Lecture 14: Modeling Cyberphysical Systems
Hybrid systems

Huan Zhang

huan@huan-zhang.com

Slides adapted from Prof. Sayan Mitra’s slides in Fall 2021

Deadlines

Homework 2 due 3/10, 11:59 pm CT

Two writing problems + two programming problems

YOU SHOULD START TODAY! (if you haven't started working on it)

Class project

Over 25+ project proposals recieved

You might be contacted by me for (optionally) forming a team of 2 if your proposed project is the same as another student.

Midterm project presentation: **3/26** and **3/28**.

- **5-min** presentations for each team. (5% of final grade)
- Slides due on **3/25**. We will compile all slides into a single file for fast switching.
- Presentation includes problem setting, proposed methodology, and initial results.
- I will give you some feedback after your presentation (1-min)

In addition: each person should give feedback for 5 projects that interest you most on each day. (total 10 feedbacks; count towards the **5% class participation** grades. Feedback will be submitted to Canvas and also shared to peers. Feedback template will be given.)

Review: dynamical systems

Behaviors of physical processes are described in terms of instantaneous laws

Common notation: $\frac{dx(t)}{dt} = f(x(t), u(t), t)\ -\ Eq. (1)$

where time $t \in \mathbb{R}$; **state** $x(t) \in \mathbb{R}^n$; $\textbf{input}\ \ u(t) \in \mathbb{R}^m$; $f: \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R} \rightarrow \mathbb{R}^n$

Example. $\frac{dx(t)}{dt} = v(t)\ ; \frac{dv(t)}{dt} =- g$

Initial value problem: Given system (1) and initial state $x_0 \in \mathbb{R}^n,\ t_0 \in \mathbb{R}$, and input u: $\mathbb{R} \rightarrow \mathbb{R}^m$, find a state trajectory or *solution* of (1).

Review: Linear time invariant system

$$\dot{x}(t) = Ax(t) + Bu(t)$$

Define Matrix exponential:

$$e^{At} = 1 + At + \frac{1}{2!}(At)^2 + \ \dots = \sum_{0}^{\infty} \frac{1}{k!}(At)^k$$

Theorem. $\xi(t, x_0, u) = \Phi(t)x_0 + \int_{t_0}^{t} \mathrm{e}^{A(t-\tau)} Bu(\tau)d\tau$

Zero input

Zero state

Here $\Phi(t) := e^{At}$ is the state-transition matrix

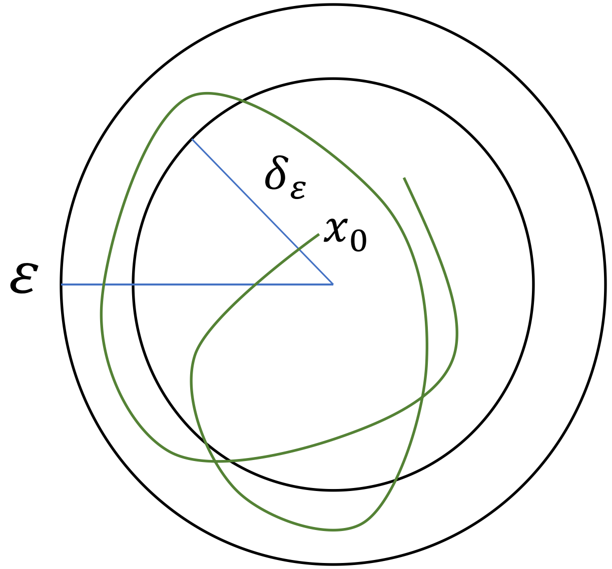
Review Lyapunov stability

Lyapunov stability: The system $\dot{x}(t) = f\big(x(t)\big)$ is said to be ***Lyapunov stable*** (at the origin) if

$\forall \varepsilon > 0,\ \exists\ \delta_\varepsilon > 0$ such that $|x_0| \leq \delta_\varepsilon \Rightarrow \forall$ t $\geq 0,\ |\xi(x_0, t)| \leq \varepsilon$.

"if we start the system close enough to the equilibrium, it remains close enough"

How is this related to invariants and reachable states ?

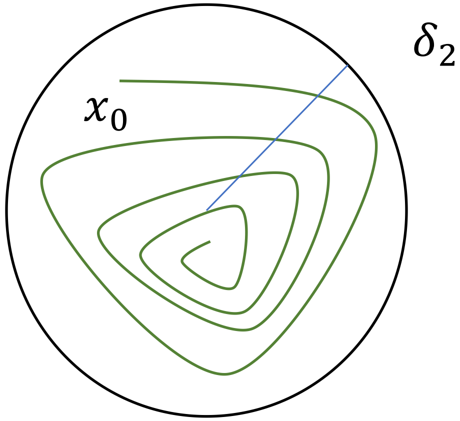


Review: Asymptotically stability

The system $\dot{x}(t) = f\big(x(t)\big)$ is said to be ***Asymptotically stable*** *(at the origin)* if it is Lyapunov stable and

$\exists \delta_2 > 0$ such that $\forall |x_0| \leq \delta_2$ as t $\rightarrow \infty$, $|\xi(x_0, t)| \rightarrow \mathbf{0}$.

