Analysis of Zeno Behaviors in Hybrid Systems

Michael Heymann\textsuperscript{1}, Feng Lin\textsuperscript{2}, George Meyer\textsuperscript{3} and Stefan Resmerita\textsuperscript{4}

\textsuperscript{1}Department of Computer Science
Technion, Israel Institute of Technology
Haifa 32000, Israel
e-mail: heymann@cs.technion.ac.il

\textsuperscript{2}Department of Electrical and Computer Engineering
Wayne State University
Detroit, MI 48202
e-mail: flin@ece.eng.wayne.edu

\textsuperscript{3}NASA Ames Research Center
Moffett Field, CA 94035
e-mail: gmeyer@mail.arc.nasa.gov

\textsuperscript{4}Department of Computer Science
Technion, Israel Institute of Technology
Haifa 32000, Israel
e-mail: stefan@cs.technion.ac.il

March 7, 2002

\textsuperscript{*}This research is supported in part by NSF under grant ITR-0082784 and NASA under grant NAG2-1043 and in part by the Technion Fund for Promotion of Research. The work by the first author was completed while he was visiting NASA Ames Research Center, Moffett Field, CA 94035, under a grant with San Jose State University.
Abstract

In this paper we investigate conditions for existence of Zeno behaviors in hybrid systems. These are behaviors that sometimes arise in hybrid systems when a discrete controller unsuccessfully attempts to satisfy specified state invariance constraints and forces the system to undergo an unbounded number of discrete transitions in a finite and bounded length of time. We also study in some detail the relation between the possibility of existence of Zeno behaviors and the problem of existence of viable safety controllers for the system, that can satisfy the state invariance conditions indefinitely. Our analysis is based on studying the trajectory set of a certain continuous time system that is associated with the dynamic equations of the hybrid system. We investigate conditions for strong Zenoess of uncontrolled hybrid systems, when no controller can enforce the specified safety specification for an unbounded length of time. We show that when a hybrid system has Zeno behaviors but is not strongly Zeno, then some legal controller exists, but a minimally interventive controller may not exist. Moreover, in this case, standard controller synthesis procedures may be inadequate for controller design but more ad-hoc methods can be employed successfully.

Keywords: Hybrid systems, Zenoess, control

1 Introduction

In recent years, various algorithms have been proposed for the synthesis of safety controllers for hybrid systems [1], [2], [3], [4], [5], [6], [11], [13], [14]. These are controllers aimed at achieving specified state-space invariance constraints such as, for example, confining the system to remain within a given bounded region of the operating space.

Various controller-synthesis procedures have been proposed for design of such safety controllers (see e.g. [8] [12] [14]). While these algorithms differ somewhat in their technical details, they all share the basic approach of first computing the maximal control-invariant set which (when it exists) is the largest subset of the operating region (usually of the state space), from within which the system is not forced to violate the safety constraint. Then the controller is implemented as a device that switches discrete configurations whenever the boundary of this maximal invariant set is reached. The computation of the maximal control-invariant set is an iterative procedure, which starts with the set of all legal states (given by the specification) as the initial candidate. It then removes, in each iteration, the states from which the system can uncontrollably reach, in one discrete transition or by a continuous flow, either an illegal state or a state already removed in a previous iteration. The algorithm terminates when (and if) a fixed point is attained; that is, when an iteration step is reached in which no new states are thus removed. However, the algorithm is not
guaranteed to terminate finitely. When it terminates, there are two possibilities: (1) the result is a non-empty control-invariant set that includes the initial state, and a controller may (but, as discussed below, need not) exist, or (2) the result is the empty set or it does not include the initial state, in which case a safety controller does not exist.

As stated, once the maximal invariant set has been computed as described above, a controller is designed to take action and switch configurations only whenever the boundary of this set is reached, so as to insure that the system’s state remains within the invariant set. However, sometimes controllers, and in particular controllers synthesized as described above, cannot satisfy the invariance constraint for an indefinite length of time. They may force the system to undergo an unbounded (infinite) number of discrete configuration changes (switches) in a finite length of time and then violate the constraints. This phenomenon is called Zenoness\(^1\) (or a Zeno behavior), and can be thought of as a type of instability of hybrid systems that constitutes a major impediment to “proper” system behavior, and is an obstacle to successful controller synthesis, even in cases when controllers actually exist. In fact, it has been shown already in [6] that when the controlled system has possible Zeno behaviors, an incorrect result may be obtained from the computation of the maximal control-invariant set and the synthesized controller may be invalid. Furthermore, when the system has Zeno behaviors, a maximal invariant set may not exist at all (sometimes even when non-empty invariant sets exist).

When the system does not have any Zeno behaviors, a controller synthesized as described above that switches on the boundary of the maximal control-invariant set, is minimally interventive (or minimally restrictive)\(^7\) in that any other safety controller would preempt it and take earlier (and more frequent) action by possibly switching configurations while still in the interior of the maximal invariant set. However, the possible presence of Zeno behaviors changes and complicates the situation substantially.

With the aim of bypassing the difficulties created by the Zenoness phenomenon, several researchers proposed controller synthesis approaches, that limit the maximal switching rate of the synthesized controller, thereby yielding controlled systems that switch configurations at or below a specified upper rate. Such switching rate limitation is accomplished by imposing various structural constraints on either the system or on the controller [2], [3], [5], [14]. Yet, while such approaches guarantee that a synthesized controller will never yield a Zeno system, they do not answer the basic questions associated with the Zenoness phenomenon. Specifically, when controllers with the imposed switching rate constraint exist, are they necessarily minimally interventive for the system when no switching rate constraints are imposed? When controllers with the imposed switching rate constraint do not exist,

---

\(^{1}\)After the Greek philosopher Zeno whose famous paradox about the race between Achillis and the turtle resembles the said behavior.
what conclusions can be drawn regarding the existence and nature of controllers for the unconstrained system? Are Zeno behaviors inherently possible in the unconstrained system? When a safety controller for the constrained system exists, does there also exist a minimally interventive controller for the unconstrained one? If the answer to this latter question is affirmative, how are the two controllers related?

Thus, the possible presence of Zeno behaviors raises various essential questions regarding both the existence of and the nature of safety controllers for a given hybrid system. Some specific issues can be related directly to the algorithm for computation of the maximal control-invariant sets. These include the following:

- When the algorithm terminates finitely and gives a non-empty control invariant set, is the system controlled by synthesized controller nonZeno?
- If the algorithm terminates successfully but the synthesized controller is Zeno, do there exist other safety controllers for the system that are nonZeno?
- If the synthesized controller is Zeno can there exist a minimally interventive controller for the system?
- If the synthesis algorithm does not terminate finitely, does this mean that there exists no safety controller for the system?
- If the synthesis algorithm does not terminate finitely, can this mean that there exists safety controllers but no minimally interventive ones?
- If the synthesis algorithm does not provide the desired result (i.e., a minimally interventive controller), what other means can be employed for designing controllers if and when they exist?

In the present paper, we address some of the questions raised above. We confine our attention to controllers that can only trigger discrete transitions in the plant. Moreover, we assume that all the transitions in the plant can be triggered by a controller. We begin our investigation by examining constant rate systems in which each of the dynamic (state) variables has a constant rate in every discrete configuration. We then extend our investigation to bounded rate systems where the rate of each state variable is specified to lie within constant upper and lower bounds. Finally, we show that our approach also applies to more general hybrid systems with nonlinear dynamics.

Our approach is based on a simple but crucial observation that a state of the hybrid system is reachable at a given time if and only if it is reachable at the same time in an “equivalent” continuous system that is obtained as a suitable weighted combination of the
dynamic equations of the hybrid system in the different discrete configurations. Thus, instead of a difficult investigation of the rather complicated class of behaviors of the hybrid system, we examine the very simple class of behaviors of the “equivalent” continuous system.

