Reasoning about Stability

Mahesh Viswanathan

University of Illinois, Urbana-Champaign

MVD, October 2014

Reasoning about Stability

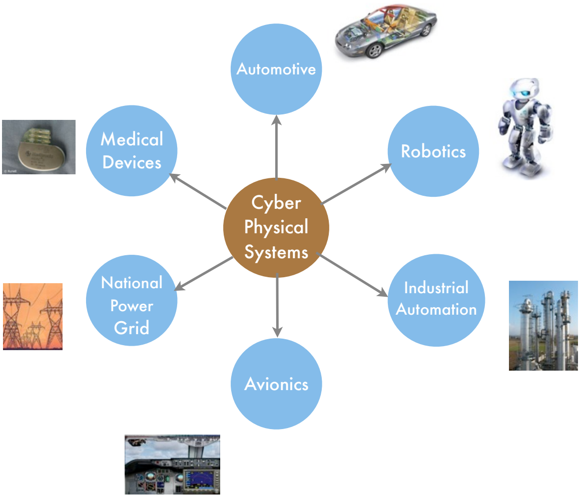
Mahesh Viswanathan

P. Prabhakar G.E. Dullerud N. Roohi

University of Illinois, Urbana-Champaign

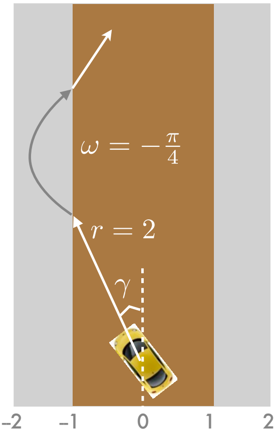
MVD, October 2014

Cyber Physical Systems

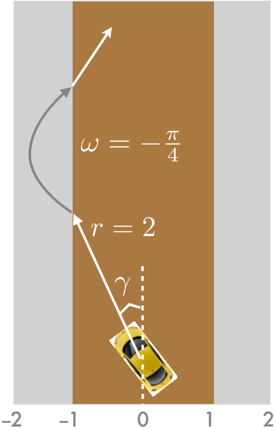


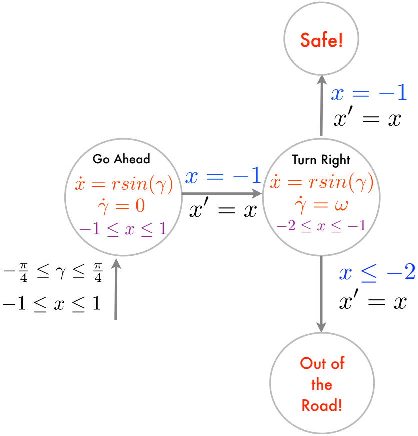
Computing devices controlling physical processes arise everywhere

Hybrid Automata

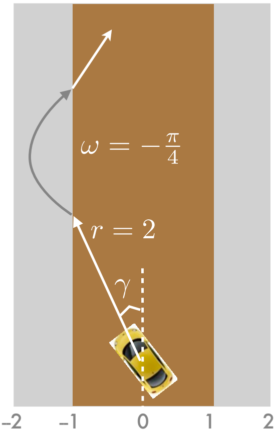


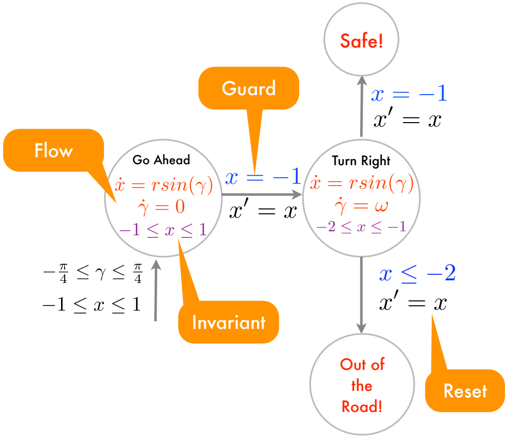
Hybrid Automata





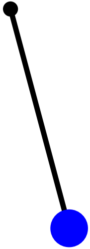
Hybrid Automata





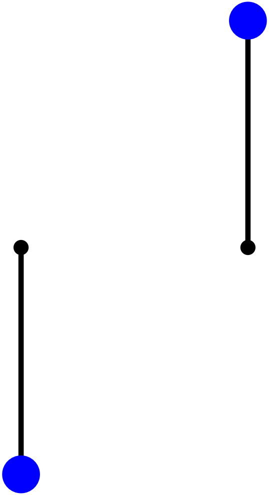
Equilibria in a Pendulum

Equilibria in a Pendulum

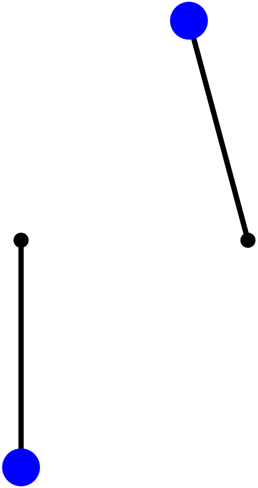


Equilibria in a Pendulum

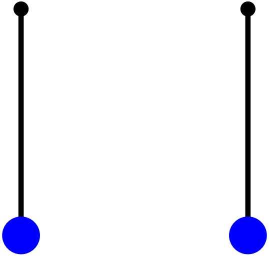
Equilibria in a Pendulum



Equilibria in a Pendulum

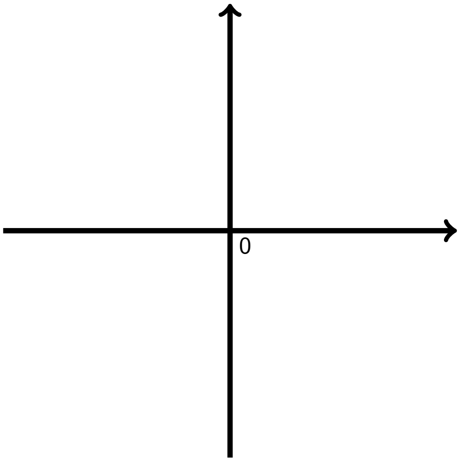


Equilibria in a Pendulum



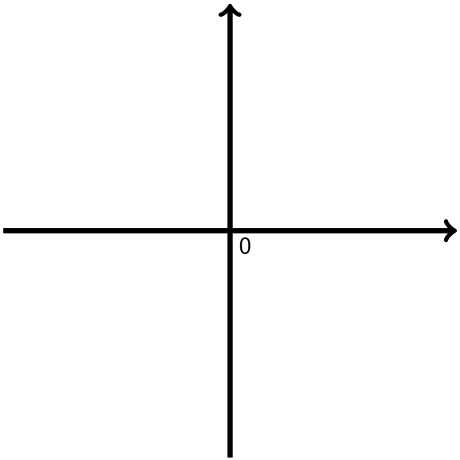
Lyapunov Stability [Aleksandr Lyapunov 1892]

- An equilibrium point is a state from which no executions leave



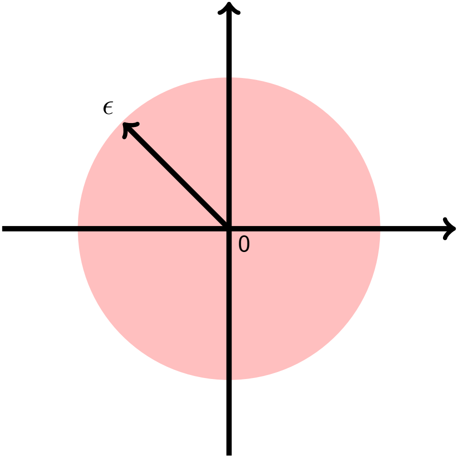
Lyapunov Stability [Aleksandr Lyapunov 1892]

