Proceedings of the 33rd
Conference on Decision and Control
Lake Buena Vista, FL - December 1994

Digital Control from a Hybrid Perspective

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

Abstract

The framework of hybrid control systems can be used to examine digital control systems. This paper shows how digital control systems can be modeled as hybrid control systems. A technique for designing interfaces for hybrid control systems is presented. An example is employed to demonstrate how this technique, as well as a DES controller design methodology developed earlier, can be applied to improve a digital control system.

1 Introduction

In a digital control system, a continuous-time plant is controlled by a digital computer. The plant output is sampled and quantized to provide the input to the controller, and the control signal is passed through a zero-order hold to provide the plant input. We show that such a digital control system is a special case of a hybrid control system, defined in [1–3]. This more general hybrid framework is used to discuss problems encountered in digital control such as chattering and limit cycles. Section 3 presents a method for designing the interface of a hybrid control system using the natural invariants of the system. Next this method is used as an alternative to the usual quantization technique of digital control. An example illustrates the application of this method to a digital control problem; it also shows the application of DES controller design to digital control.

2 Hybrid Control Systems

A hybrid control system contains a plant, interface and controller as shown in Figure 1. The plant contains the continuous dynamics and the controller contains the discrete, symbolic dynamics. This model for hybrid control systems has appeared earlier in [1–3] among others. After the hybrid control system model has been briefly described below for convenience, it will be used to model digital control systems. Note that the material in Section 3, on interface design using invariants, appears here for the first time.

The plant is a continuous-time system governed by

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

where $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{r} \in \mathbb{R}^m$ are the state and input vectors respectively. $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is a Lipschitz continuous function.

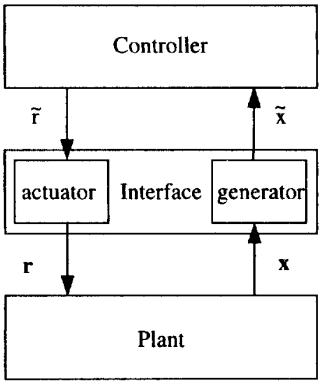

Figure 1: Hybrid Control System

The controller is a discrete event system which is modeled as a deterministic automaton. This automaton is specified by a quintuple, $(\tilde{S}, \tilde{X}, \tilde{R}, \delta, \phi)$, where $\tilde{S}$ is the set of states, $\tilde{X}$ is the set of *plant symbols*, $\tilde{R}$ is the set of *controller symbols*, $\delta : \tilde{S} \times \tilde{X} \to \tilde{S}$ is the state transition function, and $\phi : \tilde{S} \to \tilde{R}$ is the output function. The symbols in set $\tilde{R}$ are called controller symbols because they are generated by the controller. Likewise, the symbols in set $\tilde{X}$ are called plant symbols and are generated based on events in the plant. The action of the controller is described by the equations

$$\tilde{s}[n] = \delta(\tilde{s}[n-1], \tilde{x}[n]) \tag{2}$$

$$\tilde{r}[n] = \phi(\tilde{s}[n]) \tag{3}$$

where $\tilde{s}[n] \in \tilde{S}$, $\tilde{x}[n] \in \tilde{X}$, and $\tilde{r}[n] \in \tilde{R}$.

The controller and plant communicate through an interface. The interface consists of two simple subsystems, the generator and actuator.

The *generator* converts the continuous-time output (state) of the plant to an asynchronous, symbolic input for the the controller. To perform this task, two processes must be in place. The first process is a triggering mechanism determining when a plant symbol should be generated. The second process determines determine which plant symbol is to be issued.

