Hybrid automaton: execution

HYBRID AUTOMATA: EXECUTION

To define an execution of a hybrid system, we need to introduce first the notions of

• hybrid time set
• hybrid trajectory
  (which includes hybrid time set among its components)

We shall see that
an execution of a hybrid automaton $H$ is a hybrid trajectory
that is "consistent" with the definition of $H$
A hybrid time set is a finite or infinite sequence of intervals \( \tau = \{ I_i, i=0,1,\ldots, M \} \) such that

- \( I_i = [\tau_i, \tau_i'] \) for \( i < M \)
- \( I_M = [\tau_M, \tau_M'] \) or \( I_M = [\tau_M', \tau_M) \) if \( M < \infty \)
- \( \tau_i' = \tau_{i+1} \)
- \( \tau_i \leq \tau_i' \)

\( \tau \) represent times of discrete transitions

consecutive intervals, without gaps

intervals can be degenerate

to represent multiple transitions at the same time
A hybrid time set is a finite or infinite sequence of intervals \( \tau = \{ I_i, i=0,1,\ldots, M \} \) such that

- \( I_i = [\tau_i, \tau_i'] \) for \( i < M \)
- \( I_M = [\tau_M, \tau_M'] \) or \( I_M = [\tau_M, \tau_M'] \) if \( M < \infty \)
- \( \tau_i' = \tau_{i+1} \)
- \( \tau_i \leq \tau_i' \)

The elements of \( \tau \) are linearly ordered:

\[ t_1 < t_2 < t_3 < t_4 \]

HYBRID TIME SET: LENGTH

Two notions of length for a hybrid time set \( \tau = \{ I_i, i=0,1,\ldots, M \} \):

- Discrete extent:
  \[ <\tau> = M+1 \quad \text{number of discrete transitions} \]
- Continuous extent:
  \[ ||\tau|| = \sum_{i=0,1,\ldots, M} |\tau_i' - \tau_i| \quad \text{total duration of intervals in } \tau \]

\[ <\tau> = 4 \]
\[ ||\tau|| = \tau_3' - \tau_0 \]
HYBRID TIME SET: CLASSIFICATION

A hybrid time set $\tau = \{I_i, i=0,1,\ldots, M\}$ is

• Finite: if $\langle \tau \rangle$ is finite and $I_M = [\tau_M, \tau_M']$
• Infinite: if $\langle \tau \rangle$ is infinite or $||\tau||$ is infinite
• Zeno: if $\langle \tau \rangle$ is infinite but $||\tau||$ is finite

HYBRID TRAJECTORY

A hybrid trajectory is a triple $(\tau, q, x)$ that consists of:

• A hybrid time set $\tau = \{I_i, i=0,1,\ldots, M\}$
• Two sequences of functions $q = \{q_i(\cdot), i=0,1,\ldots, M\}$ and $x = \{x_i(\cdot), i=0,1,\ldots, M\}$ such that
  
  $q_i: I_i \rightarrow Q$
  
  $x_i: I_i \rightarrow X$
HYBRID TRAJECTORY

A hybrid trajectory is a triple \((\tau, q, x)\) that consists of:

- A hybrid time set \(\tau = \{I_i, i=0,1,\ldots, M\}\)
- Two sequences of functions \(q = \{q_i(\cdot), i=0,1,\ldots, M\}\) and \(x = \{x_i(\cdot), i=0,1,\ldots, M\}\) such that:
  - \(q_i: I_i \rightarrow Q\)
  - \(x_i: I_i \rightarrow X\)

Remarks:
- mixture of the two notions of continuous and discrete evolution
- the hybrid signals \(x\) and \(q\) can take multiple values at the same time instant

HYBRID AUTOMATA: EXECUTION

A hybrid trajectory \((\tau, q, x)\) is an execution (solution) of the hybrid automaton \(H = (Q,X,f,\text{Init},\text{Dom},E,G,R)\) if it satisfies the following conditions:

