Modeling Cyberphysical Systems

Sayan Mitra

Verifying cyberphysical systems

mitras@illinois.edu

Outline

- Modeling cyber-physical systems
  - Hybrid automata
  - Executions
  - Urgency
- Zeno
- Hybrid stability

Why combine discrete and continuous time models?

Engineering systems can have dynamics with widely different time constants

- Software for AV updates at 1 GHz --- human scale motion 10 Hz
- Electrical circuit breakers 50 Hz --- power system dynamics 0.1 – 1 Hz

Widely different time constants can make computation numerically unstable

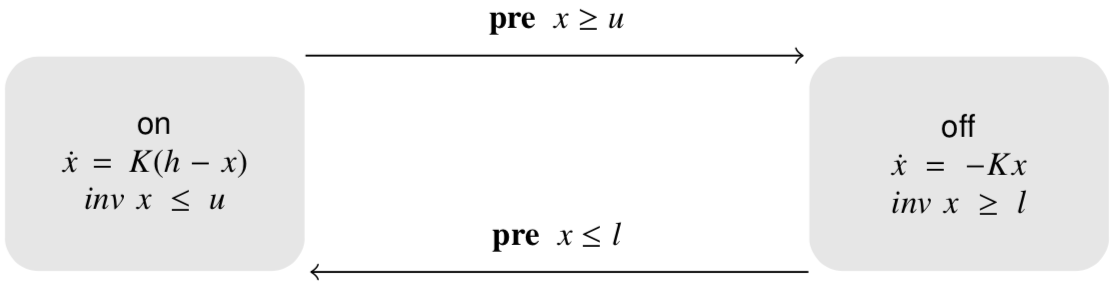
Some state changes are naturally described by automata:

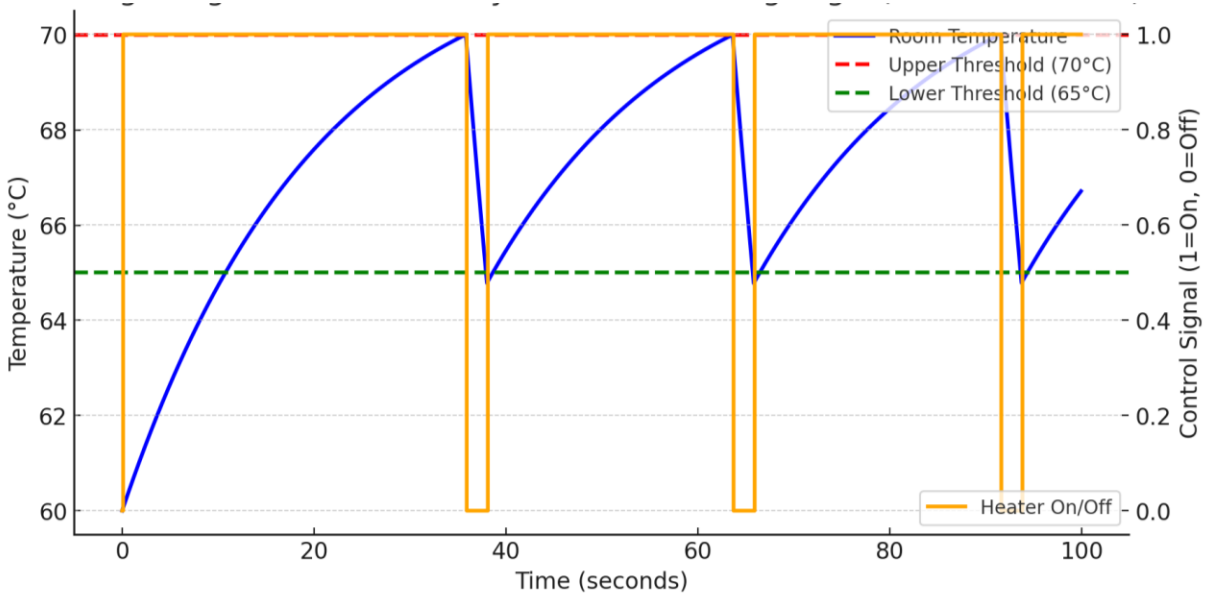
- programs, automata, SMT, switching rules, etc.

while others are naturally described by ODEs:

- motion of planets and satellites, physical laws, weather

Example. Thermostat automaton





automaton Thermostat(u, l, K, h : Real) where $u > l$
type *Status* enumeration [*on*, *off*]
actions
turnOn; turnOff;

variables
x: Real := l *loc*: *Status* := *on*
transitions
turnOn
pre $x \leq l \wedge loc = off$
eff *loc* := *on*

trajectories
modeOn
evolve $d(x) = K(h - x)$
invariant $loc = on \wedge x \leq u$

turnOff
pre $x \geq u \wedge loc = on$
eff *loc* := *off*

modeOff
evolve $d(x) = -Kx$
invariant $loc = off \wedge x \geq l$

Determinism vs nondeterminism
mode invariants

Recall language for automata

An **automaton** tuple $\mathcal{A} = \langle X, \Theta, A, \mathcal{D} \rangle$

- X is a set of variables; each $x \in X$ is associated with a type, $type(x)$
  - A **valuation** for X maps each variable in X to its type
  - Set of all valuations: $val(X)$ is the **state space**
- $\Theta \subseteq val(X)$ set of **initial** or **start states**
- A is a set of names of **actions** or **labels**
- $\mathcal{D} \subseteq val(X) \times A \times val(X)$ is the set of **transitions**
  - a transition is a triple (u, a, u')
  - defined by **pre and eff**
  - We write it as $u \rightarrow_a u'$