The generator's triggering mechanism is based on the idea of *plant events*. A plant event occurs whenever the

0-7803-1968-0/94$4.00©1994 IEEE

plant state trajectory crosses a hypersurface in a particular direction. These hypersurfaces bound "significant" regions of the continuous-state plant's state space. The generator issues an plant symbol when the plant state crosses one of these hypersurfaces. The set of plant events is therefore given formally as a set of smooth functionals, $\{h_i : \mathbb{R}^n \rightarrow \mathbb{R}\}$. Each functional is assumed to satisfy the condition that $\nabla_x h_i(\mathbf{x}) \neq 0$ for all $\mathbf{x}$ such that $h_i(\mathbf{x}) = 0$. This condition ensures that the functional's null space, $\mathcal{N}(h_i)$ forms an $n-1$ dimensional smooth manifold separating the state space into two n-dimensional halfspaces. When the plant's state enters the open region $\{\mathbf{x} : h_i(\mathbf{x}) < 0\}$, the generator issues the plant symbol associated with that particular hypersurface.

Let $\tau_e[n]$ denote the time (with respect to the plant clock) of the plant state's nth crossing of an event hypersurfaces. At each time in the sequence $\tau_e[n]$, a plant symbol is generated according to the function $\alpha_i : \mathcal{N}(h_i) \rightarrow \tilde{X}$. The sequence of plant symbols can therefore be expressed as

$$\tilde{x}[n] = \alpha_i(\mathbf{x}(\tau_e[n])) \tag{4}$$

where i identifies the hypersurface which was crossed.

The actuator converts the sequence of controller symbols to a plant input signal according to the following formula

$$\mathbf{r}(t) = \sum_{n=0}^{\infty} \gamma(\tilde{r}[n]) I(\tau_c[n], \tau_c[n+1]) \tag{5}$$

where $I(\tau_1, \tau_2)$ is a characteristic function taking on the value of unity over time interval (τ_1, τ_2) and zero elsewhere. The mapping of individual control symbols onto constant reference vectors $\mathbf{r}$ is determined by the function $\gamma : \tilde{R} \rightarrow \mathbb{R}^m$. The sequence of times $\tau_c[n]$ denotes the times when the nth control symbol was issued by the controller. It will be assumed that $\tau_e[n] < \tau_c[n] < \tau_e[n+1]$.

3 Interface Design Using Invariants

There are cases where some of the interface hypersurfaces may be selected by the designer. This section discusses the selection of these hypersurfaces. Assume that the control goal is to transition the plant's state between two disjoint regions of the state space, referred to as the starting region and the target region. The objective is to find a set of hypersurfaces, $\{h_i\}$, so that a transition between these two regions can be achieved. One way to do this is to use hypersurfaces which are dynamical invariants of the control policy, $f(\mathbf{x})$.

The target region, T, is specified as

$$T = \{\mathbf{x} : \forall i \in I_T, h_i(\mathbf{x}) < 0\}, \tag{6}$$

where I_T is the index set indicating which hypersurfaces bound the target region. For a given target region and control policy, find a set of hypersurfaces bounding another open region, B, as follows.

$$B = \{\mathbf{x} : \forall i \in I_B, h_i(\mathbf{x}) < 0, h_e(\mathbf{x}) > 0\}, \tag{7}$$

where I_B is the index set and h_e defines the exit boundary for B. B should be selected to include only states whose trajectories lead to the target region. Figure 2 shows an example of this where $I_T = \{1\}$ and $I_B = \{2, 3\}$.

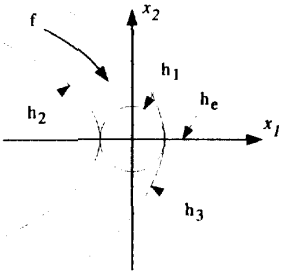

Figure 2: Target Region and Invariants

The following proposition gives sufficient conditions for the hypersurfaces bounding B and T to ensure that all state trajectories in B will reach the target region.

