Hierarchical Design of Piecewise Linear Hybrid Dynamical Systems Using a Control Regulator Approach

Xenofon D. Koutsoukos
Department of EECS
Vanderbilt University
Nashville, TN 37235, USA
Tel. +1-615-322-8283
Fax +1-615-343-5459
Xenofon.Koutsoukos@vanderbilt.edu

Panos J. Antsaklis
Department of Electrical Engineering
University of Notre Dame
Notre Dame, IN 46556, USA
Tel. +1-574-631-5792
antsaklis.1@nd.edu

Abstract

This paper presents a novel framework for hierarchical control of piecewise linear hybrid dynamical systems. The main characteristic of this class of hybrid systems is that the continuous dynamics are described by linear difference equations, the discrete dynamics by finite automata, and the interaction between the continuous and the discrete part is defined by piecewise linear maps. Control design is formulated as a regulator problem and algorithms for the synthesis of dynamical controllers are developed. Control specifications are modeled as finite automata. Both static specifications that do not change as time progresses and dynamic specifications that include sequencing of events and eventual execution of actions are considered. Control design is implemented using finite automata and linear programming techniques. Simulation results of a tank system are used to illustrate the approach.

1 INTRODUCTION

This paper presents a novel framework for hierarchical control of piecewise linear hybrid dynamical systems. Control design is formulated as a regulator problem and algorithms for the synthesis of dynamical controllers are developed. Our work is motivated by the need to address challenging problems in the control and coordination of modern complex engineering applications such as chemical and manufacturing plants, autonomous vehicles, and multiple robotic systems. A mathematical model that can capture both discrete and continuous phenomena that arise in such systems is presented. The continuous dynamics are described by linear time invariant difference equations and the discrete dynamics by finite automata. The interaction between the continuous and discrete parts is defined by piecewise linear maps characterized by sets of linear equalities and inequalities. We refer to this class of systems as *piecewise linear hybrid dynamical systems* in order to emphasize the hybrid nature of the systems and problems of interest. Piecewise linear hybrid dynamical systems are a class of systems that is general enough to describe important engineering applications,

but simple enough to facilitate the development of analysis, and more importantly, synthesis tools.

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, for example, guaranteeing that two interacting robots will not collide. Also reachability problems where the controller drives the plant from an initial operating region or state to a desired one; this is the case for example in the startup procedure of a chemical plant. Safety and reachability specifications can be characterized as static specifications since they do not change as time progresses. In this paper, we also present a formal framework for dynamic specifications that 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 while each task is satisfying safety specifications.

The proposed framework is based on a formulation of the regulator problem for piecewise linear hybrid dynamical systems. In general, a regulator requests certain types of outputs from the plant and these are to be attained in the presence of disturbances. The desired outputs can be described as the outputs of another dynamical system, called the *exosystem*. Our objective is to design a controller so that the closed loop system consisting of the plant and the controller 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 on the plant by a control mechanism*. Based on the proposed notion of attainability for the desired behavior of piecewise linear hybrid systems, we present a systematic procedure for controller design. We present a convenient representation for the controller as a dynamical system which consists of three agents: the *event generator*, the *control automaton*, and the *actuator*. The plant and the exosystem are linked by a controller to form a regulator. A feedback controller can be designed to regulate the system. Simulation results are used to illustrate the proposed methodology using a tank system.

We present a new control design framework for piecewise linear hybrid systems. We follow a hierarchical approach that separates the control task into two levels. The task at the higher level aims at establishing conditions that guarantee that safety and reachability specifications are attainable, meaning that there exist appropriate control inputs for the plant so that the closed loop system satisfies the specifications. These are conditions that can be tested off-line based on the feasibility of appropriate optimization problems and have been presented in [1]. This paper focuses on the lower level task of designing a control policy for selecting control input signals assuming that the specification is attainable. The task of computing the control signals is performed on-line by solving quadratic and linear programming problems. In addition, we design hybrid controllers for a broad class of specifications that are described by deterministic I/O finite automata. Our methodology results in controllers that can force events and guarantee the eventual execution of actions. This framework leads naturally to an input-output representation of the constituent systems which is more similar to classical control design than the supervisory control framework. Note that this paper reports results from [2]. Early results and applications of the methodology have been reported in [3, 4, 5]. Software

tools and the extension of the approach to a class of uncertain hybrid systems have been presented in [6, 7] respectively.

The approach presented in this paper is directly related to supervisory control framework for hybrid systems [8, 9]. Similar approaches based on approximations of the continuous dynamics by a discrete event system have also been proposed in [10, 11, 12, 13]. Supervisory control of hybrid systems is based on the fact that if undesirable behaviors can be eliminated from the discrete model then these behaviors can likewise be eliminated from the actual system. Therefore, the supervisory control framework is not suitable when we want to guarantee that the plant will achieve its goals and it will eventually execute the desired actions. A related approach for hierarchical control of hybrid systems based on the notion of dynamical consistency has been studied in [14]. The use of dynamical consistency aims at the computation of abstractions that preserve the controllability properties of hybrid control systems. A lattice of hierarchical partitions is defined in [14] and used to investigate dynamical consistency. However, no constructive algorithms for the computation of the partitions are given.

The first investigations of piecewise linear hybrid systems can be found in [15, 16, 17]. The main problems studied in this framework were stability, controllability, and input-output regulation. Piecewise linear hybrid systems also arise in the switching control paradigm [18] where the behavior of the plant is controlled by switching between different controllers for each region of the state space. It should be noted that the class of piecewise linear systems has been studied extensively in the circuit theory community; see for example [19] and the references therein. Here, we are interested in approaches that have been developed for modeling, analysis, and synthesis of hybrid control systems. Piecewise linear dynamical systems have been considered also in [20] where a methodology for approximating the reachable states is developed and a supervisory control framework is used for controller design. Mixed logical dynamical systems [21] can represent hybrid systems consisting of linear dynamic equations interacting with linear threshold events, automata, and logic propositions. A comparison between these models and piecewise linear systems can be found in [22]. Methods and tools for optimal control of such systems have been developed based on optimization algorithms that involve real and integer variables including an approach driven by the computation of reachable sets [23, 24]. Piecewise linear systems were also studied in [25] to develop computational algorithms for the analysis of nonlinear and uncertain dynamical systems.

The hybrid system model used in this paper can be viewed as a input-output hybrid automaton evolving in discrete-time. Hybrid automata provide a general modeling formalism for the formal specification and algorithmic analysis of hybrid systems [26]. Formalisms for input/output hybrid automata have been also proposed in [27, 28]. Here, we consider a larger class of inputs which may contain both discrete and continuous control inputs as well as discrete and continuous disturbances. Computational methods for reachability analysis of hybrid systems have been also presented in [29, 30] where the continuous flow of the hybrid system with arbitrary dynamics is approximated using polygonal flow pipes. Finite-state approximations are then used for the verification of the hybrid system properties. Safety and reachability of piecewise linear hybrid systems based on discrete abstractions of the continuous dynamics that does not require approximation of reachable sets has been presented in [1]. Computation of safe sets for a class of discrete-time linear

systems using an elimination of quantifiers approach has been also presented in [31]. Conditions for reachability of continuous-time piecewise linear hybrid systems on simplices and rectangles have been presented in [32]. Finally, several approaches for optimal control of switched and hybrid systems have been recently proposed [33, 34, 35]. In these approaches, the control objective is to minimize a predetermined performance cost and is achieved by making simplifying assumptions that restrict the switching sequences of the system. In our approach, the control specifications are described by a finite automaton, the performance costs are computed on-line, and all feasible switching sequences are allowed.

This paper is organized as follows. In Section 2, we present the modeling framework for piecewise linear hybrid dynamical systems. The regulator problem for hybrid systems is formulated in Section 3. Section 4 presents the notion of attainability as well techniques for testing attainability based on linear programming. Section 5 presents the control design framework. Finally, concluding remarks are presented in Section 6.

2 PIECEWISE LINEAR HYBRID DYNAMICAL SYSTEMS

In the following, we define the class of *piecewise linear hybrid dynamical systems*. The main characteristic of this class is that the continuous dynamics are described by linear difference equations, the discrete dynamics by finite automata, and the interaction between the continuous and the discrete part is defined by piecewise linear maps. First, we present some basic notions and the necessary notation that are used in the modeling formalism.

A *piecewise-linear (PL) subset* [16] of a finite dimensional vector space V is the union of a finite number of sets defined by (finitely many) linear equations $f(x) = a$ and linear inequalities $f(x) > a$. A *PL relation* $R \subseteq X \times Y$ between PL sets is one whose graph is a PL set. A *PL map* is defined similarly. Equivalently, the map $f : X \to Y$ is PL if there exists a partition of X by PL subsets X_i such that the restrictions $f|X_i$ are all affine (linear + translation).