If the property holds for any δ_2 then **Globally Asymptotically Stable**



Review: Verifying Stability

Theorem. (Lyapunov) Consider the system (1) with state space $x \in \mathbb{R}^n$ and suppose there exists a positive definite, continuously differentiable function $V: \mathbb{R}^n \to \mathbb{R}$. The system is:

1. Lyapunov stable if $\dot{V}(x) = \frac{\partial V}{\partial x} f(x) \leq 0$, for all $x \neq 0$
2. Asymptotically stable if $\dot{V}(x) < 0$, for all $x \neq 0$
3. It is globally AS if V is also radially unbounded.

(V is radially unbounded if $||x|| \to \infty \Rightarrow V(x) \to \infty$)

Today's lecture: Hybrid systems

Discrete transition systems, automata

Dynamical systems
Differential inclusions

Markov chains

↓

Probabilistic automata, Markov decision processes (MDP)

↓

Continuous time, continuous state MDPs

Hybrid systems

↓ ↓

Stochastic Hybrid systems

Slides adapted from Prof. Sayan Mitra's slides in Fall 2021

Outline

- Hybrid automata
- Executions
- Special kinds of executions: admissible, Zeno
- Hybrid stability

Recall from Lecture 2. language defines an automaton

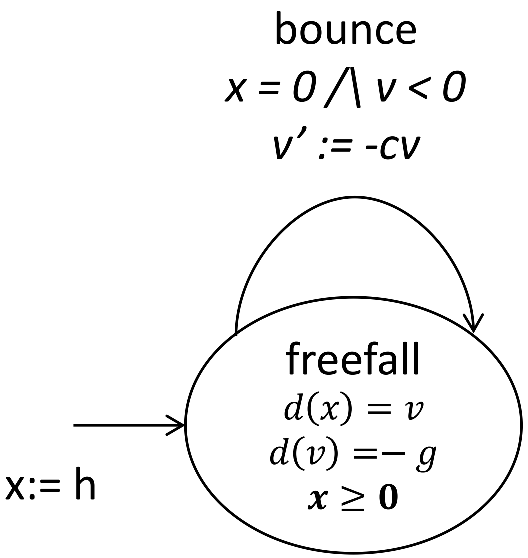
An **automaton** is a tuple $\mathcal{A} = \langle X, \Theta, A, \mathcal{D} \rangle$ where

