Hierarchical Control of Piecewise Linear Hybrid Dynamical Systems Based on Discrete Abstractions*

Xenofon D. Koutsoukos
Xerox Palo Alto Research Center
3333 Coyote Hill Road
Palo Alto, CA 94304, USA
Tel. +1-650-812-4385
Fax +1-650-812-4334
koutsouk@parc.xerox.com

Panos J. Antsaklis
Department of Electrical Engineering
University of Notre Dame
Notre Dame, IN 46556, USA
Tel. +1-219-631-5792
Fax +1-219-631-4393
antsaklis.1@nd.edu

Abstract

In this paper, a novel methodology for analysis of piecewise linear hybrid systems based on discrete abstractions of the continuous dynamics is presented. An important characteristic of the approach is that the available control inputs are taken into consideration in order to simplify the continuous dynamics. Control specifications such as safety and reachability specifications are formulated in terms of partitions of the state space of the system. The approach provides a convenient general framework not only for analysis, but also for controller synthesis of hybrid systems. The research contributions of this paper impact the areas of analysis, verification, and synthesis of piecewise linear hybrid systems.

1 Introduction

A great amount of research work has already been done in the hybrid systems area during the past decade; see for example [3] and the references there in. A survey of different models and methodologies can be found in [4]. The approach presented in this paper is directly related to supervisory control framework for hybrid systems [38, 39, 20]. Similar approaches based on approximations of the continuous dynamics by a discrete event system have also been proposed in [31, 33, 12, 24]. The hybrid system model typically used in the supervisory control framework consists of a plant described by nonlinear differential or difference equations, a discrete event controller described by a deterministic finite automaton, and an interface which provides the means for the communication between the plant and the controller. In the model proposed in the present work, we restrict ourselves to linear dynamics evolving in discrete-time. However, we consider a plant that may also contain discrete dynamics. More importantly, we consider a larger class of control inputs which may contain both discrete and continuous inputs. In addition, we take into consideration both discrete and continuous disturbances.

In this paper, new methodologies to solve important control problems in hybrid systems are presented. Our work is motivated by the need to address challenging problems in the control and coordination of modern complex engineering

*The partial financial support of the National Science Foundation (ECS99-12458) and the Army Research Office (DAAG55-98-1-0199) is gratefully acknowledged.

applications such as autonomous vehicles, chemical and manufacturing plants, and multiple robotic systems. Hybrid systems are modeled as discrete-time dynamical systems. A mathematical model that can capture both discrete and continuous phenomena is formulated. The continuous dynamics are described by linear 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 is an important class of systems with many practical applications. The introduced model is general enough to describe important engineering applications, but simple enough to facilitate the development of analysis, and more importantly, synthesis tools. Piecewise linear hybrid dynamical systems have an efficient representation for modeling and simulation. Furthermore, current modeling tools such as MATLAB,SIMULINK, and STATEFLOW offer the necessary flexibility for modeling and simulation of this class of systems.

A systematic methodology for analysis of piecewise linear hybrid systems based on discrete abstractions of the continuous dynamics is presented. Analysis and synthesis methodologies based on discrete abstractions have been studied extensively in the hybrid system literature; see for example [2, 20] and the references there in. In order to analyze hybrid systems and design control algorithms, it is desirable to induce dynamical systems in finite quotient spaces that preserve the properties of interest and then study the simplified models. In this paper, we propose a new methodology for the construction of discrete abstractions of the continuous dynamics. An important characteristic of the approach is that the available control inputs are taken into consideration in order to simplify the system. The main mathematical tool to be used is the *predecessor operator* applied recursively to subsets of the hybrid state space. The application of the predecessor operator corresponds to partition refinement into finer partitions that allow the formulation of conditions that guarantee the existence of appropriate controls for the objectives of interest.

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. In order to study safety specifications for piecewise hybrid dynamical systems, we introduce the notion of quasideterminism. Quasideterminism represents the case when the future behavior only for the next time interval of the given system can be uniquely determined by the current state of the induced system. We show that this property can be used to formulate conditions for safety specifications for piecewise linear hybrid dynamical systems. The safety conditions can be tested using efficient linear programming techniques. We also present an algorithm for the computation of the maximal safe set based on the approach in [41, 25]. Reachability conditions are also formulated. Our approach is based on conditions that guarantee that the state can be forced to reach a desirable region of the state space by selecting appropriate controls. It should be emphasized that we are interested only in the case when reachability between two regions is defined so that the state is driven to the target region without entering a third region.This is a problem of great 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.

Piecewise linear systems arise very often as mathematical models for practical applications. For example, piecewise linear systems can be used to model systems with discontinuous dynamics that arise because of saturation constraints, hysteresis, friction in mechanical systems and so on. For another example, in order to avoid dealing directly with a set of nonlinear differential equations one may choose to work with linear equations and switch among these simpler models. Furthermore, piecewise linear systems arise in the switching control paradigm [27, 28] 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 [22] and the references therein. Here, we are interested in approaches that have been developed for modeling, analysis, and synthesis of hybrid control systems. The first investigations of piecewise linear hybrid systems can be found in [35, 36, 37]. The main problems studied in this framework are stability, controllability, and input-output regulation. Piecewise linear dynamical systems have been considered also in [10, 5, 6]. A methodology for approximating the reachable states is developed and a supervisory control framework is used for controller design. A class of hybrid systems which is similar to piecewise linear hybrid systems is considered in [7, 8, 9]. These systems are described by linear dynamic equations subject to linear inequalities involving real and integer variables. Finally, piecewise linear systems were also studied in [15] 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 [1]. Formalisms for input/output hybrid automata have been also proposed in [26, 41, 23]. A related approach to the work presented in this paper is based on the modeling formalism of hybrid automata and uses bisimulations to study the decidability of verification algorithms [14, 21, 2]. Bisimulations are quotient systems that preserve the reachability properties of the original hybrid system and therefore, problems related to the reachability of the original system can be solved by studying the quotient system. The idea of using finite bisimulations for the analysis and synthesis of hybrid systems is similar to the approximation of the continuous dynamics with discrete event systems.

The main contributions of the paper are the following. An algebraic system theoretical framework is developed for the analysis, verification, and synthesis of piecewise linear hybrid dynamical systems. This framework enables us to develop a novel methodology for analysis of piecewise linear hybrid systems based on discrete abstractions of the continuous dynamics. Our approach is based on systematic methodology for refinement of the state space partition. The main characteristic of the approach is that the available control inputs are taken into consideration in order to simplify the continuous dynamics. Algorithms for reachability analysis of discrete-time piecewise linear hybrid systems are presented in detail. It should be noted that these algorithms can be applied in the general case when the discrete dynamics contain controllable and uncontrollable events and the continuous dynamics contain control inputs and disturbances. The research contributions of this work impact the areas of reachability analysis, verification, and synthesis of piecewise linear hybrid systems. Note that the main results of this paper have appeared in [16]; early results have been reported in [18, 17, 19].

This paper is organized as follows. In Section 2, we present the modeling framework for discrete-time hybrid dynamical systems. Section 3 contains the necessary mathematical preliminaries that are used to formally define piecewise linear hybrid dynamical systems. Our mathematical model is presented in Section 4 and is illustrated using a temperature control system. In Section 5, we use an algebraic system theory framework to describe our motivation for using discrete abstractions for the analysis of hybrid systems. We also describe the proposed hierarchical control architecture using an algebraic system theory framework and we formalize the conditions under which such a hierarchical scheme can be used for control design of hybrid systems. In Section 6, we present a methodology for backward reachability analysis of piecewise linear hybrid systems. First, we formally define the notion of partition refinement by characterizing the set of polyhedral partitions as a lattice. Then, we define the predecessor operator for PLHDS, and we present computer algorithms for backward reachability analysis based on the predecessor operator. In Section 7, we study the safety problem for piecewise linear hybrid systems. We define the notion of quasideterminism and how it can be used to formulate safety conditions. We also describe an algorithm based on linear programming techniques for testing the safety conditions, and we illustrate the approach using the temperature control system. In addition, we

present an algorithm for the computation of the maximal safe set of piecewise linear hybrid dynamical systems. In Section 8, we study the reachability problem and we formulate conditions that guarantee reachability between piecewise linear regions. We also present an approximation technique for the computation of the coreachable set of a piecewise linear region based on the quantization of the state space. Finally, concluding remarks are presented in Section 9.

2 Discrete-time Hybrid Dynamical Systems

Hybrid systems are modeled by the discrete-time dynamical system

$$x(t+1) = f(q(t), x(t), u(t)) \quad (1)$$

$$q(t+1) = \delta(q(t), x(t), \sigma(t)) \quad (2)$$

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

where

- $t \in \{0, 1, 2, \ldots\} \subset \Re$ is the time index,
- $x \in X \subseteq \Re^n$ is the continuous state,
- $q \in Q$ is the discrete state or *mode* of the system, where the set Q is assumed to be finite,
- $u \in U \subset \Re^m$ is the continuous input,
- $\sigma \in \Sigma$ are the input events,
- $y \in Y$ is the output of the hybrid system,
- $f : Q \times X \times U \to X$ is the continuous state transition function,
- $\delta : Q \times X \times \Sigma \to Q$ is the discrete state transition function, and
- $g : Q \times X \to Y$ is the output function.

Note that the key characteristic of the hybrid system model is the two-sided interaction between the continuous and discrete dynamics. Often, it is desirable to distinguish between controlled and uncontrolled inputs, and we may include in the continuous state transition function both continuous controls $u \in U$ and continuous disturbances $d \in D$. 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. In our modeling framework, these events are viewed as discrete disturbances. Finally, in the case when the measurements are different from the outputs, a measurement set and a measurement function can be included in the system's description.

The dynamic evolution of the system is defined as follows. While the system is at mode (discrete state) q, the continuous state evolves according to the difference equation (1) driven by the control input $u(t)$. A change in the discrete state of the system can be caused by two type of events. First, an input event $\sigma(t)$ generated by either the controller or the environment. Second, an event $e(t)$ generated by the continuous dynamics when the continuous state enters a prescribed region of the continuous state space X.

The events generated by the continuous dynamics are defined with respect to a partition of the continuous state space. The state space $X = \Re^n$ is partitioned into a finite number of regions. When the continuous state enters a new region, an event $e(t)$ is triggered and may cause a discrete (state) transition. For every region, there exists a set of feasible discrete transitions. Conversely, each discrete transition can take place only in a specific region of the state space, which is usually called the *guard* for this transition. These notions will be explicitly defined for piecewise-linear systems later in Section 4.

The state transitions are synchronized by a clock. At every clock tick an event $\sigma(t)$ may be triggered and an event $e(t)$ caused by the continuous dynamics may occur. Therefore, every change in the state occurs synchronously to a clock. 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 fact 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.

Presently, we have focused on *piecewise-linear systems* [35, 37] to facilitate the development of analysis and synthesis tools. These systems arise when the state set and/or the input set are partitioned into regions described by linear equalities and inequalities and the dynamics at each region are described by linear (or affine) state transitions. Output and measurement maps can be defined also in a similar way. The class of piecewise-linear systems is quite general as it includes linear systems, finite state machines, and their interconnections. They can be used also in many instances as approximations of more general systems. In the following, we present some necessary background material in order to formally define the mathematical model of piecewise linear hybrid dynamical systems.

3 Preliminaries

In this section, we present some basic notions and the necessary notation that are used in the modeling formalism of piecewise linear hybrid dynamical systems.

A *piecewise-linear (PL) subset* [36] 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* $R : X \to Y$ between PL sets is one whose graph is a PL set (as a subset of $X \times Y$). Similarly for a PL map. Equivalently, the map $f : X \to Y$ is PL if there exists a covering of X by PL subsets X_i such that the restrictions $f|X_i$ are all affine (linear + translation).

Consider the state space X and define the mapping $\pi : X \to 2^X$ from X into the power set of X. The mapping π defines an equivalence relation E_π on the set X in the natural way

$$x_1 \, E_\pi \, x_2 \;\; \text{iff} \;\; \pi(x_1) = \pi(x_2).$$

The image of the mapping π is called the *quotient space* of X by E_π and is denoted by X/E_π. Adopting this notation we can write $\pi : X \to X/E_\pi$ where π is understood as the *projection* of X onto X/E_π. The mapping π generates a partition of the state set X into the equivalence classes of E_π and will be called *generator*.

More specifically, we are interested in the case when $X = \Re^n$ and the generator is defined by a set of hyperplanes. Note that such piecewise-linear regions arise in many practical applications. Consider the collection $\{h_i\}_{i=1,2,\ldots,\ell}$, $h_i : \Re^n \to \Re$ 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

$$H_i = \ker(h_i) = \{x \in \Re^n : h_i(x) = g_i^T x - w_i = 0\} \tag{4}$$

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

$$\hat{h}_i(x) = \begin{cases} -1 & \text{if} \quad h_i(x) < 0 \\ 0 & \text{if} \quad h_i(x) = 0 \\ 1 & \text{if} \quad h_i(x) > 0 \end{cases} \tag{5}$$

Then, the generator is defined by $\pi(x) = [\hat{h}_1(x), \ldots, \hat{h}_\ell(x)]^T$. Although the generator has been defined as $\pi : \Re^n \to \{-1, 0, 1\}^\ell$ there is a bijection between $\{-1, 0, 1\}^\ell$ and the quotient set X/E_π (they are the same set). The quotient set can be represented as $X/E_\pi = \{P_i\},\ i = 1, \ldots, |\pi|$ where each P_i corresponds to a polyhedral region of $\Re^n$.