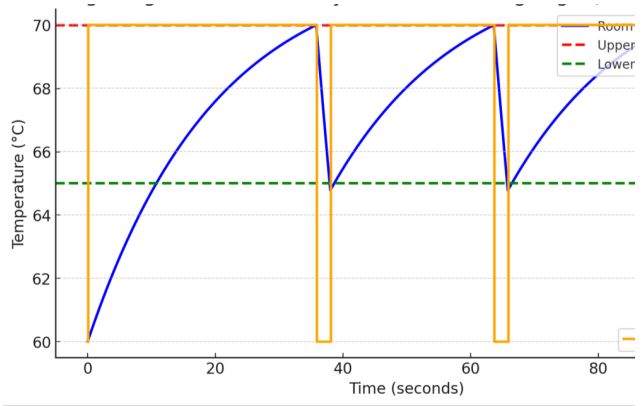
```
automaton DijkstraTR(N:Nat, K:Nat), where K > N
 type ID: enumeration [0,...,N-1]
 type Val: enumeration [0,...,K]
 actions
   update(i:ID)
 variables
   x:[ID -> Val]
 transitions
   update(i:ID)
     pre i = 0 /\ x[i] = x[N-1]
     eff x[i] := (x[i] + 1) %  K

    update(i:ID)
      pre i >0  /\ x[i] ~= x[i-1]
      eff x[i] := x[i-1]
```

Trajectories

For set of variables X and time interval J of the form $[0,T], [0,T) or\ [0,\infty)$, a ***trajectory*** for X is a function $\tau\colon\ J \to val(X)$

We will specify τ as *solutions of* differential equations



first state of $\tau.fstate := \tau(0)$

If τ is right closed then the **limit state** of $\tau.lstate = \tau(T)$

If τ is finite then **duration** of τ is $\tau.dur = T$

Domain of $\tau.dom = J$

A **point trajectory** is a trajectory with $\tau.dom = [0,0]$

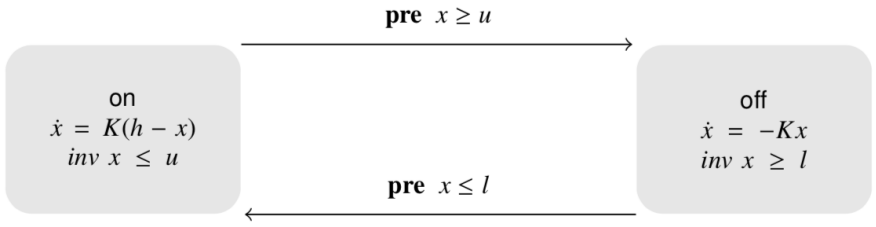
A **prefix** τ' of $\tau\colon [0,T] \to val(X)$, is a function $\tau'\colon [0,T'] \to val(X)$ such that $T' \leq T$ and $\tau'(t) = \tau(t)$ for all $t \in \tau'.dom$

Hybrid Automaton

$\mathcal{A} = (X, \Theta, A, \mathcal{D}, \mathcal{T})$

- X: set of **state variables**
  - $Q \subseteq val(X)$ set of **states**
- $\Theta \subseteq Q$ set of **start states**
- set of **actions**, A= E ∪ H
- $\mathcal{D} \subseteq Q \times A \times Q$
- $\mathcal{T}$: set of **trajectories** for X which is closed under **prefix, suffix, and concatenation**

Semantics: Executions and Traces



An ***execution*** of $\mathcal{A}$ is an (possibly infinite) alternating sequence $\alpha = \tau_0\, a_1\, \tau_1 a_2 \tau_2\ \ldots$ where

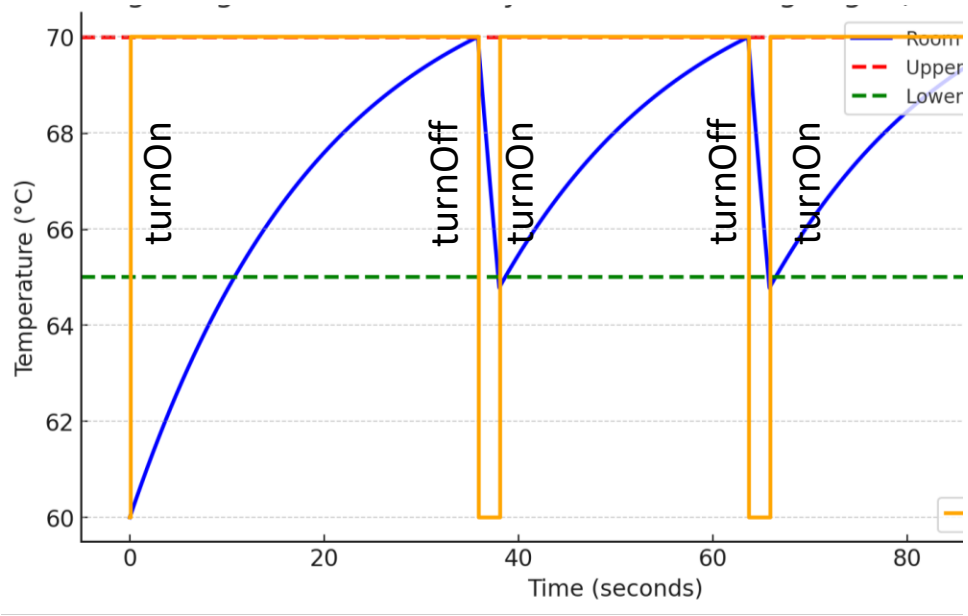
- $\forall$ i, $\tau_i.lstate \xrightarrow{a_{i+1}} \tau_{i+1}.fstate$
- If τ_0.fstate $\in \Theta$ then α is an **execution**