- X is a set of names of variables; each variable $x \in X$ is associated with a type, $type(x)$
  - A **valuation** for X maps each variable in X to its type
  - Set of all valuations:$val(X)$ this is sometimes identified as the **state space** of the automaton
- $\Theta \subseteq val(X)$ is the 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')
  - 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]
```

Bouncing Ball: Hello world of CPS



Graphical Representation used in many articles

automaton Bouncingball(c,h,g)

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

actions: bounce

transitions:

bounce

pre *x = 0* ∧ *v < 0*

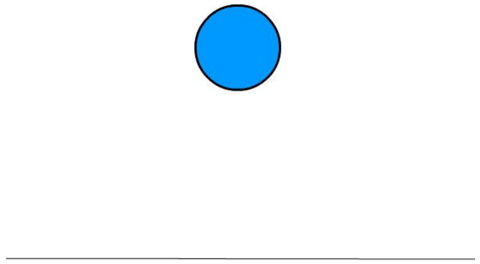
eff v := -cv

trajectories:

freefall

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

invariant $\boldsymbol{x} \geq \boldsymbol{0}$



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

Trajectories

Given a set of variables X and a **time interval** J which can be of the form $[0, T]$, $[0, T) or$ $[0, \infty)$, a ***trajectory*** for X is a function τ: $J \rightarrow val(X)$

We will specify τ as **solutions of differential equations**

The **first state** of a trajectory $\tau.fstate$: $= \tau(0)$

