Proceedings of the 31st Conference
on Decision and Control
Tucson, Arizona • December 1992

# MODELING AND ANALYSIS OF HYBRID CONTROL SYSTEMS

James. A. Stiver
Dept. of Electrical Engineering
University of Notre Dame
Notre Dame, IN 46556

Panos J. Antsaklis
Dept. of Electrical Engineering
University of Notre Dame
Notre Dame, IN 46556

## ABSTRACT

A hybrid control system contains both continuous-time and discrete event components. Specifically, the plant is a continuous-time system modeled by differential equations, and the controller is a discrete event system modeled by an automaton. This paper presents a framework for modeling hybrid control systems including the necessary interface between the plant and controller. A method to represent the entire system as a discrete event system is shown, and the concept of determinism is used to analyze hybrid control system behavior and guide in hybrid control system design.

## 1. INTRODUCTION

In this paper, a Hybrid Control System consists of a conventional system which is being controlled by a Discrete Event System (DES). By conventional system, we mean a continuous-time system described by a set of differential equations[1]. Note that the use of the term "hybrid" is distinct from another common use in the control field to refer to systems with both analog and digital components. A common example of a hybrid control system is the heating and cooling system of a typical home. Here the furnace and air conditioner can be modeled as continuous-time systems which are being controlled by a discrete event system; the thermostat. Recently attempts have been made to study hybrid control systems in a unified, analytical way [2, 6, 7, 10, 11].

In this work, we first present a flexible and tractable way of modeling hybrid control systems. Our aim is to develop a model which can adequately represent a wide variety of hybrid control systems, while remaining simple enough to permit analysis. Second, we present a few methods which can be used to analyze and aid in the design of hybrid control systems. These methods relate to the design of the interface which is a necessary component of a hybrid system and reflects both the dynamics of the plant and the aims of the controller.

## 2. HYBRID CONTROL SYSTEM MODEL

A hybrid control system, can be divided into three parts as shown in Figure 1. This section discusses the models we use for each of these three parts, as well as the way they interact. In the cases of the plant and controller, existing models for continuous-time systems and discrete event systems are used, because such models are well known. There exist no commonly used models for the required interface however, so a new one is developed here.

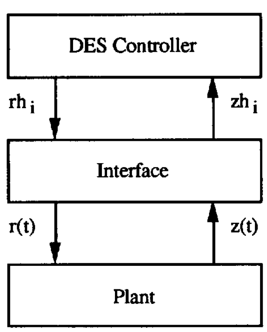


Figure 1: Hybrid Control System

[1] A time-invariant, discrete-time system can also be used.

### 2.1 Plant

The plant is modeled as a time-invariant, continuous-time system. Though called the plant, this part of the hybrid control system is intended to contain the entire continuous-time portion of the system, possibly including a continuous-time controller. With respect to the entire system however, it is the plant. Mathematically, the plant is represented by the familiar equations

$$\dot{x}(t) = f(x(t), r(t)) \tag{1}$$
$$z(t) = g(x(t)) \tag{2}$$

where x(t), r(t), and z(t) are the state, input, and output respectively. For the purposes of this work we assume that $z(t) = x(t)$. Note that the plant input and output are continuous-time signals.

### 2.2 Controller

The controller is a discrete event system which we model as a deterministic automaton. This automaton can be specified by a quintuple, $\{S, E, C, \delta, \phi\}$, where S is the (possibly infinite) set of states, E is the set of *plant events*, C is the set of *controller events*, $\delta : S \times E \rightarrow S$ is the state transition function, and $\phi : S \rightarrow C$ is the output function. The events in set C are called controller events because they are generated by the controller. Likewise, the events in set E are generated by conditions in the plant. The action of the controller can be described by the equations

$$t_i = \delta(t_{i-1}, zh_i) \tag{3}$$
$$rh_i = \phi(t_i) \tag{4}$$

where $t_i \in S$, $zh_i \in E$, and $rh_i \in C$. The index i is analogous to a time index in that it specifies the order of the states or events in a sequence. The input and output signals associated with the controller are asynchronous sequences of events, rather than continuous-time signals. Notice that there is no delay in the controller. The state transition, from $t_{i-1}$ to $t_i$, and the controller event, $rh_i$, are generated immediately after the plant event $zh_i$ occurs.

