Proceedings of the 41st IEEE
Conference on Decision and Control
Las Vegas, Nevada USA, December 2002

Controller Synthesis for a class of Uncertain Piecewise Linear Hybrid Dynamical Systems [1]

Hai Lin[†2], Panos J. Antsaklis[†]

[†] *EE Dept, Univ of Notre Dame, Notre Dame, IN 46556 USA*

Abstract

In this paper, we consider the controller synthesis problem for a class of uncertain hybrid dynamical systems. The goal is for the closed loop system to exhibit desired behavior under dynamic uncertainty and exteriors disturbances. The main question is whether there exists a controller such that the closed loop system satisfies the specification. The notion of *attainability* is introduced to refer to the specified behavior that can be forced to the plant by a control mechanism. We give a method for attainability checking by employing the predecessor operator and backward reachability analysis, and a procedure for controller design by using finite automata and linear programming techniques.

1 Introduction

In this paper, a novel methodology for analysis and synthesis of *uncertain piecewise linear hybrid dynamical systems* based on backward reachability analysis is presented. Piecewise linear (affine) systems have been widely studied in the literature, see for example [10, 5, 3, 4, 6] and the references therein. The issues studied include modeling, stability, observability and controllability etc. Piecewise linear systems arise often from linearization of nonlinear systems. Note that a large class of systems with uncertainty or parameter variations, or systems with strong nonlinearities are often of interest. If we use ordinary piecewise linear systems to approximate and study such nonlinear systems, we have to shrink the operating region of the linearization. And this results in a large number of linearizations (modes) which makes the subsequent analysis and synthesis computationally expensive or even untractable. So we propose to introduce a bundle of linearizations, whose convex hull cover the original (maybe uncertain) nonlinear dynamics, instead of approximating with just a single linearization. This way, we may keep the operating region from shrinking, and so we may study uncertain nonlinear systems in a systematic approach and with less computational burden.

[1]The partial support of the National Science Foundation (NSF ECS99-12458 & CCR01-13131), and of the DARPA/ITO-NEST Program (AF-F30602-01-2-0526) is gratefully acknowledged.

[2]Corresponding author. hlin1@nd.edu.

In our earlier work [7], we formulated and analyzed a class of uncertain, or parameter-variant piecewise linear systems[1]. Here we consider the controller synthesis problem for uncertain piecewise linear hybrid dynamical systems with polytopic continuous dynamics uncertainty. The control objective is for the closed loop system to follow a desired behavior. The main question is whether there exists a controller so that the closed loop system follows the specification. The notion of *attainability* is introduced to refer to the specified behavior that can be forced to the plant by a control mechanism. We give a method for attainability checking by employing the predecessor operator and backward reachability analysis, and a procedure for controller design by using finite automata and linear programming techniques.

The structure of this paper is as follows. Section 2 defines the uncertain hybrid dynamical systems. Section 3 considers the predecessor operator and backward reachability analysis. Section 4 deals with control specifications. The regulator problem for piecewise linear hybrid dynamical systems is formulated in Section 5. The controller design methodology is described and simulation results of a temperature control system is presented to illustrate the validity of the design methodology. Finally, concluding remarks are made.

2 Model

In the following, we define a class of piecewise linear hybrid dynamical systems with polytopic uncertainty. The discrete dynamics are described by finite automata, and the interaction between the continuous and the discrete part is defined by piecewise linear maps. The exact definition is as follows.

Definition 2.1 *Consider the* Uncertain Piecewise Linear Hybrid Dynamical Systems *(uncertain PLHDS) defined by*

$$\begin{aligned} x(t+1) &= \bar{A}_{q(t)}x(t) + B_{q(t)}u(t) + E_{q(t)}d(t) && (2.1)\\ q(t+1) &= \delta(q(t), \pi(x(t)), \sigma_c(t), \sigma_u(t)) && (2.2)\\ y(t) &= g(q(t), x(t)) && (2.3) \end{aligned}$$

where $q \in Q = \{q_1, q_2, \cdots, q_s\}$ and Q is the collection of discrete states (modes); $x \in X \subset \mathbb{R}^n$ and X stands for the continuous state space, $y \in Y \subset \mathbb{R}^r$ and Y stands for the

