Stability of HIOA

1 Introduction

2 Stability Theorems

In this section we state stability theorems for hybrid I/O automata(HIOA) without any input variables. Such systems are called *autonomous systems* in control theory. For this section we assume that a HIOA $\mathcal{A}$ consists of :

- A set Z of variables, which is partitioned into the set Y of *output* variables and the set X of *internal* or sate variables. Often it is convenient to think of a set of variables S as a vector (with elements of S in some arbitrary fixed order) and we denote this vector by $\mathbf{s}$.
- A set A of actions partitioned into *internal* H, *input* I, and *output actions* O.
- A set of states $Q \subseteq val(X)$.
- A non-empty set of *start states* $\Theta \subseteq Q$.
- A set of *discrete transitions* $\mathcal{D} \subseteq Q \times A \times Q$. A transition $(\mathbf{x}, a, \mathbf{x}') \in \mathcal{D}$ is written in short as $\mathbf{x} \xrightarrow{a}_{\mathcal{A}} \mathbf{x}'$.
- A set of *trajectories* $\mathcal{T}$ for Z, such that for every trajectory τ in $\mathcal{T}$, and for every $t \in \tau.dom$, $\tau(t).X \in Q$. It is required that $\mathcal{T}$ is closed under prefix, suffix, and concatenation. The first state $\tau(0).X$ of trajectory is denoted by $\tau.fstate$. If $\tau.dom$ is finite then $\tau.lstate = \tau(\tau.ltime).X$.

In addition, a hybrid I/O automaton also satisfies the input action enabling property, which prevents it from blocking any input action. However, the input trajectory enabling condition is not relevant here because there are no input variables.

We further assume:(a) all the variables in Z are either discrete or continuous. For a set of variables S, we denote its discrete and continuous subsets by S_d and S_c. (b) There is a family of sufficiently regular[1] functions $f_p : \mathsf{R}^n \to \mathsf{R}^n$, $p \in P$ such that every trajectory of $\mathcal{A}$ satisfy one of these functions. That is, along any trajectory $\tau \in \mathcal{T}$: $\dot{\mathbf{x}}_c = f_\sigma(\mathbf{x}_c)$ for some $\sigma \in P$. Here $\mathbf{x}_c$ is the vector of n continuous state variables of $\mathcal{A}$. (c) Discrete transitions *do not* change the valuation of the variables in $\mathbf{x_c}$, but it can change the value of σ.

An *execution fragment* of $\mathcal{A}$ is a finite or infinite sequence of actions and trajectories $\alpha = \tau_0, a_1, \tau_1, a_2 \ldots$, where each $\tau_i \in \mathcal{T}$, and if τ_i is not the last trajectory in α then τ_i is finite and $\tau_i.lstate \overset{a_{i+1}}{\to} \tau_{i+1}.fstate$. An *execution* is an execution fragment with $\tau_0.fstate \in \Theta$. An execution is *closed* if the sequence is finite and the domain of the final trajectory is a finite closed interval. The length of an execution is the number of elements (actions and trajectories) in the sequence. The limit time *ltime* of an execution is defined as $\alpha.ltime = \sum_{i=0}^{n} \tau_i.ltime$. With slight abuse of notation we denote the valuation of the continuous variables X_c at time t in the execution α by $\alpha(t)$. Note that since the discrete actions do not alter the valuation of the continuous variables there is an unique value of $\alpha(t)$ for every $t \in \alpha.ltime$.

[1] Locally Lipschitz

2.1 Some Definitions

We are assuming that the individual subsystems have origin as the common equilibrium point: $f_p(0) = 0$ for all $p \in P$. Now, we define what it means for the HIOA $\mathcal{A}$ to be stable.

The origin is said to be a *stable* equilibrium point of HIOA $\mathcal{A}$, in the sense of Lyapunov, if for every $\epsilon > 0$, there exists a $\delta > 0$, such that for every execution α of $\mathcal{A}$, we have

$$|\alpha(0)| \leq \delta \Rightarrow |\alpha(t)| \leq \epsilon \qquad \forall t \;\; 0 \leq t \leq \alpha.ltime, \tag{1}$$

and we say that $\mathcal{A}$ is *stable*. The HIOA $\mathcal{A}$ is *asymptotically stable* if it is stable and δ can be chosen so that

$$|\alpha(0)| \leq \delta \Rightarrow \alpha(t) \to 0 \quad as \quad t \to \infty \tag{2}$$

If the above condition holds for all δ then $\mathcal{A}$ is *globally asymptotically stable*. The HIOA $\mathcal{A}$ is said to be *exponentially stable* if there exist positive constants δ, c, and λ such that all executions with $|\alpha(0)| \leq \delta$ satisfy the inequality

$$|\alpha(t)| \leq c|\alpha(0)|e^{-\lambda t} \qquad \forall t \;\; 0 \leq t \leq \alpha.ltime. \tag{3}$$

If the above holds for all δ then $\mathcal{A}$ is said to be *globally exponentially stable*. *Uniform stability* is a concept which guarantees that the stability property in question holds, not just for executions (starting from initial states), but for any execution fragment. Therefore, $\mathcal{A}$ is uniformly stable in the sense of Lyapunov, if for every $\epsilon > 0$ there exists a constant $\delta > 0$, such that for any execution fragment α

$$|\alpha(t_0)| \leq \delta \Rightarrow |\alpha(t)| \leq \epsilon \quad \forall t_0, t, \;\; 0 \leq t_0 \leq t \leq \alpha.ltime \tag{4}$$

The HIOA $\mathcal{A}$ is said to be uniformly asymptotically stable if for all $\epsilon > 0$, there exists a $\delta > 0$ and a time T, such that for any execution fragment α,

$$|\alpha(t_0)| \leq \delta \Rightarrow |\alpha(t)| \leq \epsilon, \;\; \forall t \geq t_0 + T \tag{5}$$

It is said to be *globally uniformly asymptotically stable* if the above holds for all δ. Global and local uniform exponential stability are defined similarly.

