ESE601 – Hybrid Systems
Hybrid System Models

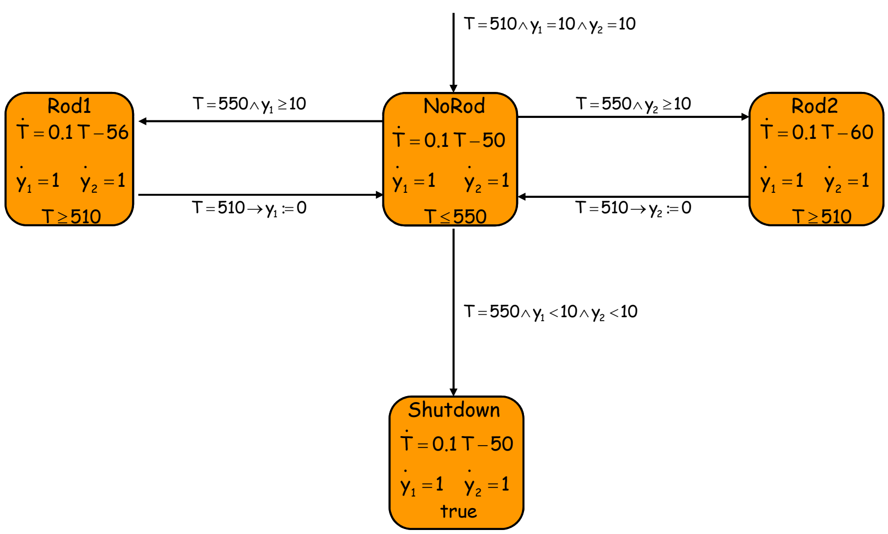
George J. Pappas

Department of Electrical and Systems Engineering

University of Pennsylvania

pappasg@seas.upenn.edu

Recall an example



Hybrid Automata

A hybrid system $H = (V, \Re^n, X_0, F, Inv, R)$ consists of

- V is a finite set of states
- $\Re^n$ is the continuous state space
- $X = V \times \Re^n$ is the state space of the hybrid system
- $X_0 \subseteq X$ is the set of initial states
- $F(l, x) \subseteq \Re^n$ maps a diff. inclusion to each discrete state
- $Inv(l) \subseteq \Re^n$ maps invariant sets to each discrete state
- $R \subseteq X \times X$ is a relation capturing discontinuous changes

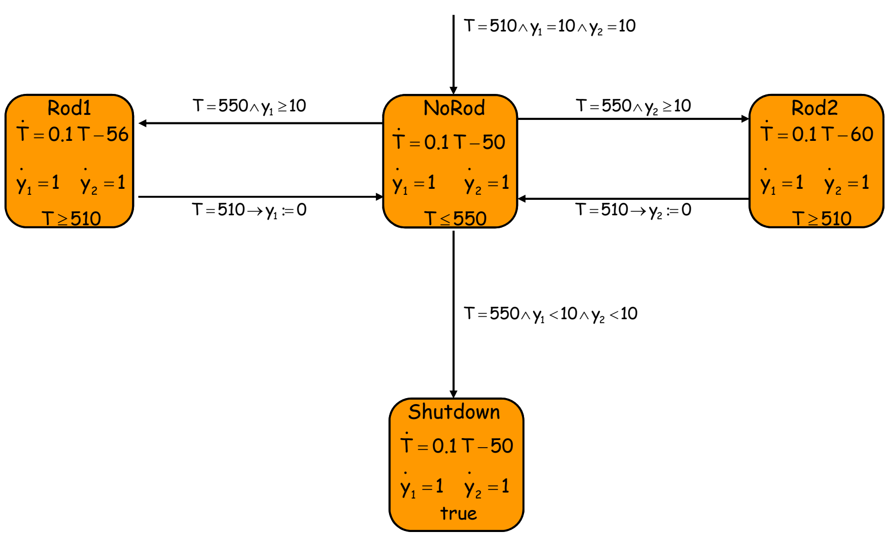
Define $E = \{(l, l') | \; \exists x \in Inv(l), x' \in Inv(l') \; ((l, x), (l', x')) \in R\}$

$$Init(l) = \{x \in Inv(l) \mid (l, x) \in X_0\}$$

$$Guard(e) = \{x \in Inv(l) | \; \exists x' \in Inv(l') \; ((l, x), (l', x')) \in R\}$$

$$Reset(e, x) = \{x' \in Inv(l') | \; ((l, x), (l', x')) \in R\}$$

An example



Transitions of Hybrid Systems

Hybrid systems can be embedded into transition systems

$H = (V, \Re^n, X_0, F, Inv, R) \longrightarrow T_H = (Q, Q_0, \Sigma, \rightarrow, O, <\cdot>)$

