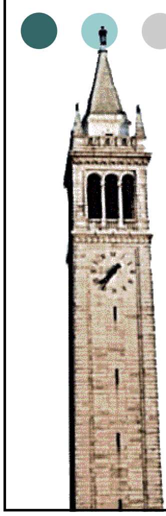


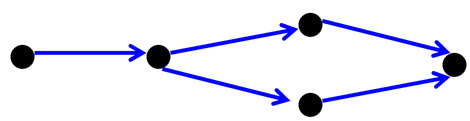
Introduction to Embedded Systems

Edward A. Lee & Sanjit A. Seshia
UC Berkeley
EECS 124
Spring 2008

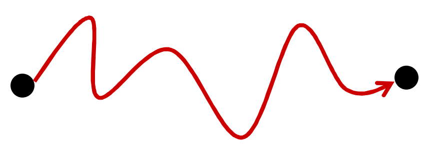
Lecture 6: Hybrid Systems, Part I

Material drawn from notes by T. Henzinger, J. Lygeros, S. Sastry, C. Tomlin

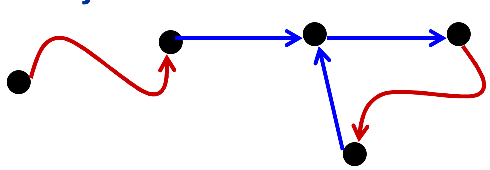
Discrete System (FSM)

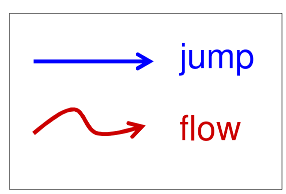


Continuous System



Hybrid System





Topics in Today's Lecture

- Examples of Hybrid Systems
- The *Hybrid Automaton* Model
  - properties of this kind of model
- Next time: focus on special case called Timed Automata (TA), and bisimulation of TA with FSM

A Thermostat

State has both discrete and continuous components:

$x \in \mathbb{R}$	temperature
$h \in \{\text{on}, \text{off}\}$	heating mode

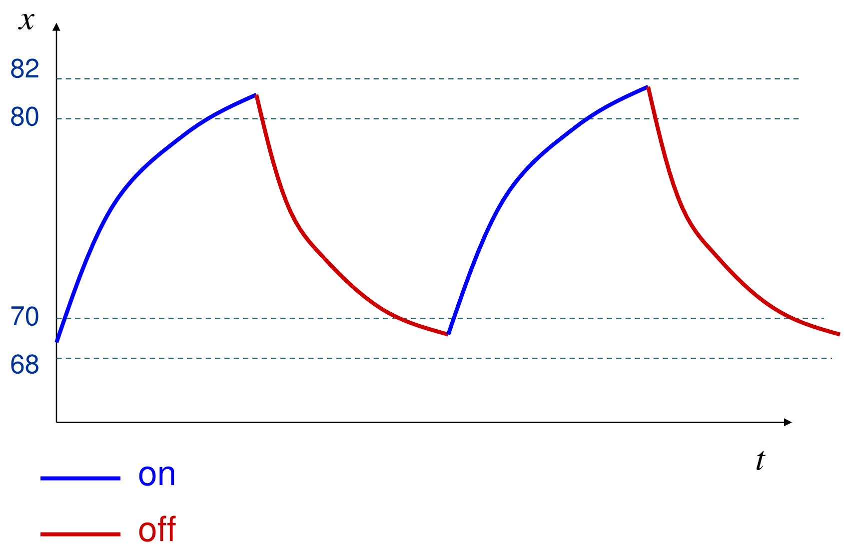
Flow in each mode is:

$h = \text{on} \wedge x < 82$	$\dot{x} = K(100 - x)$
$h = \text{off} \wedge x > 68$	$\dot{x} = -Kx$

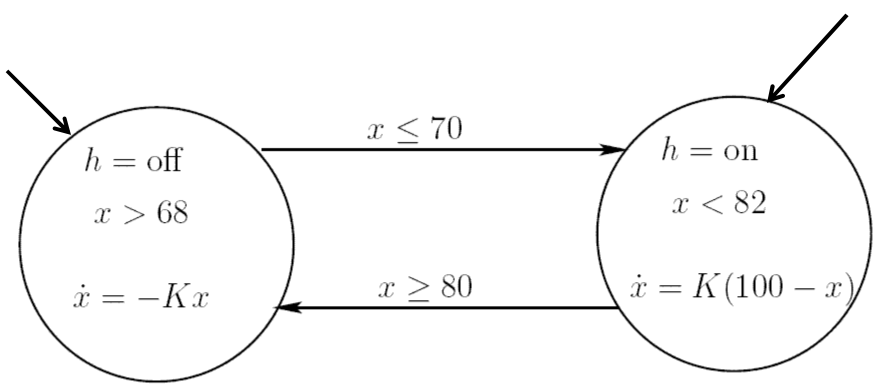
Jumps between modes: (happen instantaneously)