- An equilibrium point is a state from which no executions leave
- Lyapunov Stability is defined as



Lyapunov Stability [Aleksandr Lyapunov 1892]

- An equilibrium point is a state from which no executions leave
- Lyapunov Stability is defined as

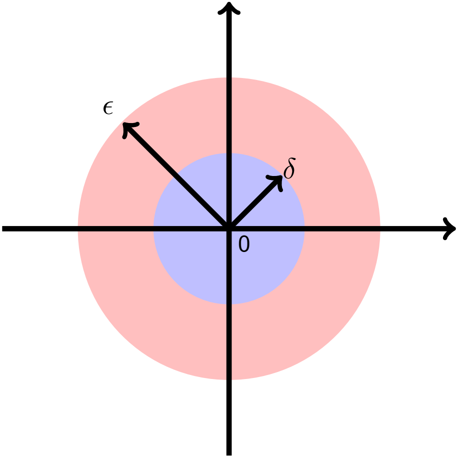


$$\forall \epsilon > 0$$

Lyapunov Stability [Aleksandr Lyapunov 1892]

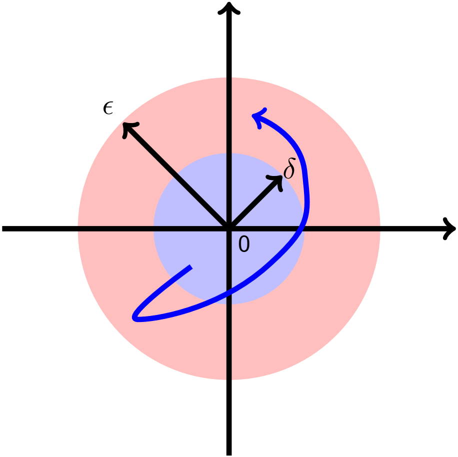
- An equilibrium point is a state from which no executions leave
- Lyapunov Stability is defined as

$$\forall \epsilon > 0 \exists \delta > 0.$$



Lyapunov Stability [Aleksandr Lyapunov 1892]

- An equilibrium point is a state from which no executions leave
- Lyapunov Stability is defined as



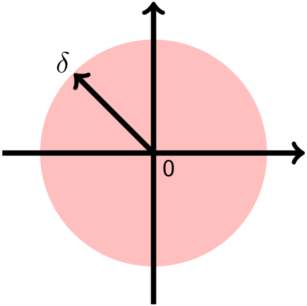
$$\forall \epsilon > 0 \exists \delta > 0.\ \forall \sigma.\ (\sigma(0) \in B_\delta(0)) \rightarrow (\forall t.\ \sigma(t) \in B_\epsilon(0))$$

Asymptotic Stability

Lyapunov Stability +
Convergence

Asymptotic Stability

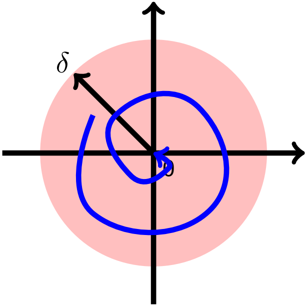
Lyapunov Stability $+$
Convergence



$\exists \delta > 0$

Asymptotic Stability

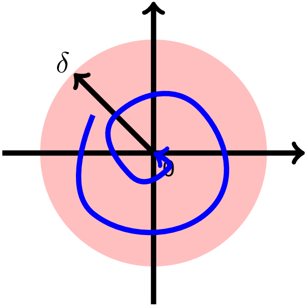
Lyapunov Stability +
Convergence



$$\exists \delta > 0 \forall \sigma. \ (\sigma \in B_\delta(0)) \rightarrow \mathrm{Conv}(\sigma)$$

Asymptotic Stability

Lyapunov Stability +
Convergence



$$\exists \delta > 0 \forall \sigma.\ (\sigma \in B_\delta(0)) \rightarrow \mathrm{Conv}(\sigma)$$
$$\mathrm{Conv}(\sigma) \equiv \forall \epsilon > 0. \exists T. \forall t \geq T.\ \sigma(t) \in B_\epsilon(0)$$

Talk Outline

- Why is the problem of checking stability different?

Talk Outline

- Why is the problem of checking stability different?
- How difficult is checking stability computationally?

Talk Outline

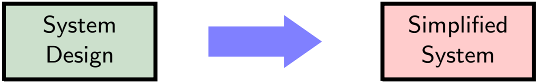
- Why is the problem of checking stability different?
- How difficult is checking stability computationally?
- What are proof principles to check stability?

Part I

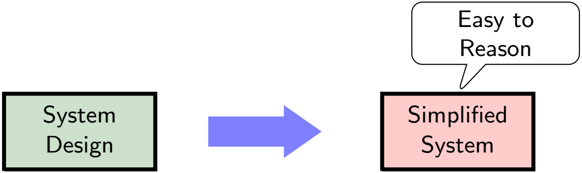
Why is stability different?

Prabhakar-Dullerud-Viswanathan

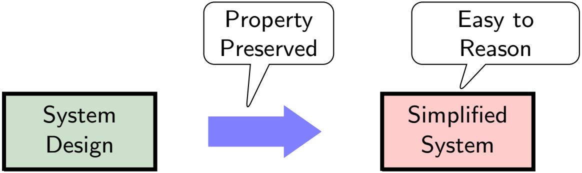
Process Preorders and Equivalences



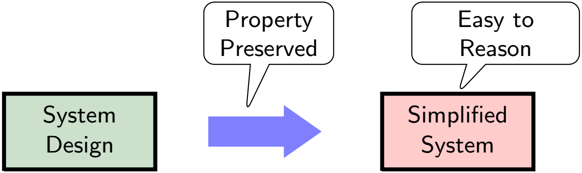
Process Preorders and Equivalences



Process Preorders and Equivalences



Process Preorders and Equivalences



Process preorders and equivalences characterize the relationship between complex and simple systems that preserve properties of interest

Transition Systems

$\mathcal{T} = (S, \Sigma, \longrightarrow)$, where S is a set of states, Σ is a set of actions, and $\longrightarrow \subseteq S \times \Sigma \times S$ is the transition relation.

Transition Systems

$\mathcal{T} = (S, \Sigma, \longrightarrow)$, where S is a set of states, Σ is a set of actions, and $\longrightarrow \subseteq S \times \Sigma \times S$ is the transition relation.

Bisimulation

$R \subseteq S_1 \times S_2$ is a bisimulation between $\mathcal{T}_1 = (S_1, \Sigma, \longrightarrow_1)$ and $\mathcal{T}_2 = (S_2, \Sigma, \longrightarrow_2)$ iff for every $(p, q) \in R$

Bisimulation

For Labeled Transition Systems

Transition Systems

$\mathcal{T} = (S, \Sigma, \longrightarrow)$, where S is a set of states, Σ is a set of actions, and $\longrightarrow \subseteq S \times \Sigma \times S$ is the transition relation.