It is assumed that the partition defined by the mapping π is appropriate for extraction of important information for the system and it will be called the *primary partition*. The primary partition is determined by considering the regions which are used to describe the control specifications and the interaction between the continuous and discrete part of the open loop hybrid system.

We are interested in characterizing the events that occur when the continuous state enters a new region of the state space. The set of events generated by the continuous dynamics is called the set of *plant events* and is denoted by E. Since our hybrid model evolves in discrete-time, the generator will not be able to identify the exact moment that a hypersurface is crossed. It identifies the first sample after a crossing has occurred. The sequence of time instants when plant events occur is given by the following equations:

$$\tau_e[0] = 0 \tag{6}$$

$$\tau_e[n] = \min\{t > \tau_e[n-1] : \exists i, h_i(x(t))h_i(x(\tau_e[n-1])) < 0\} \tag{7}$$

Each plant event is generated according to

$$e[n] = \ell\left(x(\tau_e[n]), x(\tau_e[n-1])\right) \tag{8}$$

$$e(t) = \begin{cases} e[n] & \text{if } t = \tau_e[n] \\ \epsilon \text{ (null)} & \text{otherwise} \end{cases} \tag{9}$$

where $\ell : X \times X \to E$ is a function labeling the plant events.

4 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.

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. A very important consequence of this characteristics is that it is possible to define open loop and closed loop connections between the plant and the controller and try to exploit the advantages of feedback.

First, we formally describe the interaction between the discrete and continuous components of a piecewise linear hybrid system. For each discrete mode, we assign a region of the state space using the mapping

$$\text{inv} : Q \to 2^{X/E_\pi}. \tag{10}$$

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 $\text{inv}(q)$ are called *invariants*. It is assumed that the invariants are regions of the primary partition. These regions arise from the control specifications that do not allow certain modes in a region of the state space. They can also arise from discontinuities in the continuous dynamics when, for example, saturation or sign functions are used to model the physical processes.

Note that in our modeling framework, the invariants do not necessarily correspond to disjoint regions of the state space. This is a realistic assumption, since many times in modeling of practical applications, it is not straightforward to assign a unique difference equation to each region of the state space. This is a task to be accomplished by the controller depending on the control specifications.

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

$$\text{act} : X/E_\pi \to 2^Q. \tag{11}$$

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

$$P \in \text{inv}(q) \Leftrightarrow q \in \text{act}(P). \tag{12}$$

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

$$\begin{aligned} x(t+1) &= A_{q(t)}x(t) + B_{q(t)}u(t) + E_{q(t)}d(t) && (13)\\ q(t+1) &= \delta(q(t), \pi(x(t)), \sigma_c(t), \sigma_u(t)),\ q(t+1) \in \text{act}(\pi(x(t))) && (14)\\ y(t) &= g(q(t), x(t)) && (15) \end{aligned}$$

where $x(0) = x_0 \in \Re^n, q(0) = q_0 \in Q$ and

- $\pi : X \to X/E_\pi$ partitions the continuous state space $\Re^n$ into polyhedral equivalence classes,
- $\text{act} : X/E_\pi \to 2^Q$ defines the active mode set,
- $A_q \in \Re^{n\times n}, B_q \in \Re^{n\times m}$, and $E_q \in \Re^{n\times p}$ are the system matrices for the discrete state q,
- $\delta : Q \times E \times \Sigma_c \times \Sigma_u \to Q$ is the discrete state transition function, and
- $g : Q \times X \to Y$ is the output function which is assumed to be piecewise linear.

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 $q \to q'$ (or (q, q')) may occur. Each feasible discrete state transition is 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 may 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 controllable or uncontrollable event the next discrete state can be uniquely determined.

The *guard* $G(q, q')$ of the transition (q, q') is defined as the set of all states (q, x) such that $q' \in \text{act}(\pi(x(t)))$ 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 $\sigma_u \in \Sigma_u$. The

guard of the transition describes the region of the hybrid state space where the transition can be forced to take place independently of the disturbances generated by the environment.

It should be noted that the above model does not include jumps in the continuous state that may occur when certain state variables are discontinuously reset, for example, upon crossing a hyperplane. Jumps can be added in the modeling formalism described above and in the subsequent analysis if they can be represented by piecewise linear maps. However, the notation becomes tedious, and the ideas and methodologies presented harder to follow.

Example - Temperature Control System We present a temperature control system to illustrate the piecewise linear hybrid system model. An electrical analog of a temperature control system is used by considering the temperature being analogous to electric voltage, heat quantity to current, heat capacity to capacitance, and thermal resistance to electrical resistance. The system consists of a furnace that can be switched on and off. When the furnace is on, a continuous input controls the produced heat. The control objective is to control the temperature at a point of the system by applying the heat input at a different point.

When the furnace is on, the system is described by the electrical circuit shown in Figure 1. Let x_1 and x_2 denote the voltages across the capacitors C_1 and C_2 respectively. Suppose that the (voltages) temperatures x_1 and x_2 are to be controlled by changing the (current) heat input u, which takes values in the set $U \subset \Re$. The temperature x_2 is also affected by the temperature d of the environment which is modeled as a continuous disturbance.

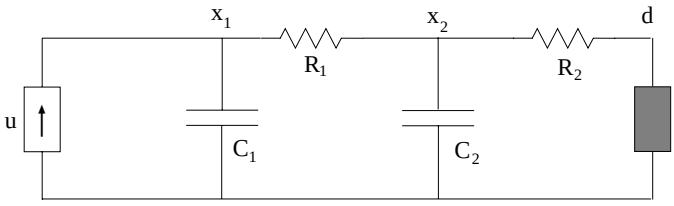


Figure 1: Electric circuit describing the case when the furnace is on.

When the furnace is turned off, the temperature is decreasing and the behavior of the system is described by the electrical circuit shown in Figure 2. The values of the resistors and the capacitors model the time constants of the system. The time constants are, in general, different depending on whether the temperature is increasing or decreasing.

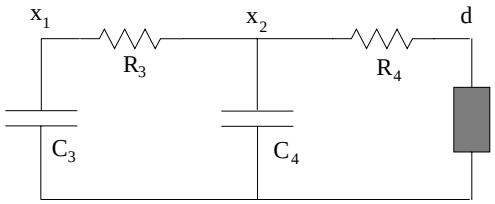


Figure 2: Electric circuit describing the case when the furnace is off.

The voltages (temperatures) x_1 and x_2 can be affected by either the continuous control input $u \in U$ or by switching on (mode q_1) or off (mode q_0) the furnace as illustrated in Figure 3. For example, we can consider a safety specification where the goal is to maintain the temperature x_2 between appropriate levels described by the tolerance interval $[lt, ht]$.

A safety guard may be included in the system representation, so that the furnace is switched off automatically whenever the temperature x_1 exceeds a prescribed level M.

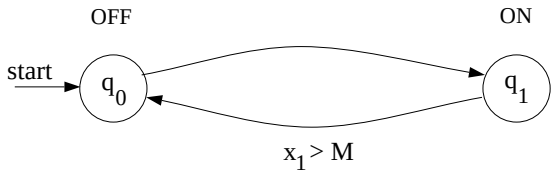


Figure 3: Temperature control system.

In the case when the system is on, the system is described by the state-space equation (using Kirchhoff's laws)

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -\frac{1}{R_1 C_1} & \frac{1}{R_1 C_1} \\ \frac{1}{R_1 C_2} & -\frac{1}{R_{12} C_2} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} \frac{1}{C_1} \\ 0 \end{bmatrix} u + \begin{bmatrix} 0 \\ \frac{1}{R_2 C_2} \end{bmatrix} d \tag{16}$$

where $R_{12} = \frac{R_1 R_2}{R_1 + R_2}$. When the system is off, then the state-space representation of the system takes the form

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -\frac{1}{R_3 C_3} & \frac{1}{R_3 C_3} \\ \frac{1}{R_3 C_4} & -\frac{1}{R_{34} C_4} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{R_4 C_4} \end{bmatrix} d. \tag{17}$$

A partition of the continuous state space is obtained by considering the hyperplanes $h_1 = x_1 - M,\ h_2 = x_2 - lt,\ h_3 = x_2 - ht$, and $h_4 = x_1$ that describe the safety guard and the control specifications of the system. The partition of the continuous state space is shown in Figure 4. Discrete-time representations of the continuous dynamics for each mode are obtained using (zero-order hold) sampling. A piecewise linear hybrid dynamical system which models the temperature control system is described by the following equations:

$$x(t+1) = A_{q(t)} x(t) + B_{q(t)} u(t) + E_{q(t)} d(t),\ x_0 = x(0) \tag{18}$$

$$q(t+1) = \delta(q(t), \pi(x(t)), \sigma(t)),\ q_0 = q(0) \tag{19}$$

$$y(t) = x_2(t) \tag{20}$$

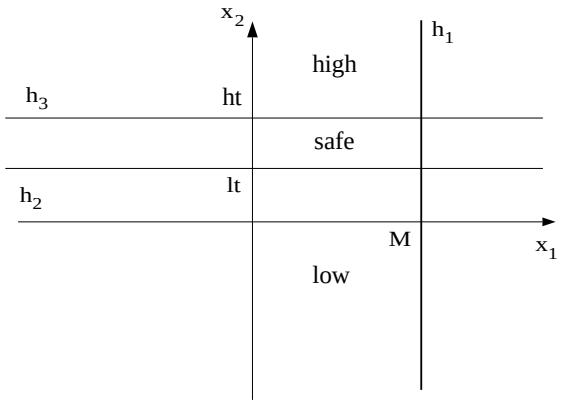


Figure 4: Partition for the temperature control system.

The discrete state transition function is described by the automaton of Figure 3. The transitions can be forced by controllable events issued by a control mechanism. The transition (q_1, q_0) may also occur because of the safety guard. The safety guard is described using the mapping act : $X/E_\pi \to 2^Q$ that defines the enabled transitions for each region of the primary partition. For the temperature control system, we have that the system cannot be at discrete state q_1 (on) if $x_1 > M$.

The temperature control system example is used to illustrate the partition refinement methodolody for safety specifications in Subection 7.2 and for reachability specifications in Subsection 8.1. □

5 Discrete Abstractions

This section describes an algebraic system theoretical framework that enable us to formalize the partition refinement methodology. The main contribution is a framework for constructing discrete abstractions for piecewise linear hybrid systems that take into consideration the control inputs, both continuous and discrete.

In order to analyze hybrid systems and design control algorithms, it is desirable to induce dynamical systems in finite quotient spaces that preserve the properties of interest and then study the simplified models. In general, piecewise linear hybrid dynamical systems cannot be induced in finite quotient spaces by preserving the reachability properties [21]. The solution we propose is to take advantage of the available control inputs in order to simplify the system. More specifically, we want to formulate conditions on the available control inputs in order to construct meaningful discrete abstractions of the hybrid system. The main mathematical tool to be used is the *predecessor operator* applied recursively to subsets of the hybrid state space. The application of the predecessor operator corresponds to the refinement of the primary partition into finer partitions that allow the formulation of conditions that guarantee the existence of appropriate controls for the objectives of interest.

In general, the design of the partition depends not only on the plant to be controlled, but also on the control policies available, as well as on the control goals to be attained. Certain control goals may require, for example, detailed feedback information while for others coarser quantization levels of the signals may be sufficient. The former case corresponds to finer partitioning of the feedback signal space, while the latter corresponds to coarser partitioning. The fact that different control goals may require different types of information about the plant is not surprising, as it is rather well known that to stabilize a system, for example, requires less detailed information about the system's dynamic behavior than to do tracking. Note that in general, the fewer the distinct regions in the partitioned signal space, the simpler (fewer states) the resulting induced system will be, and this will result in a simpler controller design. Since the systems to be controlled via hybrid controllers are typically complex, it is important to make every effort to use only the necessary information to attain the control goals. The question of systematically determining the minimum amount of information needed from the plant in order to achieve particular control goals is an important and largely open question; our work only partially resolves this question.

5.1 Induced Dynamical Systems

Let f be the state transition function of a dynamical system and assume that the inputs are fixed. Consider the diagram in Figure 5. Intuitively, the map π is used to coarsen the state set of the system. The question that arises is whether the system f can follow this abstraction. This question is concerned with the existence of a mapping $\tilde{f} : X/E_\pi \to X/E_\pi$ that makes the diagram commute. It is shown in [34] that $\tilde{f}$ exists if and only if

$$x_1 \, E_\pi \, x_2 \quad \Rightarrow \quad (\pi \circ f)(x_1) = (\pi \circ f)(x_2) \tag{21}$$

(where $\circ$ denotes function composition) and moreover, if (21) is satisfied then $\tilde{f}$ is unique. Note that the above result does not require any structure on the set X or the mappings π and f. Using equivalence relations on the state set X, it is possible to define new dynamical systems in the derived quotient spaces. These systems are called *induced dynamical systems* [34].

