# Cyberphysical Systems: Part 2

Sayan Mitra

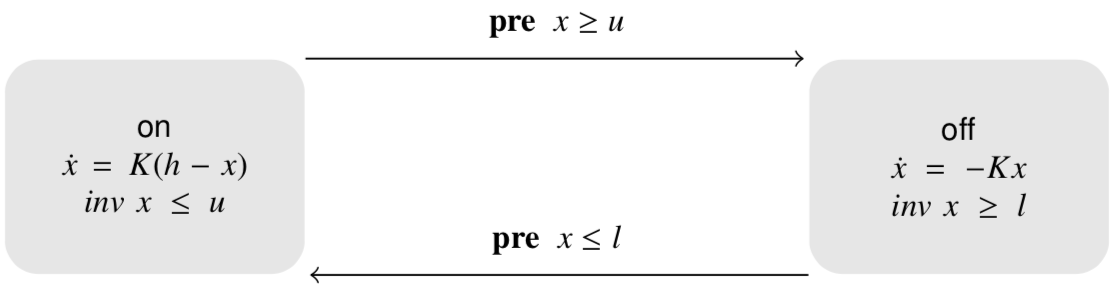
Verifying cyberphysical systems

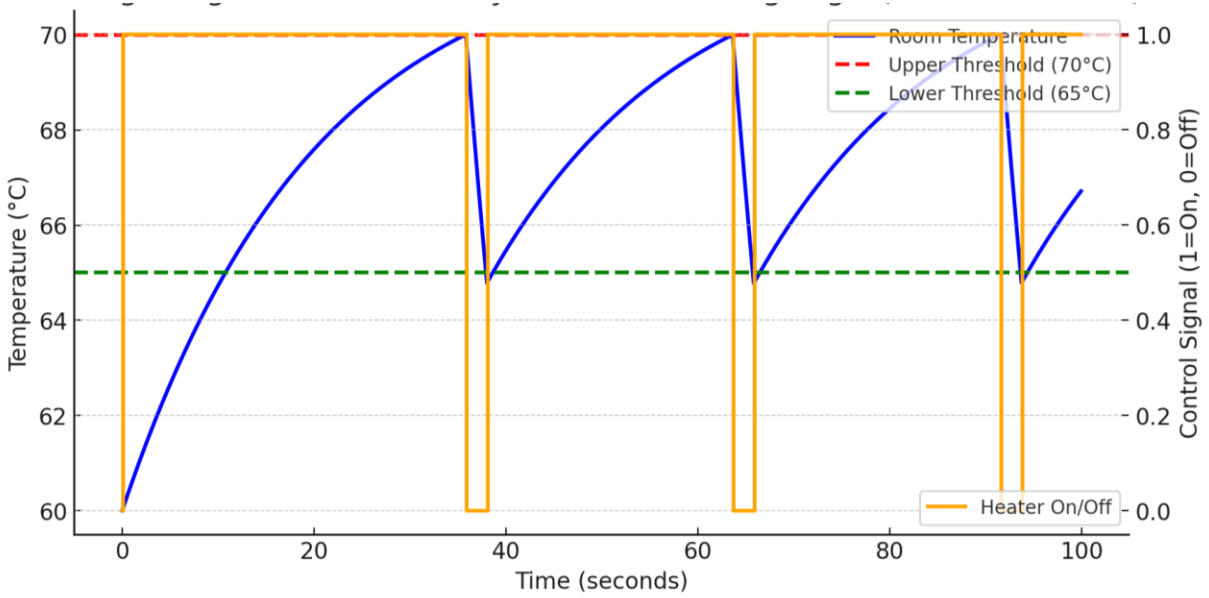
mitras@illinois.edu

# Outline

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

# 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

# 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

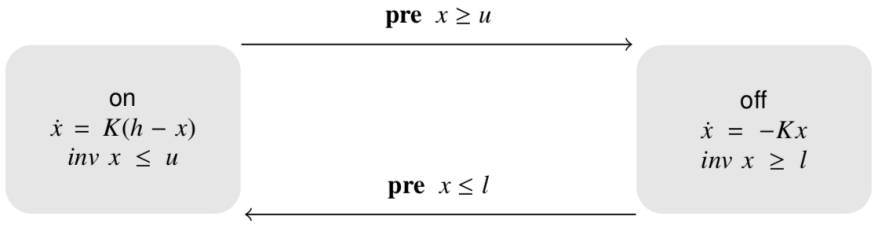
- $\forall$ i, $\tau_i.lstate \xrightarrow{a_{i+1}} \tau_{i+1}.fstate$
- If $\tau_0$.fstate $\in \Theta$ then $\alpha$ is an **execution**

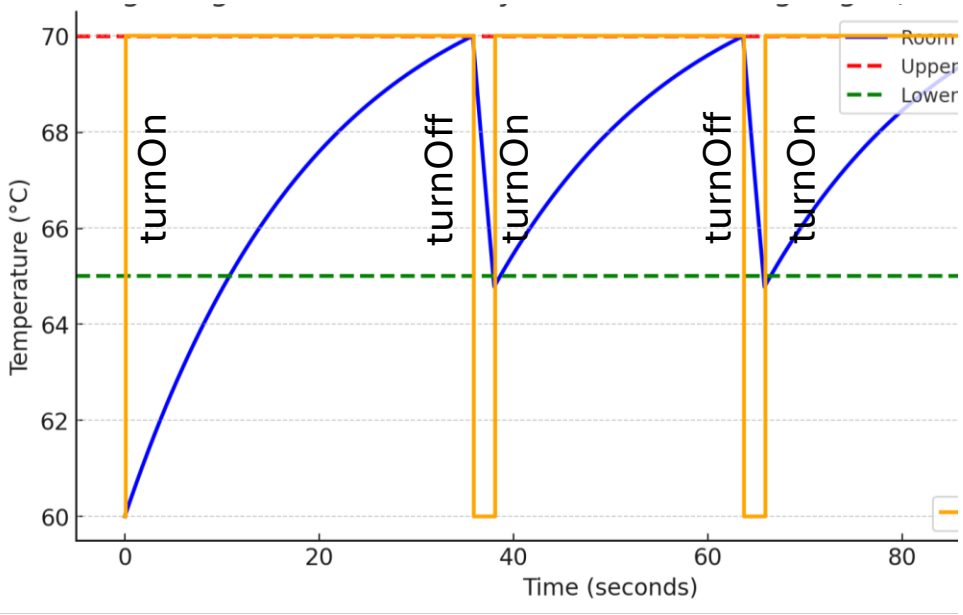
**Execs**$_{\mathcal{A}}$ set of all executions