2.2 Common Lyapunov Function

Definition 1. *Given a positive definite continuously differentiable function $V : \mathsf{R}^n \to \mathsf{R}^n$, we say that it is a* common Lyapunov fucntion *for the HIOA $\mathcal{A}$ if there exists a positive definite continuous fucntion $W : \mathsf{R}^n \to \mathsf{R}^n$, such that we have*

$$\dot{V}(\mathbf{x}_c) \leq -W(\mathbf{x}_c) \quad \forall \mathbf{x}_c, \quad \forall p \in P, \tag{6}$$

$$where \quad \dot{V}(x) = \frac{\partial V}{\partial x} f_p(x) \tag{7}$$

Theorem 1 (2.1 [3]). *If HIOA $\mathcal{A}$ has a radially unbounded common Lyapunov function then $\mathcal{A}$ is globally uniformly asymptotically stable.*

2.3 Multiple Lyapunov Functions

Let us define a function $M : \mathcal{T} \to P$ which gives the index p of the function f_p, which is active over the trajectory τ. In [3] and [2], σ is a piece-wise constant "switching signal" taking values on the index set P. We define the notion of switching signal and switching times with respect to hybrid executions. Whenever a discrete action a_i occurs such that $M(\tau_{i-1}) \neq M(\tau_i)$, the HIOA $\mathcal{A}$ is said to undergo a *switch*. For any execution fragment $\alpha = \tau_0 a_1 \tau_1 \ldots$, an instant of time $t \in \alpha.dom$ is called a *switching time* if there exists i such that $t = \tau_i.ltime$, and $M(\tau_i) \neq M(\tau_{i+1})$.

Theorem 2 (3.1 [3]). *Let V_p be the radially bounded Lyapunov function corresponding to the globally asymptotically stable system $\dot{x} = f_p(x)$ for each $p \in P$. If there exists a family of positive definite continuous functions $W_p, p \in P$ satisfying the following property along all executions,* *Consider successive pair of switching times t, t' and the corresponding trajectories τ_i and τ_j, such that $M(\tau_i) = M(\tau_j) = p$, and $M(\tau_k) \neq p, \forall\, k, i < k < j$, then $V_p(\tau_j(t')) - V_p(\tau_i(t)) \leq -W_p(\tau_i(t))$.*

Then $\mathcal{A}$ is globally uniformly asymptotically stable.

2.4 Stability Under Slow Switching : Dwell time and Average Dwell time

Definition 2. *Let $t_1, t_2, \ldots$ be the switching times of an execution α of HIOA $\mathcal{A}$. The execution α has a dwell time $\tau_d > 0$ if it satisfies the inequality $t_{i+1} - t_i \geq \tau_d$, for all i. If all executions from all starting states of $\mathcal{A}$ have dwell times $\geq \tau_d$ then HIOA $\mathcal{A}$ has a dwell time τ_d.*

Definition 3. *Let $N(\alpha)$ denote the number of switches over an execution α of HIOA $\mathcal{A}$. The execution has an average dwell time $\tau_a > 0$ if there exists a positive number N_0 such that:*

$$N(\alpha) \leq N_0 + \frac{\alpha.ltime}{\tau_a}. \tag{8}$$

If all executions from all starting states of $\mathcal{A}$ have average dwell times $\geq \tau_a$ then the HIOA has an average dwell time of τ_a.

The following theorem gives a sufficient condition for stability with respect to the average dwell time of the system. Since dwell time is a special case of average dwell time with $N_0 = 1$, a separate theorem for dwell time is not necessary.

Theorem 3 (3.2 [3]). *Consider HIOA $\mathcal{A}$ with its trajectories specified by a family of functions $f_p, p \in P$. Suppose there exist continuously differentiable functions $V_p : \mathsf{R}^n \to \mathsf{R}^n, p \in P$, two class $\mathcal{K}_\infty$ functions γ_1 and γ_2 and positive numbers λ_0 and μ such that:*

$$\gamma_1(|\mathbf{x}_c|) \leq V_p(\mathbf{x_c}) \leq \gamma_\mathbf{2}(|\mathbf{x_c}|), \quad \forall \mathbf{x_c}, \quad \forall \mathbf{p} \in \mathbf{P} \tag{9}$$

$$\frac{\partial V_p}{\partial \mathbf{x}_c} f_p(\mathbf{x}_c) \leq -\lambda_0 V_p(\mathbf{x}_c), \quad \forall \mathbf{x}_c, \quad \forall p \in P \tag{10}$$

$$V_p(\mathbf{x}_c) \leq \mu V_q(\mathbf{x}_c), \qquad \forall \mathbf{x}_x, \quad \forall p, q \in P. \tag{11}$$

Then $\mathcal{A}$ is globally asymptotically stable if it has an average dwell time $\tau_a > \frac{\log \mu}{2\lambda_0}$.