### 2.3 Interface

The controller and plant cannot communicate directly in a hybrid control system because they each utilize a different type of signal. Thus an interface is required which can convert continuous-time signals to sequences of events and vice versa. The interface consists of two memoryless maps, $\gamma$ and $\alpha$. The first map, $\gamma : C \rightarrow \mathbf{R}^m$, converts each controller event to a constant plant input as follows

$$r(t) = \gamma(rh_i) \tag{5}$$

where $rh_i$ is the most recent controller event, previous to time t. The plant input, r(t), can only have certain values, where each value is associated with a particular controller event. Thus the plant input is a piecewise constant signal which may change only when a controller event occurs.

The second map, $\alpha : \mathbf{R}^n \rightarrow E$, is a function which maps the state space of the plant to the set of plant events.

$$zh_i = \alpha(x(t)) \tag{6}$$

It would appear from equation (6) that, as x changes, zh also continuously changes. That is, there is a continuous generation of plant events by the interface because each state is mapped to an event. This is not the case due to two reasons. First, $\alpha$ must be a function which induces equivalence classes on $\mathbf{R}^n$, which form contiguous regions. Second, a plant event is generated only when the state first enters one of these regions. The overall effect is that the state space of the plant is partitioned into a number of regions and each is associated with a unique plant event which is generated whenever the state enters that region. For example, if $x(t) \in \mathbf{R}^1$, $\alpha$ could map all positive x to one event and all negative x to a second event. As this system operated a plant event would be generated whenever x changed sign.

*2.4 DES Plant*

If the plant and interface of a hybrid control system are viewed as a single component, this component behaves like a discrete event system. It is advantageous to view a hybrid control system this way because it allows it to be modeled as two interacting discrete event systems which are more easily analyzed than the system in its original form. The discrete event system which models the plant and interface is called the *DES Plant* and is modeled as an automaton similar to the controller. The automaton is specified by a quintuple, $\{P, E, C, \psi, \lambda\}$, where P is the set of states, E and C are the sets of plant events and controller events, $\psi : P \times C \to P$ is the state transition function, and $\lambda : P \to E$ is the output function. The behavior of the DES plant is as follows

$$q_{i+1} = \psi(q_i, rh_i) \tag{7}$$

$$zh_i = \lambda(q_i) \tag{8}$$

where $q_i \in P$, $rh_i \in C$, and $zh_i \in E$. There are two differences between the DES plant and the controller. First, as can be seen from equation (7), the state transitions in the DES plant do not occur immediately when a controller event occurs. This is in contrast to the controller where state transitions occur immediately with the occurrence of a plant event. The second difference is that the automaton which models the DES plant may be non-deterministic, meaning $q_{i+1}$ in equation (7) is not determined exactly but rather is limited to a subset of P. The reason for these differences is that the DES plant is a simplification of a continuous-time plant and an interface. This simplification results in a loss of information about the internal dynamics, leading to non-deterministic behavior.

The set of states, P, of the DES plant is based on the set of equivalence classes of $\alpha$. Specifically, each state in P corresponds to a region, in the state space of the continuous-time plant, which is equivalent under $\alpha$. Thus there is a one-to-one correspondence between the set of states, P, and the set of plant events, E. This can be used to develop an expression for the state transition function, $\psi$. Starting with the continuous-time plant, we integrate equation (1) to get the state after a time t, under constant input $r(t) = \gamma(c_k)$

$$x(t) = F_k(x_0, t) \tag{9}$$

Here $x_0$ is the initial state, t is the elapsed time, and $c_k \in C$. $F_k(x_0, t)$ is obtained by integrating $f(x(t), r(t))$, with $r(t) = \gamma(c_k)$. Next we define

$$\hat{F}_k(x_0) = F_k(x_0, t), \text{ where } t = \min_t \{t \mid \alpha(F(x_0, t)) \neq \alpha(x_0)\} \tag{10}$$

Equation (10) gives the state, x(t), where it will first cross into a new region. Now the dynamics of the DES plant can be derived from equations (6, 7, and 8).