$Q = V \times \Re^n$

$Q_0 = X_0$

$\Sigma = E \cup \{\tau\}$

$\rightarrow \subseteq Q \times \Sigma \times Q$

Observation set and map depend on desired properties

Discrete transitions

$(l_1, x_1) \xrightarrow{e} (l_2, x_2)$ iff $x_1 \in Guard(e), x_2 \in Reset(e, x_1)$

Continuous (time-abstract) transitions

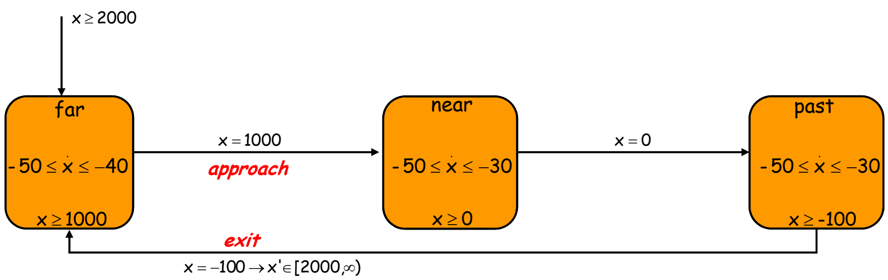
$(l_1, x_1) \xrightarrow{\tau} (l_2, x_2)$ iff $l_1 = l_2$ and $\exists \delta \geq 0 \quad x(\cdot) : [0, \delta] \rightarrow \Re^n$

$x(0) = x_1, x(\delta) = x_2,$ and $\forall t \in [0, \delta]$

$\dot{x} \in F(l_1, x(t))$ and $x(t) \in Inv(l_1)$

Rectangular hybrid automata

Rectangular sets : $\bigwedge_i x_i \sim c_i \quad \sim \in \{<, \leq, =, \geq, >\}, c_i \in Q$

