X.D. Koutsoukos and P.J. Antsaklis, "A Hybrid Feedback Regulator Approach to Control an Automotive Suspension System," H ybrid S ystems: C omputation a nd C ontro l , N. Lynch, B. Krogh Eds., pp. 188-201 , Lecture Notes in Computer Science, LNCS 1790, Springer-Verlag, 2000. Presented at the Third International Workshop H ybrid S ystems: C omputation a nd C ontrol (HSCC200 0), Pittsburgh, PA. , March 23-25, 2000.

A Hybrid Feedback Regulator Approach to Control an Automotive Suspension System*

Xenofon D. Koutsoukos and Panos J. Antsaklis

Department of Electrical Engineering
University of Notre Dame
Notre Dame, IN 46556
{xkoutsou,antsaklis.1}@nd.edu

Abstract. In this paper, we demonstrate a novel hybrid control synthesis approach using an automotive suspension system. Discrete abstractions are used to approximate the continuous dynamics and emphasis is placed on the nondeterministic nature of the abstracting models. The regulator problem for hybrid systems is formulated for safety specifications and algorithms for control design are presented.

1 Introduction

In this paper, a novel systematic methodology for hybrid control synthesis is presented and an example of an automotive suspension system is used to illustrate the approach. The main advantage of the approach is that it provides a convenient general framework for hybrid systems not only for analysis, but more importantly for controller synthesis. Discrete abstractions of the continuous dynamics are studied and the emphasis is placed on the nondeterministic nature of the abstracting models. The notion of quasideterminism is used to characterize discrete abstractions that can be used for control design. The class of systems we are particularly interested in is the class of piecewise-linear systems. Note that the analysis and synthesis algorithms have been implemented using general purpose software, namely Matlab, Simulink, and Stateflow.

Early results of the approach have appeared in [7, 6]. The approach has been influenced particularly by [1] where a feedback architecture of a continuous plant with a discrete-event controller is used for hybrid control design. Piecewise-linear systems evolving in discrete-time have been studied in [11, 13] and they represent an important class of systems with many practical applications. Recently, the class of piecewise-linear systems has attracted the attention of many researchers, see for example [5, 2]. Analysis and synthesis methodologies based on discrete abstractions have been studied extensively in the hybrid system literature [9, 8].

The paper is organized as follows. The automotive suspension system is introduced in Section 2. In Section 3, the modeling formalism is briefly outlined. In Section 4, the deterministic nature of the discrete abstractions is discussed and

* The partial financial support of the National Science Foundation (ECS95-31485) and the Army Research Office (DAAG55-98-1-0199) is gratefully acknowledged.

X.D. Koutsoukos and P.J. Antsaklis, "A Hybrid Feedback Regulator Approach to Control an Automotive Suspension System," Hybrid Systems: Computation and Control, N. Lynch, B. Krogh Eds., pp. 188-201 , Lecture Notes in Computer Science, LNCS 1790, Springer-Verlag, 2000. Presented at the Third International Workshop Hybrid Systems: Computation and Control (HSCC2000), Pittsburgh, PA. , March 23-25, 2000.

algorithms for the computation of the discrete approximations are presented. Finally, the regulator problem for hybrid systems is formulated in Section 5.

2 Automotive Suspension System

This example describes a simplified model of an automotive suspension for an independent wheel. The diagram of Figure 1 illustrates the modeled characteristics. We represent the suspension as a spring/damper system equipped with a compressor and an escape valve. We concentrate only on bounce degrees of freedom, which are represented in the model by the vertical displacement and velocity. The chassis level is raised by pumping air into the system and lowered by opening an escape valve. The suspension influences the bounce according to the equations

$$F = -2k(z+h) - 2c\dot{z} \tag{1}$$

$$m\ddot{z} = F - mg + u \tag{2}$$

where $z, \dot{z}$, and $\ddot{z}$ are the vertical displacement, velocity, and acceleration respectively. The spring and damping rate of the system are represented by the constants k and c. There are two inputs to the model. The first input is the road height h caused by irregularities in the road surface and the second input is the force u caused by the air pressure of the compressor or the escape valve.

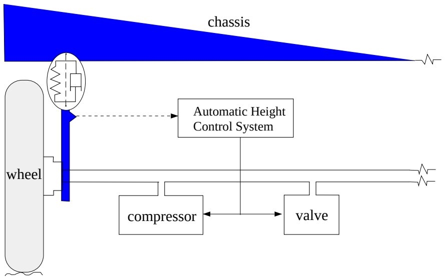


Fig. 1. Automatic height control system

The principal objective in this example is to design an automatic height control system, which increases driving comfort, allows the driver to select the chassis level according to off-road and on-road conditions, and does not violate driving safety. We consider two driving modes for the system, *straight* and *curve*. While in straight driving mode, the driver or a higher level control system in an autonomous vehicle, selects the set-point (sp) for the vertical displacement. The objective of the controller is to guarantee that the vertical displacement remains in a tolerance interval $[sp-lt, sp+ht]$ for any road disturbance from a prescribed bounded set. While in curve mode, the requirement is that the control system does not influence the chassis level, using either the compressor or the escape valve, so not to violate the safety of the system.

