M. D. Lemmon and P. J. Antsaklis, "A Computationally Efficient Framework for Hybrid Controller Design," P roc o f t he 1 994 I EEE/IFAC J oint S ymposium o n C o mputer-Aided C ontrol S ystem D esig n , pp 333-338, Tucson, AZ, March 7-9, 1994.

A Computationally Efficient Framework for Hybrid Controller Design

Michael Lemmon *and Panos Antsaklis
Department of Electrical Engineering
University of Notre Dame
Notre Dame, IN 46556

Abstract

A hybrid control system uses a logical DES controller to supervise the behaviour of a continuous-state plant. An important issue in hybrid control systems concerns precisely what constitutes "logically stable" behaviour in such systems. The challenge concerns the formulation of computationally efficient procedures for designing such logically stable systems. This paper provides a formal definition of hybrid system stability and outlines how this definition leads to a polynomial time procedure for hybrid control system synthesis. **Keywords:** hybrid systems, inductive inference, stability

1 Introduction

Hybrid control systems arise when a discrete event system (DES) is used to supervise the behaviour of a continuous state system (CSS). Such systems arise in the supervisory control of large scale systems found in semiconductor manufacturing, power distribution, flexible manufacturing, and chemical process control. One approach to the design of hybrid control systems extracts a logical model of the continuous-state system's behaviour. This logical model is called the *DES plant*. The hybrid control system design approach then synthesizes a logical DES controller for the DES plant. Examples of this approach will be found in [4], [1], and [7].

In [7], the design of a DES controller is emphasized. In particular, a DES plant is extracted from a continuous-state system and an appropriate controller is synthesized using an extension of the Ramadge-Wonham model (RWM) framework. In this framework, a DES controller is synthesized by using a fixed point recursion which produces a supremal controllable DES plant model consistent with a formal specification on plant behaviour. For the RWM framework [5], the plant symbols can be divided into controllable and uncontrollable events, where controllable events are plant symbols which can be disabled by an appropriately designed supervisor. The RWM framework generally assumed that the DES plant was controllable so that any combination of plant symbols could be disabled by the controller. In general, however, a hybrid system will not have a controllable DES plant. Therefore, if the RWM framework is to be used, a method is needed to identify those DES plant symbols which are indeed controllable by a DES controller.

The preceding statements imply that if supervision of the equivalent DES plant is to be sufficient to supervise the original CSS plant, it is necessary that the DES plant be a logical *model* of the CSS plant. The DES plant can be viewed as a logical "interpretation" of the CSS plant's behaviour of interest. If that interpretation accurately or *truthfully* predicts the desired symbolic behaviour of the CSS plant, then it is said to be a *valid interpretation* or *model* of the CSS plant's desired behaviour. An important issue in the design of hybrid control system is therefore the verification of a DES plant's validity. This issue is often referred to as model verification.

The notion of *model verification* explored in this paper is significantly different from verification frameworks found in the computer science literature. In general, verification means that a logical system is being tested to see whether or not it exhibits a specified logical behaviour such as liveness or permissiveness. Such approaches to model verification can be seen as top-down protocols whose objective is to verify macroscopic logical behaviours. In this paper, a *bottom-up* perspective is adopted to the verification problem. Rather than verifying desired logical behaviour, the techniques developed in this paper are concerned with deciding the valid-

*The partial financial support of the National Science Foundation (IRI91-09298 and MSS92-16559) is gratefully acknowledged.

0-7803-1800-5/94/$3.00 © 1994 IEEE

Authorized licensed use limited to: UNIVERSITY NOTRE DAME. Downloaded on October 7, 2009 at 21:04 from IEEE Xplore. Restrictions apply.

