X.D. Koutsoukos and P.J. Antsaklis, "Hierarchical Control of Piecewise Linear Hybrid Dynamical Systems Based on Discrete Abstractions," ISIS Technical Report ISIS-2001-001, February 2001. A revised version titled "Safety and Reachability of Piecewise Linear Hybrid Dynamical Systems Based on Discrete Abstractions," appears in the Journal of Discrete Event Dynamic Systems (DEDS), pp 203-243, Vol. 13, No. 3, July 2003.

Discrete Event Dynamic Systems is published quarterly.

Discrete Event Dynamic Systems: Theory and Applications is abstracted and/or indexed in ***Current Contents: Engineering, Computing, and Technology, The ISI Alerting Services, COMPUMATH Citation Index, Science Citation Index Expanded, INPEC Information Services, Mathematical Reviews database.***

Subscription Rates

The Institutional subscription price for 2003: volumes 13 (4 quarterly issues), including postage and handling is:

Print OR Electronic Version: EURO 497.00/US $498.00 per year

For more information about our 2003 journal access and pricing policies, visit www.kluweronline.com

Ordering Information/Sample Copies

Subscription orders and requests for sample copies should be sent to:

Kluwer Academic Publishers
101 Philip Drive
Assinippi Park
Norwell, MA 02061 USA
phone: (781) 871-6600
fax: (781) 871-6528
e-mail: kluwer@wkap.com

or

Kluwer Academic Publishers
P.O. Box 322
3300 AH Dordrecht
The Netherlands

or to any subscription agent.

Periodicals postage paid at Rahway, NJ USPS: 017-351

U.S. mailing agent: Mercury Airfreight International Ltd.
365 Blair Road
Avenel, NJ 07001 U.S.A.

Published by Kluwer Academic Publishers

Postmaster: Please send all address corrections to: *Discrete Event Dynamic Systems*, c/o Mercury Airfreight International Ltd., 365 Blair Road, Avenel, NJ 07001, U.S.A.

ISSN: 0924-6703

Printed on acid-free paper

Discrete Event Dynamic Systems: Theory and Applications, 13, 203–243, 2003
© 2003 Kluwer Academic Publishers. Manufactured in The Netherlands.

Safety and Reachability of Piecewise Linear Hybrid Dynamical Systems Based on Discrete Abstractions

XENOFON D. KOUTSOUKOS xenofon.koutsoukos@vanderbilt.edu
Department of Electrical Engineering and Computer Science, Vanderbilt University, Nashville, TN 37235 USA

PANOS J. ANTSAKLIS antsaklis.1@nd.edu
Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN 46556, USA

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

In this paper, a systematic methodology for analysis of piecewise linear hybrid systems based on discrete abstractions of the continuous dynamics is presented. Our work is motivated by the need to address challenging problems in the control and coordination of modern complex engineering 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. The introduced model is general enough to describe important engineering applications, but simple enough to facilitate the development of analysis, and 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.

Analysis and synthesis methodologies based on discrete abstractions have been studied extensively in the hybrid system literature: see, for example, Alur et al. (2000) and Koutsoukos et al. (2000). 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 Tomlin et al. (1998) and Lygeros et al. (1999). 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 (Morse, 1996, 1997) 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, Leenaerts and Van Bokhoven (1998) 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 Sontag (1981, 1982, 1996). The main problems studied in this framework are stability, controllability, and input-output regulation. Piecewise linear dynamical systems have been considered also in Bournez et al. (1999) and Asarin et al. (2000a, 2000b). 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 Bemporad and Morari (1999a, 1999b) and Bemporad et al. (2000). 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 Johansson (1999) to develop computational algorithms for the analysis of nonlinear and uncertain dynamical systems.

A great amount of research work has already been done in the hybrid systems area during the past decade; see, for example, Antsaklis (2000) and the references therein. A survey of different models and methodologies can be found in Antsaklis et al. (1998). The approach presented in this paper is directly related to supervisory control framework for hybrid systems (Stiver, 1995; Stiver et al., 1996b; Koutsoukos et al., 2000). Similar approaches based on approximations of the continuous dynamics by a discrete event system have also been proposed in Nerode and Kohn (1993), Raisch and O'Young (1998), Cury et al. (1998), and Lunze et al. (1999). 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 consider a plant that contain discrete dynamics and both discrete and continuous inputs as well as discrete and continuous disturbances.

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 Alur et al. (1995). Formalisms for input/output hybrid automata have been also proposed in Lynch et al. (1996), Tomlin et al. (1998), and Lemmon (2000). A related approach to the work presented in this paper uses bisimulations to study the decidability of verification algorithms (Henzinger, 1995; Lafferriere et al., 1999; Alur et al., 2000). 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. However, the use of bisimulations in practical control systems is limited by the requirements for very simple continuous dynamics (Lafferriere et al., 1999). The related notion of dynamical consistency for hierarchical control systems has been studied in Caines and Wei (1998). The use of dynamical consistency aims at the computation of abstractions that preserve the controllability properties of hybrid control systems. A lattice of hierarchical partitions is defined in Caines and Wei (1998) and used to investigate dynamical consistency. However, no constructive algorithms for the computation of the partitions are given. Computational methods for reachability analysis of hybrid systems have been also presented in Chutinan and Krogh (1999a, 1999b) where the continuous flow of the hybrid system with arbitrary dynamics is approximated using polygonal flow pipes. Finite-state approximations are then used for the verification of the hybrid system properties. In our paper, reachability analysis of discrete-time piecewise linear hybrid systems is carried out without approximations using Fourier–Motzkin elimination and linear programming techniques. The initial partition is refined based on the existing control resources and disturbances. The refinement terminates when it is guaranteed that the control specifications can be satisfied enabling the design of control algorithms.

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. 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 Koutsoukos (2000); early results have been reported in Koutsoukos and Antsaklis (1999a, 1999b, 2000).

This paper is organized as follows. In Section 2, we present the modeling framework for piecewise linear hybrid dynamical systems. In Section 3, we use an algebraic system theory framework to describe our motivation for using discrete abstractions for the analysis of hybrid systems. In Section 4, 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 5, we study the safety problem for piecewise linear hybrid systems. In Section 6, we study the reachability problem and we formulate conditions that guarantee reachability between piecewise linear regions. Finally, concluding remarks are presented in Section 7.

2. Piecewise Linear Hybrid Dynamical Systems

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

A piecewise-linear (PL) subset (Sontag, 1982) 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 $X \times Y$). A PL map is defined similarly. 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 $f \mid X_i$ 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 $x_1 E_\pi x_2$ 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 stateset X into the equivalence classes of E_π and will be called generator.