Proposition 1 *Given a set of trajectories described by a smooth vector field, $f(\mathbf{x})$, a non null target region, T, and a set of smooth functionals, $\{h_i : i \in I_B\} \cup \{h_e\}$, with the set $B = \{\mathbf{x} : \forall i \in I_B, h_i(\mathbf{x}) < 0, h_e(\mathbf{x}) > 0\}$ non null. Then for any trajectory, $\mathbf{x}$ which solves $\dot{\mathbf{x}} = f(\mathbf{x})$, $\mathbf{x}(t_0) \in B$ ensures that for some finite $t > t_0$, $\mathbf{x}(t) \in T$, if the following three conditions are true.*

- $\forall i \in I_B, \nabla_x h_i(\mathbf{x}) \cdot f(\mathbf{x}) = 0$
- $\exists \epsilon > 0 : \forall \mathbf{x} \in B, \nabla_x h_e(\mathbf{x}) \cdot f(\mathbf{x}) < -\epsilon$
- $\{\mathbf{x} : \forall i \in I_B, h_i(\mathbf{x}) < 0, h_e(\mathbf{x}) = 0\} \subset T$

Proof: The first condition of the the proposition, which can be rewritten as

$$\frac{dh_i(\mathbf{x})}{dt} = 0, \tag{8}$$

precludes the state trajectory crossing any hypersurface indexed by the set I_B, thus ensuring no trajectory in B will leave B except through the remaining boundary. The second condition, which can be rewritten as

$$\frac{dh_e(\mathbf{x})}{dt} < -\epsilon, \tag{9}$$

ensures that within a finite time, $\frac{h_e(\mathbf{x})}{\epsilon}$, the trajectory will cross the exit boundary. The final condition guarantees that any trajectory leaving B through the exit

boundary will be in the target region when it does so. Together these conditions are sufficient to guarantee that state trajectories in B will enter the target region in finite time. □

The problem is to find the set of hypersurfaces bounding B which satisfies Proposition 1. Consider the hypersurfaces defined by $\{h_i : i \in I_B\}$. First, these hypersurfaces must be invariant under the vector field, $f(\mathbf{x})$. This is achieved by choosing the hypersurfaces to be integral manifolds of an $n-1$ dimensional distribution which is invariant under f. An $n-1$ dimensional distribution, $\Delta(\mathbf{x})$, is invariant under f if it satisfies

$$[f(\mathbf{x}), \Delta(\mathbf{x})] \subset \Delta(\mathbf{x}), \tag{10}$$

where the $[f(\mathbf{x}), \Delta(\mathbf{x})]$ indicates the Lie bracket. Of the invariant distributions, those that have integral manifolds as we require, are exactly those which are involutive (according to Frobenius). This means

$$\forall \delta_1(\mathbf{x}), \delta_2(\mathbf{x}) \in \Delta(\mathbf{x}) \Rightarrow [\delta_1(\mathbf{x}), \delta_2(\mathbf{x})] \in \Delta(\mathbf{x}). \tag{11}$$

Therefore by identifying the involutive distributions which are invariant under the vector field, f, we have identified a set of candidate hypersurfaces. For details of these relationships between vector fields and invariant distributions, see [4].

For a single vector field, $f(\mathbf{x})$, there are many invariant hypersurfaces. The set of all invariant hypersurfaces can be found in terms of $n-1$ functionally independent mappings which form the basis for the desired set of functionals, $\{h_i : i \in I_B\}$. This basis is obtained by solving the characteristic system of ordinary differential equations,

$$\frac{dx_1}{f_1(\mathbf{x})} = \frac{dx_2}{f_2(\mathbf{x})} = \cdots = \frac{dx_n}{f_n(\mathbf{x})}, \tag{12}$$

where $f_i(\mathbf{x})$ is the ith element of $f(\mathbf{x})$. The solutions can be written in the form

$$g_1(\mathbf{x}) = c_1, g_2(\mathbf{x}) = c_2, \cdots, g_{n-1}(\mathbf{x}) = c_{n-1} \tag{13}$$

where c_i is a constant of integration [5].