$$zh_{i+1} = \lambda(\psi(q_i, rh_i)) \tag{11}$$

$$zh_{i+1} = \alpha(\hat{F}_k(x_0)) \tag{12}$$

$$\psi(q_i, rh_i) = \lambda^{-1}(\alpha(\hat{F}_k(x_0))) \tag{13}$$

where $rh_i = c_k$ and $x_0 \in \{x \mid \alpha(x) = \lambda(q_i)\}$. As can be seen, the only uncertainty in equation (13) is the value of $x_0$. $x_0$ is the state of the continuous-time plant at the time of the last plant event, $zh_i$, i.e. the time that the DES plant entered state $q_i$. It is only known to within an equivalence class of $\alpha$. The condition for a deterministic DES plant is that the state transition function, $\psi$, must be invariant to this uncertainty.

Theorem 1: The DES plant will be deterministic iff given any $q_i \in P$ and $c_k \in C$, there exists $q_{i+1} \in P$ such that for every $x_0 \in \{x \mid \alpha(x) = \lambda(q_i)\}$ we have $\alpha(\hat{F}_k(x_0)) = \lambda(q_{i+1})$.

*Proof*

To prove that the theorem is sufficient, notice that the set $\{x \mid \alpha(x) = \lambda(q_i)\}$ represents the set of all states, x, in the continuous-time plant which could give rise to the state $q_i$ in the DES plant. The theorem guarantees that the subsequent DES plant state, $q_{i+1}$, is unique for a given input and thus the DES plant is deterministic.

To prove that the theorem is necessary, assume that it does not hold. There must then exist a $q_i \in P$ and $c_k \in C$, such that no $q_{i+1}$ exists to satisfy the condition: $\alpha(\hat{F}_k(x_0)) = \lambda(q_{i+1})$ for every $x_0 \in \{x \mid \alpha(x) = \lambda(q_i)\}$. This is not a deterministic system because there is uncertainty in the state transition for at least one state and input. □

Theorem 1 states that the DES plant will be deterministic if all the state trajectories in the continuous-time plant, which start in the same region and are driven by the same input, move to the same subsequent region.

## 3. PARTITIONING

A particular problem in the design of a hybrid control system is the selection of the function $\alpha$, which partitions the state-space of the plant into various regions. Since this partition is used to generate the plant events, it must be chosen to provide sufficient information to the controller to allow control without being so fine that it leads to an unmanageably complex system or simply degenerates the system into an essentially conventional control system.

The partition must accomplish two goals. First it must give the controller sufficient information to determine whether or not the current state is in an acceptable region. For example, in an aircraft these regions may correspond to climbing, diving, turning right, etc. Second, the partition must provide enough additional information about the state, to enable the controller to drive the plant to an acceptable region. In an aircraft, for instance, the input required to cause the plane to climb may vary depending on the current state of the plane. So to summarize, the partition must be detailed enough to answer: 1) is the current state acceptable; and 2) which input can be applied to drive the state to an acceptable region.

To design a partition, we can start by designing a primary partition to meet the first goal mentioned above. This primary partition will identify all the desired operating regions of the plant state space, so its design will be dictated by the control goals. The final partition will represent a refinement of the primary partition which enables the controller to regulate the plant to any of the desired operating regions, thus meeting the second goal.

An obvious choice for the final partition is one which makes the DES plant deterministic and therefore guarantees that the controller will have full information about the behavior of the plant. In addition to being very hard to meet, this requirement is overly strict because the controller only needs to regulate the plant to the regions in the primary partition, not the final partition. For this reason we define quasideterminism, a weaker form of determinism. In the DES plant, the states which are in the same region of the primary partition can be grouped together, and if the DES plant is deterministic with respect to these groups, then we say it is quasideterministic. So if the DES plant is quasideterministic, then we may not be able to predict the next state exactly, but we will be able to predict its region of the primary partition and thus whether or not it is acceptable.