In this paper, 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

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

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

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

Then, 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 $\Re^n$.

Let $X \subseteq \Re^n$ denote the continuous state space, Q the finite set of discrete states or modes of the system, $U \subset \Re^m$ the continuous input space, Σ the set of input events, and Y the output set of the hybrid system. Often, it is desirable to distinguish between controlled and uncontrolled inputs, and we may include both a space of continuous inputs $U \subset \Re^m$ and a space of continuous disturbances $D \subset \Re^p$. 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. Note that this definition is different than the definition of supervisory control (Ramadge, 1990) where uncontrollable events are events that can be disabled by the controller. The output set can also contain a discrete and a continuous part. 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.

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

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

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

$$y(t) = g(q(t), x(t)) \tag{3}$$

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

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

The interaction between the discrete and continuous components of a piecewise linear hybrid system is formally defined as follows. For each discrete mode, we assign a region of the state space using the mapping $\mathrm{inv}: Q \to 2^{X/E_\pi}$. The continuous state may evolve according to the difference equation determined by the discrete state q only if $x(t) \in \mathrm{inv}(q)$. The regions $\mathrm{inv}(q)$ 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 $\mathrm{act} : X/E_\pi \to 2^Q$. From the definition of the invariants and the active mode sets, it follows that for each discrete state $q \in Q$ and for each region of the primary partition $P \in X/E_\pi$ we have $P \in \mathrm{inv}(q) \Leftrightarrow q \in \mathrm{act}(P)$.

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

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.

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. 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 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 B of the system by applying the heat input at a different point A. The temperature at point B is also affected by the temperature at a point C of the environment.

The temperature control system has two modes q_0 and q_1 that correspond to the furnace being on and off respectively. When the furnace is on, the system can described by the electrical circuit shown in Figure 1. Let x_1 and x_2 denote the voltages (temperatures) across the capacitors C_1 and C_2 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. For the mode q_1, the system is governed by the state-space equation (using Kirchhoff's laws)

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -\frac{1}{R_1C_1} & \frac{1}{R_1C_1} \\ \frac{1}{R_1C_2} & -\frac{1}{R_{12}C_2} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} \frac{1}{C_1} \\ 0 \end{bmatrix} u + \begin{bmatrix} 0 \\ \frac{1}{R_2C_2} \end{bmatrix} d$$

where $R_{12} = R_1R_2/R_1 + R_2$.

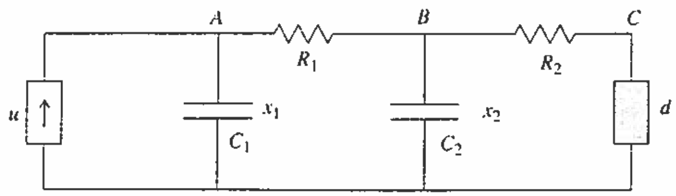

Figure 1. Electric circuit describing the case when the furnace is on.

When the furnace is turned off, the temperature is decreasing and the behavior of the system can be 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. The state-space representation of the system for the mode q_0 takes the form

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} = \begin{bmatrix} -\frac{1}{R_3C_3} & \frac{1}{R_3C_3} \\ \frac{1}{R_3C_4} & -\frac{1}{R_{34}C_4}\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} + \begin{bmatrix}0\\ \frac{1}{R_4C_4}\end{bmatrix} d$$

where $R_{34} = R_3R_4/R_3 + R_4$.

The voltages (temperatures) x_1 and x_2 can be affected either by the continuous control input $u \in U$ or by switching on or off the furnace using the control input events σ_{on} and σ_{off}. It is assumed that a relief switch is used to protect point A from overheating, so that the furnace is switched off automatically whenever the temperature x_1 exceeds a prescribed level ub. The control objective for the system is to maintain the temperature x_2 between appropriate levels described by the interval $[lt, ht]$.

A partition of the continuous state space is obtained by considering the hyperplanes $h_1 = x_1 - ub, h_2 = x_2 - lt, h_3 = x_2 - ht$, 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. The discrete part of the system is described by the automaton of Figure 3. The system switches between modes q_0 and q_1 upon receiving the control input events σ_{on} and σ_{off}. Moreover, a plant event issues when $x_1 > ub$ will switch the system to the off mode.

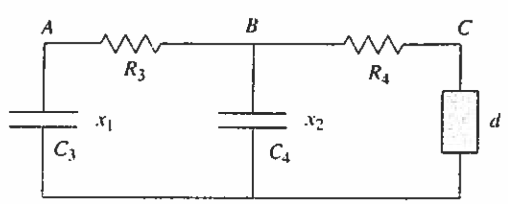

Figure 2. Electric circuit describing the case when the furnace is off.

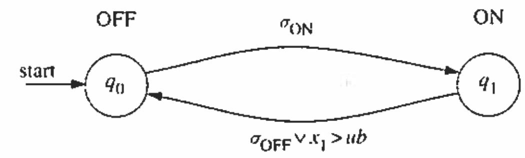

Figure 3. Temperature control system.

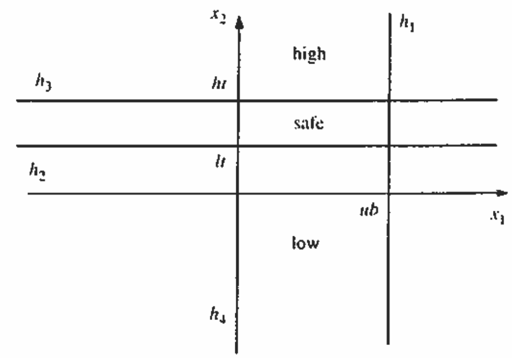

Figure 4. Partition for the temperature control system.

The temperature control system example is used to illustrate the partition refinement methodolody for safety specifications in Subsection 5.2 and for reachability specifications in Subsection 6.1. A case study of hybrid controller synthesis for a related heating system is presented in Balluchi et al. (1999), where a heating system is modeled as a hybrid automaton with continuous dynamics described by first-order differential equations and controller that guarantees safety is designed using game theory. In our example, the objective is to control the temperature at of the system by applying the heat input at a different point, and therefore the system can only be described using second-order dynamics. It should be noted that the example and our approach can be generalized to an n-th-order piecewise linear hybrid system.

3. 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. The solution we propose is to take advantage of the available control inputs in order to simplify the system. 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.