Bisimulation

$R \subseteq S_1 \times S_2$ is a bisimulation between $\mathcal{T}_1 = (S_1, \Sigma, \longrightarrow_1)$ and $\mathcal{T}_2 = (S_2, \Sigma, \longrightarrow_2)$ iff for every $(p, q) \in R$

- If $p \xrightarrow{a}_1 p'$ then there is $q' \in S_2$ s.t. $q \xrightarrow{a}_2 q'$ and $(p', q') \in R$, and

Bisimulation

For Labeled Transition Systems

Transition Systems

$\mathcal{T} = (S, \Sigma, \longrightarrow)$, where S is a set of states, Σ is a set of actions, and $\longrightarrow \subseteq S \times \Sigma \times S$ is the transition relation.

Bisimulation

$R \subseteq S_1 \times S_2$ is a bisimulation between $\mathcal{T}_1 = (S_1, \Sigma, \longrightarrow_1)$ and $\mathcal{T}_2 = (S_2, \Sigma, \longrightarrow_2)$ iff for every $(p, q) \in R$

- If $p \xrightarrow{a}_1 p'$ then there is $q' \in S_2$ s.t. $q \xrightarrow{a}_2 q'$ and $(p', q') \in R$, and
- If $q \xrightarrow{a}_2 q'$ then there is $p' \in S_1$ s.t. $p \xrightarrow{a}_1 p'$ and $(p', q') \in R$.

Hybrid Transition Systems

$\mathcal{T} = (S, \Sigma, \longrightarrow, \Delta)$, where S is a set of states, Σ is a set of actions, $\longrightarrow \subseteq S \times \Sigma \times S$ is the transition relation, and $\Delta \subseteq \{\sigma \mid \sigma : [0, T] \to S\}$ is a set of trajectories.

Bisimulation

For Hybrid Transition Systems

Hybrid Transition Systems

$\mathcal{T} = (S, \Sigma, \longrightarrow, \Delta)$, where S is a set of states, Σ is a set of actions, $\longrightarrow \subseteq S \times \Sigma \times S$ is the transition relation, and $\Delta \subseteq \{\sigma \mid \sigma : [0, T] \to S\}$ is a set of trajectories.

Bisimulation

$R \subseteq S_1 \times S_2$ is a bisimulation between $\mathcal{T}_1 = (S_1, \Sigma, \longrightarrow_1, \Delta_1)$ and $\mathcal{T}_2 = (S_2, \Sigma, \longrightarrow_2, \Delta_2)$ iff R is a bisimulation w.r.t. the discrete transitions

Hybrid Transition Systems

$\mathcal{T} = (S, \Sigma, \longrightarrow, \Delta)$, where S is a set of states, Σ is a set of actions, $\longrightarrow \subseteq S \times \Sigma \times S$ is the transition relation, and $\Delta \subseteq \{\sigma \mid \sigma : [0, T] \to S\}$ is a set of trajectories.

Bisimulation

$R \subseteq S_1 \times S_2$ is a bisimulation between $\mathcal{T}_1 = (S_1, \Sigma, \longrightarrow_1, \Delta_1)$ and $\mathcal{T}_2 = (S_2, \Sigma, \longrightarrow_2, \Delta_2)$ iff R is a bisimulation w.r.t. the discrete transitions and for every $(p, q) \in R$

- If $\tau_1 \in \Delta_(p)$ then there is $\tau_2 \in \Delta_2(q)$ s.t. $\forall t$, $(\tau_1(t), \tau_2(t)) \in R$, and
- "Conversely"

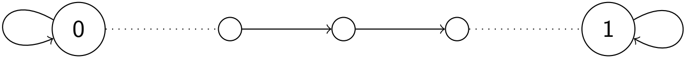
Features of Bisimulation

- The cannonical notion of equivalence between systems
- Preserves all categories of properties like safety, liveness, branching time, etc.
- Basis for minimization and decidability results

Is stability preserved by bisimulation?

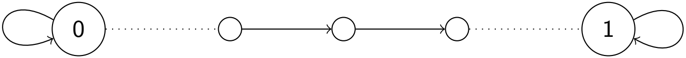
Is stability preserved by bisimulation?
No!

Stability not Bisimulation Invariant



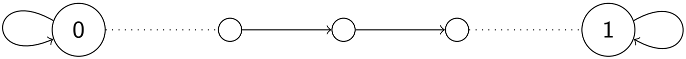
- Dynamics: $\sigma(x_0, t) = x_0^{\frac{1}{2t}}$

Stability not Bisimulation Invariant



- Dynamics: $\sigma(x_0, t) = x_0^{\frac{1}{2t}}$
- Bisimulation: $R = [0,1] \times [0,1]$ is a bisimulation

Stability not Bisimulation Invariant



- Dynamics: $\sigma(x_0, t) = x_0^{\frac{1}{2t}}$
- Bisimulation: $R = [0,1] \times [0,1]$ is a bisimulation
- 1 is Lyapunov/Asymptotically stable but 0 is not!

Need continuity!

Need continuity!

Definition

A uniformly continuous bisimulation between $\mathcal{T}_1$ and $\mathcal{T}_2$ is a binary relation R such that R is a bisimulation between $\mathcal{T}_1$ and $\mathcal{T}_2$ and R is a uniformly continuous relation

Need continuity!

Definition

A uniformly continuous bisimulation between $\mathcal{T}_1$ and $\mathcal{T}_2$ is a binary relation R such that R is a bisimulation between $\mathcal{T}_1$ and $\mathcal{T}_2$ and R is a uniformly continuous relation, i.e.,

$$\forall \epsilon > 0 \exists \delta > 0 \forall x \in \mathrm{dom}(R).\ R(B_\delta(x)) \subseteq B_\epsilon(R(x))$$

Invariance under Uniformly Continuous Bisimulations

Theorem

Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be hybrid transition systems with 0 as an equilibrium point. Suppose R is a uniformly continuous bisimulation such that $(0,0) \in R$ then

$\mathcal{T}_1$ is Lyapunov stable w.r.t. 0 iff $\mathcal{T}_2$ is Lyapunov stable w.r.t. 0

Invariance under Uniformly Continuous Bisimulations

Theorem

Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be hybrid transition systems with 0 as an equilibrium point. Suppose R is a uniformly continuous bisimulation such that $(0,0) \in R$ then

> *$\mathcal{T}_1$ is Lyapunov (Asymptotically) stable w.r.t. 0 iff $\mathcal{T}_2$ is Lyapunov (Asymptotically) stable w.r.t. 0*

Invariance under Uniformly Continuous Bisimulations

Theorem

Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be hybrid transition systems with 0 as an equilibrium point. Suppose R is a uniformly continuous bisimulation such that $(0, 0) \in R$ then

> *$\mathcal{T}_1$ is Lyapunov (Asymptotically) stable w.r.t. 0 iff $\mathcal{T}_2$ is Lyapunov (Asymptotically) stable w.r.t. 0*

- Above observation generalizes to stronger notions of stability

Invariance under Uniformly Continuous Bisimulations

Theorem

Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be hybrid transition systems with 0 as an equilibrium point. Suppose R is a uniformly continuous bisimulation such that $(0,0) \in R$ then

> *$\mathcal{T}_1$ is Lyapunov (Asymptotically) stable w.r.t. 0 iff $\mathcal{T}_2$ is Lyapunov (Asymptotically) stable w.r.t. 0*

- Above observation generalizes to stronger notions of stability
- Uniformly continuous simulations reflect stability notions