$h = \text{on} \wedge x \geq 80$	$\rightarrow$	$h := \text{off}$
$h = \text{off} \wedge x \leq 70$	$\rightarrow$	$h := \text{on}$

Dynamics of Thermostat



Hybrid Automaton for Thermostat

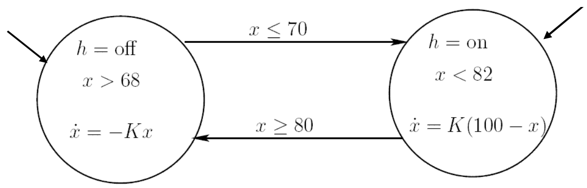


Is this automaton deterministic?

Formal Representation of Hybrid Automaton

A hybrid automaton is a tuple: $(Q, X, \Sigma, U, Init, F, J, Inv)$

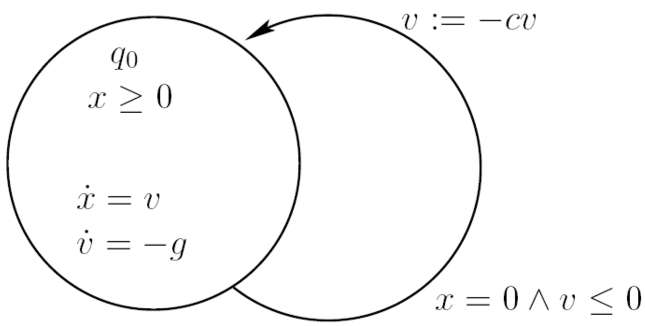
Q	finite set of modes
X	finite set of continuous state variables $\{x_1, x_2, \ldots, x_n\}$, $x_i \in \mathbb{R}$
Σ	set of discrete input symbols
U	set of continuous input signals, $\{u_1, u_2, \ldots, u_k\}$, $u_i \in \mathbb{R}$
$Init$	initial condition, $Init \subseteq Q \times \mathbb{R}^n$
F	flows, defining differential equations for each variable in each mode
J	jumps, $J : Q \times Guards \to Q \times Resets$ where an element of $Guards$ is a subset of $\Sigma \times \mathbb{R}^n \times \mathbb{R}^k$, and $Resets$ is a set of assignments of the form $x_i := expr(X, U)$
Inv	mode invariant, mapping a state to the subspace of $\mathbb{R}^n$ in which the X variables can take values



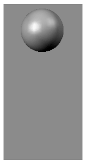
Where do Hybrid Systems arise?

- Digital controller of physical "plant"
  - thermostat
  - intelligent cruise control in cars
  - aircraft auto pilot
- Phased operation of natural phenomena
  - bouncing ball
  - biological cell growth
- Multi-agent systems
  - ground and air transportation systems
  - interacting robots (e.g., RoboSoccer)

Hybrid Automaton for Bouncing Ball



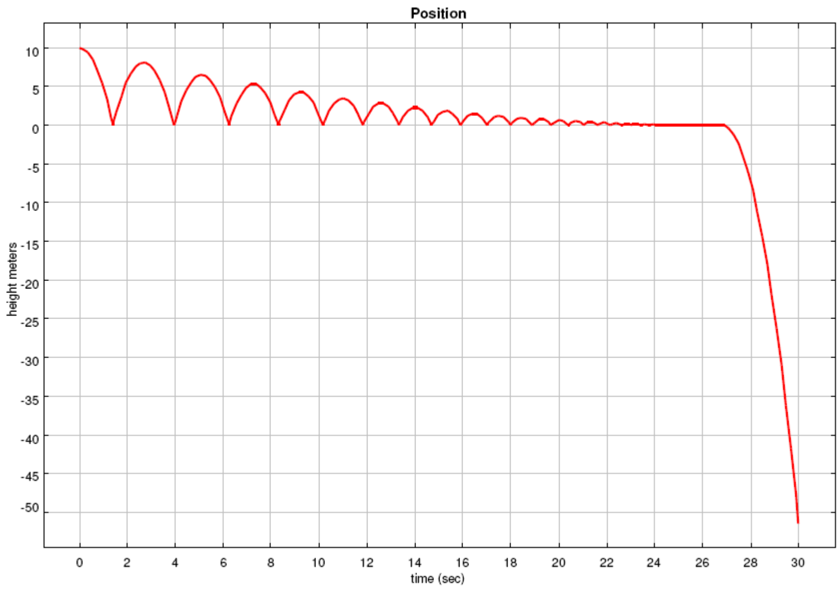
x – vertical distance from ground (position)
v – velocity
c – coefficient of restitution, $0 \leq c \leq 1$



Simulation of Bouncing Ball Automaton: Plot position x as a function of time t

What kind of plot would you expect?