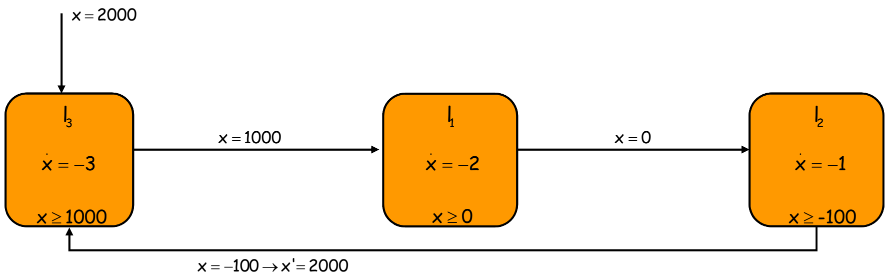


Rectangular hybrid automata are hybrid systems where

$$Init(l), Inv(l), F(l, x), Guard(e), Reset(e, x)_i$$

are rectangular sets

Multi-rate automata

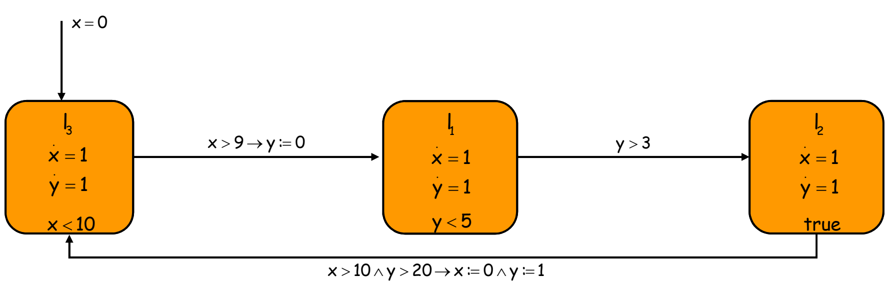


Multi-rate automata are rectangular hybrid automata where

$$Init(l), F(l,x), Reset(e,x)_i$$

are singleton sets

Timed automata



Timed automata are multi-rate automata where

$$F(l, x_i) = 1$$

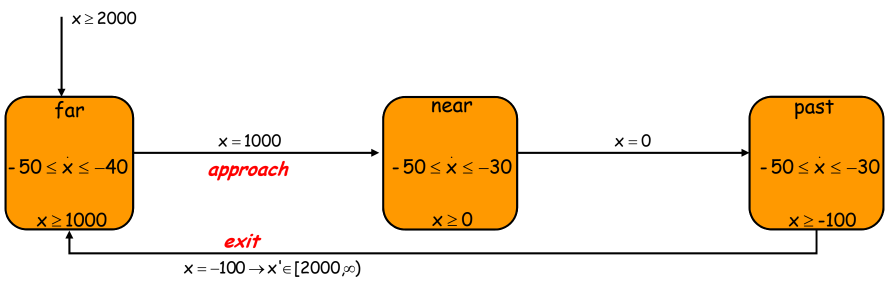
for all locations l and all variables.

Initialized automata

Rectangular hybrid automata are initialized if the following holds:

After a discrete transition, if the differential inclusion (equation) for a variable changes, then the variable must be reset to a fixed interv

Timed automata are always initialized.

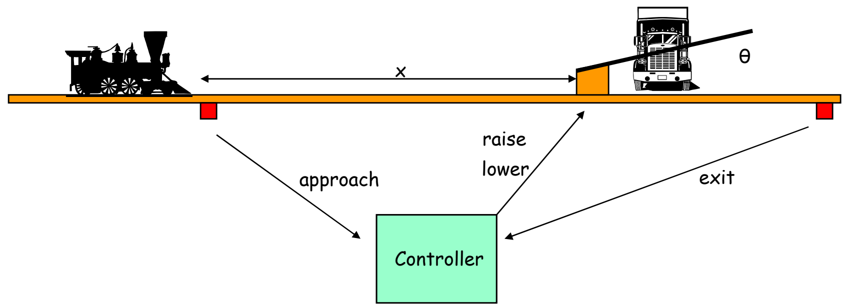


Hybrid automata are compositional

Partial synchronization

(Concurrency)

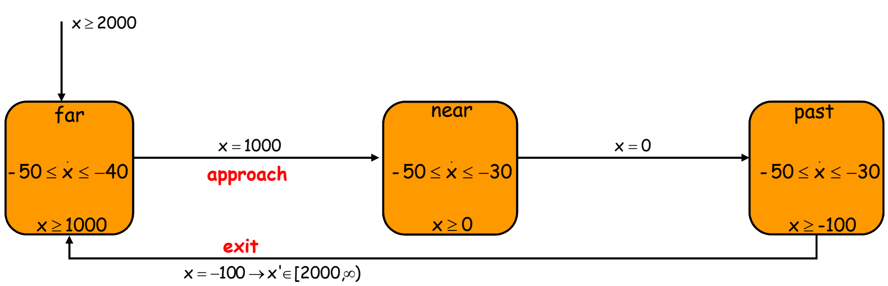
The train gate



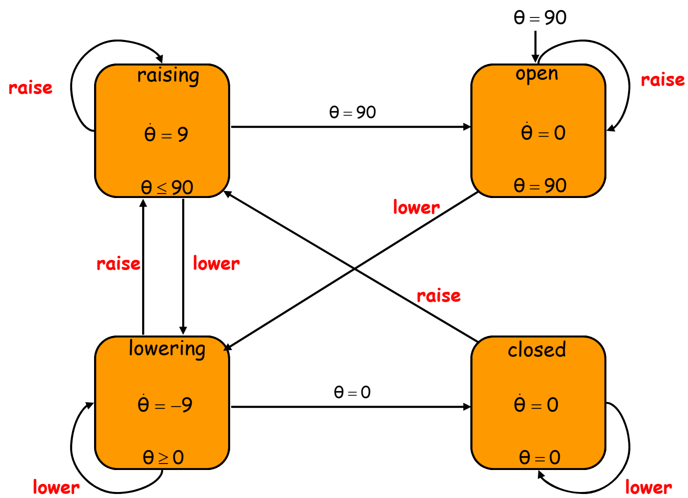
System = Train || Gate || Controller

Safety specification : If train is within 10 meters of the crossing, then gate should completely closed.
Liveness specification : Keep gate open as much as possible.