Execs$_{\mathcal{A}}$ set of all executions

First state of α is $\alpha.fstate = \tau_0.fstate$

If α is **finite and closed** $\tau_0\, a_1\, \tau_1 a_2 \tau_2\ \ldots \tau_k$ then $\alpha.lstate = \tau_k.lstate$

A state $\boldsymbol{x}$ is *reachable* if there exists an execution α with $\alpha.lstate = \boldsymbol{x}$

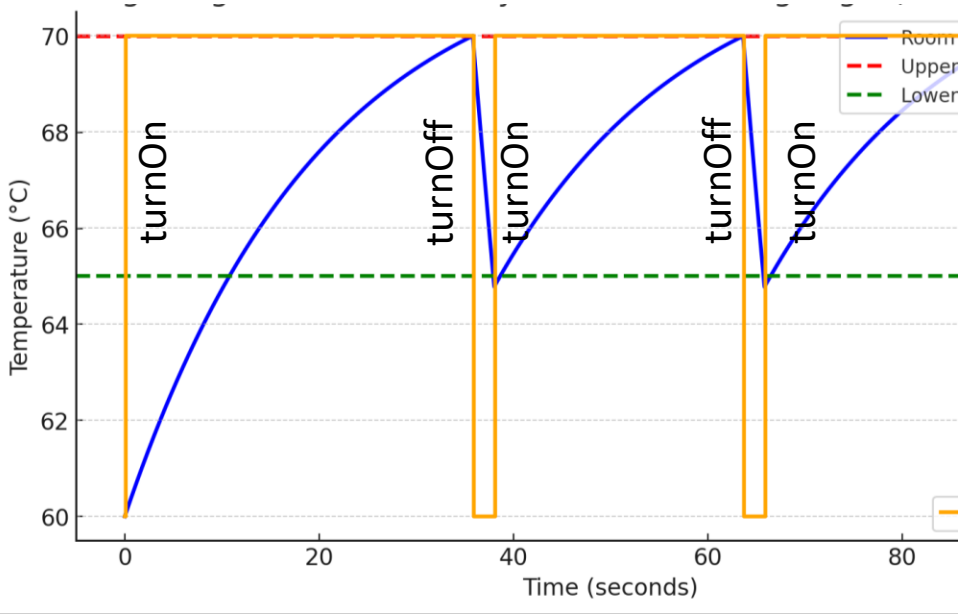


Mode invariants: Urgency in transitions

The trajectories in a hybrid automaton are often specified as solutions of ODEs

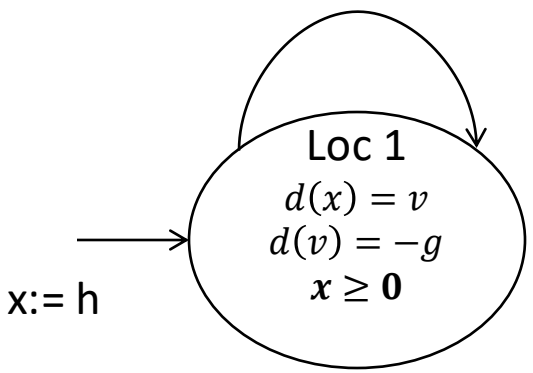
- $\dot{x} = f(x)$
- $\dot{x} = Ax$
- $\dot{x} = c$

Solutions of ODEs $\tau(x_0, t)$ define a family of trajectories which are closed under prefix, suffix, and concatenation



Nothing prevents a trajectory to continue forever unless an action is forced to occur by a **mode invariant**