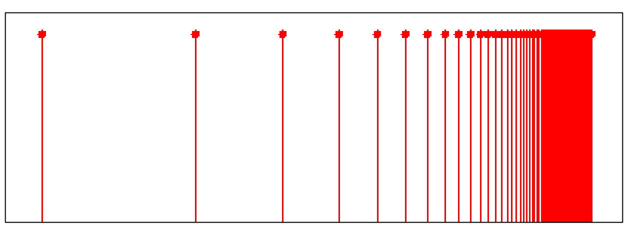
Simulation of Bouncing Ball Automaton in Ptolemy II / HyVisual



Zeno Behavior

Informally:

The system makes an infinite number of jumps in finite time



A Run/Execution of a Hybrid Automaton

time ↓

$$
\begin{array}{ll}
\tau_0 & (q_0, \mathbf{x}_0) \\
\tau_0' & \rightsquigarrow (q_0, \mathbf{x}_0') \\
= & \downarrow \\
\tau_1 & (q_1, \mathbf{x}_1) \\
\tau_1' & \rightsquigarrow (q_1, \mathbf{x}_1') \\
 & \downarrow \\
 & \vdots \\
 & \downarrow \\
\tau_N & (q_N, \mathbf{x}_N) \\
\tau_N' & \rightsquigarrow (q_N, \mathbf{x}_N') \\
 & \downarrow \\
 & \vdots
\end{array}
$$

$$\tau = \tau_0, \tau_1, \tau_2, \ldots, \tau_N[, \ldots]$$

Continuous extent of τ:

$$|\tau| = \sum_{i=0}^{\infty} \tau_{i+1} - \tau_i$$

Discrete extent of τ:

$$\langle \tau \rangle = \begin{cases} N & \text{if } \tau \text{ is a finite sequence of length } N \\ \infty & \text{if } \tau \text{ is an infinite sequence} \end{cases}$$

Zeno Behavior: Formal Definition

time ↓

$$
\begin{array}{ll}
\tau_0 & (q_0, \mathbf{x}_0) \\
\tau_0' & \rightsquigarrow (q_0, \mathbf{x}_0') \\
= & \downarrow \\
\tau_1 & (q_1, \mathbf{x}_1) \\
\tau_1' & \rightsquigarrow (q_1, \mathbf{x}_1') \\
 & \downarrow \\
 & \vdots \\
 & \downarrow \\
\tau_N & (q_N, \mathbf{x}_N) \\
\tau_N' & \rightsquigarrow (q_N, \mathbf{x}_N') \\
 & \downarrow \\
 & \vdots
\end{array}
$$

An execution of a hybrid automaton with time set τ is **zeno** iff $\langle \tau \rangle = \infty$ but $|\tau| < \infty$.

Analysis of Zeno Behavior of Bouncing Ball

If $c < 1$ all infinite executions are Zeno. The first bounce occurs at time:

$$\tau_1 = \tau_0 + \frac{v(\tau_0) + \sqrt{v^2(\tau_0) + 2gx(\tau_0)}}{g}$$

The second bounce occurs at time:

$$\tau_2 = \tau_0 + \tau_1 + \frac{2v(\tau_1)}{g}$$

where $v(\tau_1) = -cv(\tau_0') = \sqrt{v^2(\tau_0) + 2gx(\tau_0)}$.

More generally, the Nth bounce occurs at time:

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

For $c \in [0, 1)$, we have $\sum_{i=1}^{\infty} c^{i-1} = \frac{1}{1-c}$.

Thus $\lim_{N \to \infty} \tau_N < \infty$.

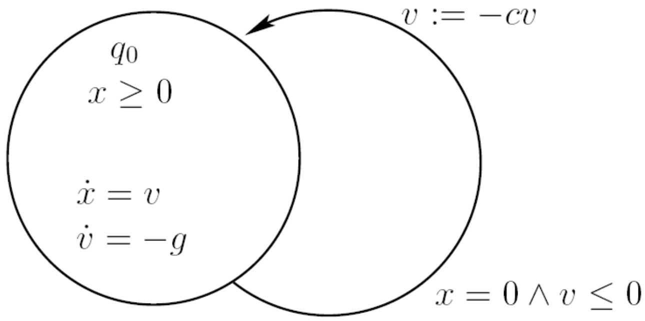
Why does Zeno Behavior Arise?

Our model is a mathematical artifact

Zeno behavior is mathematically possible, but it is infeasible in the real, physical world

Points to some unrealistic assumption in the model

Hybrid Automaton for Bouncing Ball: What's Unrealistic about this model?