Definition 1: The DES plant will be quasideterministic iff given any $q_i \in P$ and $c_k \in C$, there exists $Q \subset P$ such that for every $x_0 \in \{x \mid \alpha(x) = \lambda(q_i)\}$ we have $\alpha_p(\hat{F}_k(x_0)) = \lambda_p(q_{i+1})$ where $q_{i+1} \in Q$ and $\lambda_p(q)$ is the same for all $q \in Q$. □

The functions $\alpha_p$ and $\lambda_p$ are analogous to $\alpha$ and $\lambda$ but apply to the primary partition. They are useful for comparing states but they are never implemented and their actual values are irrelevant. For example, if $\alpha_p(x(1)) = \alpha_p(x(2))$, then x(1) and x(2) are in the same region of the primary partition. Or, if $\alpha_p(x(1)) = \lambda_p(q_1)$, then x(1) is in the same region of the primary partition as $q_1$ in the DES plant. When used with $\alpha_p$ we define $\hat{F}$ as

$$\hat{F}_k(x_0) = F_k(x_0, t), \text{ where } t = \min_t \{t \mid \alpha_p(F(x_0, t)) \neq \alpha_p(x_0)\} \quad (14)$$

We would like to find the coarsest partition which meets the conditions of Definition 1 for a given primary partition. Such a partition is formed when the equivalence classes of $\alpha$ are as follows,

$$E[\alpha] = \inf\{E[\alpha_p], E[\alpha_p \circ \hat{F}_k] \mid c_k \in C\}. \quad (15)$$

Where we use E[•] to denote the equivalence classes of •. The infimum, in this case, means the coarsest partition which is at least as fine as any of the partitions in the set.

Theorem 2: The regions described by equation (15) form the coarsest partition which generates a quasideterministic DES plant.

*Proof*

First we will prove that the partition does, in fact, lead to a quasideterministic system. For any two states, $x_1$ and $x_2$, which are in the same equivalence class of $\alpha$, we apply some control $r(t) = \gamma(c_k)$. The two states will subsequently enter new regions of the primary partition at $\hat{F}_k(x_1)$ and $\hat{F}_k(x_2)$ respectively. The actual regions entered are $\alpha_p(\hat{F}_k(x_1))$ and $\alpha_p(\hat{F}_k(x_2))$. Now according to equation (15), if $x_1$ and $x_2$ are in the same equivalence class of $\alpha$, then they are also in the same equivalence class of $\alpha_p \circ \hat{F}_k$. Therefore $\alpha_p(\hat{F}_k(x_1)) = \alpha_p(\hat{F}_k(x_2))$ and the system is deterministic.

Next we will prove that the partition is as coarse as possible. Assume there is a coarser partition which also generates a quasideterministic system. That is, there exists two states, $x_3$ and $x_4$, in the same region of the primary partition such that $\alpha(x_3) \neq \alpha(x_4)$, but $\alpha_p(\hat{F}_k(x_1)) = \alpha_p(\hat{F}_k(x_2))$ for *any* possible input. These two states would lie in the same equivalence class of $\alpha_p \circ \hat{F}_k$ for all $c_k \in C$ and therefore in the same equivalence class of $\inf\{E[\alpha_p], E[\alpha_p \circ \hat{F}_k] \mid c_k \in C\}$. This violates the assumption that $x_3$ and $x_4$ do not lie in the same equivalence class of $\alpha$, so two such states could not exist and therefore a coarser partition can not exist. □

Note that the partition described in equation (15) and discussed in Theorem 2 is not dependent upon any specific sequence of controller events. It is intended to yield a DES plant which is as 'controllable' as possible, given the continuous-time plant and available inputs. If the specific control goals are know, it may be possible to derive a coarser partition which is still adequate. This can be done in an ad hoc fashion, for instance, by combining regions which are equivalent under the inputs which are anticipated when the plant is in those regions.

## 4. EXAMPLES

*4.1 Surge Tank*

This first example will illustrate how a simple hybrid control system can be modeled. The system consists of a surge tank which is draining through a fixed outlet valve, while the inlet valve is being controlled by a discrete event system. The controller allows the tank to drain to a minimum level and then opens the inlet valve to refill it. When the tank has reached a maximum level, the inlet valve is closed. The surge tank is modeled by a differential equation,

$$\dot{x}(t) = r(t) - [x(t)]^{1/2} \quad (16)$$

where x(t) is the liquid level and r(t) is the inlet flow. The interface partitions the state space into three regions as follows

$$\alpha(x(t)) = \begin{cases} e_1 & x(t) \geq \max \\ e_2 & \min < x(t) < \max \\ e_3 & x(t) \leq \min \end{cases} \quad (17)$$

Thus when the level reaches max, plant event $e_1$ is generated, and when the level falls to min, plant event $e_3$ is generated. The interface provides for two inputs corresponding to the two controller events $c_1$ and $c_2$ as follows

$$\gamma(rh_i) = \begin{cases} 1 & rh_i = c_1 \\ 0 & rh_i = c_2 \end{cases} \quad (18)$$

Since $r(t) = \gamma(rh_i)$, this means the inlet valve will be open following controller event $c_1$, and closed following controller event $c_2$.