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

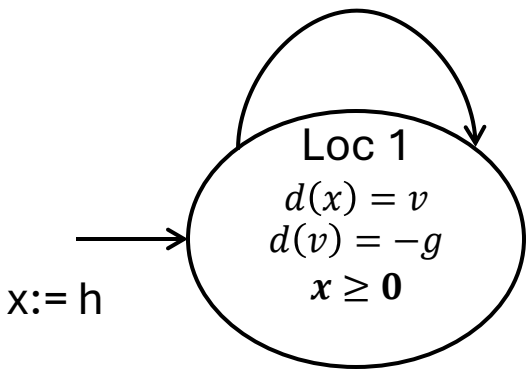
If $\alpha$ is **finite and closed** $\tau_0\ a_1\ \tau_1 a_2 \tau_2\ \ldots \tau_k$ then $\alpha.lstate = \tau_k.lstate$

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



# Example. Bouncing Ball




**automaton** Bouncingball(c,h,g)

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

**actions:** bounce

**transitions:**

bounce

**pre** *x = 0 /\ v < 0*

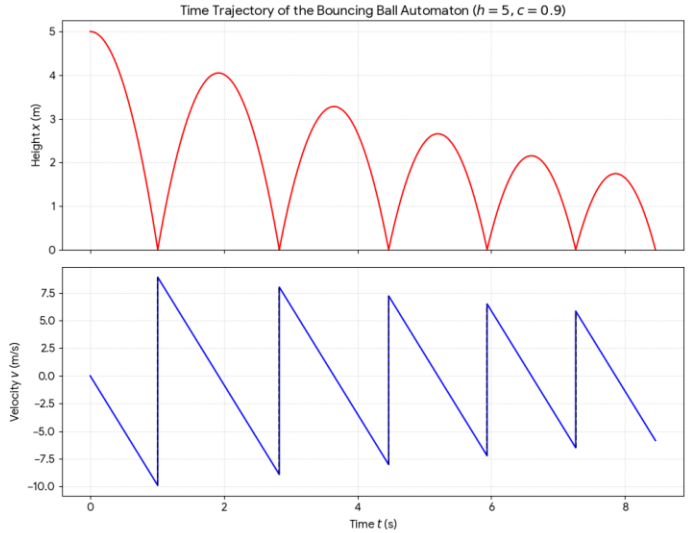
**eff** v := -cv

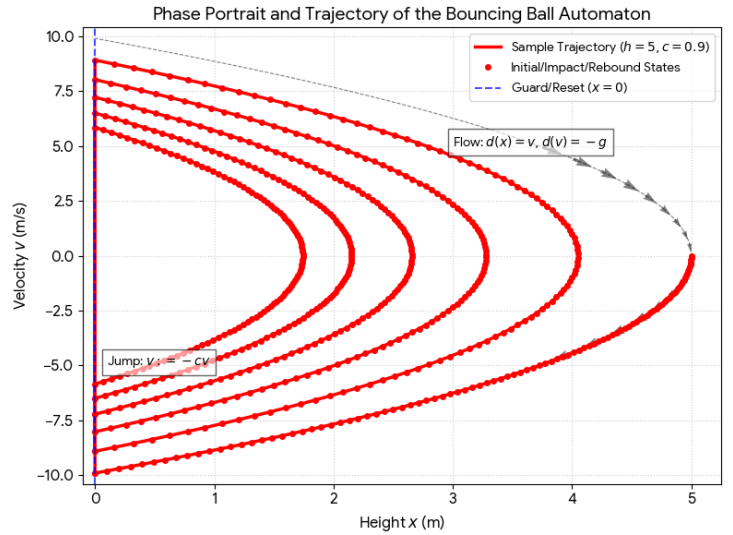
**trajectories:**

Loc1

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

**invariant** $x \geq 0$

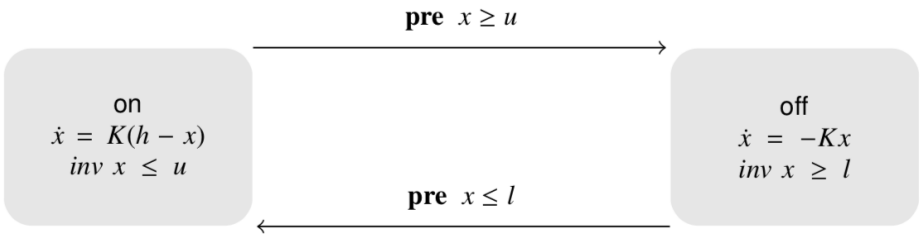



mode invariant, not to be confused with invariants of the automaton

Graphical Representation used in many articles

# 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;

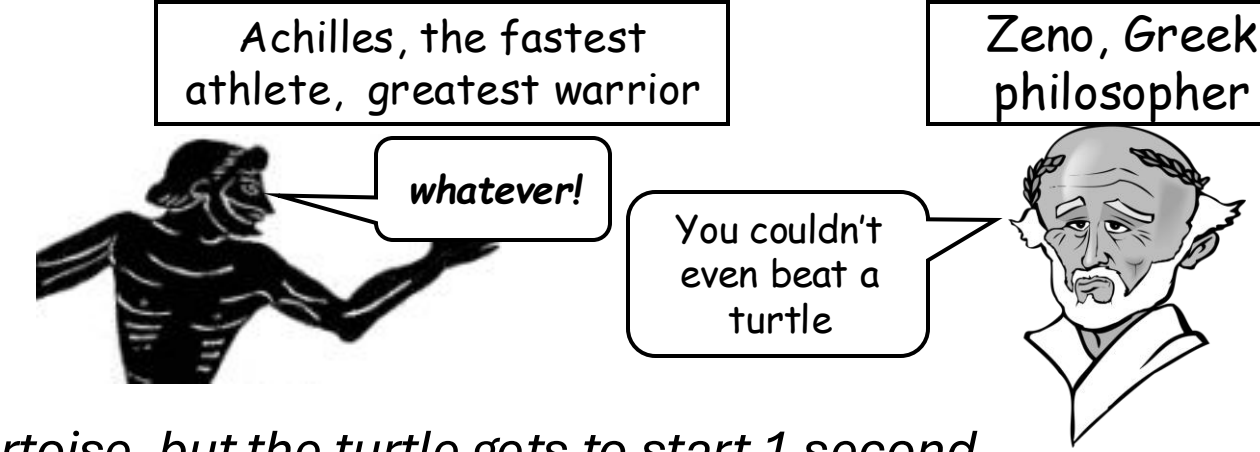
**variables**
$x$: Real := $l$ *loc*: *Status* := *on*
**transitions**
turnOn
**pre** $x \leq l \wedge loc = off$
**eff** *loc* := *on*

