Formal synthesis through set computation

With an application to adaptive cruise control

Petter Nilsson
pettni@umich.edu

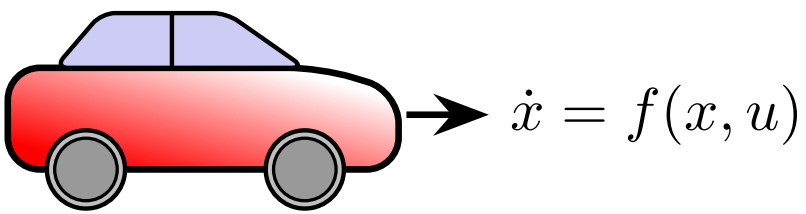
EECS Control Seminar, University of Michigan

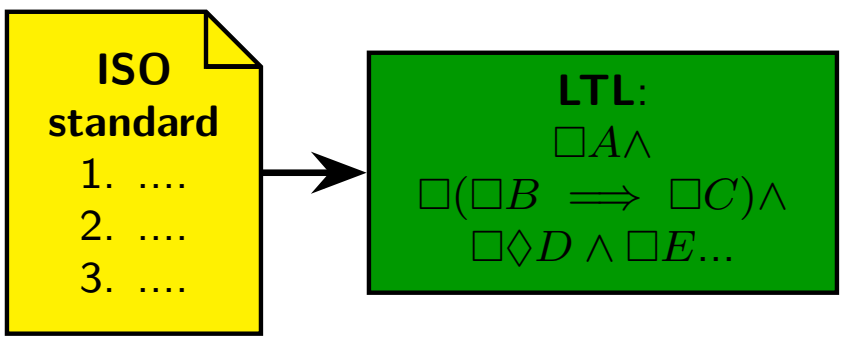
December 11, 2015

Correct-by-construction: How to pose the problem?

- Engineering: system and textual specifications
- Math: system model (e.g., controlled ODE's) and formal specification (e.g. in LTL)

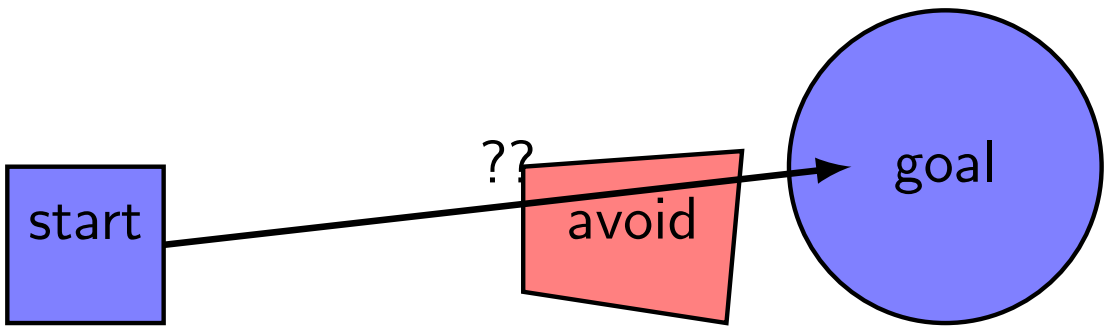
We know that modeling can be difficult, and the same is true for specification "translation"





Correct-by-construction: How to solve the problem?

- Need guarantees on transient behavior (far from steady state)
- Reachability problem is undecidable in general
- Typically have to settle for conservative results



Outline

Set-valued reachability operator

For a system $\Xi: \ \dot{x} = f(x, u)$, the **flow operator** $\phi_\Xi(t, y, u)$ satisfies

$$\phi_\Xi(0, y, u) = y, \qquad \frac{\mathrm{d}}{\mathrm{d}t}\phi_\Xi(t, y, u) = f\left(\phi(t, y, u), u\right)$$

Set-valued reachability operator

For a system $\Xi: \ \dot{x} = f(x,u)$, the **flow operator** $\phi_\Xi(t,y,u)$ satisfies

$$\phi_\Xi(0,y,u) = y, \qquad \frac{\mathrm{d}}{\mathrm{d}t}\phi_\Xi(t,y,u) = f\left(\phi(t,y,u),u\right)$$

For sets X, S, define a set-valued operator for safe reachability:

$$\mathrm{Rch}_S^{\infty,\Xi}(X) = \{y : \exists \text{ control } u(x), T > 0 \text{ s.t.}$$
$$\phi_\Xi(t,y,u) \in S \text{ for } t \in [0,T] \text{ and } \phi_\Xi(T,y,u) \in X\}$$

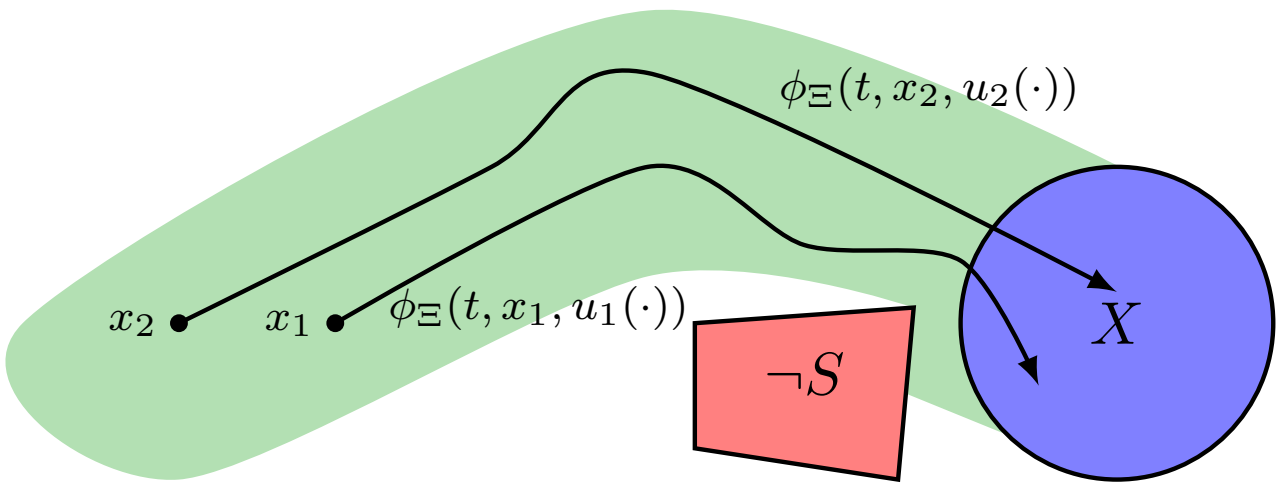
Set-valued reachability operator

For a system $\Xi: \ \dot{x} = f(x,u)$, the **flow operator** $\phi_\Xi(t,y,u)$ satisfies

$$\phi_\Xi(0,y,u) = y, \qquad \frac{\mathrm{d}}{\mathrm{d}t}\phi_\Xi(t,y,u) = f\left(\phi(t,y,u),u\right)$$

For sets X, S, define a set-valued operator for safe reachability:

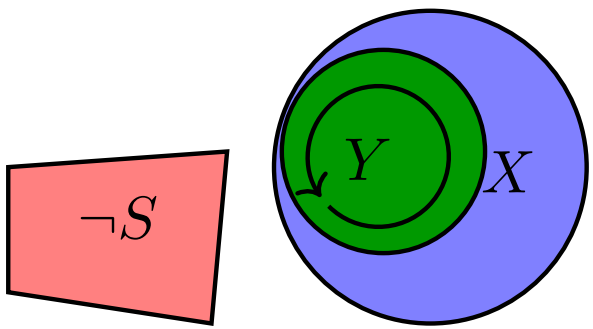
$$\mathrm{Rch}_S^{\infty,\Xi}(X) = \{y : \exists \text{ control } u(x), T > 0 \text{ s.t.}$$
$$\phi_\Xi(t,y,u) \in S \text{ for } t \in [0,T] \text{ and } \phi_\Xi(T,y,u) \in X\}$$