3.1. *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 Sain (1981) that $\tilde{f}$ exists if and only if

$$x_1 E_\pi x_2 \Rightarrow (\pi \circ f)(x_1) = (\pi \circ f)(x_2) \tag{4}$$

$$\begin{tikzcd} X \arrow[r, "f"] \arrow[d, "\pi"'] & X \arrow[d, "\pi"] \\ X/E_\pi \arrow[r, dashed, "\tilde{f}"] & X/E_\pi \end{tikzcd}$$

Figure 5. Induced dynamical systems.

(where $\circ$ denotes function composition) and moreover, if (4) 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 (Sain, 1981).

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. 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.

In general, piecewise linear hybrid dynamical systems cannot be induced in finite quotient spaces by preserving the reachability properties (Lafferriere et al., 1999). However, there are some cases when a mapping π and the induced system $\tilde{f}$ can be computed. A special case arises when the mapping π is defined using the natural invariants of the continuous dynamics (Stiver et al., 1996a; Broucke, 1999; Stiver et al., 2001). However, it is very difficult to compute such partitions, and moreover, the control specifications are not necessarily defined using the invariant sets of the system. The solution we propose is to take advantage of the available control inputs in order to simplify the system. We want to formulate conditions on the available control inputs in order to induce piecewise linear hybrid dynamical systems in finite quotient spaces. 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, 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 Koutsoukos (2000), but it is not presented in this paper due to space limitations.

First, we describe our approach using an algebraic system theory setting. Consider the diagram shown in Figure 6. 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

$$\begin{tikzcd} X \times U \arrow[r, "f"] \arrow[d, "\pi_E = \pi_{E_X} \times \pi_{E_U}"'] & X \arrow[d, "\pi_{E_X}"] \\ X/E_X \times U/E_U \arrow[r, dashed, "\tilde{f}"] & X/E_X \end{tikzcd}$$

Figure 6. Function diagram including control inputs.

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 our two-level approach 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

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

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 step. In order to do that, we consider an upper bound $N \in \mathbb{N}$ 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 (Sontag, 1996). We introduce the following notation.

$$[t_1, t_2] = \{t_1, t_1 + 1, \ldots, t_2 - 1, t_2\}, \quad t_1 \leq t_2,$$

$$u^*[t_0, t_1] = \{u(t_0), \ldots, u(t_1)\}$$

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

$$(x_1, u_1^*[t_0, t_1])\, E\, (x_2, u_2^*[t_0, t_2]) \Rightarrow \exists u_1^*[t_0, t_1], u_2^*[t_0, t_2] \in U^N$$

$$(\pi_{E_X} \circ \phi)(x_1, u_1^*[t_0, t_1]) = (\pi_{E_X} \circ \phi)(x_2, u_2^*[t_0, t_2])$$

$$\begin{tikzcd} X \times U^N \arrow[r, "\phi"] \arrow[d, "\pi_E"'] & X \arrow[d, "\pi_{E_X}"] \\ X/E_X \times U^N/E_{U^N} \arrow[r, dashed, "\tilde{\phi}"] & X/E_X \end{tikzcd}$$

Figure 7. Function diagram including control input sequences.

Our objective is to compute a partition of the state space so that the diagram shown in Figure 7 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 7 is only required with respect to the equivalence classes that are formed from the control specifications.

3.2. *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). Note that a lattice of partitions has been used to study dynamical consistency of hybrid control systems in Caines and Wei (1998). In this paper, we only consider polyhedral partitions and we present a constructive methodology for partition refinement.

In the following, we present some basic notions from algebraic system theory (Sain, 1981) 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 Sain, 1981). The lattice $(E(X), \leq, \wedge, \vee)$ is called the equivalence lattice.

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 $\mathcal{H}_1 = \{h_i\}_{i=1,\dots,d_1}$ and $\mathcal{H}_2 = \{h'_i\}_{i=1,\dots,d_2}$ respectively. The meet of E_1 and E_2 is defined as the set theoretic intersection $E = (E_1, E_2) = E_1 \cap E_2$. 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

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

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 $_i(E_i)$ 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.

4. 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.

4.1. 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 $R = (M, P)$ where $M \subseteq Q$ is a set of modes and $P \subset \Re^n$ is a piecewise linear set satisfying the following property. For every $x \in P$ there exists $q \in M$ such that $q \in \operatorname{act}(\pi(x))$. This condition guarantees that for every state in the region R there is a possible evolution of the system. Given the region $R = (M, P)$, we define the predecessor operator pre : $2^{Q \times X} \to 2^{Q \times X}$ to compute the set of states for which there exists a control input so that the state will be driven in R for every disturbance. The action of the operator is described by

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

The set pre(R) is piecewise linear and can be always represented using only linear equalities and inequalities. Such a description is based on the fact that piecewise-linear algebra admits elimination of quantifiers (Sontag, 1982) which means that any PL set defined using quantifiers can be also defined using only propositional connectives.

4.2. 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. Our results are based on combinations of three different mathematical tools. Fourier–Motzkin elimination (Motzkin, 1952) for computing appropriate projections, linear programming techniques (Nash and Sofer, 1996) for eliminating redundant constraints, and equivalences from predicate logic (Nerode and Shore, 1993) to combine the constraints.

Consider the region $R = (M, P)$. A PL set is not necessarily polyhedral. However, every PL set P can be written as a finite union of polyhedral sets $P = \bigcup_{i=1}^{p} P_i$, for example, by writing the linear constraints in disjunctive normal form. In order to show that there exists a constructive algorithm for elimination of quantifiers, we have essentially to consider only the logical formula

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

Algorithms for elimination of quantifiers for more complicated logical formulas can then be derived using logical equivalences (Chang, 1990). 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. Since any PL set can be written as the finite union of polyhedral sets, 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. In order to simplify the notation, in the remaining of this subsection we assume that the discrete state q is fixed and we consider the restriction of the predecessor operator to the continuous state space $\mathrm{pre}_c : 2^X \to 2^X$.

4.3. *Continuous Control Inputs*

Consider the system $x(t+1) = Ax(t) + Bu(t)$ 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 $U = \{u \in \Re^m \mid Fu \leq v\}$, $F \in \Re^{\mu \times m}$, $v \in \Re^{\mu}$. Consider the polyhedral set $P \subseteq \Re^n$ given by $P = \{x \in \Re^n \mid Gx \leq w\}$, $G \in \Re^{\nu \times n}$, $w \in \Re^{\nu}$. Our objective is to present a systematic methodology to compute the predecessor operator set

$$\mathrm{pre}_c(P) = \{x \in \Re^n \mid \exists u \in U, Ax + Bu \in P\}$$

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

PROPOSITION 2 *The set* $\mathrm{pre}_c(P)$ *is given by* $\mathrm{pre}_c(P) = \mathrm{Pr}(W)$ *where* $W \subseteq X \times U$ *is defined as* $W = \{(x,u) \mid (GAx + GBu \leq w) \wedge (Fu \leq v)\}$.