M. D. Lemmon and P. J. Antsaklis, "A Computationally Efficient Framework for Hybrid Controller Design," P roc o f t he 1 994 I EEE/IFAC J oint S ymposium o n C o mputer-Aided C ontrol S ystem D esig n , pp 333-338, Tucson, AZ, March 7-9, 1994.

ity or rather *stability* of controlled logical transitions within the DES plant. Essentially, this paper provides a way for testing to see whether or not a specified transition in the plant DES can be controlled in a stable manner by a continuous-control vector. Consequently, the verification procedures discussed below are intimately connected with the underlying continuous-state dynamics of the plant, rather than the discrete logical behaviour desired of the plant.

The model verification approach presented in this paper represents the first step in a bottom-up design of hybrid system controllers. The bottom-up approach originates in a consideration of the continuous-state plant's dynamics and the limited set of control policies (agents) we have to control that plant. This approach attempts to supervise the CSS plant by building up a logical DES model of the plant. The logical model is built up as the controller attempts to transition the CSS plant in a consistent and stable manner between various logical states. The potential benefits of this approach are found in the computational complexity of the resulting scheme. Rather than starting with a large logical model and reducing it (as is done in the RWM framework), one starts with a formally specified behaviour and builds up. The resulting approach is therefore a constructive "search" technique.

This paper poses the model verification problem and shows how convex optimization procedures can solve the "link"-verification problem for large classes of hybrid systems. It is shown that DES plant validity is closely related to the Lyapunov stability of the CSS plant. For this reason, model (DES plant) validity is referred to as transition or T-stability. This paper presents a set of sufficient conditions guaranteeing DES plant T-stability. For many types of hybrid systems, these conditions can be represented as linear inequality constraints on a suitable parameterization of the interface between the CSS plant and DES controller. Feasible points of this inequality system therefore represent interfaces which insure the T-stability of transitions in the DES plant. It is shown how these T-stabilizing interfaces can be determined using the central-cut ellipsoid method. The resulting algorithm provides a method for inductively determining T-stabilizing interfaces in finite time where the convergence time scales in a polynomial manner with CSS plant complexity.

The results presented in this paper provide the foundations of a framework for the integration of human and machine operators. The basic framework consists of the following steps; 1) Translation of engineering specifications and operator procedures into a logical DES specification on the plant's desired behaviour, 2) Determing a DES controller which implements the specification in a logically stable manner, and 3) translating the plant DES obtained in step 2 into a a set of operator procedures. Step 3 represents the most important step in the design framework introduced above. This step shows how such plant DES design methods can be practically applied in existing plant environments. Progress on this front will allow for human operators to interact with exsting automatic DES controller synthesis methods in a way which takes advantage of both knowledge domains. Progress along this front will allow for the migration of DES synthesis methodologies onto the factory floor in a way which has yet to be realized. The results and concepts introduced in this paper suggest one way in which this integration might be achieved in a computationally efficient manner.

The remainder of this paper is organized as follows. Section 2 defines the hybrid control systems under consideration. Section 2 discusses the semantics of the hybrid control system. Section 3 presents a definition which insures the stability of the hybrid control system's logical behaviour. Section 4 discusses a procedure for the synthesis of stable DES controllers.

2 Hybrid Control Systems

A hybrid control system consists of four interconnected subsystems; the plant, the controller, the actuator, and the generator. More precisely, we define a hybrid control system, $\mathcal{H}$, as the ordered 4-typle, $\mathcal{H} = (\mathcal{P}_c, \mathcal{C}_d, \mathcal{A}, \mathcal{G})$ where $\mathcal{P}_c$ is the CSS plant, $\mathcal{C}_d$ is the DES controller, $\mathcal{G}$ is the interface generator, and $\mathcal{A}$ is the interface actuator.

The plant, $\mathcal{P}_c$, is the part of the model which represents the entire continuous-state (CSS) portion of the hybrid control system. The plant evolves over a subset $\bar{X}$ of $\Re^n$. The plant's trajectory, $\bar{x}(t)$, through $\bar{X}$ can be described by a set of ordinary differential equations.

$$\dot{\bar{x}}(t) = f(\bar{x}(t), \bar{r}(t)) \tag{1}$$

where $\bar{x} \in \Re^n$ is the CSS plant state and $\bar{r}$ is the CSS plant's control vector.