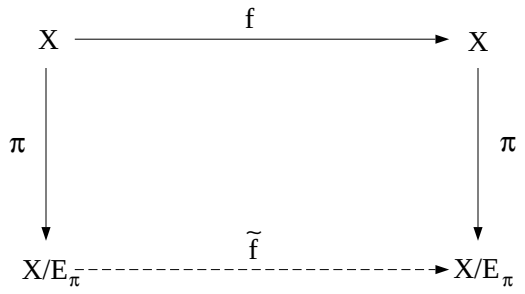


Figure 5: Induced dynamical systems.

In the hybrid systems case, the properties of the original system are not preserved, in general, in the induced system. One of the main difficulties arises because abstractions of continuous systems in finite quotient spaces usually result in nondeterniministic discrete event systems. Consider, for example, two continuous states $x_1, x_2 \in \Re^n,\ x_1 \neq x_2$ such that $\pi(x_1) = \pi(x_2) = P \in X/E_\pi$. The states x_1 and x_2 may be driven even using the same control input to different equivalence classes of the quotient space X/E_π, see Figure 6. Therefore, in general we have that $(\pi \circ f)(x_1) \neq (\pi \circ f)(x_2)$ and a mapping $\tilde{f}$ that makes the diagram commute does not exist. The induced system defined by the mapping $\tilde{f} : X/E_\pi \to X/E_\pi$ can be viewed as a nondeterministic system.

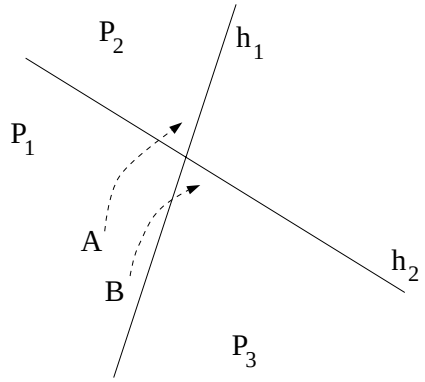


Figure 6: Nondeterminism of the induced dynamical system.

In general, piecewise linear hybrid dynamical systems cannot be induced in finite quotient spaces by preserving the reachability properties [21]. However, there are some cases when a mapping π and the induced system $\tilde{f}$ can be computed. A special case arise when the mapping π is defined using the natural invariants of the continuous dynamics [40]. However, it is very difficult to compute such partitions, and more importantly, the control specifications are not necessarily defined using the invariant sets of the system.

5.2 Hierarchical Control

The solution we propose is to take advantage of the available control inputs in order to simplify the system. More specifically, we want to formulate conditions on the available control inputs in order to induce piecewise linear hybrid dynamical systems in finite quotient spaces. In order to illustrate our approach, we use the hierarchical architecture shown in Figure 7. The design of hybrid control systems is decomposed in two levels. In the higher level, we are concerned only with the existence of appropriate control inputs. The implementation of the controller and therefore,

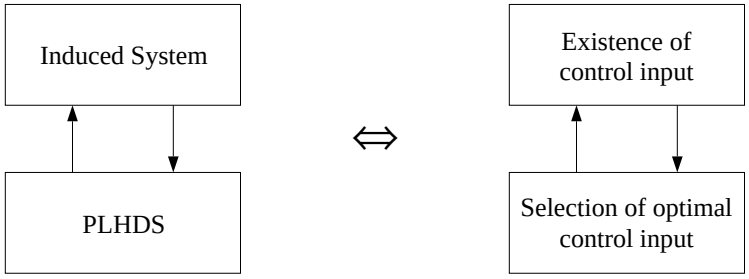


Figure 7: Hierarchical control of hybrid systems.

the selection of the control input signal is done by the lower level. First, we want to formulate efficient algorithms that guarantee the existence of appropriate control inputs for safety and reachability specifications. Second, we want to develop systematic methodologies for the design of the (lower level) controller.

In this paper, we concentrate on the first problem and we formulate conditions for the existence of appropriate control inputs for safety and reachability specifications. The conditions are expressed as the feasibility of an optimization problem. The lower level problem is concerned with the selection of the optimal control inputs and it is a by-product of the optimization algorithm. A systematic design methodology for the selection of optimal control inputs that results in a feedback control architecture has been developed in [16], but it is not presented in this paper due to space limitations.

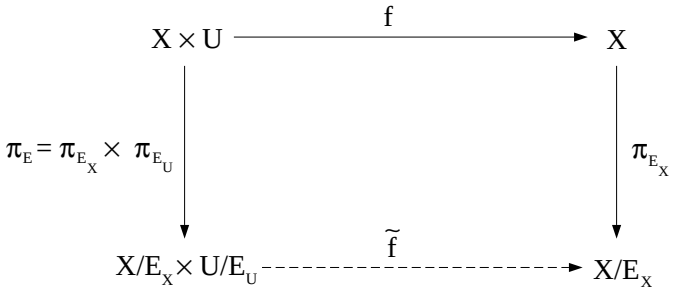


Figure 8: Function diagram including control inputs.

First, we describe our approach using an algebraic system theory setting. Consider the diagram shown in Figure 8. The equivalence relation E is defined by the mapping $\pi_E : X \times U \to X/E_X \times X/E_U$ as follows. The restriction of π_E in the state space X is the mapping which describes the primary partition of the system. The restriction of π_{E_U} separates the input space U into two equivalence classes. The first equivalence class consists of all control inputs available to the system and the second class consists of all the remaining elements of the input space. In practical applications, physical constraints such as saturation constraints restrict the control inputs that can be applied to the system. For example, the current input in the temperature control system example is constrained based on the available current source. Many times, we even consider a finite set of inputs corresponding to specific commands as, for example, in a valve can be closed, half open, open, and so on. Therefore, (x_1, u_1) is equivalent to (x_2, u_2) if and only if $\pi(x_1) = \pi(x_2)$ and the control inputs u_1, u_2 can be applied to the system. Note that the equivalence relation of the input space is defined in accordance with the hierarchical control architecture of Figure 7, since the higher control level is concerned only with the existence of controls. All available control inputs are equivalent at this level of abstraction.

The induced dynamical system $\tilde{f}$ exists if and only if

$$(x_1, u_1)\, E\, (x_2, u_2) \quad \Rightarrow \quad \exists u_1, u_2 \in U, (\pi_{E_X} \circ f)(x_1, u_1) = (\pi_{E_X} \circ f)(x_2, u_2). \tag{22}$$

The interpretation of the above condition is that $\tilde{f}$ exists if and only if there exist control inputs so that states that belong to the same polyhedral equivalence class of the primary partition, will remain equivalent in the next time step.

Of course, it is desirable to consider the dynamic evolution of the system in more than one steps. In order to do that, we consider an upper bound $N \in \mathbf{N}$ on the number of time steps that defines the length of the time horizon of interest. The length is assumed to be finite, since infinite-time problems in piecewise linear systems are, in general, undecidable [37].

We introduce the following notation.

$$\begin{aligned} [t_1, t_2] &= \{t_1, t_1+1, \ldots, t_2-1, t_2\},\ t_1 \leq t_2, \\ u^*[t_0, t_1] &= \{u(t_0), \ldots, u(t_1)\}. \end{aligned}$$

The function diagram in this case is shown in Figure 9.

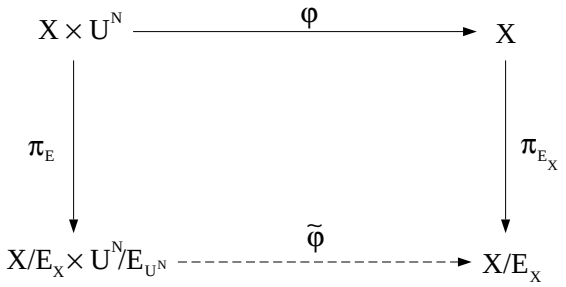


Figure 9: Function diagram including control input sequences.

The equivalence relation on the input space is now defined as follows. The input sequences $u_1^*[t_0, t_1]$ and $u_2^*[t_0, t_2]$ are equivalent if and only if $u_1(t), t \in [t_0, t_1]$ and $u_2(t), t \in [t_0, t_2]$ are available control inputs, and in addition we have $t_1 - t_0 \leq N$ and $t_2 - t_0 \leq N$. The system mapping denoted by $\phi : X \times U^N \to X$ is the extension of the map $f : X \times U \to X$, so that it can be applied to sequence segments $u^*[t_0, t],\ t - t_0 \leq N$.

The induced dynamical system $\tilde{\phi}$ exists if and only if

$$(x_1, u_1^*[t_0, t_1])\, E\, (x_2, u_2^*[t_0, t_2]) \quad \Rightarrow \quad (\pi_{E_X} \circ \phi)(x_1) = (\pi_{E_X} \circ \phi)(x_2). \tag{23}$$

Our objective is to compute a partition of the state space so that the diagram shown in Figure 9 commutes. Our approach is to refine the initial partition that is used to describe the specifications, until we can guarantee that there exist appropriate control resources that guarantee that the specifications are satisfied. Note that we consider only the regions of the state space that appear in the specifications. Consequently, the commutativity of the diagram in Figure 9 is only required with respect to the equivalence classes that are formed from the control specifications.

5.3 Partition Refinement

In this section, we characterize the set of all the partitions of the state space with polyhedral equivalence classes as a lattice and we define the notion of partition refinement with respect to the partial order of the lattice. The characterization of the partition refinement in a lattice framework is very important for the following reasons. First, by formally defining the partition refinement as a lattice operation it is clarified how the regions of are combined to form the final partition of

the system. Second, it illustrates the difficulty of using a partition of the state space to abstract the continuous dynamics (see Proposition 1).

In the following, we present some basic notions from algebraic system theory [34] that are needed for to formalize the partition refinement methodology. A *binary relation* on X is defined as a subset $B \subset X \times X = X^2$. A *poset* is defined as a set X with a partial order relation $\leq$ on X and is denoted by $(X, \leq)$. A *lattice* $(X, \leq, \wedge, \vee)$ is a poset $(X, \leq)$ for which any two elements have a greatest lower bound (*infimum*) denoted by the binary operation $x \wedge y$ (*meet*), and a least upper bound (*supremum*) denoted by the binary operation $x \vee y$ (*join*). A lattice is said to be *complete* if $\inf(Y)$ and $\sup(Y)$ exist for every $Y \subset X$. Let Y be a subset of the lattice $(X, \leq, \wedge, \vee)$, then $(Y, \leq, \wedge, \vee)$ is said to be a *sublattice* of $(X, \leq, \wedge, \vee)$ if Y is closed with respect to the binary operations meet and join.

Denote by $B(X)$ the set of all binary relations on the set X. We can define the poset $(B(X), \leq)$ where the partial order relation $\leq$ on $B(X)$ defined as $B_1 \leq B_2$ if $(x_1, x_2) \in B_1 \Rightarrow (x_1, x_2) \in B_2$. A lattice structure $(B(X), \leq, \wedge, \vee)$ can be developed in the poset $(B(X), \leq)$ by introducing meet and join operations (corresponding to the set theoretic intersection and union in X^2). The lattice $(B(X), \leq, \wedge, \vee)$ is complete and is referred to as the *relational lattice.* Let $E(X)$ be the set of all equivalence relations on X. We have that $E(X) \subset B(X)$ and $E(X)$ inherits the partial order of $B(X)$, that is for $E_1, E_2 \in E(X)$ $E_1 \leq E_2$ if $x_1 E_1 x_2 \Rightarrow x_1 E_2 x_2$. A lattice structure can also be developed on the set of all equivalence relations on X (for more details see [34]). The lattice $(E(X), \leq, \wedge, \vee)$ is called the *equivalence lattice*.

Now we prove the following proposition which illustrates how the partition refinement can be used for constructing discrete abstractions for piecewise linear hybrid systems.

Proposition 1 *The set $E_P(X)$ of all equivalence relations on X induced by mappings $\pi : X \to X/E_\pi$ which are defined using finite collections of $(n-1)$-dimensional hyperplanes and thus, they separate the state space X into polyhedral equivalence classes, is a sublattice of the equivalence lattice $E(X)$, and will be called polyhedral equivalence lattice. Furthermore, $E_P(X)$ is not complete.*

Proof Consider the equivalence relations $E_1, E_2 \subset X$ defined by the finite collections of affine functions $\mathcal{H}_1 = \{h_i\}_{i=1,\ldots,d_1}$ and $\mathcal{H}_2 = \{h'_i\}_{i=1,\ldots,d_2}$ respectively. The meet of E_1 and E_2 is defined as the set theoretic intersection $E = \inf(E_1, E_2) = E_1 \cap E_2$. E is clearly the equivalence relation defined by $\mathcal{H} = \mathcal{H}_1 \cup \mathcal{H}_2$ and its equivalence classes are polyhedral sets since they are defined by the intersection of the equivalence classes of E_1 and E_2. Therefore, $E \in E_P(X)$. The join E' of E_1 and E_2 is defined as the intersection of all equivalence relations $E'_i \in E_P(X)$ that are larger than E_1 and E_2 with respect to the partial order of the equivalence lattice

$$E' = \sup(E_1, E_2) = E_1 \cup E_2 = \bigcap_i E'_i, \; E_1, E_2 \leq E'_i \tag{24}$$

The intersection of infinite number of equivalence relations from $E_P(X)$ does not necessarily belong to $E_P(X)$. However, in this case we can define E' to be the equivalence relation induced by the finite collection $\mathcal{H}' = \mathcal{H}_1 \cap \mathcal{H}_2$ of affine functions. Then clearly, $E_1, E_2 \leq E'$ and $E' \in E_P(X)$. Note that in the case E_1 and E_2 do not have any common hyperplanes, their join is the equivalence relation that corresponds to X^2.