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

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

The domain of $\tau.dom = J$

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

Operations on trajectories: prefix, suffix, concatenation

A **prefix** τ' of a trajectory τ: $[0, T] \rightarrow val(X)$, is a function τ': $[0, T'] \rightarrow val(X)$ 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

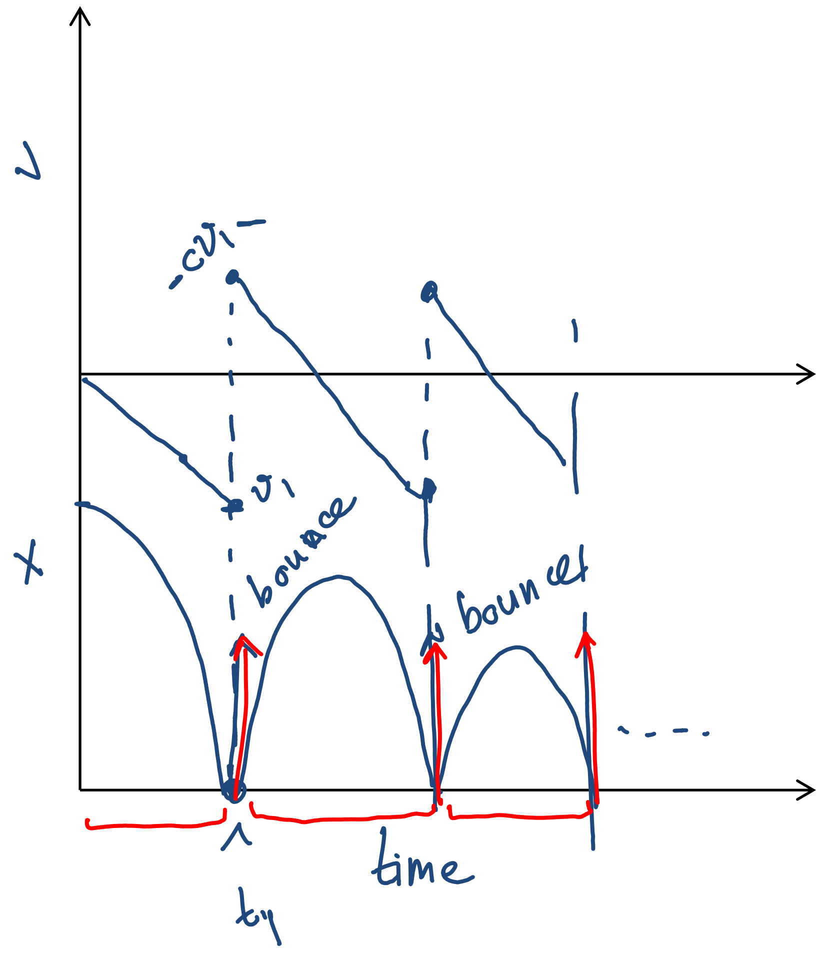
Closed: For every τ in $\mathcal{T}$, any suffix τ' or τ is also in $\mathcal{T}$

Semantics: Executions and Traces

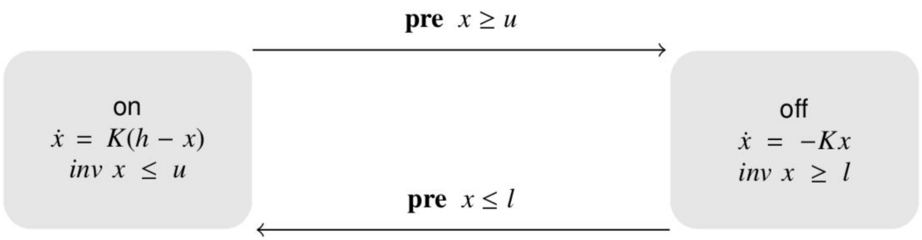
- An ***execution fragment*** of $\mathcal{A}$ is an (possibly infinite) alternating (A, X)-sequence $\alpha = \tau_0\ a_1\ \tau_1 a_2 \tau_2\ \ldots$ where
  - ∀ i, $\tau_i.lstate \overset{a_{i+1}}{\longrightarrow} \tau_{i+1}.fstate$
- If τ_0.fstate $\in\ \Theta$ then α is an **execution**
- **Execs**$_{\mathcal{A}}$ set of all executions
- The first state of an execution α is: $\alpha.fstate = \tau_0.fstate$
- If the execution 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 $\alpha.lstate = \boldsymbol{x}$

