to the 'simple' cost-to-go, the target sets which need to be considered at each iteration step are larger and fewer in number than those which would be obtained if a cost optimal controller were to be computed. Thus, both complexity of the feedback law as well as computation time are greatly reduced, in general.

If the proposed algorithm terminates, the associated feedback controller will cover the maximal controllable set $\mathcal{K}_\infty^{\text{PWA}}(\mathcal{X}_{inv})$.

Definition 2: The set $\mathcal{K}_\infty^{\text{PWA}}(\mathcal{X}_{inv})$ denotes the maximum controllable set for a PWA system (4), i.e., it contains all states which can be steered into $\mathcal{X}_{inv}$. Specifically,

$$\begin{aligned}\mathcal{K}_\infty^{\text{PWA}}(\mathcal{X}_{inv}) = \{&x(0) \in \mathbb{R}^n |\ \exists u(k) \in \mathbb{R}^m, \text{s.t.}\\ &L_i x(k) + E_i u(k) \leq W_i, \text{if } x(k) \in \mathcal{D}_i,\\ &x(k+1) = f_{\text{PWA}}(x(k), u(k)),\\ &x(N) \in \mathcal{X}_{inv},\ \forall k \geq 0, N \to \infty\}.\end{aligned}$$

A. Off-Line Computation

Before presenting the algorithm, some preliminaries will be introduced.

Assume a possibly non-convex union $\mathcal{U}^0$ of polytopes $\mathcal{X}_j^0$, i.e. $\mathcal{U}^0 = \bigcup_{j\in\mathcal{L}^0} \mathcal{X}_j^0$, where the set $\mathcal{L}^0 = \{1, 2, \ldots, L_f^0\}$ contains L_f^0 positive integers. All states which can be driven into the set $\mathcal{U}^0$ are defined by:

$$\begin{aligned}Pre(\mathcal{U}^0) &= \{x \in \mathbb{X} \mid \exists u \in \mathbb{U}, f_{\text{PWA}}(x, u) \in \mathcal{U}^0\}\\ &= \bigcup_{i\in\mathcal{I}} \bigcup_{j\in\mathcal{L}^0} \Big\{x \in \mathbb{X} \mid \exists u \in \mathbb{U},\\ &\qquad x \in \mathcal{D}_i,\ A_i x + B_i u + f_i \in \mathcal{X}_j^0\Big\}.\end{aligned}$$

For a fixed i and j, the target set $\mathcal{X}_j^0$ is convex and the dynamics affine, such that it is possible to apply standard multi-parametric programming techniques to solve the problem at hand [5]. Therefore the set $Pre(\mathcal{U}^0)$ is a union of polytopes and can be computed by solving $I_f \cdot L_f$ multi-parametric programs. In addition to the set $Pre(\mathcal{U}^0)$, we then also obtain an associated feedback law which provides feasible inputs as a function of the state (see Theorem 1). Note that the various controller partitions may overlap, but that each controller will drive the state into $\mathcal{U}^0$ in one time step. We will henceforth use the following notation $\mathcal{U}^{iter+1} = Pre(\mathcal{U}^{iter}) = \bigcup_{j\in\mathcal{L}^{iter+1}} \mathcal{X}_j^{iter+1}$.

In the following, the Algorithm for computing the minimum time controller for PWA systems will be introduced.

Algorithm 4.1: Computation: Minimum Time Controller

1) Compute the invariant set $\mathcal{X}_{inv}$ around the origin (see Figure 1(a)) and an associated Lyapunov function as described in Section III.
2) Initialize the set list $\mathcal{U}^0 = \mathcal{X}_{inv}$ and initialize the iteration counter $iter = 0$.
3) Compute $\mathcal{U}^{iter+1} = Pre(\mathcal{U}^{iter}) = \bigcup_{j\in\mathcal{L}^{iter+1}} \mathcal{X}_j^{iter+1}$, by solving a sequence of multi-parametric programs. Thus, a feedback controller partition $\{\mathcal{P}_j^{iter+1}\}_{r=1}^R$ is associated to each obtained set $\mathcal{X}_j^{iter+1}$. Obviously, the number of regions R of each partition are a function of $iter$ and j (see Figure 1(b)).
4) For all $j^* \in \mathcal{L}^{iter+1}$: If $\mathcal{X}_{j^*}^{iter+1} \subseteq \bigcup_{j\in\mathcal{L}^{iter+1}\setminus\{j^*\}} \mathcal{X}_j^{iter+1}$, then discard $\mathcal{X}_{j^*}^{iter+1}$

and set $\mathcal{L}^{iter+1} = \mathcal{L}^{iter+1} \setminus \{j^*\}$ (see Figure 1(c)).