Relation to formal specifications

Many LTL specs can be stated in terms of reachability

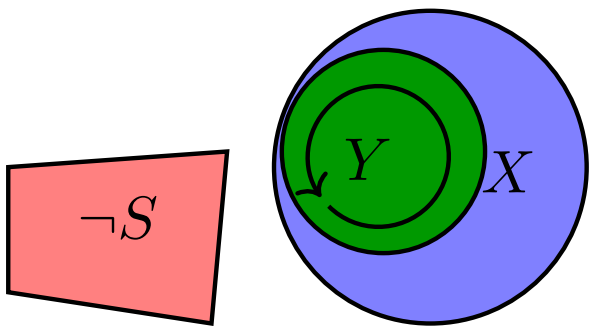
- Safety ($\Box X$): Controlled invariance of X: find $Y : Y = \mathsf{Rch}_X^{\infty,\Xi}(Y)$



Relation to formal specifications

Many LTL specs can be stated in terms of reachability

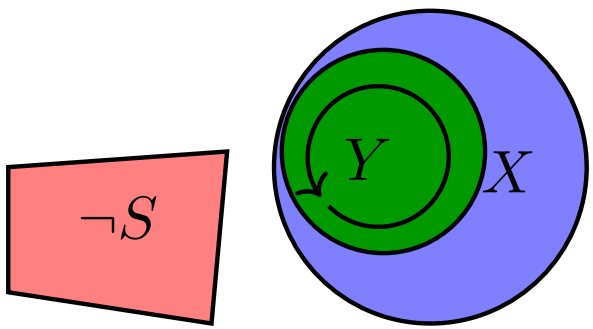
- Safety ($\Box X$): Controlled invariance of X: find $Y : Y = \mathsf{Rch}_X^{\infty,\Xi}(Y)$



Relation to formal specifications

Many LTL specs can be stated in terms of reachability

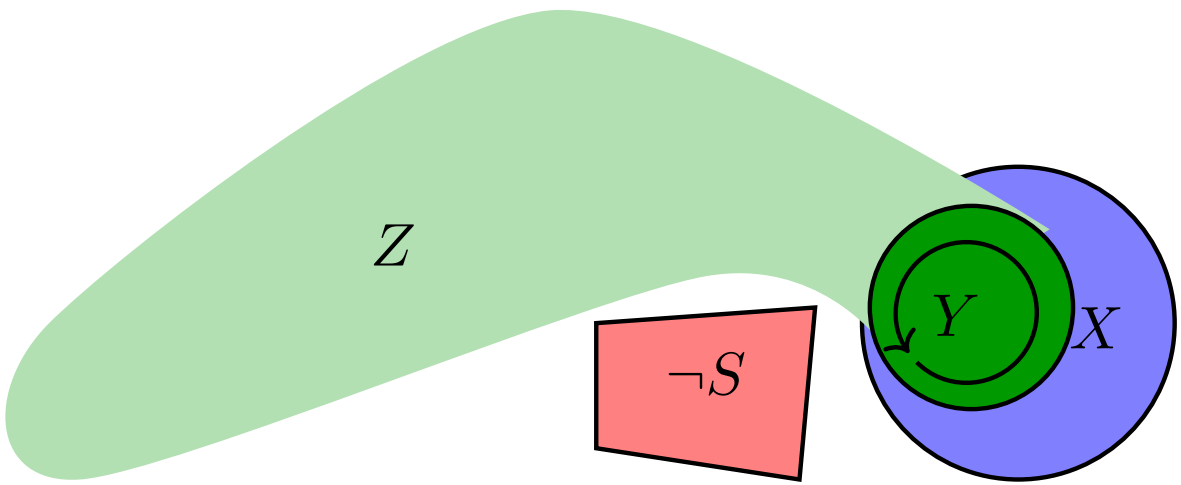
- Safety ($\Box X$): Controlled invariance of X: find $Y : Y = \mathsf{Rch}_X^{\infty,\Xi}(Y)$
- Convergence ($\Diamond\Box X$): Reach X and stay there: $Z : Z = \mathsf{Rch}_S^{\infty,\Xi}(Y)$



Relation to formal specifications

Many LTL specs can be stated in terms of reachability

- Safety ($\Box X$): Controlled invariance of X: find $Y : Y = \mathsf{Rch}_X^{\infty,\Xi}(Y)$
- Convergence ($\Diamond\Box X$): Reach X and stay there: $Z : Z = \mathsf{Rch}_S^{\infty,\Xi}(Y)$



Relation to formal specifications

Many LTL specs can be stated in terms of reachability

- Safety ($\square X$): Controlled invariance of X: find $Y : Y = \mathsf{Rch}_X^{\infty,\Xi}(Y)$
- Convergence ($\lozenge\square X$): Reach X and stay there: $Z : Z = \mathsf{Rch}_S^{\infty,\Xi}(Y)$
- Surveillance ($\wedge_i \square\lozenge G_i$): Circulate between two sets G_1, G_2:

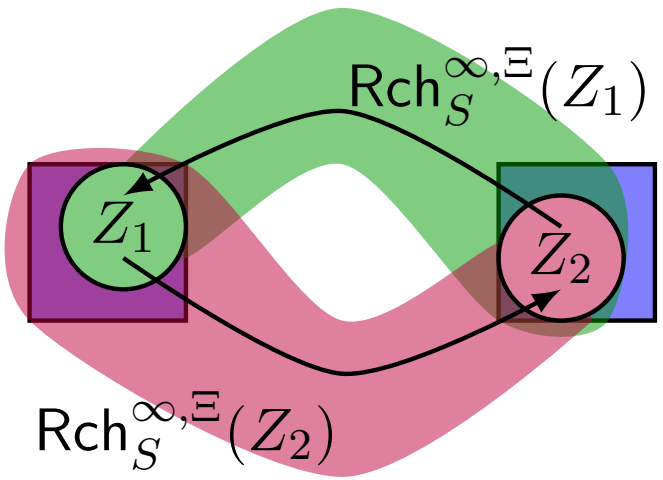
$$Z_1, Z_2 : \; Z_1 \subset G_1 \cap \mathsf{Rch}_S^{\infty,\Xi}(Z_2), \; Z_2 \subset G_2 \cap \mathsf{Rch}_S^{\infty,\Xi}(Z_1)$$

Relation to formal specifications

Many LTL specs can be stated in terms of reachability

- Safety ($\square X$): Controlled invariance of X: find $Y : Y = \mathsf{Rch}_X^{\infty,\Xi}(Y)$
- Convergence ($\lozenge\square X$): Reach X and stay there: $Z : Z = \mathsf{Rch}_S^{\infty,\Xi}(Y)$
- Surveillance ($\wedge_i \square\lozenge G_i$): Circulate between two sets G_1, G_2:

$$Z_1, Z_2 : \; Z_1 \subset G_1 \cap \mathsf{Rch}_S^{\infty,\Xi}(Z_2), \; Z_2 \subset G_2 \cap \mathsf{Rch}_S^{\infty,\Xi}(Z_1)$$

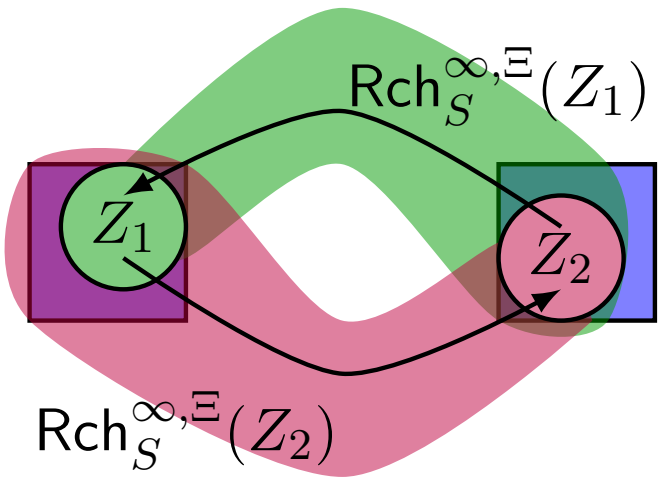


Relation to formal specifications

Many LTL specs can be stated in terms of reachability

- Safety ($\square X$): Controlled invariance of X: find $Y : Y = \mathsf{Rch}_X^{\infty,\Xi}(Y)$
- Convergence ($\lozenge\square X$): Reach X and stay there: $Z : Z = \mathsf{Rch}_S^{\infty,\Xi}(Y)$
- Surveillance ($\wedge_i \square\lozenge G_i$): Circulate between two sets G_1, G_2:

$$Z_1, Z_2 : \; Z_1 \subset G_1 \cap \mathsf{Rch}_S^{\infty,\Xi}(Z_2), \; Z_2 \subset G_2 \cap \mathsf{Rch}_S^{\infty,\Xi}(Z_1)$$



- Difficulty: How to compute $\mathsf{Rch}_S^{\infty,\Xi}$??