Example. Bouncing Ball



automaton Bouncingball(c,h,g)

variables: x: Reals := h, v: Reals := 0

actions: bounce

transitions:

bounce

pre *x = 0 /\ v < 0*

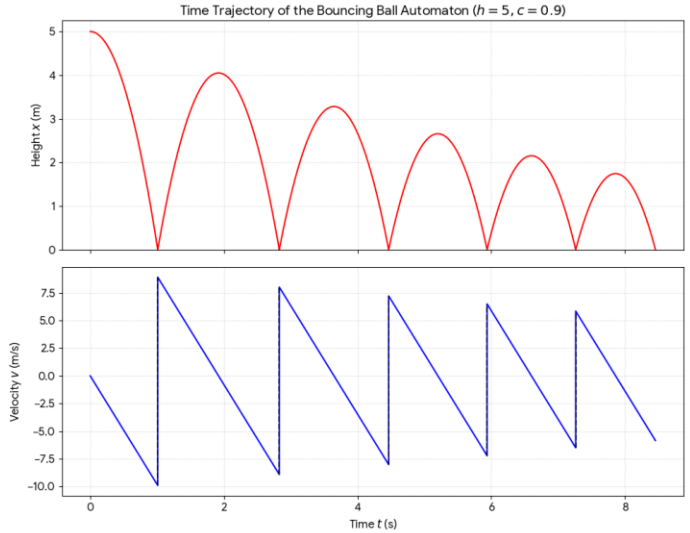
eff v := -cv

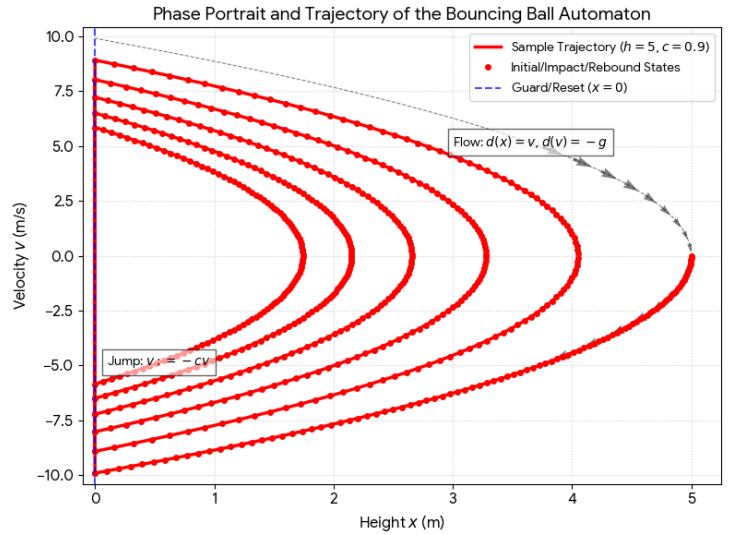
trajectories:

Loc1

evolve d(x) = v; d(v) = -g

invariant $x \geq 0$

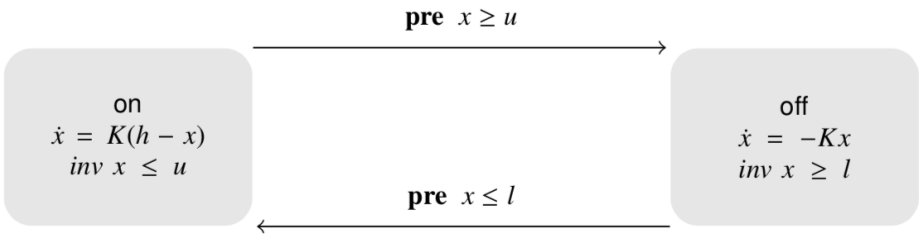




Graphical Representation used in many articles

mode invariant, not to be confused with invariants of the automaton

Urgency: Must transitions



An ***urgent*** transition (or action) is an action that has to occur as soon as it is enabled

Introduce special syntax for creating urgent transitions, but we can use mode invariants to accomplish this

automaton Thermostat(u, l, K, h,d : Real) where
$u > l$
type *Status* enumeration [*on*, *off*]
actions
turnOn; turnOff;

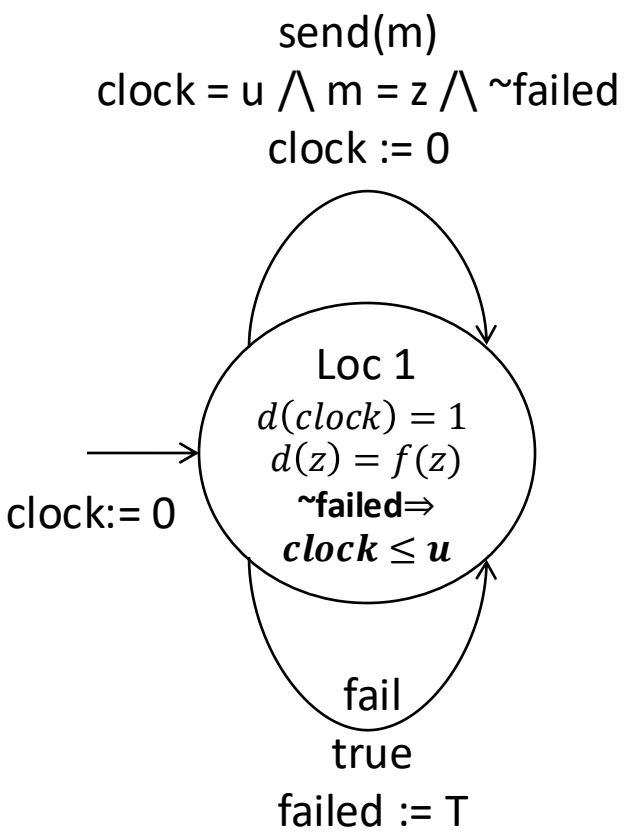
variables
x: Real := l *loc*: *Status* := *on*
transitions
turnOn
pre $x \leq l \wedge loc = off$
eff *loc* := *on*