Our control design approach is based on PL partitions of the state space. Consider the collection $\{h_i\}_{i=1,2,\ldots,\ell}, \quad h_i : \Re^n \to \Re$ of real-valued functions of the form $h_i(x) = g_i^T x - w_i$, where $g_i \in \Re^n$ and $w_i \in \Re$. Let

$$H_i = \ker(h_i) = \{x \in \Re^n : h_i(x) = g_i^T x - w_i = 0\}$$

and assume that H_i is an $(n-1)$-dimensional hyperplane ($\nabla h_i(x) = g_i^T \neq 0$). We define the function $\hat{h}_i : \Re^n \to \{-1, 0, 1\}$ by

$$\hat{h}_i(x) = \begin{cases} -1 & \text{if} \quad h_i(x) < 0 \\ 0 & \text{if} \quad h_i(x) = 0 \\ 1 & \text{if} \quad h_i(x) > 0 \end{cases}$$

Then, a PL partition is defined by $\pi(x) = [\hat{h}_1(x), \ldots, \hat{h}_\ell(x)]^T$. The mapping π defines an equivalence relation E_π on the set X in the natural way $x_1 \, E_\pi \, x_2$ iff $\pi(x_1) = \pi(x_2)$. The image of the mapping π is called the *quotient space* of X by E_π, is denoted by X/E_π, and can be represented as $X/E_\pi = \{P_i\}, \; i = 1, \ldots, |\pi|$ where each P_i corresponds to a polyhedral region of $\Re^n$ and $|\pi|$ denotes the number of equivalence classes.

Let $X \subseteq \Re^n$ denote the continuous state space, Q the finite set of discrete states or *modes* of the system, $U \subset \Re^m$ the continuous input space, Σ the set of input events, and Y the output set of the hybrid system. Often, it is desirable to distinguish between controlled and uncontrolled inputs, and we may include a space of continuous disturbances $D \subset \Re^p$ which are assumed to be unknown but measurable. It is assumed that the input set U and the disturbance set D are bounded. Furthermore, the set of input events can be written as $\Sigma = \Sigma_c \cup \Sigma_u$. The set Σ_c represents the *controllable* events which are associated with discrete state transitions which can be issued by a control mechanism. The set Σ_u contains the *uncontrollable* events generated by the environment. Note that this definition is different than the definition of supervisory control [36] where uncontrollable events are events that can be disabled by the controller. The output set may also contain a discrete and a continuous part.

Definition 1 A *piecewise linear hybrid dynamical system* (*PLHDS*) is defined by

$$x(t+1) = A_{q(t+1)}x(t) + B_{q(t+1)}u(t) + E_{q(t+1)}d(t) \quad (1)$$

$$q(t+1) = \delta(q(t), \pi(x(t)), \sigma_c(t), \sigma_u(t)),\ q(t+1) \in \text{act}(\pi(x(t))) \quad (2)$$

$$y(t) = g(q(t), x(t)) \quad (3)$$

where $x(0) = x_0 \in \Re^n, q(0) = q_0 \in Q$ and $\pi : X \rightarrow X/E_\pi$ partitions the continuous state space $\Re^n$ into polyhedral equivalence classes, act : $X/E_\pi \rightarrow 2^Q$ defines the active mode set for every region of the partition, $A_q \in \Re^{n\times n}, B_q \in \Re^{n\times m}$, and $E_q \in \Re^{n\times p}$ are the system matrices for the discrete state q, $\delta : Q \times X/E_\pi \times \Sigma_c \times \Sigma_u \rightarrow Q$ is the discrete state transition function, and $g : Q \times X \rightarrow Y$ is the output function which is assumed to be piecewise linear.

The dynamic evolution of the system is defined as follows. A change in the discrete state of the system can be caused by two type of events. First, an input event generated by either the controller or the environment. Second, an event generated by the continuous dynamics when the continuous state enters a polyhedral region of the continuous state space defined by the partition. The set of events generated by the continuous dynamics is called the set of *plant events*. After a discrete transition, the system is at mode (discrete state) q and the continuous state evolves according to the difference equation (1) driven by the control input $u(t)$.

The interaction between the discrete and continuous dynamics is defined as follows. For each discrete mode, we assign a region of the state space using the mapping inv : $Q \rightarrow 2^{X/E_\pi}$. The continuous state may evolve according to the difference equation determined by the discrete state q only if $x(t) \in \text{inv}(q)$. The regions inv(q) are often called *invariants*. In our modeling framework, the invariants do not necessarily correspond to disjoint regions of the state space. The system may switch from q_1 to q_2 at time t upon receiving an external command $\sigma(t)$ if $x(t) \in \text{inv}q_1 \cap \text{inv}q_2$. An alternative way to describe the notion of invariants that will be useful in our analysis is by defining the set of feasible modes for each region of the primary partition. The *active mode set* is defined by the mapping act : $X/E_\pi \rightarrow 2^Q$. From the definition of the invariants and the active mode sets, it follows that for each discrete state $q \in Q$ and for each region of the primary partition $P \in X/E_\pi$ we have $P \in \text{inv}(q) \Leftrightarrow q \in \text{act}(P)$.

Assume that the current discrete state is q and that $q' \in \text{act}(\pi(x(t)))$ for some state $x(t) \in \Re^n$, then q' is a possible new state, and the transition $q \rightarrow q'$ (or (q, q')) may occur. These are discrete state transitions that are associated either with a controllable event $\sigma_c \in \Sigma_c$ or an uncontrollable event $\sigma_u \in \Sigma_u$. A controllable event is issued by a control mechanism and forces the transition to occur. An uncontrollable event is generated by the environment and can also force a discrete state transition. As it is described in the previous definition, the discrete state transition function is assumed to be deterministic which means that for a given plant event or input the next discrete state can be uniquely determined. The following definition guarantees that for every state in the region R there is a possible evolution of the system.

Definition 2 A *region* of the state space is defined as $R = (M, P)$ where $M \subseteq Q$ is a set of modes and $P \subset \Re^n$ is a piecewise linear set and for every $x \in P$ there exists $q \in M$ such that $q \in \text{act}(\pi(x))$.

Remark The state transitions are synchronized by a clock. At every clock tick an input event may be triggered and an event caused by the continuous dynamics may occur. Therefore, every change in the state occurs synchronously to a clock. Since the hybrid model evolves in discrete-time, the generator will not be able to identify the exact moment that a hyperplane is crossed. It identifies the first sample after a crossing has occurred. In many physical systems, however, events occur asynchronously at time instants that do not necessarily coincide with the clock ticks. Discrete-time systems can be used as approximations of physical processes. The approximation is based on the assumption that events that occur asynchronously are detected in the next clock tick (using digital computers). In many situations, the discrepancy in the time instants of the event occurrences can be studied by considering continuous disturbances in the model. Discrete-time modeling offers significant computational advantages, however, it cannot be used to study the behavior of the system between sampling instants. For example, it is possible that a sequence of two or more plant events will occur in a sampling interval. In our model, it is assumed that the plant events are generated based only on the value of the state at the sampling instants.

Example We present a tank system to illustrate the piecewise linear hybrid model and to demonstrate our approach later in the paper. This example has been proposed as a benchmark for control reconfiguration in [37] and has been used for demonstrating estimation, fault detection, and control reconfiguration methods for hybrid systems in [38, 39]. The system consists of three identical cylindrical tanks filled with water. The tanks are connected by two pipes at levels 0 and h as shown in Figure 1. Tank 3 is used as redundant hardware in the case tank 1 fails. The input flow Q_{in} is provided by a pump to tank 1. Switching of the valves controls the flow between the pipes and the outflow. We consider a configuration where we can switch only V_a while V_1 and V_2 are always open. We assume that the flow q through a valve is linearly related to the water level across the valve Δx according to $q = \frac{\Delta x}{R}$ where R is the valve resistance. The flow through the valve V_a, if it is open, depends on the water levels x_1 and x_2 at tank 1 and tank 2 respectively as

$$Q_a = \begin{cases} 0 & \text{if } x_1 \leq h \wedge x_2 \leq h \quad \text{(mode 1)} \\ \frac{x_1 - h}{R_a} & \text{if } x_1 > h \wedge x_2 \leq h \quad \text{(mode 2)} \\ -\frac{x_2 - h}{R_a} & \text{if } x_1 \leq h \wedge x_2 > h \quad \text{(mode 3)} \\ \frac{|x_1 - x_2|}{R_a} & \text{if } x_1 > h \wedge x_2 > h \quad \text{(mode 4)} \end{cases}$$

