EECE 571M/491M, Spring 2008
Lecture 4

Modeling Hybrid Systems

Meeko Oishi, Ph.D.

Electrical and Computer Engineering

University of British Columbia, BC

http://courses.ece.ubc.ca/491

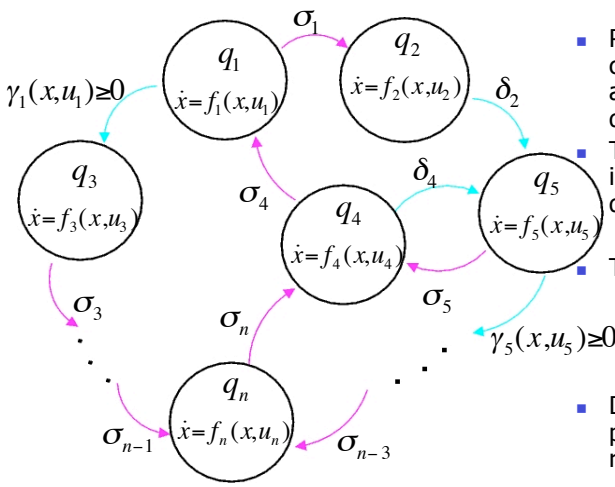
moishi@ece.ubc.ca

Tomlin LN 1,2,3

Review: Contin. Sys. & DES

Element	Continuous	Discrete
Dynamics	$\dot{x} = f(x,u)$	$R(q_k,\sigma) = q_{k+1}$
State	$x \in R^n$	$q \in Q$
Control	$u \in R^m$	$\sigma \in \Sigma$
Output	$y = h(x,u)$	$Q_M \subseteq Q$
Initial Condition	$x(0) = x_0 \in R^n$	$q[0] \in Q_0 \subseteq Q$
Solution	x(t) for u(t), x(0) known	q[k] for σ[k], q(0) known

Review: Hybrid Systems



- Potentially different continuous dynamics associated with each discrete mode
- Transitions can be input-driven or state-driven
- Transitions are
  - enabled through guard conditions
  - forced by definition of the domain
- Discontinuities are possible through the reset map

Review: Hybrid systems

The hybrid system H is a collection with the following entities:

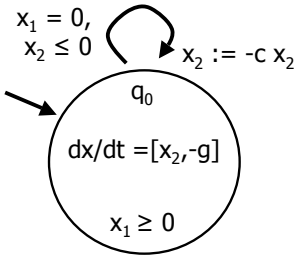
- Discrete state $q \in Q$
- Continuous state $x \in R^n$
- Discrete inputs $\sigma \in \Sigma$
- Continuous inputs $u \in U \subseteq R^m$
- Continuous dynamics $\dot{x} = f(q,x,u)$
- Discrete dynamics $R : Q \times R^n \times \Sigma \times R^m \rightarrow 2^{Q \times X}$
- Initial state $\text{Init} \subseteq Q \times R^n$
- Domain (combinations of states and inputs for which continuous evolution is allowed) $\text{Dom} \subseteq Q \times R^n \times \Sigma \times U$

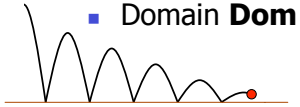
- Interpretation: R **enables** transitions, Dom **forces** them

Review: Bouncing ball

The hybrid system H_{ball} is a collection with the following entities:

- Discrete states **Q**
- Continuous state **x**
- Continuous dynamics **dx/dt = f(q,x)**
- Discrete dynamics **R(q,x)**
- Initial state **Init**
- Domain **Dom**