Theorem 3 roughly states that a hybrid system is stable if the discrete switches are between modes which are individually stable, provided that the switches do not occur *too frequently on the average.* In the next section we develop techniques for proving that a given HIOA satisfies this average dwell time criterion.

3 Invariant Proof

3.1 Overview

Consider a HIOA $\mathcal{A}$ with a single discrete variable *mode*, which takes values[2] from a finite set $\mathcal{P}$. When $mode = p$, $p \in \mathcal{P}$, we say that $\mathcal{A}$ *is in mode* p. In mode p, the trajectories of the continuous state and the output variables are described by:

$$\dot{x} \;=\; A_p x, \tag{12}$$

$$y \;=\; C_p x, \quad i \in \mathcal{P}. \tag{13}$$

Let the set of output actions be $O \subseteq \{a_{ij} \mid i \neq j, \;\; i, j \in \mathcal{P}\}$, discrete action a_{ij} occurs if and only if there is a mode switch from $mode = i$ to $mode = j$.

In order to apply Theorem 3 to automaton $\mathcal{A}$, we have to show that there exits τ_a (average dwell time) such that Equation (8) holds for every execution of $\mathcal{A}$. Since Equation (8) is a property of the executions of $\mathcal{A}$, namely the number of mode switches, and not an invariant property, we first transform $\mathcal{A}$ to a new automata $\mathcal{A}'$ in which the same property can be stated as an invariant.

[2]Each value of *mode* corresponds to a *location* in the sense of Alur-Henzinger Hybrid Automaton [1], and the continuous evolution corresponding to a particular value of *mode* can be written as an *activity* using the language constructs proposed in [5].

3.2 Transformation

We transform $\mathcal{A}$ to a new HIOA A' as follows: In addition to all the variables of A, automaton A' has two new internal variables; a counter Q and a timer t, both initialized 0. The counter Q counts the number of mode switches, and the timer reduces the count by 1 in every τ_a time. For every discrete transition $s \xrightarrow{a_{ij}}_{\mathcal{A}} s'$ of $\mathcal{A}$, automaton $\mathcal{A}'$ has a corresponding transition $s \xrightarrow{a_{ij}}_{\mathcal{A}'} s'$, such that $s'.Q = s.Q + 1$. In addition $\mathcal{A}'$ also has internal action a_{ii}, with pre-condition $t = \tau_a$, and effect $t := 0; Q := Q - 1$, for all $i \in \mathcal{P}$. Finally, for every trajectory τ' of $\mathcal{A}'$, the projection of τ on the set of continuous variables of $\mathcal{A}$ is a trajectory of $\mathcal{A}$, i.e., $\tau' \downarrow Z_c \in \mathcal{T}_{\mathcal{A}}$, and $\dot{t} = 1$. The transformation of a HIOA with $\mathcal{P} = \{1, 2, 3\}$ is shown in Figure 1.

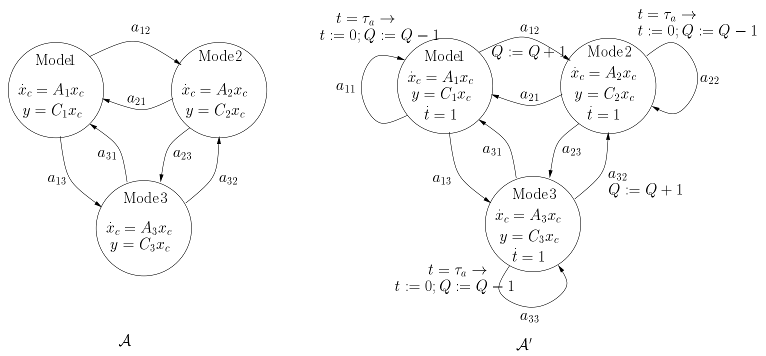

Figure 1: The transformation

Lemma 1. *All closed executions of $\mathcal{A}$ satisfy Equation (8) if and only if $Q \leq N_0$ in all reachable states of $\mathcal{A}'$.*

Proof. For the if part, consider a closed execution α of $\mathcal{A}$ and let α' be the "corresponding" execution of $\mathcal{A}'$. Let s' be the last state of α, therefore from the invariant we know that $s'.Q \leq N_0$. From construction of $\mathcal{A}'$ we know that, $N(\alpha) = N(\alpha')$ and $\alpha'.ltime = \alpha.ltime$ and therefore $s'.Q = N(\alpha') - \lfloor \frac{\alpha'.ltime}{\tau_a} \rfloor$. It follows that $N(\alpha) - \frac{\alpha.ltime}{\tau_a} \leq N_0$.

