Proceedings of the
American Control Conference
San Francisco, California • June 1993

State Space Partitioning for Hybrid Control Systems

James A. Stiver and Panos J. Antsaklis
Department of Electrical Engineering
University of Notre Dame, Notre Dame, IN 46556

Abstract

In this paper, the state space partitioning in the interface of the hybrid control system is modeled using covering halfspaces; this formulation extends the model introduced in [1]. The concept of determinism is then defined and linked to controllability and observability.

1 Introduction

A hybrid control system consists of a continuous-time system which is being controlled by a discrete event system. The continuous-time system is referred to as the plant because it is the system under control. The plant is usually termed a "conventional" system as it has a continuous state space and it is described by a set of differential (or difference) equations. The discrete event system, called the controller, has a discrete state space and symbolic input and output. In addition to the plant and controller, there is an interface which provides communication between them. The interface generates symbols for the controller as the state of the plant moves through a partitioned state space. It also converts symbols from the controller into plant inputs. The interface, and in particular the partition which it implements, is very important because it governs the interaction of the controller and plant. The design of the partition will depend upon the control goals and the available control actions, and it relates to the basic question of how much information is required to control a system.

In this work, using the model for hybrid control systems originally described in [1] and [2], we present a mathematical description for the generator in the interface using covering halfspaces. This model is used to examine the adequacy of the interface via the concept of determinism. Determinism is then linked to controllability and observability.

It should be noted that [2] also contains an extended list of references on hybrid control systems which cannot be included here due to lack of space.

2 Hybrid Control System Model

A hybrid control system consists of three parts. The modeling and interactions of these parts are now described.

2.1 Plant

The plant is modeled as a time-invariant, continuous-time system, represented by the familiar equations,

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

where $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{r} \in \mathbb{R}^m$, and $\mathbf{z} \in \mathbb{R}^p$ are the state, input, and output vectors respectively. For the purposes of this work we assume that $\mathbf{z} = \mathbf{x}$.

2.2 Controller

The controller is a discrete event system which is modeled as a deterministic automaton specified by the quintuple, $\{\tilde{S}, \tilde{Z}, \tilde{R}, \delta, \phi\}$, where $\tilde{S}$ is the (possibly infinite) set of states, $\tilde{Z}$ is the set of plant symbols which are generated by events in the plant and make up the controller input, $\tilde{R}$ is the set of controller symbols which constitute the controller's output set, $\delta : \tilde{S} \times \tilde{Z} \to \tilde{S}$ is the state transition function, and $\phi : \tilde{S} \to \tilde{R}$ is the output function. The behavior of the controller is described by

$$\tilde{s}[n] = \delta(\tilde{s}[n-1], \tilde{z}[n]) \tag{3}$$
$$\tilde{r}[n] = \phi(\tilde{s}[n]) \tag{4}$$

where $\tilde{s}[n] \in S, \tilde{z}[n] \in \tilde{Z}$, and $\tilde{r}[n] \in \tilde{R}$. The index n indicates the order of the symbols.

2.3 Interface

The interface of a hybrid control system consists of two memoryless maps which convert between continuous-time signals and symbolic signals. The first map, called the actuator, $\gamma : \tilde{R} \to \mathbb{R}^m$, converts a sequence of controller symbols to a piecewise constant plant input as follows

$$\mathbf{r} = \gamma(\tilde{r}) \tag{5}$$

The plant input, $\mathbf{r}$, can change only when a plant symbol occurs.

The second map, called the generator, $\alpha : \mathbb{R}^n \to \tilde{Z}$, is a function which maps the state space of the plant to the set of plant symbols as follows

$$\tilde{z} = \alpha(\mathbf{x}) \tag{6}$$

The generator is based upon a partition of the state space, where each region of the partition is an open subset of the state space, called an event. Each of these events is associated with a unique plant symbol. A plant symbol is generated only when the state first enters the associated event. The set of events is denoted by, Z, and the the plant symbol associated with event $\mathbf{z}_i \in \mathbf{Z}$ is $\tilde{z}_i$.

The partition is formed by a set of $(n-1)$ dimensional hypersurfaces, which are described by a set of functions, $h_i : \mathbb{R}^n \to \mathbb{R}$. Each function, h_i, divides the state space into two halfspaces, $\{\mathbf{x} : h_i(\mathbf{x}) > 0\}$ and $\{\mathbf{x} : h_i(\mathbf{x}) < 0\}$. Each point in the state space which does not lie on a hypersurface can be associated with a binary vector, where the ith element of the vector is: 0 if $h_i(\mathbf{x}) < 0$ or 1 if $h_i(\mathbf{x}) > 0$. For $\{\mathbf{x} : h_i\mathbf{x} = 0\}$, α does not generate a symbol as described above. These binary vectors make up the set of plant symbols. For example $\tilde{z}_5 = [101]$ would be the plant symbol associated with the event $\mathbf{z}_5 = \{\mathbf{x} : h_1(\mathbf{x}) > 0, h_2(\mathbf{x}) < 0, h_3(\mathbf{x}) > 0\}$.

2.4 DES Plant Model

When the plant and interface of a hybrid control system are viewed as a single system, this system appears as a discrete event system called the DES plant model [1]. Like the controller, the DES plant is modeled as an automaton, $\{\tilde{P}, \tilde{Z}, \tilde{R}, \psi, \lambda\}$, where $\tilde{P}$ is the set of states, $\tilde{Z}$ and $\tilde{R}$ are as before, $\psi : \tilde{P} \times \tilde{R} \rightarrow \tilde{P}$ is the state transition function, and $\lambda : \tilde{P} \rightarrow \tilde{Z}$ is the output function. We are interested in the state transition function of the DES plant which was found in [1] to be

$$\psi(\tilde{p}_i, \tilde{r}_k) = \lambda^{-1}(\alpha(\hat{F}_k(\mathbf{x}_0))) \tag{7}$$

where $\mathbf{x}_0$ is the current state of the plant, $\tilde{r}_k$ is the controller symbol, and $\hat{F}_k(\mathbf{x}_0)$ is the value of the state when it will next cross a boundary. Since this equation depends upon $\mathbf{x}_0$ and $\mathbf{x}_0$ could be anywhere in the event associated with the previous plant symbol, the DES plant will be nondeterministic in general.