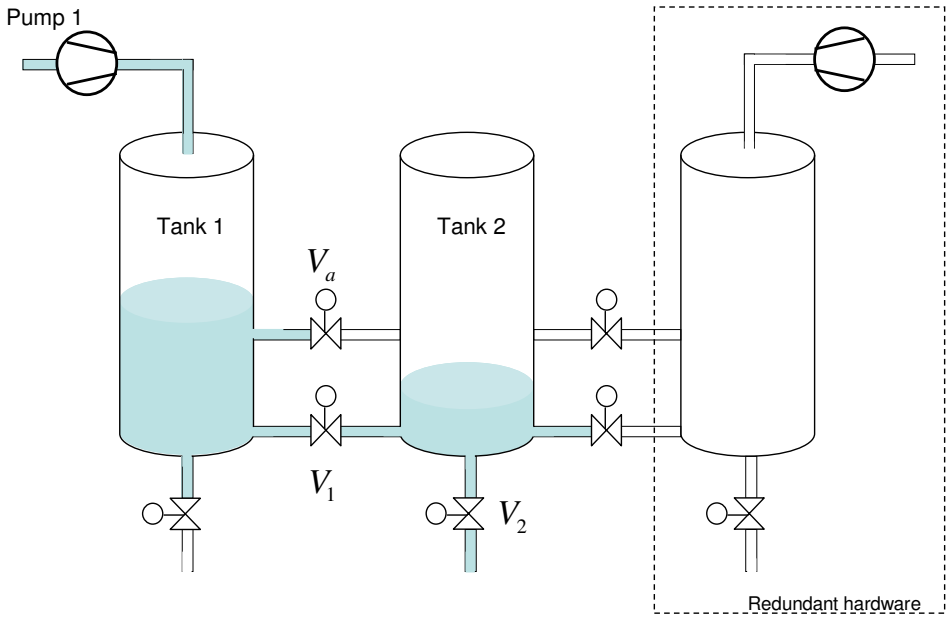

Figure 1: Tank system

Table 1: Parameters for the tank system

PARAMETER	DESCRIPTION	VALUE
T_s	sampling period	0.1 sec
$R_1 = R_2$	valve resistance	$5 \cdot 10^3 m^{-2} sec$
R_a	valve resistance	$10 \cdot 10^3 m^{-2} sec$
h	height of the connecting pipe for valve V_a	0.3 m
A	base area of each tank	0.0154 m^2
$Q_{in,max}$	maximum input flow	$0.1 \times 10^{-3} m^3/sec$

If V_a closes, the system transitions to the mode $q = 1$ where there is no flow through V_a. Figure 2 shows a hybrid system model for the tank system where the continuous dynamics at each mode describe the rate of change of the water level $\dot{x}_i = (\text{in-flow}_i - \text{out-flow}_i)/A$ where A is the base area of each tank. Discrete-time representations of the continuous dynamics are obtained using zero-order hold sampling. The transition between modes are triggered either by plant events, for example, when $x_1 > h$ or by control events, for example, when a control command closes V_a. The parameters of the tank system are summarized in Table 1.

Let's assume that for safety we require that the water level x_1 is between 0.3 and 0.5 and the water level x_2 is between 0.2 and 0.5. This is a safety specification described by the set $P = \{x \in \Re^2 : 0.3 \leq x_1 \leq 0.5 \wedge 0.2 \leq x_2 \leq 0.5\}$. A partition of the continuous state space is obtained by considering the hyperplanes that define the mode transitions and the safety specification and it is shown in Figure 3. The safe set P consists of two polyhedral regions $P_1 = \{x \in \Re^2 : 0.3 \leq x_1 \leq 0.5 \wedge 0.2 \leq x_2 \leq 0.3\}$, and $P_2 = \{x \in \Re^2 : 0.3 \leq x_1 \leq 0.5 \wedge 0.3 < x_2 \leq 0.5\}$. The active mode sets are $\{q_1, q_2\} \in \text{act}(P_1)$, $\{q_1, q_4\} \in \text{act}(P_2)$ respectively.

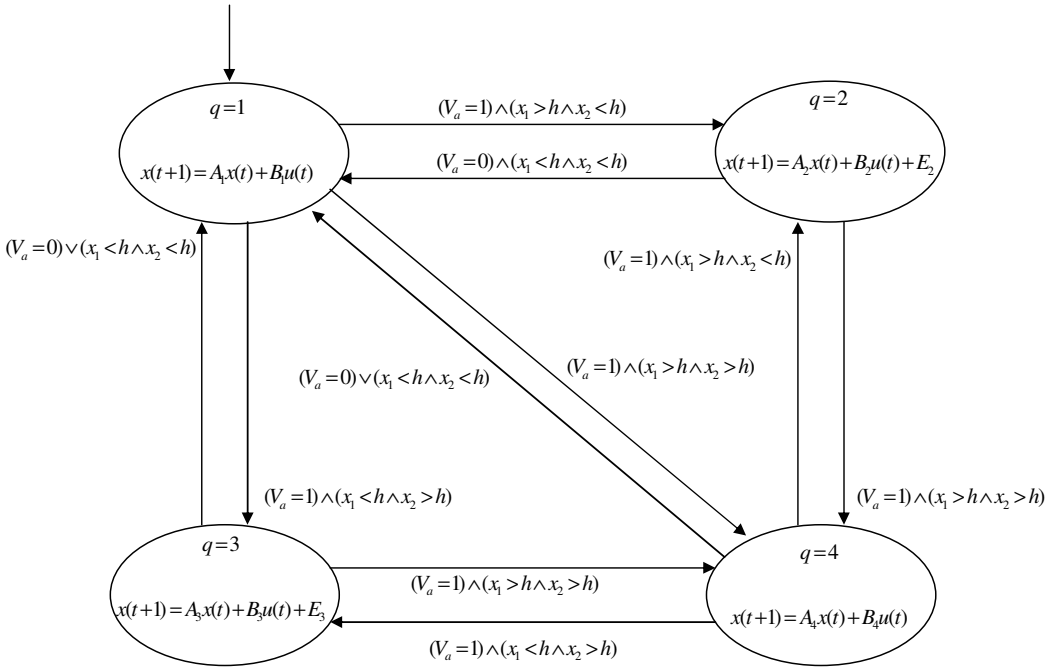

Figure 2: Hybrid model of the tank system

3 HYBRID SYSTEM REGULATOR

In this section, we present a hierarchical control framework for hybrid systems based on a formulation of the regulator problem. In general, a regulator requests certain types of outputs from the plant so that these are attained in the presence of disturbances. The desired outputs can be described as the outputs of another dynamical system, called the *exosystem*. In this paper, we assume that the exosystem is represented by a deterministic input-output (I/O) finite automaton. The control specifications are described using the language generated by the automaton. The objective is to design a controller so that the closed loop system shown in Figure 4 follows the behavior of the exosystem. Note that by convention it is assumed that the disturbances, which are unknown but measurable, are described as outputs of the exosystem.

The control specifications are modeled 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, $\delta_e : X_e \times V_e \rightarrow X_e$ is the state transition function, $\lambda_e : X_e \rightarrow Y_e$ is the output function returning the output associated with each state, and R_0 is the initial state.

The set of states is defined as $X_e = \{R_1, R_2, \ldots, R_M\}$ where $R_i = (Q_i, P_i)$ 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 $R_i \subseteq Q \times X/E_\pi$. Note that in order to reduce the number of states of the finite automaton that models the specifications, each region R_i may contain more than one discrete modes and/or more than one regions of the continuous state space if those are adjacent. The regions R_i are disjoint as subsets of the hybrid state space $Q \times X$. Therefore, each state (q, x) corresponds to exactly one region R_i. We assume that the function δ_e is *non-total*, which means that not every input can be applied

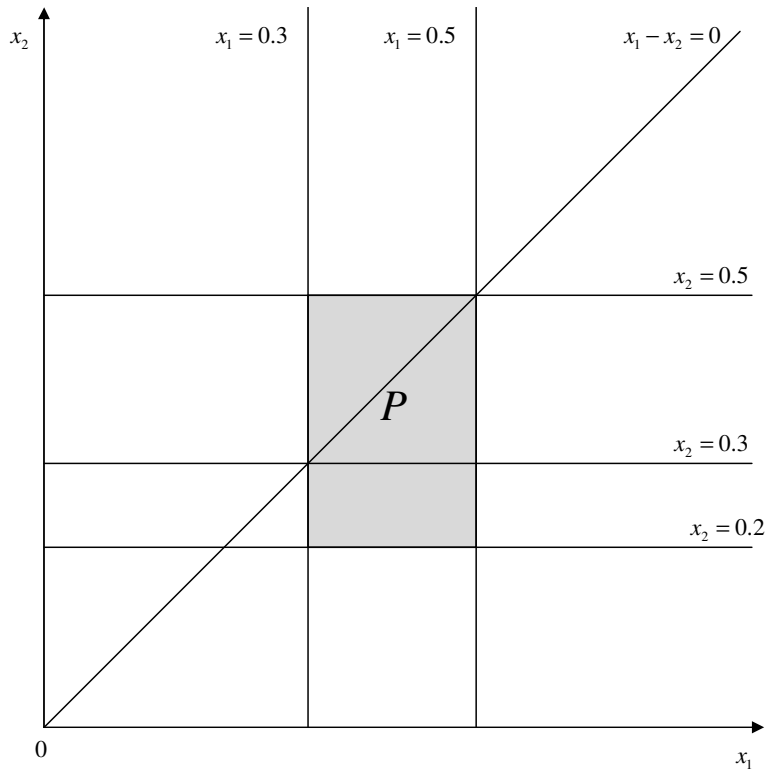

Figure 3: Partition for the tank system

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. Note that the I/O finite automaton which describes the specifications is a deterministic Moore automaton [40].