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

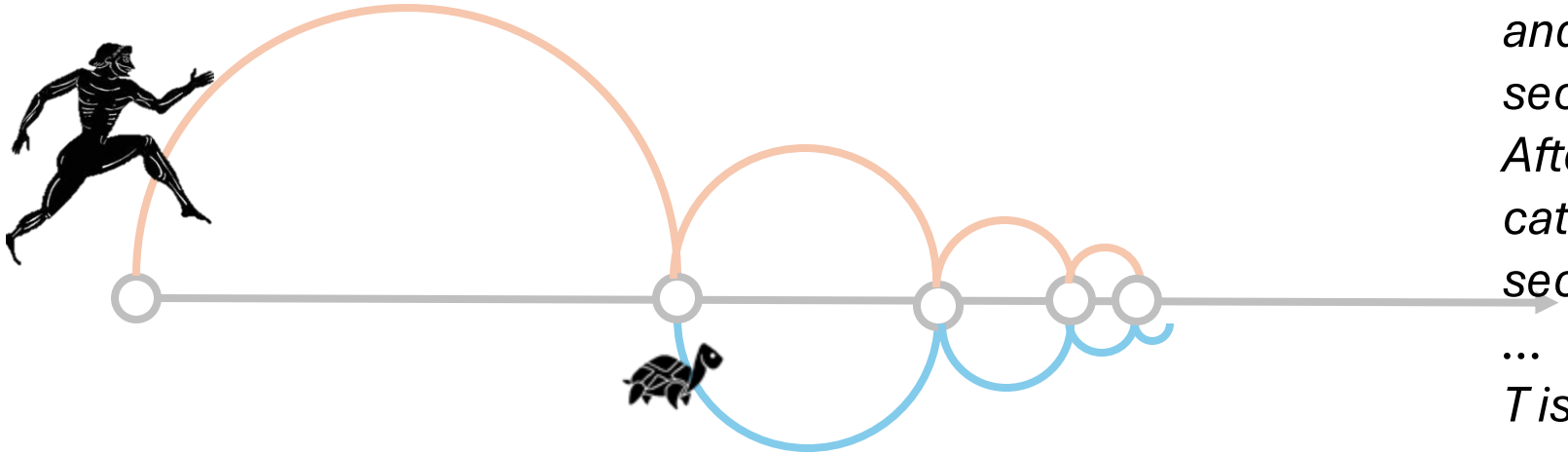
**trajectories**
modeOn
**evolve** $d(x) = K(h - x)$
**invariant** $loc = on \wedge x \leq u + d$

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

# 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?*



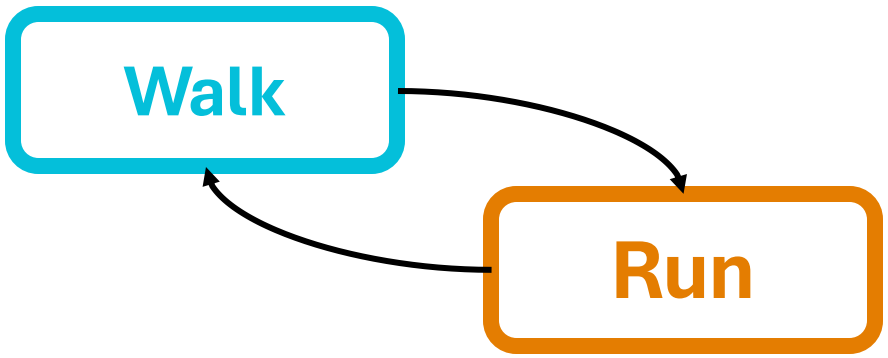
*After 1/10$^{th}$ of a second, Achilles reaches where the Turtle (T) started, and T has a head start of 1/10$^{th}$ second.*
*After another 1/100$^{th}$ of a second, A catches up to where T was at t=1/10 sec, but T has a head start of 1/100$^{th}$ ...*
*T is always ahead ...*