2 The Hybrid Machine Model

In this section we briefly review the Hybrid-Machine formalism as described e.g. in [8]. A hybrid machine is denoted by

\[ HM = (Q, \Sigma, D, E, I, (q_0, x_0)) \]

The elements of HM are as follows.

- \( Q \) is a finite set of configurations.
- \( \Sigma \) is a finite set of event labels. An event is an input event, denoted by \( \sigma \) (underlined), if it is received by the HM from its environment; and an output event, denoted by \( \overline{\sigma} \) (overlined), if it is generated by the HM and transmitted to the environment.
- \( D = \{d_q = (x_q, y_q, u_q, f_q, h_q) : q \in Q\} \) is the dynamics of the HM, where \( d_q \), the dynamics at the configuration \( q \), is given by:
  \[
  \dot{x}_q = f_q(x_q, u_q), \\
  y_q = h_q(x_q, u_q),
  \]
  with \( x_q, u_q, \) and \( y_q, \) respectively, the state, input, and output variables of appropriate dimensions. \( f_q \) is a Lipschitz continuous function and \( h_q \) a continuous function. (A configuration need not have dynamics associated with it; that is, we permit \( d_q = \emptyset \), in which case we say that the configuration is static.) Note that the dynamics, and in particular the dimension of \( x_q \), can change from configuration to configuration.
- \( E = \{(q, G \land \sigma \rightarrow \overline{\sigma'}, q', x_q^0) : q, q' \in Q\} \) is a set of edges (or transition-paths), where \( q \) is the configuration exited, \( q' \) is the configuration entered, \( \sigma \) is the input event, and \( \overline{\sigma'} \) the output event. \( G \) is the guard, formally given as a Boolean combination of inequalities (called atomic formulas) of the form \( \Sigma_i a_i s_i \geq C_j \) or \( \Sigma_i a_i s_i \leq C_j \), where the \( s_i \) are signal variables, to be defined shortly, and the \( a_i \) and \( C_j \) are real constants. Finally, \( x_q^0 \) is the initialization value for \( x_{q'} \) upon entry to \( q' \).

Signal variables consist of output variables of configuration \( q \), as well as signals received from the environment. The set of signal variables defines a signal space \( S \).
An edge \((q, G \land \sigma \rightarrow \overline{\sigma}', q', x^0_{q'})\) is interpreted as follows: If the guard \(G\) is true and the event \(\sigma\) is received as an input, then the transition to \(q'\) takes place at the instant \(\sigma\) is received\(^2\), with the assignment of the initial condition \(x_{q'}(t_0) = x^0_{q'}\) (where \(t_0\) denotes the time at which the configuration \(q'\) is entered and \(x^0_{q'}\) is either a specified constant vector, or a function of \(x_q\)). The output event \(\overline{\sigma}'\) is transmitted as a side-effect at the same time.

There are a variety of special cases as follows. If \(\overline{\sigma}'\) is absent, then no output event is transmitted. If \(x^0_{q'}\) is absent (or partially absent), then the initial condition is inherited (or partially inherited) from \(x_q\) (assuming \(x_q\) and \(x_{q'}\) represent the same physical object, and hence are of the same dimension).

If \(\sigma\) is absent, then the transition takes place immediately upon \(G\) becoming true. Such a transition is called dynamic and is sometimes abbreviated as \((q, G, q')\) when \(\overline{\sigma}'\) and \(x^0_{q'}\) are either absent or understood. The guard associated with a dynamic transition is called a dynamic guard. If \(G\) is absent, the guard is always true and the transition will be triggered by the input event \(\sigma\). Such a transition is called an event transition and is sometimes abbreviated as \((q, \sigma, q')\) when \(\overline{\sigma}'\) and \(x^0_{q'}\) are either absent or understood. When both \(G\) and \(\sigma\) are present, the transition is called a guarded event transition.

- \(I = \{I_q : q \in Q\}\) is a set of invariants. For each \(q \in Q\), \(I_q\) is defined as \(I_q = cl(\neg (G_1 \lor \ldots \lor G_k))\), where \(G_1, \ldots, G_k\) are the dynamic guards at \(q\), and where \(cl(.)\) denotes set closure\(^3\).

- \((q_0, x_0)\) denotes the initialization condition: \(q_0\) is the initial configuration, and \(x_{q_0}(t_0) = x_0\).

The invariant \(I_q\) of a configuration \(q\) expresses the condition under which the HM is permitted to reside at \(q\); that is, the condition under which all of the dynamic guards are false (and the system is not forced out of \(q\) by a true dynamic guard). In particular, from the definition of \(I_q\) as \(I_q = cl(\neg (G_1 \lor \ldots \lor G_k))\), it follows that each of the configurations of the HM is completely guarded. That is, every invariant violation implies that some dynamic guard becomes true, triggering a transition out of the current configuration. (It is, in principle, permitted that more than one guard become true at the same instant. In such a case the transition that is actually selected is resolved nondeterministically.) It is further permitted that, upon entry into \(q\), one or more of the guards at \(q\) be already true. In such a case, the HM will immediately exit \(q\) and enter a configuration specified by (one of) the true guards. Such a transition is considered instantaneous.

---

\(^2\)If \(\sigma\) is received as an input while \(G\) is false, then no transition is triggered.

\(^3\)To avoid fruitless need to distinguish between open sets and closed sets, we shall always insist that invariants and guards be derived as closed sets - by taking their closure.
The HM runs as follows: At a configuration $q$, the continuous dynamics evolves according to $d_q$ until either a dynamic transition is triggered by a dynamic guard becoming true, or an event transition is triggered by the environment (through an input event, while the associated guard is either absent or true).

Since a guarded event transition can be treated as a dynamic transition followed by an event transition [8], we shall only need to consider two types of transitions: (1) dynamic transitions, that are labeled by dynamic guards only, and (2) event transitions, that are labeled by events only.

A run of the HM is a sequence

$$q_0 \xrightarrow{e_1,t_1} q_1 \xrightarrow{e_2,t_2} q_2 \xrightarrow{e_3,t_3} \ldots$$

where $e_i$ is the $i$th transition and $t_i(\geq t_{i-1})$ is the time when the $i$th transition takes place. For each run, we define its trajectory, time stamp, path and traces as follows.

- The trajectory of the run is the sequence of the vector time functions of the (state) variables:

$$x_{q_0}, x_{q_1}, x_{q_2}, \ldots$$

where $x_{q_i} = \{x_{q_i}(t) : t \in [t_i, t_{i+1})\}$.

- The time stamp of the run is a (column) vector function $In(t), t \geq 0$, where $\dim(In(t)) = \dim(Q)$. If at time $t \geq 0$ HM is in the $i$th configuration, then $In(t)$ has value 1 in its $i$th entry and zeros in all others.

- The path of the run is the sequence of the configurations.

- The input trace of the run is the sequence of the input events.

- The output trace of the run is the sequence of the output events.

We say that a path is irreducible if for any two consecutive configurations $q, q'$ in the sequence, either $q$ and $q'$ have different dynamics ($d_q \neq d_{q'}$), or, upon entry into $q'$, if $x^0_{q'} \neq \emptyset$, the state variable is (at least partially) re-initialized. A run is irreducible if its associated path is irreducible.

We shall call a run of a HM dynamic if all its transitions are dynamic transitions. If a dynamic run is reducible, i.e., if its associated path has consecutive configurations $q$ and $q'$ with identical dynamics and no re-initialization upon transition from $q$ to $q'$, the run can be reduced by combining $q$ and $q'$ into a single configuration. Thus, every dynamic run can be reduced to an irreducible one. An unbounded irreducible dynamic run

$$q_0 \xrightarrow{e_1,t_1} q_1 \xrightarrow{e_2,t_2} q_2 \xrightarrow{e_3,t_3} \ldots$$
is called a Zeno run if
\[ \lim_{i \to \infty} t_i = T < \infty \]

A HM is called Zeno if it possesses Zeno runs. Otherwise it is called non-Zeno or viable. A hybrid machine all of whose runs are Zeno is called strongly Zeno.