[1]This work can be seen as extensions of our group's previous work [6] to uncertain systems and to more general cases.

continuous output space, the continuous control $u \in \mathcal{U} \subset \mathbb{R}^m$, the continuous disturbance $d \in \mathcal{D} \subset \mathbb{R}^p$, and $\mathcal{U}$, $\mathcal{D}$ are bounded convex polyhedral set; and

- $\tilde{A}_q \in \mathbb{R}^{n\times n}$. *The entries in $\tilde{A}_q$ are unknown, but $\tilde{A}_q$ can be expressed as a convex combination of N_q $\mathbb{R}^{n\times n}$ matrices $\{A_q^1, A_q^2, ..., A_q^{N_q}\}$, that is*

$$\tilde{A}_q = \sum_{i=1}^{N_q} \lambda_i A_q^i, \quad \lambda_i \geq 0, \sum_{i=1}^{N_q} \lambda_i = 1 \tag{2.4}$$

- $B_q \in \mathbb{R}^{n\times m}$, *and* $E_q \in \mathbb{R}^{n\times p}$ *are the system matrices for the discrete state q,*
- $\pi : X \to X/E_\pi$ *partitions the continuous state space $\mathbb{R}^n$ into polyhedral equivalence classes.*
- $q(t+1) \in act(\pi(x(t)))$, *where $act : X/E_\pi \to 2^Q$ defines the active mode set,*
- $\delta : Q \times X/E_\pi \times \Sigma_c \times \Sigma_u \to Q$ *is the discrete state transition function. Here $\sigma_c \in \Sigma_c$ denotes a controllable event and Σ_u the collection of uncontrollable events.*
- $g : Q \times X \to Y$ *is the output function.*
- *The* guard $G(q, q')$ *of the transition (q, q') is defined as the set of all states (q, x) such that $q' \in act(\pi(x(t)))$ and there exist controllable event $\sigma_c \in \Sigma_c$ such that $q' = \delta(q, \pi(x), \sigma_c, \sigma_u)$ for every uncontrollable event $\sigma_u \in \Sigma_u$.*

3 Backward Reachability Analysis

The main mathematical tool to be used for backward reachability analysis is the *predecessor operator* applied recursively to subsets of the hybrid state space.

3.1 The Predecessor Operator

A region of the state space is defined as $R \subset Q \times X$. We are interested in computing the set of all the states that can be driven to R by either continuous or discrete transitions. We assume that the region is represented by $R = (q, P)$ where $q \in Q$ and $P \subset \mathbb{R}^n$ is a piecewise linear set. Let's assume that P can be represented by $P = \{x \in \mathbb{R}^n | Gx \leq w\}$ [2]. The dynamic evolution of the system is defined by discrete and continuous transitions.

Discrete Transitions. The predecessor operator for discrete transitions is denoted by $pre_d : 2^{Q\times X} \to 2^{Q\times X}$ and it is used to compute the set of states that can be driven to the region R by a discrete instantaneous transition $q' \to q$ that can be forced by the controller for any uncontrollable event. The predecessor operator in this case is defined as follows:

$$pre_d(R) = \{(q', x) \in Q \times X | \exists \sigma_c \in \Sigma_c, \forall \sigma_u \in \Sigma_u, \; q = \delta(q', x, \sigma_c, \sigma_u)\}$$

For every discrete transition that can be forced by a controllable event we have that

$$pre_d(R) = \bigcup_{q' \in act(P)} G(q', q) \cap (\{q'\} \times P)$$

where $G(q', q)$ is the guard of transition $q' \to q$.