The behavior of the exosystem is described by $B_e \subseteq (V_e \times Y_e)^I$ and consists of all the input-output pairs that the exosystem can generate [41]. The "time" axis is described by the index set I which represents an ordering of events related to the specifications. The specification is defined only with respect to the output set and it is represent by $B_{sp} \subset Y_e^I$. The output behavior of the exosystem can be described by the set of sequences of symbols from Y_e that it can generate. Denote by Y^* the set of all strings formed by concatenation of symbols from the output alphabet Y where the * operation is called the Kleene closure [40]. A *language* is formally defined as a subset of Y_e^*. The output behavior of the exosystem can be described by the output language $L \subset Y^*$. The usual set operations, such as union, intersection, difference, complement (with respect to Y^*) are applicable to languages, details can be found in [40]. In addition, the *prefix-closure* of L, denoted by $\bar{L}$ is defined as the set of all prefixes of strings in L. The language L is said to be *prefix-closed* if all the prefixes of the language are also in L, or equivalently if $L = \bar{L}$. Note that the language describing the output behavior is different than the language *accepted* by an automaton. The former is defined with respect to the output symbols generated by the output function, while the latter is defined with respect to the input symbols and the state transition function [40]. The language generated by the exosystem is defined as follows.

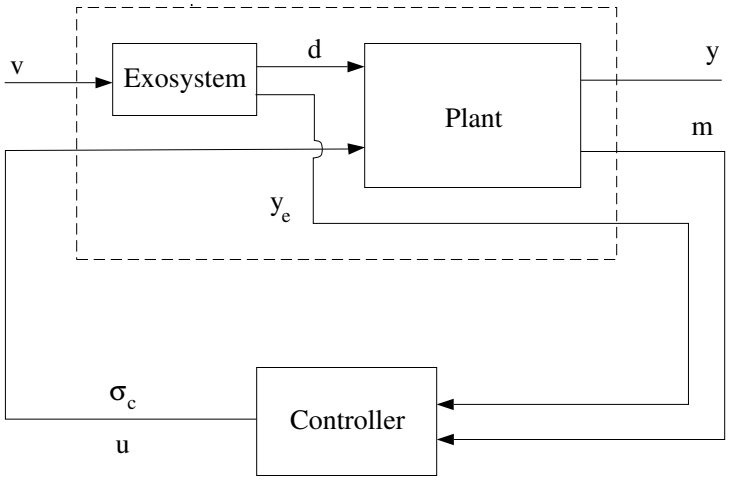

Figure 4: Hybrid system regulator.

Definition 3 Given the finite set Y_e and the sequence $y : I \rightarrow Y_e$, then $y \in L$ if there exist $x_e \in X_e$ and $v_e \in V_e$ such that the following conditions hold:

$$\begin{aligned} x_e[n+1] &= \delta_e(x_e[n], v_e[n]), \quad \forall n \in I \\ y[n] &= \lambda_e(x_e[n]), \forall n \in I \end{aligned}$$

Each specification can be described by a language $K \subset L$. A language generated by the exosystem may contain two types of symbols. First, it may contain *terminating* output symbols that represent safe regions of the state space. If the hybrid state reaches a safe region then it remains in that region indefinitely. Note that this blocking behavior describes the safety of a region and it is not undesirable as in discrete event systems. Second, the language K may contain *non-terminating* output symbols. The transition from one state of the exosystem to the next represents reachability for the corresponding regions of the hybrid state space.

Example We consider the tank system presented in Section 2. The safety specification for the system is described by the region

$$\begin{aligned} R_1 &= \Big(\{q_1, q_2\}, \{x \in \Re^2 | 0.3 < x_1 \leq 0.5 \wedge 0.2 < x_2 \leq 0.3\}\Big) \\ &\cap \Big(\{q_1, q_4\}, \{x \in \Re^2 | 0.3 < x_1 \leq 0.5 \wedge 0.3 < x_2 \leq 0.5\}\Big). \end{aligned} \tag{4}$$

Dynamic specifications involve also sequencing of events. For example, in the startup procedure, we may require that x_1 crosses the connecting pipe at $h = 0.3$ before $x_2 > 0$. We describe the additional regions as

$$R_2 = \Big(\{q_1, q_2\}, \{x \in \Re^2 | 0.3 < x_1 \leq 0.5 \wedge 0 < x_2 \leq 0.2\}\Big), \tag{5}$$

and

$$R_3 = \Big(\{q_1\}, \{x \in \Re^2 | 0 < x_1 < 0.3 \wedge 0 < x_2 < 0.2\}\Big). \tag{6}$$

The specifications are modeled by the finite automaton shown in Figure 5. The output function is defined by $\lambda(R_3) = a,\ \lambda(R_2) = b,$ and $\lambda(R_1) = c$. We consider the following specifications that can be described

using the exosystem shown in Figure 5. For safety the desired behavior is described by the language $K_1 = c$, where c is viewed here as a constant function from the index set I to Y_e. During the startup procedure the desired behavior can be described by the language $K_2 = abc$. This a piecewise constant function from I to Y_e. During the operation of the system we may require a periodic behavior described by the language $K_3 = \overline{(abcb)^*}$.

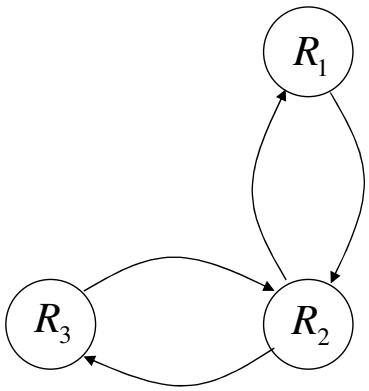

Figure 5: Exosystem for the tank system.

The problem considered in this paper is the design of a controller so that the closed loop system shown in Figure 4 exhibits the same behavior as the exosystem. Next, we formally define the composition of the plant with the controller and the behavior of interest of the closed loop system.

Plant The plant is a piecewise linear hybrid dynamical system represented by $\mathcal{P} = (X_p, U_p, Y_p, M_p; f_p, g_p, m_p)$ where $X_p = Q \times X$ is the hybrid state space, $U_p = (\Sigma_u \times \Sigma_c) \times (D \times U)$ is the input set, Σ_u is the set of events generated by the environment and Σ_c the set of events generated by the controller, D is the set of continuous disturbances, U is the set of continuous control inputs, $Y_p = \{R_1, \ldots, R_M\}$ is the output set consisting of the regions of the hybrid state space which are used to describe the control specifications, and $M_p = Q \times X$ is the measurement space. The hybrid state transition function $f_p : X_p \times U_p \to X_p$ is described by (1) and (2). The output function $g_p : X_p \to Y_p$ described by (3) is implemented by a filter that determines the membership of the state into a region R_i of the state space. The measurement function is $m_p : X_p \to M_p$ assuming full state feedback.

Controller The controller is represented by $\mathcal{C} = (X_c, U_c, Y_c; f_c, g_c)$ where X_c is the set of states, $U_c = M_p$ is the input set which coincides with the measurement set of the plant, $Y_c = \Sigma_c \times U$ is the output of the controller consisting of the continuous control input and the controller events that may trigger a (feasible) discrete transition, $f_c : X_c \times U_c \to X_c$ is the state transition function, and $g_c : X_c \times U_c \to U_c$ is the output function of the controller.

Composition The plant and the controller are connected in a feedback configuration as shown in Figure 4. The closed loop system is described by $\mathcal{CL} = (X_{cl}, U_{cl}, Y_{cl}; f_{cl}, g_{cl})$ where $X_{cl} = X_p \times X_c$ is the state set of the closed loop system, $U_{cl} = \Sigma_u \times D$ is the set of exogenous inputs, $Y_{cl} = Y_p$ is the output set, $f_{cl} : X_{cl} \times U_{cl} \to X_{cl}$ is the state transition function, and $g_{cl} : X_{cl} \times U_{cl} \to Y_{cl}$ is the output

function of the closed loop system. Since the control requirement is for the closed loop system to exhibit the same (output) behavior as the exosystem, we define the behavior of the plant with respect to the output set $Y_{cl} = \{R_1, \ldots, R_M\}$. Since the closed loop system is a discrete-time dynamical system, in order to compare the behaviors of the event-driven exosystem and the time-driven closed loop system, we abstract the time information by defining an index function α_{cl}. Consider the index $k \in \mathbf{N}$. The state of the discrete-time dynamical system is associated with the index k meaning the state at time $t = kT$. Then define $\alpha_{cl} : \mathbf{N} \rightarrow I$ as follows:

$$\alpha_{cl}[0] = 0 \tag{7}$$

$$\alpha_{cl}[n] = \min\{t = kT > \alpha_{cl}[n-1] : y_{cl}(t) \neq y_{cl}(\alpha_{cl}[n-1])\} \tag{8}$$

Let $I = \text{Im}\alpha_{cl}$ denote the image of the function α_{cl}, then the behavior of the closed loop system is defined as $B_{cl} \subseteq Y_{cl}^I$.

Definition 4 Given the exosystem $\mathcal{E}$ and the piecewise linear hybrid dynamical system $\mathcal{P}$, we say that the *regulator problem* has a solution if there exists a controller $\mathcal{C}$ such that the output behavior of the closed loop system follows the specified behavior of the exosystem, that is $B_{cl} = B_{sp}$.