In hybrid control systems, the plant will be supervised by the application of logical directives. In this regard, the control vector trajectory, $\bar{r}(t)$ is piecewise constant. Using this fact, we can therefore represent the CSS plant,$\mathcal{P}_c$, of a hybrid system, $\mathcal{H}$, as the ordered $m+1$ tuple,

$$\mathcal{P}_c = (\bar{X}, \Phi_t^{\bar{r}_1}, \cdots, \Phi_t^{\bar{r}_m}) \tag{2}$$

Authorized licensed use limited to: UNIVERSITY NOTRE DAME. Downloaded on October 7, 2009 at 21:04 from IEEE Xplore. Restrictions apply.

M. D. Lemmon and P. J. Antsaklis, "A Computationally Efficient Framework for Hybrid Controller Design," Proc of the 1994 IEEE/IFAC Joint Symposium on Computer-Aided Control System Design, pp 333-338, Tucson, AZ, March 7-9, 1994.

$\Phi_t^{\bar{r}_i} : \bar{X} \rightarrow \bar{X}$ (for $i = 1, \ldots, m$) is the ith family (indexed by time, t) of transition operators generated by the differential equation

$$\dot{\bar{x}}(t) = f(\bar{x}(t), \bar{r}_i) \tag{3}$$

where $\bar{r}_i$ is a constant vector in $\Re^m$ and $f : \Re^n \times \Re^m \rightarrow \Re^n$ is a Lipschitz continuous function serving as the infinitesimal generator of the transition operators $\Phi_t^{\bar{r}_i}$.

The controller is a discrete event system. The controller will often be called the *supervisor*. It receives as inputs a sequence $\tilde{x}[n]$ of symbols generated by the interface drawn from a finite alphabet $\tilde{X}$ of *plant symbols*. The supervisor outputs a sequence $\tilde{r}[n]$ of symbols drawn from a finite alphabet $\tilde{R}$ of *control symbols*. The controller's dynamics are assumed to be modeled by a deterministic finite automaton (DFA). The DES controller, $\mathcal{C}_d$, for hybrid system $\mathcal{H}$, is the ordered 4-tuple, $\mathcal{C} = (\tilde{S}, \tilde{X}, \tilde{R}, \Phi, Q)$, where $\tilde{S}$ is a finite alphabet of controller symbols, $\tilde{X}$ and $\tilde{R}$ are finite alphabets of plant and control symbols, respectively. $\Phi : \tilde{S} \times \tilde{X} \rightarrow \tilde{S}$ is the controller's transition operator. $Q : \tilde{S} \rightarrow \Re$ is the controller enabling function. These two functions are assumed to yield the following recursive equations.

$$\tilde{s}[n] = \Phi(\tilde{s}[n-1], \tilde{x}[n]) \tag{4}$$

$$\tilde{r}[n] = \text{argmax}_{\tilde{r} \in \tilde{R}} Q(\tilde{s}[n]) \tag{5}$$

The generator transforms continuous-time state trajectories, $\bar{x}(t)$, into a sequence of plant symbols, $\tilde{x}[n]$. The interface generator, $\mathcal{G}$, for hybrid system $\mathcal{H}$ is the ordered 4-tuple, $\mathcal{G} = (\tilde{X}, \mathcal{B}, \alpha, T_x)$ where

- $\tilde{X}$ is a finite alphabet of plant symbols.
- $\mathcal{B}$ is called the interface's event basis. It is a family of p continuously differentiable functionals, $b_i : \Re^n \rightarrow \Re$ over the state space such that $\nabla_{\bar{x}} b_i(\bar{x}) \neq 0$ when $b_i(\bar{x}) = 0$ and such that these functionals separate the state space.
- $\alpha : \Re^n \rightarrow \tilde{X}$ is a mapping from the CSS plant state onto the DES plant symbol.
- and $T_x : \Re \times \tilde{X} \times \Re \rightarrow \Re$ is a recursive mapping used to generate a sequence of times, $\tau_x[n]$, called plant instants, representing the times (measured with respect to the CSS plant's clock) when a plant symbol is issued by the generator. In particular, T_x maps the nth plant instant, $\tau_x[n]$, conditioned on the CSS plant's state trajectory, $\bar{x}(t)$, onto the $n+1$dt plant instant, $\tau_x[n+1] = T_x(\tau_x[n], \bar{x}(t))$.