Continuous Transitions. In the case of continuous transitions, given the region $R = (q, P)$ we define the predecessor operator $pre_c : 2^{Q\times X} \to 2^{Q\times X}$ to compute the set of states for which there exists a control input so that the continuous state will be driven in the set P for every disturbance, while the system is at the discrete mode q. The action of the operator is described by

$$pre_c^q(R) = \{q\} \times \{x \in X | \exists u \in \mathcal{U}, \forall d \in \mathcal{D}, \forall \tilde{A}_q \in Conv_{i=1}^{N_q}(A_q^i), \; \tilde{A}_q x + B_q u + E_q d \in P\}$$

Let's denote $pre_{c,i}^q(R)$ for $1 \leq i \leq N_q$ as $pre_{c,i}^q(R) = \{q\} \times \{x \in X | \exists u \in \mathcal{U}, \forall d \in \mathcal{D}, A_q^i x + B_q u + E_q d \in P\}$[3], and assume that the piecewise linear set $P = \{x \in \mathbb{R}^n | Gx \leq w\}$, where $G \in \mathbb{R}^{v\times n}$, $w \in \mathbb{R}^v$. Then, we have a proposition:

Proposition 3.1

$$pre_c^q(R) = \bigcap_{i=1,\cdots,N_q} pre_{c,i}^q(R)$$

Remark: The significance of the proposition is that the calculation for the continuous predecessor for the polytopic uncertain PLHDS can be boiled down to the finite intersection of continuous predecessor set of the polytope vertices PLHDS. The algorithm for calculating the robust predecessor set can be found in [7, 8].

Remark: Please note that the set $pre(R)$ is piecewise linear and is described using a finite set of linear inequalities. Therefore, we can apply the predecessor operator to compute the set of all states that can be driven to $pre(R)$ to get $pre(pre(R))$. Following the same procedure, we define successive applications of the predecessor operator as:

$$pre^M(R) = \underbrace{pre(...pre(R))}_{M times} \tag{3.5}$$

Remark: For a given region R,we define the *coreachable set* $CR(R)$ as the set of all states that can be driven to R. The coreachable set for a region of the hybrid state space can be computed by successive application of the predecessor operator

$$CR(R) = pre^*(R) \tag{3.6}$$

In general, the proposed procedure is semi-decidable and its termination is not guaranteed. We will return to this matter in the reachability problem shortly after.

4 Control Specifications

In this section, we present a modeling formalism for control specifications based on finite automata models, called the *exosystem*. We consider both static and

[2] Please note that $a \leq b$ means that all entries in the vector $(a - b)$ are all non-positive.

[3] The $pre_{c,i}^q(R)$ is nothing but continuous predecessor set of the i-th vertex A_q^i.

dynamic specifications. *Static specifications* describe desired outputs that do not change as time progresses. For example, *safety* and *reachability* are static specifications. *Dynamic specifications* involve sequencing of events and eventual execution of actions. In a manufacturing system, for example, the assembly of a component may require that a set of tasks is executed in a specific order. Our control objective is that the closed loop system exhibits the same behavior as the exosystem. The main question is whether there exists a controller so that the closed loop system follows the behavior of the exosystem. This question is directly related to the existence of appropriate control resources in order for the plant to achieve the desired behavior. We formalize this notion using the *attainability* of the specified behavior. In this work, attainable behavior refers to behavior that can be forced to the plant by a control mechanism.

4.1 Static Specifications

Typical control specifications investigated in this paper are formulated in terms of partitions of the state space of the system. Examples include safety problems, where the controller guarantees that the plant will not enter an unsafe region.

Safety. At first, we focus on the safety problem and we show how the refinement of the state space partition can be used to formulated conditions for safety. Given a set of states described by the region $R \subset Q \times X$ and an initial condition $(q_0, x_0) \in R$, we say that the system is *safe* if $(q(t), x(t)) \in R$ for every $t \geq t_0$. The conditions that guarantee that a given region of the hybrid state space is safe can be described as following.

Theorem 4.1 [4] *An uncertain PLHDS is safe with respect to the region $R \subseteq Q \times X$ if and only if $R \subseteq pre(R)$.*

Reachability. Secondly, we study the reachability problem for uncertain piecewise linear hybrid dynamical systems. It should be emphasized that we are interested only in the case when reachability between two regions R_1 and R_2 is defined so that the state is driven to R_2 directly from the region R_1 in finite steps without entering a third region. This is a problem of practical importance in hybrid systems since it is often desirable to drive the state to a target region of the state space while satisfying constraints on the state and input during the operation of the system.