The design of the controller can be decomposed in two levels. In the higher level, we are concerned only with conditions for the existence of appropriate control inputs for safety and reachability specifications. Such conditions are based on the feasibility of appropriate optimization problems [1] and can be tested off-line. Assuming that there exists appropriate control resources to satisfy the specifications, the implementation of the controller and therefore, the selection of the actual control input signal is done by solving on-line optimization problem. In this paper, we focus on the second problem of selecting control inputs by solving appropriate optimization problems. The solution to the first problem has been presented in [1] where conditions for the existence of appropriate control inputs for safety and reachability specifications.

4 ATTAINABILITY AND THE REGULATOR PROBLEM

Our control objective is that the closed loop system consisting of the plant and the controller 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 on the plant by a control mechanism*.

Definition 5 The specification behavior B_{sp} is said to be *attainable* if there exists solution to the regulator problem, that is a controller such that the output behavior of the closed loop system satisfies $B_{cl} = B_{sp}$.

Practically, the attainability of the specification behavior B_{sp} can be tested off-line using algorithms for reachability analysis of piecewise linear hybrid systems. Given the region $R = (M, P)$, we consider the

predecessor operator pre : $2^{Q\times X} \rightarrow 2^{Q\times X}$ to compute the set of states for which there exists a control input so that the state will be driven in R for every disturbance. The action of the operator is described by

$$\text{pre}(R) = \{q \in M\} \times \{x \in X | \exists u \in U, \forall d \in D, A_q x + B_q u + E_q d \in P\}.$$

The set pre(R) is piecewise linear and can be always represented using only linear equalities and inequalities. Such a description is based on the fact that *piecewise-linear algebra* admits elimination of quantifiers [16] which means that any PL set defined using quantifiers can be also defined using only propositional connectives. The elimination of quantifiers can be performed using Fourier-Motzkin elimination [42] for computing appropriate projections, linear programming techniques for eliminating redundant constraints, and equivalences from predicate logic [43] to combine the constraints. Details can be found in [1].

Since the set pre(R) is piecewise linear, we can apply the predecessor operator recursively to obtain

$$\text{pre}^N(R) = \overbrace{\text{pre}(\cdots \text{pre}(R))}^{N \text{ times}}.$$

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 represented by successive application of the predecessor operator $CR(R) = \text{pre}^*(R)$ where $*$ denotes the fixed point of the predecessor operator. It should be noted that the algorithm for the computation of the coreachable set for a region R is semi-decidable. The procedure produces the correct answer if it terminates, but its termination is not guaranteed. Infinite time problems for piecewise linear systems are, in general, undecidable [17]. For finite time problems, backward reachability algorithms for piecewise linear hybrid systems are NP-complete [17]. This follows from the definition of the predecessor operator which is formulated using the existential quantifier over all possible inputs. The number of linear constraints that are used to represent the coreachable region grows exponentially at every iteration of the algorithm. Practically, many constraints are redundant and can be eliminated by performing Fourier-Motzkin elimination at every time step. For example, reachability analysis between two regions R_1 and R_2 is based on the computation of the set pre$(R_2) \cap R_1$ at every time step. Some of the constraints for pre(R_2) will be redundant and can be eliminated for the computation of the set pre(pre$(R_2) \cap R_1$). The elimination of redundant constraints is itself computationally expensive, however, the attainability tests are performed only at design-time. If the specification is attainable, the run-time controller presented in Section 5 needs to perform only one-step optimization algorithms that are very efficient.

Testing attainability involves testing safety for the regions that correspond to terminating symbols of the specification language K and reachability for non-terminating symbols. In the following, we briefly present the related results from [1].

Definition 6 [1] Given a set of safe 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.

Lemma 1 *[1] A PLHDS is safe with respect to the region $R \subseteq Q \times X$ if and only if $R \subseteq$ pre(R).*

The safety condition is illustrated in Figure 6 where the piecewise linear set pre(R) contains the safe set R. Let $R|X$ and pre$(R)|X$ be the projection of R and pre(R) into the continuous state space X. The sets $R|X$ and pre$(R)|X$ are piecewise linear but not polyhedral, and therefore they are not necessarily convex. In order to test whether $R|X \subseteq$ pre$(R)|X$, we represent the constraints in disjunctive normal form (DNF) and we test the feasibility of finite set of linear programming problems.

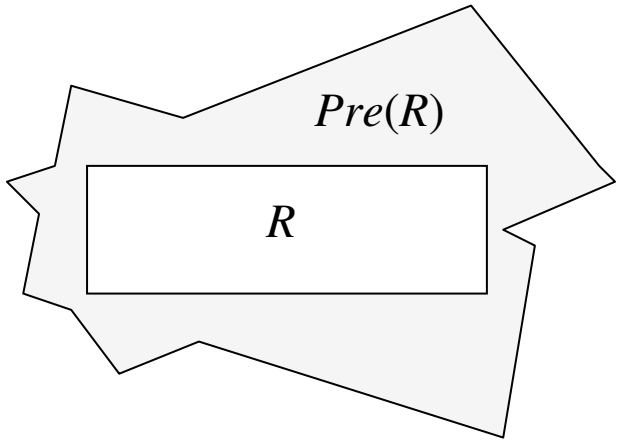

Figure 6: Safety condition.

Next, we consider non-terminating symbols of the specification language K and the corresponding regions of the state space. 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 while staying in $R_1 \cup R_2$. 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. Further, we only consider regions of the form $R_1 = (Q_1, P_1)$ and $R_2 = (Q_2, P_2)$ for which P_1 and P_2 are adjacent polyhedral regions of the primary partition. In this case, the regions P_1 and P_2 have a common boundary which is represented by a $(n-1)$-dimensional hyperplane $h(x) = g^T x - w$. The reachability problem between any two regions can be solved by finding a path consisting of adjacent reachable regions.

The problem of deciding if 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. Our approach is based on conditions that guarantee that state can be forced to cross the hyperplane $h(x)$ in finite time by selecting appropriate controls. For this purpose, we consider a finite time horizon defined by NT where T is the sampling period and $N \in \mathbf{N}$. Consider a PLHDS described by the equations (1)-(3) and assume that the initial condition is $(q(t_0), x(t_0)) \in R_1$.

Definition 7 [1] The region R_2 is *directly N-reachable* from R_1 if for every initial state $(q(t_0), x(t_0)) \in R_1$ there exist control inputs for the PLHDS and $k \in \mathbf{N}, 0 < k \leq N$ so that $(q(t), x(t)) \in R_1$ for $t_0 \leq t < t_0 + kt$ and $(q(t_0 + kt), x(t_0 + kt)) \in R_2$.

We denote by $CR^N_{R_1}(R_2)$ the *coreachable set* of all states that can be driven from the region R_1 to R_2 in the finite time $t \leq NT$ without entering a third region. The set $CR^N_{R_1}(R_2)$ can be computed using the

predecessor operator pre : $2^{Q\times X} \rightarrow 2^{Q\times X}$. Given the regions R_1 and R_2, we compute all the states that can driven from R_1 to R_2. At every iteration k of the algorithm we consider the intersection of the set pre(R^k) with the set R_1 since we are interested only in states that can be driven to R_2 directly from the region R_1 without entering a third region. At every iteration of the algorithm we apply the predecessor operator to a piecewise linear region of the state space and we take the intersection between two piecewise linear sets. Hence, the resulting region is still piecewise linear, it can be represented using only linear equalities and inequalities, and the following holds.

Lemma 2 *[1] Consider a PLHDS described by (1)-(2) and the piecewise linear regions $R_1 = (Q_1, P_1)$ and $R_2 = (Q_2, P_2)$. Then, the region R_2 is directly N-reachable from R_1 if and only if $R_1 = CR^N_{R_1}(R_2)$.*

The N-reachability condition is illustrated in Figure 7 where the region R_2 can be reached from R_1 in at most 4 steps. Since the set $CR^N_{R_1}(R_2)$ is piecewise linear, the reachability problem between R_1 and R_2 can be solved using linear programming techniques similarly to the safety conditions.

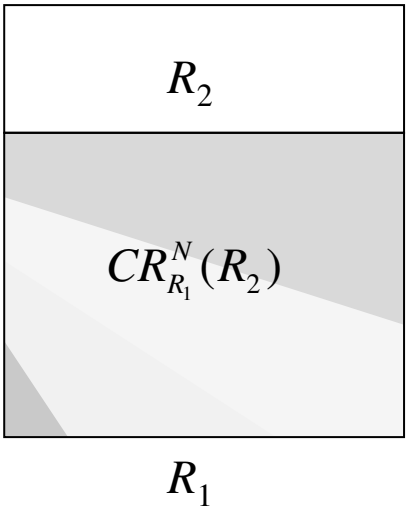

Figure 7: Reachability condition.

Based on the safety and reachability properties describe above, we formulate attainability conditions for dynamic specifications defined by the languages generated by the exosystem. Let $y = y_0 y_1 \dots y_{n-1} \in K$ denote a sequence of symbols from the specification language K.