turnOff
pre $x \geq u \wedge loc = on$
eff *loc* := *off*

trajectories
modeOn
evolve $d(x) = K(h - x)$
invariant $loc = on \wedge x \leq u + d$

modeOff
evolve $d(x) = -Kx$
invariant $loc = off \wedge x \geq l - d$

Another Example: Periodically Sending Process



Automaton PeriodicSend(u)

variables: analog

clock: Reals := 0, z:Reals, failed:Boolean := F

actions: send(m:Reals), fail

transitions:

send(m)

pre clock = u /\ m = z /\ ~failed

eff clock := 0

fail

pre true

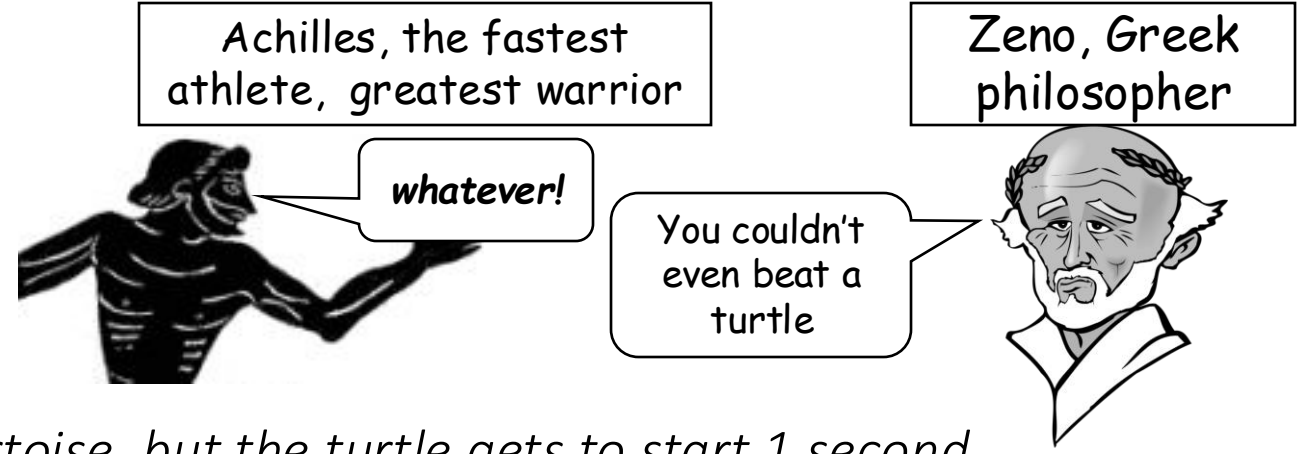
eff failed := T

trajectories:

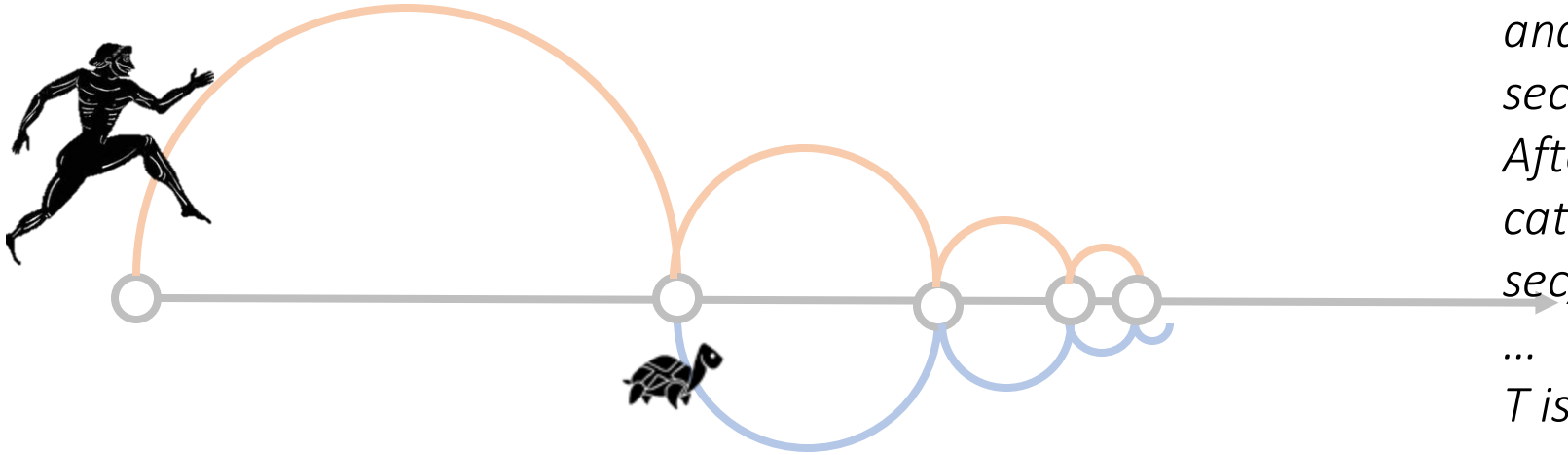
evolve d(clock) = 1, d(z) = f(z)

invariant failed \/ clock<=u

Zeno's Paradox



Achilles runs 10 times faster than than the tortoise, but the turtle gets to start 1 second earlier. Can Achilles ever catch Turtle?