4 Digital Control in a Hybrid Framework

A digital control system is a special case of hybrid control in which the plant, interface, and controller obey certain constraints. Since the events are triggered at regular (or irregular if desired) intervals of time, there must be a clock present in the system. Normally in digital control this clock is not modeled explicitly, but in a hybrid control framework it appears as part of the plant. A convenient way to do this is to use a two state oscillator such as,

$$\dot{\mathbf{x}}_c = \begin{bmatrix} 0 & \omega \\ -\omega & 0 \end{bmatrix} \mathbf{x}_c, \tag{14}$$

where ω is the clock frequency. If required, ω can be made a function of the plant input, $\mathbf{r}$, and then the clock frequency can be changed by the controller. To implement the sampling, hypersurfaces of the type

$$h_i(\mathbf{x}_c) = k_1 x_{c1} + k_2 x_{c2} \tag{15}$$

are used, where k_1 and k_2 are real constants. A single hypersurface with $k_1 = 1$ and $k_2 = 0$ would model the typical case where the sampling rate is given by $\omega/2\pi$. By adding more hypersurfaces each with its own measurement function, α_i, multirate sampling can be modeled.

When modeling digital control, the measurement function(s), α_i, is a quantizer. The value of each state at the sampling instant is truncated or rounded and restricted to the range of values acceptable to the digital controller. For example, a controller implemented on an 8 bit computer might only have the capability to measure a state to 256 possible values. In that case we might have a set of plant symbols given by

$$\tilde{X} = \{0, 1, ..., 255\}, \tag{16}$$

and a measurement function given by

$$\alpha_1(\mathbf{x}(t)) = \text{trunc}(\mathbf{x}(t) + 128) \text{ modulo } 256. \tag{17}$$

Quantization of this type will form a grid-like partition of the state space into regions, each associated with the same plant symbol. Figure 4 shows an example for a 2 dimensional state space.

On the other side of the interface, the actuator models a zero-order hold. Controller symbols, representing quantized plant input values, are converted to piecewise constant plant inputs. In an example akin to the one given above, the set of controller symbols is given by

$$\tilde{R} = \{0, 1, ..., 255\}, \tag{18}$$

and γ is given by

$$\mathbf{r}(\tau_c[n]) = \tilde{r}[n] - 128. \tag{19}$$

Finally, the controller is an automaton which implements the desired control strategy for the digital controller.

4.1 Comments on HCS Modeling Digital Control Systems

When a digital control system is modeled as a hybrid control system, the resulting hybrid control system has the following characteristics. First, the event triggering hypersurfaces are based only on the clock states. This means that they are unaffected by the other states of the plant. Because of this, these states can vary arbitrarily without triggering an event and the sampling will occur

at rates which are not dependent on the state of the plant (except for the state of the clock).

The second characteristic of digital control is that the state measurements (plant symbols) are formed by quantizing the state values in a grid like fashion. That is, each state is quantized individually to create a grid over the state space. Each cell in the grid is an n-dimensional box in the state space.

These two characteristics are actually the same except that one applies to the clock. The first one applies to the state(s) of the clock and is a consequence of the clock based sampling. The second one applies to the remaining plant states as a result of the way states are quantized in the analog to digital converters found in digital control. Together the two characteristics create a grid style partition over the entire state space (including clock states) of the plant.

The designer is free to choose the sampling rate(s) and the quantization level(s) for each state. These choices determine the dimensions of the n-dimensional boxes which fill up the state space. What the designer cannot do is change the basic shape of the boxes which form the partition; they will always be rectangular.

The requirement that a grid-like partition be used is a limitation of digital control, but it does give the advantage of a framework for design. The interface can be designed by selecting quantization levels and sampling rates for each state. Also, continuous-time control laws can be adapted for use in digital controllers via techniques like Euler's approximation or zero-order hold equivalence. The trade-off is that most hybrid control design possibilities cannot be realized in digital control because of the use of a grid like partition.