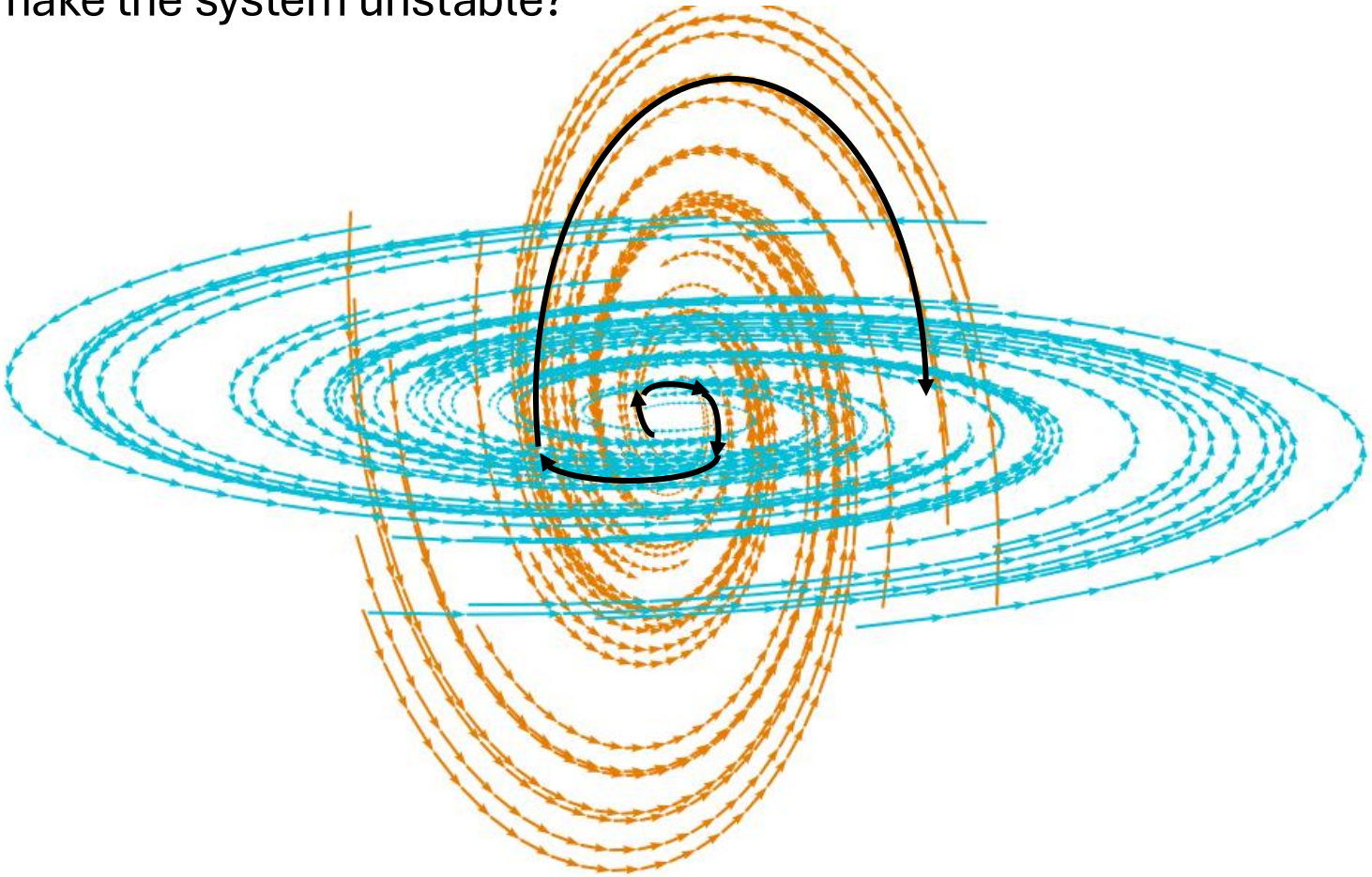
*Lesson: Mixing discrete transitions with continuous motion can be tricky!*

# 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?

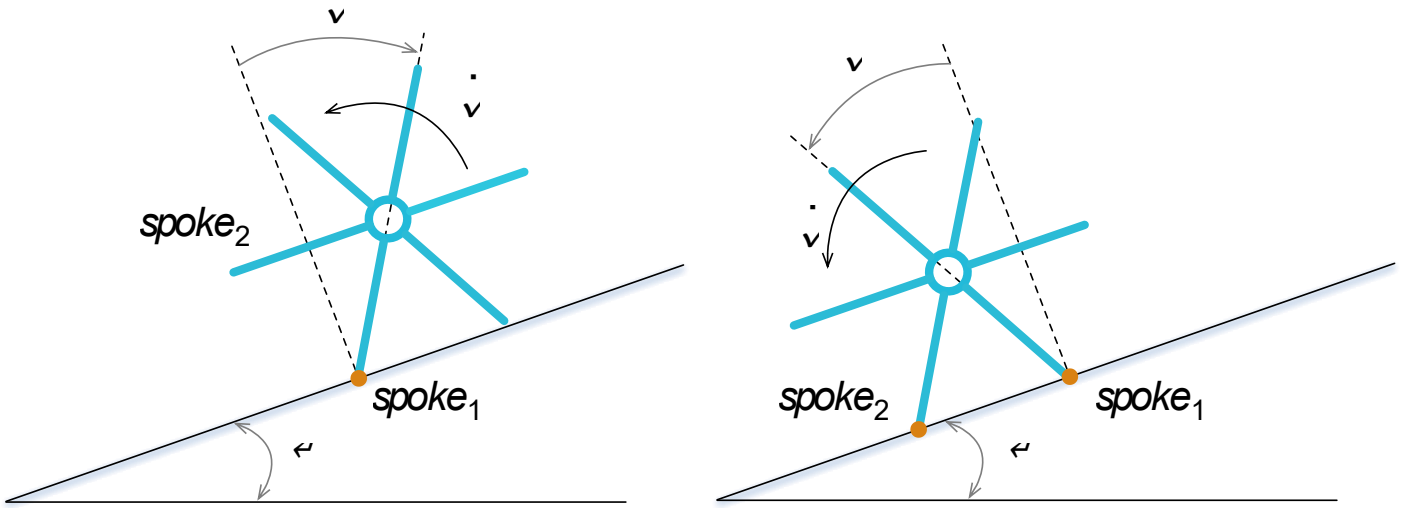


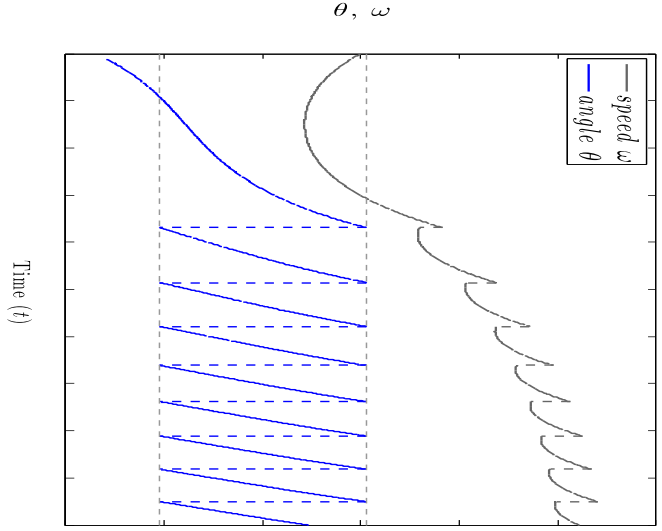



# Special kinds of executions

- **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

# Rimless wheel




**automaton** RimlessWheel($\alpha, \mu$: Real, $n$: Nat)
 const $\beta$: Real := $2\,\pi/n$
 **type** Spokes: enumeration [1,…,n]
 **actions**
  impact
 **variables**
  pivot: Spokes :=1
  $\theta$:Real := 0
  $\omega$: 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 $\mathcal{A}$ is ***reachable*** if ∃ an execution $\alpha$ with $\alpha.lstate = \boldsymbol{x}$