For the only if part, consider a reachable state s' of $\mathcal{A}'$. There exists an execution α' such that s' is the last state of α'. Let α be an execution of $\mathcal{A}$ "corresponding" to α'. Since $N(\alpha) \leq N_0 + \lfloor \frac{\alpha.time}{\tau_a} \rfloor$ implies $N(\alpha') \leq N_0 + \lfloor \frac{\alpha'.ltime}{\tau_a} \rfloor$, it follows that $s'.Q \leq N_0$. □

Theorem 4. *All executions of $\mathcal{A}$ satisfy Equation (8) if and only if $Q \leq N_0$ in all reachable states of $\mathcal{A}'$.*

Proof. We show that all executions of $\mathcal{A}$ satisfy Equation (8) if and only if all closed executions of $\mathcal{A}$ satisfy (8). The reverse direction is trivial. For the forward direction, we have to show that if any execution α of $\mathcal{A}$ violates (8), then there exists a closed execution α' of $\mathcal{A}$ that violates (8) as well. First, note that if α is infinite, then there is a closed prefix of α that violates (8). Secondly, if α is finite and open, then the closed prefix of α excluding the last trajectory of α, also violates (8). □

3.3 Remarks

1. We can now prove the average dwell time criterion for $\mathcal{A}$ by proving the invariant $Q \leq N_0$ of A'. Not that this new problem is *easier*, but in this form we can use the standard invariant proof techniques in the HIOA framework. In some extreme cases (linear, rectangular, etc.) this can be verified by model-checking, for example.

2. In general N_0 is not known and the objective is to show that Q is bounded, however, for applying the invariant proof techniques we need to know a bound on N_0.

3. The above transformation allows Q to be negative, and then rapidly return to zero. This is acceptable for asymptotic stability, but does not guarantee uniform stability. For uniform stability we want all execution fragments of $\mathcal{A}$ to satisfy (8). That is,

$$\begin{aligned} N(t,t_0) &\leq N_0 + \frac{t-t_0}{\tau_a} \\ Q(t,t_0) &\leq N_0 \end{aligned}$$

That is, we need to show that the total change in Q between any two reachable states is bounded by N_0 (some constant). We can prove this by introducing an additional variable Q_{min}, initially set to zero, which stores the largest negative value Q ever gets in an the execution. Then, $Q(t,t_0) \leq N_0$ in $\mathcal{A}$ if and only if $Q - Q_{min} \leq N_0$ in all reachable states of $\mathcal{A}'$.

3.4 Case Study: Hysteresis Switching

In this section we apply the technique described in Section 3 to an example which is a part of a supervisory control system taken from Chapter 6 of [3]. We consider a finite set of continuous monitoring signals μ_p, $p \in \mathcal{P}$ satisfying the following properties:

$$\begin{aligned} \mu_p(0) &\geq \xi_\mu && (14) \\ \mu_{p^*}(t) &\leq \bar{\mu}_{p^*}(0) + C_1 + \xi_\mu e^{2\lambda t}, \quad \text{for some } p^* \in \mathcal{P} && (15) \end{aligned}$$

where $\bar{\mu_p}, C_1, \xi_\mu$ are positive constants for each $p \in \mathcal{P}$. Equation (14) gives a lower bound on the initial values of all the monitoring signals, and Equation (15) states that there exists some $p^* \in \mathcal{P}$ which satisfies the exponential upper bound.

The switching logic unit implements the scale independent switching based on the monitoring signals μ_p, $p \in \mathcal{P}$. At an instant of time when the input to the plant $u = u_\sigma$, for some $\sigma \in \mathcal{P}$, if there exists a $p \in \mathcal{P}$ such that $\mu_p(1+h) \leq \mu_\sigma$ for some fixed hysteresis constant h, then the supervisory controller *switches to* $\sigma = p$ and the plant input is set to $u = u_p$.

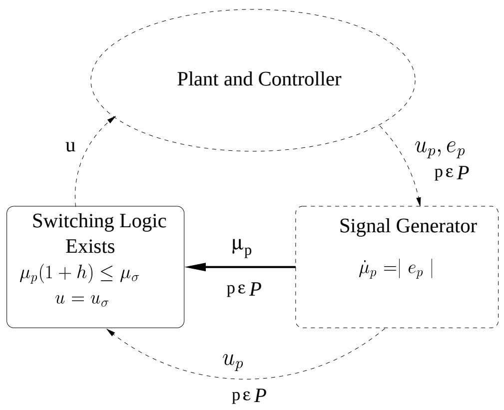

Figure 2: HIOA model of Supervisory Controller: Plant, controller, monitoring signal generator, and the hysteresis switching logic.

We are interested in the switching logic unit of Figure 2, since this is part which switches between the different controllers and can potentially violate the dwell time criterion.

3.4.1 Description of the HIOA

The HIOA model of the hysteresis switch is shown in Figure 3. The automaton receives the continuous monitoring signals as the input variables μ_p, $p \in \mathcal{P}$, and produces the discrete(piece-wise constant) signal σ

hybridautomaton `SwitchedController`(h:`PosReal`, $\mathcal{P}$:`IndexSet`)

signature
  output $switch_p$ for each $p \in \mathcal{P}$,
  internal $dequeue$

transitions
  output $switch_p$ for each $p \in \mathcal{P}$
  pre $(1+h)\mu_p \le \mu_\sigma$
  eff $\sigma := p$; $c := c + 1$;
    $c_p := c_p + 1$; $\mu_p^{c_p} := \mu_p$; $t_p := 0$