Clearly Zeno HMs are ill defined, in that they may uncontrollably execute an unbounded number of transitions in a finite (and bounded) time interval and thus describe systems whose lifetime is limited, contrary to our intention of modeling ongoing behaviors (that never terminate). In the next sections we shall explore conditions under which hybrid machines possess Zeno behaviors.

### 3 Zenoness

In certain applications, the state variables \( x_q \) represent similar (or sometimes the same) physical objects or phenomena in all configurations. In such cases the vectors \( x_q \) are of the same dimension in all configurations. When this is the case and if \( x_q \) is never re-initialized, we shall denote \( x_q \) simply by \( x \), and we shall call such systems homogeneous hybrid systems.

In the remainder of the paper we shall consider, without further mention, only homogeneous hybrid systems.

We shall assume that the system has \( n \) configurations; that is, \( \text{dim}(Q) = n \), and that the dynamics in the \( i \)th configuration is given by \( \dot{x} = f_i(x, u) \), \( y(t) = x(t) \). Thus, at each configuration, the state variable is also the output, so that the signal space, as defined above, is the state space.

For a run that starts at the initial state \( x(0) = x_0 \), the dynamics of \( x(t) \) for \( t \geq 0 \) can then be expressed as
\[
\dot{x} = F(x, u, t) := [f_1(x, u) \ f_2(x, u) \ldots f_n(x, u)] In(t).
\] (1)

This description, which resembles the dynamic representation of a continuous system, will be used below to derive various results on Zenoness.

To illustrate some aspects of the Zeno phenomenon, let us examine the following example.

---

Technion - Computer Science Department - Technical Report CIS-2002-03 - 2002
Example 1 Consider the hybrid system shown in Figure 1(a).

It consists of three configurations labeled by 1, 2, and 3. There are three continuous variables $x_1$, $x_2$, and $x_3$. The rates of changes of these variables are displayed in each configuration (thus, in configuration 1, $\dot{x}_1 = 100$, $\dot{x}_2 = -90$, $\dot{x}_3 = 1$, etc.). When a variable reaches some lower bound and the corresponding guard becomes true, a dynamic transition is triggered that takes the system to a different configuration (e.g., when $x_2$ becomes zero in configuration 1, a transition is triggered to configuration 2) as shown in Figure 1(a).

Note that in each configuration of the system, at least one variable is decreasing and will eventually cause the system to change configuration. We call such a variable an active variable.

This example is an extension of the two water-tank example that we first proposed in [8] and was later used by others [10]. However, the behavior of this system is much more complex than the two water-tank example, as can be seen in Figure 1. It is not very straightforward to deduce intuitively from the dynamics whether the system is Zeno. Indeed, the switching among the three configurations is highly irregular as shown by the simulation results in Figure 1(d) and the “water level” in each tank (the value of the variables) does not show

---

4Without loss of generality, we assume that the lower bounds are 0 in this paper.
an obvious pattern as can be seen in Figure 1(c). However, as can be seen in Figure 1(b), "time converges", that is, an unbounded number of transitions takes place in bounded time and hence the system is Zeno.

We are motivated, by this simple example and many others, to investigate the complex phenomenon of Zenoness. The first question that we would like to answer is how to check whether a system is Zeno or not, and the related question whether a safety controller exists.

### 3.1 Conditions forZenoness of Constant Rate Systems

To examine the Zenoness phenomenon and its relation to control synthesis, we review the concept of *instantaneous configuration cluster* (ICC) [8]. Let \( v = [s_1, \ldots, s_m] \in \mathcal{S} \) be a valuation of the signal vector (in our case the state vector) and let \( q \) be a configuration. Suppose that \( q \) is entered by a dynamic transition guarded by \( G \), whose value is true at \( v \). Assume further that \( q \) has an outgoing dynamic transition guarded by \( G' \), which becomes (or is) true at the entry value of the signal vector to \( q \). (In the present setup this value will be \( v \) since the signal vector is not re-initialized). Since \( G' \) follows \( G \) instantaneously, we say that the transition associated with \( G' \) is triggered by that associated with \( G \). A sequence of transitions \( G_1, G_2, \ldots \) is triggered by \( v \) if \( G_1 \) is true at \( v \) and \( G_{i+1} \) is triggered by \( G_i \) for all \( i \geq 1 \). For a signal value \( v \), consider all transition sequences in the HM triggered by \( v \). Let \( \text{HM}(v) \) denote the HM obtained by deleting all transitions that are not elements of transition sequences triggered by \( v \). A strongly connected component (SCC)\(^5\) of \( \text{HM}(v) \) that consists of two or more configurations is called an ICC. The triggering value \( v \) of the signal vector will be called a Zeno point of the HM. Note that there may exist more than one ICC for a given Zeno point and there may be more than one Zeno point for an ICC. In Example 1, \( v = x = [0, 0, 0] \) is a Zeno point associated with an ICC which includes configurations 1, 2 and 3. As stated earlier, for the systems described in this paper, the signal vector is equal to the state vector, since we assumed that all state variables are output variables.

In [8] it is shown that existence of a Zeno point and its associated ICC is a necessary condition for Zenoness, although it is not sufficient. Clearly, once at a Zeno point, the behavior of the HM is necessarily Zeno. Thus, the question that must be examined is whether if initialized outside (or away from) a Zeno point, a possible run will enter the Zeno point after a bounded length of time. We shall say that a Zeno point is a Zeno attractor whenever there exist initializations of the HM outside the Zeno point such that for some run, the Zeno point will be reached in bounded time. Clearly, a HM is non-Zeno if and only if it has no Zeno attractor. Thus, the problem of checking Zenoness of a HM consists

\[^5\text{An SCC is a set of configurations for which there is a directed path from any configuration to any other.}\]
of identifying its ICCs, if any, and checking whether they include Zeno attractors. In this paper, we address the latter issue.

We consider a homogeneous hybrid system with \( n \) configurations and \( m \) continuous variables. We confine our attention first to constant rate hybrid systems, for which the continuous dynamics in configuration \( j, j = 1, 2, \ldots, n \), is given by

\[
\begin{bmatrix}
\dot{x}_1 \\
\dot{x}_2 \\
\vdots \\
\dot{x}_m
\end{bmatrix} = \begin{bmatrix}
k_{1j} \\
k_{2j} \\
\vdots \\
k_{mj}
\end{bmatrix},
\]

and we shall consider systems that satisfy the following assumption:

**Assumption 1**

1. The legal region of the system is the nonnegative orthant \( \mathbb{R}_+^m = \{ x \in \mathbb{R}^m : x_i \geq 0, i = 1, 2, \ldots, m \} \).
2. All the system’s configurations are in an ICC with respect to the Zeno point \( x = 0 \).
3. Every variable is active in some configurations.
4. In every configuration, there is at least one active variable.
5. In a given configuration, a unique transition is associated with each active variable \( x_i \). This transition is triggered either by an event (generated by a controller) or by the associated guard \([x_i \leq 0]\) becoming true. Each transition leads the system to a configuration where the triggering variable \( x_i \) is not active.

In the above Assumption, (1) implies that a variable is active if and only if its derivative is negative, (2) states that every configuration is relevant to the Zeno behavior, (3) states that every variable is relevant to the Zeno behavior of the system, (4) ensures that the hybrid system cannot stay in any configuration indefinitely and hence the system is forced to perform an unbounded number of transitions over an unbounded interval of time, and (5) states that the hybrid system can be forced to exit a configuration at any time before \([x_i \leq 0]\) becomes true.