Semantics: Executions and Traces

- An ***execution fragment*** of $\mathcal{A}$ is an (possibly infinite) alternating (A, X)-sequence $\alpha = \tau_0\ a_1\ \tau_1 a_2 \tau_2 \ldots$ where
  - ∀ 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
- The first state of an execution α is: $\alpha.fstate = \tau_0.fstate$
- If the execution 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 $\alpha.lstate = \boldsymbol{x}$



Thermostat variations



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 $x \leq l$ ∧ *loc*=*off*
eff *loc* := *on*

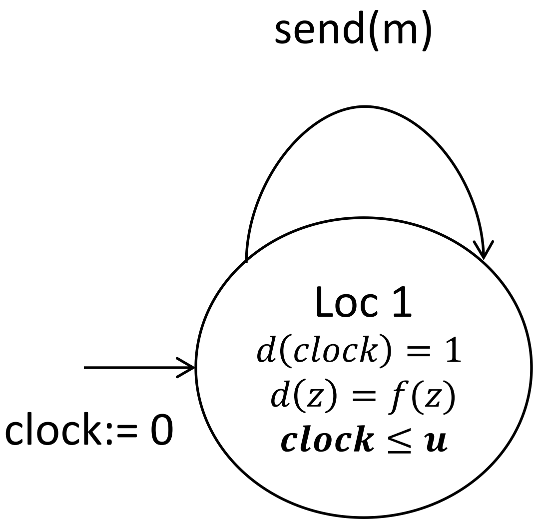
turnOff
pre $x \geq u$ ∧ *loc*=*on*
eff *loc* := *off*

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

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

- Determinism vs nondeterminism
- mode invariants

Another Example: Periodically Sending Process



Automaton PeriodicSend(u, f)

variables:

clock: Reals := 0, z:Reals

actions: send(m:Reals)

transitions:

send(m)

pre (clock = u) ∧ (m = z)

eff clock := 0

trajectories:

Loc1

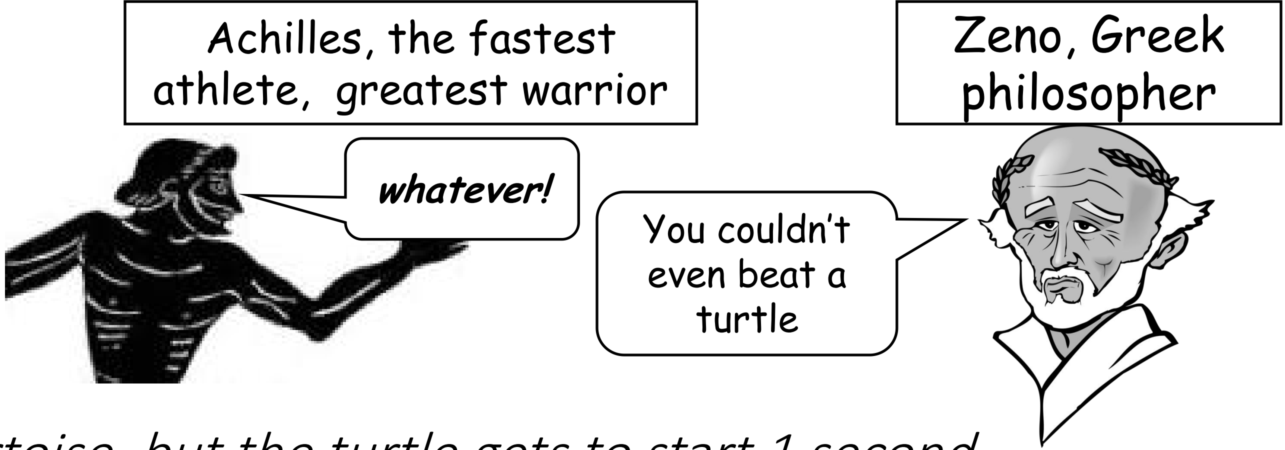
evolve d(clock) = 1, d(z) = f(z)

invariant clock<=u

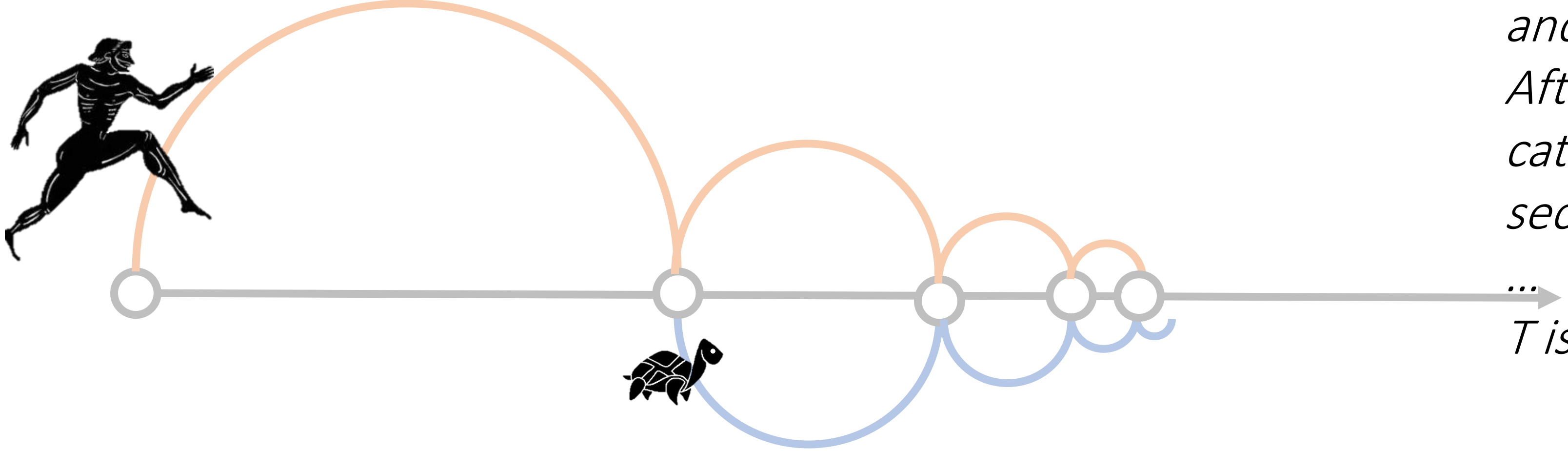
Special kinds of executions