In this paper, the design of the controller that selects the action of the compressor and the escape valve is formulated as a hybrid control synthesis problem. A controller is designed based on discrete abstractions of the continuous dynamics using the refinement algorithm presented in Section 5. The controller is responsible for generating the control laws that guarantee that the chassis level will track the set-point within the prescribed tolerance while in straight-driving mode and will suspend the active control while in turning mode. Note that pneumatic suspension system examples have been used in the hybrid system literature to illustrate verification algorithms in a linear hybrid automata setting [4, 14, 3].

3 Modeling of Hybrid Systems

3.1 Hybrid system model

We propose to model hybrid systems as *set-dynamical systems* [10]. A *set-dynamical system* (SDS) is denoted as $(X, U, Y; f, g)$ where X is the state set of the system, U is the input set, Y is the output set, $f : X \times U \to X$ is the state transition function, and $g : X \times U \to Y$ is the output function. It is important to distinguish between the controlled and the uncontrolled inputs (disturbances) of an SDS. Furthermore, in the case when the measurements are different than the outputs, a measurement set M and a measurement function m can be included in the system's description.

In order to describe the behavior of a dynamical system, the notion of time must be included in the system's representation and this is accomplished with an index set J equipped with a simple order relation. Assume that the *index set* J is given. Define *index functions* $\alpha : \mathbb{N} \to J$. An index function is said to be *admissible* if $n_1 \leq n_2 \Rightarrow \alpha(n_1) \leq \alpha(n_2)$ (i.e. α is order preserving), and $n_1 \neq n_2 \Rightarrow \alpha(n_1) \neq \alpha(n_2)$ (i.e. α is injective). The state $x \in X$ is associated with an index $j(n)$ meaning the state at time $j(n)$.

A *hybrid dynamical system* (HDS) is defined as an SDS where the constituent sets consist of a continuous and a discrete part. We assume that the continuous part is a subset of a finite dimensional vector space and that the discrete part is finite.

Definition 1. *A hybrid dynamical system is defined by* $(X, U, D, Y, M; f, g, m)$ *where* $X = X_c \times X_d$ *is the state set;* $U = U_c \times U_d$ *is the set of control inputs consisting in general of continuous and discrete controls;* $D = D_c \times D_d$ *is the set of disturbances;* $Y = Y_c \times Y_d$ *is the output set;* $M = M_c \times M_d$ *is the measurement set;* $f : X \times U \times D \to X$ *is the state transition function;* $g : X \times U \times D \to Y$ *is the output function; and* $m : X \times U \times D \to M$ *is the measurement function.*

Presently, we have focused on *piecewise-linear systems* [11, 13] to facilitate the development of analysis and synthesis tools. These systems arise when the state set and/or the input set are partitioned into regions described by linear equalities and inequalities and the dynamics at each region are described by linear (or affine) state transitions. Output and measurement maps can be defined also in a similar way. The class of piecewise-linear systems is quite general as it includes linear systems, finite state machines, and their interconnections. They can be used also in many instances as approximations of more general systems.

Control specifications and primary partition Control specifications for hybrid systems can include safety requirements that are usually formulated with respect to a partition of the state space of the system. Consider the state set X of an SDS and define the mapping $\pi : X \to \mathbb{P}(X)$ 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 state set X into the equivalence classes of E_π and will be called *generator*. We assume that the partition defined by π is appropriate for extraction of important information for the system and it will be called the *primary partition*. More specifically, we are interested in the case when $X = \mathbb{R}^n$ and the generator is defined by a set of hyperplanes in $\mathbb{R}^n$. Note that such piecewise-linear regions arise in many applications.Consider the collection $\{h_i\}_{i=1,2,\ldots,\ell}, \;\; h_i : \mathbb{R}^n \to \mathbb{R}$ of real-valued functions of the form $h_i(x) = g_i^T x - w_i, \; i = 1, 2, \ldots, \ell$ where $g_i \in \mathbb{R}^n$ and $w_i \in \mathbb{R}$. Let $H_i = \ker(h_i) = \{x \in \mathbb{R}^n : h_i(x) = g_i^T x - w_i = 0\}$ and assume that H_i is an $(n-1)$-dimensional hyperplane $(\nabla h_i(x) = g_i^T \neq 0)$. We define the function $\hat{h}_i : \mathbb{R}^n \to \{-1, 0, 1\}$ by

$$\hat{h}_i(x) = \begin{cases} -1 \; if \; h_i(x) < 0 \\ \;\; 0 \;\; if \; h_i(x) = 0 \\ \;\; 1 \;\; if \; h_i(x) > 0 \end{cases} \tag{3}$$

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 $\pi : \mathbb{R}^n \to \{-1, 0, 1\}^\ell$ there is a bijection between $\{-1, 0, 1\}^\ell$ and the quotient set X/E_π (they are the same set).