For the sublattice $(E_P(X), \leq, \wedge, \vee)$ to be complete, every subset of $E_P(X)$ should have an infimum and a supremum. Consider a infinite set $\{E_i\}$ of equivalence relations in $E_P(X)$, then $\inf_i(E_i)$ does not necessarily belong to $E_P(X)$ since infinite intersections of polyhedral sets may not be polyhedral. □

Partition refinement is defined with respect to the order relation of the polyhedral equivalence lattice. A partition defined by the mapping π' is finer than the partition defined by π, if the induced equivalence relations considered as

elements of the equivalence lattice satisfy the condition $E_{\pi'} \leq E_{\pi}$. The partition refinement methodology starts from the initial partition of the system and computes finer partitions by incorporating additional hyperplanes. In the lattice framework, given the primary partition, we refine the state space using the "meet" operation of $E_P(X)$. The fact that the polyhedral equivalence lattice is not complete implies that in order for the final partition to be a polyhedral equivalence relation, the partition refinement must use only a finite number of "meet" operations. It should be emphasized that the control specifications,the invariants, and the guards of the hybrid model are represented using the polyhedral regions of the primary partition.

6 Backward Reachability Analysis

In this section, we describe a backward reachability analysis approach for partition refinement. The main contribution is an efficient algorithm for partition refinement of piecewise linear hybrid systems based on the predecessor operator.

6.1 The Predecessor Operator for PLHDS

In this section, we define the predecessor operator for PLHDS. We also present the technical results that are necessary for the computation of the operator. These results are used for the development of computer algorithms for backward reachability analysis of PLHDS.

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. In the case of piecewise linear hybrid dynamical systems, it suffices to assume that the region is represented by $R = (q, P)$ where $q \in Q$ and $P \subset \Re^n$ is a piecewise linear set. The dynamic evolution of the system is defined by discrete and continuous transitions. We first define and compute the predecessor operator for discrete transitions.

6.1.1 Discrete Transitions

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

$$\text{pre}_d(R) = \{(q', x) \in Q \times X | \exists \sigma_c \in \Sigma_c, \forall \sigma_u \in \Sigma_u, q = \delta(q', x, \sigma_c, \sigma_u)\}. \tag{25}$$

It should be noted that the null event is included in the event set $\Sigma = \Sigma_c \cup \Sigma_u$. A controllable and an uncontrollable event may occur at the same time instant. If only a controllable event occurs at a time instant, then the uncontrollable event at this time instant is assumed to be the null event ϵ and vice versa. Furthermore, by the definition of the PLHDS, for the transition $q' \rightarrow q$ to be feasible, it is required that $q' \in \text{act}(\pi(x))$.

The refinement partition algorithm consists of recursive applications of the predecessor operator starting with the regions defined by the primary partition. For every discrete transition that can be forced by a controllable event we have that

$$\text{pre}_d(R) = \bigcup_{q' \in \text{act}(P)} G(q', q) \tag{26}$$

where $R = (q, P)$ and $G(q', q)$ is the guard of transition $G(q', q)$.

Since we assumed that the guards are described by the polyhedral equivalence classes of the primary partition, no refinement is necessary for the discrete transitions.

6.1.2 Continuous Transitions

In the case of continuous transitions, given the region $R = (q, P)$ we define the predecessor operator $\text{pre}_c : 2^{Q\times X} \to 2^{Q\times X}$ 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

$$\text{pre}_c(R) = \{q\} \times \{x \in X | \exists u \in U, \forall d \in D, A_q x + B_q u + E_q d \in P\}. \tag{27}$$

The set $\text{pre}_c(R)$ 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 [36]. In order to illustrate this result, we consider an alternative way to define PL sets [36].

Definition 2 Let $\mathcal{L}$ be the first-order language defined by (i) a set of (countably many) variables $\{x_1, x_2, \ldots\}$, (ii) the connective symbols $\neg$ and $\to$, (iii) the quantifier $\forall$, the parentheses (and) and the comma, (iv) A set of constants $\{r\}$ for each real number r, (v) A set of unary functions $\{r \cdot ()\}$ for each real number, the binary function $+$, (vi) the relational symbols $>$ and $=$.

Lemma 1 *[36] Every sentence in $\mathcal{L}$ defines a PL set and conversely, every PL subset of $\Re^n$ can be defined in this fashion.*

The conclusion of the above lemma is that any set defined using quantifiers can be also defined using only propositional connectives. Therefore, we can represent the predecessor operator of any piecewise linear region of the hybrid state space without quantifiers.

6.2 Computation of the Predecessor Operator

In the following, we present algorithms to carry out the elimination of quantifiers for the computation of the predecessor operator for piecewise-linear hybrid dynamical systems. As it was explained in Section 6.1, the predecessor operator for discrete transitions is given by the union of the guards of those transitions that are feasible and can be forced by a control mechanism. Since the guards are regions of the state space that are included in the description of the primary partition, here we concentrate on predecessor operator for the continuous transitions. Our results are based on combinations of three different mathematical tools. Fourier-Motzkin elimination [29] for computing appropriate projections, linear programming techniques [30] for eliminating redundant constraints, and equivalences from predicate logic [32] to combine the constraints.

Consider the region $R = (q, P)$ where $q \in Q$ and P is a PL set. A PL set is not necessarily polyhedral. However, every PL set P can be written as

$$P = \bigcup_{i=1}^{p} P_i \tag{28}$$

where P_i are polyhedral sets, as shown in the following lemma. The proof of the lemma is constructive and is used in the developed algorithms for backward reachability analysis.

Lemma 2 *Every PL set can be written as a finite union of polyhedral sets.*

Proof By Lemma 4.1, every PL set can be written as a sentence in $\mathcal{L}$. Such a sentence is a logical formula without quantifiers and can be written in *disjunctive normal form* (see for example [32]) as

$$(\phi_{11} \wedge \phi_{12} \wedge \ldots) \vee \ldots \vee (\phi_{p1} \wedge \phi_{p2} \wedge \ldots) \tag{29}$$

where ϕ_{ij} is logical formula describing either a linear equality or inequality.

Consider a linear constraint ϕ_{ij} and assume without loss of generality that can be represented by the linear inequality $g_{ij}^T x < w_{ij}$. Then, equivalently we can represent ϕ_{ij} by the inequality $-g_{ij}^T x > -w_{ij}$. Therefore, every constraint in Equation (29) can be represented using, for example, the relational symbols $>$ and $=$. But every conjunction of linear constraints $(\phi_{i1} \wedge \phi_{i2} \wedge \ldots)$ is a polyhedral set as the intersection of halfspaces and hyperplanes. Therefore, every PL set can be written as a finite union of polyhedral sets by representing the set as a logical formula in disjunctive normal form. □

In order to simplify the notation, we consider only the restriction of the predecessor operator in the continuous state space $\text{pre}_c : 2^X \to 2^X$. It should be noted that in order to show that there exists a constructive algorithm for elimination of quantifiers, we have essentially to consider only the logical formula

$$(\exists u \in U)(\phi_{11}(x,u) \wedge \phi_{12}(x,u) \wedge \ldots) \vee \ldots \vee (\phi_{p1}(x,u) \wedge \phi_{p2}(x,u) \wedge \ldots). \tag{30}$$

Algorithms for elimination of quantifiers for more complicated logical formulas can then be derived using logical equivalences [11]. In the case of PLHDS, we are interested in elimination of quantifiers for formulas of the form $(\exists u \in U)$ and $(\forall d \in D)$ for the control inputs and disturbances respectively. By Lemma 4.2, it suffices to show how the predecessor operator is applied to a union of polyhedral sets. We compute the predecessor operator of a PL set in two steps. First, we consider only polyhedral sets and second, unions of polyhedral sets.

Consider the system

$$x(t+1) = Ax(t) + Bu(t) \tag{31}$$

where $A \in \Re^{n \times n}$ and $B \in \Re^{n \times m}$. It is assumed that the control input takes values in the polytope (bounded polyhedral) U described by

$$U = \{u \in \Re^m | Fu \leq v\}, \;\; F \in \Re^{\mu \times m}, \; v \in \Re^{\mu}. \tag{32}$$

Consider the polyhedral set $P \subseteq \Re^n$ given by

$$P = \{x \in \Re^n | Gx \leq w\}, \;\; G \in \Re^{\nu \times n}, \; w \in \Re^{\nu}. \tag{33}$$

Our objective is to present a systematic methodology to compute the predecessor operator set

$$\text{pre}_c(P) = \{x \in \Re^n | \exists u \in U, Ax + Bu \in P\}. \tag{34}$$

We denote $\text{Pr} : X \times U \to X$ the *projection* from the set $X \times U = \Re^n \times \Re^m$ to the state space $X = \Re^n$.

Proposition 2 *The set $pre_c(P)$ is given by*

$$pre_c(P) = \text{Pr}(Q) \tag{35}$$

where $Q \subseteq X \times U$ is defined as

$$Q = \{(x,u) | (GAx + GBu \leq w) \wedge (Fu \leq v)\}. \tag{36}$$

Proof By direct substitution, we have that

$$\text{pre}_c(P) = \{x | \exists u \in U, GAx + GBu \leq w\} \tag{37}$$

Then, we have that if $x \in \Pr(Q)$, there exits $u \in U$ such that $(x, u) \in Q$, and therefore $x \in \text{pre}_c(P)$. Conversely, if $x \in \text{pre}_c(P)$, then by definition of the predecessor operator there exists control input $u \in U$ such that $(x, u) \in Q$, which implies that $x \in \Pr(Q)$. Therefore, we have shown that $\text{pre}_c(P) = \Pr(Q)$. □

The projection of the set Q into the continuous state space $X = \Re^n$ can be computed using the *Fourier-Motzkin elimination method* [29, 13, 43]. We project the polyhedron $Q \subset \Re^n \times \Re^m$ into the space $\Re^n$ by eliminating the variables u_i of the control input vector. According to Fourier's method, in order to eliminate a variable from a set of inequalities, we must consider all pairs of inequalities in which the variable has opposite sign and eliminate the variable between each pair.

Since U is bounded, all the control variables u_i will appear with opposite sign in at least one pair of inequalities from the constraints $Fu \leq v$. In order to see that, consider that there exists u_i that appear with the same sign in all the constraints. Assume without loss of generality that u_i appears with a positive sign in all the constraints $Fu \leq v$. Then, u_i can be decreased indefinitely without violating any of the constraints. Therefore, the set U is unbounded which is a contradiction.

Example Consider the following set of linear inequalities

$$x_1 + x_2 + u \leq 1 \tag{38}$$
$$2x_1 + x_2 + u \leq 1 \tag{39}$$
$$u \leq 1 \tag{40}$$
$$-u \leq -.5 \tag{41}$$

for which we want to eliminate the variable u. We consider all pairs of inequalities in which the variable u has opposite signs and eliminate between each pair. To demonstrate this, the inequalities (38) and (41) can be written as

$$.5 \leq u \leq 1 - x_1 - x_2. \tag{42}$$

Therefore, we have that

$$.5 \leq 1 - x_1 - x_2 \tag{43}$$

which can be written as

$$x_1 + x_2 \leq .5. \tag{44}$$

Therefore, if there is a solution to the inequalities (38) and (41), there must be a solution to the derived inequality (44). Conversely, if there is a solution to (44), then by writing the inequality in the form (42), it follows that there exists u such that the initial inequalities are satisfied. Note that the inequality (44) can be easily derived by adding (38) and (41) (after possible multiplication by a positive number).

Repeating this procedure for all the pairs of inequalities in which u has different signs we obtain the following set of linear inequalities, which represents the projection of the set of solutions to the (x_1, x_2) space.

$$x_1 + x_2 \leq .5 \tag{45}$$
$$2x_1 + x_2 \leq .5 \tag{46}$$
$$0 \leq .5 \tag{47}$$

Note that the constraint $0 \leq .5$ is redundant. □

A piecewise linear set, however, is not necessarily polyhedral, but it can be written as the union of polyhedral sets using the proof of Lemma 4.2. Consider, the set $P = \bigcup_{i=1}^{p} P_i$ where P_i are polyhedral sets. Then, the set $\text{pre}_c(P)$ can be computed by the following lemma.

Lemma 3 *Consider the piecewise linear set $P = \bigcup_{i=1}^{p} P_i$, where P_i are polyhedral sets, then the predecessor operator of P can be computed by $\text{pre}_c(P) = \bigcup_{i=1}^{p} \text{pre}_c(P_i)$.*

Proof

$$\begin{aligned}
\text{pre}_c(P') &= \text{pre}_c\left(\bigcup_{i=1}^{p} P_i\right) && (48)\\
&= \{x | \exists u \in U, Ax + Bu \in \bigcup_{i=1}^{p} P_i\} && (49)\\
&= \{x | \exists u \in U, Ax + Bu \in P_1 \vee \ldots \vee \exists u \in U, Ax + Bu \in P_p\} && (50)\\
&= \bigcup_{i=1}^{p} \text{pre}_c(P_i) && (51)
\end{aligned}$$

□

Therefore, the predecessor operator commutes with unions of piecewise linear sets. Note that this lemma is a consequence of the equivalence $(\exists x)(\phi(x) \vee \psi(x)) \leftrightarrow (\exists x)\phi(x) \vee (\exists x)\psi(x)$ in predicate logic.