- **Infinite**: Infinite sequence of transitions and trajectories $\tau_0\ a_1\ \tau_1 a_2 \tau_2\ \ldots$
- **Closed**: Finite with final trajectory with closed domain $\tau_0\ a_1\ \tau_1 a_2 \tau_2\ \ldots \tau_k$ and $\tau_k.dom = [0, T]$
- **Admissible**: Infinite duration
  - May or may not be infinite
  - $\tau_0\ a_1\ \tau_1 a_2 \tau_2\ \ldots$
  - $\tau_0\ a_1\ \tau_1 a_2 \tau_2\ \ldots \tau_k$ with $\tau_k.dom = [0, \infty)$
- **Zeno**: Infinite but not admissible
  - Infinite number of transitions in finite time

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

…

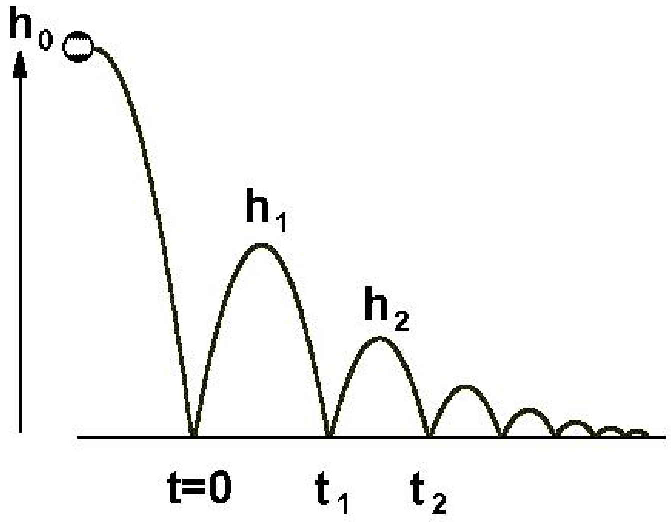
T is always ahead …

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

Zeno's Paradox: bouncing ball

- Infinite number of bounces to come to reset!
- Yet it needs finite time to come to reset

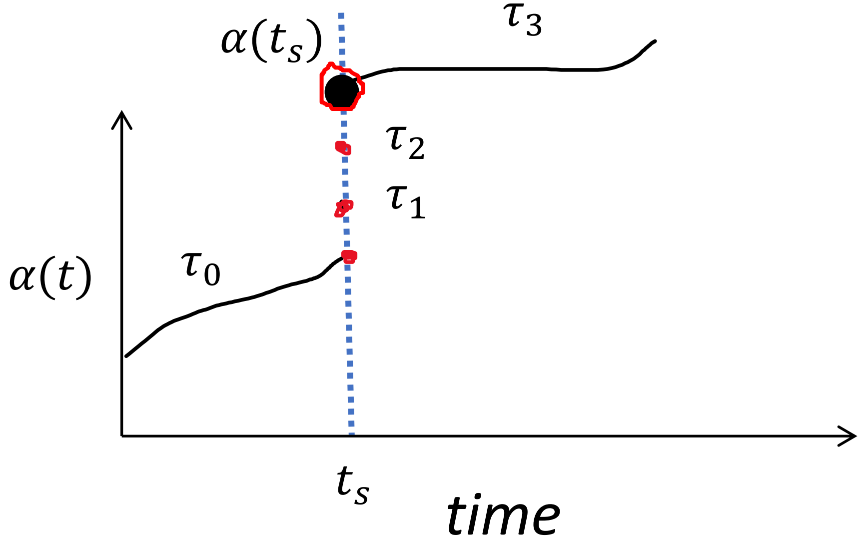
- The time for each bounce (e.g., t_1 - t_0, t_2 - t_1, etc) is a geometric series



Read: https://www.millersville.edu/physics/experiments/045/

Defining stability of hybrid automata

- Given an *admissible* (infinite duration) execution $\alpha = \tau_0 \; a_1 \; \tau_1 a_2 \tau_2 \; \ldots$