  internal $dequeue$
  pre $now = k\tau_a$
  eff $d := d + 1$

variables
  input analog μ_p : `Real` for each $p \in \mathcal{P}$,
  output $\sigma : \mathcal{P}$, initially $\sigma = arg\ min_{p\in\mathcal{P}}\mu_p$,
  internal analog now : `Real`, initially 0,
  internal c, d : `Int`, initially 0,
  internal μ_p^i : `Real` $\cup \{\perp\}$, for $p \in \mathcal{P}$ and $i \in \{0, 1, 2, \ldots\}$,
    initially $\mu_p^0 = \mu_p$, $\mu_\sigma^1 = \mu_\sigma$, otherwise $\mu_p^i = \perp$,
  internal c_p : `Int`, initially 0, for $p \neq \sigma$, and $c_\sigma := 1$
  internal t_p : `Real`, initially 0, for each $p \in \mathcal{P}$,

derived variables
  const m : `Int` $= |\mathcal{P}|$
  Q : `Int` $= c - d$

trajectories
  activity *flow*
    evolve $d(now) := 1$; $d(t_p) := 1$
    stop at $(\exists p, (1+h)\mu_p \le \mu_\sigma) \vee (now = k\tau_a)$

Figure 3: HIOA specification hysteresis switching in supervisory controller

as output. The value of σ changes to p whenever the action $switch_p$ is enabled. We also have variable c and d which count the number of mode switches and the number of τ_a periods elapsed. Together, c and d define Q, the number of "extra switches" (corresponding to Q of Section 3.2). In addition we have introduced some additional variables to facilitate the invariant proofs. The discrete variables μ_p^i store the values of μ_p at the instants when σ became equal to p for the i^{th} time. The variable c_p counts the number of intervals in which $\sigma = p$, and the real variable t_p is a reset timer measuring how long $\sigma = p$, most recently.

3.4.2 Invariants

In this section we prove a sequence of invariant properties which lead to the establishment of the average dwell time criterion for the `SwitchedController` automaton, which in turn establishes its stability. If the observation made in Remark 3 of Section 4.3 is correct, then (I think) we can tweak these invariants to get uniform stability as well.

Invariant 1. $Q \le c - \frac{now}{\tau_a} + 1$.

Proof. Initial states satisfy. The $switch_q$ action does not affect the invariant. Consider the *dequeue* action $s \xrightarrow{a} s'$. We know $s'.Q = s.Q - 1$ and the right hand side of the inequality does not change, therefore the invariant is preserved. All trajectories preserve the invariant because only the right hand side of the inequality increases. □

Invariant 2. *When the system is in mode q, then μ_q is less than or equal to $(1+h)$ times μ_p for all p's in $\mathcal{P}$. In addition, at the instant when σ changes to a new value q, μ_q is less than or equal to μ_p for all $p \in \mathcal{P}$. Formally,*

$$(1) \qquad \forall\, q \in \mathcal{P},\ \sigma = q \Rightarrow \forall\, p \in \mathcal{P}, \mu_q \le (1+h)\mu_p, \tag{16}$$

$$(2) \qquad \forall\, q \in \mathcal{P},\ \sigma = q \wedge c_q > 0 \wedge t_q = 0 \Rightarrow \forall\, p \in \mathcal{P}, \mu_q \le \mu_p. \tag{17}$$

Proof. Part(1): Initial states satisfy. For the induction step we need to consider only discrete transitions $s \xrightarrow{a} s'$, where $a = switch_q$. Let $s.\sigma = r$, we know that $s'.\sigma = q$.

$$\text{By inductive hypothesis} \qquad s.\mu_r \le (1+h)s.\mu_p, \quad \forall p \in \mathcal{P} \tag{18}$$

$$\begin{aligned} \text{by precondition of } switch_q \quad & (1+h)s.\mu_q \leq s.\mu_r, \\ \text{by continuity of } \mu_p\text{'s and (18)} \quad & (1+h)s'.\mu_q \leq s'.\mu_r \leq (1+h)s'.\mu_p, \quad \forall p \in \mathcal{P} \end{aligned} \tag{19}$$

From (19) it follows that $s'.\mu_q \leq (1+h)s'.\mu_p$, for all $p \in \mathcal{P}$. The stopping condition of activity *flow* ensures that the invariant is preserved over all trajectories. For any trajectory τ, $\tau(t).\mu_\sigma \leq (1+h)\tau(t).\mu_p$, for all $t \in \tau.dom$ and for all $p \in \mathcal{P}$, because if there exists a p such that $\tau(t).\mu_\sigma = (1+h)\tau(t).\mu_p$, then $t = \tau.ltime$.