Proposition 1 *The specification behavior B_{sp} is* attainable *if and only there exists a language K that satisfies the following conditions:*

(i) $y_0 = g(q_0, x_0) = g(q(0), x(0))$,

(ii) *if y_{n-1} is a terminating output symbol, then for every $t > \alpha_{cl}[n-1]$ there exist control input $u(t)$ and controllable events $\sigma_c(t)$ so that for every disturbance $d(t)$ and uncontrollable event $\sigma_u(t)$, we have that $y(t) = g(q(t), x(t)) = y_{n-1}$, and*

(iii) if y_{n-1} is not a terminating output symbol, then there exists $t' > \alpha_{cl}[n-1]$ so that for every t such that $\alpha_{cl}[n-1] < t < t'$, there exist $u(t)$ and $\sigma(t)$ such that for every $d(t)$ and $\sigma_u(t)$ we have $y_n \in K$, $\alpha_{cl}[n] = t'$ where

$$\begin{aligned} y(t') &= g(q(t'), x(t')) = y_n \\ y(t) &= g(q(t), x(t)) = y_{n-1},\ \alpha_{cl}[n-1] < t < t'. \end{aligned}$$

Proof The above conditions require that for every prefix of the specification language K, there exist controls, that will force the next output symbol of the closed loop system to remain in K and therefore, B_{sp} is *attainable*.

Proposition 2 *The specification behavior B_{sp} is attainable if and only if every terminating state y_{n-1} corresponds to a region R_{n-1} that is safe and for every non-terminating state y_{n-1}, there exists y_n so that, for the corresponding regions we have that R_n is directly reachable from R_{n-1}.*

Proof First, we have $y_0 = g(q_0, x_0) \in K$ by the definition of attainability. Next, if y_{n-1} is a terminating output symbol, consider $R_{n-1} \subset Q \times X$ the corresponding region of the state space. Then, the attainability of the language K implies that the region R_{n-1} is safe and by the definition of safety, there exists a control policy that will force the state to remain in R_{n-1} for every disturbance. Finally, if y_{n-1} is not a terminating output symbol, consider the regions R_{n-1} and R_n corresponding to the output symbols y_{n-1} and y_n respectively. Since K is attainable, there exist a control policy so that R_n is reachable from R_{n-1}, and therefore there exist a control policy so that the sequence $y = y_0 y_1 \dots y_{n-1} y_n \in K$.

5 CONTROLLER DESIGN

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. A controller is designed as a dynamical system to implement the desired control policy. We have already shown that if the specification behavior is attainable, there exists control policy so that the closed loop system will satisfy the specification. Our objective is to build a convenient representation of the controller. The design of the controller is based on the regions $\{R_1, \dots, R_M\}$ that are used to define the control specifications. The proposed representation for the controller is shown in Figure 8. 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 which is applied to the hybrid plant. The control input consists of a continuous component $u \in U$ and a discrete component $\sigma_c \in \Sigma_c$ which triggers feasible discrete transitions. In the following, we formally define the controller.

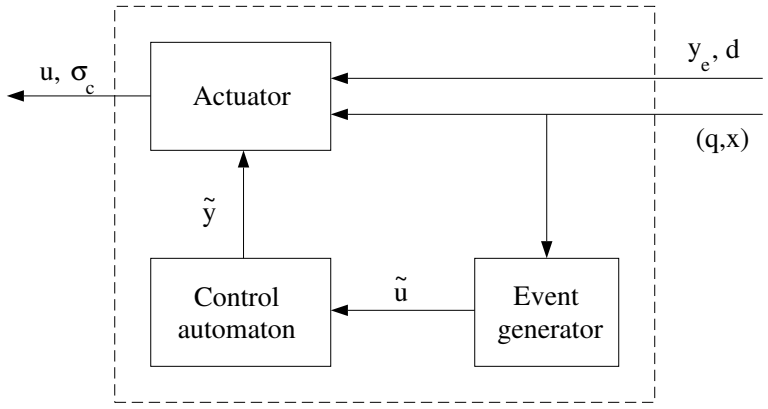

Figure 8: Controller diagram.

Event Generator The event generator abstracts the discrete-time signal $(q(t), x(t))$ from the plant to a sequence of events that describe the membership of the state to the regions R_i. The event generator is defined by the following equations:

$$\begin{aligned} \alpha_c[0] &= 0 \\ \alpha_c[n] &= \min\{t > \alpha_c[n-1] : (q(t), x(t)) \in R_j \neq R_i \ni (q(\alpha_c[n-1]), x(\alpha_c[n-1]))\} \\ \widetilde{u}[n] &= \ell_c(R_i, R_j) \end{aligned}$$

where $\alpha_c : \mathbf{N} \to I$ is an index function representing the order of events and $\ell_c : X_e \times X_e \to \widetilde{U}$ a (non-total) label function assigning an event to every pair of regions with adjacent continuous parts.

Control Automaton The control automaton is an I/O deterministic (Moore) finite automaton [40] defined as $(X_c, \widetilde{U}, \widetilde{Y}, \widetilde{\delta}, \widetilde{\lambda})$ where $X_c = \{R_1, \ldots, R_M\}$ is the set of states, $\widetilde{U}$ is the set of input events, $\widetilde{Y} = \{\widetilde{y}_1, \ldots, \widetilde{y}_M\}$ is the set of output events, $\widetilde{\delta} : X_c \times \widetilde{U} \to X_c$ is the state transition function, and $\widetilde{\lambda} : X_c \to \widetilde{Y}$ is the output function.

The control automaton is deterministic and therefore, the next state can be uniquely determined from the current state and the input event which is an abstraction of the state of the hybrid plant. This is a realistic assumption for practical applications of hybrid systems. The input events represent the measurements from the hybrid plant. An event $\widetilde{u}$ is generated when the state crosses to a new region R_i of the state space.

The Actuator 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 output of the actuator is a discrete-time control signal $(\sigma_c(t), u(t))$. 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. Consider the specification behavior described by the language $K = y_0 y_1 \ldots y_{n-1},\ y_i \in Y_e$ and let R_i be the corresponding region to the output symbol y_i.

First, we consider terminating output symbols that represent safety conditions for the corresponding region of the state space. Let y_{n-1} be a terminating state and $R_{n-1} = (S_{n-1}, P_{n-1}) \subset Q \times X$ the

corresponding region. We define the cost functional $J_{n-1} : Q \times \Re^n \times \Re^m \times \Re^p \to \Re$

$$J_{n-1}(q, x, u, d) = ||A_{q(t)}x(t) + B_{q(t)}u(t) + E_{q(t)}d(t) - \bar{x}_{n-1}||_2^2 \tag{9}$$

where $\bar{x}_{n-1} \in P_{n-1}$ is a point selected by the designer. For example, $\bar{x}_{n-1}$ can be selected as the epicenter of P_{n-1}. It should be noted that he cost functional can include weighting matrices for the case where the states have different physical units or importance. The control signal is selected as the solution to the following optimization problem:

$$\begin{array}{c} \min \\ q \in \text{act}(\pi(x(t))) \\ u \in U \end{array} \quad \begin{array}{c} J_{n-1}(q, x, u, d) \\ \text{s.t. } A_q x(t) + B_q u(t) + E_q d(t) \in P_{n-1} \end{array} \tag{10}$$

The constraints and the cost functional are computed at every time step. The control action is selected by solving a quadratic programming problem for each feasible discrete mode $q \in \text{act}(\pi(x(t)))$. Let q' be the mode that corresponds to the minimum cost, then the control input is selected as $(\sigma_c(t), u(t))$ where $q' = \delta(q(t), \pi(x(t)), \sigma_c(t), \epsilon)$ and $u = \text{argmin}_{u \in U} J_{n-1}(q', x, u, d)$.

Next, we consider two non-terminating output symbols y_k and y_{k+1} which describe a reachability specification between the regions $R_k = (S_k, P_k)$ and $R_{k+1} = (S_{k+1}, P_{k+1})$. The control objective is to drive every state in R_k to R_{k+1}. As it is explained in Section 4, we can assume that P_k and P_{k+1} are adjacent polyhedral regions of the state space and we denote $h(x) = g^T x - w$ their common boundary. Let P be the polyhedral set defined by all the linear constraints that define P_k except $h(x)$. It is assumed without loss of generality that $h(x) > 0$ for every $x \in P$. We define the cost functional $J_h : Q \times \Re^n \times \Re^m \times \Re^p \to \Re$

$$J_h(q, x, u, d) = g^T[A_{q(t)}x(t) + B_{q(t)}u(t) + E_{q(t)}d(t)] - w. \tag{11}$$

The control signal is selected as the solution to the following optimization problem:

$$\begin{array}{c} \min \\ q \in \text{act}(\pi(x(t))) \\ u \in U \end{array} \quad \begin{array}{c} J_h(q, x, u, d) \\ \text{s.t. } A_q x(t) + B_q u(t) + E_q d(t) \in P \end{array} \tag{12}$$

Similarly, this problem can be solved very efficiently by solving a linear programming problem for each feasible discrete mode $q \in \text{act}(\pi(x(t)))$, Let q' be the mode that corresponds to the minimum cost, then the control input is selected as $(\sigma_c(t), u(t))$ where $q' = \delta(q(t), \pi(x(t)), \sigma_c(t), \epsilon)$ and $u = \text{argmin}_{u \in U} J_h(q', x, u, d)$.