The interface generator's output is obtained from the following equation,

$$\tilde{x}[n] = \alpha(\bar{x}(\tau_x[n])) \tag{6}$$

The *actuator* converts a sequence $\tilde{r}[n]$ of control symbols into a plant input, $\bar{r}(t)$. The interface actuator, $\mathcal{A}$, for the hybrid system $\mathcal{H}$, is the ordered tuple, $\mathcal{A} = (\bar{R}, \tilde{R}, \gamma, T_r)$, where $\gamma : \tilde{R} \rightarrow \bar{R}$ is a mapping from the control symbols to the constant reference vectors input to the plant, $\bar{R}$ and $\tilde{R}$ are the control (reference) spaces and alphabets, respectively. $T_r : \Re \rightarrow \Re$ is a recursive function generating a sequence of control instants, $\tau_r[n]$. The actuator output is therefore represented as

$$\bar{r}(t) = \sum_{n=0}^{\infty} \gamma(\tilde{r}[n]) I(t, \tau_r[n], \tau_r[n+1]) \tag{7}$$

where $I(t)$ is an indicator function which is unity over the interval $(\tau_r[n], \tau_r[n+1]]$ and is zero elsewhere. It is assumed that the sequence of control and plant instants obey the following relation.

$$\tau_x[n] < \tau_r[n] < \tau_x[n+1] \tag{8}$$

3 Semantics

In a hybrid control system, the plant and interface behave like a discrete event system. The combined plant/interface system is referred to as the *DES plant*. The inputs to the DES plant are the sequence of *control events* $(\tilde{r}[n], \tau_r[n])$. The outputs of the DES plant are the sequence of *plant events* $(\tilde{x}[n], \tau_x[n])$. Note that a *hybrid system event* is an ordered pair consisting of a symbol and the associated time when that symbol was generated or issued.

Definition 1 *A* DES plant *of the hybrid control system $\mathcal{H}$ is the labeled digraph, $\mathcal{P}_d = (\tilde{Z}, A)$. The set of vertices, $\tilde{Z}$, is a finite alphabet of DES plant* state symbols. *The set of arcs, A is a subset of $\tilde{Z} \times \tilde{Z}$. Each arc is labeled by a* control event, $(\tilde{r}, \tau_r) \in \tilde{R} \times \Re$ *and a* plant event $(\tilde{x}, \tau_x) \in \tilde{X} \times \Re$.

The "semantic" meaning of the DES plant refers to the underlying entities which $\mathcal{P}_d$'s vertices and transitions are intended to represent. A concrete semantic interpretation can be constructed once specific choices are made for the event basis, $\mathcal{B}$, and the interface mappings, α and γ. In this paper, the event basis will be assumed to generate an open covering of the CSS plant's state space. Let $\mathbf{b}_i^+$ denote the subset of the state space for which $b_i(\bar{x}) > 0$.

Authorized licensed use limited to: UNIVERSITY NOTRE DAME. Downloaded on October 7, 2009 at 21:04 from IEEE Xplore. Restrictions apply.

M. D. Lemmon and P. J. Antsaklis, "A Computationally Efficient Framework for Hybrid Controller Design," P roc o f t he 1 994 I EEE/IFAC J oint S ymposium o n C o mputer-Aided C ontrol S ystem D esig n , pp 333-338, Tucson, AZ, March 7-9, 1994.