4.2 Problems Encountered in Digital Control

The combination of sampling and quantized state measurements creates problems in digital control. Among the problems are chattering and limit cycles. These effects can be described from the point of view of hybrid control systems.

Chattering - digital control systems, like hybrid systems in general can suffer from chattering. Chattering occurs along a surface which separates two different quantization levels or regions in the plant state space. If each of the two regions contains a control law which sends the plant state back to the other region, then the state will oscillate back and forth at the sampling rate. Again the culprit is non-deterministic behavior arising from state space partitioning. In this case, a pair of control laws which are probably appropriate for the states at the center of a region are not entirely appropriate for the states lying toward the boundaries of the region.

Limit cycles - a limit cycle occurs when the plant state can not reach the equilibrium point or when the equilibrium point is not associated with the appropriate control law to make it an invariant. The state oscillates about the equilibrium point.

5 Example: Double Integrator

Since the problems encountered in digital control are largely a result of the quantization, it is reasonable to try to solve them by changing the way the quantization is done. The usual quantization forms a grid-like partition of the state space, forming an array of n-dimensional boxes each with its own symbol. The problem is that these quantization levels have no relationship to the state trajectories which flow through them. To reduce this problem, the grid-like partition can be replaced with a partition based on the natural invariants of the system. The following example explores the use of invariants for quantization.

In this example we have a double integrator which we wish to stabilize. We examine three cases. First, the classical case of digital control is shown. Next, the classical quantization is used but the controller is designed using discrete event system techniques developed for hybrid systems. Third, the interface is designed to quantize based on system invariants and the controller is again designed with methods from hybrid control.

In all cases, we start with a double integrator plant.

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \mathbf{r} \tag{20}$$

5.1 Case 1

As an example of classical digital controller design, the above plant will be converted to its discrete time, zero-order hold equivalent. Then a discrete time controller will be designed, and finally the digital controller will be obtained by quantization of the discrete-time controller. The zero-order hold equivalent of the plant follows.

$$\mathbf{x}[k+1] = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \mathbf{x}[k] + \begin{bmatrix} \frac{1}{2}T^2 \\ T \end{bmatrix} \mathbf{r}[k] \tag{21}$$

This discrete time plant can be stabilized by the following feedback controller.

$$\mathbf{r}[k] = \begin{bmatrix} -2 & -3.2 \end{bmatrix} \mathbf{x}[k] \tag{22}$$

With a sampling period, T, of 0.2, the controller moves the poles of the system from $z = 1$ to $z = .66 \pm .32$, leading to a system whose states (should) spiral gently in to the origin.

The final step to obtain the digital controller is to quantize the signals. The value of the plant state is quantized to 64 levels before being fed to the controller. Each state, x_1 and x_2, is quantized to the nearest .25 on the interval $[-1, .75]$ for a total of 64 possible values.

$$\tilde{x}[k] = \begin{bmatrix} .25\text{round}(\mathbf{x}_1[k]/.25) \\ .25\text{round}(\mathbf{x}_2[k]/.25) \end{bmatrix}, \tag{23}$$

The plant input is also quantized to the nearest .25. We now have a digital controller, albeit a crude one, for the plant. The controller will stabilize the plant in the neighborhood of the origin but rather than a gentle spiral, the state chatters in to the origin and then exhibits limits cycles near the origin. A state trajectory for this system is shown in Figure 3.

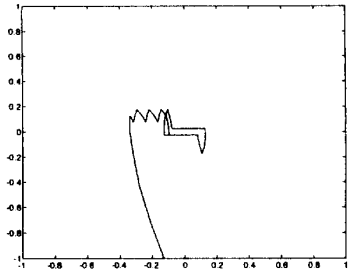

Figure 3: State Trajectory for Case 1

It should not be concluded that this example represents the best that can be done with conventional digital control, it does not. It is used here to illustrate the sorts of problems which are typical in digital control.