Proof: By direct substitution, we have that $\mathrm{pre}_c(P) = \{x \mid \exists u \in U, GAx + GBu \leq w\}$. Then, we have that if $x \in \mathrm{Pr}(W)$, there exits $u \in U$ such that $(x,u) \in W$, and therefore $x \in \mathrm{pre}_c(P)$. Conversely, if $x \in \mathrm{pre}_c(P)$, then by definition of the predecessor operator there exists control input $u \in U$ such that $(x,u) \in W$, which implies that $x \in \mathrm{Pr}(W)$. Therefore, we have shown that $\mathrm{pre}_c(P) = \mathrm{Pr}(W)$. ∎

The projection of the set W into the continuous state space $X = \Re^n$ can be computed using the Fourier-Motzkin elimination method (Motzkin, 1952; Duffin, 1974; Williams, 1986). We project the polyhedron $W \subset \Re^n \times \Re^m$ 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

$$x_1 + x_2 + u \leq 1 \tag{5}$$

$$2x_1 + x_2 + u \leq 1 \tag{6}$$

$$u \leq 1 \tag{7}$$

$$-u \leq -0.5 \tag{8}$$

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 (5) and (8) can be written as

$$0.5 \leq u \leq 1 - x_1 - x_2 \tag{9}$$

Therefore, we have that

$$0.5 \leq 1 - x_1 - x_2 \tag{10}$$

which can be written as

$$x_1 + x_2 \leq 0.5 \tag{11}$$

Therefore, if there is a solution to the inequalities (5) and (8), there must be a solution to the derived inequality (11). Conversely, if there is a solution to (11), then by writing the inequality in the form (9), it follows that there exists u such that the initial inequalities are satisfied. Note that the inequality (11) can be easily derived by adding (5) and (8) (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.

$$x_1 + x_2 \leq 0.5$$
$$2x_1 + x_2 \leq 0.5$$
$$0 \leq 0.5$$

Note that the constraint $0 \leq 0.5$ is redundant.

A piecewise linear set, however, is not necessarily polyhedral, but it can be written as the union of polyhedral sets. 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 1 *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:

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

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

4.3.1. *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 $x(t+1) = Ax(t) + Bu(t) + Ed(t)$ 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:

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

In this case, the predecessor operator takes the form

$$\text{pre}_c(P) = \{x \in X \mid \exists u \in U, \quad \forall d \in D, \quad Ax + Bu + Ed \in P\} \tag{12}$$

Consider the following linear programming problems for $i = 1, \ldots, \nu$:

$$\begin{aligned}
\min \quad & -g_i^T Ed \\
\text{s.t.} \quad & d \in D
\end{aligned}$$

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 $\text{pre}_c(P)$ is given by $\text{pre}_c(P) = \text{Pr}(V)$ where $V \subseteq X \times U$ is defined as $V = \{(x, u) \mid \bigwedge_{i=1,\ldots,\nu} g_i^T Ax + g_i^T Bu \leq w_i - g_i^T Ed_i^*\}$.*

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

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

Therefore, there exists $u \in U$ such that $(x, u) \in V$, which implies that $x \in \text{Pr}(V)$.

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

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

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

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

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

$$\forall d \in D, Ax + Bu + Ed \in P_1 \cup P_2 \longleftrightarrow \neg(\exists d \in D, Ax + Bu + Ed \in P_1^c \cap P_2^c)$$

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

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

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 $\mathrm{pre}_c(P)$ 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

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

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 5.

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 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 Koutsoukos and Antsaklis (2000).

4.4. *Algorithms for Backward Reachability Analysis*

Consider a PLHDS described by the equations (1)–(3) and a region $R = (M, P)$. 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 as defined in equation (12). The following algorithm computes all the states of the hybrid system that can be driven to R in one time-step.

Algorithm for the computation of pre(R)

INPUT: $R = (M, P), S = \emptyset, T = \emptyset$;
for $i = 1, \ldots, |\pi|$
  $Q_i = P \cap P_i$;
  if $Q_i \neq \emptyset$
    for $q \in M \cap \mathrm{act}(P_i)$
      $S = S \cup \mathrm{pre}_{c,q}(Q_i)$;
      $T = T \cup \{q\}$;
    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. 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 and we consider only the active mode set for each region. Note that by construction, for every $x \in S$ there exists $q \in S$ such that $q \in (\pi(x))$ and therefore, the set $\mathrm{pre}(R)$ is a region.

We have shown that the set $\mathrm{pre}(R)$ is piecewise linear since it can be 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 $\mathrm{pre}(R)$ to get $\mathrm{pre}\,(\mathrm{pre}(R))$. Following the same procedure, we define successive applications of the predecessor operator as

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

For a given region R, we define the coreachable set $\mathrm{CR}(R)$ 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 $CR(R) = \mathrm{pre}^*(R)$ where * denotes the fixed point of the predecessor operator. It should be noted that the algorithm for the computation of the coreachable set for a region R is semi-decidable. The procedure produces the correct answer if it terminates, but its termination is not guaranteed. Infinite time problems for piecewise linear systems are, in general, undecidable (Sontag, 1996). In Section 6, we present a grid-based approximation technique that can be used to formulate a termination condition for the successive application of the predecessor operator.

For finite time problems, backward reachability algorithms for piecewise linear hybrid systems are NP-complete (Sontag, 1996). 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 5 that we can formulate conditions that guarantee that a piecewise linear region is safe by considering only one iteration.

5. 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 2: *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.

5.1. 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 (Lafferriere et al., 1999). 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.

5.1.1. 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 $\pi(x(t+1)) \in X/E_\pi$. 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 quasideterminism is illustrated in Figure 8. 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 in Figure 8 in order to analyze the reachability properties with respect to the safety specifications.

DEFINITION 3: *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 $(q, q'), q' \in \mathrm{act}(\pi(x(t)))$, control action $u \in U$ and disturbance $d \in D$.*

$$\begin{tikzcd} X \arrow[r, "f"] \arrow[d, "\pi_F"'] & X \arrow[d, "\pi"] \\ X/E_{\pi_F} \arrow[r, "\tilde{f}"] & X/E_\pi \end{tikzcd}$$

Figure 8. Quasideterminism and the partitions of the state space.