3 Determinism

Determinism is the ability to uniquely predict future states based on the current state and input. Unlike conventional systems, the DES plant model is not generally deterministic. The following conditions can be used to test for determinism. Note we have allowed $h_j(\mathbf{z}_n) = h_j(\mathbf{x})$ where $\mathbf{x} \in \mathbf{z}_n$.

Definition 1 *A surface, h_i, is an exit boundary of event $\mathbf{z}_n$ under $\tilde{r}_k$ if $\exists \mathbf{x} \in \{\mathbf{x} : h_i(\mathbf{x}) = 0, h_j(\mathbf{x}) = h_j(\mathbf{z}_n) \forall j \neq i\}$ such that $h_i(\mathbf{z}_n)(f_k(\mathbf{x}) \cdot \nabla h_i(\mathbf{x})) < 0$.*

Definition 2 *A surface, h_i, is an entry boundary of event $\mathbf{z}_n$ under $\tilde{r}_k$ if $\exists \mathbf{x} \in \{\mathbf{x} : h_i(\mathbf{x}) = 0, h_j(\mathbf{x}) = h_j(\mathbf{z}_n) \forall j \neq i\}$ such that $h_i(\mathbf{z}_n)(f_k(\mathbf{x}) \cdot \nabla h_i(\mathbf{x})) > 0$.*

In view of the above definitions, the proof of the following theorem is straight forward.

Theorem 1 *The DES plant model is deterministic if there is no event with more than one exit boundary under any given controller symbol.*

Lack of determinism implies that the system is not fully controllable. Full controllability requires that the system can be driven from one state to another with appropriate inputs, but this cannot be achieved when the system's response to a given input is not predictable. A relaxed form of controllability, in which a system can be driven from one state to a subset of states, is possible as described in [2].

Lack of determinism also renders the notion of observability less useful, as the knowledge of an initial state and input will still not uniquely predict future states. Thus even full observability will not make the system's future behavior known.

4 Examples

4.1 Example 1 - Nondeterministic System

The plant is a double integrator

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \mathbf{r} \tag{8}$$

where $\mathbf{r} \in \{-1, 0, 1\}$ which yields

$$f_1(\mathbf{x}) = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ -1 \end{bmatrix} \tag{9}$$

$$f_2(\mathbf{x}) = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \mathbf{x} \tag{10}$$

$$f_3(\mathbf{x}) = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{11}$$

The events are formed by the following two hypersurfaces

$$h_1(\mathbf{x}) = x_1 \tag{12}$$

$$h_2(\mathbf{x}) = x_2 \tag{13}$$

These two hypersurfaces form four events which are simply the four quadrants. This system can be shown to be nondeterministic by showing that one of the events, corresponding to $\tilde{z} = [01]$, has two exit boundaries under $\tilde{r}_1$.

$$f_1(\mathbf{x}) \cdot \nabla h_1(\mathbf{x}) = x_2 > 0 \tag{14}$$

$$f_1(\mathbf{x}) \cdot \nabla h_2(\mathbf{x}) = -1 < 0 \tag{15}$$

In this case $h_1(\mathbf{z}) < 0$ and $h_2(\mathbf{z}) > 0$ so both h_1 and h_2 are exit boundaries for this event.

4.2 Example 2 - Deterministic System

Using the same plant as in Example 1, we can obtain a deterministic system by replacing the first hypersurface, h_1, by

$$h_1 = x_1 + \frac{x_2^2}{2} \tag{16}$$

With four events, two boundaries per event, and three controller symbols, there are 24 equations which need to be evaluated to assess determinism. Here are the six equations pertaining to the event represented by plant symbol $\tilde{z} = [01]$.

$$f_1(\mathbf{x}) \cdot \nabla h_1(\mathbf{x}) = 0 \tag{17}$$

$$f_1(\mathbf{x}) \cdot \nabla h_2(\mathbf{x}) = -1 < 0 \tag{18}$$

$$f_2(\mathbf{x}) \cdot \nabla h_1(\mathbf{x}) = x_2 < 0 \tag{19}$$

$$f_2(\mathbf{x}) \cdot \nabla h_2(\mathbf{x}) = 0 \tag{20}$$

$$f_3(\mathbf{x}) \cdot \nabla h_1(\mathbf{x}) = 2x_2 > 0 \tag{21}$$

$$f_3(\mathbf{x}) \cdot \nabla h_2(\mathbf{x}) = 1 > 0 \tag{22}$$

As in the previous example, $h_1(\mathbf{z}) < 0$ and $h_2(\mathbf{z}) > 0$, so it can be seen that this event has no more than one exit region under each controller symbol. This is true for the other three events as well, meaning this system is deterministic.

References

[1] J. A. Stiver, P. J. Antsaklis, "Modeling and Analysis of Hybrid Control Systems", *Proceedings of the 31st Conference on Decision and Control*, pp. 3748-3751, Tucson AZ, December 1992.

[2] P. J. Antsaklis, M. D. Lemmon, J. A. Stiver, "Hybrid System Modeling and Event Identification", *Technical Report of the ISIS Group*, ISIS-93-002, University of Notre Dame, Notre Dame IN, January 1993.