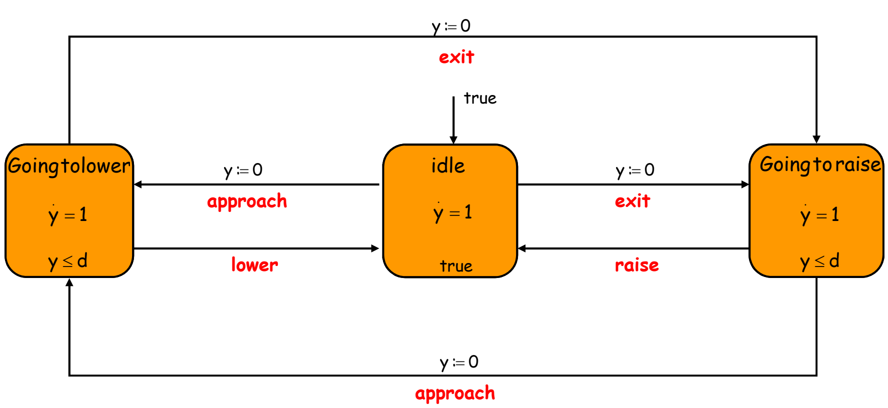
Train model



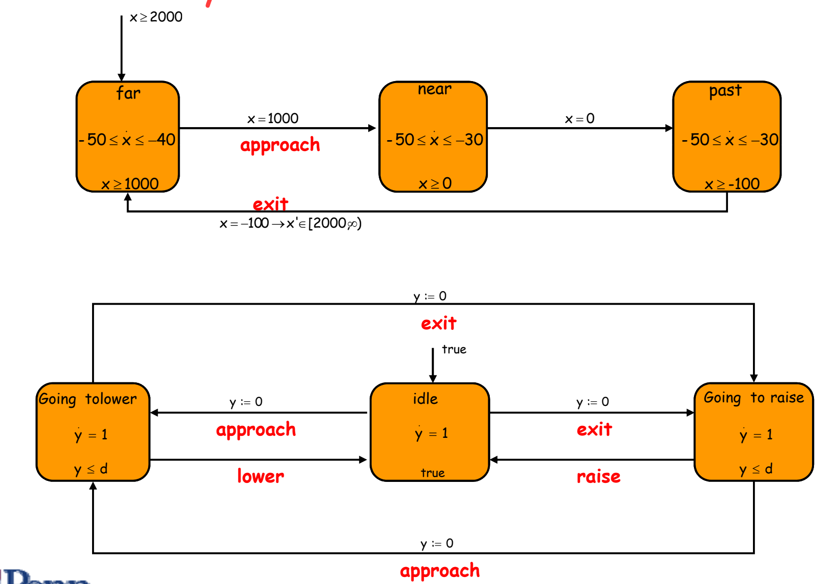
Gate model



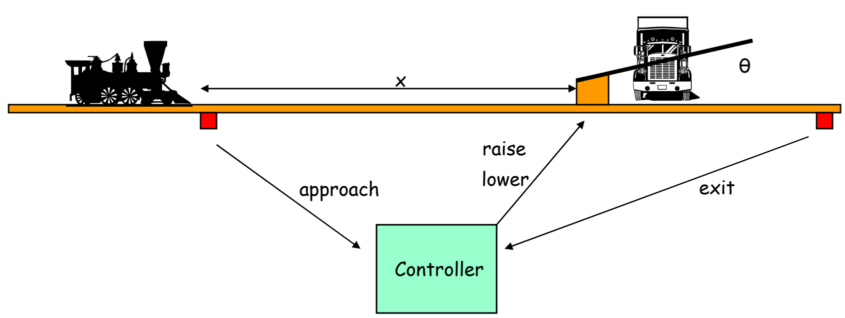
Controller model



Synchronized transitions



Verifying the controller



$System = Train \,||\, Gate \,||\, Controller$

Safety specification : Can we avoid the set $\theta > 0 \wedge (-10 \le x \le 10)$?