Measurements and final partition Suppose that at time k we have that $\tilde{y}(k) = \pi(x(k)) \in X/E_\pi$. If it is agreed that the granularity of the partition generated by the mapping π 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 $\tilde{y}(k+1) = \pi(x(k+1)) \in X/E_\pi$. This can be accomplished by considering a finer partition than the partition defined by the generator π to obtain better estimates for the continuous state. This partition will be called the *final partition* and will be determined using the quasideterminism property discussed below. The generator π_F is defined in a similar way as the output function 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. The function $z = \pi_F(x)$ can be viewed as a *measurement function* that provides some information about the continuous state. Intuitively, our ability to make decisions to influence the behavior of the system depend on the amount of information contained in the measurement signal.

Example - The automotive suspension system The system contains continuous dynamics due to the spring/damper subsystem and discrete dynamics due to pneumatic part of the suspension. Furthermore, the control specifications contain constraints for both the continuous and discrete variables. For these reasons, the automotive suspension system is modeled as the hybrid dynamical system $(X, U, D, Y, M; f, g, m)$. The state space of the system is $X = X_c \times X_d = \mathbb{R}^2 \times \{straight, curve\}$ representing the displacement and the velocity of the system, and the driving mode. The set of control actions is $U = \{u_0, u_1, u_2\}$ corresponding to the case when the controller is suspended, the compressor is on, and the escape valve is open respectively (the compressor and the valve can not operate simultaneously). The set of exogenous input (that cannot be controlled) is $D = D_c \times D_d = \mathbb{R} \times \{turn, resume\}$ representing the road height and the selection for the driving mode respectively. The output set is $Y = \mathbb{R}$ representing the chassis level. The measurement set is described as the quotient set X/E_{π_F} induced by the final partition π_F that is to be determined in Section 4. The state transition function $f : X \times U \times D \to X$ is described by $x(k+1) = Ax(k) + Bu(k) + Ed(k)$ where x_1 is the displacement of the chassis, x_2 is the velocity, u is the applied force due to either the compressor or the escape valve, and d is the road height. The parameters of the system A, B, and E are derived from the differential equations (2) by sampling at a prescribed rate T. Finally, the output function is $y(k) = Cx(k)$ where $C = [1, 0]$ and the measurement function $z(k) = \pi_F(x(k))$ returns the membership of the state in one of the equivalences classes of the final partition.

3.2 Control Specifications

Regulatory feedback control of hybrid dynamical systems is based on a representation of the control specifications as a set-dynamical system which is usually called the exosystem. In this paper, we focus on the case when the exosystem is described by a finite automaton. The case when hard time constraints on the transitions of the exosystem are necessary can also be studied in this framework by including clocks in the description of the plant.

Example - The automotive suspension system The control specifications for the automotive suspension system are now described. While in straight driving mode, the driver or a higher level control system in an autonomous vehicle, selects the set point (sp) for the vertical displacement. The objective of the controller is to guarantee that the vertical displacement remains within a tolerance interval $[sp - lt, sp + ht]$ for any road disturbance from a prescribed bounded set. While in curve mode, the requirement is that the control system does not influence the chassis level, using either the compressor or the escape valve, so not to violate the safety of the system. The control specifications can be described formally by the finite automaton shown in Figure 2(i). The state e_0 corresponds to the case the driving mode is straight, where the requirement for the chassis height is to be inside the tolerance interval $[sp - lt, sp + ht]$. The states e_1 corresponds to the case when the driving mode is curve. The input alphabet is $\Sigma = \{turn, resume, \epsilon\}$ where ϵ is a void event.

The primary partition can be derived from the control specifications in a straightforward manner and is described by $h_1(x) = x_1 - (sp + ht)$ and $h_2(x) = x_1 - (sp - lt)$. Then the generator is defined by $\pi(x) = [\hat{h}_1(x), \hat{h}_2(x)]^T$ where the function $\hat{h}_i$ is defined in Equation (3) and it separates the state space into five equivalence classes. For simplicity, we will consider that the safe region is described by the closed interval $[sp - lt, sp + ht]$ and will consider only three regions corresponding to *safe, high, and low* chassis levels as shown in Figure 2(ii).