After 1/10th of a second, Achilles reaches where the Turtle (T) started, and T has a head start of 1/10th second.
After another 1/100th of a second, A catches up to where T was at t=1/10 sec, but T has a head start of 1/100th
…
T is always ahead …

Lesson: Mixing discrete transitions with continuous motion can be tricky!

Defining stability of hybrid automata

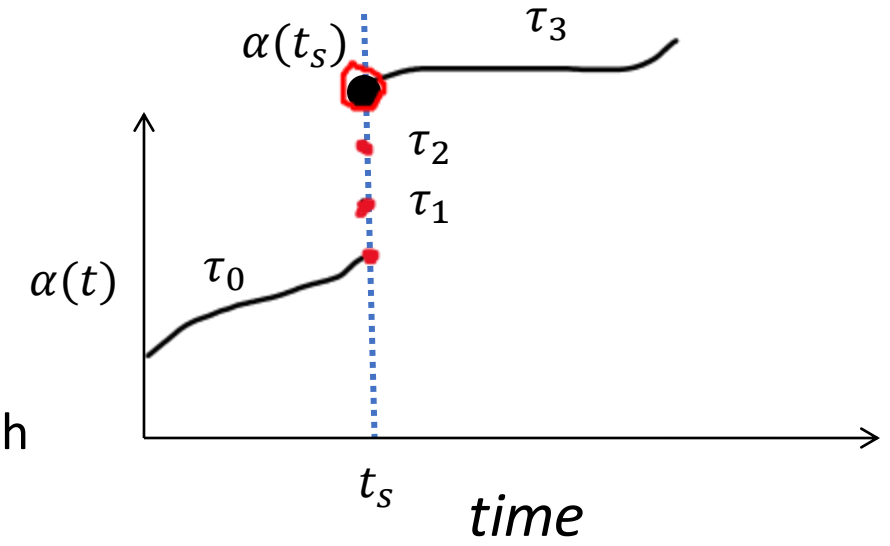
Given an admissible execution $\alpha = \tau_0\ a_1\ \tau_1 a_2 \tau_2\ \ldots$

Recall, stability is defined in terms of $|\alpha(t)|$; two issues

- We would like to view an execution as $\alpha: [0, \infty) \to val(X)$ But, how to define $\alpha(t)$?
- define $\alpha(t_s) = \alpha'.lstate$ where α' is the longest **prefix** of α with $\alpha'.ltime = t_s$
- Also, $\alpha(t)$ has valuations of discrete variables (loc) and continuous variable (x): ignore discrete variable when discussing stability $\alpha(t) \equiv \alpha(t)\lceil x$

Now, we can import all the definitions of Lyapunov stability and asymptotic stability from dynamical system models to hybrid automata

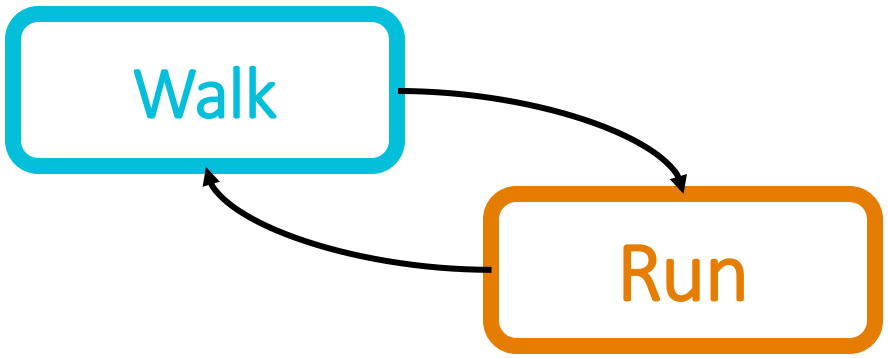
Question. How can we check stability of a hybrid automaton?