6.2.1 Continuous Disturbances

Here, we consider that continuous disturbances are present in the description of the system which for a fixed discrete mode is given by

$$x(t+1) = Ax(t) + Bu(t) + Ed(t) \tag{52}$$

where $A \in \Re^{n \times n}$, $B \in \Re^{n \times m}$, and $E \in \Re^{n \times p}$. It is also assumed that the control input u and the disturbance d take values in the polyhedral and bounded sets U and D respectively.

Consider the polyhedral set P represented by the following set of linear inequalities:

$$\begin{array}{rcl} g_1^T x & \leq & w_1 \\ \vdots & & \vdots \\ g_\nu^T x & \leq & w_\nu \end{array}$$

In this case, the predecessor operator takes the form

$$\text{pre}_c(P) = \{x \in X | \exists u \in U, \forall d \in D, Ax + Bu + Ed \in P\}. \tag{53}$$

Consider the following linear programming problems:

$$\begin{array}{ll} \min & -g_i^T Ed \\ \text{s.t.} & d \in D \end{array} \tag{54}$$

Since D is a bounded set the above linear programming problems have finite solutions. The corresponding solutions are denoted by $d_i^* = \text{argmin}_{d \in D}\{-g_i^T Ed\}$, $i = 1, \ldots, \nu$.

Proposition 3 *The set $pre_c(P)$ is given by*

$$pre_c(P) = \Pr(Q) \tag{55}$$

where $Q \subseteq X \times U$ is defined as

$$Q = \{(x,u) | \bigwedge_{i=1,\ldots,\nu} g_i^T Ax + g_i^T Bu \leq w_i - g_i^T Ed_i^*\}. \tag{56}$$

Proof If $x \in \text{pre}_c(P)$ then by definition there exists control input $u \in U$ such that the following set of inequalities holds for every $d \in D$, and therefore for $d = [d_1^*, \ldots, d_\nu^*]^T$ we have that

$$\begin{array}{rcl} g_1^T Ax + g_1^T Bu & \leq & w_1 - g^T Ed_1^* \\ \vdots & & \vdots \\ g_\nu^T Ax + g_\nu^T Bu & \leq & w_\nu - g_\nu^T Ed_\nu^* \end{array}$$

Therefore, there exists $u \in U$ such that $(x,u) \in Q$, which implies that $x \in \Pr(Q)$.

Conversely, assume that $x \in \Pr(Q)$ but $x \notin \text{pre}_c(P)$. Then, there exists $\bar{d} \in D$ and $i \in [1, \ldots, \nu]$ such that for every $u \in U$

$$g_i^T Ax + g_i^T Bu > w_i - g_i^T E\bar{d}. \tag{57}$$

But by the assumption that $x \in \Pr(Q)$ we have that there exists $u \in U$ such that

$$g_i^T Ax + g_i^T Bu \leq w_i - g_i^T Ed_i^* \leq w_i - g_i^T E\bar{d} \tag{58}$$

which is a contradiction. □

Note that we could first apply the Fourier-Motzkin elimination method for the elimination of control variables, and then solve the linear programming problems for the disturbance.

In the case the set P is piecewise linear but not polyhedral, then we can can compute the set $\text{pre}_c(P)$ without quantifiers by using appropriate equivalences from predicate logic. For example, in order to eliminate the universal quantifier of the disturbances for the set $P_1 \cup P_2$, we can use the logical equivalence

$$\forall d \in D, Ax + Bu + Ed \in P_1 \cup P_2 \leftrightarrow \neg(\exists d \in D, Ax + Bu + Ed \in P_1^c \cap P_2^c). \tag{59}$$

Then, the existential quantifier can be eliminated by writing the set $P_1^c \cap P_2^c$ in disjunctive normal form and apply the Fourier-Motzkin elimination method for each set of conjunctive constraints. Note that since the control variables $u \in U$ are independent of the disturbance variable $d \in D$, we can select the order for the elimination of quantifiers.

Example In order to illustrate, that the predecessor operator can be computed in a closed-form in a straightforward manner, we consider a piecewise linear set described by the logical formula

$$(\phi_1(x,u,d) \wedge \phi_2(x,u,d)) \vee (\phi_3(x,u,d) \wedge \phi_4(x,u,d)) \tag{60}$$

where ϕ_i corresponds to the linear constraint $g_i^T Ax + g_i^T Bu + g_i^T Ed \leq w_i$.

The computation of the set $\text{pre}_c(P)$ is equivalent to the quantifier elimination for the formula

$$(\exists u)(\forall d)(\phi_1(x,u,d) \wedge \phi_2(x,u,d)) \vee (\phi_3(x,u,d) \wedge \phi_4(x,u,d).$$

By applying simple logical equivalences we have

$$\begin{aligned} & (\exists u)(\forall d)(\phi_1(x,u,d) \wedge \phi_2(x,u,d)) \vee (\phi_3(x,u,d) \wedge \phi_4(x,u,d)) \\ \Leftrightarrow \quad & (\forall d)\left((\exists u)(\phi_1(x,u,d) \wedge \phi_2(x,u,d)) \vee (\phi_3(x,u,d) \wedge \phi_4(x,u,d)\right) \\ \Leftrightarrow \quad & (\forall d)\left((\exists u)(\phi_1(x,u,d) \wedge \phi_2(x,u,d))\right) \vee (\exists u)(\phi_3(x,u,d) \wedge \phi_4(x,u,d))\,. \end{aligned}$$

The elimination of the control variables can be accomplished by applying Fourier-Motzkin elimination. The resulting set can be written in disjunctive normal form to obtain the logical formula $\Psi(x,d)$. Then, the disturbance variables can be eliminated using the logical equivalence $(\forall d \in D)(\Psi(x,d)) \Leftrightarrow (\neg(\exists d \in D)\neg(\Psi(x,d))$. □

We have presented constructive algorithms for the computation of the predecessor operator for any piecewise linear region of the continuous state space. These algorithms use the Fourier-Motzkin elimination method, linear programming techniques, and simple equivalences from predicate logic. The algorithms were presented in analytical form and they can be implemented by software in a straightforward manner. These algorithms have been applied for reachability analysis of practical examples using MATLAB in Section 7.

Remark A special case of particular interest is the class of hybrid systems for which the control inputs take values in a finite set. This is a rather important class of systems since it can be used to model many practical applications. For example, chemical processes usually involve actuators that can be modeled using discrete variables such as valves and compressors. Discrete control inputs arise also in the motion control of many systems such as satellites or underwater vehicles. Note that in this case the projection $\mathrm{Pr}(Q)$ can be computed as the union of the sets that result by substituting each possible value for the control input. This method, however, will lead to many redundant constrains. The procedure to eliminate these redundant constrains requires additional computational effort. A methodology for reachability analysis in the case of discrete control inputs based on mathematical programming techniques has been presented in [19].

6.3 Algorithms for Backward Reachability Analysis

Consider a PLHDS described by the equations (13) - (15) and a region $R = (q,P)$. We denote the quotient space X/E_π induced by the primary partition as $X/E_\pi = \{P_i\}, i = 1,\ldots,|\pi|$. In addition, let $\mathrm{pre}_{c,q} : 2^X \to 2^X$ denote the predecessor operator for a continuous transition described by the discrete mode q. The following algorithm computes all the states of the hybrid system that can be driven to R in one time-step. The algorithm is implemented using the technical results presented earlier in this section.

Algorithm for the computation of pre(R)

INPUT: $R = (q,P),\ S = \emptyset,\ T = \emptyset$;
for $i = 1,\ldots,|\pi|$
  $Q_i = P \cap P_i$;
  if $Q_i \neq \emptyset$
    for $q' \in \mathrm{act}(P_i)$
      $S = S \cup \mathrm{pre}_{c,q'}(Q_i)$;
      $T = T \cup \{q'\}$;
    end
end
OUTPUT: $\mathrm{pre}(R) = (T,S)$

The algorithm computes all the regions of the state space for which the state can be driven to R by a continuous transition. In order to consider only the discrete modes that are feasible at each region of the state space, we write the set P as a union of regions of the initial partition. Then, we add also the states that can be driven to R by discrete transitions. If the initial region R contains more than one discrete states, then the algorithm is applied for each individual state.

We have shown 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

$$\text{pre}^N(R) = \overbrace{\text{pre}(\cdots \text{pre}(R))}^{N \text{ times}}. \tag{61}$$

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

$$CR(R) = \text{pre}^*(R). \tag{62}$$

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. In Section 8, we present a grid-based approximation technique that can be used to formulate a termination condition for the successive application of the predecessor operator.

In this section, we have presented a systematic approach for the computation of the predecessor operator for piecewise linear hybrid dynamical systems. The developed algorithms can be used for backward reachability analysis and partition refinement. The predecessor operator can be computed in a closed form for any piecewise linear region of the hybrid state space. Here, we comment on the computational complexity of the presented algorithms for backward reachability analysis.

Infinite time problems for piecewise linear systems are, in general, undecidable [37]. We have presented semi-decidable procedures for backward reachability analysis. For finite time problems, backward reachability algorithms for piecewise linear hybrid systems are NP-complete [37]. This follows from the definition of the predecessor operator which is formulated using the existential quantifier over all possible inputs. Practically, the number of linear constraints that are used to represent the coreachable region grows exponentially at every iteration of the algorithm.

The developed algorithms can be used for practical applications if they involve only a reasonable number of iterations. For example, it is shown in Section 7 that we can formulate conditions that guarantee that a piecewise linear region is safe by considering only one iteration.

7 Safety

In the following, 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.

Definition 3 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.

Our objective is to formulate conditions on the available controls, so that a given set is safe for a PLHDS. In order to study safety specifications for piecewise hybrid dynamical systems, we introduce the notion of quasideterminism.

Quasideterminism represents the case when the future behavior only for the next time interval of the actual system can be uniquely determined by the current state of the induced system. We show that this property can be used to formulate conditions for safety specifications for piecewise linear hybrid dynamical systems.

7.1 Quasideterminism

Quasideterminism can be viewed as a desirable property of the partition of the continuous state space. The central characteristic of quasideterministic systems is that only the reachability properties with respect to the safety specifications are preserved in the quotient system. Quasideterminism is a weaker requirement than the existence of a finite bisimulation. A partition that results in quasideterminism can be always computed for piecewise-linear systems, while recent results have shown that finite bisimulations exist only for limited classes of systems [21]. In both approaches an algorithm is used to refine the state space. A bisimulation corresponds to a fixed point of the refinement algorithm. In quasideterminism, we do not require the existence of a fixed point but we stop the refinement at a prescribed fixed iteration. The disadvantage of that is that in this case the quotient system does not completely preserve the reachability properties of the original system, however this is not needed for controller design for an interesting class of problems as this work demonstrates.

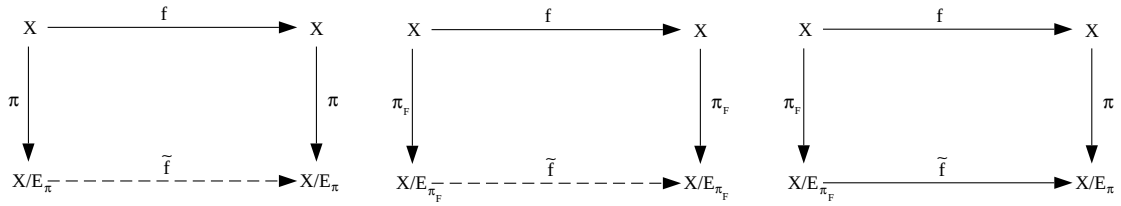
7.1.1 Measurements and Final Partition

Suppose that at time t, $\pi(x(t)) \in X/E_\pi$ is known. The signal $x(t)$ represents the state of the system at the t^{th} successive iteration of the system. If it is agreed that the granularity of the primary partition is appropriate for the extraction of useful information regarding the system's behavior, then it is desirable to uniquely determine the state at the next iteration up to its membership on an equivalence class $y(t+1) = \pi(x(t+1)) \in X/E_\pi$. This can be accomplished by considering a finer partition than the primary partition defined by the generator π to obtain better estimates for the continuous state. This partition will be called the *final partition*.

The final partition is defined by a mapping $\pi_F : X \to 2^X$ in a similar way as the primary partition is defined by π. Given a partition defined by a finite set of $(n-1)$-dimensional hyperplanes, the generator $\pi_F : X \to X/E_{\pi_F}$ separates the state space into a finite number of equivalence classes which correspond to polyhedral regions in $\Re^n$. The function $z = \pi_F(x)$ can be seen as a *measurement function* that provides the membership of the state to one of the equivalence classes of E_{π_F}. Intuitively, our ability to make decisions to influence the behavior of the system depend on the amount of information contained in the measurement signal.

In the case when the estimates of the state at time t provide sufficient information to uniquely determine the membership of the state of the induced system at time $t+1$ on an equivalence class of E_π, the system is said to be quasideterministic. The notion of quasidetermism is illustrated in Figure 10. Although we do not compute an equivalence relation that guarantees the existence of a mapping $\tilde{f}$ that preserves the reachability properties of the original system, we exploit the commutativity of the diagram (c) in Figure 10 in order to analyze the reachability properties with respect to the safety specifications.