- Initial condition: \((q_0(\tau_0), x_0(\tau_0)) \in \text{Init}\)
- Continuous evolution:
  - for all \(i\) such that \(\tau_i < \tau_i'\)
    - \(q_i: I_i \rightarrow Q\) is constant
    - \(x_i: I_i \rightarrow X\) is the solution to the ODE associated with \(q_i(\tau_i)\)
    - \(x_i(t) \in \text{Dom}(q_i(\tau_i)), t \in [\tau_i, \tau_i')\)
- Discrete evolution:
  - \((q_i(\tau_i),q_{i+1}(\tau_{i+1})) \in E\) transition is feasible
  - \(x_i(\tau_i) \in G(q_i(\tau_i),q_{i+1}(\tau_{i+1}))\) guard condition satisfied
  - \(x_i(\tau_{i+1}) \in R((q_i(\tau_i),q_{i+1}(\tau_{i+1})),x_i(\tau_i))\) reset condition satisfied
EXAMPLE: THERMOSTAT

\[ Q = \{ \text{OFF, ON} \}; \quad X = \mathbb{R}; \]
\[ f(\text{OFF}, x) = -0.2x; \quad f(\text{ON}, x) = -0.2x + 6 \]
\[ \text{Init} = \{(\text{OFF}, x): x \geq 18\} \cup \{(\text{ON}, x): x \leq 22\} \]
\[ \text{Dom}(\text{OFF}) = [18, \infty); \quad \text{Dom}(\text{ON}) = (-\infty, 22] \]
\[ E = \{ (\text{OFF}, \text{ON}), (\text{ON}, \text{OFF}) \} \]
\[ G((\text{OFF}, \text{ON})) = (-\infty, 18]; \quad G((\text{ON}, \text{OFF})) = [22, \infty) \]
\[ R(e, x) = \{x\} \text{ for any } e \in E \]
EXAMPLE: THERMOSTAT

- Execution accepted by the thermostat hybrid automaton
- Infinite hybrid time set
- Evolution of the heater status starting from the initial condition \( x(0) = 5 \) with the heater ON
- Evolution of the temperature \( x \) starting from the initial condition \( x(0) = 5 \) with the heater ON

HYBRID AUTOMATA EXECUTIONS

What can go wrong?
HYBRID AUTOMATA EXECUTIONS

What can go wrong?

Problems of the ODE solution (existence, uniqueness, finite escape) avoided by the globally Lipschitz assumption

Problems due the hybrid nature!!!

- Zeno
- chattering
- blocking
- nondeterministic
HYBRID EXECUTION: ZENO

Let \((\tau, q, x)\) be an execution of \(H\).

\((\tau, q, x)\) is called **Zeno execution** if

\(\tau = \{ I_i, i=0,1,\ldots, M \} \) is Zeno (infinite number of discrete transitions in finite time)

EXAMPLE: BOUNCING BALL

\[
\begin{align*}
x_1 &= 0 \land x_2 < 0 \\
x_2 &= -cx_2 \\
x_1 &= x_2 \\
\dot{x}_2 &= -g \\
x_1 &> 0 \lor (x_1 = 0 \land x_2 \geq 0) \\
c &\in (0,1)
\end{align*}
\]
EXAMPLE: BOUNCING BALL

\[
x_1 = 0 \wedge x_2 < 0
\]

\[
x_1 \geq 0
\]

\[
x_2 := -cx_2
\]

\[
\dot{x}_1 = x_2
\]

\[
\dot{x}_2 = -g
\]

\[
x_1 > 0 \lor (x_1 = 0 \land x_2 \geq 0)
\]

\[
c \in (0,1)
\]

\[
x_1(0) = 0, \ x_2(0) > 0
\]

\[
\tau_0^* = 2 \frac{x_2(0)}{g}
\]
HYBRID EXECUTION: CHATTERING

Let \((\tau, q, x)\) be an execution of \(H\).

\((\tau, q, x)\) is called chattering execution if

- \(\tau = \{l_i, i=0,1,…, M\}\) is Zeno (infinite number of discrete transitions in finite time)
- After some \(k \geq 0\), all intervals \(l_i, i \geq k\), are singletons

\[
\begin{align*}
q_1 & : \dot{x} = 1, \quad x \leq 0 \\
q_2 & : \dot{x} = -1, \quad x \geq 0
\end{align*}
\]
HYBRID EXECUTION: CHATTERING

Let \((\tau, q, x)\) be an execution of \(H\).