Let us consider a run of a hybrid system HM initialized at state \( x(0) = x_0 \). We assume that \( x_0 \) is in \( \text{int}(\mathbb{R}_+^m) \), the interior of \( \mathbb{R}_+^m \). Using equation (1), we obtain the state \( x(t) \) at \( t \geq 0 \) as

\[
x(t) = \int_0^t KI n(\tau)d\tau + x_0,
\]

(2)
where $K$ is the rate matrix

$$K = \begin{bmatrix}
  k_{11} & k_{12} & \ldots & k_{1n} \\
  k_{21} & k_{22} & \ldots & k_{2n} \\
  \vdots & \vdots & & \vdots \\
  k_{m1} & k_{m2} & \ldots & k_{mn}
\end{bmatrix}.$$  

Equation 2 can be rewritten as

$$x(t) = \int_0^t K I_n(\tau) d\tau + x_0 = K \int_0^t I_n(\tau) d\tau + x_0 = K t \alpha(t) + x_0, \quad (3)$$

where $\alpha(t) = \frac{1}{t} \int_0^t I_n(\tau) d\tau =: [\alpha_1(t), \alpha_2(t), \ldots, \alpha_n(t)]^t$. Note that $\alpha_i(t) \geq 0, i = 1, 2, \ldots, n$, and $\alpha_1(t) + \alpha_2(t) + \ldots + \alpha_n(t) = 1$. Thus, $\alpha_i(t)$ represents the fraction of time (up to time $t$), that the HM resides in configuration $i; i = 1, 2, \ldots, n$. In other words,

$$\alpha(t) \in \mathcal{A} := \{\alpha \in \mathbb{R}_+^n | \sum_{i=1}^n \alpha_i = 1\}.$$

It is readily noted that $x(t) = \int_0^t K I_n(\tau) d\tau + x_0$ is also the solution of the following constant rate dynamical system

$$\begin{cases}
  \dot{x} = K \alpha \\
x(0) = x_0
\end{cases} \quad (4)$$

for $\alpha = \alpha(t)$. This much simpler "equivalent" system will serve us below to investigate the Zenoness properties of the hybrid system HM. In particular, we will show that the existence of Zenoness is closely related to the existence of solutions to the inequality $K \alpha \geq 0, \alpha \in \mathcal{A}$.

We shall make use of the following simple observation.

**Lemma 1** Let HM be a homogeneous constant rate hybrid system satisfying Assumption 1 with initial state $x(0) = x_0 \in \text{int}(\mathbb{R}_+^m)$. Let $x \in \text{int}(\mathbb{R}_+^m)$ be any point. Then there exists a run of HM reaching $x$ with a trajectory wholly contained in $\mathbb{R}_+^m$ if and only if for some $\alpha \in \mathcal{A}$ there exists a solution to system (4) starting at $x_0$ and reaching $x$. Moreover, in that case, the time $T$ at which HM reaches $x$ (i.e., $x(T) = x$) is the same as the time at which the equivalent system (4) reaches $x$.

**Proof**

(Only if) Suppose there exists a state trajectory of HM, wholly contained in $\mathbb{R}_+^m$, starting at $x_0$ and reaching $x$ at time $T$; that is, $x(T) = x$. Then, the solution of system (4) starting at $x_0$ at time 0, with the value of $\alpha$ taken as $\alpha(T)$ from Equation 3, will reach the state $x$ at time $T$.  

12
(If) If there exists a trajectory of system (4), for some \( \alpha^* \in \mathcal{A} \), starting at \( x_0 \) and reaching \( x \), then this trajectory is a line segment with endpoints \( x_0 \) and \( x \). Assume \( x \) is reached at time \( T \) (i.e., \( x(T) = x \)). Then any trajectory of HM satisfying \( \alpha(T) = \alpha^* \) will be a trajectory from \( x_0 \) to \( x \). Although not all such trajectories are contained in \( \mathbb{R}^m_+ \), we will see that there exist trajectories that are. Note that since the line segment connecting \( x_0 \) and \( x \) is wholly contained in the open set \( \text{int}(\mathbb{R}^m_+) \), there exists \( \epsilon > 0 \), for which the \( \epsilon \)-neighborhood of this line segment is also contained in \( \text{int}(\mathbb{R}^m_+) \). We can construct a run of HM whose trajectory stays within this \( \epsilon \)-neighborhood (and hence in \( \mathbb{R}^m_+ \)) as follows. We first partition the line segment \([x_0, x]\) into \( N \) equal sections. The end points of these sections are denoted by \( x^1, x^2, ..., x^N = x \). Let \( t_i \) be the time when \( x^i \) is reached: \( x(t_i) = x^i, i = 1, 2, ..., N \). Let a run of HM be such that \( \alpha(t_i) = \alpha^*, i = 1, 2, ..., N \). Then the trajectory of the run will intersect with the line segment at \( x^1, x^2, ..., x^N \). Since we can make each section sufficiently small by selecting sufficiently large \( N \), we can ensure that the deviation of the trajectory from the line segment \([x_0, x]\) is sufficiently small.

By investigating the equivalent system (4) instead of the original hybrid system HM, we can simplify the problem of determining Zenoness significantly. In particular, we have the following necessary and sufficient condition for strong Zenoness.

**Theorem 1** Let HM be a homogeneous constant-rate hybrid machine satisfying Assumption 1 with initial state \( x(0) = x_0 \in \text{int}(\mathbb{R}^m_+) \). Then HM is strongly Zeno if and only if \( K\alpha \geq 0 \) has no solutions in \( \mathcal{A} \).

**Proof**

(If) Assume that \( K\alpha \geq 0 \) has no solutions in \( \mathcal{A} \), but that HM has some non-Zeno run such that for all \( t \geq 0 \),

\[
x(t) = K\alpha(t)t + x_0 \in \mathbb{R}^m_+.
\]

Let \( \{t_i\}_{i \in \mathbb{N}}, t_{i+1} > t_i \), be an unbounded sequence of times. Then, since \( \alpha(t_i) \in \mathcal{A} \) for all \( i \), and since \( \mathcal{A} \) is compact, the sequence \( \alpha(t_i) \) has a convergent subsequence \( \alpha(t_{i_j}) \) with limit \( \alpha^* \in \mathcal{A} \). Let \( v = K\alpha^* \). Since, by assumption, \( K\alpha \geq 0 \) has no solutions in \( \mathcal{A} \), it follows that \( v_j < 0 \) for some \( j \in \{1, ..., m\} \). Hence, there exists \( 0 < t^* < \infty \), such that at least one component of \( x(t) = K\alpha^*t + x_0 \) will become negative for all \( t > t^* \). But then, since \( K\alpha t + x_0 \) is continuous in \( \alpha \), also some component of \( x(t) = K\alpha(t)t + x_0 \) will become negative for finite \( t \), contradicting our assumption that a non-Zeno run exists.

(Only if) Suppose there exists \( \alpha^* \in \mathcal{A} \) such that \( K\alpha^* \geq 0 \). Then for \( x_0 \in \text{int}(\mathbb{R}^m_+) \), the trajectory \( x(t) = K\alpha t + x_0 \in \text{int}(\mathbb{R}^m_+) \) for all \( t \geq 0 \). By Lemma 1 there exists then a run of HM starting at \( x_0 \), which is wholly contained in \( \mathbb{R}^m_+ \), in contradiction with the assumption that HM is strongly zeno.
The condition of Theorem 1 (which is the standard feasibility condition for solution of a linear program) can easily be checked using standard available software. If $K^\alpha \geq 0$ has solutions, the HM is not strongly Zeno and there exist switching policies resulting in non-Zeno runs of the system. However, without externally forced switching, the dynamic runs may still be Zeno. We shall discuss the control issues in Section 4.