Some work related to reachability

- Zonotopes[1]
- Occupation measures/positive polynomials[2]
- Reference governors[3]
- Bernstein methods[4]
- Hamilton-Jacobi equation[5]
- Polytopic reachability for discrete time linear systems

---

[1] A. Girard. "Reachability of Uncertain Linear Systems Using Zonotopes". In: *Proc. HSCC*. 2005, pp. 291–305.

[2] V. Shia, **R. Vasudevan**, R. Bajcsy, and R. Tedrake. "Convex Computation of the Reachable Set for Controlled Polynomial Hybrid Systems". In: *Proc. IEEE CDC*. 2014, pp. 1499–1506.

[3] U. V. Kalabić, I. V. Kolmanovsky, and **E. G. Gilbert**. "Reduced order extended command governor". In: *Automatica* 50.5 (2014), pp. 1466–1472.

[4] T. Dang and R. Testylier. "Reachability analysis for polynomial dynamical systems using the Bernstein expansion". In: *Reliable Computing* 17.2 (2012), pp. 128–152.

[5] I. M. Mitchell, A. M. Bayen, and C. J. Tomlin. "A time-dependent Hamilton-Jacobi formulation of reachable sets for continuous dynamic games". In: *IEEE TAC* 50.7 (2005), pp. 947–957.

Some work related to reachability

- Zonotopes[1]
- Occupation measures/positive polynomials[2]
- Reference governors[3]
- Bernstein methods[4]
- Hamilton-Jacobi equation[5]
- Polytopic reachability for discrete time linear systems

---

[1] A. Girard. "Reachability of Uncertain Linear Systems Using Zonotopes". In: *Proc. HSCC*. 2005, pp. 291–305.

[2] V. Shia, **R. Vasudevan**, R. Bajcsy, and R. Tedrake. "Convex Computation of the Reachable Set for Controlled Polynomial Hybrid Systems". In: *Proc. IEEE CDC*. 2014, pp. 1499–1506.

[3] U. V. Kalabić, I. V. Kolmanovsky, and **E. G. Gilbert**. "Reduced order extended command governor". In: *Automatica* 50.5 (2014), pp. 1466–1472.

[4] T. Dang and R. Testylier. "Reachability analysis for polynomial dynamical systems using the Bernstein expansion". In: *Reliable Computing* 17.2 (2012), pp. 128–152.

[5] I. M. Mitchell, A. M. Bayen, and C. J. Tomlin. "A time-dependent Hamilton-Jacobi formulation of reachable sets for continuous dynamic games". In: *IEEE TAC* 50.7 (2005), pp. 947–957.

Reachability for linear discrete-time dynamics

- Linear system: $\Sigma: \; x(t+1) = Ax(t) + Bu(t) + Ed(t)$
- Input restriction $u \in \mathcal{U} = \{u : H_u u \leq h_u\}$
- Disturbance assumption $d \in \mathcal{D} = \{d : H_d d \leq h_d\}$
- Final set $X = \{x : Hx \leq h\}$

Reachability for linear discrete-time dynamics

- Linear system: $\Sigma: \; x(t+1) = Ax(t) + Bu(t) + Ed(t)$
- Input restriction $u \in \mathcal{U} = \{u : H_u u \leq h_u\}$
- Disturbance assumption $d \in \mathcal{D} = \{d : H_d d \leq h_d\}$
- Final set $X = \{x : Hx \leq h\}$

Valid state-input-disturbance combinations:

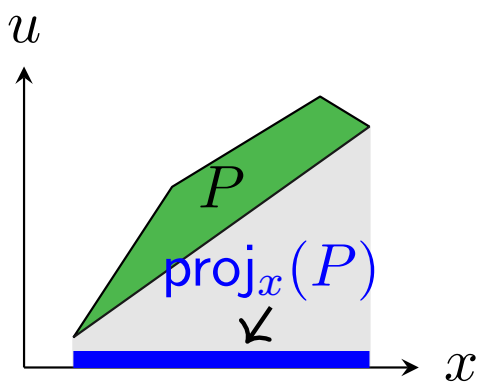
$$P = \{(x, u, d) : Ax + Bu + Ed \in X, \; u \in \mathcal{U}\}$$

Reachability for linear discrete-time dynamics

- Linear system: $\Sigma: \; x(t+1) = Ax(t) + Bu(t) + Ed(t)$
- Input restriction $u \in \mathcal{U} = \{u : H_u u \leq h_u\}$
- Disturbance assumption $d \in \mathcal{D} = \{d : H_d d \leq h_d\}$
- Final set $X = \{x : Hx \leq h\}$

Valid state-input-disturbance combinations:

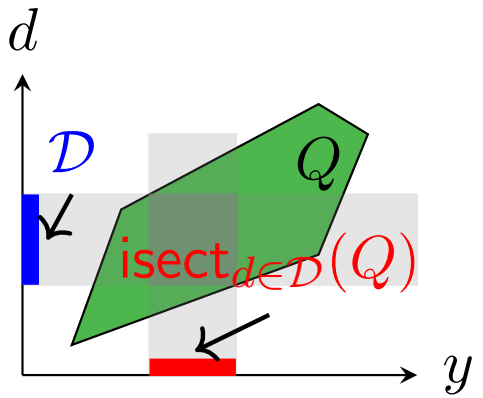
$$P = \{(x, u, d) : Ax + Bu + Ed \in X, \; u \in \mathcal{U}\}$$



Valid initial states:

$$\text{Rch}_S^{1,\Sigma}(X) = S \cap \text{proj}_x \circ \text{isect}_{d \in \mathcal{D}} \; (P)$$

Need only polyhedron projection and intersection

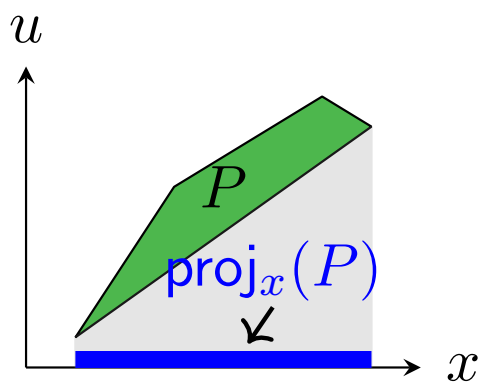


Reachability for linear discrete-time dynamics

- Linear system: $\Sigma: \; x(t+1) = Ax(t) + Bu(t) + Ed(t)$
- Input restriction $u \in \mathcal{U} = \{u : H_u u \leq h_u\}$
- Disturbance assumption $d \in \mathcal{D} = \{d : H_d d \leq h_d\}$
- Final set $X = \{x : Hx \leq h\}$