5) If $\mathcal{U}^{iter} \neq \mathcal{U}^{iter+1}$, set $iter = iter + 1$ and goto step 3.
6) For all $k \in \{1, \ldots, iter-1\}$ and $r \in \mathbb{N}^+$ discard all controller regions $\{\mathcal{P}_j^{k+1}\}_r$ for which $\{\mathcal{P}_j^{k+1}\}_r \subseteq \bigcup_{i\in\{1,\ldots,k\}} \mathcal{U}^i$ since the associated control law will never be applied.

The index $iter$ corresponds to the number of steps in which a state trajectory will enter the terminal set $\mathcal{X}_{inv}$ if a RHC policy is applied. If the algorithm terminates in finite time, the union of all controlled sets $\mathcal{U}^{iter}$ is the maximum controllable set $\mathcal{K}_\infty^{\text{PWA}}(\mathcal{X}_{inv})$ as given in Definition 2. Note however, that Algorithm 4.1 may not terminate in finite time (e.g. if states are unbounded). It is therefore advisable to specify a maximum step distance which can be used as a termination criterion in step 5 of Algorithm 4.1.

B. On-Line Application

In the minimum time algorithm presented in this paper, we can take advantage of some of the algorithm features to speed up the on-line region identification procedure. We propose a three-tiered search tree structure which serves to significantly speed up the region identification. Unlike the search tree proposed in [14], the tree structure proposed here is computed automatically by Algorithm 4.1, i.e., no post-processing is necessary. The three levels of the search tree are as follows:

Algorithm 4.2: On-Line Application of Minimum Time Controller

1) Identify active dynamics i, such that $x \in \mathcal{D}_i$.
2) Identify controller set $\mathcal{X}_j^{iter}$ associated with dynamic i which is 'closest' to the target set, i.e., $\min_{iter,j} \;\; iter, \;\;$ s.t. $x \in \mathcal{X}_j^{iter}$.
3) Extract the controller partition $\{\mathcal{P}_j^{iter}\}_{r=1}^R$ with the corresponding feedback laws F, G and identify the region r which contains the state $x \in \{\mathcal{P}_j^{iter}\}_r$.
4) Apply the control input $u = F_r x + G_r$. Goto 1.

Note that the association of controller partitions $\mathcal{X}_j^{iter}$ to active dynamics in step 2 is trivially implemented by building an appropriate lookup-table during the off-line computation in Algorithm 4.1.

Theorem 2: The controller obtained with Algorithm 4.1 and applied to a PWA system (4) in a receding horizon control fashion according to Algorithm 4.2, guarantees stability and feasibility of the closed loop system, provided $x(0) \in \mathcal{K}_\infty^{\text{PWA}}(\mathcal{X}_{inv})$.

Proof: Assume the initial state $x(0)$ is contained in the set $\mathcal{U}^{iter}$ with a step distance to $\mathcal{X}_{inv}$ of $iter$. The control law at step 4 of Algorithm 4.2 will drive the state into a set $\mathcal{U}^{iter-1}$ in one time step (see step 3 of Algorithm 4.1). Therefore, the state will enter $\mathcal{X}_{inv}$ in $iter$ steps. Once the state enters $\mathcal{X}_{inv}$ the feedback controllers associated with the common quadratic Lyapunov ensures stability. □

The proof stretches the classic definition of stability, since the Lyapunov function is discontinuous and assumes only discrete values for $x \notin \mathcal{X}_{inv}$. However, this is not a problem, since Lyapunov functions do not need to be continuous for discrete time systems.

V. Controller with Reduced Number of Switches

It is possible to obtain even simpler controllers and faster computation times by modifying Algorithm 4.1. Instead of computing a minimum time controller, an alternative scheme which aims at reducing the number of switches can be applied. A change in the active system dynamic $\mathcal{D}_i \to \mathcal{D}_j$, $(i \neq j)$ is referred to as a switch. The proposed procedure does not guarantee the minimum number of switches, though straightforward modifications to the algorithm would yield such a solution. The "minimum number of switches" solution was not pursued in this paper since computation time was the primary objective.