3.2 Regular Systems

Although the problem of finding necessary and sufficient conditions for Zenoness (rather than strong Zenoness) is still open, we can solve the problem for regular systems, which satisfy both Assumption 1 and the following:

Assumption 2 The number of continuous (state) variables is equal to the number of configurations (that is, $n = m$). Each state variable is active in exactly one configuration. Furthermore, the rate matrix is of full rank (that is, $\text{rank}(K) = n$).

To present our results, let us consider all convex cones in $\mathbb{R}^n$ rooted at the origin. Denote by

$$CONE(v_1, v_2, \ldots, v_l) = \{v \in \mathbb{R}^n : v = \beta_1 v_1 + \beta_2 v_2 + \ldots + \beta_l v_l \text{ for some } \beta_1 \geq 0, \beta_2 \geq 0, \ldots, \beta_l \geq 0\}$$

the convex cone generated by vectors $v_i \in \mathbb{R}^n, i = 1, 2, \ldots, l$.

Let $u_i = [0 \ldots 1 \ldots 0]^T$ be the $n$-vector with 1 in its $i$th position and 0 elsewhere. Denote

$$PO = CONE(u_1, u_2, \ldots, u_n)(= \mathbb{R}^n_+)$$
$$NE = CONE(-u_1, -u_2, \ldots, -u_n).$$

If $\text{rank}[v_1 v_2 \ldots v_l] = r$, then the dimension of $CONE(v_1, v_2, \ldots, v_l)$ is $r$. Its boundary consists of $r$ surfaces. Each surface is a part of a supporting hyperplane, generated by some $r - 1$ independent vectors in $\{v_1, v_2, \ldots, v_l\}$.

**Lemma 2** Let $C_1$ and $C_2$ be two cones. If the surfaces of $C_1$ intersect $C_2$ only at the origin, then either $C_2$ is contained in $C_1$, or $C_1$ is contained in the complement of $C_2$.

**Proof**

Elementary.

Denote the column vectors of $K$ by $k_i$: $K = [k_1 k_2 \ldots k_n]$.

**Lemma 3** Under Assumption 2, the surfaces of $CONE(k_1, k_2, \ldots, k_n)$ and $NE$ intersect only at the origin.
Proof

Under Assumption 2, the matrix $[k_i, k_{i_2} \ldots k_{i_{n-1}}]$ consisting of any $n-1$ columns of $K$, $k_{i_1}, k_{i_2}, \ldots, k_{i_{n-1}}$, has at least one row all of whose elements are nonnegative. Therefore, the surface generated by the vectors $k_{i_1}, k_{i_2}, \ldots, k_{i_{n-1}}$ intersects with $NE$ only at the origin.

Lemma 4 Under Assumption 2, $K\alpha \geq 0$ has no solution in $\mathcal{A}$ if $K\alpha < 0$ has a solution in $\mathcal{A}$.

Proof

By Lemmas 2 and 3, $NE$ is either contained in $CONE(k_1, k_2, \ldots, k_n)$, or is contained in the complement of $CONE(k_1, k_2, \ldots, k_n)$.

Suppose $K\alpha < 0$ has a solution in $\mathcal{A}$. This means that $CONE(k_1, k_2, \ldots, k_n) \cap NE \neq \{0\}$. Therefore, $NE$ is contained in $CONE(k_1, k_2, \ldots, k_n)$ and hence $CONE(k_1, k_2, \ldots, k_n) \cap PO = \{0\}$. Because $K$ is of full rank, $K\alpha \geq 0$ has no solution in $\mathcal{A}$.

With these three lemmas, we can prove the following theorem that gives a necessary and sufficient condition for Zenoness of regular systems.

Theorem 2 Under Assumptions 1 and 2, a homogeneous constant-rate hybrid system HM is Zeno if and only if $K\alpha \geq 0$ has no solution in $\mathcal{A}$.

Proof

If $K\alpha \geq 0$ has no solution in $\mathcal{A}$, then by Theorem 1 HM is strongly Zeno and hence Zeno. If HM is Zeno, then it has a Zeno run. Let $\alpha_z \in \mathcal{A}$ be associated with that run. Clearly $K\alpha_z < 0$. By Lemma 4, the system of inequalities $K\alpha \geq 0$ has no solution in $\mathcal{A}$.

Note that for systems satisfying both Assumption 1 and Assumption 2, Zenoness and strong Zenoness are equivalent; that is, there exists a Zeno run of a system if and only if all its runs are Zeno. Also note that for systems satisfying Assumption 1 but not Assumption 2, no conclusion can be drawn just from the existence of solutions in $\mathcal{A}$ to the inequality $K\alpha \geq 0$, as to whether the system is Zeno or not. In the next subsection, we shall provide illustrative examples to demonstrate different aspects of Zenoness for such cases.

3.3 Illustrative Examples

Zeno behaviors have a complex nature even for systems satisfying Assumption 1 (but not Assumption 2) as we will illustrate by the following examples. Note that when the conditions of Theorem 1 or Theorem 2 are satisfied, then the results are independent of the initial
conditions and the exact layout of connections between configurations. However, when these conditions are not satisfied, a dynamic run may or may not be Zeno depending on the initial conditions and on the exact layout of connections and guards between configurations. This is illustrated in Examples 2 and 3.

**Example 2** This example shows a hybrid system in which certain dynamic runs are Zeno and others are not, depending on the initial condition. The system is shown in Figure 2.

**Figure 2:** A system where Zenoness depends on the initial state

This system satisfies Assumption 1 but is not regular, since the second configuration has two active variables. Notice further, that while $K\alpha \geq 0$ has solutions in $\mathcal{A}$ and $K\alpha < 0$ has no solutions in $\mathcal{A}$, Zeno behaviors are possible. To understand the dynamic behavior of this system, observe that the loop consisting of configurations 1 and 2 (denoted by $1 \leftrightarrow 2$) has active variables $x_2$ and $x_3$. The submatrix corresponding to these variables is

$$K^{L}_{sub} = \begin{bmatrix} -90 & 130 \\ 1 & -90 \end{bmatrix},$$

and represents a Zeno regular HM; that it, $K^{L}_{sub}$ satisfies Assumption 2 and $K^{L}_{sub}\alpha \geq 0$ has no solutions in $\mathcal{A}^{L}_{sub} := \{\alpha_2, \alpha_3 | \alpha_2 \geq 0, \alpha_3 \geq 0, \alpha_2 + \alpha_3 = 1\}$. Thus, if a dynamic run is “trapped” in the loop $1 \leftrightarrow 2$, Zeno behavior must occur.

On the other hand, the loop $2 \leftrightarrow 3$ consisting of configurations 2 and 3, has active variables 1 and 2 with associated submatrix

$$K^{R}_{sub} = \begin{bmatrix} -90 & 70 \\ 130 & -90 \end{bmatrix}$$

which represents a non-Zeno regular HM ($K^{R}_{sub}\alpha \geq 0$ has solutions in $\mathcal{A}^{R}_{sub}$). Hence, if a dynamic run is “trapped” in the loop $2 \leftrightarrow 3$, it will be non-Zeno.
One can see that the system of Figure 2 will be trapped in one of the two loops after a number of initial transitions. Suppose that the initial configuration is 1. When $x_2 = 0$, a transition takes the system to configuration 2. Now suppose $x_3$ hits its guard before $x_1$ (i.e., $x_3 = 0$ is reached while $x_1 > 0$) and the system switches back to configuration 1, where the rate of $x_1$ is greater than the rate of $x_3$. After a while, the transition to configuration 2 takes place again, where $x_1$ and $x_3$ have the same negative rate, and therefore $x_3$ will again become zero before $x_1$, forcing the system back to configuration 1, and so on.