Valid state-input-disturbance combinations:

$$P = \{(x, u, d) : Ax + Bu + Ed \in X, \; u \in \mathcal{U}\}$$

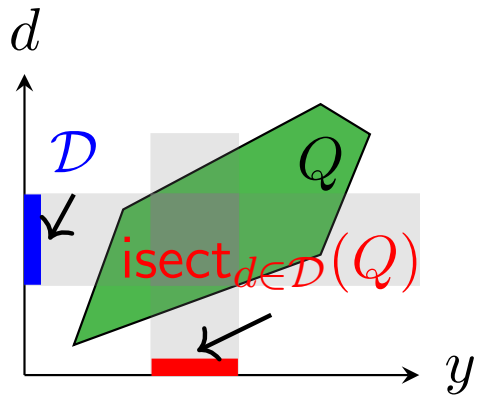


Valid initial states:

$$\text{Rch}_S^{1,\Sigma}(X) = S \cap \text{proj}_x \circ \text{isect}_{d\in\mathcal{D}}\,(P)$$

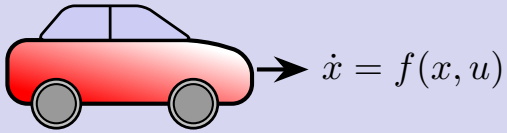
Need only polyhedron projection and intersection

$$\text{Rch}_S^{\infty,\Sigma}(X) = \bigcup_{k\geq 0} \underbrace{\text{Rch}_S^{1,\Sigma} \circ \ldots \circ \text{Rch}_S^{1,\Sigma}}_{k \text{ times}}(X)$$



Outline

Model: continuous dynamics



ACC car dynamics

$$m\dot{v} = F_w - F_r(v)$$
$$\dot{h} = v_L - v$$
$$\dot{v}_L = a_L$$

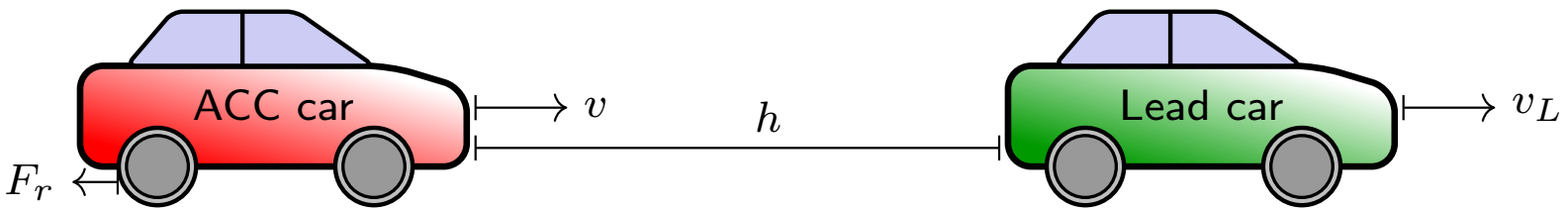
F_r polynomial drag term

Input bound

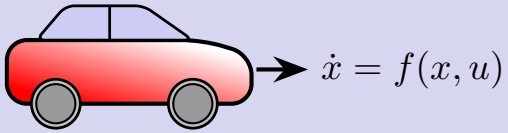
$$F_w \in [-0.3mg, 0.2mg]$$

Lead car assumptions

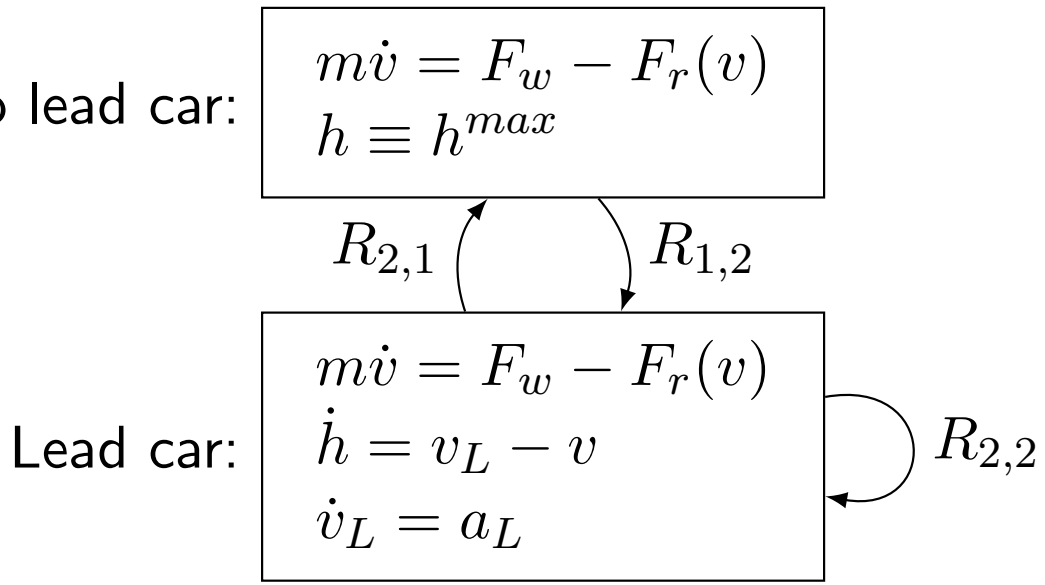
$$v_L \in [v_L^-, v_L^+]$$
$$a_L \in [a_L^-, a_L^+]$$



Model: discrete dynamics



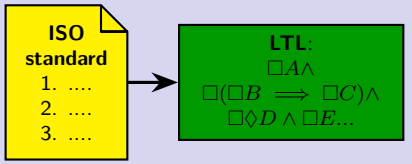
- Hybrid system



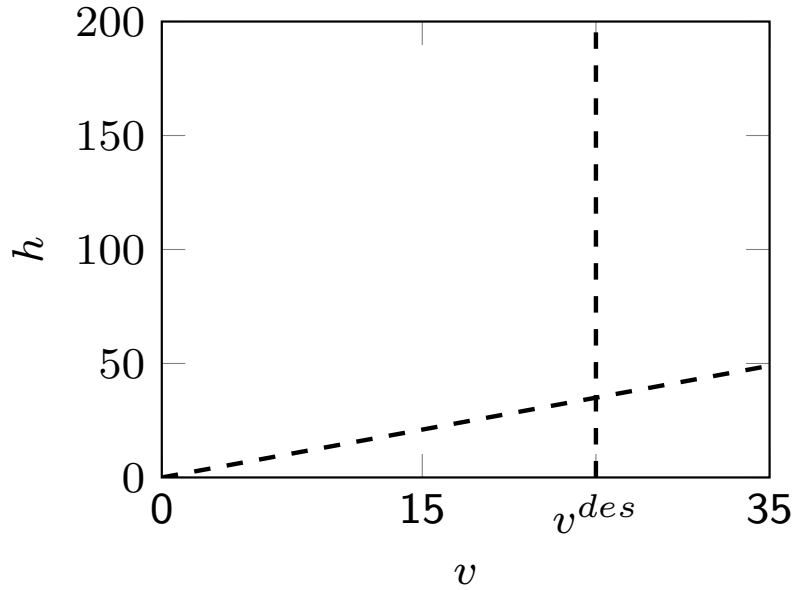
- Reset maps

$$R_{1,2}(v, h^{max}) = \{(v, \bar{h}, \bar{v}_L) : (\bar{h}, \bar{v}_L) \in \mathcal{H} \times \mathcal{V}_L\}$$
$$R_{2,1}(v, h, v_L) = \{(v, h^{max})\}$$
$$R_{2,2}(v, h, v_L) = \{(v, \bar{h}, \bar{v}_L) : (\bar{h}, \bar{v}_L) \in \mathcal{H} \times \mathcal{V}_L\}$$

Textual to formal specification

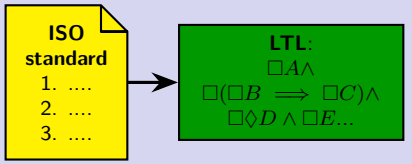


ISO: "When the ACC is active, the vehicle speed shall be controlled automatically either to maintain a time gap (h/v) to a forward vehicle, or to maintain the set speed v_{des}, whichever speed is lower."