Let the set $\mathbf{b}_i^-$ denote the subset of the state space for which $b_i(\bar{x}) < 0$. This collection of sets, $\mathbf{b}_i^{\pm}$, associated with $\mathcal{B}$ are usually assumed to form an open cover of the CSS plant's state space. Define the following symbols,

$$\tilde{b}_i^+ = \text{EnterRegion}(\mathbf{b}_i^+) \quad (9)$$

$$\tilde{b}_i^- = \text{EnterRegion}(\mathbf{b}_i^-) \quad (10)$$

$$\tilde{\lambda} = \text{NULL} \quad (11)$$

The set $\mathbf{b}_i^{\pm}$, will be said to be the *preimage* of plant symbol $\tilde{b}_i^{\pm}$.

The plant symbol alphabet, $\tilde{X}$, used in this paper will be defined with respect to the event basis. Let

$$\tilde{X} = \left\{\tilde{b}_i^+\right\}_{i=1,\ldots,p} \cup \left\{\tilde{b}_i^-\right\}_{i=1,\ldots,p} \cup \left\{\tilde{\lambda}\right\}, \quad (12)$$

where $\tilde{\lambda}$ represents a "null" symbol. The generator mapping α is also defined with respect to the assumed event basis. It is assumed that this mapping will be used to generate plant symbols only when the plant state crosses the boundary of some $\mathbf{b}_i^{\pm}$ set. The generator mapping α, therefore takes the following form

$$\alpha(\bar{x}) = \begin{cases} \tilde{b}_i^+ & \text{if } b_i(\bar{x}) = 0 \text{ and } \nabla_x b_i(\bar{x})\dot{\bar{x}}(t) > 0 \\ \tilde{b}_i^- & \text{if } b_i(\bar{x}) = 0 \text{ and } \nabla_x b_i(\bar{x})\dot{\bar{x}}(t) < 0 \\ \tilde{\lambda} & \textit{otherwise} \end{cases} \quad (13)$$

The plant instant generator T_x is a recursive function generating a sequence of *plant instant* times, $\tau_x[n]$. In this paper, the plant instants represent times when the boundary of sets $\mathbf{b}_i^{\pm}$ are crossed by the CSS plant's state vector.

With the preceding definitions, the plant events labeling the DES plant arcs represent the symbols issued by the generator when the CSS plant state crosses the boundary of the symbol's pre-image. In this regard, the semantic interpretation of hybrid system plant symbols is grounded in the CSS plant state's entry into one of the elements, $\mathbf{b}_i$, of the event basis.

The DES plant states, $\tilde{Z}$ are defined in terms of the plant symbols, $\tilde{X}$. Let

$$\mathcal{Z} = \left(\tilde{X}, [,], \sim, \vee\right) \quad (14)$$

be a logical system in which the plant symbols, $\tilde{X}$, form the propositional variables and the other symbols ([,],$\sim$,and $\vee$) are used to form propositional logical formulae from the symbols in $\tilde{X}$. With these definitions, the DES plant's state space, $\tilde{Z}$ will be taken to be any collection of wffs for the propositional logical system $\mathcal{Z}$.

The DES controller's state symbols, $\tilde{z}$ can also be given a semantic interpretation grounded in subsets of the CSS plant's state space. Let $\tilde{x}$ and $\tilde{y}$ be elements of $\tilde{X}$ with pre-images $\mathbf{x}$ and $\mathbf{y}$, respectively. Then the following definitions allow us to formulate the preimage of any plant state.

- the preimage of $\sim \tilde{x}$ is the opening of $\bar{X} - \mathbf{x}$.
- the preimage of $[\tilde{x} \vee \tilde{y}]$ is $\mathbf{x} \cup \mathbf{y}$.

Note that with the preceding definition of the DES plant's state symbol, $\tilde{z}$, the symbol has a semantic interpretation that the CSS plant's state is in the preimage of that symbol.

4 Transition Stability

An important issue in hybrid systems concerns the validity of the DES plant. The DES plant can be seen as a "logical" interpretation of the CSS plant's symbolic behaviour. If that interpretation accurately and reliably predicts the CSS plant's behaviour, then it is a valid interpretation or *model* of the CSS plant. As one approach [7] to hybrid control system design involves synthesizing DES controllers for an extracted DES plant, the "validity" of the DES plant is a crucial question which has to be answered.