Figure 3: Representative Runs
(A) Zeno Run: $x_0 = [2, 90, 130]$, $q_0 = 1$; (B) Non-Zeno Run: $x_0 = [1, 90, 131]$, $q_0 = 1$
Thus, the behavior of the system is given by the matrix $K_{\text{sub}}$, corresponding to $x_2$ and $x_3$ in configurations 1 and 2. On the other hand, if after the first transition, $x_1$ becomes zero before $x_3$, a similar argument shows that the behavior depends only on the matrix $K_{\text{sub}}^R$ corresponding to $x_1$ and $x_2$ in configurations 2 and 3. Therefore, we conclude that the run will or will not be Zeno, depending on the initial state. A simple calculation shows that, for $q_0 = 1$, the run is Zeno if $x_{01} > x_{03} - (129/90)x_{02}$, and it is non-Zeno if $x_{01} < x_{03} - (129/90)x_{02}$. In the case of equality, then after the first transition (from configuration 1 to configuration 2), both variables $x_1$ and $x_3$ become zero in configuration 2 at the same instant, and the system chooses its next configuration (either 1 or 3) non-deterministically, thereby becoming Zeno if it switches to configuration 1 and non-Zeno if it switches to configuration 3. Two sample runs that demonstrate Zeno and non-Zeno behaviors of this system are shown in Figure 3.

Example 3 This example shows two systems with identical configurations and dynamics (i.e., with the same rate matrix $K$), as well as identical invariants, that differ in their connection-layouts between configurations. One of these layouts is shown in Figure 4(a) and yields a non-Zeno system, while the other layout is shown in Figure 5(a) and yields a Zeno system.
Notice that when the system of Figure 4(a) is in configuration 2, the condition \([x_1 \leq 0]\) triggers a transition to configuration 1, where \(\dot{x}_1 = 130\). On the other hand, in the system of Figure 5(a), the condition \([x_1 \leq 0]\) triggers a transition to configuration 3, where \(\dot{x}_1 = 1\) (which is much smaller than 130).

Let us compare the individual loops in the two systems. For the system in Figure 4(a), the loop 1 \(\leftrightarrow\) 2 has active variables \(x_1\) and \(x_2\), whose rate-matrix corresponds to non-Zeno behavior. Similarly, the loop 2 \(\leftrightarrow\) 3 has active variables \(x_2\) and \(x_3\), whose rate-matrix also corresponds to non-Zeno behavior. On the other hand, in the system shown in Figure 5(a), the loop 1 \(\leftrightarrow\) 2 has active variables \(x_2\) and \(x_3\) and the loop 2 \(\leftrightarrow\) 3 has active variables \(x_1\) and \(x_2\). The rate matrices of both these loops correspond to Zeno behaviors.

Using reasoning similar to that in Example 2, one can see that the system shown in Figure 5(a) will be trapped in one of the two loops following some initial transitions. In either loop, the system is Zeno.

On the other hand, an analysis of the system shown in Figure 4(a) reveals that the system is never trapped in one of the two loops. Still, in spite of the non-Zeno run exhibited in Figure 4, no general conclusion can be drawn regarding the Zenoness of the system. (Recall Example 1, where the system had two non-Zeno loops yet the system was still Zeno.)
Example 4  This example shows that even for a Zeno system that has only one loop (and hence only one switching sequence), there may exist non-Zeno runs when switched properly.

The system is shown in Figure 6(a). Its dynamic run (i.e., when switched by the guards becoming true) is Zeno as shown in Figure 6(b)– Figure 6(d). However, $K\alpha \geq 0$ has solutions in $\mathcal{A}$. For example, one solution is $\alpha^* = [0.125, 0.125, 0.5, 0.25]^T$. Therefore, if the system is switched to remain in the proximity of the line emanating from $x_0$ in the direction of $\alpha^*$ (as discussed in the proof of Lemma 1), the run will be non-Zeno.

4 Zenoness in Controlled Hybrid Systems

In the previous section, we have examined various conditions for Zeno behaviors and derived necessary and sufficient conditions for Zenoness in constant-rate Hybrid Machines. In this section, we shall use these results to investigate the existence and synthesis of controllers for such systems.

A controller $C$ of a hybrid system $HM$ is another hybrid machine that runs in parallel with $HM$. The controlled system (also called closed-loop system) is the composite of the two
running in parallel and is denoted by $HM|C$. We assume that $C$ and HM interact by event synchronization only. That is, $C$ controls HM only by triggering (event) transitions and does not interfere otherwise with the continuous dynamics of HM $[8]$. Controllers are used to ensure the satisfaction of safety $[1] [4] [8] [11] [13]$, liveness $[9]$ and optimality specifications of systems. A safety specification is a state-invariance constraint that specifies a ‘legal’ region of operation in which the system must remain at all time. A safety controller is aimed at ensuring that the system never leave the specified legal region. Various algorithms have been proposed in the literature for synthesis of safety controllers for hybrid systems. Essentially, all these are iterative “layer peeling” algorithms that employ the following basic approach: Let $L$ and $I$ denote the legal and illegal regions, respectively, of the operating space. Then at the $i^{th}$ iteration, the algorithm computes the set of all initial states (or, region in the operating space) $S_i$, from which there exists no control policy under which the safety requirement can be satisfied for more than $i - 1$ discrete (switching) steps. Consequently, when starting in any state of $S_i$, it will take, for any control policy, at most $i$ steps before the system state will enter $I$. The $(i + 1)^{th}$ iteration of the algorithm consists of computing the set $S_{i+1} \subseteq L - S_i$ of states from which, in at most one step, the system will be forced to enter $I \cup S_i$, so that $S_{i+1} = S_{i+1} \cup S_i$. The algorithm terminates at step $i$ either if $(q_0, x_0) \in S_i$ and a safety controller does not exist, or if a fixed point is reached; that is, $S_{i+1} = S_i (=: S^*)$. There is, of course, also the possibility that the algorithm never terminates and no conclusion can be drawn at all. In case, the algorithm terminates finitely, and a fixed point is reached, a safety controller exists and the system can be controlled to satisfy the safety specification indefinitely, provided the system is non-Zeno $[6]$. We then say that the controlled system is viable $[8]$. Moreover, the controller $C$ is obtained from the synthesis algorithm as the hybrid machine that forces configuration transitions in the controlled system when the boundary of $L - S^*$ is reached, thereby avoiding entrance into the illegal region $I$. Such a controller, is then minimally interventive $[6]$.

The situation gets to be complicated when the system to be controlled is Zeno. In that case, a controller may or may not exist even if the synthesis algorithm terminates finitely. Moreover, even if a safety controller for a Zeno system exists, there may not exist a minimally interventive one as will be seen below. Finally, as mentioned above, there remains the case in which the algorithm does not converge finitely and the existence of a safety controller cannot be resolved algorithmically.

In this section we discuss the issue of controller existence and synthesis in some detail. For the purpose of our discussions, we shall assume that the system to be controlled is a constant rate HM with rate matrix $K$ that satisfies Assumption 1. We assume that all dynamic transitions can be preempted by controls (i.e., the transitions can be triggered by the controller at or before the corresponding guards become true). This implies, in
particular, that the configurations are an instantaneous configuration cluster (in the sense that any configuration can be reached instantaneously from any other by a suitable sequence of controlled transitions).

For systems that satisfy the above conditions, the standard synthesis algorithm terminates in one iteration and generates a controller $C$, that switches configurations whenever the boundary of the legal region is reached; namely, whenever the active variable becomes zero.

The obtained controller $C$ may or may not be viable, depending on whether the closed-loop system $HM||C$ is Zeno or not. For systems that satisfy Assumption 1, we can reach the following conclusions from the results of the previous section.

1. If $K\alpha \geq 0$ has no solution in $A$, then by Theorem 1, any run of HM is Zeno. Therefore $HM||C$ is Zeno. In fact, no viable controller for HM exists and there is no point in even trying to synthesize a controller.