Part(2) : Initial states satisfy the invariant because $q = arg\ min_{p\in\mathcal{P}}\mu_p$. For the induction step, consider a discrete transition $s \xrightarrow{a} s'$, where $a = switch_q$. Let $s.\sigma = r$, we know that $s'.\sigma = q$.

$$\begin{aligned} \text{From Part (1)} \quad & s.\mu_r \leq (1+h)s.\mu_p, \quad \forall p \in \mathcal{P} && (20) \\ \text{by precondition of } switch_q \quad & (1+h)s.\mu_q \leq s.\mu_r, \\ \text{by continuity of } \mu_p's \text{ and (20)} \quad & s'.\mu_q \leq s'.\mu_p, \quad \forall p \in \mathcal{P}. \end{aligned}$$

We note that the *dequeue* actions do not alter any of the variables involved in the invariant. Now, consider any trajectory τ. If τ is a point trajectory, then the invariant holds. If τ is not a point trajectory, then the invariant holds vacuously because $\tau.lstate.t_q \neq 0$. □

Corollary 1. *If σ equals some r in $\mathcal{P}$ and the stopping condition of flow holds for some q in $\mathcal{P}$ then $\mu_r = (1+h)\mu_q$. Formally, $\forall\ q, r \in \mathcal{P},\ \sigma = r \wedge (1+h)\mu_q \leq \mu_r \Rightarrow (1+h)\mu_q = \mu_r$.*

Proof. From Invariant 2 Part(1), $\mu_r \leq (1+h)\mu_q$, and we have $\mu_r \geq (1+h)\mu_q$, therefore $\mu_r = (1+h)\mu_q$. □

Invariant 3. *For all q in $\mathcal{P}$, μ_q^{i+1} is greater than or equal to $(1+h)$ times μ_q^i.*
Formally, $\forall\ q \in \mathcal{P},\ c_q \geq 2 \Rightarrow \mu_q^{c_q} \geq (1+h)\mu_q^{c_q-1}$.

Proof. Initial states have $c_p \leq 1$ for all $p \in \mathcal{P}$. For the induction step we only have to consider discrete transitions of the form $s \xrightarrow{a} s'$, where $a = switch_q$. Let $s.\sigma = r$, $s.c_q = k+1$, that is, a is the $(k+1)^{st}$ $switch_q$ action.

$$\text{From the transition relation, and Corollary 1} \quad s'.\mu_q^{k+1} = s'.\mu_q = (1+h)s'.\mu_r \tag{21}$$

Let s'' be the post state of the k^{th} $switch_q$ action (ugly!).

$$\begin{aligned} \text{From Invariant 2 Part(1)} \quad & s''.\mu_q = s''.\mu_q^k \leq (1+h)s''.\mu_r && (22) \\ \text{from monotonicity of } \mu_r \quad & s''.\mu_r \leq s'.\mu_r. \\ \text{Since } \mu_q^k \text{ is not changed after } s'' \quad & s'.\mu_q^k \leq s'.\mu_r && (23) \\ \text{combining (21) and (23)} \quad & s'.\mu_q^{k+1} \geq (1+h)s'.\mu_q^k \end{aligned}$$

□

Corollary 2. $\forall\ q \in \mathcal{P}, \mu_q^{c_q} \geq (1+h)^{c_q-1}\mu_q^1$.

Proof. Follows from Invariant 3. □

Invariant 4. *There exists q in $\mathcal{P}$ such that c_q is at least $\lceil \frac{c-1}{m} \rceil$.*

Proof. We observe that, the counter c is incremented every time a $switch_p$ action occurs for any $p \in \mathcal{P}$, and for each $p \in \mathcal{P}$ the counter c_p is incremented when the corresponding $switch_p$ action occurs. Since there are m possible values of p, it follows that at least $\lceil \frac{c-1}{m} \rceil$ out of the c *switch* actions would correspond to some q in $\mathcal{P}$. □

Invariant 5. *If we select the average dwell time τ_a to be $\frac{\log(1+h)}{2\lambda m}$ then the number of extra switches Q is bounded by*

$$Q \leq 2 + m + \frac{m}{\log(1+h)} \log\left(\frac{\bar{\mu}_{p^*}(0) + C_1}{\xi_\mu} + 1\right) \tag{24}$$

Proof. Consider any reachable state s, from Corollary 2, Invariant 4 and monotonicity of μ_p's we know that there exists q in $\mathcal{P}$ such that $s.\mu_q^{c_q} \geq (1+h)^{\lceil \frac{c-1}{m} \rceil - 1} s.\mu_q^1$. Taking logarithm and rearranging we have,

$$s.c \leq 1 + m + \frac{m}{\log(1+h)} \log\left(\frac{s.\mu_q^{c_q}}{s.\mu_q^1}\right)$$

Let s' be the post state of the c_p^{th} $switch_q$ action, then $s.\mu_q^{c_p} = s'.\mu_q$. From Invariant 2 Part(2) and monotonicity it follows that $s.\mu_q^{c_q} = s'.\mu_q^{c_q} \leq s'.\mu_l \leq s.\mu_l$. It follows that,

$$s.c \leq 1 + m + \frac{m}{\log(1+h)} \log\left(\frac{s.\mu_l}{s.\mu_q^1}\right), \quad \forall l \in \mathcal{P}$$

Form monotonicity and property (14) of the monitoring signals, $\mu_q^1 \geq \mu_q^0 \geq \xi_\mu$. Therefore,

$$s.c \leq 1 + m + \frac{m}{\log(1+h)} \log\left(\frac{s.\mu_l}{\xi_\mu}\right), \quad \forall\, l \in \mathcal{P}$$

Replacing l with p^* of (15), we get

$$\begin{aligned} s.c &\leq 1 + m + \frac{m}{\log(1+h)} \log\left(\frac{\bar{\mu}_{p^*}(0) + C_1 + \xi_\mu e^{2\lambda\ s.now}}{\xi_\mu}\right) \\ &\leq 1 + m + \frac{m}{\log(1+h)} \log\left(\frac{\bar{\mu}_{p^*}(0) + C_1}{\xi_\mu} + 1\right) + \frac{2\lambda m\ s.now}{\log(1+h)} \end{aligned}$$