$Reach_{\mathcal{A}}(\Theta)$ all reachable state from $\Theta$

$Reach_{\mathcal{A}}(\Theta, T)$ states reachable within time $T$

$Reach_{\mathcal{A}}(\Theta, k)$ states reachable within $k$ transitions

$Reach_{\mathcal{A}}(\Theta, T, k)$ states reachable up to time $k$ transitions and time $T$

An invariant $I \subseteq val(X)$ is a set of states that contains $Reach_{\mathcal{A}}(\Theta)$

# How to prove invariants of hybrid automata

Recall the theorem we used for proving *I* is an invariant for automaton $\mathcal{A} = \langle X, \Theta, A, D \rangle$

**Theorem 7.1**. Given an automaton $\mathcal{A} = \langle X, \Theta, A, \boldsymbol{D} \rangle$, if a set of states $I \subseteq val(X)$ satisfies the following:

- (Start condition) For any starting state $\boldsymbol{x} \in \Theta, \boldsymbol{x} \in I$ and
- (Transition closure) For any action a $\in A$, if and $\boldsymbol{x} \rightarrow_{\boldsymbol{a}} \boldsymbol{x}'$ and $\boldsymbol{x} \in I$ then $\boldsymbol{x}' \in I$, and

Then $I$ is an inductive invariant of $\mathcal{A}$.

# How to prove invariants of hybrid automata

**Theorem 7.1**. Given an HIOA $\mathcal{A} = \langle X, \Theta, A, \boldsymbol{D}, \boldsymbol{T} \rangle$, if a set of states $I \subseteq val(X)$ satisfies the following:

- (Start condition) For any starting state $\boldsymbol{x} \in \Theta, \boldsymbol{x} \in I$ and
- (Transition closure) For any action a $\in A$, if and $\boldsymbol{x} \rightarrow_{\boldsymbol{a}} \boldsymbol{x}'$ and $\boldsymbol{x} \in I$ then $\boldsymbol{x}' \in I$, and
- (Trajectory closure) For any trajectory $\tau \in \boldsymbol{T}$ if $\tau.fstate \in I$ then $\tau.lstate \in I$

Then $I$ is an inductive invariant of $\mathcal{A}$.

# How to prove invariants of hybrid automata

**Theorem 7.1**. Given an HIOA $\mathcal{A} = \langle X, \Theta, A, \boldsymbol{D}, \boldsymbol{T} \rangle$, if a set of states $I \subseteq val(X)$ satisfies the following:

- (Start condition) For any starting state $\boldsymbol{x} \in \Theta, \boldsymbol{x} \in I$ and
- (Transition closure) For any action a $\in A$, if and $\boldsymbol{x} \rightarrow_{\boldsymbol{a}} \boldsymbol{x}'$ and $\boldsymbol{x} \in I$ then $\boldsymbol{x}' \in I$, and
- (Trajectory closure) For any trajectory $\tau \in \boldsymbol{T}$ if $\tau.fstate \in I$ then $\tau.lstate \in I$

Then $I$ is an inductive invariant of $\mathcal{A}$.

**Proof.** Consider any reachable state $x \in Reach_{\mathcal{A}}$. By the definition of a reachable state, there exists an execution $\alpha$ of $\mathcal{A}$ with $\alpha.lstate = x$. We proceed by induction on the length of the execution $\alpha$. For the base case, $\alpha$ consists of a single starting state $\boldsymbol{x} \in \Theta$, and, by the *start condition*, $\boldsymbol{x} \in I$. For the inductive step, we consider two subcases:

**Case 1:** $\alpha = \alpha' a\, p(\boldsymbol{x})$, where $a \in A$ and $p(\boldsymbol{x})$ is a point trajectory at $\boldsymbol{x}$ .

By the induction hypothesis, we know that $\alpha'.lstate \in I$.

By invoking the *transition closure*, we obtain $\boldsymbol{x} \in I$.

**Case 2:** $\alpha = \alpha'\tau$, where $\tau$ is a trajectory of $\mathcal{A}$ and $\tau.lstate = \boldsymbol{x}$

By the *induction hypothesis*, $\alpha'.lstate \in I$ and by

invoking the *trajectory closure*, we deduce that $\tau.lstate = \boldsymbol{x} \in I$

# An application

**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** $x \geq 0$

Candidate invariant: ``stays above ground''

$I_0 : x \geq 0 \equiv \{\boldsymbol{u} \in val(\{x, v\}) |\ \boldsymbol{u}\lceil x \geq 0\ \}$

Applying Theorem 7.1:

- Consider any initial state $\boldsymbol{u} \in \Theta; \boldsymbol{u}\lceil x = h \geq 0$
  - $\boldsymbol{u} \in I_0$
- Consider any transition $\boldsymbol{u} \rightarrow_{bounce} \boldsymbol{u'}$
  - From precondition we know $\boldsymbol{u}\lceil x = 0$; from effect we know $\boldsymbol{u'}.x = \boldsymbol{u}.x$ therefore $\boldsymbol{u'}\lceil x = 0 \geq 0$
  - $\boldsymbol{u'} \in I_0$
- Consider any trajectory $\tau \in T$
  - From mode invariant we know that for $\forall t \in \tau.dom\ , \tau(t)\lceil x \geq 0$
  - It follows that $\tau.lstate\lceil x \geq 0$
- What part of Bouncingball was used ? What could be changed?