5.2 Case 2

In this case the system will be modeled as a hybrid control system so that the controller can be designed using logical DES techniques as described in [3].

The clock appears in the plant as described in Section 4 as a two state periodic system.

$$\dot{\mathbf{x}}_c = \begin{bmatrix} 0 & \frac{2\pi}{.2} \\ -\frac{2\pi}{.2} & 0 \end{bmatrix} \mathbf{x}_c \tag{24}$$

The two states of the clock, along with the states of the double integrator given in equation 20, constitute the augmented state space of the plant in the hybrid control system.

The design of the generator is based on imitating the sampling of a conventional digital control system. A plant event should occur at intervals of the sampling period, $T = \frac{2\pi}{\omega}$. This is accomplished by slipping a single hypersurface into the generator of the system.

$$h_1(\mathbf{x}_c) = x_{c1} \tag{25}$$

Note that a symbol is only produced when the hypersurface is crossed in the negative direction, this avoids generating two symbols per clock period. There are no hypersurfaces associated with the states of the integrator itself, $\mathbf{x}$, because only the clock triggers events. The plant symbols are determined by α.

$$\alpha(\mathbf{x}) = .25\text{round}(4\mathbf{x}[k]) \tag{26}$$

This generator quantizes the plant state in the same way as the the first case and as shown in Figure 4.

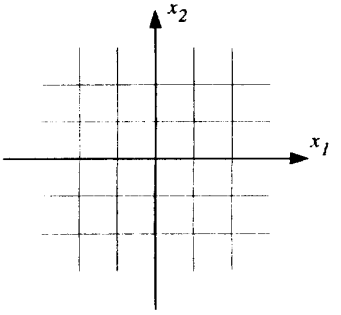

Figure 4: Quantization for Case 2

The actuator of this system will convert the controller symbols into one of three possible inputs to the plant. We use a subset of the input values which were used by the digital controller in the first case of this example, that is

$$\tilde{r}[k] \in \{-2.0, 0.0, 2.0\}. \tag{27}$$

Now to design the controller, we first need to extract the DES plant model. This procedure is described in [3] and results in an automaton which models the double integrator together with the interface. Using the DES plant, a controller is designed via techniques developed for the control of discrete event systems. The controller operates by choosing a control policy which enables only desired transitions. It this case, the desired transitions are those which move the state toward the origin in a roughly clockwise direction (using the orientation of Figure 4). The following equation provides the control policy.

$$\tilde{r}[k] = \begin{cases} -2.0 & \text{if} \quad x_1 > 0, x_2 > 0 \\ 2.0 & \text{if} \quad x_1 \leq 0, x_2 \leq 0 \\ 0.0 & \text{otherwise} \end{cases} \tag{28}$$

Of course, the actual value of the state, $\mathbf{x}(t)$, is unavailable to the controller, so the quantized value, $\tilde{x}[k]$, is used. A state trajectory for this case is shown in Figure 5. The chattering evident in case 1 has been eliminated.

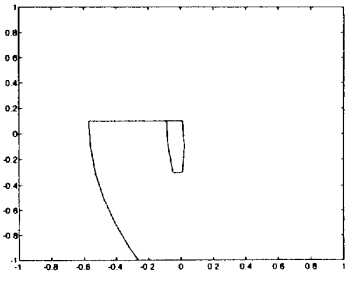

Figure 5: State Trajectory for Case 2

5.3 Case 3

In the final case we design both the interface and the controller. This time the natural invariants of the system are be used in the quantization. Three possible inputs are used to control the double integrator, $\{-2, 0, 2\}$, and they are used to compute three invariant functions for the system. The invariant function associated with the input -2 is computed as follows. The control policy is

$$f_1(\mathbf{x}) = x_2, \tag{29}$$
$$f_2(\mathbf{x}) = -2. \tag{30}$$

This gives the characteristic equation

$$\frac{dx_1}{x_2} = \frac{dx_2}{-2}, \tag{31}$$

with a solution

$$g_1(\mathbf{x}) = x_1 + .25{x_2}^2. \tag{32}$$

The remaining two invariant functions can be computed similarly.