The problem of deciding whether a region R_2 is directly reachable from R_1 can be solved by recursively computing all the states that can be driven to R_2 from R_1 using the predecessor operator. As we have discussed, the proposed procedure is semi-decidable and its termination is not guaranteed. In order to formulate a constructive algorithm for reachability, we consider two approaches. First, we consider an upper bound on the time horizon and we examine the reachability only for the predetermined finite horizon. Second, we formulate a termination condition for the reachability algorithm based on a grid-based approximation of the piecewise linear regions of the state space [6].

Theorem 4.2 [4] *Consider an uncertain PLHDS described by definition 2.1 and the regions $R_1 = (\mathbf{q}_1, P_1)$ and $R_2 = (\mathbf{q}_2, P_2)$ then the region R_2 is directly reachable from R_1 if and only if $R_1 \subseteq CR(R_2)$.*

4.2 Dynamic Control Specifications

In this section, we present a modeling formalism for control specifications based on finite automata models, and we consider dynamic specifications.

Exosystem. We consider specifications that are described with respect to regions of the hybrid state space. We define the set X_e as $X_e = \{R_1, R_2, ..., R_M\}$ where $R_i = (\mathbf{q}_i, P_i)$ are piecewise linear regions of the hybrid state space. Since we assume that the primary partition is fine enough to describe the specifications, for every region we can write $R_i \subseteq Q \times X/E_\pi$. In the following, we use a formal automaton model to represent the specifications of interest as [6].

Definition 4.1 *The control specifications are modelled by an input-output (I/O) deterministic finite automaton described by $\mathcal{E} = (X_e, V_e, Y_e, \delta_e, \lambda_e, R_0)$ where X_e is the set of states, V_e is the input alphabet, Y_e is the output alphabet, $\delta_e : X_e \times V_e \to X_e$ is the state transition function, $\lambda_e : X_e \to Y_e$ is the output function returning the output associated with each state, and R_0 is the initial state.*

We assume that the function δ_e is non-total, which means that not every input can be applied to every state of the automaton. We also assume that every state is reachable and therefore, there exists appropriate input sequences so that every state can be reached. The I/O finite automaton which describes the specifications is a deterministic Moore automaton and is called the *exosystem.*

Attainability. Our control objective is that the closed loop system consisting of the plant and the controller exhibits the same behavior [5] as the exosystem. And the main question is if there exists a controller so that the closed loop system follows the behavior of the exosystem. We formalize this notion using the *attainability* of the specified behavior. In the following we present the necessary and sufficient condition for attainability. In this work, attainable behavior refers to behavior that can be forced to the plant by a control mechanism.

[4]The proof is analogous to proof for the corresponding theorem on [6].

[5]A dynamical system can be described as a triple (T, W, B) where T is the time axis, W is the signal space, and $B \subset W^T$ (the set of all functions $f : T \to W$) the behavior[11].

Theorem 4.3 [4] *The specification behavior B_{sp} is attainable* if and only if *the following conditions hold: First, Every terminating state y_n corresponds to a region R_n that is safe; and secondly for every non-terminating state y_k, there exists y_{k+1} so that, for the corresponding regions we have that R_{k+1} is reachable from R_k.*
Furthermore, if B_{sp} is attainable then there exists a controller C so that the regulator problem has a solution.

5 Hybrid Systems Controller Design

In this section, we present a systematic procedure for *controller* design. Assume that exact state measurement (q,x) is available. An *admissible control input* (or *law*) is one which satisfies the input constraints $(\Sigma_c, \mathcal{U})$. In this section, we present a systematic procedure for *controller* design. It is assumed that the desired behavior is attainable and therefore there exists a control policy so that the plant will follow the output of the exosystem. The design of the controller is based on the regions $\{R_1, \cdots, R_M\}$ that are used to define the control specifications, which means starting from region R_1 the state $(q(t), x(t))$ directly reaches region R_2 and son on, until it enters the region R_M, then the state $(q(t), x(t))$ should stay within R_M. Follows [6], the proposed representation for the controller is shown below.