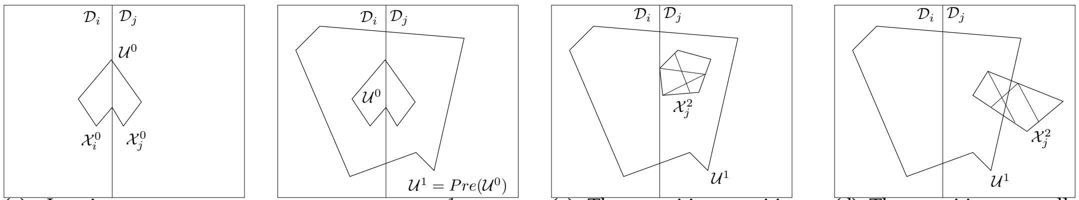
The proposed reduced switch controller will avoid switching the active dynamics for as long as possible. We will here introduce the following operator

$$\begin{aligned} Pre_i(\mathcal{X}) =& \{x \in \mathbb{X} \mid \exists u \in \mathbb{U}, \\ & x \in \mathcal{D}_i, \; A_i x + B_i u + f_i \in \mathcal{X}\}. \end{aligned}$$

Once the $j-th$ controller set $\mathcal{X}_j^{iter,i}$ associated to dynamic i and obtained at iteration $iter$ is computed, the set is subsequently used as a target set for as long as the controllable set of states can be enlarged. With this scheme, the total number of convex sets needed to describe the controlled set $\mathcal{U}^{iter}$ remains constant while the size of $\mathcal{U}^{iter}$ increases. Therefore, this scheme generally results in fewer sets during the dynamic programming iterations compared to Algorithm 4.1. The proposed scheme is guaranteed to work at least as well as Algorithm 4.1 with respect to controller complexity. Specifically, the algorithm works as follows:

Algorithm 5.1: Computation: Controller with Reduced Number of Switches

1) Compute the invariant set $\mathcal{X}_{inv}$ around the origin and an associated Lyapunov function as described in Section III.
2) Initialize the set list $\mathcal{U}^0 = \mathcal{X}_{inv} = \bigcup_{j\in\mathcal{L}^0} \mathcal{X}_j^0$ and initialize the iteration counter $iter = 0$.



(a) Invariant target set $\mathcal{X}_{inv}$. (b) Set of states $\mathcal{U}^1$ which enter $\mathcal{U}^0$ in one time step. (c) The transition partition does not expand the controlled set of states. (d) The transition controller expands the controllable set of states.

Fig. 1. Description of Algorithm 4.1.

3) Execute the following for all $i \in \mathcal{I}$ and $j \in \mathcal{L}^{iter}$:
    a) Initialize counter $c = iter$ and set $\mathcal{C}^c = \mathcal{X}_j^c$.
    b) Compute $\mathcal{C}^{c+1} = Pre_i(\mathcal{C}^c)$ by using multi-parametric programming and store the associated controller partition. Thus, a feedback controller partition $\{\mathcal{P}_j^{c+1,iter}\}_{r=1}^R$ is obtained.
    c) If $\mathcal{C}^c \subseteq \mathcal{C}^{c+1}$, set $c = c + 1$ and goto step 3b.
    d) If $c = iter$ set $\mathcal{U}^{iter+1} = \mathcal{U}^{iter+1} \cup \mathcal{C}^{c+1}$, else $\mathcal{U}^{iter+1} = \mathcal{U}^{iter+1} \cup \mathcal{C}^c$.
4) If $\mathcal{U}^{iter+1} \neq \mathcal{U}^{iter}$, set $iter = iter + 1$ an goto 3.
5) For all $k \in \{1, \ldots, iter - 1\}$, $c \in \mathbb{N}$ and $r \in \mathbb{N}^+$ discard all controller regions $\{\mathcal{P}_j^{c,k+1}\}_r$ for which $\{\mathcal{P}_j^{c,k+1}\}_r \subseteq \bigcup_{i\in\{1,\ldots,k\}} \mathcal{U}^i$ since the associated control law will never be applied.

The on-line computation is identical to the scheme described in Section IV-B.

Remark 1: If we always have $\mathcal{C}^c \nsubseteq \mathcal{C}^{c+1}$ in step 3c of Algorithm 5.1, then Algorithm 5.1 is identical to Algorithm 4.1. However if $\mathcal{C}^c \subseteq \mathcal{C}^{c+1}$, it is possible to perform a large part of the computations on convex sets, which makes Algorithm 5.1 significantly more efficient than Algorithm 4.1.