Parametric HyTech verification : YES if $d \le \frac{49}{5}$

Properties of trajectories

Blocking (or existence)

Determinism (or uniqueness)

Zeno (or finite escape time)

Compositional semantics (behaviors)

Compositional properties

Penn

An alternative notion of hybrid trajectory

Penn

Hybrid systems

Dynamical systems with discrete and continuous state and/or input variables

$$q \in Q = \{q_1, q_2, q_3\}$$

$$x \in \mathsf{R}^n$$

q changes discretely

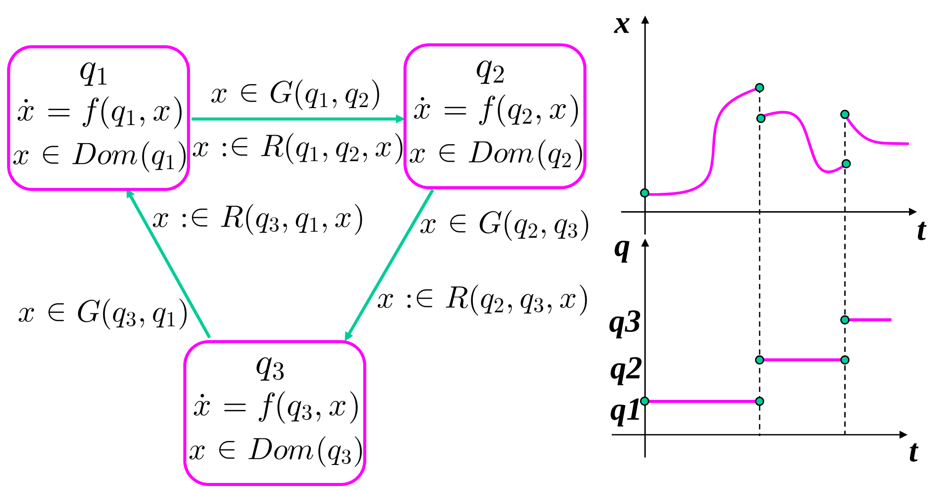
$$q(t^-) \mapsto q(t^+)$$

x changes either discretely, or continuously

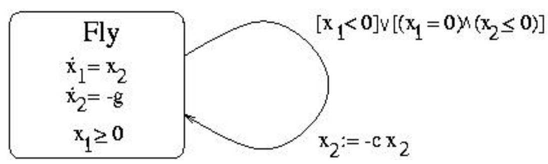
$$x(t^-) \mapsto x(t^+)$$

$$\dot{x}(t) = f(x(t), q(t))$$

System evolution

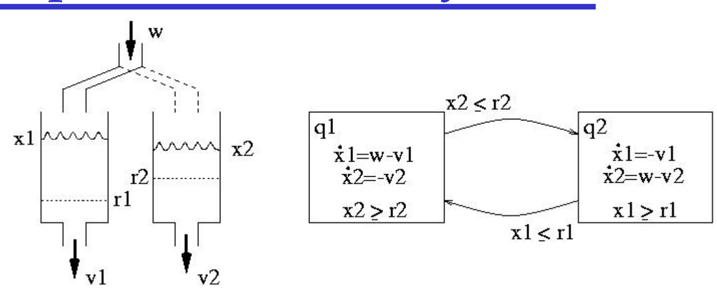


Example: Bouncing Ball



- Model of ball bouncing on level surface
- x_1 ball height, x_2 vertical ball velocity
- Fraction of energy lost at each impact

Example: Water Tank System



- Model of two leaky buckets
- Water supply dedicated either to one or the other bucket
- Water leaks at constant rate
- Supply at constant rate
- Controller switches supply to bucket that empties

Time axis

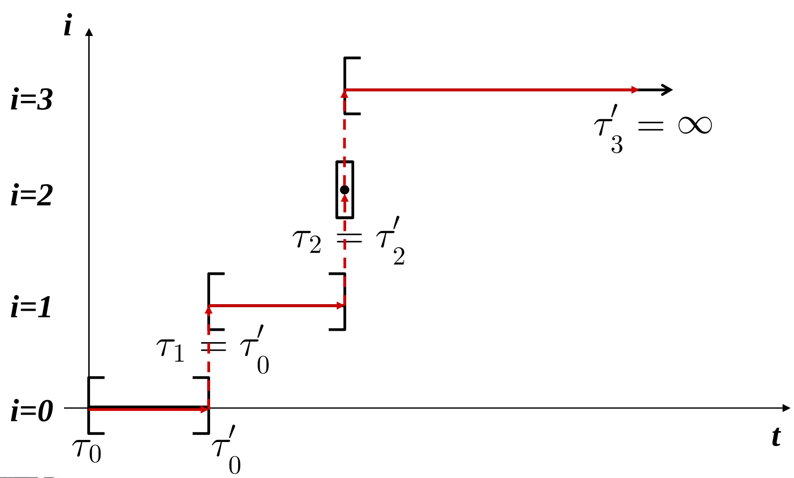
- Evolution both in continuous time and even driven
- Need time set richer that either R or N

$$\tau = \{I_i\}_0^N$$

- Hybrid time set:
  - Finite or infinite sequence of intervals
  - $I_i = [\tau_i, \tau_i']$ if $i < N$
  - $I_N = [\tau_N, \tau_N']$ or $I_N = [\tau_N, \tau_N')$ if $N < \infty$
  - $\tau_i \le \tau_i' = \tau_{i+1}$