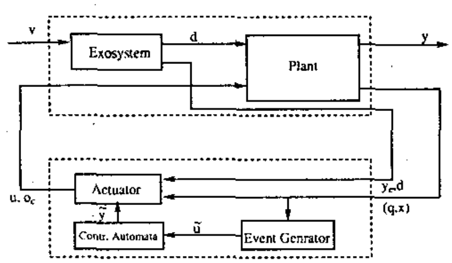

Figure 1: Hybrid systems controller diagram.

The controller consists of three agents. The *event generator* receives the discrete-time measurement signal of the hybrid plant, and issues appropriate events when the state $(q(t), x(t))$ enters a new region R_i of the hybrid state space. The *control automaton* is a finite automaton whose states correspond to the regions R_i and its main purpose is to select an appropriate cost functional based on the control objective. Finally, The *actuator* determines the control input to be applied to the plant using an optimization algorithm based on the desired output provided by the exosystem. The control input consists of a continuous component $u \in \mathcal{U}$ and a discrete component $\sigma_c \in \Sigma_c$ which triggers feasible discrete transitions. At every time step, the control input is selected as the solution to a mathematical programming problem. In the following, we formulate the optimization problem that is used by the actuator, considering the specification behavior described by $\{R_1, \cdots, R_M\}$.

Safety Controller. First, we consider terminating output symbols that represent safety conditions for the corresponding region, R_M, of the state space. We define the cost functional $J_M : Q \times [0,1]^{N_q} \times \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p \to \mathbb{R}$

$$J_M(q, \lambda, x, u, d) = c^T[\sum_{i=1}^{N_q}(\lambda_i A_q^i)x(t) + B_q u(t) + E_q d(t)]$$

where $c \in \mathbb{R}^n$ a coefficient vector, which is constructed by the matrix of the constrained region P_M. Assume $P_M = \{x \in \mathbb{R}^n | G_M x \leq w_M\}$, then c may be selected of the form $w^T G_M$, where $w^T \in \mathbb{R}^n$ called a weighted vector. Without loss of generality, w is selected in such a way that make $w^T G_M x \leq 0$ for all $x \in P_M$. The control signal is selected as the solution to the following optimization problem:

$$\min_{u \in \mathcal{U}} \max_{\lambda \in [0,1]^{N_q},\, d \in \mathcal{D}} J_M(q, \lambda, x, u, d)$$
$$s.t.\ \tilde{A}_q x(t) + B_q u(t) + E_q d(t) \in P_M$$

The optimal action of the controller is one that tries to minimize the maximum cost, and try to counteract the worst disturbance and the worst model uncertainty. Here, the disturbance and uncertainty is given the advantage: the control plays first and disturbance and uncertainty play second with the knowledge of the controller's play. This kind of solution is referred to as *Stackelberg solution.*

By following similar arguments as in the proof of Proposition 3.1[7], the above optimization problems can be boiled down into a linear programming problem,

$$\min_{u \in \mathcal{U}} c^T B_q u(t)$$
$$s.t. \begin{cases} G_M[A_q^1 x(t) + B_q u(t) + E_q d(t)] \leq w_M \\ G_M[A_q^2 x(t) + B_q u(t) + E_q d(t)] \leq w_M \\ \cdots\cdots \\ G_M[A_q^{N_q} x(t) + B_q u(t) + E_q d(t)] \leq w_M \\ u \in \mathcal{U},\ d \in \mathcal{D} \end{cases}$$

The above problem can be solved very efficiently. The following algorithm describe the procedure for the synthesis of safety controller for an given initial condition (q_0, x_0) containing in a specified region $R_M = (\mathbf{q}_M, P_M)$.

Algorithm 5.1 Safety Controller
INPUT: $R_M = (\mathbf{q}_M, P_M)$, (q_0, x_0);
if $\min_u \max_\lambda J_M(q_0, x_0, \lambda, u)$ feasible
  $u^* = arg \min_u J_M(q_0, x_0, \lambda, u)$
  $q^* = q_0$
else
  for i=1, $\cdots$, $|\mathbf{q}_M|$,

$q_i = \mathbf{q}_M(i)$
if $x_0 \in G_{q_0}^{q_i}$
$J_M^{q_i} = \min_u \max_\lambda J_M(q_i, x_0, \lambda, u)$
end
end
$q^* = arg\min_{q_i \in \mathbf{q}_M} J_M^{q_i}$
$u^* = arg\min_u J_M^{q_i}$
end
OUTPUT: u^*, q^*