DES plant validity can be viewed in terms of the invariance of the plant and control event sequence to small perturbations in the CSS plant's state. An arc of the DES plant represents a transition of the CSS plant's state between two subsets of the CSS plant's state space. The labeling of that arc represents the symbolic behaviour of that transition. A valid DES plant would preserve that labeling under small perturbations of the initial CSS plant state. In particular, this means that the plant symbols generated by the transition must be unchanged (invariant) with respect to transition perturbations. In addition, this means that the plant instants, must vary in a smooth (continuous) manner with the assumed perturbations. This notion of "validity" is closely related to the Lyapunov stability of the underlying CSS plant and hence the question of DES plant validity can be viewed as a question about the hybrid system's *stability*.

The preceding discussion can be formalized in the following definition of hybrid system transition or T-stability. This definition is a formalization of earlier notions of "supervisability" presented in [2]. Related definitions of hybrid system stability will be found in [4].

Authorized licensed use limited to: UNIVERSITY NOTRE DAME. Downloaded on October 7, 2009 at 21:04 from IEEE Xplore. Restrictions apply.

M. D. Lemmon and P. J. Antsaklis, "A Computationally Efficient Framework for Hybrid Controller Design," P roc o f t he 1 994 I EEE/IFAC J oint S ymposium o n C o mputer-Aided C ontrol S ystem D esig n , pp 333-338, Tucson, AZ, March 7-9, 1994.

Definition 2 *Let $\mathcal{P}_d = (V, A)$ be a DES plant for the hybrid system $(\mathcal{P}_c, \mathcal{C}_d, \mathcal{A}, \mathcal{G})$. Consider an arc $(\tilde{u}, \tilde{w}) \in A$ which is labeled with plant event, $(\tilde{x}, \tau_x)$, and control event, $(\tilde{r}, \tau_r)$. Let* $\mathbf{u}$ *and* $\mathbf{w}$ *be the pre-images of the DES plant states, $\tilde{u}$ and $\tilde{w}$, respectively. Let $\bar{r} = \gamma(\tilde{r})$ where γ is the hybrid system's actuator mapping.*

*The arc $(\tilde{u}, \tilde{w})$ is transition or T-*stable *if and only if for all $\bar{x}_u \in \mathbf{u}$ there exists an open neighborhood of size ϵ, $N(\bar{x}_u, \epsilon)$, and a finite time $0 < T < \infty$ such that*

- *$\Phi_T^r(\bar{x})$ is an open neighborhood in* $\mathbf{w}$
- *and the plant symbols, $\tilde{x}$, issued by the transition are identical*

for all $\bar{x} \in N(\bar{x}_u, \epsilon)$.

When the DES plant is T-stable, it becomes a *model* or *valid interpretation* for the CSS plant. With such a model it is then possible to supervise the CSS plant by supervising its DES plant. Such a controller can be synthesized using existing methods for DES controller synthesis. In this approach to hybrid control system design, it is critical that the DES plant be T-stable.

For the case when the controls enter linearly, the following proposition has been shown to provide a sufficient condition for T-stability [2]. Proof of this will be found in [3].

Proposition 1 *Consider a DES plant, $\mathcal{P}_d = (V, A)$ of hybrid system whose CSS plant's transition operators are generated by the differential equation*

$$\dot{\bar{x}} = f_0(\bar{x}) + \sum_{i=1}^{m} r_i f_i(\bar{x}) \tag{15}$$

where r_i is the ith component of control vector $\bar{r}$ and $f_i : \Re^n \rightarrow \Re^n$ are a family of Lipschitz continous functions.