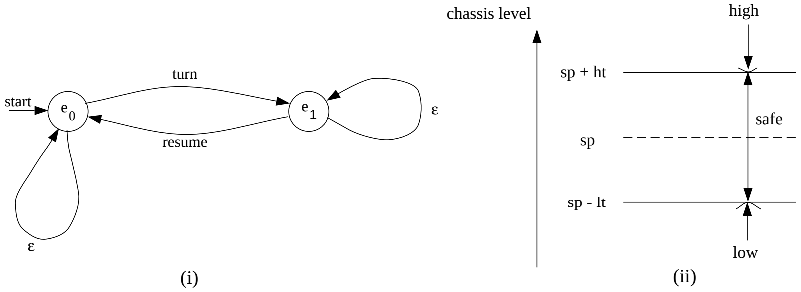


Fig. 2. (i) Exosystem, (ii) Primary Partition

The finite automaton of Figure 2(i) can be represented by the set-dynamical system $(X_e, V_e, Y_e, M_e; f_e, g_e, m_e)$ where $X_e = \{e_0, e_1\}$ is the state set, $V_e = \{turn, resume, \epsilon\}$ is the set of exogenous inputs, $Y_e = \{turn, resume, \epsilon\}$ is the output set (which characterizes part of the exogenous inputs to the plant), and $M_e = X/E_\pi$ is the set of output requests. The state transition function $f_e : X_e \times V_e \to X_e$ is the state transition of the automaton, the output function $g_e : X_e \times V_e \to Y_e$ is defined as $g_e(e, v) = v$ for every $e \in X_e$ and $v \in V_e$. Finally, the output request (measurement) function is defined as follows.

$$m_e(e, v) = \begin{cases} safe & \text{for } e = e_0,\ \forall v \in V_e, \\ \tilde{y} \in \{safe, low, high\} & \text{for } e = e_1,\ \forall v \in V_e \end{cases} \tag{4}$$

4 Partition Refinement and Discrete Abstractions

4.1 Motivation

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. Let f be the state transition function of an SDS and assume that the inputs are fixed. Consider the diagram in Figure 3-(a). 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 [10] that $\tilde{f}$ exists if and only if

$$x_1 \, E_\pi \, x_2 \quad \Rightarrow \quad (\pi \circ f)(x_1) = (\pi \circ f)(x_2) \tag{5}$$

(where $\circ$ denotes function composition) and moreover, if (5) 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*.

4.2 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 control specifications are preserved in the quotient system resulting in more efficient algorithms to partition the state space that are applicable to larger classes of hybrid systems. Quasideterminism is a weaker requirement than the existence of a finite bisimulation. A partition that results in a quasideterminism can be always be computed for piecewise-linear systems, while recent results have shown that finite bisimulations exist only for limited classes of systems [8]. 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.

Suppose that at time k, $\pi(x(k)) \in X/E_\pi$ is known. In the case when the estimates of the state at time k provide sufficient information to uniquely determine the membership of the state of the induced system at time $k+1$ on an equivalence class of E_π, the system is said to be quasideterministic. The notion of quasidetermism is illustrated in Figure 3. Although we do not compute an equivalence relation that guarantees the existence of a mapping $\tilde{f}$ that preserves the reachability properties of the original system, we exploit the commutativity of the diagram (c) in Figure 3 in order to analyze the reachability properties with respect to the control specifications. The formal definition for the concept of quasideterminism is given in later in the section.

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 $B(X)$ is defined as $B_1 \leq B_2$ if $(x_1, x_2) \in B_1 \Rightarrow (x_1, x_2) \in B_2$. Let $E(X)$ be the set of all equivalence

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

(a) Primary Partition

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

(b) Final Partition

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

(c) Quasideterministic Partition

Fig. 3. Quasideterminism and the partitions of the state space

relations on X. We have that $E(X) \subset B(X)$ and $E(X)$ inherits the partial order of $B(X)$. A lattice structure can be developed on the set of all equivalence relations on X (for more details see [10]). 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.*

Definition 2. *The hybrid system $(X, U, D, Y, M; f, g, m)$ 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 $\tilde{z} \in X/E_{\pi_F}$ and for all states $x \in X$ such $\pi_F(x) = \tilde{z}$, there exists unique region of the primary partition $\tilde{y} \in X/E_\pi$ such that $\tilde{y} = \pi(f(x, u, d))$ for every control action $u \in U$ and exogenous input $d \in D$.*

If the hybrid system $(X, U, D, Y, M; f, g, m)$ 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 $E_{\pi_q} \leq E_{\pi_F}$. Refinement of the state space partition will terminate if we can guarantee that there is a control policy to satisfy the specifications.

4.3 Partition Refinement

In the following, we present some basic results that will be used in the theoretical analysis of the algorithms for the partition refinement. A *piecewise-linear (PL) subset* [12] 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 $f(x) > a$. An alternative way to define PL sets which is important for our discussion is the following [12].