Theorem 3: A controller computed according to Algorithm 5.1 and applied to a PWA system (4) according to Algorithm 4.2, guarantees stability and feasibility of the closed loop system, provided $x(0) \in \mathcal{K}_\infty^{\text{PWA}}(\mathcal{X}_{inv})$.

Proof: Follows from Theorem 2. □

VI. Numerical Examples

As was shown in [8] and will also be illustrated in this section, computing minimum time controllers instead of optimal controllers may serve to significantly reduce computation time, since in general, fewer regions are obtained than with the algorithm in [6]. We will demonstrate efficiency of Algorithms 4.1 and 5.1 on the following examples.

Example 1: Consider the 2-dimensional piece-wise linear system $x(k+1) = A_i x(k) + B_i u(k)$, such that:

$$i = \begin{cases} 1, & \text{if } x_1(k) \geq 0 \ \& \ x_2(k) \geq 0 \\ 2, & \text{if } x_1(k) \leq 0 \ \& \ x_2(k) \leq 0 \\ 3, & \text{if } x_1(k) \leq 0 \ \& \ x_2(k) \geq 0 \\ 4, & \text{if } x_1(k) \geq 0 \ \& \ x_2(k) \leq 0 \end{cases}$$

$$A_1 = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \quad B_1 = \begin{bmatrix} 1 \\ 0.5 \end{bmatrix},$$

$$A_2 = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \quad B_2 = \begin{bmatrix} -1 \\ -0.5 \end{bmatrix},$$

$$A_3 = \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix}, \quad B_3 = \begin{bmatrix} -1 \\ 0.5 \end{bmatrix},$$

$$A_4 = \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix}, \quad B_4 = \begin{bmatrix} 1 \\ -0.5 \end{bmatrix},$$

One can observe, that the system is a double integrator in the discrete time domain, with different orientation of the vector field in the different quadrants. The state and input constraints, respectively, are: $-5 \leq x_1(t) \leq 5$, $-5 \leq x_2(t) \leq 5$, $-1 \leq u(k) \leq 1$, and the weight matrices for the optimization problem are $\mathcal{Q} = I$, and $\mathcal{R} = 1$.

Example 2: Consider the following 3-dimensional PWA system [11]:

$$x(k+1) = A_i x(k) + B_i u(k) + f_i, \quad i = \begin{cases} 1, & \text{if } \ x_2(k) \leq 1 \\ 2, & \text{else} \end{cases}$$

$$A_1 = \begin{bmatrix} 1 & 0.5 & 0.3 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}, \quad B_1 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad f_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} 1 & 0.2 & 0.3 \\ 0 & 0.5 & 1 \\ 0 & 0 & 1 \end{bmatrix}, \quad B_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad f_2 = \begin{bmatrix} 0.3 \\ 0.5 \\ 0 \end{bmatrix}$$

Subject to constraints $-10 \leq x_1(k) \leq 10$, $-5 \leq x_2(k) \leq 5$, $-10 \leq x_3(k) \leq 10$, and $-1 \leq u(k) \leq 1$. Again, weights in the cost function are $\mathcal{Q} = I$, $\mathcal{R} = 0.1$.

To initialize the algorithm described in Section IV, one first needs to compute a control invariant set $\mathcal{X}_{inv}$ around the origin. Once the set $\mathcal{X}_{inv}$ is computed, Algorithms 4.1 and 5.1 are applied to Examples

	Algorithm 4.1		Algorithm 5.1		Dynamic Programming [6]	
	Run time	# regions	Run time	# regions	Run time	# regions
Example 1	71 sec.	174	39 sec.	138	91 hours	3904
Example 2	791 sec.	642	151 sec.	293	⋆	⋆

TABLE I

COMPARISON OF THE CPU-TIME AND THE NUMBER OF REGIONS FOR DIFFERENT SYSTEMS. ⋆ DENOTES THAT THE COMPUTATION FOR THE PARTICULAR PROBLEM HAD NOT CONVERGED AFTER 7 DAYS.