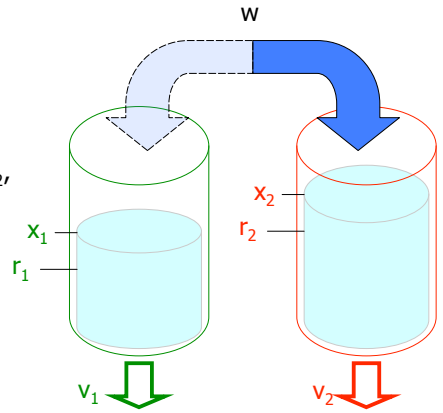




Guard condition **Guard(q,q')**
Reset map **Reset(q,x)**

Review: Water tanks

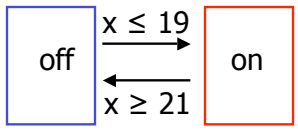
- One hose fills two tanks at constant rate *w*
- Hose switches instantaneously
- Tanks leak at rates v_1, v_2, respectively
- Goal: Keep water levels x_1, x_2 above r_1, r_2, respectively
- Question: How can this be modeled as a hybrid system?



dx/dt = w-v

Review: Thermostat

- Heater in a room is either on or off
- Temperature rises or falls depending on the heater's status
- Inaccuracies in temperature measurement
- Goal: Keep temperature near 20 C while avoiding **chattering**
- Question: How can this be modeled as a hybrid system?



dx/dt = -ax dx/dt= a(30-x)
x ≥ 18 x ≤ 22

Today's lecture

- Review
  - Mathematical formulation of hybrid systems
  - Three examples
    - Bouncing ball
    - Water tank
    - Thermostat
- Hybrid systems
  - 'Time' in a hybrid system
  - Trajectories (executions)
    - Important properties
    - Existence and uniqueness
    - Zenoness

Hybrid time sets

- Sequence of intervals $\mathcal{I}_i = [\tau_i, \tau_i']$

$$\begin{aligned}\tau &= \{I_0, I_1, \cdots, I_i, \cdots I_N\} \\ &= \{I_i\}_0^N\end{aligned}$$

- Each interval corresponds to time spent in a single mode
- New interval when system switches into a new mode
- Hybrid time set is
  - **Finite** if a finite number of modes are visited
  - **Infinite** if an infinite number of modes are visited

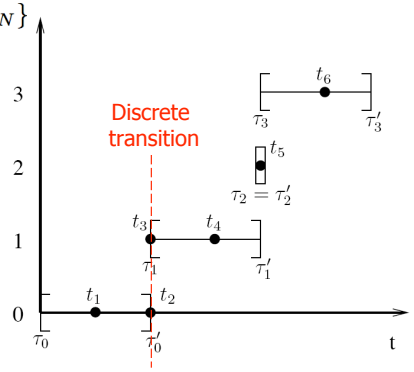
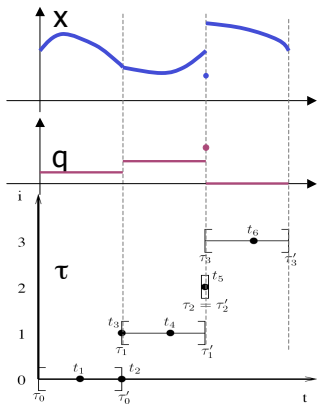


Figure 1: A hybrid time set $\tau = \{[\tau_i, \tau_i']\}_{i=0}^3$.

Hybrid executions

- Hybrid executions start at an acceptable state (in the defined initial set)
- Discrete transitions can occur only as specified in the discrete transition function
- Continuous evolution must have a unique solution (e.g., Lipschitz continuity within each mode)
- Continuous state must remain within the domain of the current mode
- The mode stays constant over the entire interval

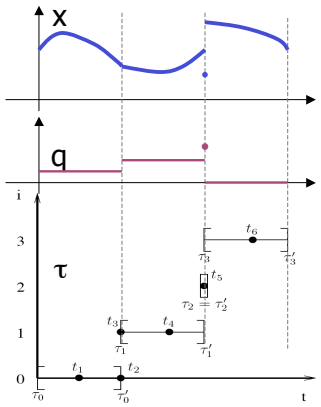


Hybrid executions

- Tuple (τ,q,x)
  - $\tau = \{\mathcal{I}_i\}_0^N$
  - $q = \{q_i(\cdot)\}_0^N, \ q_i : \mathcal{I}_i \to Q$
  - $x = \{x_i(\cdot)\}_0^N, \ x_i : \mathcal{I}_i \to R^n$
- With I.C. $(q_0, x_0) \in \text{Init}$
- Discrete evolution