Definition 3. *Let $\mathcal{L}$ be the first-order language defined by (i) a set of (countably many) variables $\{x_1, x_2, \ldots\}$, (ii) the connective symbols $\neg$ and $\to$, (iii) the quantifier $\forall$, the parentheses (and) and the comma, (iv) A set of constants $\{r\}$ for each real number r, (v) A set of unary functions $\{r \cdot ()\}$ for each real number, the binary function $+$, (vi) the relational symbols $>$ and $=$.*

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

The above lemma is proved in [12]. The conclusion of the lemma is that any set defined using quantifiers can be also defined using only propositional connectives. In order to refine the state space, we define the predecessor operator $pre : \mathbb{P}(X) \to \mathbb{P}(X)$ as

$$pre(P) = \{x | \exists u \in U, \forall d \in D, f(x,u,d) \in P\}. \tag{6}$$

The set $pre(P)$ represents all the states x for which there is a control action that will enforce the state to remain in P for any disturbance d. If the set P is piecewise-linear, then from Lemma 1 it follows that the set $pre(P)$ is also piecewise-linear and can be defined using only propositional connectives.

In the remaining of the paper, we will concentrate on the case the hybrid system is described by

$$(X, U, D, Y, M; f, g, m) \tag{7}$$

with finite input set U, bounded disturbance set D, and transition function given by $x(k+1) = Ax(k) + Bu(k) + Ed(k)$. Similar results can be developed for other classes of piecewise-linear systems.

Initially, assume that the state transition function is given by $x(k+1) = Ax(k) + Bu(k)$ where $x \in \mathbb{R}^n$ and the input u takes values in a finite set $U \subset \mathbb{R}^m$. For fixed control action $u \in U$ the dynamics of the system are described by the mapping $f_u : \mathbb{R}^n \to \mathbb{R}^n$ with $f_u(x) = Ax + Bu$. We want to compute the set of all the state x that can be driven in P by the control action u by defining the predecessor operator $pre_{f_u}(P) = \{x | f_u(x) = Ax + Bu \in P\}$.

Lemma 2. *Consider the affine function $h(x) = g^T x - w$ and the set $H = \ker(h) = \{x | g^T x - w = 0\}$. Let $H' = \{x | f_u(x) = Ax + Bu \in H\}$ be the set of all $x \in \mathbb{R}^n$ that can be driven in H by application of the affine mapping f_u. Then $H' = \ker(h')$ where $h'(x) = g^{'T} x - w'$ with $g^{'T} = g^T A$ and $w' = w - g^T Bu$. In addition, if $Y = int(K')$ is an open halfspace bounded by H', then $f_u(Y) = int(K)$, that is $f_u(Y)$ is an open halfspace bounded by H.*

Next, we define the halfspace $P(g,w) = \{x | g^T x \le w\},\ g \ne 0$ and we compute the set of all states that can be driven to P by using the predecessor operator $pre_{\exists} : \mathbb{P}(X) \to \mathbb{P}(X)$ defined as $pre_{\exists}(P) = \{x | \exists u \in U, f_u(x) = Ax + Bu \in P\}$.

Lemma 3. *Consider the set $P(g,w) = \{x | g^T x \le w\},\ g \ne 0\}$, then $pre_{\exists}(P) = \{x | g^T Ax \le w - g^T Bu^*\}$ where u^* is the maximizer of the function $w(u) = w - g^T Bu$ over the set of control actions U.*

Let $f : X \to Y$ be a mapping and consider the sets $D \subset X$ and $E \subset Y$. The *image* of D and the *inverse image* of E under the mapping f are defined by $f(D) = \{f(x) | x \in D\},\ \ f^{-1}(E) = \{x | f(x) \in E\}$. It is easily verified that the map $f^{-1} : \mathbb{P}(Y) \to \mathbb{P}(X)$ commutes with unions, intersections, and complements. The operator $pre_{f_u} : \mathbb{P}(X) \to \mathbb{P}(X)$ $(X = \mathbb{R}^n)$ clearly returns the inverse image of P under the mapping f_u for fixed input and therefore commutes with unions, intersection, and complements. The notation pre_{f_u} has been used instead of f_u^{-1} in order to be consistent with the notation when the control action is not fixed. In the case when the input set is finite, the set $pre_\exists(P)$ can be computed for any PL set as the union $\bigcup_{u_i \in U} pre_{f_{u_i}}(P)$.

Next, we consider the case when continuous disturbances are present and we assume that for a fixed discrete control action the description of the system is $x(k+1) = Ax(k) + Bd(k)$ where $x \in \mathbb{R}^n$ and $d \in D \subset \mathbb{R}^m$ a disturbance which takes values in a bounded polyhedron. We define a new predecessor operator $pre_f^d : \mathbb{P}(X) \to \mathbb{P}(X)$ by $pre_f^d(P) = \{x | \forall d \in D, f(x,d) = Ax + Bd \in P\}$. This operator returns all the states which will be in the set P at the next time step for every possible disturbance.