In Section 3, we showed that given a piecewise linear region $R \subset Q \times X$, the set $\text{pre}(R)$ of all the states that can be driven to R 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 $\text{pre}(R)$ and the partition $\pi' \in E_P(X)$ defined by those as $\pi'(x) = [\hat{h}'_1(x), \ldots, \hat{h}'_\ell(x)]^T$ where

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

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* $\text{pre}: 2^{Q\times X} \to 2^{Q\times X}$ *to the regions of the initial partition defined by $E_{\pi'}$. Then the piecewise linear hybrid dynamical system with final partition defined by $E_{\pi_F} = (E_\pi, E_{\pi'})$ 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 $x(t+1) = A_q x(t) + B_q u(t) + E_q d(t) \in P_i$ if and only if $Z_j \subseteq \text{pre}_c(P_i)$. Therefore, for each $(q, q'), q' \in (\pi(x(t)))$, 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 theorem 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 theset 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 $x(t+1)$ in an equivalence class of X/E_π can be uniquely determined.

5.2. *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 $R \subseteq \text{pre}(R)$.*

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 $(q(t), x(t)) \in R$ for every t. By definition, the set $\text{pre}(R)$ 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 $(q, x) \in R$ we have that $(q, x) \in \text{pre}(R)$. ■

In the following, we present a constructive algorithm which is used to test the condition $R \subseteq \text{pre}(R)$. Let $R|X$ and $\text{pre}(R)|X$ be the projection of R and $\text{pre}(R)$ into the continuous state space X. Similarly $R|Q$ and $\text{pre}(R)|Q$ for the discrete state space Q. In order to show that $R \subseteq \text{pre}(R)$, we need to test whether $R|Q \subseteq \text{pre}(R)|Q$ and $R|X \subseteq \text{pre}(R)|X$. Since, the sets $R|Q$ and $\text{pre}(R)|Q$ 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 $\text{pre}(R)$. The sets $R|X$ and $\text{pre}(R)|X$ 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 and we test the feasibility of finite set of linear programming problems.

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 $\text{pre}(R)|X$ can be written as

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

and

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

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

$$\begin{aligned} &\min \quad x \\ &\text{s.t.} \quad x \in C_{ij} \end{aligned}$$

We have that $P_i \cap W_j = \emptyset$ 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 $P_i \cap W_j = \emptyset$ for every $i = 1, \ldots, |P|$ and $j = 1, \ldots, |Q|$.

Table 1. Parameters for the temperature control system

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

Example—Temperature Control System: In the following, we use the temperature control system presented in Section 2 to illustrate how we can formulate the safety conditions. 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 $x(t+1) = A_1 x(t) + B_1 u(t) + E_1 d(t)$ where

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

and $u(t) \in U_1, d(t) \in D_1$. Similarly, for the discrete state q_0 (furnace off), we have $x(t+1) = A_0 x(t) + B_0 u(t) + E_0 d(t)$ where

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

and $u(t) \in U_0, d(t) \in D_0$. The partition of the state space is obtained by considering the following hyperplanes $h_1(x) = x_1 - ub$, $ub = 20$; $h_2(x) = x_2 - ht$, $ht = 5$; $h_3(x) = x_2 - lt$, $lt = 0$; and $h_4(x) = x_1$, and it is shown in Figure 9. It is assumed that the safe region is described by the set $R = \{(q_0, q_1), P\}$ where P is given by

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

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

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

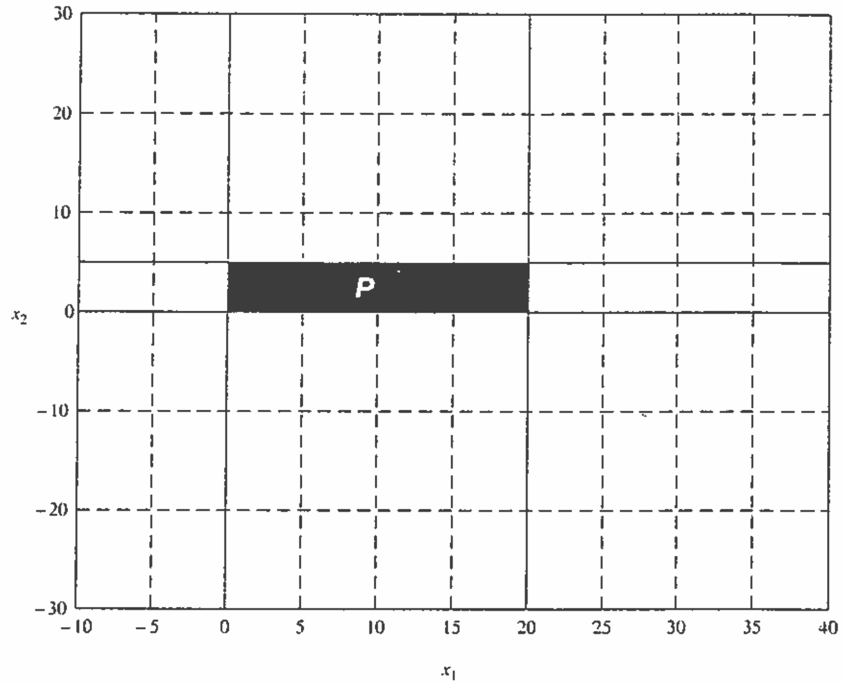

Figure 9. Primary partition for the temperature control system.

and

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

First, we compute the set

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

Note that if the system is at mode q_0, the input is $u = 0$. Using Proposition 3, we consider the following set of linear inequalities:

$$GA_0 x \le w - GE_0 d \tag{13}$$

We solve the linear programming problems

$$\begin{aligned} \min \quad & -g_i^T E_0 d \\ \text{s.t.} \quad & d \in D_0 \end{aligned}$$

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 (13) we get

$$\text{pre}_{c,q_0}(P) = \left\{ x \in \Re^n \left| \begin{bmatrix} 0.8259 & 0.1354 \\ 0.0677 & 0.5551 \\ -0.0677 & -0.5551 \\ -0.8259 & -0.1354 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \leq \begin{bmatrix} 20 \\ 5 \\ -0.3772 \\ -0.0387 \end{bmatrix} \right. \right\} \tag{14}$$

Next, we compute the set

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

We consider the following set of linear inequalities:

$$\begin{aligned} GA_1 x + GB_1 u &\leq w - GE_1 d \\ u &\leq 1 \\ -u &\leq -0.5 \end{aligned}$$