Lyapunov's Second Method

System $\dot{x} = F(x)$ with solution $\varphi(x, t)$

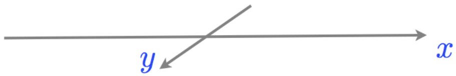
Equilibrium $F(0) = 0$

Lyapunov's Second Method

System $\dot{x} = F(x)$ with solution $\varphi(x, t)$

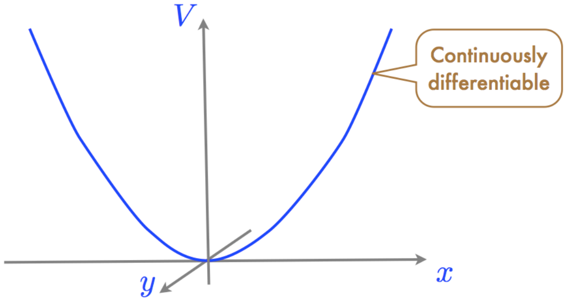
Equilibrium $F(0) = 0$

If there exists a "Lyapunov function" for the system then it is Lyapunov stable.


y
x

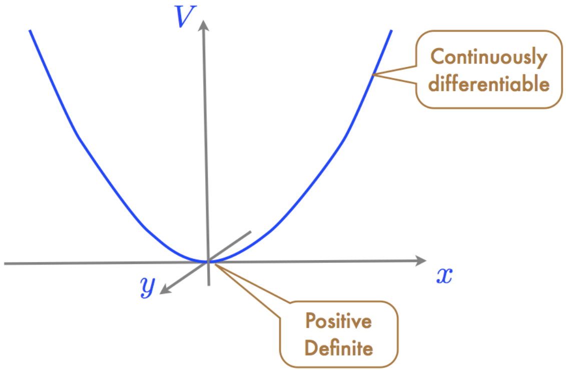
Lyapunov Function

An Illustration



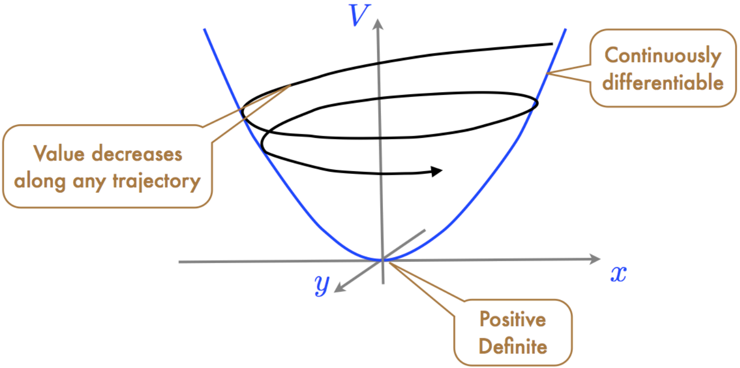
Lyapunov Function

An Illustration



Lyapunov Function

An Illustration



Lyapunov's Method as an Abstraction

$$\boxed{\begin{array}{c}\dot{x} = F(x) \\ \varphi\end{array}}$$

Lyapunov's Method as an Abstraction

$$\boxed{\begin{array}{c}\dot{x} = F(x) \\ \varphi\end{array}}$$

Exists $V : \mathbb{R}^n \to \mathbb{R}_+$ s.t.
V is positive definite C^1
$\dot{V} \leq 0$

Lyapunov's Method as an Abstraction

$$\dot{x} = F(x)$$
$$\varphi$$

$$V(\varphi)$$

Exists $V : \mathbb{R}^n \to \mathbb{R}_+$ s.t.
V is positive definite C^1
$\dot{V} \leq 0$

$v_1 \xrightarrow{t} v_2$ iff exist x_1, x_2 s.t. $x_1 \xrightarrow{t} x_2$, $V(x_1) = v_1$ and $V(x_2) = v_2$.

Lyapunov's Method as an Abstraction

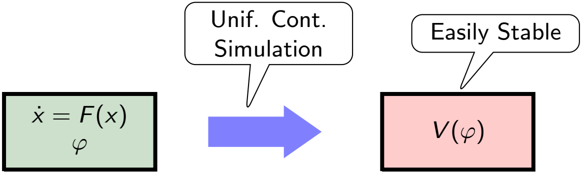
Easily Stable

$$\boxed{\begin{array}{c}\dot{x} = F(x) \\ \varphi\end{array}} \qquad \boxed{V(\varphi)}$$

Exists $V : \mathbb{R}^n \to \mathbb{R}_+$ s.t.
V is positive definite C^1
$\dot{V} \leq 0$

$v_1 \xrightarrow{t} v_2$ iff exist x_1, x_2 s.t. $x_1 \xrightarrow{t} x_2$, $V(x_1) = v_1$ and $V(x_2) = v_2$.

Lyapunov's Method as an Abstraction



Exists $V : \mathbb{R}^n \to \mathbb{R}_+$ s.t.
V is positive definite C^1
$\dot{V} \leq 0$

$v_1 \xrightarrow{t} v_2$ iff exist x_1, x_2 s.t. $x_1 \xrightarrow{t} x_2$, $V(x_1) = v_1$ and $V(x_2) = v_2$.

Other Characterizations

- Other extensions of Lyapunov's method to switched systems can also be understood in the abstraction setting
- Hartman-Grobman Theorem contructs a uniformly continuous bisimilar linearization

Expressing Stability

- Stabiity cannot be expressed in the classical modal/temporal logics like Hennessy-Milner, LTL, CTL, μ-calculus, etc.
  - Logic equivalence for these logics coincides with bisimulation

Expressing Stability

- Stabiity cannot be expressed in the classical modal/temporal logics like Hennessy-Milner, LTL, CTL, μ-calculus, etc.
  - Logic equivalence for these logics coincides with bisimulation
- First order logic over appropriate topologial structures is too strong

Expressing Stability

- Stabiity cannot be expressed in the classical modal/temporal logics like Hennessy-Milner, LTL, CTL, μ-calculus, etc.
  - Logic equivalence for these logics coincides with bisimulation
- First order logic over appropriate topologial structures is too strong
- Is there a (modal) logic that can express stability for which logic equivalence coincides with "continuous" bisimulations?

S_4: A Modal Logic for Space

Orlov[1928], Lewis[1932], Gödel[1933], Stone[1937], Tarski[1937]

$$\varphi ::= p \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \mathbf{I}\varphi \mid \mathbf{C}\varphi$$

S_4: A Modal Logic for Space

Orlov[1928], Lewis[1932], Gödel[1933], Stone[1937], Tarski[1937]

$$\varphi ::= p \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \mathbf{I}\varphi \mid \mathbf{C}\varphi$$

- Formulas interpreted as sets of points in a topological space

S_4: A Modal Logic for Space

Orlov[1928], Lewis[1932], Gödel[1933], Stone[1937], Tarski[1937]

$$\varphi ::= p \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \mathbf{I}\varphi \mid \mathbf{C}\varphi$$

- Formulas interpreted as sets of points in a topological space
- $[\![\mathbf{I}\varphi]\!]$: Interior of set defined by φ

S_4: A Modal Logic for Space

Orlov[1928], Lewis[1932], Gödel[1933], Stone[1937], Tarski[1937]

$$\varphi ::= p \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \mathbf{I}\varphi \mid \mathbf{C}\varphi$$

- Formulas interpreted as sets of points in a topological space
- $[\![\mathbf{I}\varphi]\!]$: Interior of set defined by φ
- $[\![\mathbf{C}\varphi]\!]$: Closure of set defined by φ

Bimodal Spatio-Temporal Logic

- Add to S_4 the usual temporal modalities of $\Box$ and $\Diamond$

Bimodal Spatio-Temporal Logic

- Add to S_4 the usual temporal modalities of $\Box$ and $\Diamond$
- Can stability be expressed in the resulting logic?