Lemma 4. *Consider the set* $P = P(g,w) = \{x | g^T x \leq w\}$*, then* $pre_f^d(P) = \{x | g^T Ax \leq w - g^T Bd^*\}$ *where* $d^* = argmin_{d \in D}\{-g^T Bd\}$*.*

The predecessor operator in the case of bounded disturbances commutes with the intersection of halfspaces. Note that this result is a consequence of the equivalence $(\forall x)(\phi(x) \wedge \psi(x)) \leftrightarrow (\forall x)\phi(x) \wedge (\forall x)\psi(x)$ in predicate logic.

In the following, we consider the system $x(k+1) = Ax(k) + Bu(k) + Ed(k)$ where the disturbance d takes values in a bounded polyhedral set D and the control input u takes values in a finite set U and the polyhedral set $P = \{x | g_1^T x \leq w_1 \wedge \cdots \wedge g_p^T \leq w_p\}$. Then by using the results of this section we have that

$$\begin{aligned} pre(P) &= \{x | \exists u \in U, \forall d \in D, f(x,u,d) = Ax + Bu + Ed \in P\} \\ &= \bigcup_{u_i \in U} pre_{f_{u_i}}^d(P) \\ &= \bigcup_{u_i \in U} \{x | g_1^T Ax \leq w_1 - g_1^T Bu_i - g_1^T Ed_1^* \wedge \cdots \wedge g_p^T Ax \leq w_p - g_p^T Bu_i - g_p^T Ed_p^*\} \end{aligned}$$

where $d_i^* = argmax_{d \in D}\{-g_i^T Bd\}$. Next, consider the hyperplanes $h_i'(x) = g_1^T Ax - (w_1 - g_1^T Bu_i - g_1^T Bd_1^*)$, $i = 1, \ldots, p$ and the partition $\pi' \in E_P(X)$ defined by those hyperplanes using Equation (3).

Proposition 2. *The hybrid system (7) with primary and final partition defined by* X/E_π *and* $\inf(E_\pi, E_{\pi'})$ *respectively is quasideterministic with respect to the primary partition.*

The implication of the above proposition is that for every state, every control action, and every disturbance the membership of the state at the next time step to an equivalence class of the primary partition can be uniquely determined from the current region of the final partition. Given a fixed time window repetitive applications of the predecessor operator can take into consideration more than one time steps. At this point it is possible to construct a discrete-event system based on the final partition π_F and extend supervisory control techniques in order to exploit the information that is preserved in the discrete abstraction due to quasideterminism. However, we continue with our analysis of specific control problems for which we can formulate conditions for the existence of control policies that guarantee that the specifications are satisfied.

4.4 Safety

In the following, we focus on the safety problem and we describe algorithms for the refinement of the state space partition that result in quasideterministic systems. Given a set of safe states described by the piecewise-linear set $P \subset \mathbb{R}^n$ and an initial condition $x_0 = x(0) \in P$, we say that the system is *safe* if $x(k) \in P$ for every k. The system is safe with respect to the set P if

$$P \subseteq pre(P) = \{x | \exists u \in U, \forall d \in D, f(x, u, d) \in P\}. \tag{8}$$

The validity of equation (8) can be tested using the representation of $pre(P)$ without quantifiers. Since the set $pre(P)$ is piecewise-linear but not polyhedral, the development of efficient algorithms that test if the equation (8) holds is necessary and is a topic of current research. A simple algorithm to perform this test consists of representing the complement of $pre(P)$ as the union of polyhedra $Q = [pre(P)]^c = \bigcup_{i=1,\dots,\rho} Q_i$ and then, testing if $P \cap Q_i = \emptyset$ for every $i = 1, ..., \rho$ using linear programming techniques. A simple way to express Q as the union of polyhedra is to consider all the inequalities that define Q pairwise and eliminate all the pairs that correspond to parallel hyperplanes.

Proposition 3. *Given the polyhedral set of safe states P and the hybrid system (7), if $P \cap Q_i = \emptyset, i = 1, \dots \rho$ where $Q = [pre(P)]^c = \bigcup_{i=1,\dots,\rho} Q_i$, then there exists control policy that guarantees that the system is safe.*

Example - Automotive Suspension System The automotive suspension system is safe if the chassis level is inside the interval $[sp - lt, sp + ht]$ while in straight driving mode. Our approach for the design of the controller is that given the desired-set point and therefore the primary partition, a final partition can be constructed and the conditions of Proposition 3 can be tested in an autonomous manner. If there exists a control policy that guarantees that the system is safe, then a controller that implements such a policy can be designed based on the discrete abstraction induced by X/E_{π_F}. The same approach can be used also off-line to characterize all the set-points for which there exists a control policy that guarantees safety.