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

$$\begin{aligned} \min \quad & -g_i^T E_1 d \\ \text{s.t.} \quad & d \in D_1 \end{aligned}$$

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

$$\text{pre}_{c,q_1}(P) = \left\{ x \in \Re^2 \left| \begin{bmatrix} 0.6634 & 0.1997 \\ 0.1997 & 0.2641 \\ -0.1997 & -0.2641 \\ -0.6634 & -0.1997 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \leq \begin{bmatrix} 19.4580 \\ 4.3953 \\ 0.6847 \\ 4.0507 \end{bmatrix} \right. \right\} \tag{15}$$

The sets $\text{pre}_{c,q_0}(P)$ and $\text{pre}_{c,q_1}(P)$ are shown in Figure 10. The set pre(R) is computed using the algorithm presented in Subsection 4.4 as

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

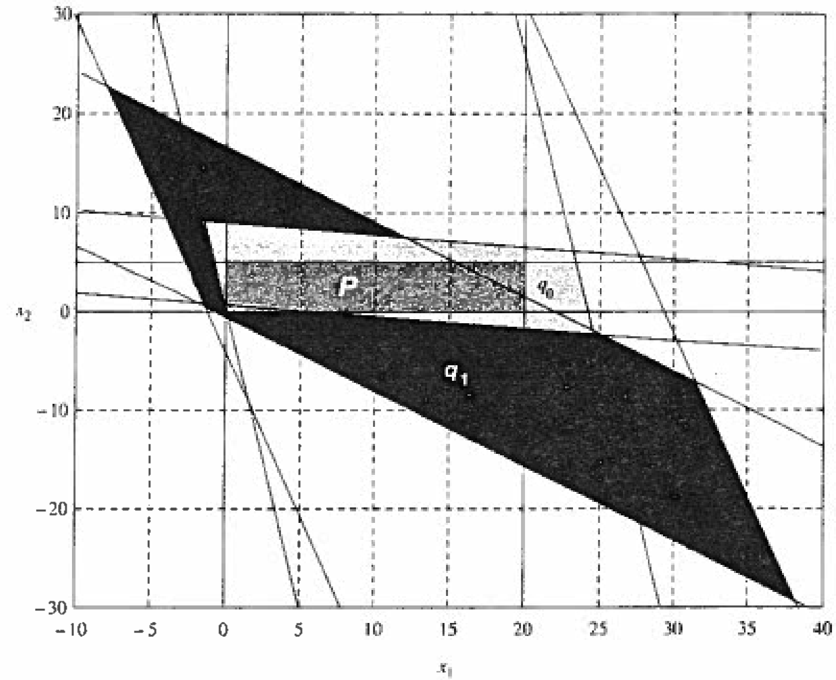

Figure 10. Final partition for the temperature control system.

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

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

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 (14) and (15) respectively. We define the set $W = [\text{pre}(R|X)]^c$. Using DeMorgan's laws, the set W can be represented by

$$\bigcup_{\substack{i=1,2,3,4, \\ j=1,2,3,4}} (\neg\phi_{0i} \wedge \neg\phi_{1j})$$

Therefore, the set W can be written as $W = \bigcup_{i,j} W_{ij}$. Each set W_{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:

$$\begin{aligned} &\min \quad x \\ &\text{s.t.} \quad x \in P \cap W_{ij} \end{aligned}$$

For the temperature control system, we have that $R|Q = \text{pre}(R)|Q = \{q_0, q_1\}$ and $P \cap W_{ij} = \emptyset$ 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 10. Note that for every continuous states x there exists at least one discrete mode that the system can switch to guarantee safety.

5.3. *Maximal Safe Set*

In Subsection 5.2, 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 has been studied for timed systems in Asarin et al. (1995) and Maler et al. (1995) and for several classes of hybrid systems in Tomlin et al. (1998), Lygeros et al. (1999), Wong-Toi (1997), and Berardi et al. (2000). Here, we present how the algorithm presented in Tomlin et al. (1998) can be applied to PLHDS.

Algorithm for the computation of maximal safe set

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

The maximal safe set is computed as a fixed point of the iterative procedure described above. At the k-th 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 Tomlin et al. (1998) and Lygeros et al. (1999), 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 3. 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 5.2 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 Vidal et al. (2000).

6. 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), stem 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 appropriately constrained to guarantee safe operation.

DEFINITION 4: *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.

6.1. 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 $N \in \mathbb{N}$. Consider a PLHDS described by the equations (1)–(3) and assume that the initial condition is $(q(t_0), x(t_0)) \in R_1$.

DEFINITION 5 *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 $k \in \mathbb{N}, 0 < k \leq N$ so that $(q(t), x(t)) \in R_1$ for $t_0 \leq t < t_0 + kt$ and $(q(t_0 + kt), x(t_0 + kt)) \in R_2$.*

We define the coreachable set $CR_{R_1}^N(R_2)$ 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 : $2^{Q \times X} \to 2^{Q \times X}$ can be used to compute the set $CR_{R_1}^N(R_2)$ using the following algorithm.