Bimodal Spatio-Temporal Logic

- Add to S_4 the usual temporal modalities of $\Box$ and $\Diamond$
- Can stability be expressed in the resulting logic? No!

Bimodal Spatio-Temporal Logic

- Add to S_4 the usual temporal modalities of $\Box$ and $\Diamond$
- Can stability be expressed in the resulting logic? No!
  - [Aiello-van Bentham, Davoren] Logic equivalence in this bi-modal logic coincides with bisimilarity under relations with "weak continuity" properties

Bimodal Spatio-Temporal Logic

- Add to S_4 the usual temporal modalities of $\Box$ and $\Diamond$
- Can stability be expressed in the resulting logic? No!
  - [Aiello-van Bentham, Davoren] Logic equivalence in this bi-modal logic coincides with bisimilarity under relations with "weak continuity" properties
- Open Question: What is the right logic?

Part II

How difficult is checking stability computationally?

Prabhakar-Viswanathan

What is known about checking Stability?

What is known about checking Stability?

Very little!

What is known about checking Stability?

Very little!

- Traditional control theoretic methods focus on identifying sufficient conditions that guarantee stability

What is known about checking Stability?

Very little!

- Traditional control theoretic methods focus on identifying sufficient conditions that guarantee stability
  - Some complexity results on how difficult it is to find these sufficient conditions

What is known about checking Stability?

Very little!

- Traditional control theoretic methods focus on identifying sufficient conditions that guarantee stability
  - Some complexity results on how difficult it is to find these sufficient conditions
- [Blondel-Tsitsiklis et. al.] Prove computational lower bounds (undecidability/NP-hardness) on checking stability of special discrete time linear switched systems

Deciding Hybrid Models

Traditional techniques to establish decidablity for various properties (like safety, liveness, etc.) of special hybrid models fail for stability

Establishing stability needs new ideas

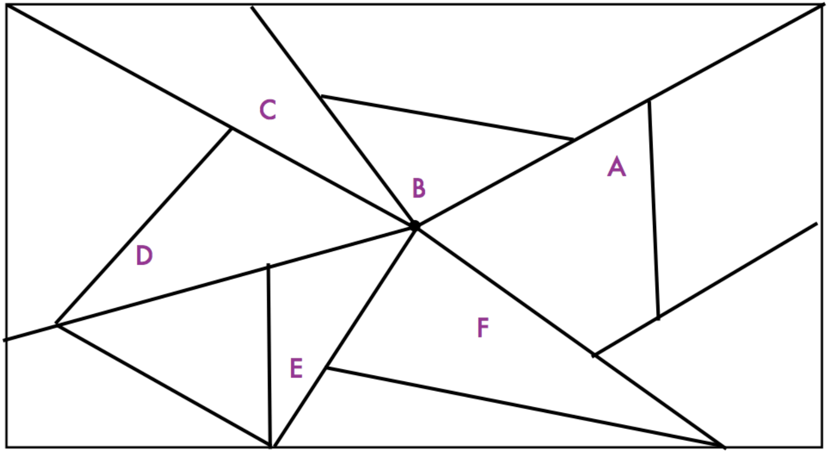
Deciding Hybrid Models

Traditional techniques to establish decidablity for various properties (like safety, liveness, etc.) of special hybrid models fail for stability

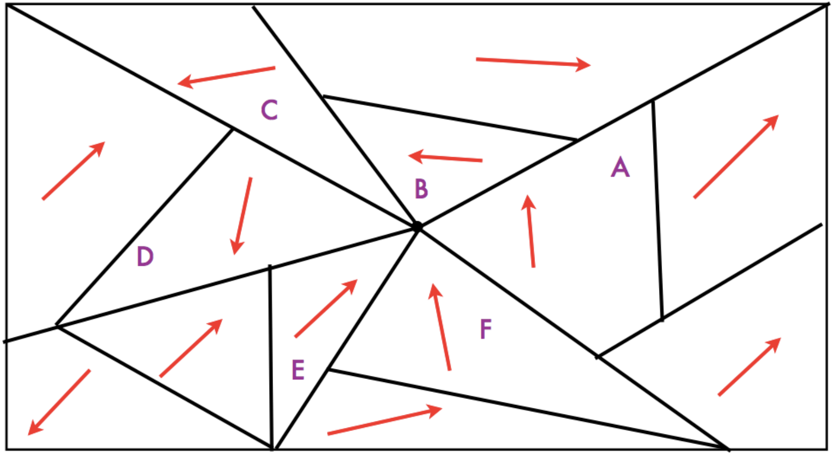
- Decidability results rely on establishing the existence of an effectively constructable finite "bisimulation" quotient

Establishing stability needs new ideas

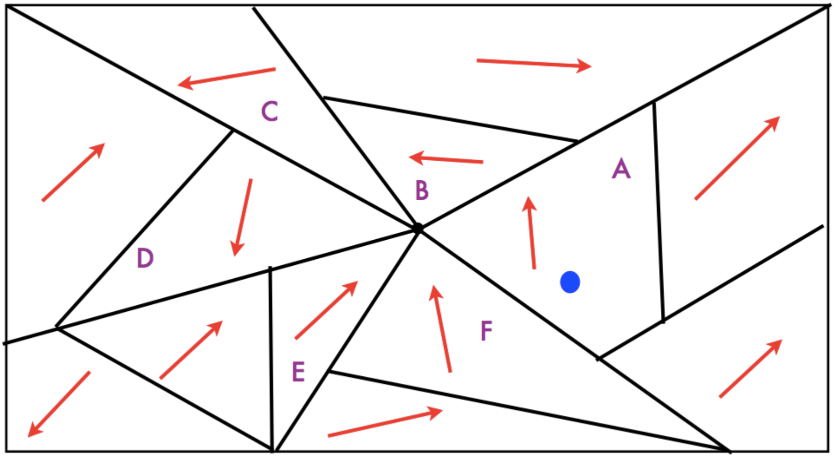
Piecewise Constant Derivative (PCD) Systems



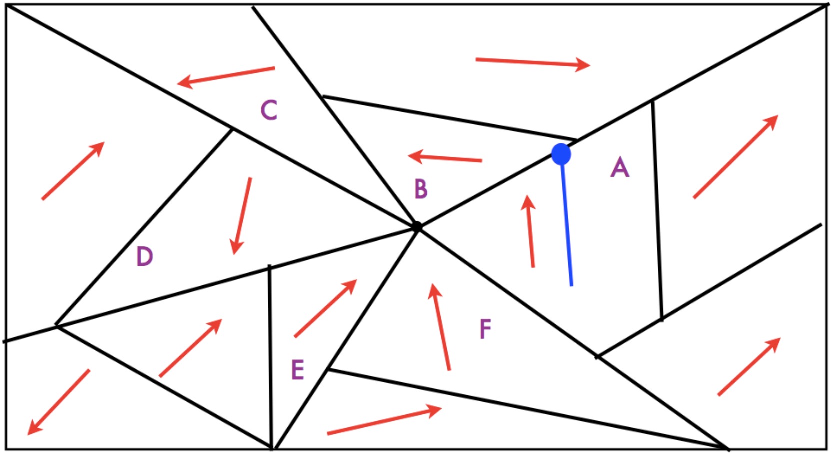
Piecewise Constant Derivative (PCD) Systems