The controller for the surge tank is a two state automaton which moves to state $s_1$ whenever $e_3$ is received, moves to state $s_2$ whenever $e_1$ is received and returns to the current state if $e_2$ is received. Furthermore $\phi(s_1) = c_1$ and $\phi(s_2) = c_2$.

*4.2 Double Integrator*

To illustrate the work presented in this paper we give this second example of a hybrid control system. The plant is a double integrator

$$\dot{x}(t) = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} r(t) \quad (19)$$

The general control goal in this system, which motivates the design of the interface, is to move the state of the double integrator between the four quadrants of the state-space. In the interface, the function $\alpha$ partitions the state space into four regions as follows,

$$\alpha(x(t)) = \begin{cases} e_1 & x_1, x_2 \geq 0 \\ e_2 & x_1 < 0, x_2 \geq 0 \\ e_3 & x_1, x_2 < 0 \\ e_4 & x_1 \geq 0, x_2 < 0 \end{cases} \quad (20)$$

and the function $\gamma$ provides three set points,

$$\gamma(rh_i) = \begin{cases} -10 & rh_i = c_1 \\ 0 & rh_i = c_2 \\ 10 & rh_i = c_3 \end{cases} \quad (21)$$

So whenever the state of the double integrator enters quadrant 1, for example, the plant event $e_1$ is generated. When the controller (which is unspecified) generates controller event $c_1$, the double integrator is driven with an input of -10.

Now we know that the DES plant will have four states because there are four regions in the state space of the actual plant. By examining the various state trajectories given by equation (22), we can find the DES plant which is shown in Figure 2. Equation (22) is obtained by integrating equation (19) and adding x(0).

$$x(t) = F_k(x_0, t) = \begin{bmatrix} 1 & t \\ 0 & 1 \end{bmatrix} x(0) + \begin{bmatrix} .5t^2 \\ t \end{bmatrix} \gamma(c_k) \quad (22)$$

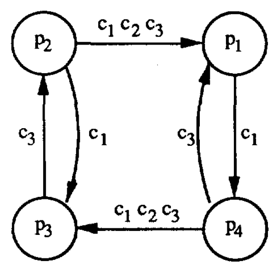


Figure 2: DES Plant of Double Integrator

As can be seen, the DES plant is not deterministic. If we consider $q_i$

$= p_2$ (corresponds to $\lambda(q_i) = e_2$) and $rh_i = c_1$, there exists no unique $q_{i+1}$ thus violating Theorem 1. This presents a problem for the controller because if it generates controller event $c_1$ after receiving plant event $e_2$, the subsequent behavior of the plant is not predictable.

If we want a quasi-deterministic plant, using the four quadrants as the primary partition, Theorem 2 can be used to obtain a new partition. This partition is shown in Figure 3 and the resulting DES plant is shown in Figure 4. The final partition refined the regions in quadrants 2 and 4. Notice that the DES plant is now quasi-deterministic (in fact completely deterministic.) All the transitions are unique.

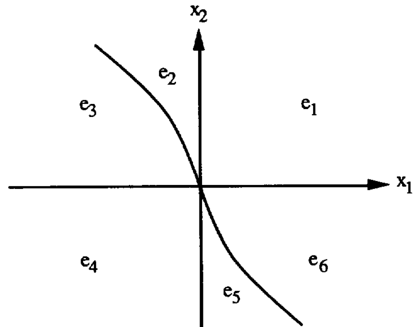


Figure 3: Partition for Quasideterministic System

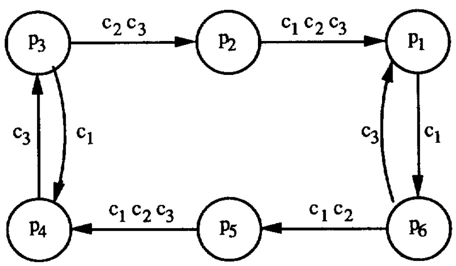


Figure 4: DES Plant of Quasideterministic Double Integrator


## ACKNOWLEDGEMENTS

This work was sponsored in part by the Center for Applied Mathematics, University of Notre Dame, by a fellowship to the first author.