Using Invariant 1, and putting $\tau_a = \frac{\log(1+h)}{2\lambda m}$,

$$\begin{aligned} s.Q &\leq 2 + m + \frac{m}{\log(1+h)} \log\left(\frac{\bar{\mu}_{p^*}(0) + C_1}{\xi_\mu} + 1\right) + \frac{2\lambda m\ s.now}{\log(1+h)} - \frac{s.now}{\tau_a} \\ &\leq 2 + m + \frac{m}{\log(1+h)} \log\left(\frac{\bar{\mu}_{p^*}(0) + C_1}{\xi_\mu} + 1\right) \end{aligned}$$

□

Theorem 5. *The supervisory controller has an average dwell time of* $\frac{\log(1+h)}{2\lambda m}$ *and is stable.*

Proof. Follows from Invariant 5 and Theorem 4. □

4 Integer Programming

4.1 Overview

In this approach we try to prove that the given HIOA $\mathcal{A}$ does *not* have an average dwell time of τ_a for any constant N_0, by constructing an execution α of $\mathcal{A}$ with $N(\alpha) \geq N_0 + \frac{\alpha.ltime}{\tau_a}$, for any N_0. If the procedure fails to produce such a counter-example then we have proved that the average dwell time of $\mathcal{A}$ is τ_a.

We note that, an execution α can serve as the above counter-example only if it is an infinite execution. If α is finite, there are finitely many mode switches, and by scaling the constant N_0 we can make $N(\alpha) \leq N_0 + \alpha.ltime/\tau_a$. In general, it is not possible to find α automatically, however, for a restricted class of HIOA, it is possible to ensure that if α exists then it will have a finite description, in which case we can hope to have an algorithm for finding it.

The intuition is that, we can abstract the infinite state space of $\mathcal{A}$ by a set $\mathcal{C}$ of *equivalence classes*. Every state in class $c \in \mathcal{C}$ satisfies the same set of enabling conditions for the discrete transitions and mode invariants and takes the same next step (transition or trajectory). We can think of HIOA $\mathcal{A}$ equivalently as a labelled graph, with the set of nodes $\mathcal{C}$, and an edge labelled λ between nodes c_i and c_j, if and only if action (or activity) λ takes every state in c_1 to some state in c_2.

Now, for any execution of $\mathcal{A}$, we can construct a sequence of classes as follows: replace each action by the class of the prestate, and each τ trajectory by a sequence of classes $c_1, c_2, \ldots$, such that, $\tau.fstate \in c_1$, and for all i, $c_i \neq c_{i+1}$, and there exists $t \in \tau.dom$, such that $\tau(t) \in c_i$ and for all $0 < \epsilon \leq t2$, $\tau(t+\epsilon) \in c_{i+1}$. By this construction, any infinite execution of $\mathcal{A}$ corresponds to an infinite sequence of classes. Since $\mathcal{C}$ is finite, therefore any infinite sequence should contain a finite sub-sequence which repeats. HIOA $\mathcal{A}$ violates the dwell time criterion if an only if it has an infinite execution with *many extra switches*. Such an execution corresponds to an infinite sequence of classes with *many extra switches*, and such a sequence of classes exits if and only if there exits a *repeatable finite sequence of classes with extra switches*.

4.2 A Simple Example

Consider the following trivial HIOA $\mathcal{A}$(Figure 4), with two modes $\mathcal{M} = \{A, B\}$, a single real variable $x \in \mathsf{R}$, and two discrete transitions $\mathcal{D} = \{a, b\}$. The mode invariants and the action preconditions of $\mathcal{A}$ split up R^1 into five segments $\mathcal{S} = \{S_1, \ldots, S_5\}$ (Figure 5). The set of equivalence classes for the problem is $\mathcal{C} = \mathcal{S} \times \mathcal{M} \times \mathcal{D}$. Lets choose $n = ||\mathcal{C}||$, this is rather conservative. So we will try to find a sequence of classes of length n which maximizes the number of mode switches and minimizes the duration of the execution.

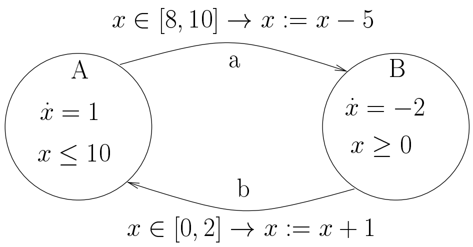

Figure 4: A trivial HIOA

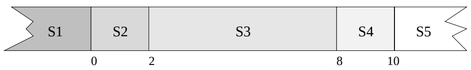

Figure 5: Disjoint sections of the continuous state-space for HIOA $\mathcal{A}$.

We define the decision variables:

1. $s_i \in \mathcal{M} \times \mathsf{R}$ — For $i = 0, 1, \ldots n$, it is the state in the execution fragment of $\mathcal{A}$ corresponding to the last state of the i^{th} class in the sequence. s_0 is the first state of the execution fragment.