# An application

**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** $x \geq 0$

Candidate invariant: ``stays above ground and below h''

$I_h: h \geq x \geq 0$

Applying Theorem 7.1:

- Consider any initial state $\boldsymbol{u} \in \Theta$; $\boldsymbol{u}.x = h$
  - $\boldsymbol{u} \in I_h$
- Consider any transition $\boldsymbol{u} \rightarrow_{bounce} \boldsymbol{u'}$
  - From precondition we know $\boldsymbol{u}.x = 0$; from effect we know $\boldsymbol{u'}.x = \boldsymbol{u}.x$ therefore $\boldsymbol{u'}.x = 0$
  - $\boldsymbol{u'} \in I_h$
- Consider any trajectory $\tau \in T$
  - From mode invariant and inductive hypothesis we know that for $\forall t \in \tau.dom, \tau(t).x \geq 0$ **and** , $\tau(0).x \in [0, h]$ and that $\tau$ is a solution of d(x) = v; d(v) = -g
  - Is this adequate to infer $\tau.lstate \in I_h$?

# Strengthened invariant

**automaton** Bouncingball(c,h,g)

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

k: Nat := 0

**actions:** bounce

**transitions:**

bounce

**pre** *x = 0 /\ v < 0*

**eff** v := -cv; k:= k + 1

**trajectories:**

Loc1

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

**invariant** $x \geq 0$

Candidate invariant: ``stays above ground and below h''

$I_v: v^2 - 2g(hc^{2k} - x) = 0$

Applying Theorem 7.1:

- Consider any initial state $\boldsymbol{u} \in \Theta; \boldsymbol{u}.x = h; \boldsymbol{u}.k = 0$
  - $\boldsymbol{u} \in I_v$
- **Exercise**: Finish the rest

# Summary so far

- Theorem 7.1 gives a sufficient condition for proving **inductive** invariants
- Not all invariants are inductive
- We often have to **strengthen** invariants to make them inductive
- Read examples in Chapter 7

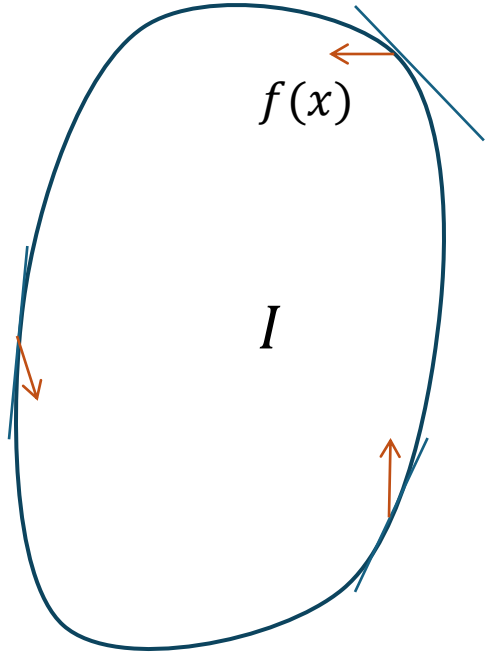
# **Sub-tangential conditions.** Checking trajectory conditions without solving ODEs [Bhatia and Szegö 67]

How to check trajectory closure: for any $\tau \in \boldsymbol{T}$ if $\tau.fstate \in I$ then $\tau.lstate \in I$ ?

Use a Lyapunov function for each mode. How? Exercise.

**Lemma.** Consider the ODE $\dot{x} = f(x)$ for state variable $x$, describing $\boldsymbol{T}$. Let $I$ be a compact set containing the initial set $\Theta$. Boundary of $I$ is defined by $\mathrm{P}(\boldsymbol{x}) = 0$**.** Then, $I$ is an inductive invariant of the above ODE if for each $x, on\ the\ boundary\ P(x) = 0$ , $f(x)$ is pointing inwards from the boundary. That is

$$\frac{\partial P(\boldsymbol{x})}{\partial \boldsymbol{x}}.f(x) \geq 0.$$




Related to the notions of Barrier Certificates and more general Control Barrier Functions.

# Proving invariance using subtangential condition

**Lemma.** Consider $\dot{x} = f(x)$ and let $P: \mathbb{R}^n \to \mathbb{R}$ be a continuous function such that $I = \{x | P(x) \leq 0\}$ is compact and contains. $\partial I := \{x \mid P(x) = 0\}$ is the boundary of $I$ and P is differentiable over $\partial I$. If

Sub-tangential condition $\frac{\partial P(x)}{\partial x} . f(x) \leq 0 \; \forall x \in \partial I$.

Then $I$ is an invariant

Proof. Fix $x(0) \in \Theta \subseteq I$. We must show $\forall t \geq 0, x(t) \in I$

Define $\psi(t) = P\big(x(t)\big)$

$\psi(0) = P\big(x(0)\big) \leq 0 \qquad [x(0) \in \Theta \subseteq I, P\big(x(0)\big) \leq 0]$