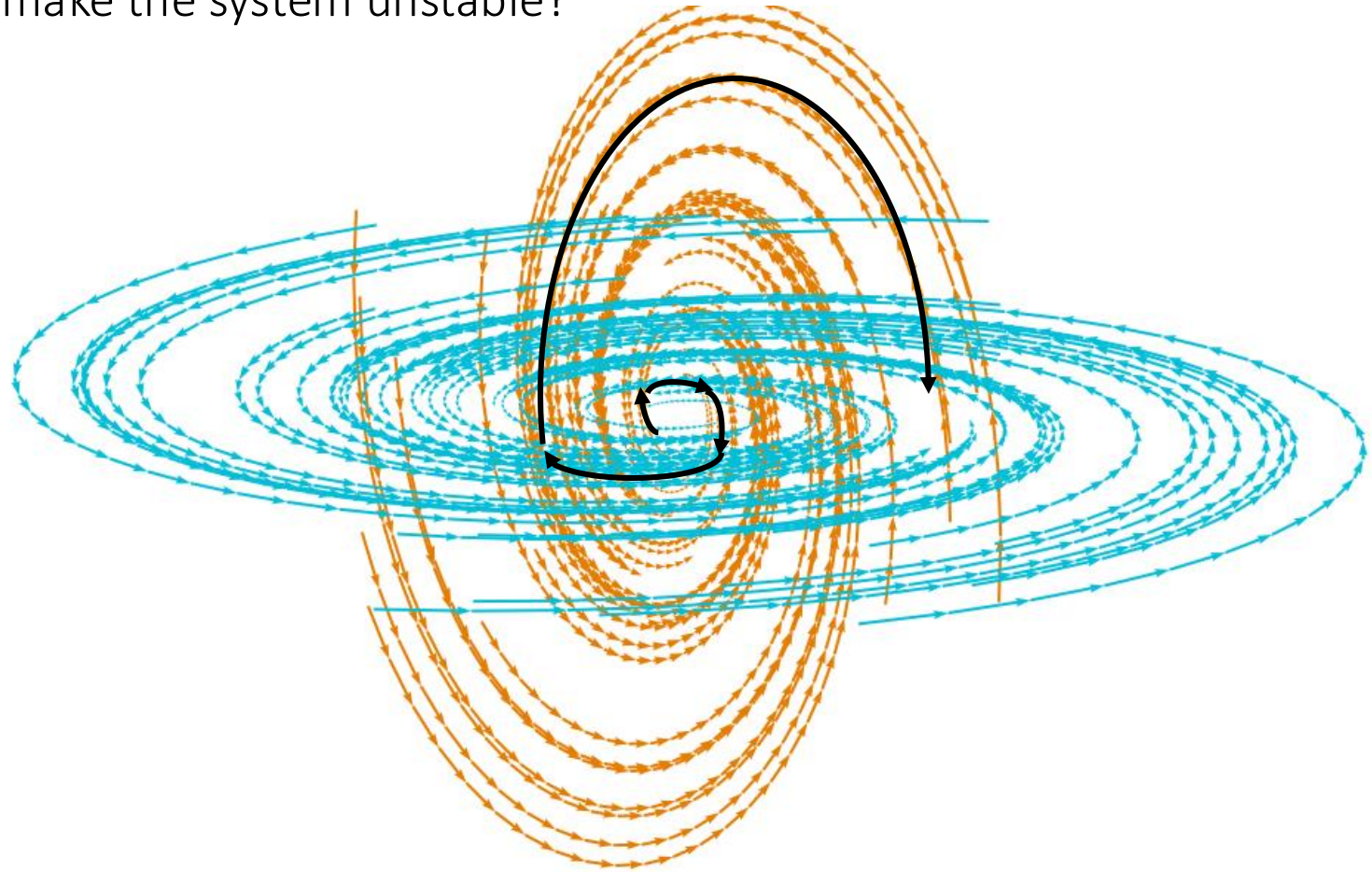


Stability of a hybrid automaton

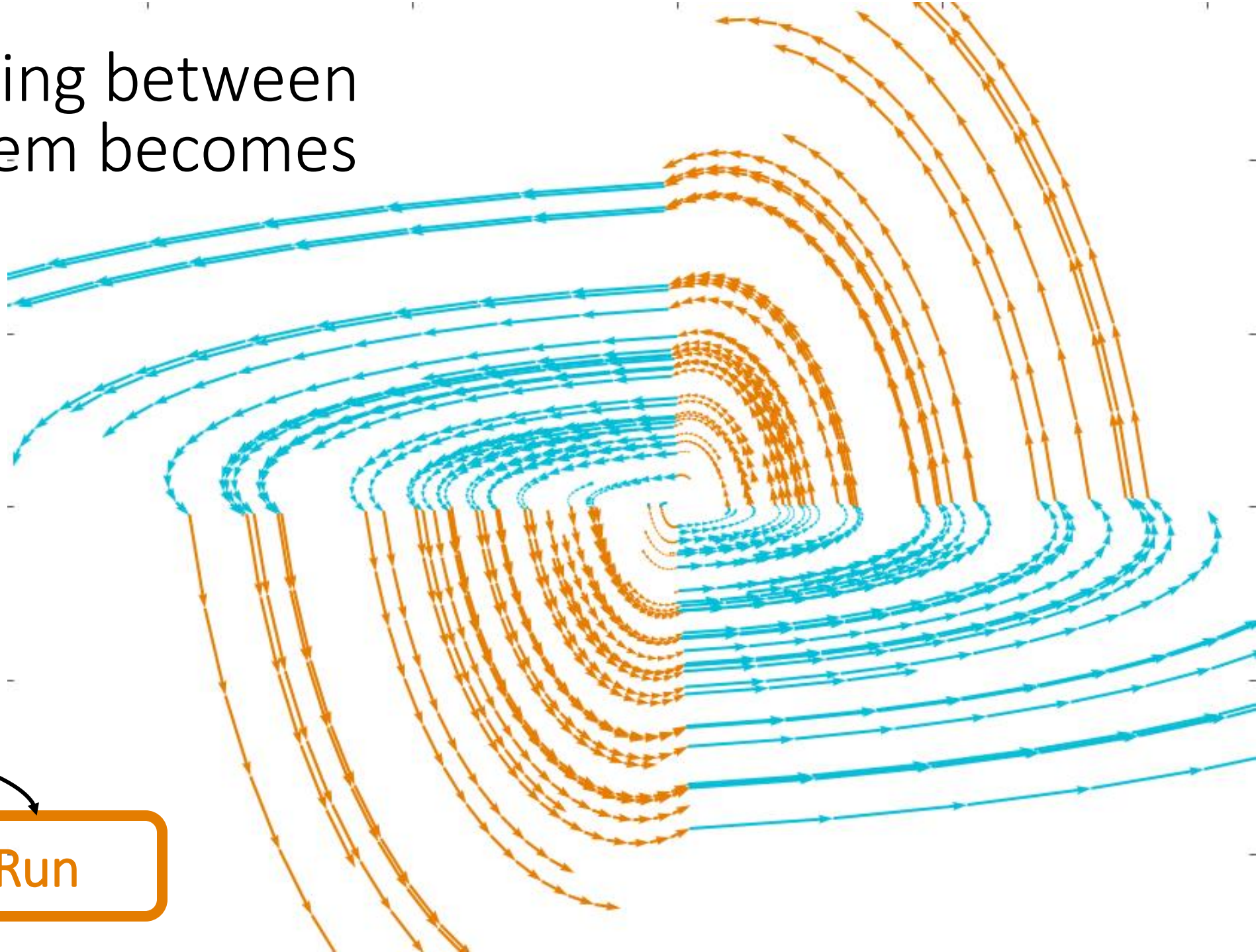
Each of the modes of a walking robot are asymptotically stable