Consider an arc $(\tilde{u}, \tilde{w})$ of the DES plant. Let the $\mathbf{w}$ *be the pre-image of $\tilde{w}$. Let* $\mathbf{v}$ *denote the pre-image of any DES plant state symbol, $\tilde{v}$ which is not equal to $\tilde{w}$. This arc will be T-stable if there exists a continuously differentiable positive definite functional $V_{\mathbf{w}} : \Re^n \rightarrow \Re$ which is zero on a closed proper subset of* $\mathbf{w}$ *such that for all $\bar{x} \in \Re^n$,*

$$\begin{pmatrix} L_0 V_{\mathbf{w}} & L_1 V_{\mathbf{w}} & \cdots & L_m V_{\mathbf{w}} \end{pmatrix} \begin{pmatrix} 1 \\ r_1 \\ \cdot \\ r_m \end{pmatrix} < 0 \tag{16}$$

and there exists a continuously differentiable positive definite functional $V_{\mathbf{v}} : \Re^n \rightarrow \Re$ such that for all $\bar{x}$ in $\mathbf{v}$,

$$\begin{pmatrix} L_0 V_{\mathbf{v}} & L_1 V_{\mathbf{v}} & \cdots & L_m V_{\mathbf{v}} \end{pmatrix} \begin{pmatrix} 1 \\ r_1 \\ \cdot \\ r_m \end{pmatrix} > 0 \tag{17}$$

where $L_i V$ is the Lie derivative of V with respect to the ith vector field, f_i.

5 DES Controller Design

The preceding section derived sufficient conditions for T-stability. In special cases where either the plant observations are linear combinations of plant states or the plant control policies enter linearly, then proposition 1 provides a set linear inequality constraints characterizing hybrid system T-stability. The solution of the inequality system will be an interface yielding a T-stable DES plant. The results of this paper therefore provide a methodology for designing interfaces which insure that the DES plant will be a valid model of the CSS plant.

For those special cases where the sufficient conditions form linear inequality systems, there is the additional advantage gained by having a variety of efficient computational methods for solving these systems. The method of centers computes a sequence of convex bodies and their centers in such a way that the computed centers converge to the feasible point. Depending on the analytic form of the convex bodies and the centers, different types of algorithms are obtained. Examples of such algorithms are the so-called ellipsoid method and interior point methods based on logarithmic barrier functions. Both of these algorithms converge after a finite number of updates and both algorithms have polynomial complexity. Therefore, not only does the inequality system allow the design of a T-stable interface, it provides a computationally efficient method which may be practical for DES plants with a large number of logical states.

The preceding discussion suggests that the use of convex programming algorithms such as the ellipsoid method can efficiently decide whether or not an arc of a given plant DES is T-stable. The plant DES, of course, is not unique. Therefore, if we are given a plant DES which represents the desired logical behaviour of the plant, the preceding algorithm can be used to decide whether or not an arc of the specified plant DES is T-stable. By repeatedly applying the algorithm to each arc of the specified plant DES, it

Authorized licensed use limited to: UNIVERSITY NOTRE DAME. Downloaded on October 7, 2009 at 21:04 from IEEE Xplore. Restrictions apply.

M. D. Lemmon and P. J. Antsaklis, "A Computationally Efficient Framework for Hybrid Controller Design," P roc o f t he 1 994 I EEE/IFAC J oint S ymposium o n C o mputer-Aided C ontrol S ystem D esig n , pp 333-338, Tucson, AZ, March 7-9, 1994.

then becomes possible to decide whether or not the specified behaviour is T-stable.

Consider for example a plant DES $\mathcal{P}_d^*$ which generates a language K^*. Assume that K^* is the desired logical behaviour which we want our hybrid control system to possess. The ellipsoidal algorithm test can be used to decide in finite time whether or not a given arc $\mathcal{P}_d^*$ is indeed T-stable. We can therefore construct a sequence of plant DES by successively testing all the arcs $\mathcal{P}_d^*$. As each T-stable arc is identified, we build up an ordered sequence of plant DES

$$\mathcal{P}_d^{(1)}, \mathcal{P}_d^{(2)}, \ldots, \mathcal{P}_d^{(i)} \tag{18}$$

whose associated languages are subsets of the specified language K^*. In this sense, therefore, the languages, $K^{(i)}$, generated by this sequence of DES plants represent better and better approximations of K^* which are guaranteed of being T-stable. In other words, we obtain the following sequence,

$$K^{(1)} \subset K^{(2)} \subset K^{(3)} \subset \ldots \subset K^{(3)} \subset K^* \tag{19}$$