Algorithm for the computation of $CR_{R_1}^N(R_2)$

```
R^0 = R_2;
CR^N_{R_1}(R_2) = ∅;
k = 0;
for k = 1, ..., N
  R^{k+1} = pre(R^k) ∩ R_1;
  if ¬(R^{k+1} ⊆ R^k)
    CR^N_{R_1}(R_2) = CR^N_{R_1}(R_2) ∪ R^{k+1};
  else
    exit
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 3. 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 (1)–(2) and the regions $R_1 = (Q_1, P_1)$ and $R_2 = (Q_2, P_2)$. Then, the setset $CR_{R_1}^N(R_2)$ is piecewise linear and the region R_2 is directly N-reachable from R_1 if and only if $R_1 \subseteq CR_{R_1}^N(R_2)$.*

Proof: The set set $CR_{R_1}^N(R_2)$ 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 $R_1 \subseteq CR_{R_1}^N(R_2)$ 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 $CR_{R_1}^N(R_2)$ is piecewise linear, the reachability problem between R_1 and R_2 can be solvedusing linear programming techniques similarly to the safety conditions (see Subsection 5.2). 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 Section 2. Consider the regions $R_1 = (\{q_0, q_1\}, P_1)$ and $R_2 = (\{q_0, q_1\}, P_2)$ where

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

and

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

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

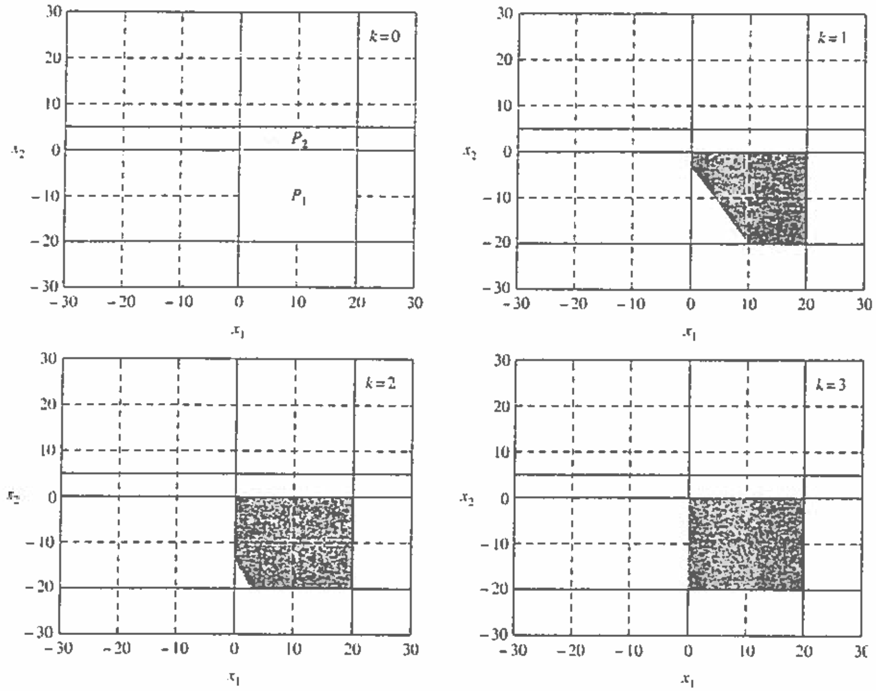

Figure 11. Coreachable set for the temperature control system.

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 6.1. The coreachable set of states for three iterations of the algorithm is shown in Figure 11. 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.

6.2. *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

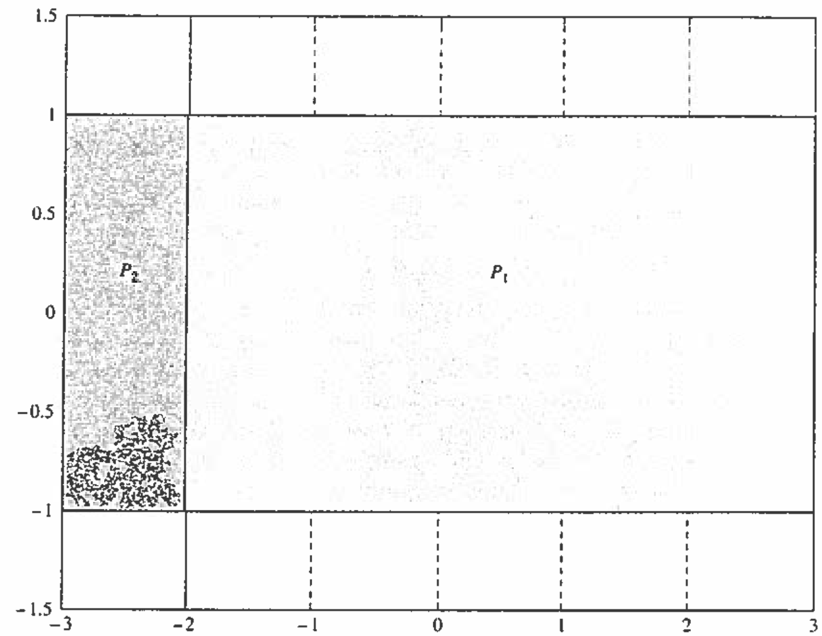

Figure 12. Primary partition for the system.

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

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

Suppose we are given the partition shown in Figure 12 described by the hyperplanes $h_1(x) = x_1 + 3, h_2(x) = x_1 + 2, h_3(x) = x_2 - 1$, and $h_4(x) = x_2 + 1$.

We consider the regions

$$P_1 = \{x \in \Re^n | (-3 \leq x_1 \leq -2) \wedge (-1 \leq x_2 \leq 1)\}$$

and

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

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

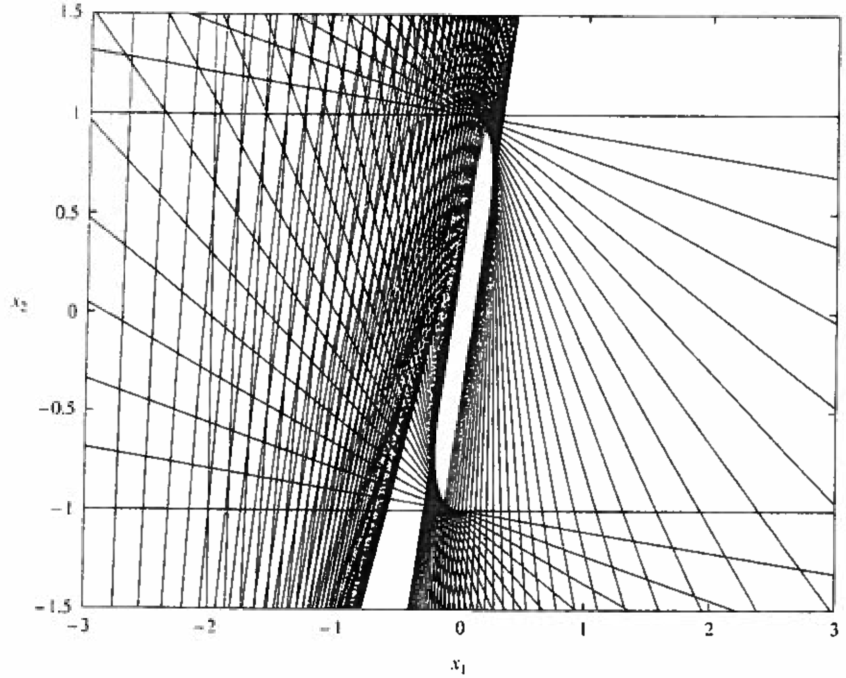

Figure 13. The backward reachability algorithm does not necessarily terminate.

states in P_1 that can be driven in P_2. At the every iteration of the reachability algorithm we have that $P^k = \mathrm{pre}_c(P)^k \cap P_1 \neq \emptyset$. Therefore, we add new states to the coreachable set of P_2 at every iteration and the algorithm will not terminate. In Figure 13 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 $\mathrm{pre}(R^k)$ is not "substantially different" than the set $\mathrm{pre}(R^{k-L})$. 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 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 bebounded. 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

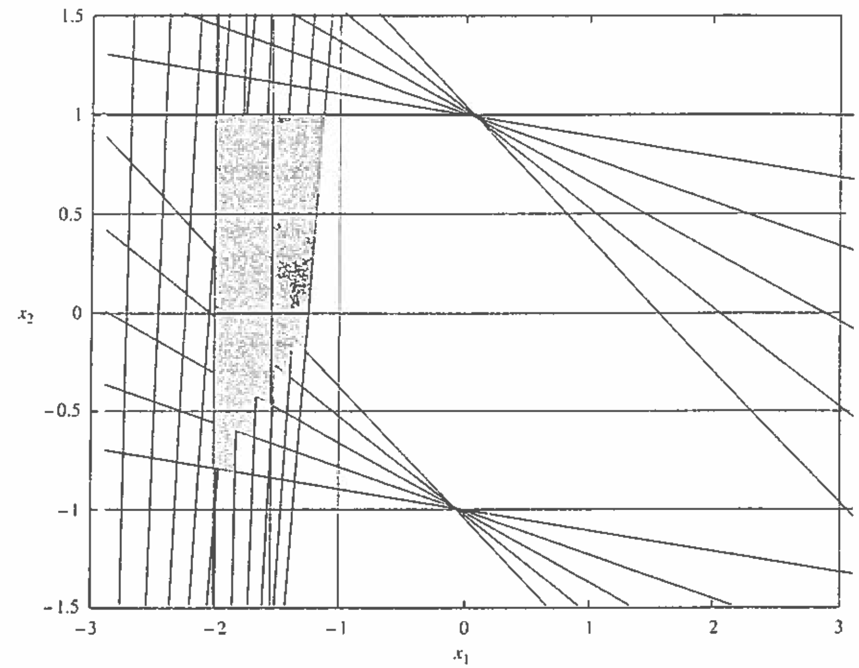

Figure 14. Grid-based approximation for reachability.

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 $\mathrm{grid}(P)$ is defined as the union of all the cells that belong to the set P. The set $\mathrm{grid}(P)$ is a conservative approximation of P since $x \in \mathrm{grid}(P)$ implies that $x \in P$.

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

$R^0 = R$;
$G^0 = \mathrm{grid}(P)$;
$T^0 = R|Q$;
while $\neg(\mathrm{pre}(R^k) \subseteq R^k)$
  $R^{k+1} = R^k \cup \mathrm{pre}(R^k)$;
  $G^{k+1} = \mathrm{grid}(P^{k+1})$;
  $T^{k+1} = R^{k+1}|Q$;

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 k-th 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.

Example: Consider the linear system $x(t + 1) = Ax(t)$ presented earlier in the section. Figure 14 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 cells were added to the coreachable set.

7. 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 interaction 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.

Acknowledgments

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

References

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

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

Antsaklis, P. (ed.) 2000. *Proceedings of the IEEE. Special Issue on Hybrid Systems: Theory and Applications*, volume 88.

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

Asarin, E., Bournez, O., Dang, T., and Maler, O. 2000a. Approximate reachability analysis of piecewise-linear dynamical systems. In N. Lynch and B. Krogh (eds), *Hybrid Systems–Computation and Control*, volume 1790 of *Lecture Notes in Computer Science*. Springer-Verlag, pp. 20–31.