- To reason about stability of an execution, we would like to view an execution as $\alpha : [0, \infty) \rightarrow val(X)$
- But, how can we define $\alpha(t)$?
- define $\alpha(t_s) = \alpha'.lstate$ where α' is the longest **prefix** of α with $\alpha'.ltime = t_s$



Defining stability of hybrid automata

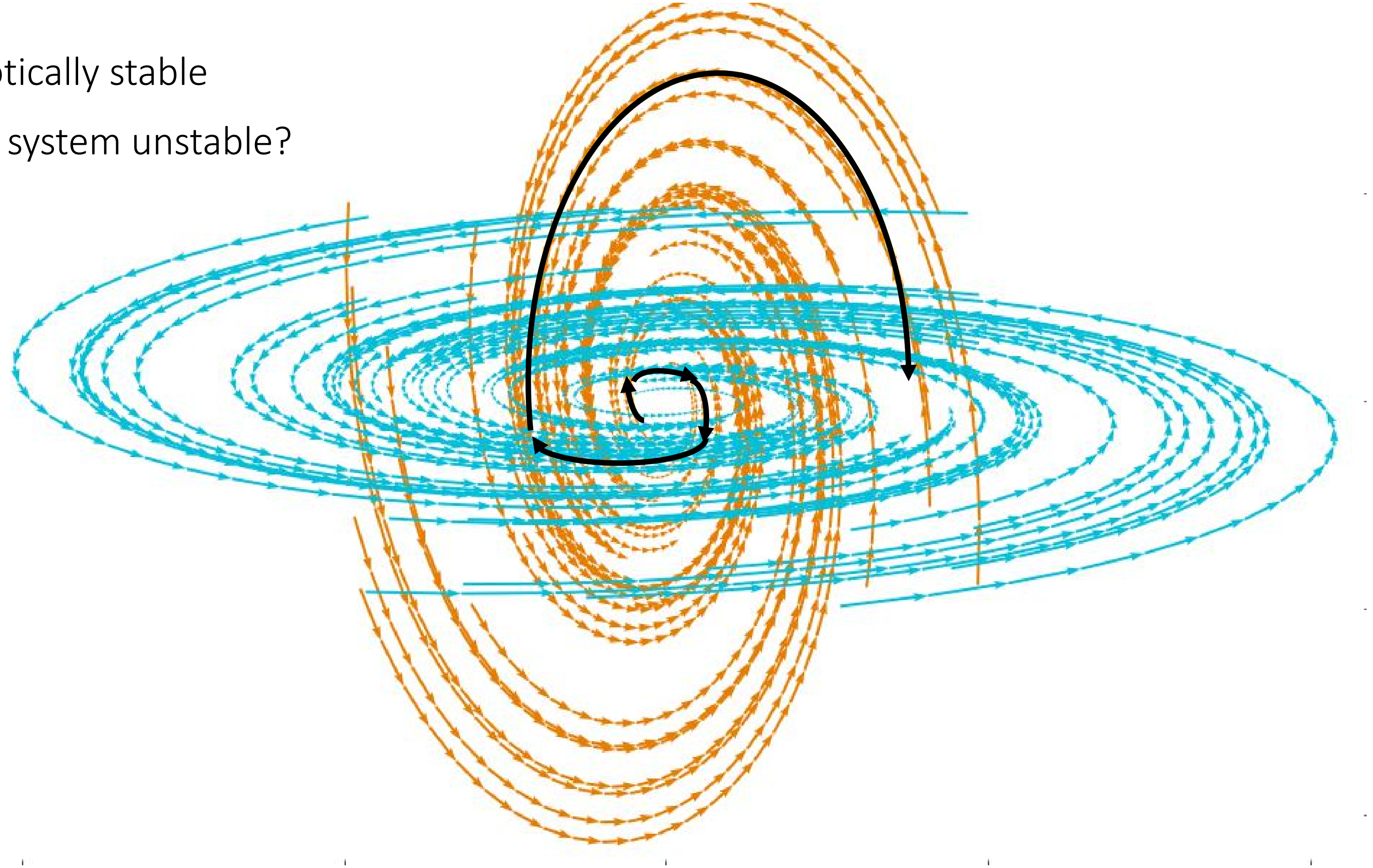
- An hybrid automata is globally uniformly asymptotically stable if:
- For any $\varepsilon > 0$ and any state v_0, there is a time T such that for any excution fragment α starting from v_0, for all $t \geq T$, $||\alpha(t)|| < \varepsilon$

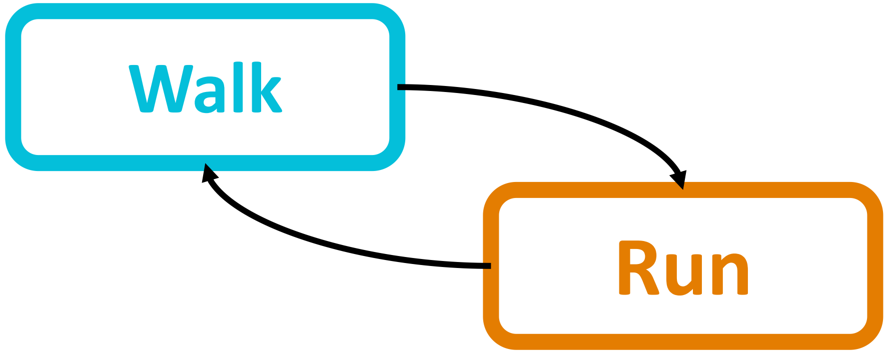
Hybrid Instability

Can we verify the stability of a hyrbid system by just verifying the stability of each mode?

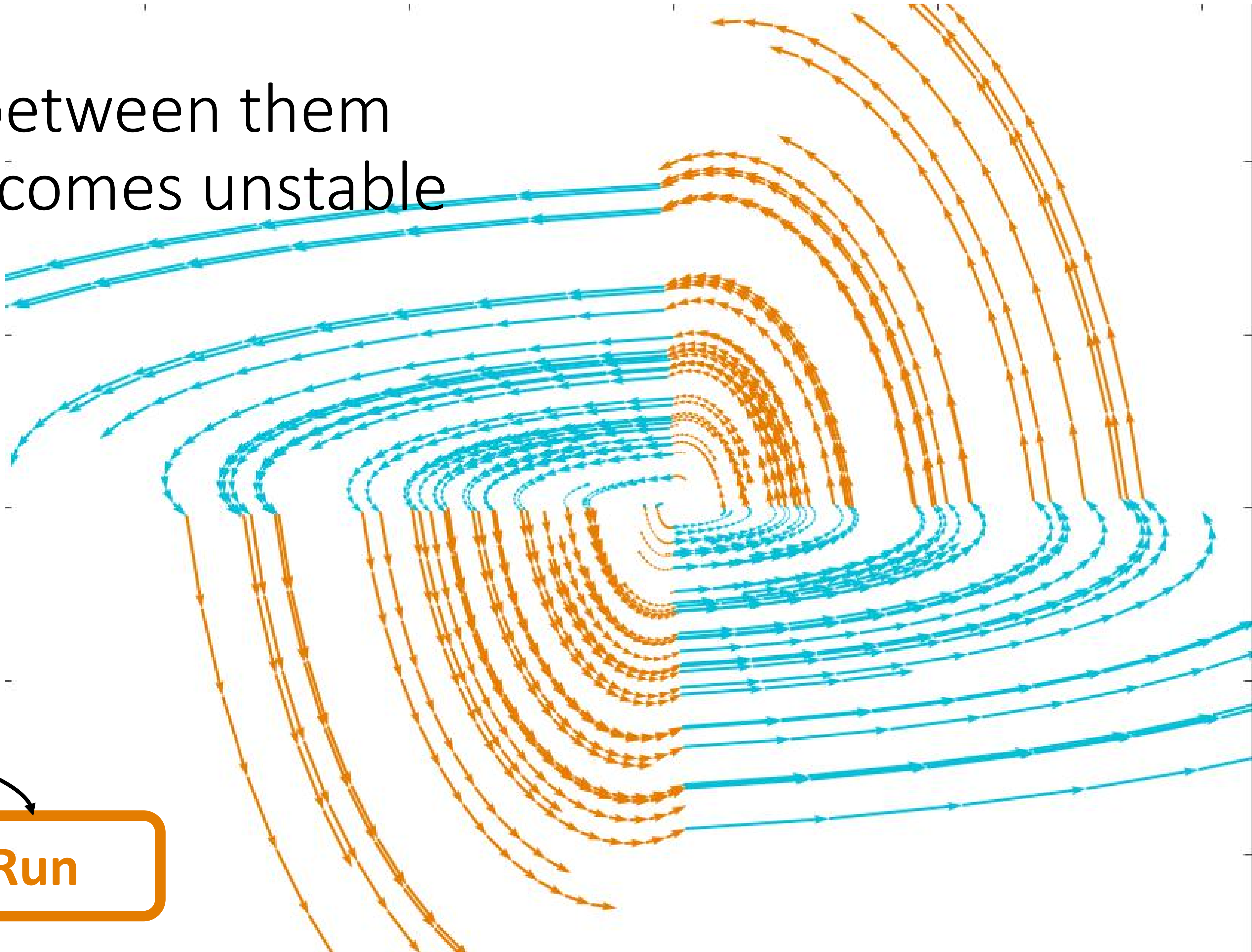
Each of the modes of a walking robot are asymptotically stable

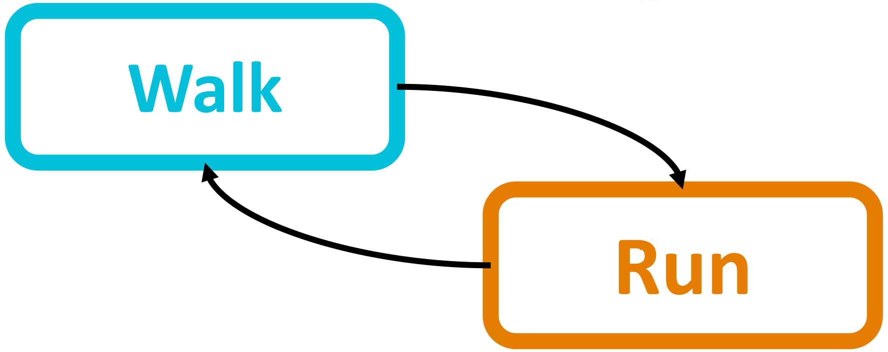
Is it possible to switch between them to make the system unstable?



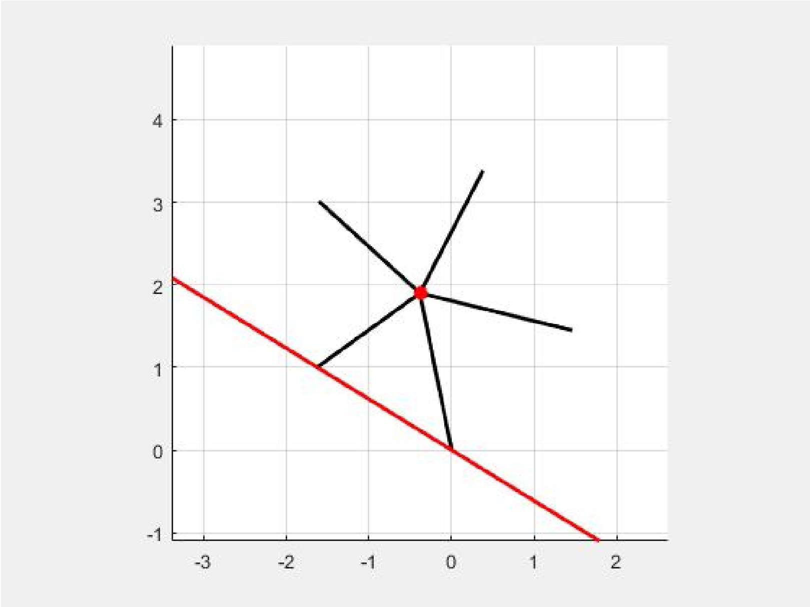


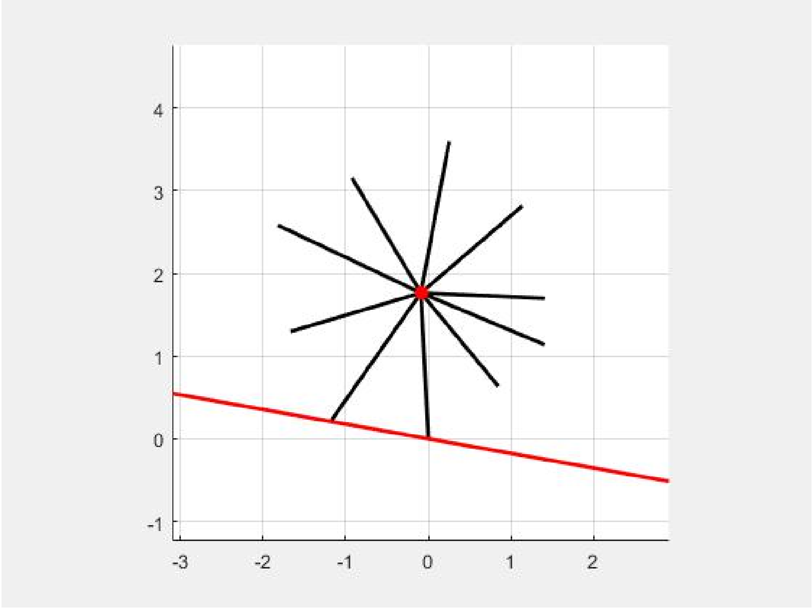
By switching between them the system becomes unstable



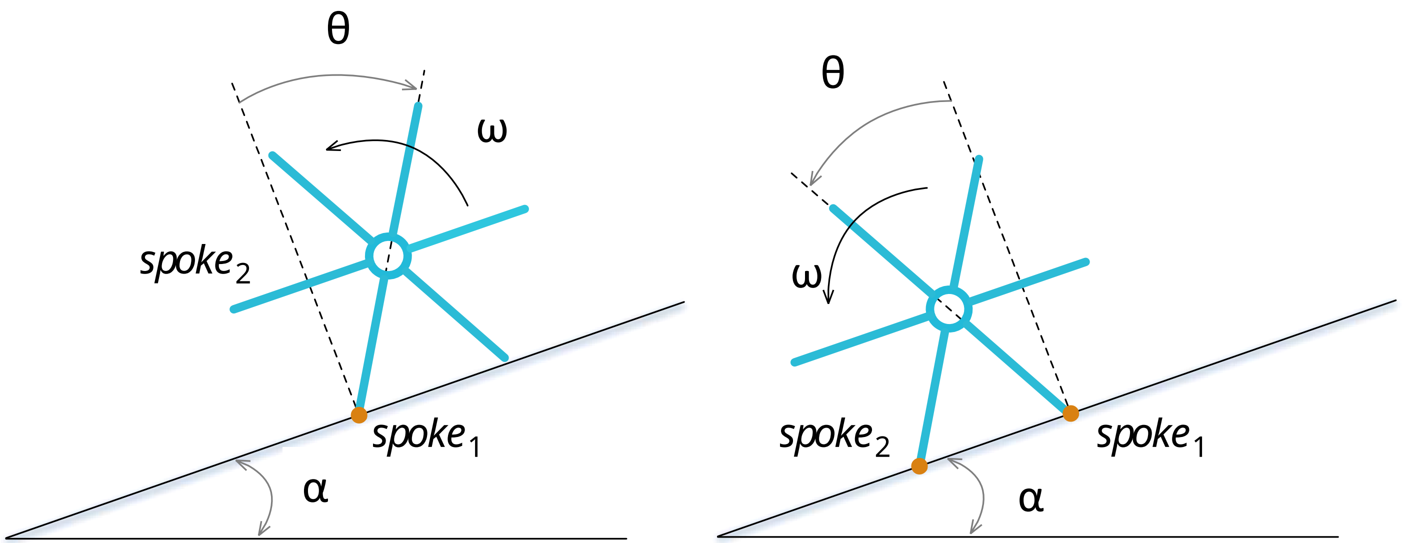


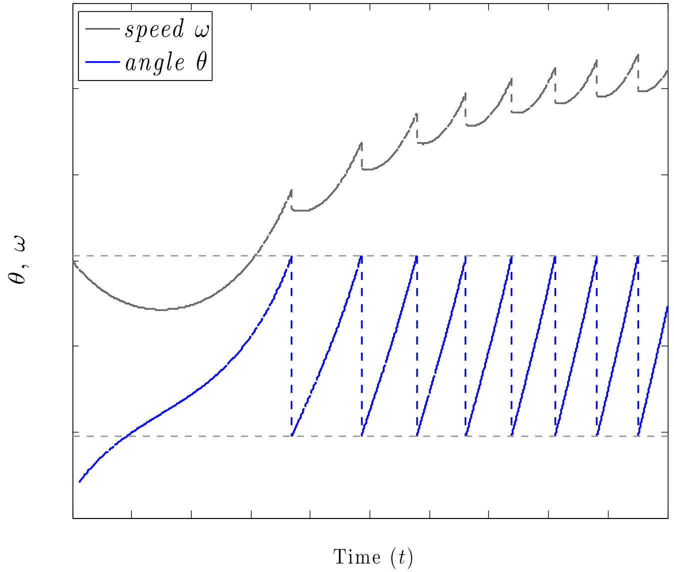