In the procedure, we first try to remain the mode and avoid switching, simply because switching maybe costy. However, sticking to mode q_0 may be not a good choice, and there may not exist feasible control signal. So the procedure try to take possible mode switching into consideration and choose the mode that can make the next continuous state farthest from the boundary. We claim that there must be at least one optimization problem be feasible[6]. Here q^* stands for the mode that corresponds to the minimum cost value J_M, then the candidate control input is selected as $(\sigma_c(t), u^*(t))$ where $q^* = \delta(q(t), \pi(x(t)), \sigma_c(t), \varepsilon)$ and u^* is the solution of the above optimization procedure.

Reachability Controller. Next, we consider two non-terminating output symbols y_k and y_{k+1} which describe a reachability specification between the regions $R_k = (\mathbf{q}_k, P_k)$ and $R_{k+1} = (\mathbf{q}_{k+1}, P_{k+1})$. The control objective is to drive every state in R_k to R_{k+1}. Let the convex polyhedral set $P_k = \{x : Gx \le w\}$. For a pair of modes $q_k \in \mathbf{q}_k$ and $q_k' \in \mathbf{q}_{k+1}$, assume the intersection of the guard set for (q_k, q_k'), $G_{q_k}^{q_k'}$, with the common region of P_k and P_{k+1}, is not empty. Let's denote this polytope as $P_C^{(q_k,q_k')} = P_k \cap P_{k+1} \cap G_{q_k}^{q_k'} = \{x : G_C x \le w_C\}$. We define the cost functional, $J_C : Q \times Q \times [0,1]^{N_q} \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p \to \mathbb{R}$

$$J_C(q_i, q_i', x, \lambda, u, d) = c^T[\sum_{i=1}^{N_{q_i}} (\lambda_i A_{q_i}^i) x(t) + B_{q_i} u(t) + E_{q_i} d(t)]$$

Where the vector c is selected in the same way as described above from polytope $P_C^{(q_k,q_k')}$. The control signal is selected as the solution to the following minmax optimization problem:

$$\min_{u\in\mathcal{U}} \max_{\lambda\in[0,1]^{N_{q_i}}} J_C(q_i, q_i', x, \lambda, u, d)$$
$$s.t.\ \tilde{A}_{q_i} x(t) + B_{q_i} u(t) + E_{q_i} d(t) \in P$$
$$u \in \mathcal{U},\ d \in \mathcal{D}$$

Similarly, this optimization problem can be reduced to the following linear programming problem:

$$\min_{u\in\mathcal{U}} c^T B_{q_i} u(t)$$
$$s.t. \begin{cases} G[A_{q_i}^1 x(t) + B_{q_i} u(t) + E_{q_i} d(t)] \le w \\ G[A_{q_i}^2 x(t) + B_{q_i} u(t) + E_{q_i} d(t)] \le w \\ \cdots\cdots \\ G[A_{q_i}^{N_{q_i}} x(t) + B_{q_i} u(t) + E_{q_i} d(t)] \le w \\ u \in \mathcal{U},\ d \in \mathcal{D} \end{cases}$$

[6]Please turn to [8] for proof.

The following algorithm design the controller to guarantee the directly reachability.

Algorithm 5.2 Reachability Controller

INPUT: $\Omega_k = (\mathbf{q}_k, P_k)$, $\Omega_{k+1} = (\mathbf{q}_{k+1}, P_{k+1})$, (q_0, x_0), *feasibility = 0;*
for *j=1,* $\cdots$, $|\mathbf{q}_{k+1}|$,
$q_j' = \mathbf{q}_{k+1}(j)$
if $\min_u \max_\lambda J_C(q_0, q_j', x_0, \lambda, u)$ feasible
$J_C^{(q_0,q_j')} = \min_u \max_\lambda J_C(q_0, q_j', x_0, \lambda, u)$
feasibility = 1
end
end
if *feasibility ==1*
$ind = arg\min_{q_j' \in \mathbf{q}_{k+1}} J_C^{(q_0,q_j')}$
$u^* = arg\min_u J_C^{(q_0,ind)}$
$q^* = q_0$
else
for *i=1,* $\cdots$, $|\mathbf{q}_k|$,
$q_i = \mathbf{q}_k(i)$
if $x_0 \in G_{q_0}^{q_i}$
for *j=1,* $\cdots$, $|\mathbf{q}_{k+1}|$,
$q_j' = \mathbf{q}_{k+1}(j)$
$J_C^{(q_i,q_j')} = \min_u \max_\lambda J_C(q_i, q_j', x_0, \lambda, u)$
end
end
end
$[q^*, q'] = arg\min_{q_i \in \mathbf{q}_k; q_j' \in \mathbf{q}_{k+1}} J_M^{(q_i,q_j')}$
$u^* = arg\min_u J_M^{q_i}$
end
OUTPUT: u^*, q^*