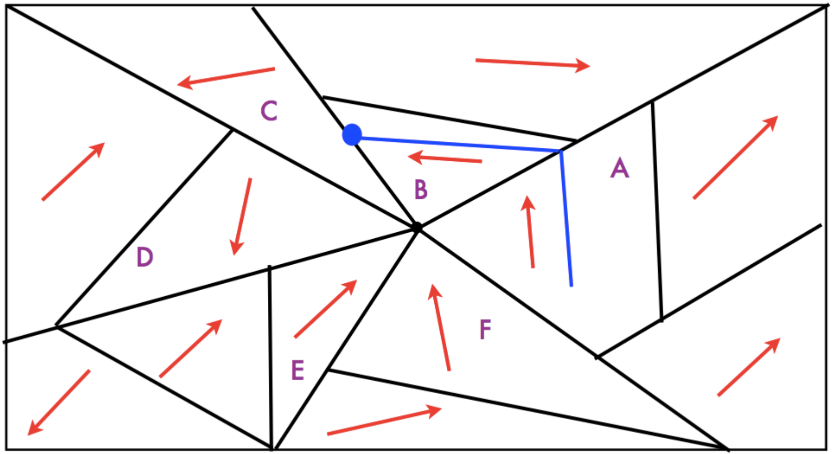
Piecewise Constant Derivative (PCD) Systems



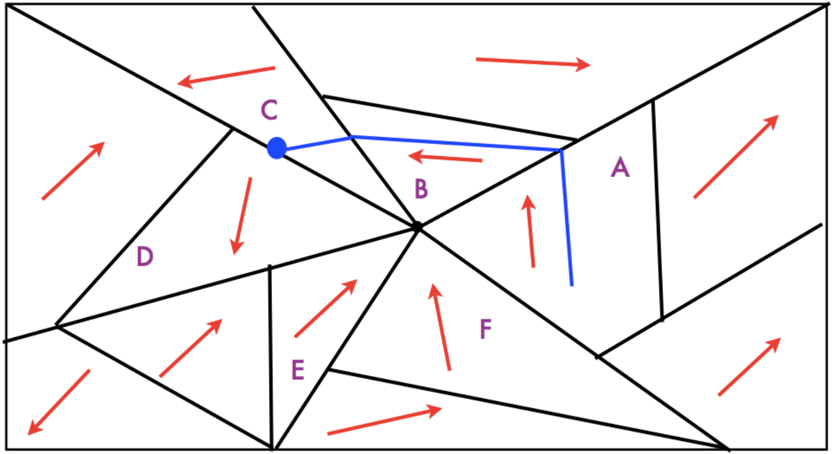
Piecewise Constant Derivative (PCD) Systems



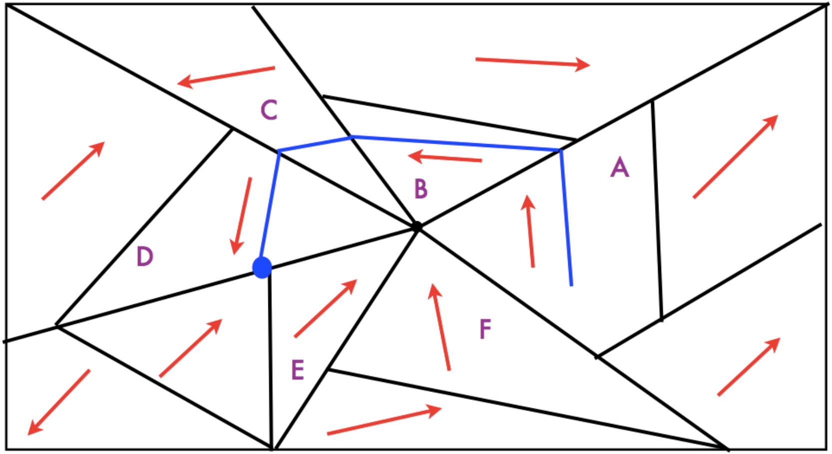
Piecewise Constant Derivative (PCD) Systems



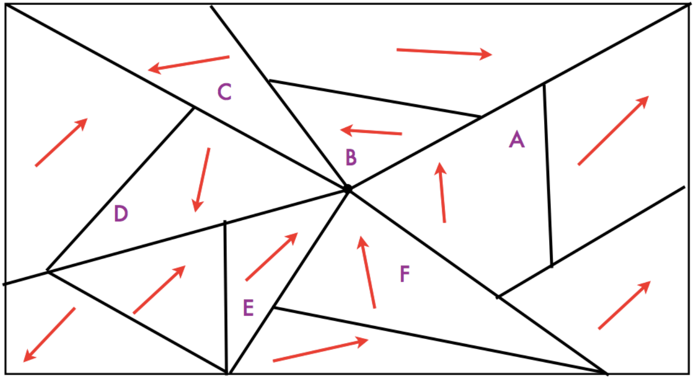
Piecewise Constant Derivative (PCD) Systems



Piecewise Constant Derivative (PCD) Systems

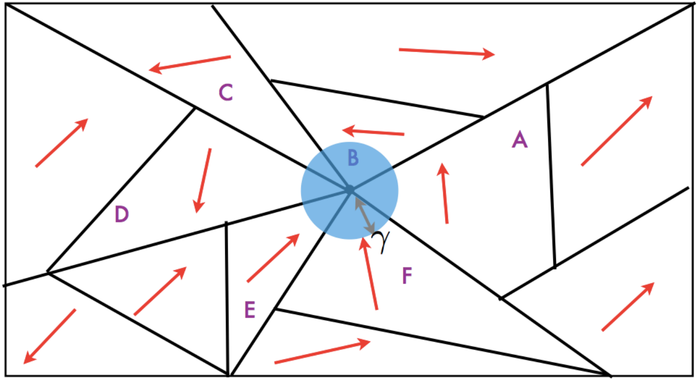


Stability in PCD



$$\forall \epsilon > 0\ \exists \delta > 0\ \forall \sigma.\ (\sigma(0) \in B_\delta(0)) \to (\forall t.\ \sigma(t) \in B_\epsilon(0))$$

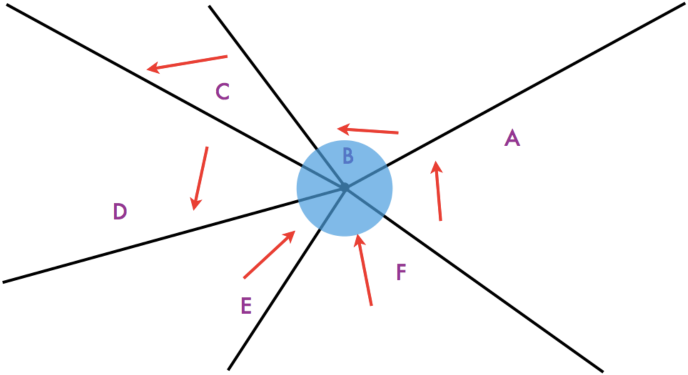
Stability in PCD



$\exists\gamma > 0.\ \forall\epsilon \in (0,\gamma]$

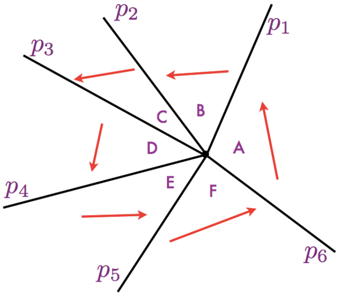
$$\cancel{\forall\epsilon > 0}\ \exists\delta > 0\ \forall\sigma.\ (\sigma(0) \in B_\delta(0)) \rightarrow (\forall t.\ \sigma(t) \in B_\epsilon(0))$$

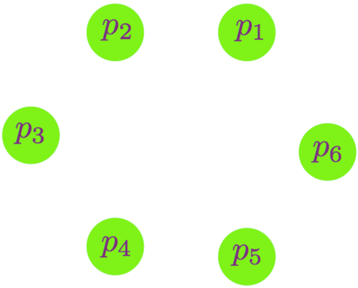
Special Structure Near the Origin



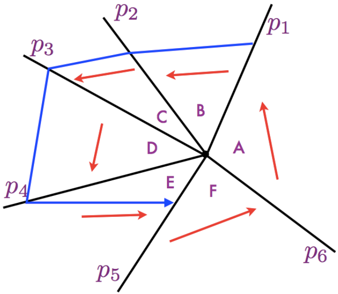
The planar partition looks like "wedges".