2. $c_{ij} \in \{0, 1\}$ — For $i = 1, \ldots, n$, $j \in \mathcal{S}$, $c_{ij} = 1$ if the i^{th} class in the sequence is in S_j, $c_{ij} = 0$ otherwise.

3. $m_{ij} \in \{0, 1\}$ — For $i = 1, \ldots, n$, $j \in \mathcal{M}$, $m_{ij} = 1$ if the i^{th} class in the sequence is in mode j, $m_{ij} = 0$ otherwise.

4. $d_{ij} \in \{0, 1\}$ — For $i = 1, \ldots, n$, $j \in \mathcal{D}$, $d_{ij} = 1$ if the j^{th} class in the sequence is an action j, $d_{ij} = 0$ otherwise.

5. $t_i \in \mathsf{R}$ — For $i = 1, \ldots, n$ is the duration of time spent in the i^{th} in the sequence. If the class corresponds to a discrete transition $d_{i*} \neq 0$ then $t_i = 0$.

The objective function we want to maximize is:

$$\sum_i \sum_j d_{ij} - \frac{1}{\tau_a} \sum_i t_i \sum_j m_{ij} \tag{25}$$

The first term sums the number of switches over the entire sequence, and the second term is the number of allowed switches with dwell time τ_a. Suppose the optimal value found is greater then zero, then this execution sequence could be repeated over and over to get an unbounded number of switches, precisely what violates the average dwell time criterion.

The following constraints are imposed on the integer program:

1. Each class in the sequence corresponds to exactly one mode or exactly one discrete transition
$$\sum_{\mathrm{J}\in\mathcal{M}} m_{ij} + \sum_{j\in\mathcal{D}} d_{ij} = 1, \quad \forall\ i. \tag{26}$$

2. Each class in the sequence corresponds to exactly one of the $S_j \in \mathcal{S}$.
$$\sum_{j} c_{ij} = 1, \quad \forall\ i. \tag{27}$$

3. Every s_i should satisfy the precondition or the mode invariant of the class it is in.
$$\left(\sum_{i\in\mathcal{D}} d_{ij} L_j\right) \leq s_i.x \leq \left(\sum_{j\in\mathcal{D}} d_{ij} U_j\right), \forall i, \tag{28}$$
where $L = [0\ \ 8], U = [2\ \ 10]$. Similarly for the mode invariants.

4. The sates s_i at the beginning of class i and state s_{i+1} should follow the conditions set by class i. If the i^{th} class corresponds to an action step a then s_{i+1} should be the post-state of action a on s_i. If the i^{th} class corresponds to a trajectory step τ, then s_{i+1} is the *lstate* of a trajectory of length t_i starting from s_i.
$$s_{i+1}.x = s_i.x + \left(t_i \sum_{j\in\mathcal{M}} m_{ij} M_j\right) + \left(\sum_{j\in\mathcal{D}} d_{ij} D_j\right), \forall\ i \tag{29}$$
where $M_A = 1; M_B = -2; D_a = -5; D_b = 1$.

5. Repeatability: The sequence of classes is repeatable if the first and the last classes are the same.
$$c_{1j} = c_{nj}, \quad d_{1j} = d_{nj}, \quad m_{n1} = m_{nj}, \qquad \forall j \tag{30}$$

If we solve the above integer program and the optimal value is positive, then we have found an finite repeatable execution fragment with more switches than allowed by (8). In addition if s_0 is reachable then we have a counter-example. On the other hand, if the optimal value is non-positive, then we can conclude that there is no counter-example and the average dwell time condition is satisfied.

4.2.1 Remarks

1. The above formulation of the integer program searches of class sequences of length n only. This can be fixed to look for sequences with length at most n.

2. The starting state of the sequence s_0 may not be reachable !! I don't know how to fix this. But the failure to find a suitable solution still shows that the dwell time criterion is satisfied.

References

[1] Rajeev Alur, Costas Courcoubetis, Nicolas Halbwachs, Thomas A. Henzinger, P.-H. Ho, Xavier Nicollin, Alfredo Olivero, Joseph Sifakis, and Sergio Yovine. The algorithmic analysis of hybrid systems. *Theoretical Computer Science*, 138(1):3–34, 1995.

[2] J. Hespanha and A. Morse. Stability of switched systems with average dwell-tim. Technical report, University of Southern California / Lab. for Control Science & Eng., Yale University, Jan 1999.

[3] Daniel Liberzon. *Switching in Systems and Control*. Systems and Control: Foundations and Applications. Birkhauser, Boston, Jan 2003.

[4] Sayan Mitra, Yong Wang, Nancy Lynch, and Eric Feron. Application of hybrid I/O automata in safety verification of pitch controller for model helicopter system. Technical Report MIT-LCS-TR-880, MIT Laboratory for Computer Science, Cambridge, MA 02139, January 2003. http://theory.lcs.mit.edu/mitras/research/QuanTR02.ps.

[5] Sayan Mitra, Yong Wang, Nancy Lynch, and Eric Feron. Safety verification of model helicopter controller using hybrid Input/Output automata. In *HSCC'03, Hybrid System: Computation Control*, Prague, the Czech Republic, April 3-5 2003. Also, long version in [4].

[6] Morse A. S. Supervisory control of families of linear setpoint controllers-part1: Exact matching. *IEEE Trans. Automatic Control*, 41(7):1413–1431, 1996.