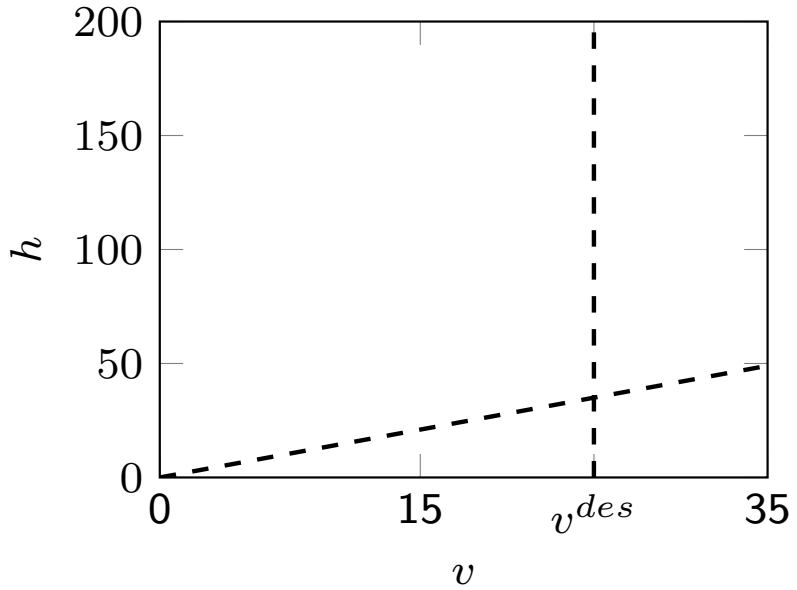


LTL : $\left(\bigwedge_{i=1}^{2} (\quad \Rightarrow \quad) \right)$

Textual to formal specification

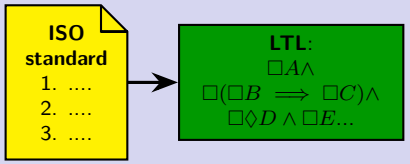


ISO: "When the ACC is active, the vehicle speed shall be controlled automatically either to maintain a time gap (h/v) to a forward vehicle, or to maintain the set speed v_{des}, whichever speed is lower."

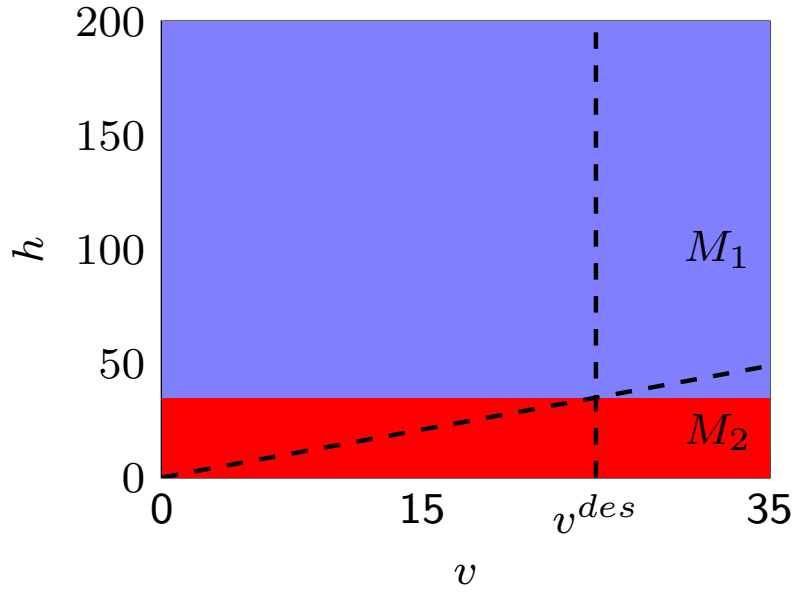


LTL : $\left(\bigwedge_{i=1}^{2} (\quad \Rightarrow \quad) \right)$

Textual to formal specification



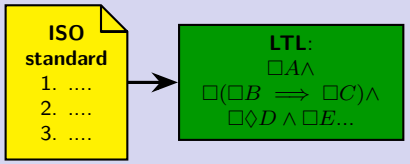
ISO: "When the ACC is active, the vehicle speed shall be controlled automatically either to maintain a time gap (h/v) to a forward vehicle, or to maintain the set speed v_{des}, whichever speed is lower."



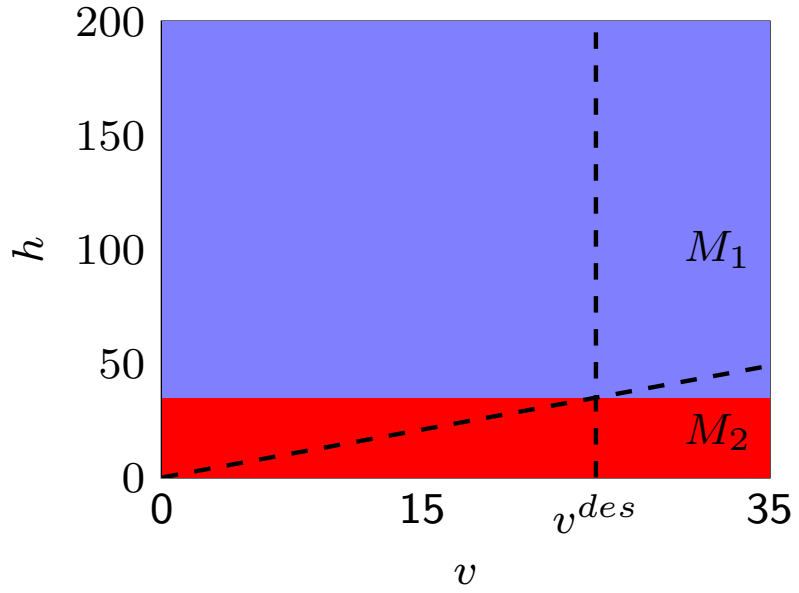
$$\text{LTL}: \quad \left(\bigwedge_{i=1}^{2} (\Box M_i \Rightarrow \qquad) \right)$$

- $\Box$ = "always"

Textual to formal specification



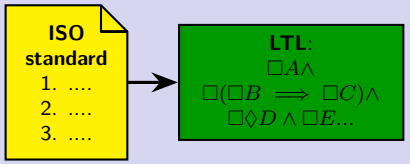
ISO: "When the ACC is active, the vehicle speed shall be controlled automatically either to maintain a time gap (h/v) to a forward vehicle, or to maintain the set speed v_{des}, whichever speed is lower."



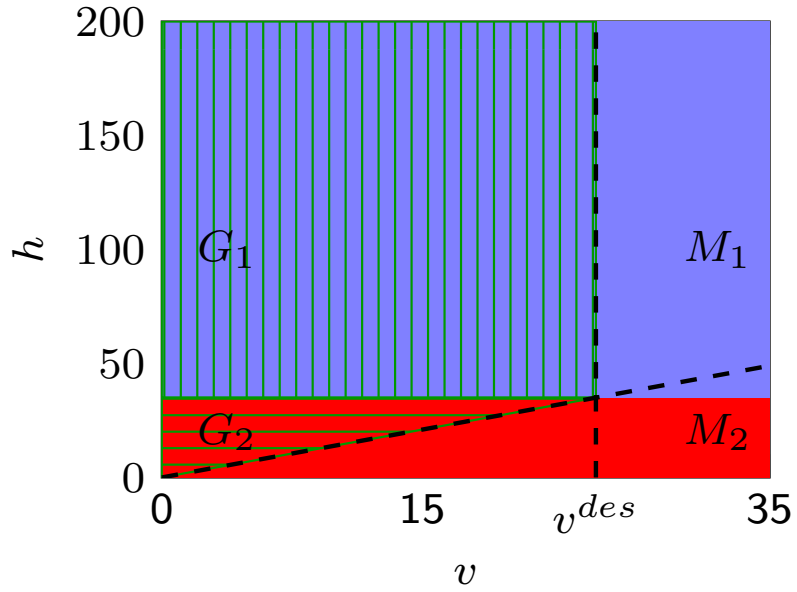
LTL : $\left(\bigwedge_{i=1}^{2} (\Box M_i \Rightarrow \quad) \right)$

- $\Box =$ "always"

Textual to formal specification



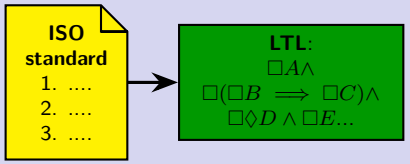
ISO: "When the ACC is active, the vehicle speed shall be controlled automatically either to maintain a time gap (h/v) to a forward vehicle, or to maintain the set speed v_{des}, whichever speed is lower."