Rimless wheel: another example of hybrid system





Rimless wheel





automaton RimlessWheel(α, μ: Real, n: Nat)
  const β: Real := 2 π/n
  type Spokes: enumeration [1,...,n]
  actions
    impact
  variables
    pivot: Spokes :=1
    θ:Real := 0
    ω: Real := 0
  transitions
    impact
      pre $\theta \geq \beta/2$
      eff pivot := pivot + 1 mod n
        $\theta :=- \beta/2$
        $\omega := \mu\omega$

  trajectories
    swing
    evolve
      $\mathrm{d}(\theta) = \omega$
      $d(\omega) = \sin(\theta + \alpha)$
    invariant $\theta \leq \frac{\beta}{2}$

Invariants and reachability

- A state $\boldsymbol{x}$ of automaton $\mathcal{A}$ is ***reachable*** if there exists an execution α with $\alpha.lstate = \boldsymbol{x}$
- $Reach_{\mathcal{A}}(\Theta)$ is the set of all reachable state from Θ
- $Reach_{\mathcal{A}}(\Theta, T)$ is the set of states reachable within time T
- $Reach_{\mathcal{A}}(\Theta, k)$ is the set of states reachable within k transitions
- $Reach_{\mathcal{A}}(\Theta, T, k)$ is the set of states reachable up to time k transitions and time T
- An invariant $I \subseteq val(X)$ is a set of states that contains $Reach_{\mathcal{A}}(\Theta)$