In order to construct the final partition, we translate the control specification from the output space to the input space to obtain the set $P_1 = \{(x_1, x_2) | sp - lt \leq x_1 \leq sp + ht\}$. Clearly, the set P_1 is unbounded in the state space $\mathbb{R}^2$. From Lemma 2 it follows that the set $pre(P_1)$ is bounded by hyperplanes that in general intersect with P_1 and therefore, it is not possible that $P_1 \subseteq pre(P_1)$. The practical implication of this observation is that if the chassis level is very close to the boundary of the set P_1, then if the chassis vertical velocity is large and directed towards the unsafe region, there will be no finite control input that will guarantee safety. In order to proceed with the controller design we have to determine a bounded approximation of the set P_1 by taking into consideration realistic bounds for the chassis vertical velocity. The final partition can be determined using the partition refinement algorithms described above. The primary and final partition for typical values of the system parameters are shown in Figure 4 where it can be seen that $P \subset pre(P)$.

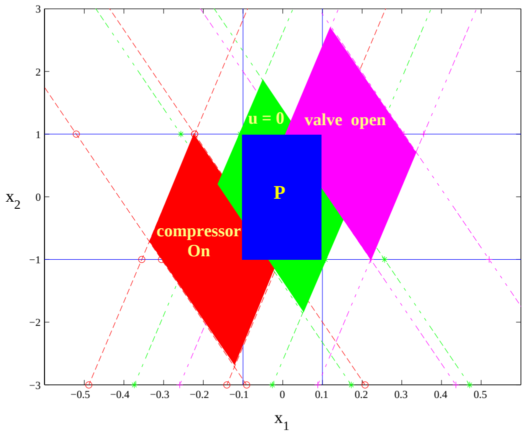


Fig. 4. Final partition

5 Hybrid System Regulator

In this section, the regulator problem for hybrid systems is formulated. In general, a regulator requests certain types of outputs from the plant so that these are attained in the presence of disturbances. The desired outputs are characterized by a regulation condition and they can be described as the outputs of another SDS, called the *exosystem*. The plant and the exosystem are linked by a controller to form a regulator as shown in the Figure 5(i). A feedback controller can be designed to regulate the system. The main characteristic of the controller is that it contains a copy of the exosystem in accordance to the “internal model principle”.

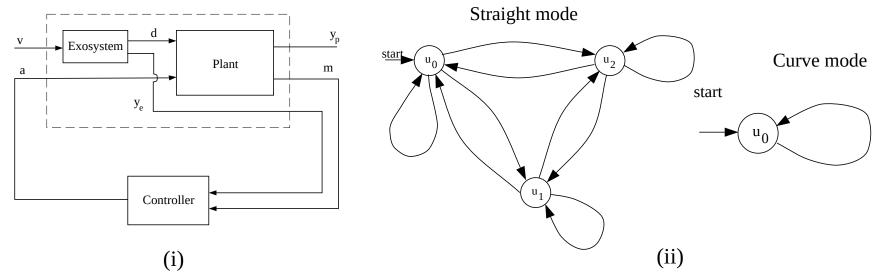
In the following, we consider the safety problem and we describe how a controller can be designed based on the discrete abstraction induced by the final partition. The state of the controller correspond to the regions of the final partition and the current state $x_c = \pi_F(x)$ can be determined by filtering the plant measurements using the inequalities that define the equivalence classes of the final partition. The controller can be described by the SDS $\mathcal{C} = (X_c, Y \times M, U; f_c, g_c)$ where X_c is the state set of the controller; $Y \times M$ is the input set of the controller consisting of pairs describing the output request and that actual plant output every time instant; U is the output set representing the control actions; $f_c : X_c \times (Y \times M) \to X_c$ is the state transition function for the controller; and $g_c : X_c \times (Y \times M) \to U$ is the output function given by $u = g_c(x_c, (m_e(x_e, v), \pi_F(x)))$. Since for some states there exist more than one control inputs that can be applied for safety, there are several ways to implement the output function of the controller. For example, the output function can

X.D. Koutsoukos and P.J. Antsaklis, "A Hybrid Feedback Regulator Approach to Control an Automotive Suspension System," H ybrid S ystems: C omputation a nd C ontro l , N. Lynch, B. Krogh Eds., pp. 188-201 , Lecture Notes in Computer Science, LNCS 1790, Springer-Verlag, 2000. Presented at the Third International Workshop H ybrid S ystems: C omputation a nd C ontrol (HSCC200 0), Pittsburgh, PA. , March 23-25, 2000.