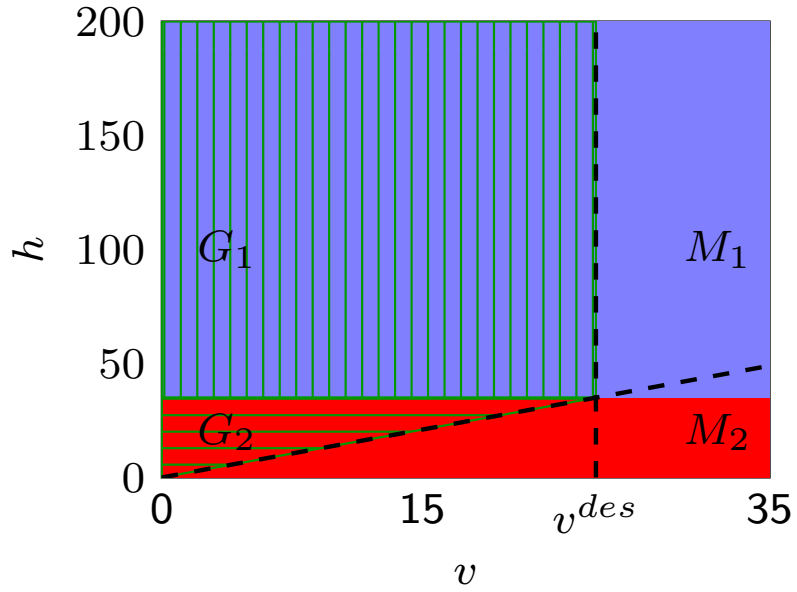
$$\text{LTL}: \quad \left(\bigwedge_{i=1}^{2} (\Box M_i \Rightarrow \Diamond \Box G_i) \right)$$

- $\Box$ = "always", $\Diamond$ = "eventually"

Textual to formal specification



ISO: "When the ACC is active, the vehicle speed shall be controlled automatically either to maintain a time gap (h/v) to a forward vehicle, or to maintain the set speed v_{des}, whichever speed is lower."



$$\text{LTL}: \Box S_U \wedge \Box \left(\bigwedge_{i=1}^{2} (\Box M_i \Rightarrow \Diamond \Box G_i) \right)$$

- $\Box =$ "always", $\Diamond =$ "eventually"
- Input constraint $S_U = \{F_w : F_w \in [-0.3mg, 0.2mg]\}$

Fixed point interpretation of specification

- Solve for specification of the form

$$\square\left((\square M_1 \implies \lozenge\square G_1) \wedge (\square M_2 \implies \lozenge\square G_2)\right)$$

For Inv^∞ controlled invariance operator, want sets C_1 and C_2 s.t.

$$C_1 \subset M_1 \cap \mathsf{Rch}_S^\infty \underbrace{\left(\mathsf{Inv}^\infty \left(G_1 \cap (C_1 \cup C_2)\right) \cup C_2\right)}_{D_1},$$

$$C_2 \subset M_2 \cap \mathsf{Rch}_S^\infty \underbrace{\left(\mathsf{Inv}^\infty \left(G_2 \cap (C_1 \cup C_2)\right) \cup C_1\right)}_{D_2}.$$

Correct control strategy if such C_1, C_2 are found:

- When in C_1, make progress toward D_1
- When in C_2, make progress toward D_2

Want to make C_1 and C_2 as large as possible to maximize controller domain

Fixed point interpretation of specification

- Solve for specification of the form

$$\square\left((\square M_1 \implies \lozenge\square G_1) \wedge (\square M_2 \implies \lozenge\square G_2)\right)$$

For Inv^∞ controlled invariance operator, want sets C_1 and C_2 s.t.

$$C_1 \subset M_1 \cap \mathsf{Rch}_S^\infty \underbrace{\left(\mathsf{Inv}^\infty \left(G_1 \cap (C_1 \cup C_2)\right) \cup C_2\right)}_{D_1},$$

$$C_2 \subset M_2 \cap \mathsf{Rch}_S^\infty \underbrace{\left(\mathsf{Inv}^\infty \left(G_2 \cap (C_1 \cup C_2)\right) \cup C_1\right)}_{D_2}.$$

Correct control strategy if such C_1, C_2 are found:

- When in C_1, make progress toward D_1
- When in C_2, make progress toward D_2

Want to make C_1 and C_2 as large as possible to maximize controller domain

Fixed point interpretation of specification

- Solve for specification of the form

$$\square\left((\square M_1 \implies \lozenge\square G_1) \wedge (\square M_2 \implies \lozenge\square G_2)\right)$$

For Inv^∞ controlled invariance operator, want sets C_1 and C_2 s.t.

$$C_1 \subset M_1 \cap \mathsf{Rch}_S^\infty \underbrace{\left(\mathsf{Inv}^\infty \left(G_1 \cap (C_1 \cup C_2)\right) \cup C_2\right)}_{D_1},$$

$$C_2 \subset M_2 \cap \mathsf{Rch}_S^\infty \underbrace{\left(\mathsf{Inv}^\infty \left(G_2 \cap (C_1 \cup C_2)\right) \cup C_1\right)}_{D_2}.$$

Correct control strategy if such C_1, C_2 are found:

- When in C_1, make progress toward D_1
- When in C_2, make progress toward D_2

Want to make C_1 and C_2 as large as possible to maximize controller domain

Fixed point interpretation of specification

- Solve for specification of the form

$$\square\left((\square M_1 \implies \lozenge\square G_1) \wedge (\square M_2 \implies \lozenge\square G_2)\right)$$

For Inv^∞ controlled invariance operator, want sets C_1 and C_2 s.t.

$$C_1 \subset M_1 \cap \mathsf{Rch}_S^\infty \underbrace{\left(\mathsf{Inv}^\infty \left(G_1 \cap (C_1 \cup C_2)\right) \cup C_2\right)}_{D_1},$$

$$C_2 \subset M_2 \cap \mathsf{Rch}_S^\infty \underbrace{\left(\mathsf{Inv}^\infty \left(G_2 \cap (C_1 \cup C_2)\right) \cup C_1\right)}_{D_2}.$$

Correct control strategy if such C_1, C_2 are found:

- When in C_1, make progress toward D_1
- When in C_2, make progress toward D_2

Want to make C_1 and C_2 as large as possible to maximize controller domain

Solution via Polyhedral Invariant Set Computation (PCIS)

- Set computations directly on the continuous state space of a linearized system
  - Conservative linearization
  - Reachability in linearized system implies reachability in original system

$$C_1^0 = M_1,\ C_2^0 = M_2$$

$$C_1^{k+1} = M_1 \cap \mathsf{Rch}_S^\infty \left(\mathsf{Inv}^\infty \left(G_1 \cap (C_1^k \cup C_2^k) \right) \cup C_2^k \right)$$