1-2. A comparison of both Algorithms 4.1 and 5.1 with the approach in [6] is given in Table I. In [6], the authors solve an optimal control problem in a dynamic programming fashion for a fixed horizon. In order to guarantee a fair comparison, the algorithm as presented in [6] was slightly modified to include an additional check for convergence. This addition was necessary to guarantee termination of the algorithm as soon as the controller covers $\mathcal{K}_{\infty}^{\mathrm{PWA}}(\mathcal{X}_{inv})$. Note that the influence of this additional check on the run-times given in Table I is negligible. As can be seen from the results in Table I, the methods proposed in this paper are superior to the approach of [6] regarding complexity. The reason for the drastic decrease in runtime is the reduced number of target sets. However, neither Algorithm 4.1 nor 5.1 guarantee optimal closed loop-performance in the sense of the cost-objective in (5).

It is not possible to put an a priori bound on the performance of Algorithms 4.1 and 5.1 with respect to the optimal solution. A performance comparison was performed on a set of examples in [7] and is omitted here for space reasons. The average performance decay was on the level of 5% whereas the worst case was above 100% versus the cost optimal trajectories. This may not seem encouraging, but as Table I illustrates, the minimum-time feedback control may often be the only type of controller which is computable in reasonable time.

VII. CONCLUSION

A novel algorithm to compute a low complexity feedback controller for constrained PWA systems was presented. Based on iterative computations of multi-parametric programs, a feedback controller is obtained which drives the state into a target set in minimum time. An alternative controller which aims at reducing the number of switches between different dynamics is also presented and the provided examples suggest that this approach may further reduce complexity significantly. The two algorithms reduce complexity versus optimal controllers [6], [10] by several orders of magnitude whilst incurring a negligible decay in the average closed-loop performance. Furthermore, a search tree for efficient on-line identification of the optimal feedback law is automatically constructed by both algorithms.

The presented algorithms as well as a more detailed report can be downloaded from `http://control.ee.ethz.ch`.

REFERENCES

[1] M. Baotić, F.J. Christophersen, and M. Morari. A new Algorithm for Constrained Finite Time Optimal Control of Hybrid Systems with a Linear Performance Index. In *European Control Conference*, Cambridge, UK, September 2003.

[2] M. Baotić and F.D. Torrisi. Polycover. Technical Report AUT03-11, Automatic Control Lab, ETHZ, Switzerland, 2003.

[3] A. Bemporad, K. Fukuda, and F.D. Torrisi. Convexity recognition of the union of polyhedra. *Computational Geometry*, 18:141–154, April 2001.

[4] A. Bemporad and M. Morari. Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35(3):407–427, March 1999.

[5] A. Bemporad, M. Morari, V. Dua, and E.N. Pistikopoulos. The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1):3–20, January 2002.

[6] F. Borrelli, M. Baotić, A. Bemporad, and M. Morari. An efficient algorithm for computing the state feedback optimal control law for discrete time hybrid systems. In *Proc. 2003 American Control Conference*, Denver, Colorado, USA, June 2003.

[7] P. Grieder, M. Kvasnica, M. Baotić, and M. Morari. Low complexity control of piecewise affine systems with stability guarantee. Technical Report AUT03-13, Automatic Control Lab, ETHZ, Switzerland, 2003.

[8] P. Grieder and M. Morari. Complexity reduction of receding horizon control. In *Proc. 42th IEEE Conf. on Decision and Control*, Maui, Hawaii, USA, December 2003.

[9] W.P.M.H. Heemels, B. De Schutter, and A. Bemporad. Equivalence of hybrid dynamical models. *Automatica*, 37(7):1085–1091, July 2001.

[10] E. C. Kerrigan and D. Q. Mayne. Optimal control of constrained, piecewise affine systems with bounded disturbances. In *Proc. 41st IEEE Conference on Decision and Control*, Las Vegas, Nevada, USA, December 2002.

[11] D. Q. Mayne and S. Raković. Model predicitive control of constrained piecewise affine discrete-time systems. *Int. J. of Robust and Nonlinear Control*, 13(3):261–279, April 2003.

[12] D. Q. Mayne, J.B. Rawlings, C.V. Rao, and P.O.M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, June 2000.

[13] S. Raković, P. Grieder, M. Kvasnica, D. Q. Mayne, and M. Morari. Computation of invariant sets for piecewise affine discrete time systems subject to bounded disturbances. *Submitted*, 2003.

[14] P. Tøndel, T.A. Johansen, and A. Bemporad. Evaluation of piecewise affine control via binary search tree. *Automatica*, 39(5):945–950, 2003.