2. If HM also satisfies Assumption 2 and hence is regular (that is, $n = m$ and each variable is active in exactly one configuration) and if $K\alpha \geq 0$ has a solution in $A$, then by Theorem 2, no run of HM is Zeno. Therefore $HM||C$ is non-Zeno and $C$ is the minimally interventive safety controller.

3. If the hybrid machine HM satisfies Assumption 1 but not assumption 2 (and hence is not regular), and $K\alpha \geq 0$ has a solution in $A$, then the behavior of the synthesis algorithm and properties of the controller cannot be determined from the results of the previous section, and some further examination is required.

In Case 3, since $K\alpha \geq 0$ has a solution in $A$, a non-Zeno safety controller exists. However, it may not be obtainable by the standard algorithmic approach. Moreover, while a controller exists, there is no guarantee that there exists a minimally interventive one. Indeed, as illustrated in Example 4, if $\alpha^*$ is a solution to $K\alpha \geq 0$, a controller can be obtained by switching configurations to remain in the proximity of the ray emanating from the initial state $x_0$ in the direction of $\alpha^*$, as was discussed earlier. However, while such a controller is viable and guarantees safety, it cannot be synthesized using the standard algorithmic approach. To see this, let us return to Example 4. Note that the controller $C$ synthesized by the standard approach switches on the boundary of $\mathbb{R}_+^m$, and the closed-loop system which is represented by Figure 6(a), is Zeno. This is also an example in which a minimally interventive safety controller does not exist. Indeed, consider a controller that operates in two phases. In the first phase the controller switches configurations on the boundary of $\mathbb{R}_+^m$ for a finite but arbitrarily large number of times. It holds the system in $\mathbb{R}_+^m$ in a minimally interventive way, while allowing the system to approach the Zeno point. The second phase starts at a point
that the system reaches in the first switching phase, and the controller begins to switch configurations so as to remain in \( \mathbb{R}^m_+ \) close to line emanating from \( x \) in the direction of \( \alpha^* = [0.125, 0.125, 0.5, 0.25]^T \). The result is a viable and non-Zeno safety controller. This procedure enables us to design a controller that allows the system to get arbitrarily close to the Zeno point (by switching on the boundary in a minimally intervention way) and then drive it away from the Zeno point (by switching “along” \( \alpha^* \)). Hence, for any safety controller of the type synthesized, there exists another controller that allows the system in the first phase to get yet closer to the Zeno point before driving it away. It follows that for any safety controller there is another safety controller that is less intervention (by staying longer in phase one). It follows that a minimally intervention controller does not exist. However, this is equivalent to the fact that a maximal control-invariant set does not exist, disproving the conjecture made in [12] (Proposition 3) that if a viable controller exists, then a unique maximal control-invariant set exists as well.

Since it has been understood for some time that system Zenoness is an impediment to controller synthesis, various ways have been sought to prevent and bypass its possible occurrence [10]. In particular, it has been argued that Zenoness is a modelling artifice and real physical systems cannot switch configurations at an arbitrarily high rate. Thus, various model “regularization” methods have been suggested in the literature that are aimed at forcing delays between successive configuration switches, thereby preventing Zenoness from occurring. It is interesting to examine what the actual effect is of model regularization and Zenoness elimination on the controller synthesis results. To this end, let us reexamine each of the three cases discussed above:

1. Clearly, in case 1, the introduction of switching delays will not help, since in this case all runs are Zeno and obviously no safety controller exists, regardless as to whether or not delays are permitted. The iterative synthesis procedure will either fail to converge or will decide finitely that a controller with the specified delay does not exist. However, there is no indication whether a controller with a smaller delay exists or not. One may falsely hope that it does!

2. In Case 2, the minimally intervention controller exists without a delay specification. Hence, the standard algorithm terminates finitely, with a viable controller design. However, a controller may not exist for a specified minimal delay and a given initial condition. Therefore, one may falsely conclude that a controller does not exist.

3. Case 3 is the only case in which the introduction of a delay may help. This can be seen from examination of Example 3. Synthesis without a delay results in a controller that switches nondeterministically either according to Figure 4(a) or according to Figure...
Such switching may (but does not need to) produce a Zeno run. If the system is “regularized”, for example by introduction of a delay, the controller synthesis algorithm will produce the correct switching pattern as shown in Figure 4(a). The introduction of the delay in the algorithm is not cost-free because we know that a non-Zeno safety controller without specified delay actually exists and can be found by the analysis presented in the present paper, which can be employed for selecting the preferred switching pattern. The introduction of a delay in systems that fall into ‘Case 3’ may have other undesirable effects as shown in Example 4. For this Zeno system the standard synthesis algorithm with delay does not converge finitely and hence does not terminate. However, we know that a safety controller exists (although not a minimally interventive one).

It is clear from the above discussion that while the introduction of a delay (or other “regularization” procedure) prevents the system from becoming Zeno, it is not an effective method for solving the safety controller design problem for Zeno (or non-Zeno) hybrid machines.

5 Extensions to Bounded-Rate Hybrid Systems

In this section, we extend the results of Section 3 to bounded-rate hybrid systems. Recall that in a bounded-rate system, the dynamics behavior is given by

\[ \dot{x}_i \in [k_{ij}^L, k_{ij}^U], i = 1, 2, ..., m, j = 1, 2, ..., n. \]

where \( k_{ij}^L \) and \( k_{ij}^U \) are the lower and upper bounds of the rate. By bounded rate we mean that all that is assumed about the dynamics is that the rate can take any (possibly time varying) value in the specified range (subject, of course, to standard integrability conditions). Let us define lower and upper rate-bound matrices

\[
K^L := 
\begin{bmatrix}
  k_{11}^L & k_{12}^L & \cdots & k_{1n}^L \\
  k_{21}^L & k_{22}^L & \cdots & k_{2n}^L \\
  \vdots & \vdots & \ddots & \vdots \\
  k_{m1}^L & k_{m2}^L & \cdots & k_{mn}^L
\end{bmatrix},
\]

\[
K^U := 
\begin{bmatrix}
  k_{11}^U & k_{12}^U & \cdots & k_{1n}^U \\
  k_{21}^U & k_{22}^U & \cdots & k_{2n}^U \\
  \vdots & \vdots & \ddots & \vdots \\
  k_{m1}^U & k_{m2}^U & \cdots & k_{mn}^U
\end{bmatrix}.
\]

For a bounded-rate system \( HM^o \), we define lower-rate system \( HM^L \) and upper-rate system \( HM^U \) as the constant rate hybrid systems obtained from \( HM^o \) by replacing the
continuous dynamics with constant-rate dynamics given by $K^L$ and $K^U$, respectively. Our objective is to show that we can investigate Zenoness of $HM^o$ by investigating Zenoness of $HM^L$ and $HM^U$. To this end, let us first prove the following lemma.

**Lemma 5** Assume $HM^L$ and $HM^U$ both satisfy Assumption 1. (1) If $HM^o$ has a non-Zeno run then $HM^U$ has a non-Zeno run. (2) If $HM^L$ has a non-Zeno run then $HM^o$ has a non-Zeno run.

**Proof**
We prove only (1) because the proof of (2) is similar. Let

$$r^o = q_0 \xrightarrow{e_1.t_1} q_1 \xrightarrow{e_2.t_2} q_2 \xrightarrow{e_3.t_3} \ldots$$

be a non-Zeno legal run of $HM^o$. Let

$$r^U = q_0 \xrightarrow{e_1.t_1} q_1 \xrightarrow{e_2.t_2} q_2 \xrightarrow{e_3.t_3} \ldots$$

be a run of $HM^U$ that switches at exactly the same times and to exactly the same configurations as $r^o$. We need to show that $r^U$ is a legal run of $HM^U$. To this end, we shall see that $x^U(t) \geq x(t) \geq 0$ for all $t$. This can be done inductively as follows: Initially, $x^U(0) = x(0) = x_0 > 0$. Let us suppose that upon entry to $q_j$, $x^U(t_i) \geq x(t_i) \geq 0$. Since at configuration $j$, $\dot{x}^U = k^U_j \geq \dot{x}$, it follows that $x^U(t) \geq x(t) \geq 0$ for $t \in [t_i, t_{i+1}]$ (where the inequality $x(t) \geq 0$ follows from the assumption that $r^o$ is a legal run).