$$(q_{i+1}(\tau_{i+1}), x_{i+1}(\tau_{i+1})) \in R(q_i(\tau_i'), x_i(\tau_i'))$$

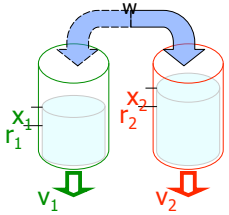
- Continuous evolution
  - $q_i(.)$ is constant for $t \in \mathcal{I}_i$
  - $x_i(.)$ is the solution to $\dot{x}_i = f(q_i(t), x_i(t))$
  - $x_i(t) \in \text{Dom}(q_i(t)) \in \mathcal{I}_i$ for $t \in \mathcal{I}_i$



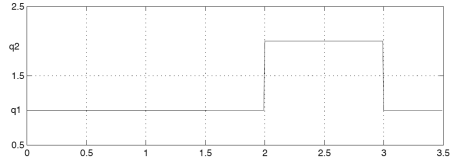
Hybrid executions

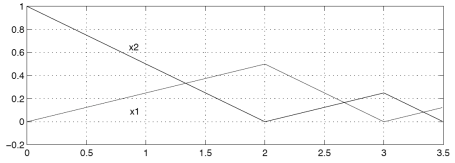
Water tank problem with

- $r_1 = r_2 = 0$
- $v_1 = v_2 = 0.5$
- $w = 0.75$
- Initial state (q_1, [0, 1])



dx/dt = w-v



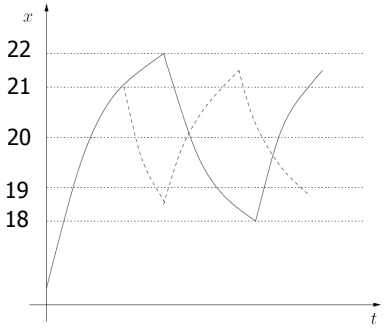


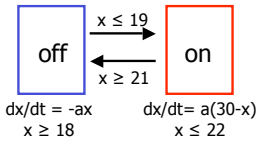
Time set τ = { [0,2], [2,3], [3,3.5] }

Hybrid executions

Thermostat problem with
- Initial state (on, 16)

Nondeterministic since more than one execution is possible from the same initial condition





Hybrid systems

- **Existence and Uniqueness Theorem:**
  - A hybrid automaton H accepts a unique infinite execution for each initial state if it is deterministic and non-blocking
- Non-blocking --> executions exist for all initial states
- Deterministic --> Infinite executions (if they exist) are unique
- Notice that this is significantly different from the requirements for existence and uniqueness of continuous dynamical systems.
- Mathematical tests exist to determine whether these conditions are met (**Tomlin LN 4**)
- You may assume existence and uniqueness are satisfied for all examples we will cover in this course, unless stated otherwise.

Finding unique executions

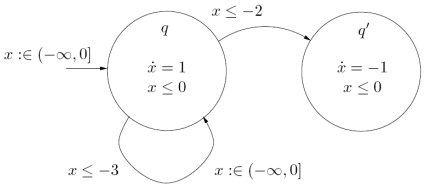
- Executions are **nondeterministic** when
  - Any discrete transition can evolve to more than one mode
  - A discrete transition is enabled while continuous evolution is possible
- Executions are **deterministic** when
  - Each discrete transition has a unique destination
  - Whenever a discrete transition is possible, continuous evolution is impossible
- The thermostat example is nondeterministic
- The water tank and bouncing ball examples are deterministic