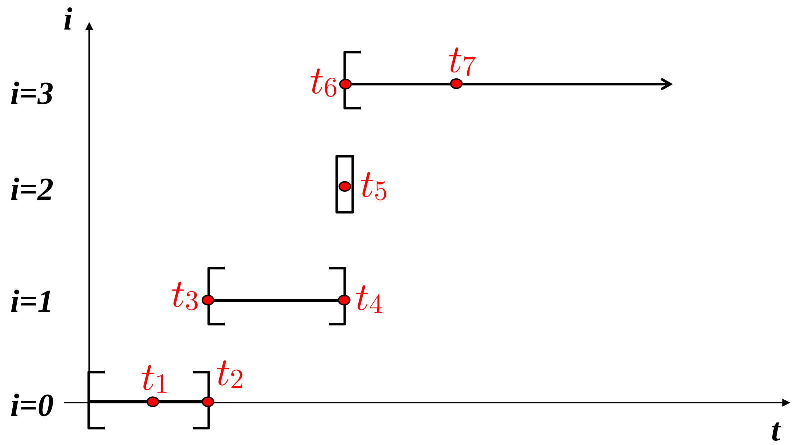
Example

$$\tau = \{I_i\}_0^3 = \{[\tau_0, \tau_0'], [\tau_1, \tau_1'], [\tau_2, \tau_2'], [\tau_3, \infty)\}$$



Properties

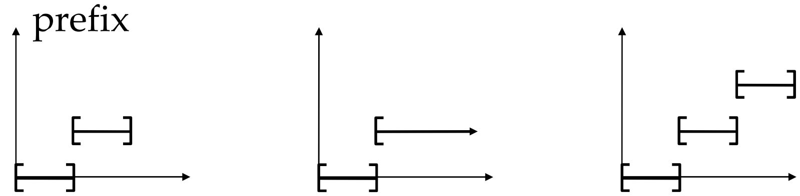
Each hybrid time set if totally ordered



$$t_1 \prec t_2 \prec t_3 \prec t_4 \prec t_5 \prec t_6 \prec t_7$$

Properties

- Set of hybrid time sets partially ordered
- Natural definition of prefix, extension, maximal prefix



τ $\hat{\tau}$ $\tilde{\tau}$

$$\tau \sqsubseteq \hat{\tau}$$
$$\tau \sqsubseteq \tilde{\tau}$$

$$\tilde{\tau} \not\sqsubseteq \hat{\tau}$$
$$\hat{\tau} \not\sqsubseteq \tilde{\tau}$$

(Autonomous) Hybrid Automata

Hybrid automaton:

$$H = (Q, X, Init, f, Dom, E, G, R)$$

- Discrete state variables $Q = \{q_1, q_2, q_3, \ldots\}$
- Continuous state variables $X = \mathsf{R}^n$
- Initial conditions $Init \subseteq Q \times X$
- Continuous dynamics $f : Q \times X \to \mathsf{R}^n$
- Domain of continuous evolution $Dom : Q \to 2^X$
- Discrete transitions $E \subseteq Q \times Q$
- Guards $G : E \to 2^X$
- Transition relation $R : E \times X \to 2^X$

ETH Zürich 27

What can it all mean?