\((\tau, q, x)\) is called chattering execution if

- \(\tau = \{I_i, i=0,1,\ldots, M\}\) is Zeno (infinite number of discrete transitions in finite time)
- After some \(k \geq 0\), all intervals \(I_i, i \geq k\), are singletons

\[
\begin{align*}
\tau &= \{[0,1],[1,1],[1,1],\ldots\} \\
q &= \{q_1,q_2,q_1,q_2, \ldots\} \\
x &= \{t-1, 0, 0, 0, \ldots\}
\end{align*}
\]

same problem of the sign function, can be solved by using a different notion of solution to ODE
ZENO HYBRID AUTOMATA

A hybrid automaton is Zeno if it accepts some Zeno execution

• difficult to simulate and analyse
• regularization methods to eliminate Zeno behavior

TEMPORAL REGULARIZATION: BOUNCING BALL

• each bounce takes $\varepsilon > 0$ time units
• an infinite number of discrete transitions cannot occur in finite time
TEMPORAL REGULARIZATION: BOUNCING BALL

**Bounce**
- \( \dot{x}_1 = 0 \)
- \( \dot{x}_2 = 0 \)
- \( \dot{x}_3 = 1 \)
- \( x_3 \leq \epsilon \)
- \( x_3 := 0 \)
- \( x_3 \geq \epsilon \)

**Fly**
- \( \dot{x}_1 = x_2 \)
- \( \dot{x}_2 = -g \)
- \( \dot{x}_3 = 0 \)
- \( x_1 > 0 \lor (x_1 = 0 \land x_2 \geq 0) \)
- \( x_2 := -cx_2 \)
- \( x_1 \geq 0 \)

\( \epsilon = 0.1 \)
TEMPORAL REGULARIZATION: BOUNCING BALL

Executions get extended beyond the Zeno time $\tau_\infty$

In the limit, as $\varepsilon \to 0$, the execution of the regularized hybrid automaton converges to

- the Zeno execution for $t < \tau_\infty$
- $x_1(t) = x_2(t) = 0$ for $t \geq \tau_\infty$
HYBRID AUTOMATA: BLOCKING vs. NON-BLOCKING

A hybrid automaton $H = (Q, X, f, \text{Init}, \text{Dom}, E, G, R)$ is non-blocking if for all initial states $(q, x) \in \text{Init}$ there is an infinite execution starting at $(q, x)$.

The execution starting from $(q_1, -1)$ gets stuck at $x = 0$ after 1 time unit.

→ blocking hybrid automaton
HYBRID AUTOMATA: BLOCKING vs. NON-BLOCKING

A hybrid automaton $H = (Q, X, f, Init, Dom, E, G, R)$ is non-blocking if for all initial states $(q, x) \in Init$ there is an infinite execution starting at $(q, x)$

Note: this concerns with global existence of a solution, not uniqueness!

---

HYBRID AUTOMATA: BLOCKING vs. NON-BLOCKING

A hybrid automaton $H = (Q, X, f, Init, Dom, E, G, R)$ is non-blocking if for all initial states $(q, x) \in Init$ there is an infinite execution starting at $(q, x)$

Note: this concerns with global existence of a solution, not uniqueness!

$x \in \{-3\}$

$q_1$

$x \geq 0$

$x = 1$

$x \leq 0$

$q_2$

$x \leq -3$

$x : (-\infty, 0]$
NON-BLOCKING HYBRID AUTOMATA

Given a hybrid automaton $H = (Q, X, f, Init, Dom, E, G, R)$, transition states are those states from which continuous evolution is impossible:

$$\text{Trans} := \{(q', x') \in Q \times X : \forall \delta > 0, \exists t \in [0, \delta) \text{ such that } x(t) \notin \text{Dom}(q')\}$$

where $x(t)$ is the solution of $\frac{dx}{dt} = f(q', x)$ with $x(0) = x'$

A hybrid automaton is non-blocking if:

1. $f(q, \cdot)$ is Lipschitz for each $q \in Q$
2. for each $(q, x) \in \text{Trans}$, there exists $q'$ such that $(q, q') \in E$ and $x \in G(q, q')$

(a discrete transition should be possible from the Trans states)

Remark: not a necessary condition, since transition states are not necessarily reached from the initial states

NON-BLOCKING HYBRID AUTOMATA

$\{(q_2, x) : x \leq 0\} \subseteq \text{Trans}$

trivial for $x < 0$ (already outside $\text{Dom}(q_2)$),

for $x = 0$ it would exit the domain

No discrete transition possible from $q_2$

Still… non blocking!
HYBRID AUTOMATA: DETERMINISTIC vs. NONDETERMINISTIC

An execution is maximal if “it cannot be extended any further”

A hybrid automaton is deterministic if for each initial state \((q,x) \in \text{Init}\) there exists at most one maximal execution starting at \((q,x)\).

Remarks:

1. infinite executions are maximal.
2. to make the notion of maximal execution precise, we should introduce the notion of prefix and a partial order on the hybrid time sets and on the set of executions of a hybrid automaton (see the Notes by John Lygeros)

What causes non-determinism?

- choice between continuous evolution and discrete jump
- jump to multiple modes
- multiple reset positions
DETERMINISTIC HYBRID AUTOMATA

A hybrid automaton is deterministic if

1. If \( x \in G(q,q') \) for some \((q,q') \in E\), then \((q,x) \in Trans\)  
   *(jump only when continuous evolution not possible)*

2. If \((q,q'), (q,q'') \in E\), then \(G(q,q') \cap G(q,q'') = \emptyset\)  
   *(no multiple jumps possible)*

3. If \((q,q') \in E\) and \(x \in G(q,q')\), then \(R((q,q'),x)\) is a  
   singleton  
   *(only one reset position when jumping)*

EXISTENCE AND UNIQUENESS OF EXECUTIONS

A hybrid automaton has a unique infinite execution for each initial state if it is non-blocking and deterministic.

Sufficient condition given by the putting together the sufficient conditions for the hybrid automaton to be non-blocking and deterministic
HYBRID AUTOMATA: FORMAL DEFINITION

A hybrid automaton $H$ is a collection

$$H = (Q, X, f, \text{Init}, \text{Dom}, E, G, R)$$

- $Q = \{q_1, q_2, \ldots\}$ is a set of discrete states (modes)
- $X = \mathbb{R}^n$ is the continuous state space
- $f: Q \times X \to \mathbb{R}^n$ is a set of vector fields on $X$
- $\text{Init} \subseteq Q \times X$ is a set of initial states
- $\text{Dom}: Q \to 2^X$ assigns to each $q \in Q$ a domain $\text{Dom}(q)$ of $X$
- $E \subseteq Q \times Q$ is a set of transitions (edges)
- $G: E \to 2^X$ is a set of guards (guard condition)
- $R: E \times X \to 2^X$ is a set of reset maps

No input and output variables....

OPEN HYBRID AUTOMATA

An open hybrid automaton $H$ is a collection

$$H = (Q, X, V, W, f, h, \text{Init}, \text{Dom}, E, G, R)$$

- $Q = \{q_1, q_2, \ldots\}$ is a set of discrete states (modes)
- $X = \mathbb{R}^n$ is the continuous state space
- $V = \Sigma \times U$ is the input space (discrete & continuous)
- $W = \Psi \times Y$ is the output space (discrete & continuous)
- $f: Q \times X \times V \to \mathbb{R}^n$ is a set of vector fields on $X$
- $h: Q \times X \to W$ is an output map
- $\text{Init} \subseteq Q \times X$ is a set of initial states
- $\text{Dom}: Q \to 2^{X \times V}$ assigns to each $q \in Q$ a domain $\text{Dom}(q)$ of $X \times V$
- $E \subseteq Q \times Q$ is a set of transitions (edges)
- $G: E \to 2^{X \times V}$ is a set of guards (guard condition)
- $R: E \times X \times V \to 2^X$ is a set of reset maps