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

Asarin, E., Maler, O., and Pnueli, A. 1995. Symbolic controller synthesis for discrete and timed systems. In P. Antsaklis, W. Kohn, A. Nerode, and S. Sastry (eds), *Hybrid Systems II*, volume 999 of *Lecture Notes in Computer Science*. Springer, pp. 1–20.

Balluchi, A., Benvenuti, L., Villa, T., Wong-Toi, H., and Sangiovanni-Vincentelli, A. 1999. A case study of hybrid controller synthesis of a heating system. In *Proceedings of the 5th European Control Conference, ECC99*. Germany: Carlsruhe.

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

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

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

Berardi, L., Santis, E. D., and Benedetto, M. D. 2000. Invariant sets and control synthesis for switching systems with safety specifications. In N. Lynch and B. Krogh (eds), *Hybrid Systems—Computation and Control*, volume 1790 of *Lecture Notes in Computer Science*. Springer-Verlag, pp. 59–72.

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

Broucke, M. 1999. A geometric approach to bisimulation and verification of hybrid systems. In F. Vaandrager and J. van Schuppen (eds), *HSCC 99: Hybrid Systems—Computation and Control*, volume 1569 of *Lecture Notes in Computer Science*. Springer-Verlag, pp. 61–75.

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

Chang, C. 1990. *Model Theory*. Elsevier.

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

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

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

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

Henzinger, T. 1995. Hybrid automata with finite bisimulations. In Z. Fülöp and G. Gécgeg (eds), *ICALP'95: Automata, Languages, and Programming*. Springer-Verlag.

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

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

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

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

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

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

Lafferriere, G., Pappas, G., and Sastry, S. 1999. Hybrid systems with finite bisimulations. In P. Antsaklis, W. Kohn, M. Lemmon, A. Nerode, and S. Sastry (eds), *Hybrid Systems V*, volume 1567 of *Lecture Notes in Computer Science*. Springer, pp. 186–203.

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

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

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

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

Lynch, N., Segala, R., Vaandrager, F., and Weinberg, H. 1996. Hybrid I/O automata. In R. Alur, T. A. Henzinger, and E. D. Sontag (eds), *Hybrid Systems III, Verification and Control*, volume 1066 of *Lecture Notes in Computer Science*. Springer, pp. 496–510.

Maler, O., Pnueli, A., and Sifakis, J. 1995. On the synthesis of discrete controllers for timed systems. In E. Mayr and C. Puech (eds), *STACS 95: Theoretical Aspects of Computer Science*, volume 900 of *Lecture Notes in Computer Science*. Springer-Verlag, pp. 229–242.

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

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

Motzkin, T. 1952. *The Theory of Linear Inequalities*. Santa Monica, CA: Rand Corp.

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

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

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

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

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

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

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

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

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

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

Stiver, J., Antsaklis, P., and Lemmon, M. 1996a. An invariant based approach to the design of hybrid control systems. In *IFAC 13th Triennial World Congress*, volume J, San Francisco, CA, pp. 467–472.

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

Stiver, J., Koutsoukos, X., and Antsaklis, P. 2001. An invariant based approach to the design of hybrid control systems. *International Journal of Robust and Nonlinear Control* 11(5): 453–478.

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

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

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

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