Definition 4 A piecewise linear hybrid dynamical system with primary and final partition defined by X/E_π and X/E_{π_F} is quasideterministic with respect to the primary partition if for every region of the final partition $Z_i \in X/E_{\pi_F}$ and for all states $x \in X$ such $\pi_F(x) = Z_i$, there exists unique region of the primary partition $P_i \in X/E_\pi$ such that $P_i = \pi(x(t+1))$ for every feasible discrete transition $(q, q'), q' \in \text{act}(\pi(x(t))$, control action $u \in U$ and disturbance $d \in D$.



(a) Primary Partition (b) Final Partition (c) Quasideterministic Partition

Figure 10: Quasideterminism and the partitions of the state space.

In Section 5, we showed that given a piecewise linear region $R \subset Q \times X$, the set pre(R) of all the states that can be driven to R by either a continuous or a discrete transition is piecewise linear, and therefore, can be described using a finite set of linear inequalities. Next, consider the hyperplanes $h'_i(x)$ that correspond to the linear inequalities that define the set pre(R) and the partition $\pi' \in E_P(X)$ defined by those hyperplanes using the following equations:

$$\pi'(x) = [\hat{h'}_1(x), \ldots, \hat{h'}_\ell(x)]^T \tag{63}$$

where

$$\hat{h'}_i(x) = \begin{cases} -1 & \text{if} \quad h'_i(x) < 0 \\ 0 & \text{if} \quad h'_i(x) = 0 \\ 1 & \text{if} \quad h'_i(x) > 0 \end{cases} \tag{64}$$

Theorem 1 *Consider a piecewise linear hybrid dynamical system with primary partition defined by E_π and let the partition generated by applying the predecessor operator pre $: 2^{Q\times X} \to 2^{Q\times X}$ to the regions of the initial partition defined by $E_{\pi'}$. Then the piecewise linear hybrid dynamical system with final partition defined by $E_{\pi_F} = \inf(E_\pi, E_{\pi'})$ is quasideterministic with respect to the primary partition.*

Proof Consider an equivalence class $Z_j \in X/E_{\pi_F}$. Z_j corresponds to an polyhedral region of $\Re^n$. Since $E_{\pi_F} = \inf(E_\pi, E_{\pi'})$, for every $P_i \in X/E_\pi$ we have that either $Z_j \subseteq \text{pre}_c(P_i)$ or $Z_j \cap \text{pre}_c(P_i) = \emptyset$. Consider a continuous state $x \in Z_j$, then by the definition of the predecessor operator we have that $x(t+1) = A_{q(t)}x(t) + B_{q(t)}u(t) + E_{q(t)}d(t) \in P_i$ if and only if $Z_j \subseteq \text{pre}_c(P_i)$. Therefore, for each $(q, q'), q' \in \text{act}(\pi(x(t))$, for each $u \in U$, and for every $d \in D$, the membership of the continuous state $x(t+1)$ in an equivalence class of X/E_π can be uniquely determined from the membership of the state $x(t)$ in an equivalence class of X/E_{π_F}. □

The implication of the above proposition is that for every state, every control action, and every disturbance the membership of the state at the next time step to an equivalence class of the primary partition can be uniquely determined from the current region of the final partition. This information can be used to determine if the set P is safe.

Remark If the PLHDS with primary and final partition defined by X/E_π and X/E_{π_F} is *quasideterministic* with respect to the primary partition π, then it is also quasideterministic if instead of E_{π_F} we use any finer final partition such that $E_{\pi_N} \leq E_{\pi_F}$. This can be shown by considering a region $Z'_i \in E_{\pi_N}$. By the definition of the partial order in the equivalence lattice, for every $Z'_i \in E_{\pi_N}$, there exists a unique $Z_j \in E_{\pi_F}$ so that $\pi_N(x) = Z'_i \Rightarrow \pi_F(x) = Z_j$. Therefore, every $Z'_i \in E_{\pi_N}$ corresponds to a unique equivalence class of E_{π_F}, for which the membership of the continuous state $x(t+1) = A_{q(t)}x(t) + B_{q(t)}u(t) + E_{q(t)}d(t)$ in an equivalence class of X/E_π can be uniquely determined.

7.2 Safety Conditions

In this section, we formulate conditions that guarantee that a given region of the hybrid state space is safe. The conditions can be efficiently tested using linear programming techniques.

Theorem 2 *A PLHDS is safe with respect to the region $R \subseteq Q \times X$ if and only if $R \subseteq pre(R)$.*

Proof If $R \subseteq \text{pre}(R)$, every state $(q, x) \in \text{pre}(R)$ and therefore every state $(q, x) \in R$ can be driven in R, either by selection of appropriate control input $u \in U$ or by triggering a discrete transition and therefore, the system is safe.

Conversely, assume that the system is safe and consider there exists control policy such that $x(t) \in R$ for every t. By definition, the set pre(R) is the set of all the states for which there exists control policy so that the next state will be in R. Therefore, since the system is safe for every $x \in R$ we have that $x \in \text{pre}(R)$. □

In the following, we present a constructive algorithm which is used to test the condition $R \subseteq \text{pre}(R)$. Let $R|X$ and pre$(R)|X$ be the projection of R and pre(R) into the continuous state space X. Similarly $R|Q$ and pre$(R)|Q$ for the discrete state space Q. Since, the sets $R|Q$ and pre$(R)|Q$ are finite, we can test whether $R|Q \subseteq \text{pre}(R)|Q$ in a straightforward manner. Next, we concentrate on the continuous part of the regions R and pre(R). 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 $R|X \subseteq \text{pre}(R)|X$, we represent the constraints in disjunctive normal form (DNF) and we test the feasibility of finite set of linear programming problems.

Using Lemma 4.2, every PL set can be written as a union of polyhedral sets using the disjunctive normal form representation. Therefore, we can assume that the set $R|X$ and the complement of the set pre$(R)|X$ can be written as

$$R|X = \bigcup_{i=1,\ldots,|P|} P_i \tag{65}$$

and

$$[\text{pre}(R)|X]^c = \bigcup_{j=1,\ldots,|Q|} Q_j \tag{66}$$

where P_i and Q_j are polyhedral, and therefore convex sets in $\Re^n$. For each pair (i, j) the set $C_{ij} = P_i \cap Q_j$ is polyhedral as the intersection of polyhedral sets. Furthermore, the condition $P_i \cap Q_j = \emptyset$ can be tested by solving the following linear programming problem:

$$\begin{array}{ll} \min & x \\ \text{s.t.} & x \in C_{ij} \end{array} \tag{67}$$

We have that $P_i \cap Q_j = \emptyset$ if and only if the above linear programming is infeasible. Therefore, we have that $R \subseteq \text{pre}(R)$ and the PLHDS is safe if and only if $P_i \cap Q_j = \emptyset$ for every $i = 1, \ldots, |P|$ and $j = 1, \ldots, |Q|$.

Example - Temperature Control System In the following, we use the temperature control system presented in Section 4 to illustrate how we can formulate the safety conditions. The temperature control system is modeled by the PLHDS described in Equations (18)- (20). We consider the system parameters shown in Table 1.

The discrete state q_1 corresponds to the case the furnace is on. Using zero-order hold sampling with $T = 1$, the continuous dynamics are described by the difference equation

$$x(t+1) = A_1 x(t) + B_1 u(t) + E_1 d(t) \tag{68}$$

Furnace ON	*Furnace OFF*
q_1	q_0
$R_1 = 2$	$R_3 = 10$
$R_2 = 1$	$R_4 = 2$
$C_1 = 1$	$C_3 = 0.5$
$C_2 = 1$	$C_4 = 1$
$U_1 = [0.5, 5]$	$U_0 = 0$
$D_1 = [0, 1]$	$D_0 = [-1, 0]$

Table 1: Parameters for the temperature control system

where

$$A_1 = \begin{bmatrix} -0.6634 & 0.1997 \\ 0.1997 & 0.2641 \end{bmatrix},\ B_1 = \begin{bmatrix} 0.8101 \\ 0.1369 \end{bmatrix},\ E_1 = \begin{bmatrix} 0.1369 \\ 0.5363 \end{bmatrix}, \tag{69}$$

and $u(t) \in U_1, d(t) \in D_1$.

Similarly, for the discrete state q_0 (furnace off), we have

$$x(t+1) = A_0 x(t) + B_0 u(t) + E_0 d(t) \tag{70}$$

where

$$A_0 = \begin{bmatrix} 0.8259 & 0.1354 \\ 0.0677 & 0.5551 \end{bmatrix},\ B_0 = \begin{bmatrix} 1.8179 \\ 0.0773 \end{bmatrix},\ E_0 = \begin{bmatrix} 0.0387 \\ 0.3772 \end{bmatrix}, \tag{71}$$

and $u(t) \in U_0, d(t) \in D_0$.

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

$$h_1(x) = x_1 - M,\ M = 20 \tag{72}$$

$$h_2(x) = x_2 - ht,\ ht = 5 \tag{73}$$

$$h_3(x) = x_2 - lt,\ lt = 0 \tag{74}$$

$$h_4(x) = x_1 \tag{75}$$

and it is shown in Figure 11.

It is assumed that the safe region is described by the set $R = \{(q_0, q_1), P\}$ where P is given by

$$P = \{x \in \Re^2 | (0 \le x_1 \le M) \wedge (lt \le x_2 \le ht)\}. \tag{76}$$

Next, we describe in detail the algorithm for the computation of the set pre(R). We represent the set P as $P = \{x | Gx \le w\}$ where

$$G = \begin{bmatrix} g_1^T \\ g_2^T \\ g_3^T \\ g_4^T \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & -1 \\ -1 & 0 \end{bmatrix} \tag{77}$$

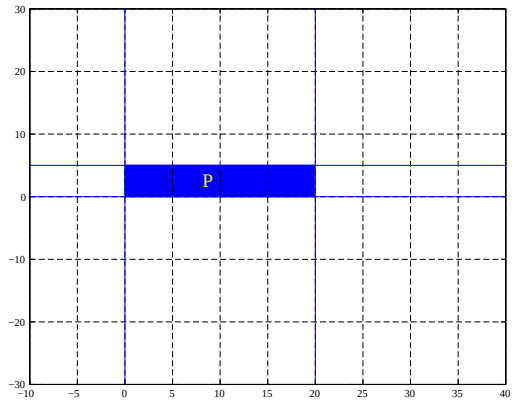


Figure 11: Primary partition for the temperature control system.

and

$$w = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix} = \begin{bmatrix} 20 \\ 5 \\ 0 \\ 0 \end{bmatrix}. \tag{78}$$

First, we compute the set

$$\text{pre}_{c,q_0}(P) = \{x | A_0 x + B_0 u + E_0 d \in P\}. \tag{79}$$

Note that if the system is at mode q_0, the input is $u = 0$. Using Proposition 4.3, we consider the following set of linear inequalities:

$$GA_0 x \leq w - GE_0 d \tag{80}$$

We solve the linear programming problems

$$\begin{array}{ll} \min & -g_i^T E_0 d \\ \text{s.t.} & d \in D_0 \end{array} \tag{81}$$

for $i = 1, 2, 3, 4$ and we obtain $[d_1^*, d_2^*, d_3^*, d_4^*] = [0, 0, -1, -1]$. By substituting in Equation (80) we get

$$\text{pre}_{c,q_0}(P) = \left\{ x \in \Re^n | \begin{bmatrix} 0.8259 & 0.1354 \\ 0.0677 & 0.5551 \\ -0.0677 & -0.5551 \\ -0.8259 & -0.1354 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \leq \begin{bmatrix} 20 \\ 5 \\ -0.3772 \\ -0.0387 \end{bmatrix} \right\}. \tag{82}$$

Next, we compute the set

$$\text{pre}_{c,q_1}(P) = \{x | A_1 x + B_1 u + E_1 d \in P\}. \tag{83}$$

We consider the following set of linear inequalities:

$$GA_1 x + GB_1 u \leq w - GE_1 d \tag{84}$$

$$u \leq 1 \tag{85}$$

$$-u \leq -0.5 \tag{86}$$

We apply the Fourier-Motzkin elimination method in order to eliminate the control variable u. We also solve the linear programming problems

$$\begin{array}{ll} \min & -g_i^T E_1 d \\ \text{s.t.} & d \in D_1 \end{array} \tag{87}$$

for $i = 1, 2, 3, 4$ and we obtain $[d_1^*, d_2^*, d_3^*, d_4^*] = [1, 1, 0, 0]$. Using Proposition 3 we have that

$$\text{pre}_{c,q_1}(P) = \left\{ x \in \Re^2 | \begin{bmatrix} 0.6634 & 0.1997 \\ 0.1997 & 0.2641 \\ -0.1997 & -0.2641 \\ -0.6634 & -0.1997 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \leq \begin{bmatrix} 19.4580 \\ 4.3953 \\ 0.6847 \\ 4.0507 \end{bmatrix} \right\}. \tag{88}$$

The sets $\text{pre}_{c,q_0}(P)$ and $\text{pre}_{c,q_1}(P)$ are shown in Figure 12.