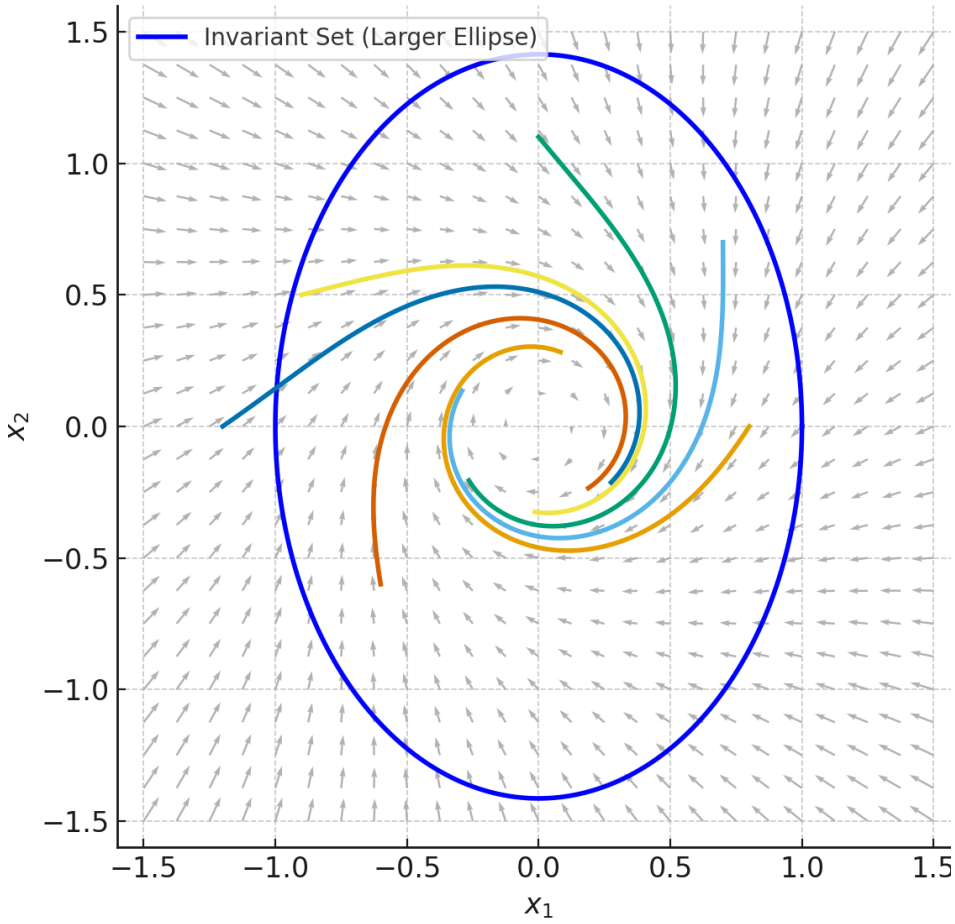
SFTC trajectory from $x(0)$ leaves $I$ at some point $t_1$

That is, $\psi(t_1) = P\big(x(t_1)\big) > 0$

By continuity of $\psi$, there exists earliest $t^* \in (0, t_1)$ such that $\psi(t) \leq 0$ f $t < t^*, \psi(t^*) = 0 \Rightarrow x(t^*) \in \partial I$ and $\psi(t) > 0$ for some $t > t^*$

$$\dot{\psi}(t^*) = \frac{d}{dt} P\big(x(t)\big)\Big|_{t=t^*} = \frac{\partial P(x(t^*))}{\partial t} . f(x(t^*)) \leq 0$$

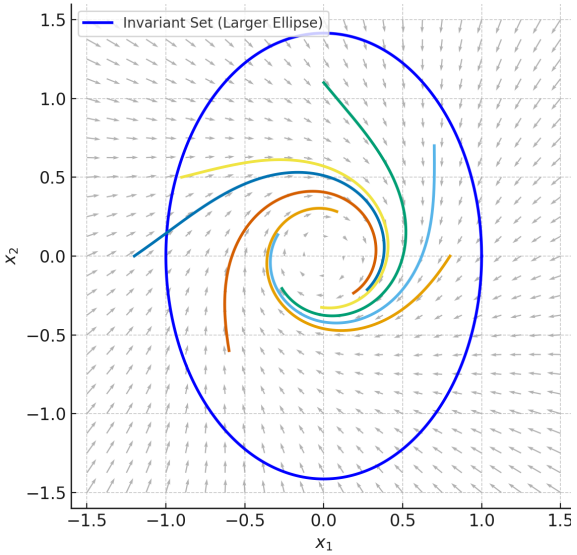
$\psi(t^*)$ is nonincreasing and it is not possible for $\psi(t) > 0$ for $t > t^*$.



**Lemma.** Consider $\dot{x} = f(x)$ and let $P: \mathbb{R}^n \to \mathbb{R}$ be a continuous function such that $I = \{x | P(x) \leq 0\}$ is compact and contains. $\partial I := \{x \mid P(x) = 0\}$ is the boundary of $I$ and P is differentiable over $\partial I$. If

Sub-tangential condition $\frac{\partial P(x)}{\partial x}.f(x) \leq 0 \; \forall x \in \partial I.$

Then $I$ is an invariant




## Example.

$\dot{x_1} = x_2 - x_1(x_1^2 + x_2^2) \qquad \dot{x}_2 = -x_1 - x_2(x_1^2 + x_2^2)$

$I = \{x \in \mathbb{R}^2 \,|P(x) \leq 0\}$ where $P(x) = 2x_1^2 + x_2^2 - 1$

$$\nabla P = \frac{\partial P}{\partial x} = \begin{bmatrix} 4x_1 \\ 2x_2 \end{bmatrix}$$

$$\nabla P.f(x) = \begin{bmatrix} 4x_1 \\ 2x_2 \end{bmatrix}.[x_2 - x_1(x_1^2 + x_2^2) \quad -x_1 - x_2(x_1^2 + x_2^2)]$$