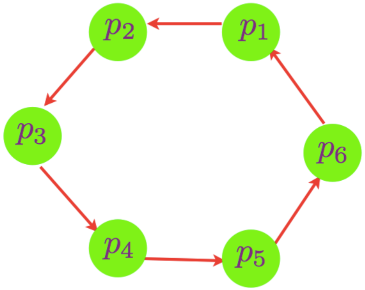
Executions near 0



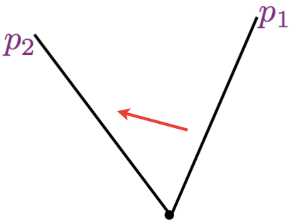


Executions near 0

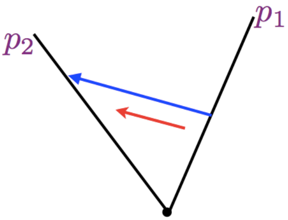




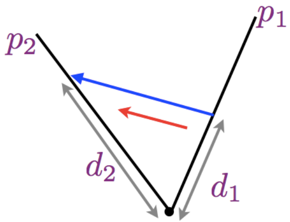
Capturing distance from origin



Capturing distance from origin

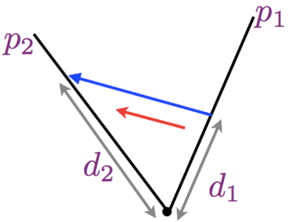


Capturing distance from origin

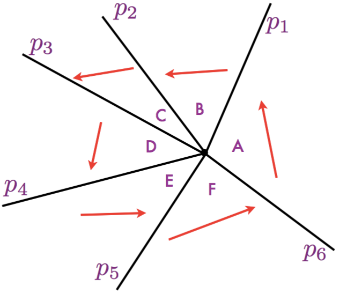


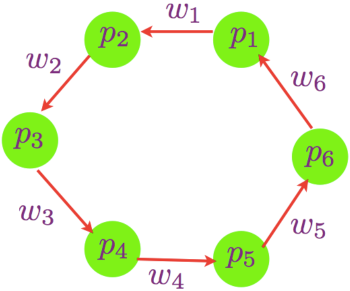
Capturing distance from origin

$$w(e) = \frac{|d_1|}{|d_2|}$$

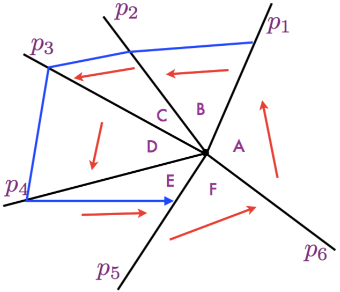


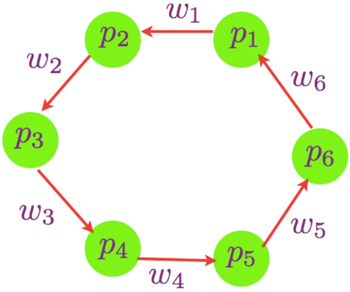
Stability Analysis
Weighted Graph





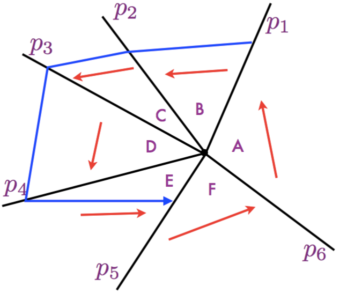
Stability Analysis
Weighted Graph

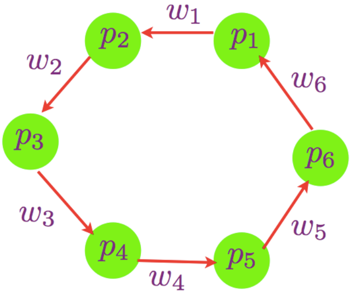




$$\sigma = \sigma(p_1, p_2)\sigma(p_2, p_3)\cdots\sigma(p_4, p_5)$$

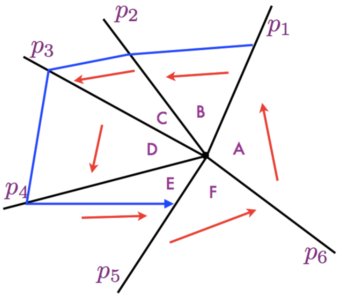
Stability Analysis
Weighted Graph

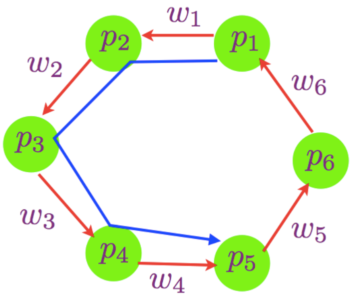




$$\sigma = \sigma(p_1, p_2)\sigma(p_2, p_3) \cdots \sigma(p_4, p_5)$$
$$w(\sigma) = \frac{|d(\sigma(T))|}{|d(\sigma(0))|}$$

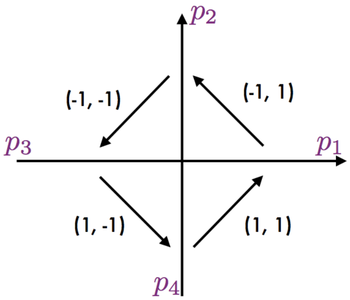
Stability Analysis
Weighted Graph

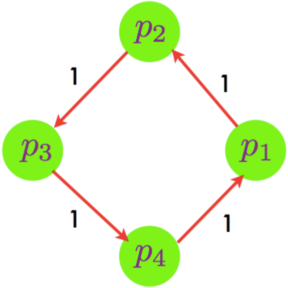




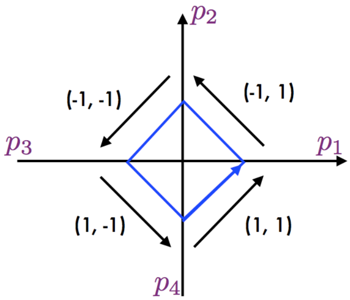
$$\sigma = \sigma(p_1, p_2)\sigma(p_2, p_3)\cdots\sigma(p_4, p_5)$$
$$w(\sigma) = \frac{|d(\sigma(T))|}{|d(\sigma(0))|} = w(p_1, p_2)w(p_2, p_3)\cdots w(p_4, p_5)$$