- 2^X power set (set of all subsets) of X
- State of the system $(q, x) \in Q \times X$
- Start with $(q, x) \in Init$
- Continuous motion $\dot{x} = f(q, x)$ …
- … provided that $x = Dom(q)$
- Discrete transition $q \mapsto q'$ only if
  - $(q, q') \in E$
  - $x \in G(q, q')$
- After discrete transition $x' \in R(q, q', x)$

ETH Zürich 28

Executions

- Solutions called executions or runs
- Defined "declaratively" (cf. "imperatively")
- Solutions defined through "acceptance conditions"
  - Select any $(q, x) \in Init$
  - Follow ODE as long as $x \in Dom(q)$
  - Discrete transition provided $x \in G(q, q')$
  - After transition select any $x' \in R(q, q', x)$
- Many solutions for some initial conditions
- No solutions for others

ETH Zürich

Executions

- Executions NOT functions of real time
- They are of the form (τ, q, x) with
  - $\tau = \{I_i\}_0^N$
  - $x = \{x_i(\cdot)\}_0^N, \; x_i(\cdot) : I_i \to X$
  - $q = \{q_i(\cdot)\}_0^N, \; q_i(\cdot) : I_i \to Q$
- Initial condition $(q_0(\tau_0), x_0(\tau_0)) \in Init$

ETH Zürich

Executions

- Continuous evolution
  - The functions $q_i(\cdot)$ are constant
  - The functions $x_i(\cdot)$ are solutions of

$$\dot{x}_i(t) = f(q_i(t), x_i(t))$$

- Discrete transitions
  - $(q_i(\tau'_i), q_{i+1}(\tau_{i+1})) \in E$
  - $x_i(\tau'_i) \in G(q_i(\tau'_i), q_{i+1}(\tau_{i+1}))$
  - $x_{i+1}(\tau_{i+1}) \in R(q_i(\tau'_i), q_{i+1}(\tau_{i+1}), x_i(\tau'_i))$

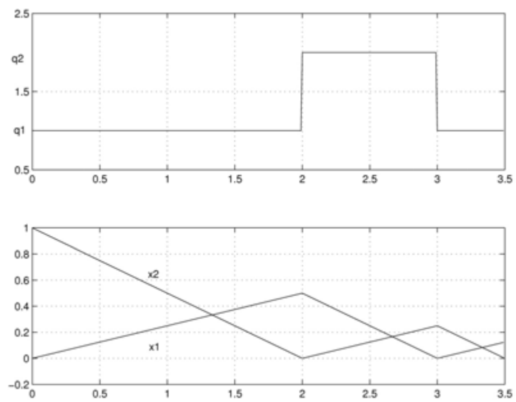
Classification

An execution (τ, q, x) with $\tau = \{I_i\}_0^N$ is called

- Finite if $N < \infty$ and $I_N = [\tau_N, \tau'_N]$
- Infinite if $N = \infty$ or $I_N = [\tau_N, \infty)$
- Zeno if $N = \infty$ and $\sum_{i=0}^{\infty}(\tau'_i - \tau_i) < \infty$
- Maximal if it is not a strict prefix of any other execution

Example: Water tanks

$$\tau = [0, 2], [2, 3], [3, 3.5]$$



Basic properties

- Basic system properties
- Existence uniqueness of solutions, etc
- Define states the system can reach

$$Reach = \{(q, x) \mid \exists(\tau, q, x),\ (q_N(\tau_N), x_N(\tau_N)) = (q, x)\}$$

- Define states where continuous evolution is impossible

$$Out = \{(q, x) \mid \forall \epsilon > 0\ \exists t \in [0, \epsilon),\ x(t) \notin Dom(q)\}$$

Existence and uniqueness of solutions

***Proposition*: Infinite executions exist for all initial conditions if for all** $(q, x) \in Reach \cap Out$ **there exists** $(q, q') \in E$ **such that** $x \in G(q, q')$

***Proposition*: Unique infinite executions exist for all initial conditions iff for all** $(q, x) \in Reach$

1. $x \in G(q, q') \Rightarrow (q, x) \in Out$

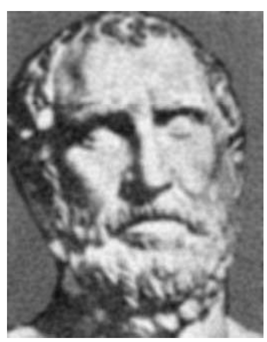
2. $x \notin G(q, q') \cap G(q, q'')$

3. $x \in G(q, q') \Rightarrow |R(q, q', x)| = 1$

***Zeno executions*: infinite number of discrete transitions in finite time**

ETH Zürich

Zeno of Elea



- ***Result of modeling over-abstraction***
- ***Surprisingly common (water tank, bouncing ball)***
- ***Can be tricky to cope with with in design problems***
- ***No satisfactory method exists for dealing with it***

ETH Zürich