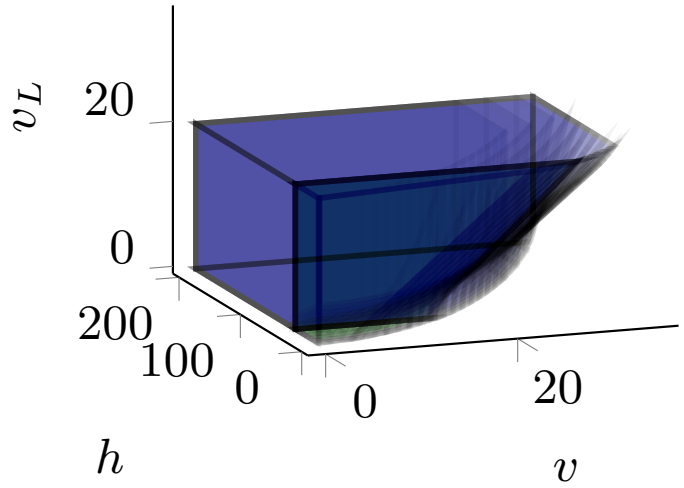
$$C_2^{k+1} = M_2 \cap \mathsf{Rch}_S^\infty \left(\mathsf{Inv}^\infty \left(G_2 \cap (C_1^k \cup C_2^k) \right) \cup C_1^k \right)$$