$$= 4x_1(x_2 - x_1(x_1^2 + x_2^2)) + 2x_2(-x_1 - x_2(x_1^2 + x_2^2))$$

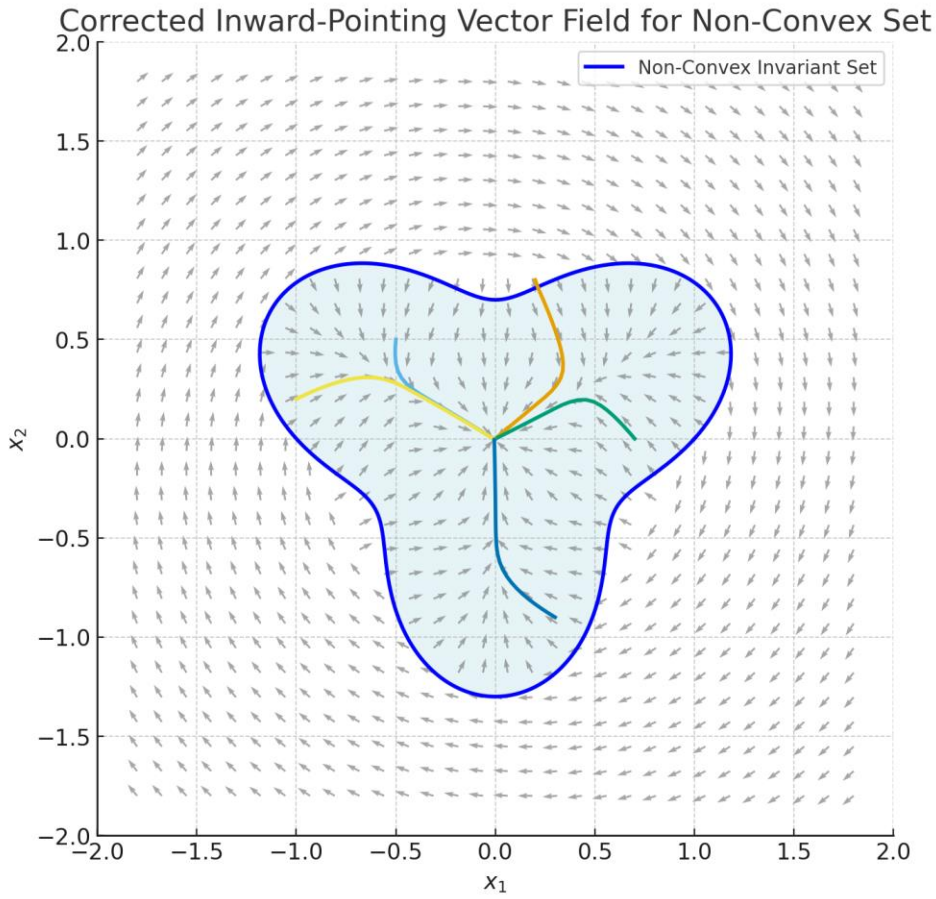
$$= 4x_1x_2 - 4x_1^2r^2 - 2x_2x_1 - 2x_2^2r^2$$

$$= 2x_1x_2 - 2r^2(2x_1^2 + x_2^2)$$

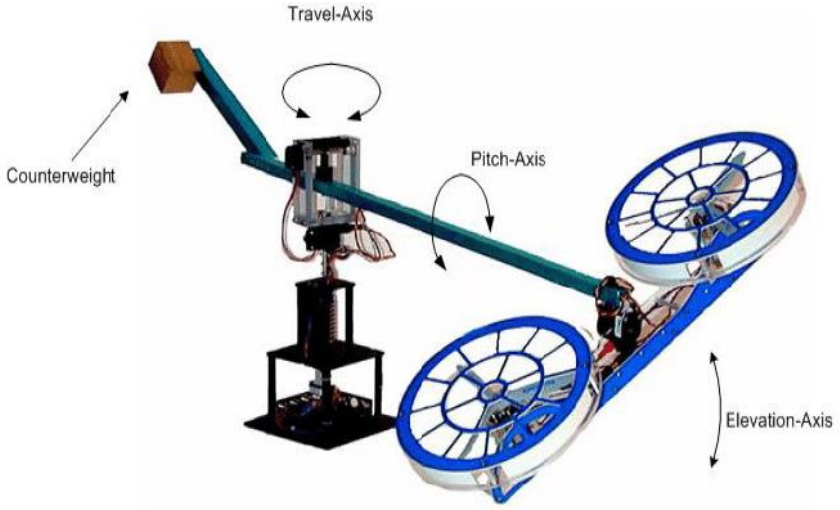
On the boundary $2x_1^2 + x_2^2 = 1, \nabla P.f(x) = 2x_1x_2 - 2r^2 = 2x_1x_2 - 2(x_1^2 + x_2^2)$

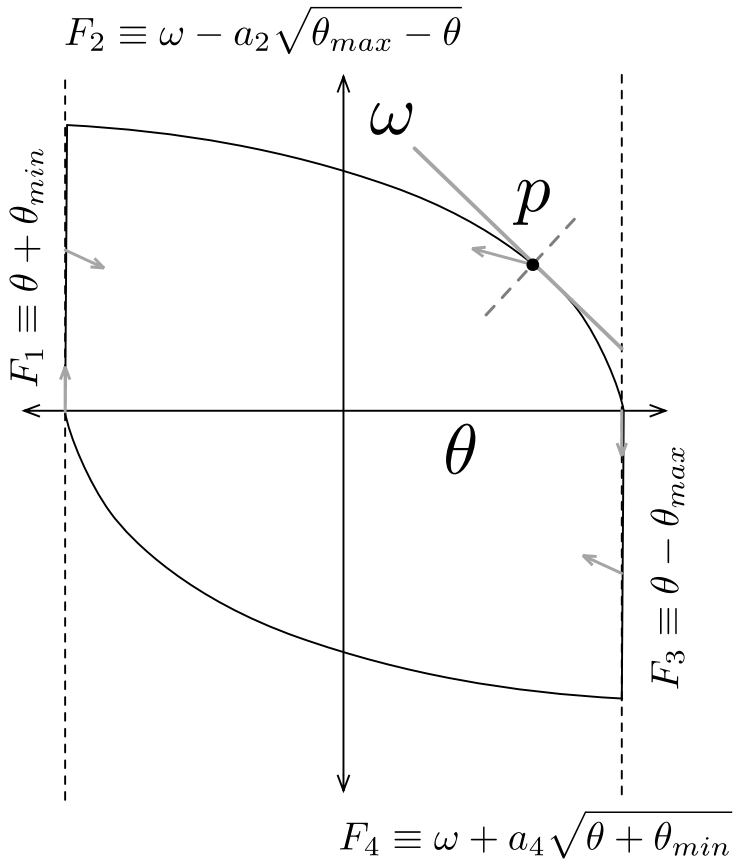
Completing the square: $\nabla P.f(x) = 2x_1x_2 - 2(x_1^2 + x_2^2) = -2\left(\frac{3}{4}x_1^2 + \left(x_2 - \frac{1}{2}x_1\right)^2\right) \leq 0$

# Another example



# Piece-wise sub-tangential conditions




If the boundary is defined in pieces, for example, one for each mode, then check the condition for each piece separately

Safety verification of model helicopter controller using hybrid input/output automata, Mitra, Wang, Lynch & Feron, HSCC 2003.

# Summary

- Definitions of reachability and invariance for hybrid automata generalize those definitions for (discrete) automata by introducing trajectories
- We can prove inductive invariants for hybrid automata by checking trajectory closure, in addition to transition closure
- Trajectory closure for trajectories defined by ODEs can be checked using the subtangential conditions which do not require solving ODEs