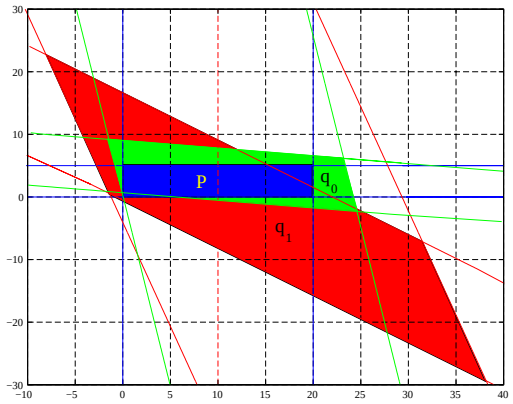


Figure 12: Final partition for the temperature control system.

The set $\text{pre}(R)$ is computed using the algorithm presented in Subsection 6.3. Since both discrete transitions (q_0, q_1) and (q_1, q_0) are feasible we get

$$\text{pre}(R) = \{(q_0, q_1), \text{pre}_{c,q_0}(P) \cup \text{pre}_{c,q_1}(P)\}. \tag{89}$$

In the following, we illustrate how we can test the safety condition $R \subseteq \text{pre}(R)$. The set $\text{pre}(R)|X$ can be represented by the logical formula

$$(\phi_{01} \wedge \phi_{02} \wedge \phi_{03} \wedge \phi_{04}) \vee (\phi_{11} \wedge \phi_{12} \wedge \phi_{13} \wedge \phi_{14}) \tag{90}$$

where the atomic formulas ϕ_{ij} correspond to the linear inequalities that define the sets $\text{pre}_{c,q_0}(P)$ and $\text{pre}_{c,q_1}(P)$ in Equations (82) and (88) respectively. We define the set $Q = [\text{pre}(R|X)]^c$. Using DeMorgan's laws, the set Q can be represented by

$$\begin{aligned} & \neg[(\phi_{01} \wedge \phi_{02} \wedge \phi_{03} \wedge \phi_{04}) \vee (\phi_{11} \wedge \phi_{12} \wedge \phi_{13} \wedge \phi_{14})] \\ \Leftrightarrow \quad & \neg(\phi_{01} \wedge \phi_{02} \wedge \phi_{03} \wedge \phi_{04}) \wedge \neg(\phi_{11} \wedge \phi_{12} \wedge \phi_{13} \wedge \phi_{14}) \\ \Leftrightarrow \quad & (\neg\phi_{01} \vee \neg\phi_{02} \vee \neg\phi_{03} \vee \neg\phi_{04}) \wedge (\neg\phi_{11} \vee \neg\phi_{12} \vee \neg\phi_{13} \vee \neg\phi_{14}) \\ \Leftrightarrow \quad & \bigcup_{i=1,2,3,4,\ j=1,2,3,4} (\neg\phi_{0i} \wedge \neg\phi_{1j}) \end{aligned}$$

Therefore, the set Q can be written as $Q = \bigcup_{i,j} Q_{ij}$. Each set Q_{ij} is described by the logical formula $(\neg\phi_{0i} \wedge \neg\phi_{1j})$ and therefore, it is polyhedral. The condition $R|X \subseteq \text{pre}(R)|X$ can be checked by testing the feasibility of the linear programming problems:

$$\begin{array}{ll} \min & x \\ \text{s.t.} & x \in P \cap Q_{ij} \end{array} \tag{91}$$

For the temperature control system, we have that $R|Q = \text{pre}(R)|Q = \{q_0, q_1\}$ and $P \cap Q_{ij} = \emptyset$ for $i = 1, 2, 3, 4$ and $j = 1, 2, 3, 4$. Therefore, the region R is safe as it can be seen in Figure 12. □

7.3 Maximal Safe Set

In Subsection 7.2, we formulated conditions that guarantee that a given region R is safe. In the case when the safety conditions are not satisfied and R is not safe, it is possible that there exists a region $R' \subset R$ which is safe. Such a region can be computed as the maximal safe set contained in R. The problem of computing the maximal safe set for hybrid systems has been studied in [41, 25, 44]. Here, we present how the algorithm presented in [41] can be applied to PLHDS.

Algorithm for the computation of maximal safe set

INPUT: $R^0 = X;\ R^1 = R;\ k = 1;$
while $R^k \neq R^{k-1}$
$\quad R^k = \text{pre}(R^{k-1}) \cap R^{k-1};$
$\quad$ k = k+1;
end
OUTPUT: $R^* = R^k$

The maximal safe set is computed as a fixed point of the iterative procedure described above. At the kth iteration of the algorithm, we compute the set $R^k = \text{pre}(R^{k-1}) \cap R^{k-1}$ which contains all the states in R^k for which there exist controls so that the state will remain in R^k. If there exists a fixed point iteration, then clearly the corresponding set R^* is safe. Furthermore, we have that $R^k \subseteq R^{k-1}$ and therefore the set R^* is the maximal safe set contained in R.

The algorithm involves the computation of the predecessor operator at every iteration. In [41, 25], this computation is accomplished by solving a Hamilton-Jacobi-Bellman equation derived from a game theoretical formulation of the problem. In the case of PLHDS, the predecessor operator can be computed using the algorithms for elimination of quantifiers presented in Section 5. The proposed procedure is semi-decidable. If the algorithm terminates, it provides the maximal safe piecewise linear set contained in R, however its termination is not guaranteed.

The advantage of computing the maximal safe set of a PLHDS, is that based on this set, a controller can be designed which is maximally permissive. Such a controller is optimal in a sense, since it does not restrict the behavior of the plant in a conservative way. However, the algorithm for computing the maximal safe set is not computationally efficient. For PLHDS, the number of linear constraints increases exponentially at each iteration of the algorithm. On the other hand, the safety conditions presented in Subsection 7.2 do not guarantee that the corresponding controller will be maximally restrictive. However, they provide constructive conditions that guarantee that a given region is safe and they can be used to determine what are the appropriate control inputs that guarantee safety. A class of discrete-time systems for which this procedure is decidable has been presented in [42].

8 Reachability

In this section, we study the reachability problem for piecewise linear hybrid dynamical systems. We present a reachability algorithm based on the successive computation of the predecessor operator. In general, 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.

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 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. Consider, for example, an unmanned underwater vehicle with control policies that allow various combinations of screw speeds (on and off), stern plane positions (up, level, down), and rudder positions (left, right, straight). A control goal for such a system can be described by a target region of the state space which represents a desirable set of displacements and velocities for the vehicle. However, while the system is driven to the target regions the dispacements and velocities must be appropriety constrained to guarantee safe operation.

Definition 5 Given two regions $R_1, R_2 \subseteq Q \times X$, we say that R_2 is *directly reachable* from R_1, if every state $(q, x) \in R_1$ can be driven in R_2 in finite time without entering a third region.

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. 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. Note that if the regions $R_1 = (q_1, P)$ and $R_2 = (q_2, P)$ have identical continuous parts, then the reachability problem can be solved by considering the set of feasible transitions for the polyhedral region P.

8.1 Finite Time Horizon

Consider the regions R_1 and R_2 and the initial state $(q, x) \in R_1$ and assume that we can disable the state from crossing all the boundaries of R_1 but $h(x)$. It is still possible that the hybrid system will be blocked in the sense that the state will never exit the region R_1 through the hyperplane $h(x)$. Note that this can happen since we want to drive the state from R_1 to R_2 without entering a third region. In this case there is a trade-off between driving the state into the target region and satisfy the constraints for the state trajectory. The risk of violating the operational conditions of a system while stirring the state to a desired operating point must be addressed. Thus, this formulation of the reachability problem that takes into consideration constraints in the state trajectory is more important than considering only the state into the target region, both in theory and in practice.

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 (13)-(15) and assume that the initial condition is $(q(t_0), x(t_0)) \in R_1$.

Definition 6 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 $t_0 \leq t_0 < t_0 + kt$ and $(q(t_0 + kt), x(t_0 + kt)) \in R_2$..

We define the *coreachable set* $CR^N_{R_1}(R_2)$ 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 predecessor operator pre : $2^{Q\times X} \rightarrow 2^{Q\times X}$ can be used to compute the set $CR^N_{R_1}(R_2)$ using the following algorithm.

Algorithm for the computation of $CR^N_{R_1}(R_2)$

$R^0 = R_2$;
$CR^N_{R_1}(R_2) = \emptyset$;
$k = 0$;
while $\neg(R^{k+1} \subseteq R^k)$ AND $k \leq N$
    $R^{k+1} = \text{pre}(R^k) \cap R_1$;
    $CR^N_{R_1}(R_2) = CR^N_{R_1}(R_2) \cup R^{k+1}$;
    $k = k + 1$;
end

Given the regions R_1 and R_2, we compute all the states that can driven from R_1 to R_2. Note that 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 have to apply the predecessor operator to a piecewise linear region of the state space. The resulting region is still piecewise linear, it can be represented using only linear equalities and inequalities, and it can be computed using the algorithms for elimination of quantifiers presented in Section 5. The above algorithm can be used to determine if the region R_2 is N-reachable from R_1 using the following theorem.

Theorem 3 *Consider a PLHDS described by (13)-(14) and the regions $R_1 = (Q_1, P_1)$ and $R_2 = (Q_2, P_2)$. Then, the set set $CR^N_{R_1}(R_2)$ is piecewise linear and the region R_2 is directly N-reachable from R_1 if and only if $R_1 \subseteq CR^N_{R_1}(R_2)$.*

Proof The set set $CR^N_{R_1}(R_2)$ is piecewise linear since it is computed using finite unions and intersections of piecewise linear sets. At the k iteration of the algorithm, the set R^k contains all the states in R_1 that can be driven in R_2 in $t \leq kT$. If $R_1 \subseteq CR^N_{R_1}(R_2)$ then there exists controls so that every state $(q, x) \in R_1$ can be driven to R_2 in $t \leq NT$ without entering a third region. □

Furthermore, 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 (see Subsection 7.2).

For regions that are not adjacent, a feasible path connecting these regions which consists of adjacent must be established. Note that this can be done at the higher level of abstraction, since the necessary information is the existence of a control policy and not the actual policy.

Example - Temperature Control System We illustrate the reachability algorithm using the temperature control system presented in Subsection 4.

Consider the regions $R_1 = (\{q_0, q_1\}, P_1)$ and $R_2 = (\{q_0, q_1\}, P_2)$ where

$$P_1 = \{x \in \Re^2 | (0 \leq x_1 \leq 20) \wedge (-20 \leq x_2 \leq 0)\} \tag{92}$$

and

$$P_2 = \{x \in \Re^2 | (0 \leq x_1 \leq 20) \wedge (0 \leq x_2 \leq 5)\}. \tag{93}$$

It is desirable that every state from R_1 can be driven to R_2 without entering a third region. Such a specification may arise, for example, at the startup procedure of the system, where it is required for the state of the system to reach the safe region $0 \leq x_2 \leq 5$ without entering the unsafe region $x_1 > 20$.

In order to compute the set of states in the region R_1 that can be driven to R_2 using appropriate control inputs, we apply the reachability algorithm presented in Subsection 8.1. The coreachable set of states for three iterations of the algorithm is shown in Figure 13.

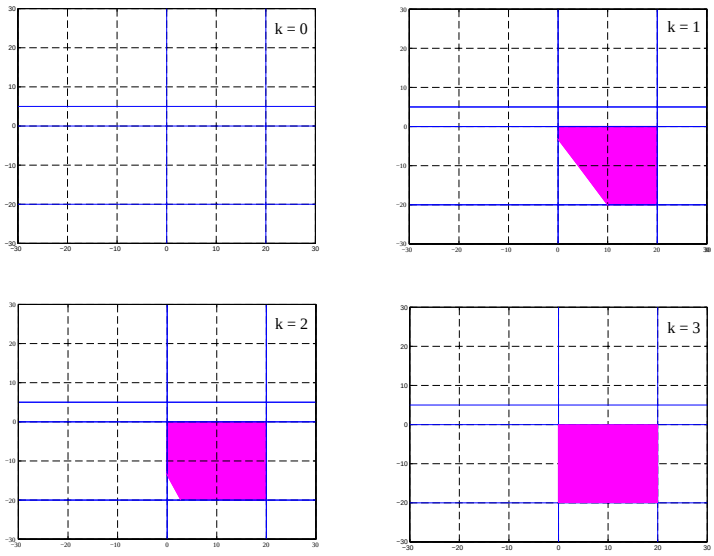


Figure 13: Coreachable set for the temperature control system.

The region R_2 is directly reachable from R_1 in $t = 3T$. Therefore, there exists control policy which selects the control input $u \in U$ and possibly forces appropriate discrete transitions so that every state $(q, x) \in R_1$ can be driven to the region R_2. □

8.2 Grid-Based Approximation

In this section, we formulate an approximation-based methodology in order to guarantee that the algorithm for the successive computation of the predecessor operator will terminate. The reachability algorithm based on the successive computation of the predecessor operator is semi-decidable and therefore, its termination is not guaranteed. In the following, we present such an example.