Attainability Controller. The following algorithm design the controller to guarantee the directly attainability for the specification behavior described by $\{R_1, \cdots, R_M\}$.

Algorithm 5.3 Attainability Controller

INPUT: $\{R_1, \cdots, R_M\}$, (q_0, x_0);
for *n =1,* $\cdots$, *M-1,*
while $x_0 \in R_n$ and $x_0 \notin R_{n+1}$
Design Reachability Controller from $\mathrm{R_n}$ to $\mathrm{R_{n+1}}$
end
end
Design Safety Controller for $\mathrm{R_M}$
OUTPUT: u^*, q^*

Proposition 5.1 [6] *Consider the controller shown in Figure 1 with the event generator, control automaton, and actuator as defined above. If the specification behavior is attainable, the output behavior of the closed loop system follows the specified behavior of the exosystem, that is* $B_{cl} = B_{sp}$.

Example 5.1 (TEMPERATURE CONTROL SYSTEM) *The system consists of a furnace that can be switched on* *and off. The control objective is to control the temperature at a point of the system by applying the heat input at a different point. So, the discrete mode only contains two states, that is the furnace "off", q_0, and the furnace is "on", q_1. The continuous dynamics is described as* [7]

$$x(t+1) = \begin{cases} \tilde{A}_0 x(t) + B_0 u(t) + E_0 d(t), & q = q_0 \\ \tilde{A}_1 x(t) + B_1 u(t) + E_1 d(t), & q = q_1. \end{cases}$$

where

$$\begin{aligned} A_0^1 &= \begin{pmatrix} 0.825 & 0.135 \\ 0.68 & 1 \end{pmatrix}, A_0^2 = \begin{pmatrix} 1 & 0.35 \\ 0.068 & 0.555 \end{pmatrix} \\ B_0 &= \begin{pmatrix} 1.8179 \\ 0.0773 \end{pmatrix}, E_0 = \begin{pmatrix} 0.0387 \\ 0.3772 \end{pmatrix} \\ A_1^1 &= \begin{pmatrix} -0.664 & 0.199 \\ 0.199 & 0.264 \end{pmatrix}, A_1^2 = \begin{pmatrix} -0.7 & 0.32 \\ 0.32 & 0.44 \end{pmatrix} \\ B_1 &= \begin{pmatrix} 0.8101 \\ 0.1369 \end{pmatrix}, E_1 = \begin{pmatrix} 0.1369 \\ 0.5363 \end{pmatrix} \end{aligned}$$

The partition of the state space is obtained by considering the following hyperplane