Finding unique executions

- **Non-blocking**
  - Infinite executions exist from all states in the initial set
    (Executions must always have somewhere next to evolve to, either through continuous or discrete dynamics -- they cannot be "blocked")
  - Slightly different definitions for deterministic vs. nondeterministic automata
- **Blocking**
  - Executions can get "stuck" -- e.g., if it is not possible to remain in a continuous mode and it is also not possible to transition to another mode
- The thermostat, water tank, and bouncing ball examples are all non-blocking

Hybrid systems

- Pathological example
  - Blocking
  - Non-deterministic



Zenoness

- Infinite transitions in a finite amount of time
- Achilles and the tortoise
- Walking towards a wall
- Not possible in discrete systems or continuous systems -- but possible in hybrid systems
- Problematic for simulation or computational algorithms

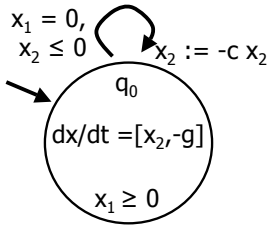
Zenoness

- Bouncing ball example
  - Deterministic
  - Non-blocking
  - Zeno
- The Nth bounce occurs at time:

$$\tau_N = \tau_1' = \tau_0 + \tau_1 + \frac{2x_2(\tau_1)}{g}\sum_{k=1}^{N} c^{k-1}$$

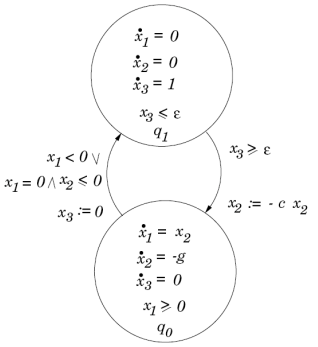
- And since for 0 ≤ c < 1, $\sum_{k=1}^{} c^{k-1} \to \frac{1}{1-c}$ as $N \to \infty$
- As N -> ∞, executions occur in the time

$$\|\tau\| = \tau_0 + \frac{x_2(\tau_0)}{g} + \frac{(1+c)\sqrt{x_2^2(\tau_0) + 2gx_1(\tau_0)}}{g(1-c)} < \infty$$



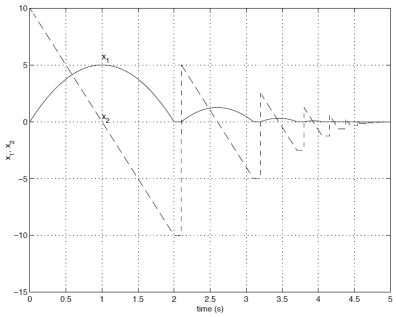
Zenoness

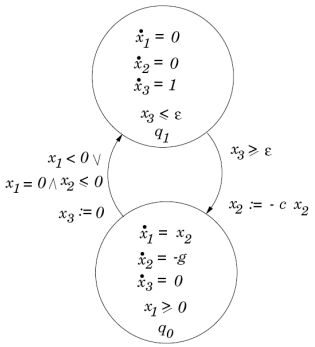
- Bouncing ball example
  - Deterministic
  - Non-blocking
  - Zeno
- One way to "fix" zenoness is to add a time delay
  - Additional "holding" mode
  - Additional state (time) with additional domain constraints to enforce a minimum **dwell time**
- Often arises due to modeling simplifications



Zenoness

- Bouncing ball example
  - Deterministic
  - Non-blocking
  - Zeno





Summary

- Executions of hybrid automata
  - Time sets
  - Trajectories
- Existence and uniqueness
  - Requires non-blocking and deterministic H
  - Can be checked by evaluating system parameters (guards, domain, reset map)
  - May require constructing *Reach* and *Trans* set, sets of reachable states and transition states, respectively
- Zenoness
  - Achilles and the tortoise
  - Chattering

Next lectures

- Continuous system stability (Linear systems)
  - Brief review of standard techniques
  - Similarity transformations
  - Diagonalization
  - Graphical analysis -- phase plane
  - Lyapunov equation
- Continuous system stability (Nonlinear systems)
  - Linearization
  - Graphical analysis -- phase plane
  - When linearization doesn't help