Example Consider the discrete-time linear system $x(t+1) = Ax(t)$ with

$$A = \begin{bmatrix} 1.1036 & -0.0315 \\ 0.1051 & 0.9984 \end{bmatrix}. \tag{94}$$

Suppose we are given the partition shown in Figure 14 described by the following hyperplanes:

$$h_1(x) = x_1 + 3 \tag{95}$$
$$h_2(x) = x_1 + 2 \tag{96}$$
$$h_3(x) = x_2 - 1 \tag{97}$$
$$h_4(x) = x_2 + 1 \tag{98}$$

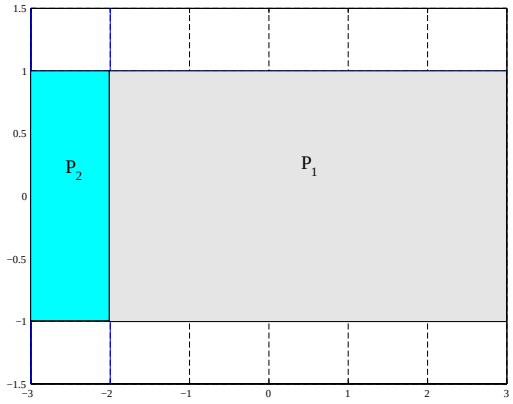


Figure 14: Primary partition for the system.

We consider the regions

$$P_1 = \{x \in \Re^n | (x_1 \leq -2) \wedge (-1 \leq x_2 \leq 1)\} \tag{99}$$

and

$$P_2 = \{x \in \Re^n | (-3 \leq x_1 \leq -2) \wedge (-1 \leq x_2 \leq 1)\} \tag{100}$$

and our objective is to test if the region P_2 is reachable from P_1.

The linear system $x(t+1) = Ax(t)$ is an unstable system with complex conjugate eigenvalues $1.0510 \pm j0.0235$. We use the reachability algorithm to compute the set of states in P_1 that can be driven in P_2. At the every iteration of the reachability algorithm we have that

$$P^k = \text{pre}_c(P)^k \cap P_1 \neq \emptyset. \tag{101}$$

Therefore, we add new states to the coreachable set of P_2 at every iteration and the algorithm will not terminate.

In Figure 15 we show the linear constraints computed by the algorithm by applying successively the predecessor operator for twenty iterations. □

In order to guarantee that the reachability algorithm will always terminate we formulate a practical termination condition. The termination condition is based on quantization of the state space. The basic idea is that the algorithm should terminate if the set pre(R^k) is not "substantially different" than the set pre(R^{k-L}). By "substantially different" we mean whether new cells of the quantized space have been added to the set of states that can be driven to R. L is a parameter selected by the designer and depends on the sampling period and the quantization of the state space. If

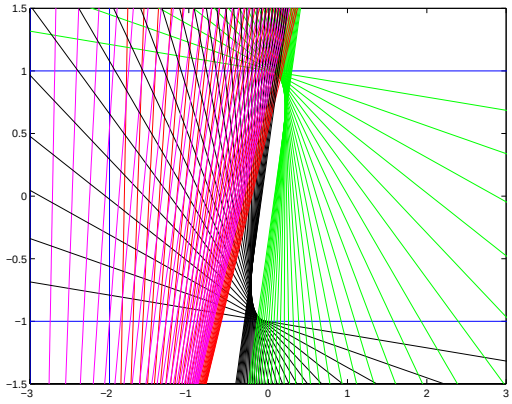


Figure 15: The backward reachability algorithm does not necessarily terminate.

the sampling period is small and the quantization levels are large, it is possible that no new states will be added in the coreachable set in one time step and we have to use a parameter $L > 1$.

First, we select quantization levels Δx_i for each continuous state $x_i \in \Re$, and the range of each state $x_{i,\min}$ and $x_{i,\max}$ which is assumed to be bounded. These choices lead to a quantization of the plant state space into a finite number of n-dimensional cells.

A given piecewise linear set $P \subset \Re^n$ is then approximated by the union of all the cells that belong to the set. The membership of a cell to the set P is defined as follows. A n-dimensional cell satisfies the constraint $g^T x \leq w$ if and only if all the vertices of the cell satisfy the constraint. The cell belongs to the set P if it satisfies all the constraints that define P. We formally define this approximation technique using the mapping grid : $2^{\Re^n} \rightarrow 2^{\Re^n}$. The set grid$(P)$ is defined as the union of all the cells that belong to the set P (see Figure 16). The set grid(P) is a conservative approximation of P since $x \in$ grid(P) implies that $x \in P$.

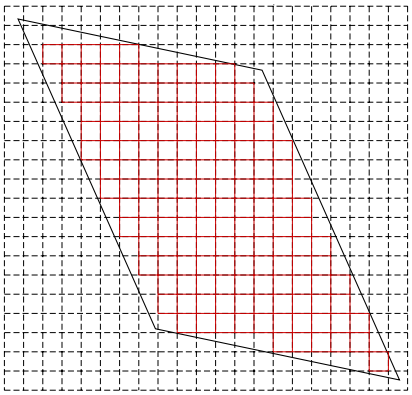


Figure 16: Grid-based approximation of a piecewise linear set

Consider a region $R = (T, P)$, we define the *coreachable set* $CR(R)$ as the set of all states that can be driven to R. The following algorithm illustrates how the grid-based approximation can be used the computation of the coreachable set $CR(R)$ for the region R.

Algorithm for the computation of $CR(R)$ using the grid-based approximation

$R^0 = R$;

$G^0 = \text{grid}(P)$;
$T^0 = R|Q$;
while $\neg(\text{pre}(R^k) \subseteq R^k)$
    $R^{k+1} = R^k \cup \text{pre}(R^k)$;
    $G^{k+1} = \text{grid}(P^{k+1})$;
    $T^{k+1} = R^{k+1}|Q$;
    if $k+1 > L$ then $j = k+1-L$ else $j = 0$;
    if $G^{k+1} \subseteq G^j$ and $T^{k+1} \subseteq T^j$ then exit;
end

The above algorithm computes the coreachable set for the region R by successive application of the predecessor operator. At the kth iteration, the algorithm computes the set $\text{pre}(R^k)$ of states that can be driven to the region R in k time steps. Note that $\text{pre}(R^k) \subseteq \text{pre}(R^{k+1})$ since at every iteration of the algorithm we add more reachable states. The algorithm will terminate if no new cells are added to the coreachable set for L iterations. The algorithm is guaranteed to terminate since by the quantization assumption we consider finitely many n-dimensional cells. Note that the approximation of the set $\text{pre}(R^k)$ is used only in the termination condition. The algorithm proceeds for the computation of the set $\text{pre}(R^{k+1})$ using the exact representation of $\text{pre}(R^k)$, therefore there is no accumulation of error due to the approximation.

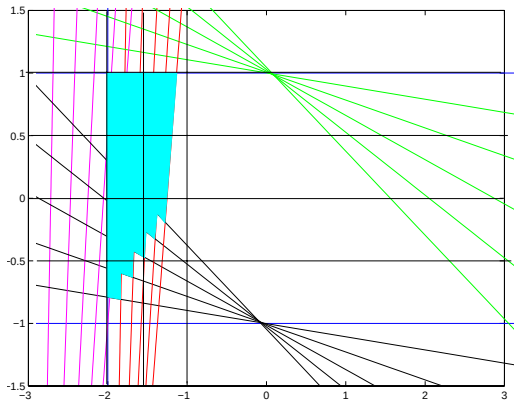


Figure 17: Grid-based approximation for reachability.

Example Consider the linear system $x(t+1) = Ax(t)$ presented earlier in the section. Figure 17 shows an approximation of the set of states from P_1 that can be driven to P_2. The quantized levels for the example are $\Delta x_i = 0.5$ and the design parameter $L = 2$. In this case the coreachable set can be underapproximated by the shaded region since in the last $L = 2$ iterations of the algorithm no new were added to the coreachable set. □

9 Conclusions

In this paper, a mathematical model that can capture both discrete and continuous phenomena is formulated. The continuous dynamics are described by linear difference equations and the discrete dynamics by finite automata. The inter-

action between the continuous and discrete part is defined by piecewise linear maps. 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. 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 new methodology for the construction of discrete abstractions of the continuous dynamics. The main characteristic of the approach is that the available control inputs are taken into consideration in order to simplify the system. The predecessor operator for piecewise linear systems is defined and computer algorithms for refining the partition of the state space are developed. Furthermore, we formulate conditions for safety and reachability specifications for piecewise linear hybrid dynamical systems. In order to study safety specifications for piecewise hybrid dynamical systems, we introduce the notion of quasideterminism. Quasideterminism represents the case when the future behavior only for the next time interval of the actual system can be uniquely determined by the current state of the induced system. We show that this property can be used to formulate conditions for safety specifications for piecewise linear hybrid dynamical systems. The safety conditions can be tested using efficient linear programming techniques. Reachability conditions are also formulated. Our approach is based on conditions that guarantee that the state can be forced to reach a desirable region of the state space by selecting appropriate controls. The main advantage of the proposed approach is that it provides a convenient general framework not only for analysis, but more importantly for controller synthesis.

Practical hybrid systems are often characterized by nonlinear continuous dynamics. The most important question that arises is whether the backward reachability analysis developed for piecewise linear hybrid dynamical systems can be applied efficiently for the analysis of nonlinear hybrid systems. Piecewise linear systems can be used to approximate the nonlinear dynamics. However, in order to obtain good approximations we may need to use a large number of subsystems and therefore the corresponding analysis and synthesis algorithms will be in general computationally inefficient. The extension of the analysis and synthesis techniques based on discrete abstractions of the continuous dynamics for nonlinear hybrid systems is a very important research direction.

References

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

[2] R. Alur, T. Henzinger, G. Lafferriere, and G. Pappas. Discrete abstractions of hybrid systems. *Proceedings of IEEE*, 88(7):971–984, July 2000.

[3] P. Antsaklis, editor. *Proceedings of the IEEE, Special Issue on Hybrid Systems: Theory and Applications*, July 2000.

[4] P. Antsaklis, X. Koutsoukos, and J. Zaytoon. On hybrid control of complex systems: A survey. *European Journal of Automation*, 32(9-10):1023–1045, 1998.

[5] E. Asarin, O. Bournez, T. Dang, and O. Maler. Approximate reachability analysis of piecewise-linear dynamical systems. In N. Lynch and B. Krogh, editors, *Hybrid Systems—Computation and Control*, volume 1790 of *Lecture Notes in Computer Science*, pages 20–31. Springer-Verlag, 2000.

[6] 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.

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

[8] A. Bemporad and M. Morari. Verification of hybrid systems via mathematical programming. In F. Vaandrager and J. van Schuppen, editors, *HSCC 99: Hybrid Systems—Computation and Control*, volume 1569 of *Lecture Notes in Computer Science*, pages 31–45. Springer-Verlag, 1999.

[9] A. Bemporad, F. Torrisi, and M. Morari. 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.

[10] O. Bournez, O. Maler, and A. Pnueli. Orthogonal polyhedra: Representation and computation. In *HSCC 99: Hybrid Systems—Computation and Control*, volume 1569 of *Lecture Notes in Computer Science*, pages 46–60. Springer-Verlag, 1999.

[11] C. Chang. *Model Theory*. Elsevier, 1990.

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

[13] R. Duffin. On Fourier's analysis of linear inequality systems. *Mathematical Programming Study I*, pages 71–95, 1974.

[14] T. Henzinger. Hybrid automata with finite bisimulations. In Z. Fülöp and G. Gécgeg, editors, *ICALP'95: Automata, Languages, and Programming*. Springer-Verlag, 1995.

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

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

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

[18] X. Koutsoukos and P. Antsaklis. 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.

[19] X. Koutsoukos and P. Antsaklis. 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.

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

[21] G. Lafferriere, G. Pappas, and S. Sastry. Hybrid systems with finite bisimulations. 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 186–203. Springer, 1999.

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

[23] M. Lemmon. On the existence of solutions to controlled hybrid automata. In N. Lynch and B. Krogh, editors, *Hybrid Systems—Computation and Control*, volume 1790 of *Lecture Notes in Computer Science*, pages 229–242. Springer-Verlag, 2000.

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

[25] J. Lygeros, C. Tomlin, and S. Sastry. Controllers for reachability specifications for hybrid systems. *Automatica*, 35(3):349–370, 1999.

[26] N. Lynch, R. Segala, F. Vaandrager, and H.Weinberg. Hybrid I/O automata. 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 496–510. Springer, 1996.

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

[28] A. Morse, editor. *Control using logic-based switching*, volume 222 of *Lecture Notes in Control and Information Sciences*. Springer, 1997.

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

[30] S. Nash and A. Sofer. *Linear and Nonlinear Programming*. McGraw-Hill, 1996.

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

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

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

[34] M. Sain. *Introduction to Algebraic System Theory*. Academic Press, 1981.

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

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

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

[38] J. Stiver. *Analysis and design of hybrid control systems*. PhD thesis, Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN, 1995.

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

[40] J. Stiver, X. Koutsoukos, and P. Antsaklis. An invariant based approach to the design of hybrid control systems. Technical Report ISIS-2000-001, ISIS Group at the University of Notre Dame, February 2000. To appear in the International Journal of Robust and Nonlinear Control.

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

[42] R. Vidal, S. Schaffert, J. Lygeros, and S. Sastry. 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.

[43] H. Williams. Fourier's method of linear programming and its dual. *American Mathematical Monthly*, 93:681–695, 1986.

[44] H. Wong-Toi. The synthesis of controllers for linear hybrid automata. In *Proceedings of the 36th IEEE Conference on Decision and Control*, pages 4607–4612, San Diego, CA, December 1997.