Proposition 3 *Consider the controller shown in Figure 8 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* $B_{cl} = B_{sp}$.

Proof Each terminating output symbol y_i represents a safety specification for the region R_i. By the attainability assumption, there exists a control policy that guarantees safety. Therefore, there exists a solution to the optimization problem (10) and the corresponding control input satisfies the safety objective. Similarly,

for non-terminating output symbols y_k and y_{k+1}. By the attainability assumption, there exists a control policy that guarantees that the region R_{k+1} is reachable from R_k. Therefore, there exists a solution to the optimization problem (12) and the corresponding control input satisfies the reachability objective.

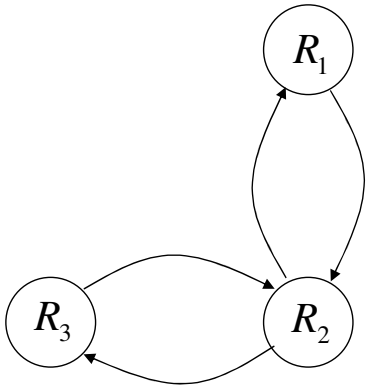

Figure 9: Control automaton for the tank system.

A major advantage of the approach is that it decouples the synthesis problem into two tasks. The first task that establishes the attainability of the specifications is computationally very expensive but it is performed off-line. The complexity of the algorithms for testing attainability of safety and reachability specifications in the worst case is exponential on the number of state space constraints and on the number of steps of the time horizon [1]. The second task that selects the control inputs signals is based on quadratic and linear programming problems that can be solved very efficiently and allow real-time implementations of the hybrid controllers.

Example In the following, we illustrate the controller design methodology using the tank system. The control specifications are described by the exosystem presented in Section 3. The attainability of the specification behavior can be verified using linear programming techniques [1]. We present a hybrid controller for the startup procedure of the tank system. During startup, it is desired for the state to transition from the initial region R_3 to the operating region R_1 through region R_2. This specification is described by the language $K_2 = abc$.

The controller consists of the event generator, the control automaton, and the actuator. The event generator determines the membership of the state (q, x) in one of the regions R_1, R_2, or R_3 described in equations (5), (5), and (6) respectively. The control automaton is shown in Figure 9. Every output of the of the control automaton corresponds to a cost functional. The control input is selected by minimizing this cost functional at every time step over all the possible control actions $u \in [0, 0.1 \times 10^{-3}]$ and $V_a \in \{0, 1\}$ while ensuring that the mode q is feasible, $q \in \text{act}(\pi(x))$, for the current state x.

In region R_3, since the only feasible mode is $q = 1$, the control input is selected as the solution to

$$\begin{array}{cc} \min & g_3^T[A_1x(t) + B_1u(t)] - w_3 \\ u \in [0, 0.1 \times 10^{-3}] & \text{s.t.} \begin{bmatrix} 0 \\ 0 \end{bmatrix} \leq A_1x(t) + B_1u(t) \leq \begin{bmatrix} 0.5 \\ 0.2 \end{bmatrix} \end{array}$$

where $g_3^T = [-1, 0]$ and $w_3 = -0.3$. In region R_2, the modes $q = 1$ and $q = 2$ are feasible. The control action includes the input flow u and position of the valve V_a and is selected as

$$\begin{array}{c} \min \\ q \in \text{act}(\pi(x)) \\ u \in [0, 0.1 \times 10^{-3}] \end{array} \quad \begin{array}{c} g_2^T[A_q x(t) + B_q u(t) + E_q] - w_2 \\ \text{s.t.} \quad \begin{bmatrix} 0.3 \\ 0 \end{bmatrix} \leq A_q x(t) + B_q u(t) + E_q \leq \begin{bmatrix} 0.5 \\ 0.2 \end{bmatrix} \end{array}$$

where $g_2^T = [0, -1]$ and $w_2 = -0.2$. Finally, in region R_1 the modes $q = 1$ and $q = 2$ are feasible when $x_2 < 0.3$ and the modes $q = 1$ and $q = 4$ are feasible when $x_2 > 0.3$. The control action is selected as

$$\begin{array}{c} \min \\ q \in \text{act}(\pi(x)) \\ u \in [0, 0.1 \times 10^{-3}] \end{array} \quad \begin{array}{c} ||A_q x(t) + B_q u(t) + E_q - \bar{x}||_2^2 \\ \text{s.t.} \quad \begin{bmatrix} 0.3 \\ 0.2 \end{bmatrix} \leq A_q x(t) + B_q u(t) + E_q \leq \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} \end{array}$$

where $\bar{x} = [0.4, 0.35]^T$. In Figure 10 we show the trajectory of the continuous state, the discrete mode, and the control input. Note that the continuous state is driven to the region R_1 and then remains inside R_1 indefinitely. The controller guarantees that every state in the region R_3 can be driven to first to the region R_2, then to the safe region R_3 and remain safe for every t.

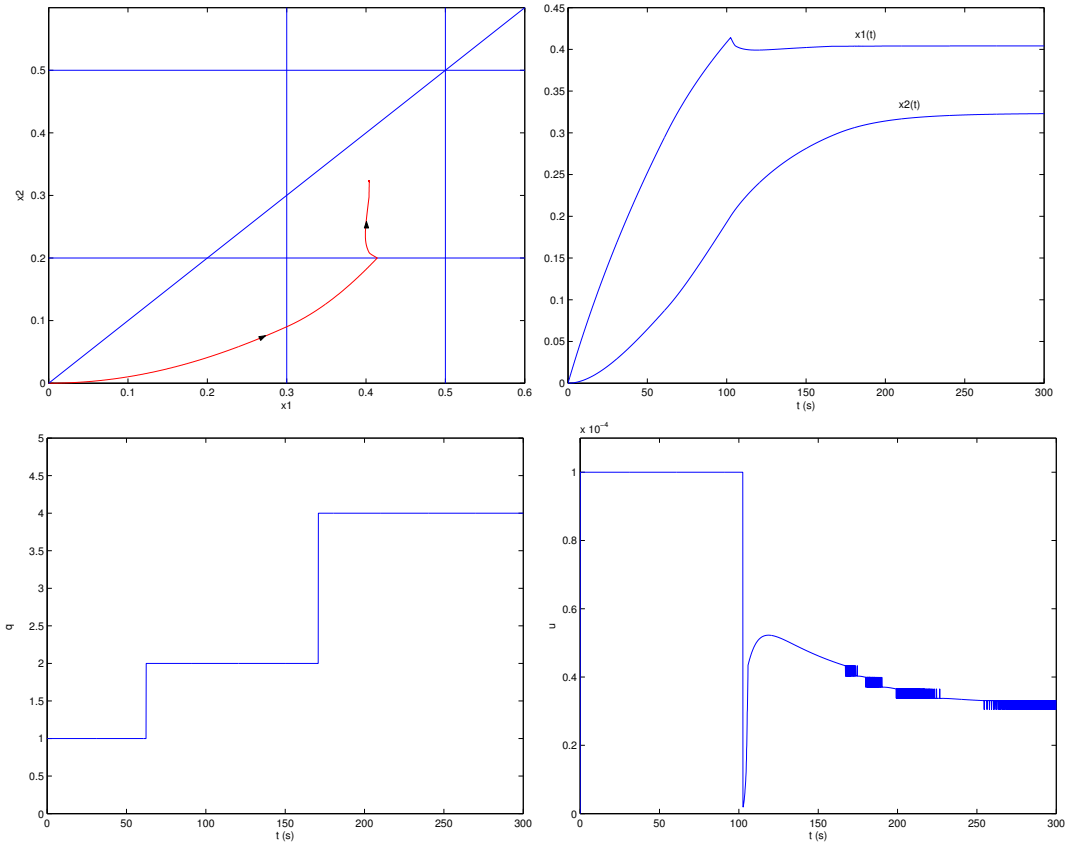

Figure 10: Tank system simulation for $x_0 = [0, 0]^T$.

6 Conclusions

In this paper, we consider hybrid systems in which the continuous dynamics are described by linear difference equations, the discrete dynamics by finite automata, and the interaction between the continuous and discrete part is defined by piecewise linear maps. The proposed modeling formalism separates the physical plant to be controlled from the control specifications and the controller. It provides the necessary mathematical tools to describe explicitly what control actions are available in order to influence the behavior of the plant so that the control specifications are satisfied. We present a novel methodology for the control design of piecewise linear hybrid dynamical systems based on a formulation of the regulator problem. We present a formal control design framework for both static specifications that do not change as time progresses and dynamic specifications that involve sequencing of events and eventual execution of actions. The main characteristic of the approach is that the feedback controller contains a control automaton that is used to select appropriate cost functionals that are minimized by selecting specific control actions.

ACKNOWLEDGEMENTS The partial support of the US National Science Foundation (NSF ECS99-12458 and CCR01-13131), of the Army Research Office (DAAG19-01-1-0743), and the DARPA/IXO-NEST Program (AF-F30602-01-2-0526) is gratefully acknowledged.