The largest $K^{(i)}$ whose language is contained within K^* forms the supremal controllable sublanguage of the K^*. If it turns out that $\mathcal{P}_d^*$ is a T-stable plant DES, then the supremal controllable sublanguage is K^*. Therefore the plant DES, $\mathcal{P}_d^*$ can be used as the basis for the DES controller of the hybrid system. If $K^{(i)}$ turns out to be smaller than K^*, then this means that our formal specification cannot be stably realized by the hybrid system. We must either settle for the behaviours realized by $\mathcal{P}_d^{(i)}$ or else, we must modify the plant and interface until the formal specification can be achieved. Basically, the procedure outlined above represents a way of constructing T-stable DES controllers for the hybrid system. The final plant DES obtained in this manner, represents a modification of the original specification, K^*, which is guaranteed of being a logically stable interpretation of the continuous-state plant's behaviour.

6 Summary

This paper has outlined a method for hybrid control system design which uses convex programming methods like the ellipsoid algorithm to decide, in finite time, whether or not a specified logical behaviour can be implemented in a "stable" manner by a DES controller. A formal definition for this notion of "logical" stability was introduced and sufficient conditions were presented guaranteeing T-stability. The proposed method for constructing the stable plant DES which most closely approximates the desired logical behaviour is interpreted as a means of modifying original operator specifications so that the resulting set of specifications are consistent with this notion of logical stability. The proposed approach represents an efficient method for determining hybrid control systems.

References

[1] P.J. Antsaklis, J.A. Stiver, and M.D. Lemmon,"Hybrid System Modeling and Autonomous Control Systems", in *Hybrid Systems*, Lecture Notes in Computer Science, Vol 736, R.L. Grossman, Anil Nerode, Anders Ravn, and Hans Rischel (editors), pp. 366-392, Springer Verlag, Berlin, 1993.

[2] M.D. Lemmon and P.J. Antsaklis (1993) "Hybrid Systems and Intelligent Control", Procedings of the *International Symposium on Intelligent Control*, Vol 1:174-179, Aug 26-28, 1993, Chicago, Illinois

[3] M.D. Lemmon and P.J. Antsaklis (1993), *On the Stability of Logical Transitions in Hybrid Control Systems*, Technical Report ISIS-93-009, Dept. of Electrical Engineering, University of Notre Dame, Notre Dame, IN 46556, December 1993.

[4] A. Nerode and W. Kohn (1993), "Models for Hybrid Systems: Automata, Topologies, Stability", in *Hybrid Systems*, Lecture Notes in Computer Science, Vol 736, R.L. Grossman, Anil Nerode, Anders Ravn, and Hans Rischel (editors), pp. 317-356, Springer Verlag, Berlin, 1993.

[5] P. Ramadge and W.M. Wonham (1989), "The control of discrete event systems", *Proceedings of the IEEE*, vol 77, no. 1, pp. 81-89, Jan. 1989

[6] J.A. Stiver and P.J. Antsaklis (1991), "A novel discrete event system approach to modeling and analysis of hybrid control systems", In *Proceedings of the Twenty-Ninth Annual Allerton Conference on Communications, Control, and Computing*, University of Illionois at Urbana-Champaign, Oct. 1991.

[7] J.A. Stiver and P.J. Antsaklis (1993), "DES Supervisor Design for Hybrid Control Systems", *Proc. of the Thirty-First Annual Allerton Conference on Communication Control and Computing*, Univ. of Illinois at Urbana-Champaign, Sept 29-Oct. 1 1993, To appear

Authorized licensed use limited to: UNIVERSITY NOTRE DAME. Downloaded on October 7, 2009 at 21:04 from IEEE Xplore. Restrictions apply.