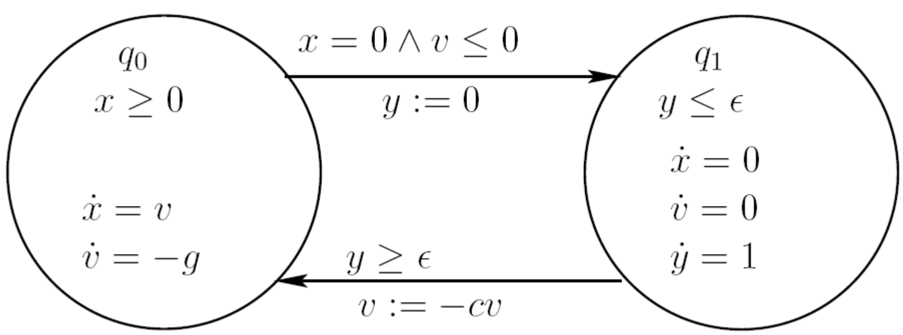


x – vertical distance

v – velocity

c – coefficient of restitution, $0 \leq c \leq 1$

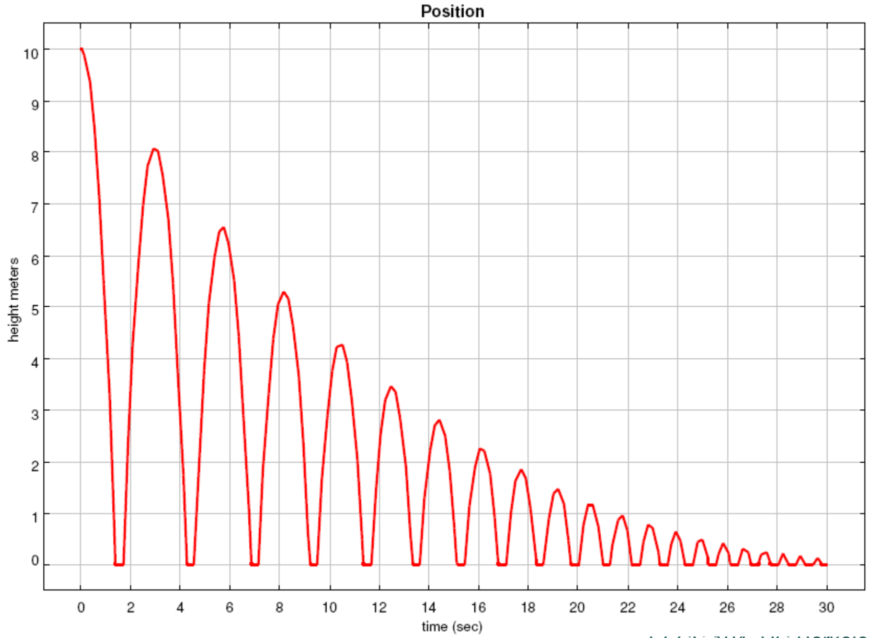
Eliminating Zeno Behavior: Regularization



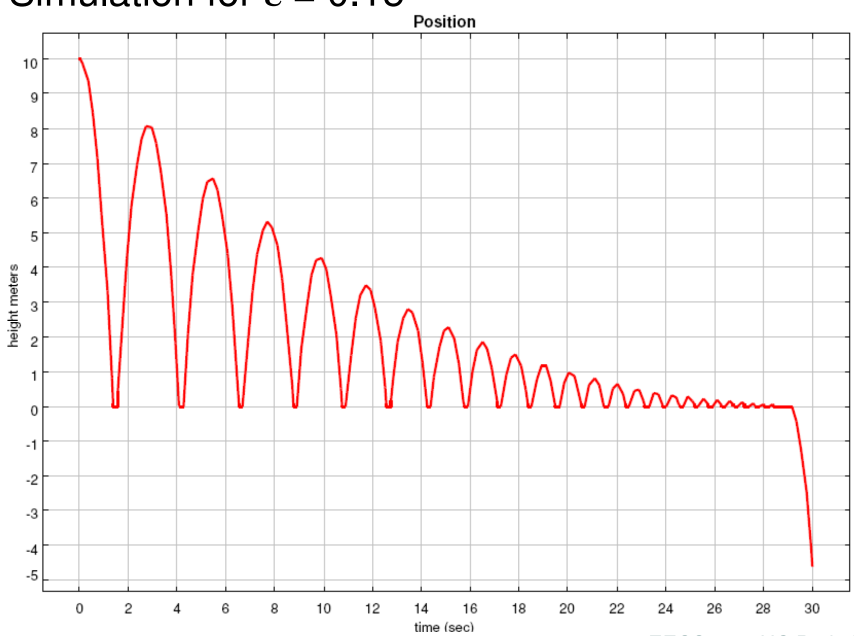
An instantaneous mode change (jump) is unrealistic

What happens as ε goes to 0?

Simulation for ε = 0.3



Simulation for ε = 0.15



Exercise: Construct a Hybrid Automaton Model of iRobot Hill Climber