$$h_1(x) = x_1 - 20, h_2(x) = x_2 - 5, h_3(x) = x_2, h_4(x) = x_1$$

Assume $u \in \mathcal{U} = [-1,1]$, $d \in \mathcal{D} = [-0.1, 0.1]$ Consider region $R_1 = (\{q_0, q_1\}, P_1)$ and $R_2 = (\{q_0, q_1\}, P_2)$, where $P_1 = \{x \in \mathbb{R}^2 | (0 \le x_1 \le 20) \wedge (-20 \le x_2 \le 0)\}$, and $P_2 = \{x \in \mathbb{R}^2 | (0 \le x_1 \le 20) \wedge (0 \le x_2 \le 5)\}$. Our control object is that for every initial sate (q_0, x_0) within region R_1 there exist control $u \in \mathcal{U}$ and $\sigma_c \in \Sigma_c$ so that from (q_0, x_0) the state can be driven to R_2 without entering a third region, then the state will stay inside R_2 [8]*, no matter what the dynamic uncertainty, continuous and discrete disturbances are. Let's check the* attainability. *We first calculate $pre(R_2)$, which cover the region R_2, so R_2 is safe. By recursively using $pre(.)$, we find that R_1 can be driven to R_2 in three steps, i.e. reachable. So the attainability of the specification is satisfied. Then, we design the controller and plot the simulation result for nominal plant (here we choose the epicenter of the state matrix , i.e. $\frac{1}{2}(A_q^1 + A_q^2)$) in Figure 2. Also the control signal output (σ_c, u) of the controller is plotted in Figure 2.*

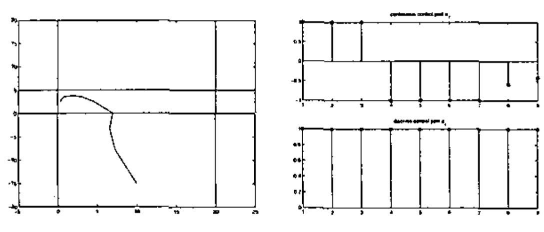

Figure 2: *Left:* Simulation for closed loop nominal plant (assuming $d = 0$). *Right:* The control signals output (σ_c, u) of the Controller.

6 Conclusion

In this paper, we consider controller synthesis for a class of uncertain hybrid systems, in which the continuous dynamics are described by linear difference equations with polytopic uncertainties, the discrete dynamics by finite automata, and the interaction between the continuous and discrete part is defined by piecewise linear maps. The existence of a controller such that the closed loop systems follow desired output of exosystem under uncertainty and disturbance is analysis first. Then, based on the proposed notion of attainability for the desired behavior of piecewise linear hybrid systems, we present a systematic procedure for controller design by using finite automata and linear programming techniques. However, in this paper we only consider the uncertainty in A_q matrix, our next step is consider the uncertainty in both A_q and B_q as in [8].

References

[1] E. Asarin, O. Bournez, T. Dang, O. Maler, and A. Pnueli. *Effective synthesis of switching controllers for linear systems.* Proceedings of IEEE, 88(7):1011-1025, July 2000.

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

[3] A. Bemporad, G. Ferrari-Trecate and M. Morari. *Observability and controllability of piecewise affine and hybrid systems,* IEEE Tran. on AC, 45(10): 1864-1876, 2000

[4] W. Heemels,B. De Schutter, A. Bemporad. *On the equivalence of classes of hybrid dynamical models,* The 40th IEEE Conference on Decision and Control, (1): 364-369 , 2001

[5] M. Johansson, *Piecewise Linear Control Systems,* Ph.D. Thesis, Lund Institute of Technology, Sweden, 1999.

[6] X.D. Koutsoukos, *Analysis and Design of Piecewise Linear Hybrid Dynamical Systems.*PhD thesis, Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN, 2000.

[7] H. Lin, X.D. Koutsoukos and P.J. Antsaklis, *Hierarchical Control of a class Uncertain Piecewise Linear Hybrid Dynamical Systems,* IFAC'2002, Barcelona, Spain.

[8] H. Lin and P.J. Antsaklis, *Analysis and Synthesis for Discrete-Time Polytopic Uncertain Piecewise Linear Hybrid Dynamical Systems,* draft, Aug. 2002 (`http://www.nd.edu/ isis/tech.html`)

[9] E. Sontag. *Remarks on piecewise-linear algebra.* Pacific Journal of Mathematics, 92(1):183-210, 1982.

[10] E. Sontag. *Interconnected automata and linear systems: A theoretical framework in discrete-time.* Hybrid Systems III, volume 1066 of Lecture Notes in Computer Science, pages 436-448. Springer, 1996.

[11] J.Willems. *Paradigms and puzzles in the theory of dynamical systems.* IEEE Transactions on Automatic Control, 36(3):259-294, 1991.

[7] using zero-order hold sampling with $T = 1$s.

[8] Of cause we can build an automata, exosystem, to describe such simple specification.