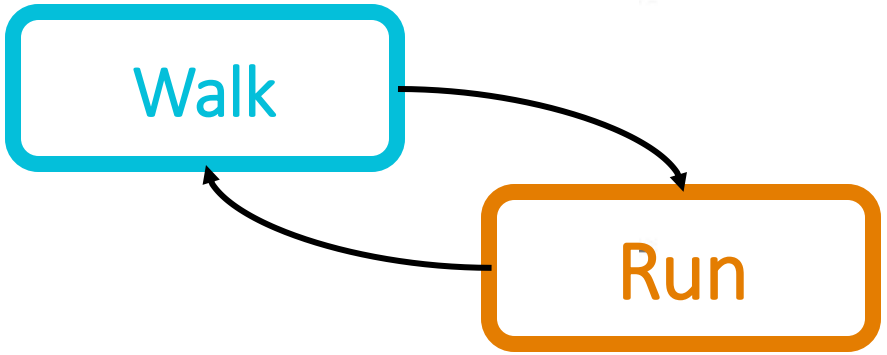
Is it possible to switch between them to make the system unstable?



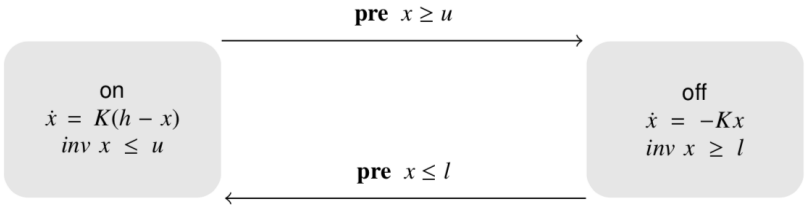


Yes! By switching between them the system becomes unstable



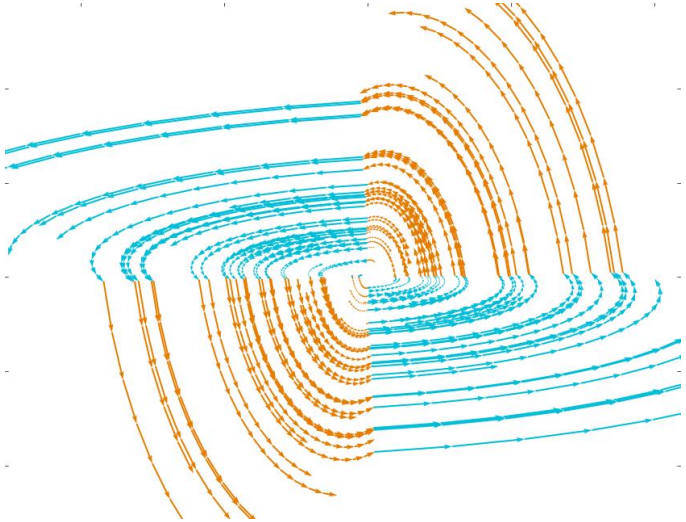


Summary

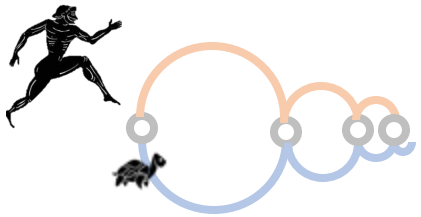


Interesting features in hybrid models

- Interaction between transitions, trajectories, and mode invariants can lead to extra sources of nondeterminism; transitions can be made urgent with mode invariants
- Mode switches can make the system unstable
- Zeno behavior



Inductive invariance proofs work with additional condition for trajectory closure



- Trajectory closure can be checked with subtangential condition, Lyapunov functions, Barrier certificates