We can now give a sufficient condition for strong Zenoness of bounded rate hybrid machines as follows.

**Theorem 3** Assume $HM^L$ and $HM^U$ satisfy Assumption 1. If $K^U\alpha \geq 0$ has no solutions in $\mathcal{A}$, then $HM^o$ is strongly Zeno.

**Proof**
Suppose that $HM^o$ is not strongly Zeno, i.e., there exists a non-Zeno run of $HM^o$. By Lemma 5, there exists a non-Zeno run of $HM^U$. This means that $HM^U$ is not strongly Zeno. By Theorem 1, $K^U\alpha \geq 0$ has a solution in $\mathcal{A}$, a contradiction.

Next, we give a necessary condition for strong Zennoness.

**Theorem 4** Assume $HM^L$ and $HM^U$ satisfy Assumption 1, if $HM^o$ is strongly Zeno, then $K^L\alpha \geq 0$ has no solutions in $\mathcal{A}$.
Proof
Suppose that $K^L \alpha \geq 0$ has a solution in $A$. By Theorem 1, $HM^L$ is not strongly Zeno, i.e., there exists a non-Zeno run of $HM^L$. By Lemma 5, there exists a non-Zeno run of $HM^o$, which means that $HM^o$ is not strongly Zeno, a contradiction.

The above results deal with strong Zenoness. We now consider Zenoness in regular bounded rate hybrid systems, namely, systems in which $HM^L$ and $HM^U$ both satisfy Assumptions 1 and 2.

**Theorem 5** Assume $HM^L$ and $HM^U$ satisfy Assumptions 1 and 2. If $K^L \alpha \geq 0$ has a solution in $A$, then $HM^o$ is non-Zeno.

**Proof**
Suppose that $HM^o$ is Zeno. Let

$$r^o = q_0 \xrightarrow{e_1,t_1} q_1 \xrightarrow{e_2,t_2} q_2 \xrightarrow{e_3,t_3} \cdots$$

be a Zeno run of $HM^o$. This run will reach the Zeno point $(x = 0)$ at some finite time $t_Z$. Let

$$r^L = q_0 \xrightarrow{e_1,t_1} q_1 \xrightarrow{e_2,t_2} q_2 \xrightarrow{e_3,t_3} \cdots$$

be a run of $HM^L$ that switches at exactly the same times and to exactly the same configurations as $r^o$. (We assume, for the purpose of the proof, that in $HM^L$ the guards $[x_i \leq 0]$ are not in effect and the legal constraints $[x_i \geq 0], i = 1, 2, ..., n$, are not binding.)

It can then be shown, in similar fashion to the proof of Lemma 5, that $x^L(t) \leq x(t)$ for all $t \leq t_Z$. In particular, $x^L(t_Z) = K^L \alpha(t_Z)t_Z + x_0 \leq 0$, where $\alpha(t_Z) \in A$. This implies that $K^L \alpha(t_Z) \leq -x_0/t_Z < 0$, or that $K^L \alpha < 0$ has a solution in $A$. By Lemma 4, it then follows that $K \alpha \geq 0$ has no solution in $A$, a contradiction.

**Example 5** This simple example shows that the above conditions cannot be extended by much. Consider the bounded-rate hybrid machine that has two configurations in one loop, with

$$K^L = \begin{bmatrix} 1 & -5 \\ -1 & 1 \end{bmatrix} \quad \text{and} \quad K^U = \begin{bmatrix} 5 & -1 \\ -1 & 1 \end{bmatrix}.$$  

One can see that $K^L \alpha \geq 0$ has no solution in $A$, while $K^U \alpha \geq 0$ has solutions in $A$. The behavior of this system is hard to predict from knowing only $K^L$ and $K^U$. If the actual rates are given by $K^L$, then the system is strongly Zeno while, if the rates are given by $K^U$, then the system is non-Zeno.

26
6  Zenoness in Hybrid Systems with Nonlinear Dynamics

In the previous sections we investigated the occurrence of Zenoness in constant rate and in bounded rate hybrid machines. In this section we examine the possibility of Zeno behavior in a class of hybrid machines with nonlinear dynamics.

We consider systems in which the dynamics of variable $i$ in configuration $j$ is given by

$$\dot{x}_i = g_{ij}(x),$$

where the $g_{ij}(x), i = 1, 2, \ldots, m; j = 1, 2, \ldots, n$, are Lipschitz continuous functions of $x$. As before, we shall assume that the legal region is the nonegative orthant; that is, $x_i \geq 0$, $i = 1, \ldots, m$, and shall focus attention on the dynamic behavior of the system in the vicinity of the Zeno point $x = 0$.

Taking the Taylor expansion of $g_{ij}$, $i = 1, 2, \ldots, m; j = 1, 2, \ldots, n$, around $x = 0$, we obtain,

$$\dot{x}_i \approx g_{ij}(0) + \dot{g}_{ij}(0)x + \ddot{g}_{ij}(0)x^2/2 + \ldots,$$

which, if $g_{ij}(0) \neq 0$, behaves, for $x$ sufficiently close to 0, like

$$\dot{x}_i = g_{ij}(0).$$

But this latter equation represents a constant rate dynamic system and hence the Zenoness of the nonlinear hybrid system is determined by its constant rate approximation near the origin. Thus, our analysis of constant rate hybrid systems applies to a fairly large class of nonlinear systems as well. Specifically, if we assume that in addition to $g_{ij}(0) \neq 0$ for all $i$ and $j$, our nonlinear dynamic equations satisfy Assumption 1 or Assumption 1 and Assumption 2 for sufficiently small $x$, then the results of the previous sections apply.

7  Conclusion

In this paper we studied various issues concerning the possible existence of Zeno behaviors in hybrid systems and the related question of existence of safety controllers that satisfy specified state invariance constraints. We first focused our attention on constant rate hybrid machines, and showed that the existence of Zeno behaviors can be examined by checking for existence of solutions to a set of linear inequalities in a specified region of $\mathbb{R}^m$. In particular, we have shown that for the class of “regular” constant rate hybrid systems Zenoness is equivalent to strong Zenoness; that is, the system has Zeno runs if and only if all its runs are Zeno. In this case, it is clear that if Zeno runs exist, no safety controller exists.
When a system has Zeno runs but is not strongly Zeno, some legal controller exists. However we have shown that, contrary to earlier belief, the existence of a safety controller in Zeno systems does not always imply the existence of a minimally interventive (or minimally restrictive) controller. This implies, in particular, that the standard iterative synthesis algorithms that have been proposed in the literature may not apply in such cases. However, as was demonstrated, controllers can still be designed by more ad-hoc procedures.

We discussed some of the shortcomings of the approach for bypassing the problems associated with controller synthesis in Zeno systems that is based on introduction of switching delays. Specifically, the synthesis algorithm may not converge because of the introduced delay, and in cases where it converges, there may exist controllers less restrictive than the synthesized one.

We extended the study of strong Zenoness and existence of minimally interventive controllers to bounded-rate hybrid systems. Because of the nondeterminism in the dynamics of such systems, a gap appears between necessary conditions for Zenoness and sufficient conditions for Zenoness. Finally, we have shown that our analysis approach also applies to hybrid systems with nonlinear dynamics.

References