References

[1] Koutsoukos, X. and Antsaklis, P.: Safety and reachability of piecewise linear hybrid dynamical systems based on discrete abstractions. *Journal of Discrete Event Dynamic Systems: Theory and Applications*, 13(3):203–243, 2003.

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

[3] Koutsoukos, X. and Antsaklis, P.: Hybrid control of a robotic manufacturing system. In *Proceedings of the 7th IEEE Mediterranean Conference on Control and Automation*, pages 144–159, Haifa, Israel, June 1999.

[4] Koutsoukos, X. and Antsaklis, P.: Design of hybrid system regulators. In *Proceedings of the 38th IEEE Conference on Decision and Control*, pages 3990–3995, Phoenix, AZ, December 1999.

[5] Koutsoukos, X. and Antsaklis, P.: A hybrid feedback regulator approach to control an automotive suspension system. In N. Lynch and B. Krogh, editors, *Hybrid Systems—Computation and Control*, volume 1790 of *Lecture Notes in Computer Science*, pages 188–201. Springer-Verlag, 2000.

[6] Lin, H., Koutsoukos, X. and Antsaklis, P.: HYSTAR: A toolbox for hierarchical control of piecewise linear hybrid dynamical systems. In *Proceedings of the American Control Conference*, pages 686–691, Anchorage, AK, 2002.

[7] Lin, H., Koutsoukos, X. and Antsaklis, P.: Hierarchical control for a class of uncertain piecewise linear hybrid dynamical systems. In *15th Triennial World Congress of the International Federation of Automatic Control*, Barcelona, 2002.

[8] Stiver, J., Antsaklis, P. and Lemmon, M.: A logical DES approach to the design of hybrid control systems. *Mathl. Comput. Modelling*, 23(11/12):55–76, 1996.

[9] Koutsoukos, X., Antsaklis, P., Stiver, J. and Lemmon, M.: Supervisory control of hybrid systems. *Proceedings of IEEE*, 88(7):1026–1049, July 2000.

[10] Nerode, A. and Kohn, W.: Models for hybrid systems: Automata, topologies, controllability, observability. In Robert L. Grossman, Anil Nerode, Anders P. Ravn, and Hans Rischel, editors, *Hybrid Systems*, volume 736 of *Lecture Notes in Computer Science*, pages 317–356. Springer-Verlag, 1993.

[11] Raisch, J. and O'Young, S.: Discrete approximation and supervisory control of continuous systems. *IEEE Transactions on Automatic Control*, 43(4):568–573, 1998.

[12] Cury, J., Krogh, B. and Niinomi, T.: Synthesis of supervisory controllers for hybrid systems based on approximating automata. *IEEE Transactions on Automatic Control*, 43(4):564–568, 1998.

[13] Lunze, J., Nixdorf, B. and Schroder, J.: Deterministic discrete-event representations of linear continuous-variable systems. *Automatica*, 35(3):396–406, 1999.

[14] Caines, P. and Wei, Y-J.: Hierarchical hybrid control systems: A lattice formulation. *IEEE Transactions on Automatic Control*, 43(4):501–508, 1998.

[15] Sontag, E.: Nonlinear regulation: The piecewise linear approach. *IEEE Transactions on Automatic Control*, 26(2):346–358, 1981.

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

[17] Sontag, E.: Interconnected automata and linear systems: A theoretical framework in discrete-time. In R. Alur, T.A. Henzinger, and E.D. Sontag, editors, *Hybrid Systems III, Verification and Control*, volume 1066 of *Lecture Notes in Computer Science*, pages 436–448. Springer, 1996.

[18] Morse, A.: Supervisory control of families of linear set-point controllers-Part 1: Exact matching. *IEEE Transactions on Automatic Control*, 41:1413–1431, 1996.

[19] Leenaerts, D. and van Bokhoven, W.: *Piecewise Linear Modeling and Analysis*. Kluwer, 1998.

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

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

[22] Heemels, W., De Schutter, B. and Bemporad, A.: Equivalence of hybrid dynamical models. *Automatica*, 37:1085–1091, 2001.

[23] Bemporad, A., Torrisi, F. and Morari, M.: Optimization-based verification and stability characterization of piecewise affine and hybrid systems. In N. Lynch and B. Krogh, editors, *Hybrid Systems—Computation and Control*, volume 1790 of *Lecture Notes in Computer Science*, pages 45–58. Springer-Verlag, 2000.

[24] Bemporad, A., Giovanardi, L. and Torrisi, F.: Performance driven reachability analysis for optimal scheduling and control of hybrid systems. In *Proceedings of the IEEE Conference on Decision and Control*, pages 969–979, 2000.

[25] Johansson, M.: *Piecewise Linear Control Systems*. PhD thesis, Lund University, Sweden, 1999.

[26] Alur, R., Courcoubetis, C., Halbwachs, N., Henzinger, T., Ho, P-H., Nicollin, X., Oliveiro, A., Sifakis, J. and Yovine, S.: The algorithmic analysis of hybrid systems. *Theoretical and Computer Science*, 138:3–34, 1995.

[27] Lynch, N., Segala, R., Vaandrager, F. and Weinberg, H.: Hybrid I/O automata. In Rajeev Alur, Thomas A. Henzinger, and Eduardo D. Sontag, editors, *Hybrid Systems III, Verification and Control*, volume 1066 of *Lecture Notes in Computer Science*, pages 496–510. Springer, 1996.

[28] Tomlin, C., Lygeros, J. and Sastry, S.: Synthesizing controllers for nonlinear hybrid systems. In T.A. Henzinger and S. Sastry, editors, *HSCC 98: Hybrid Systems—Computation and Control*, volume 1386 of *Lecture Notes in Computer Science*, pages 360–373. Springer-Verlag, 1998.

[29] Chutinan, A. and Krogh, B.: Computing approximated aytomata for a class of linear hybrid systems. In P. Antsaklis, W. Kohn, M. Lemmon, A. Nerode, and S. Sastry, editors, *Hybrid Systems V*, volume 1567 of *Lecture Notes in Computer Science*, pages 16–37. Springer, 1999.

[30] Chutinan, A. and Krogh, B.: Verification of polyhedral-invariant hybrid automata using polygonal flow pipe approximations. In F. Vaandrager and J. van Schuppen, editors, *HSCC 99: Hybrid Systems—Computation and Control*, volume 1569 of *Lecture Notes in Computer Science*, pages 76–90. Springer-Verlag, 1999.

[31] Vidal, R., Schaffert, S., Lygeros, J. and Sastry, S.: Controlled invariance of discrete time systems. In N. Lynch and B. Krogh, editors, *Hybrid Systems—Computation and Control*, volume 1790 of *Lecture Notes in Computer Science*, pages 437–450. Springer-Verlag, 2000.

[32] Habets, L. and van Schuppen, J.: Control of piecewise-linear hybrid systems on simplices and rectangles. In M. Di Benedetto and A. Sangiovanni-Vincentelli, editors, *Hybrid Systems—Computation and Control 2001*, volume 2034 of *Lecture Notes in Computer Science*, pages 261–274. Springer-Verlag, 2001.

[33] Gokbayrak, K. and Cassandras, C.: Hybrid controllers for hierarchically decomposed systems. In N. Lynch and B. Krogh, editors, *Hybrid Systems—Computation and Control*, volume 1790 of *Lecture Notes in Computer Science*, pages 117–129. Springer-Verlag, 2000.

[34] Giua, A., Seatzu, C. and van der Mee, C.: Optimal control of switched autonomous linear systems. In *Proceedings of the 40th IEEE Conference on Decision and Control*, pages 2472–2477, Orlando, FL, December 2001.

[35] Xu, X. and Antsaklis, P.: An approach to switched systems optimal control based on parameterization of the switching instants. In *15th Triennial World Congress of the International Federation of Automatic Control*, Barcelona, 2002.

[36] Ramadge, P.: On the periodicity of symbolic observations of piecewise smooth discrete-time systems. *IEEE Transactions on Automatic Control*, 35(7):807–813, 1990.

[37] Heiming, B. and Lunze, J.: Definition of the three tank benchmark problem for controller reconfiguration. In *European Control Conference*, 1999.

[38] Bemporad, A., Mignone, D., and Morari, M.: Moving horizon estimation for hybrid systems and fault detection. In *Proceedings of the American Control Conference*, pages 2471–2475, San Diego, CA, 1999.

[39] Tsuda, K., Mignone, D., Ferrari-Trecate, G, and Morari, M.: Reconfiguration strategies for hybrid systems. In *Proceedings of the American Control Conference*, pages 868–873, Arlington, VA, 2001.

[40] Hopcroft, J. and Ullman, J.: *Introduction to Automata Theory, Languages and Computation*. Addison-Wesley, 1979.

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

[42] Motzkin, T.: *The theory of linear inequalities*. Rand Corp., Santa Monica, CA, 1952.

[43] Nerode, A. and Shore, R.: *Logic for Applications*. Texts and Monographs in Computer Science. Springer-Verlag, 1993.