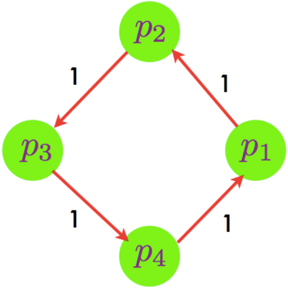
Example 1



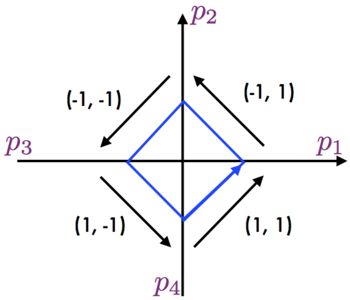


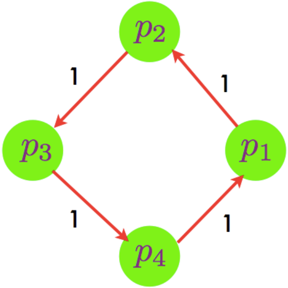
Example 1





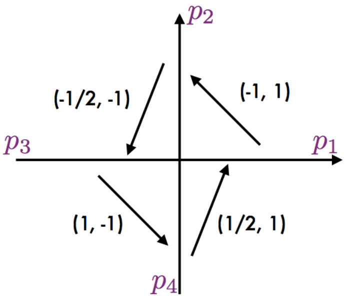
Example 1

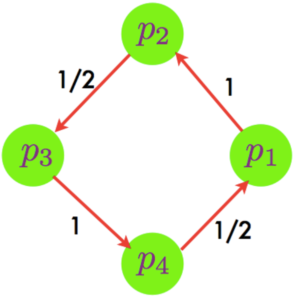




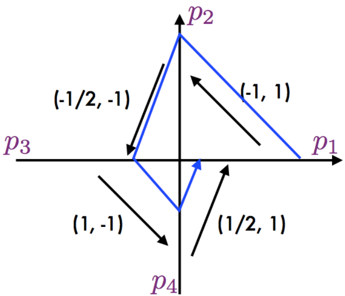
Lyapunov stable but not asymptotically stable.

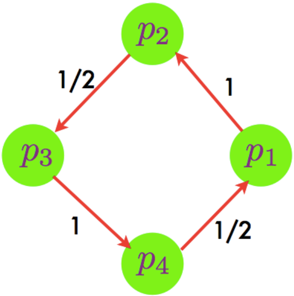
Example 2



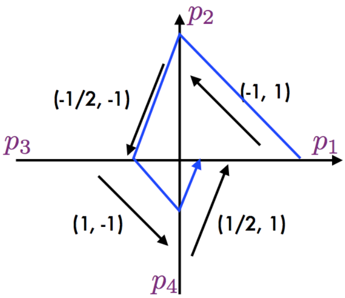


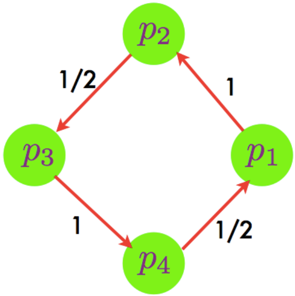
Example 2





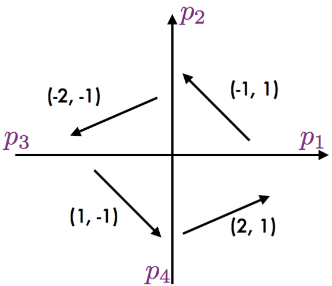
Example 2

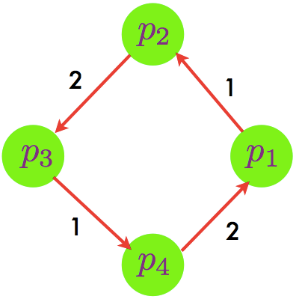




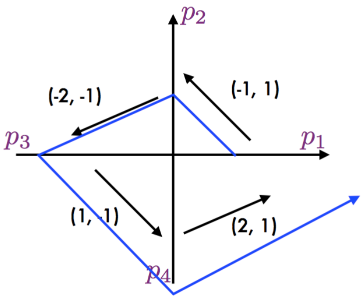
Lyapunov and asymptotically stable.

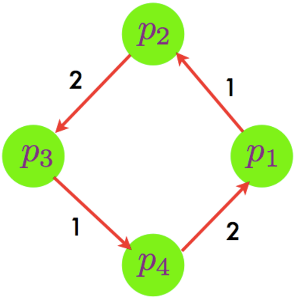
Example 3



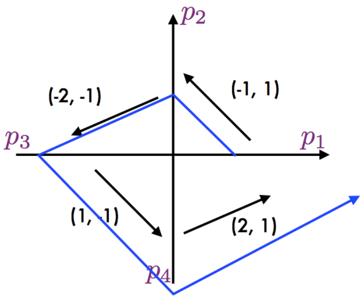


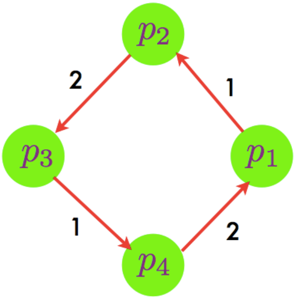
Example 3





Example 3





Neither Lyapunov nor asymptotically stable.

Stability of PCD

Theorem

A piecewise constant derivative system is Lyapunov stable iff the weighted graph does not have any cycles of weight > 1 *.*

Stability of PCD

Theorem

A piecewise constant derivative system is Lyapunov (asymptotically) stable iff the weighted graph does not have any cycles of weight > 1 (≥ 1).

Decidability Results

- Observations can be extend to show that the stability problem is decidable for planar rectangular switched systems with polyhedral invariants and guards

Decidability Results

- Observations can be extend to show that the stability problem is decidable for planar rectangular switched systems with polyhedral invariants and guards
- Stability problem is undecidable for PCD in 5 dimensions.

Stability, Safety, and Abstraction

Is stability computationally harder than checking reachability?

Is stability computationally harder than checking reachability?

- Undecidability proof obtained by reducing reachability to stability

Is stability computationally harder than checking reachability?

- Undecidability proof obtained by reducing reachability to stability
- However, reachability is not known to be decidable for planar rectangular switched systems with polyhedral guards and resets

Stability, Safety, and Abstraction

Is stability computationally harder than checking reachability?

- Undecidability proof obtained by reducing reachability to stability
- However, reachability is not known to be decidable for planar rectangular switched systems with polyhedral guards and resets; stability algorithm exploits special structure near origin

Stability, Safety, and Abstraction

Is stability computationally harder than checking reachability?

- Undecidability proof obtained by reducing reachability to stability
- However, reachability is not known to be decidable for planar rectangular switched systems with polyhedral guards and resets; stability algorithm exploits special structure near origin
- Stability algorithm can be thought of as constructing a "quantitative predicate abstraction".

Stability, Safety, and Abstraction

Is stability computationally harder than checking reachability?

- Undecidability proof obtained by reducing reachability to stability
- However, reachability is not known to be decidable for planar rectangular switched systems with polyhedral guards and resets; stability algorithm exploits special structure near origin
- Stability algorithm can be thought of as constructing a "quantitative predicate abstraction".Can be exploited to do abstraction-based checking of stabiity of more general hybrid systems.

Proof Rules for reasoning about Stability
[Roohi-Dullerud-Viswanathan]

- Traditional proof principles for stability are too conservative to reason about systems in the presence of "fair" controllers

Proof Rules for reasoning about Stability

[Roohi-Dullerud-Viswanathan]

- Traditional proof principles for stability are too conservative to reason about systems in the presence of "fair" controllers
- Compositional reasoning principles

Conclusions

Stability requires new analysis techniques and a number of questions remain open

Conclusions

Stability requires new analysis techniques and a number of questions remain open

- Concrete abstraction based methods to verify stability
- Logic to specify stability
- Computational complexity of checking stability
- Relationship between Safety and Stability
- Compositional reasoning principles