$$g_2(\mathbf{x}) = x_2 \tag{33}$$
$$g_3(\mathbf{x}) = x_1 - .25{x_2}^2 \tag{34}$$

The values of these invariant functions are quantized to form the plant symbols.

$$\alpha(\mathbf{x}) = .25\text{round}(4g(\mathbf{x})) \tag{35}$$

Now the quantization yields the partition shown in Figure 6.

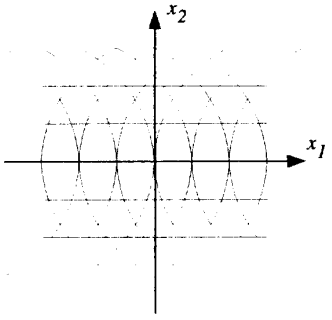

Figure 6: Quantization for Case 3

The controller for this case is similar to case 2.

$$\tilde{r}[k] = \begin{cases} -2.0 & \text{if} \quad g_3(\mathbf{x}) < 0, g_2(\mathbf{x}) < 0 \\ 2.0 & \text{if} \quad g_2(\mathbf{x}) > 0, g_3(\mathbf{x}) > 0 \\ 0.0 & \text{otherwise} \end{cases} \tag{36}$$

Figure 7 shows a state trajectory for this system. Notice that the chattering has been avoided as in Case 2; however the controller more easily follows from the interface design.

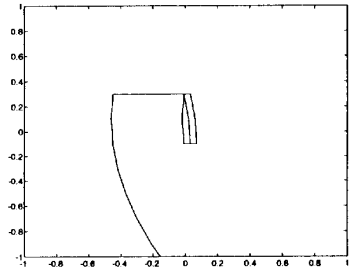

Figure 7: State Trajectory for Case 3

6 Conclusion

Techniques developed in the framework of hybrid control can be adapted and applied to digital control systems. This paper presented some early results in the use of system invariants in the design of hybrid control system interfaces. The invariants were shown to be useful in digital control systems as well, where quantization is the process analogous to the interface.

References

[1] P. Antsaklis, J. Stiver, and M. Lemmon, "Hybrid system modeling and autonomous control systems", In *Hybrid Systems*, edited by R. L. Grossman, A. Nerode, A. P. Ravn and H. Rischel, vol. 736 of *Lecture Notes in Computer Science*, pp. 366–392. Springer-Verlag, 1993.

[2] J. A. Stiver and P. J. Antsaklis, "On the controllability of hybrid control systems", In *Proceedings of the 32nd Conference on Decision and Control*, pp. 3748–3751, San Antonio, TX, Dec. 1993.

[3] J. A. Stiver, P. J. Antsaklis, and M. D. Lemmon, "A logical DES approach to the design of hybrid control systems", Technical Report of the ISIS Group ISIS-93-006, University of Notre Dame, Notre Dame, IN, October 1993.

[4] A. Isidori, *Nonlinear Control Systems*, Springer-Verlag, Berlin, 2 edition, 1989.

[5] P. J. Olver, *Applications of Lie Groups to Differential Equations*, Springer-Verlag, New York, 1986.

[6] P. J. Ramadge, "On the periodicity of symbolic observations of piecewise smooth discrete-time systems", *IEEE Transactions on Automatic Control*, vol. 35, no. 7, pp. 807–812, July 1990.

[7] C. Chase and P. J. Ramadge, "Dynamics of a switched n buffer system", In *Proceedings of the Twenty-Eighth Annual Allerton Conference on Communication, Control, and Computing*, pp. 455–464, University of Illinois at Urbana-Champaign, Oct. 1991.