## REFERENCES


[1] Antsaklis, P. J., Passino, K., Wang, S., "Towards Intelligent Autonomous Control Systems: Architecture and Fundamental Issues", *Journal of Intelligent and Robotic Systems*, pp. 315-342, 1989.

[2] Benveniste, A., Le Guernic, P., "Hybrid Dynamical Systems and the SIGNAL Language", *IEEE Transactions on Automatic Control*, Vol. 35, No. 5, pp. 535-546, May 1990.

[3] Cao, X., Ho, Y., "Models of Discrete Event Dynamic Systems", *IEEE Control Systems Magazine*, pp. 69-76, June 1990.

[4] Carroll, J., Long, D., *Theory of Finite Automata*, Prentice-Hall, N.J., 1989.

[5] Cassandras, C., Ramadge, P., "Toward a Control Theory for Discrete Event Systems", *IEEE Control Systems Magazine*, pp. 66-68, June 1990.

[6] Fishwick, P., Zeigler, B., "Creating Qualitative and Combined Models with Discrete Events", *Proceedings of The 2nd Annual Conference on AI, Simulation and Planning in High Autonomy Systems*, pp. 306-315, Cocoa Beach, FL, April 1991.

[7] Gollu, A., Varaiya, P., "Hybrid Dynamical Systems", Proceedings of the 28th Conference on Decision and Control, pp. 2708-2712, Tampa, FL, December 1989.

[8] Passino, K., "Analysis and Synthesis of Discrete Event Regulator Systems", Ph. D. Dissertation, Dept. of Electrical and Computer Engineering, Univ. of Notre Dame, Notre Dame, IN, April 1989.

[9] Passino, K., Michel, A., Antsaklis, P. J., "Lyapunov Stability of a Class of Discrete Event Systems", *Proceedings of the American Control Conference*, Boston MA, June 1991.

[10] Peleties, P., DeCarlo, R., "Modeling of Interacting Continuous Time and Discrete Event Systems : An Example", *Proceedings of the 26th Annual Allerton Conference on Communication, Control, and Computing*, pp. 1150-1159, Univ. of Illinois at Urbana-Champaign, October 1988.

[11] Peleties, P., DeCarlo, R., "A Modeling Strategy with Event Structures for Hybrid Systems", *Proceedings of the 28th Conference on Decision and Control*, pp.1308 - 1313, Tampa FL, December 1989.

[12] Ramadge, P., Wonham, W. M., "The Control of Discrete Event Systems", *Proceedings of the IEEE*, Vol. 77, No. 1, pp. 81 - 98, January 1989.

[13] Sain, M. K., *Introduction to Algebraic System Theory*, Acedemic Press, NY, 1981.

[14] Stiver, J., "Modeling of Hybrid Control Systems Using Discrete Event System Models", M.S. Thesis, Dept. of Electrical Engineering, Univ. of Notre Dame, Notre Dame, IN, May 1991.

[15] Stiver, J., Antsaklis, P., "A Novel Discrete Event System Approach to Modeling and Analysis of Hybrid Control Systems", Control Systems Technical Report #71, Dept. of Electrical Engineering, University of Notre Dame, Notre Dame, IN, June 1991.

[16] Stiver, J., Antsaklis, P., "A Novel Discrete Event System Approach to Modeling and Analysis of Hybrid Control Systems", *Proceedings of the Twenty-Ninth Annual Allerton Conference on Communication, Control, and Computing*, University of Illinois at Urbana-Champaign, Oct. 2-4, 1991.

[17] Zeigler, B., *Multifacetted Modelling and Discrete Event Simulation*, Acedemic Press, NY, 1984.

[18] Zhong, H.,Wonham, W. M., "On the Consistency of Hierarchical Supervision in Discrete-Event Systems", *IEEE Transactions on Automatic Control*, Vol. 35, No. 10, pp. 1125-1134, October 1990.