defined by

$$u = \begin{cases} u_0 & \text{if } x \in P_0 = pre^d_{f_{u_0}} \\ u_1 & \text{if } x \in P_1 = pre^d_{f_{u_1}} \setminus P_0 \\ \vdots & \vdots \\ u_N & \text{if } x \in P_N = pre^d_{f_{u_N}} \setminus P_{N-1} \end{cases} \tag{9}$$

Example - Automotive Suspension System The controller for the automotive suspension system is shown in Figure 5(ii). For the straight driving mode the controller is represented as a finite automaton with three different states corresponding to the regions of the final partition for the set P and output function defined by (9). For the curve driving mode, the controller consists of one state with constant output function $u = 0$. The controller communicates with the plant and the exosystem in a synchronous manner.



(i)

(ii)

Fig. 5. (i) Hybrid system regulator, (ii) Controller

Remark A problem related to safety is to examine if there exists a control policy that will drive the state of the system to a prescribed region. For example, since at the end of a curve the chassis level may not be inside the interval $[sp - lt, sp + ht]$, it is required that as soon as the system is in *straight* mode the chassis level must be driven to the safety region by using either the compressor or the valve. This is a reachability specification that can be also studied in the framework presented in the paper. The final partition can be constructed by repetitive applications of the predecessor operator. For the termination of the partition refinement algorithm, the reachability specifications should be characterized by bounds on the time for the state to reach the desired region.

6 Conclusions

A novel hybrid control synthesis approach is demonstrated using an automotive suspension system. Controller design is based on quasideterministic discrete abstractions of the continuous dynamics. The regulator problem for hybrid systems is formulated for safety specifications and algorithms for control design are presented. Although a second-order system was used the approach, the methodologies and the algorithms described are applicable to more complex systems. The approach has been validated with simulations using Matlab, Simulink, and Stateflow but simulation results are omitted due to length limitations. An important point is that the above approach is potentially implementable on-line for real-time control. Note that due to space limitations, detailed descriptions of the technical results were omitted, but they can be obtained by contacting the authors.

References

1. P. Antsaklis, J. Stiver, and M. Lemmon. Hybrid system modeling and autonomous control systems. In R. L. Grossman, A. Nerode, A. P. Ravn, and H. Rischel, eds., *Hybrid Systems*, Vol. 736, *LNCS*, 366–392. Springer-Verlag, 1993.
2. A. Bemporad and M. Morari. Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35(3):407–427, 1999.
3. A. Bemporad and M. Morari. Verification of hybrid systems via mathematical programming. In *HSCC 99: Hybrid Systems—Computation and Control*, Vol. 1569, *LNCS*. Springer-Verlag, 1999.
4. R. Fehnker. Automotive control revised - linear inequalities as approximations of reachable sets. In T. Henzinger and S. Sastry, eds., *HSCC 98: Hybrid Systems—Computation and Control*, Vol. 1386, *LNCS*, 110–125. Springer-Verlag, 1998.
5. M. Johansson and A. Rantzer. Computation of piecewise quadratic Lyapunov functions for hybrid systems. *IEEE Transactions on Automatic Control*, 43(1):31–45, 1998.
6. X. Koutsoukos and P. Antsaklis. Design of hybrid system regulators. In *Proceedings of the 38th IEEE Conference on Decision and Control*, 3990–3995, Phoenix, AZ, Dec. 1999.
7. X. Koutsoukos and P. Antsaklis. Hybrid control of a robotic manufacturing system. In *Proceedings of the 7th IEEE Mediterranean Confereence on Control and Automation*, 144–159, Haifa, Israel, June 1999.
8. G. Lafferriere, G. Pappas, and S. Sastry. Reachability analysis of hybrid systems using bisimulations. In *Proceedings of the 37th IEEE Conference on Decision and Control*, 1623–1628, Tampa, FL, 1998.
9. J. Raisch and S. O'Young. Discrete approximation and supervisory control of continuous systems. *IEEE Transactions on Automatic Control*, 43(4):568–573, 1998.
10. M. Sain. *Introduction to Algebraic System Theory.* Academic Press, 1981.
11. E. Sontag. Nonlinear regulation: The piecewise linear approach. *IEEE Transactions on Automatic Control*, 26(2):346–358, 1981.
12. E. Sontag. Remarks on piecewise-linear algebra. *Pacific Journal of Mathematics*, 92(1):183–210, 1982.
13. E. Sontag. 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*, Vol. 1066, *LNCS*, 436–448. Springer-Verlag, 1996.
14. T. Stauner, O. Muller, and M. Fuchs. Using HYTECH to verify an automotive control system. In O. Maler, ed., *Hybrid and Real-Time Systems*, Vol. 1201, *LNCS*, 139–153. Springer-Verlag, 1997.