- Use approximations to keep sets simple
- Iterations converge after one step

PCIS: Computations

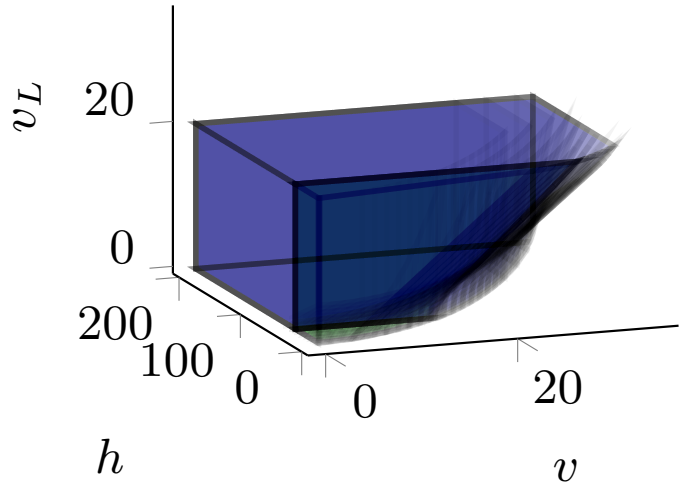
PCIS: Domain

"Normal" lead car

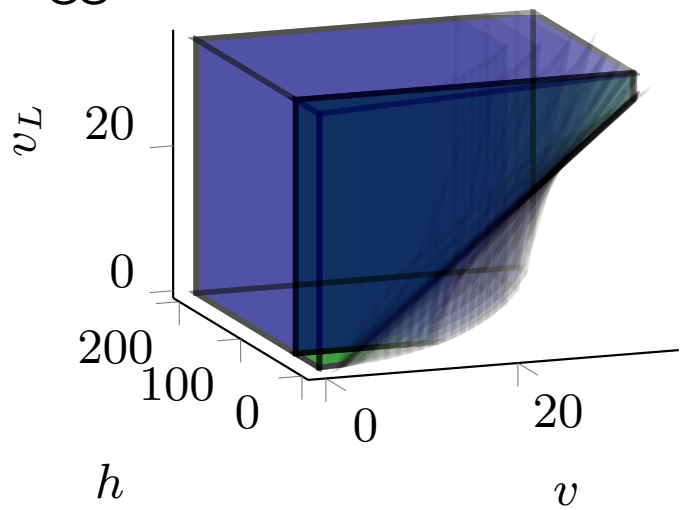


PCIS: Domain

"Normal" lead car

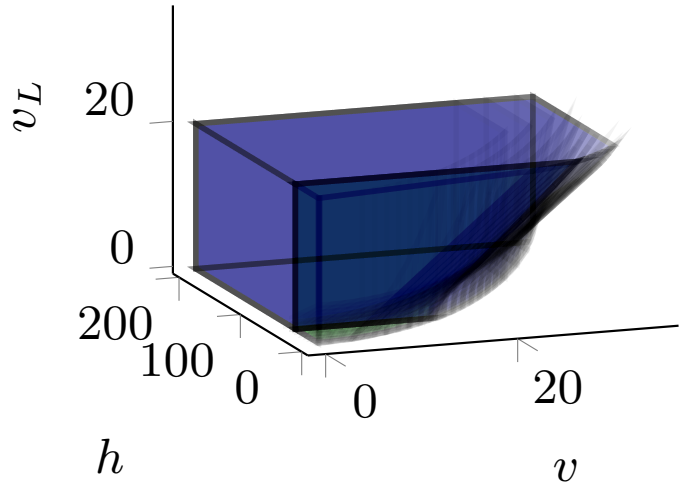


"Aggressive" lead car

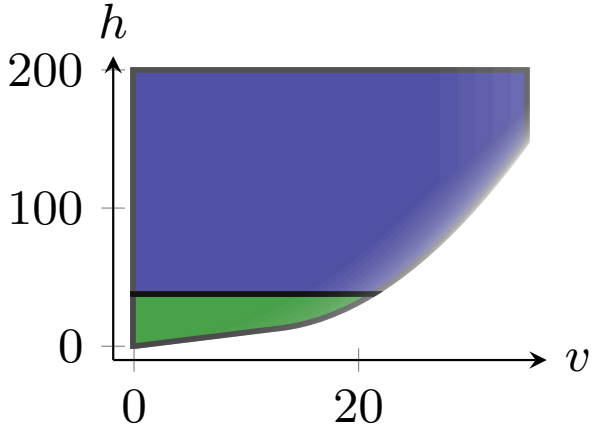


PCIS: Domain

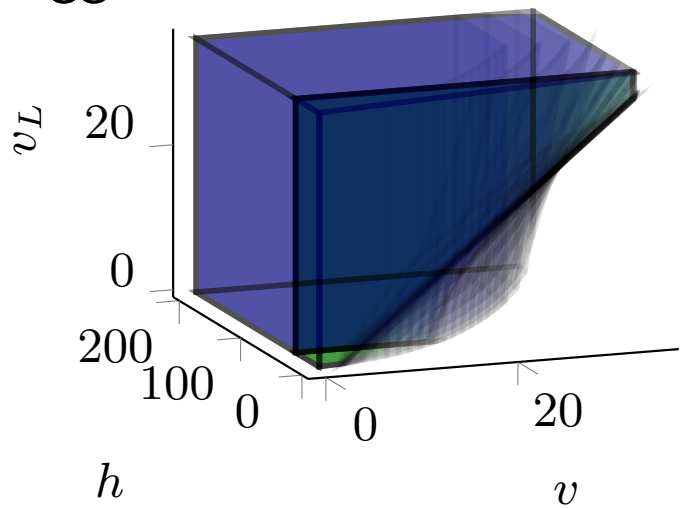
"Normal" lead car



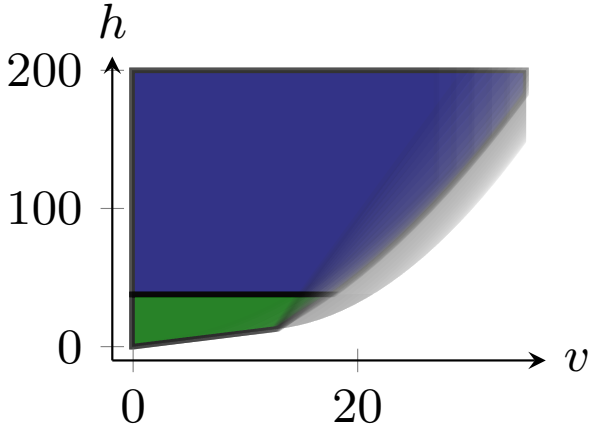
Cross section at $v_L = 10$ m/s



"Aggressive" lead car

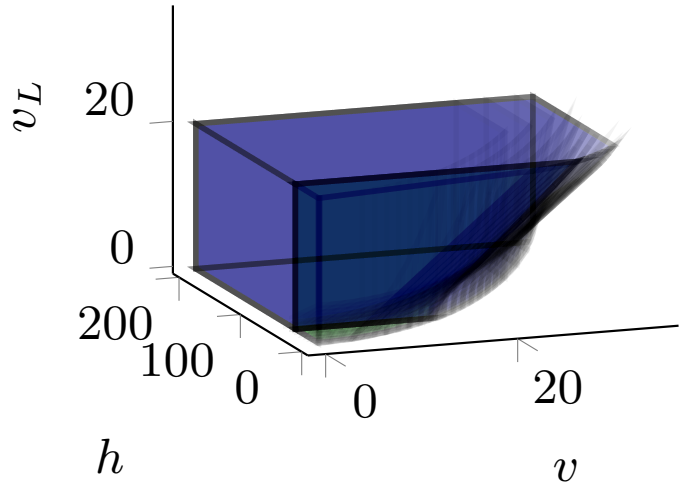


Cross section at $v_L = 10$ m/s

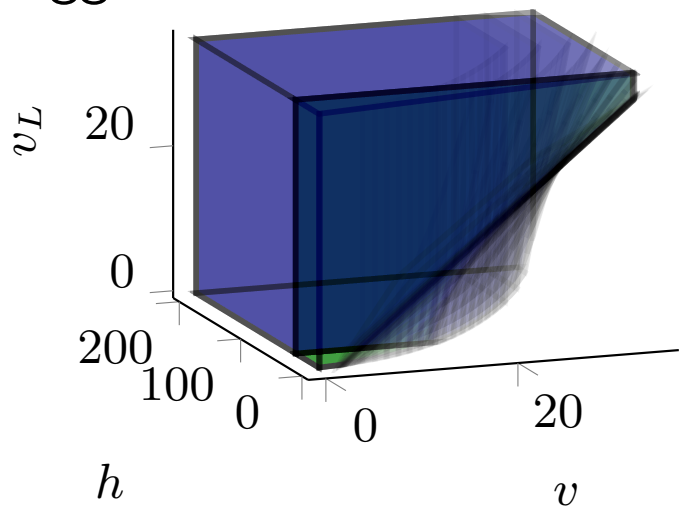


PCIS: Domain

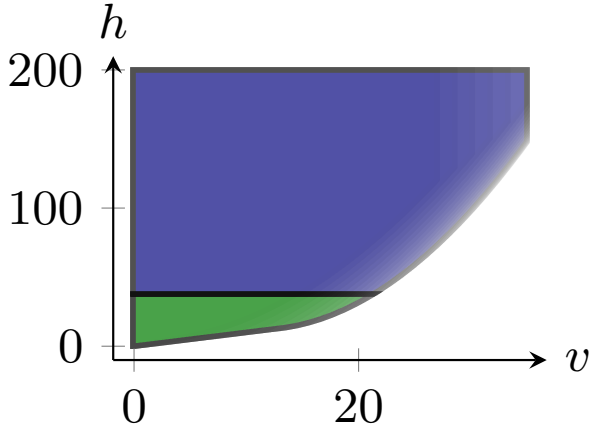
"Normal" lead car



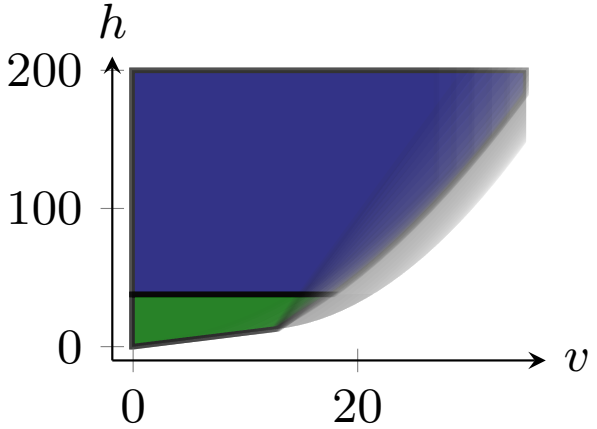
"Aggressive" lead car



Cross section at $v_L = 10$ m/s



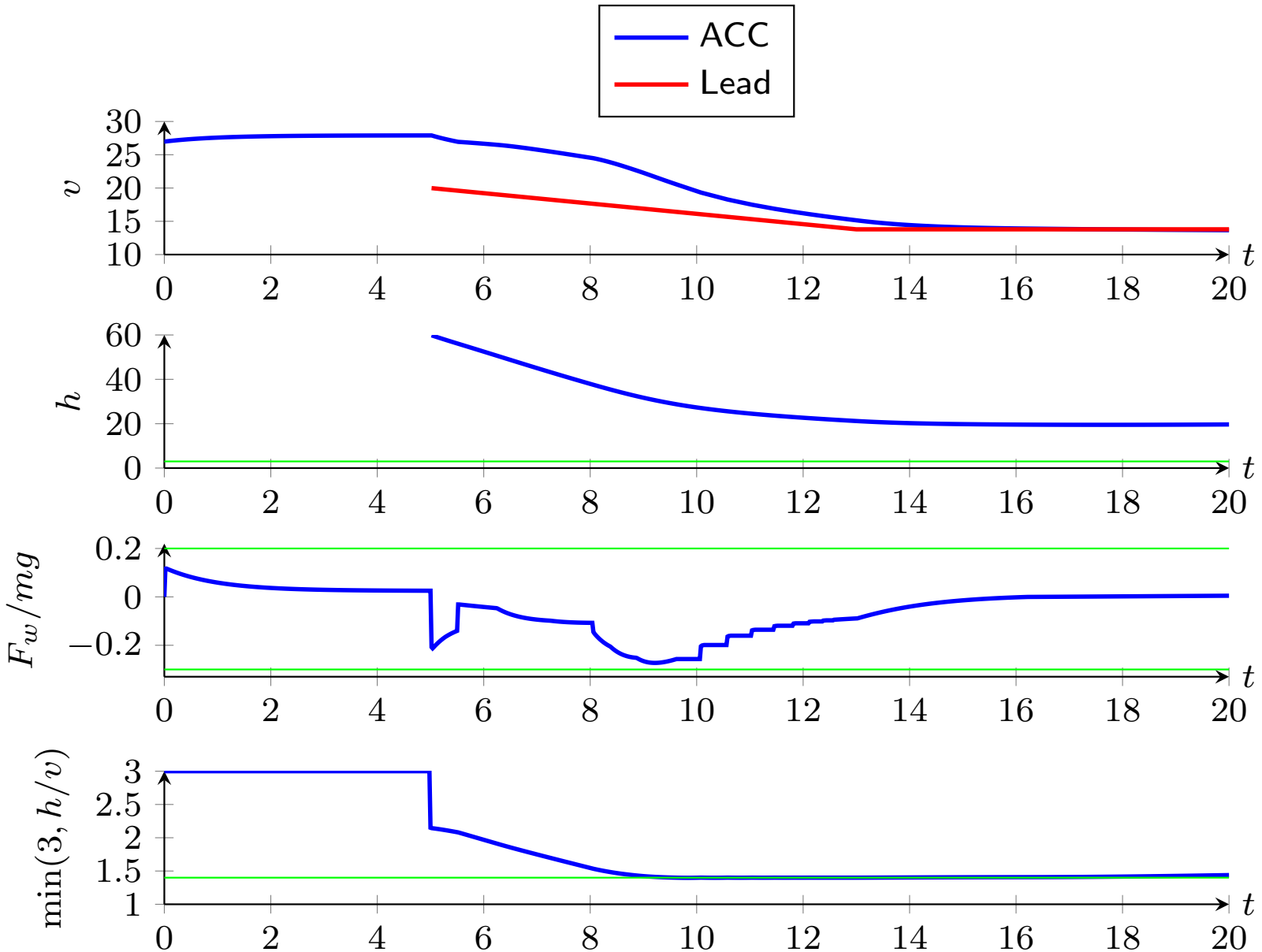
Cross section at $v_L = 10$ m/s



- Implementation: use quadratic program with constraints to move between sets

Simulations: video

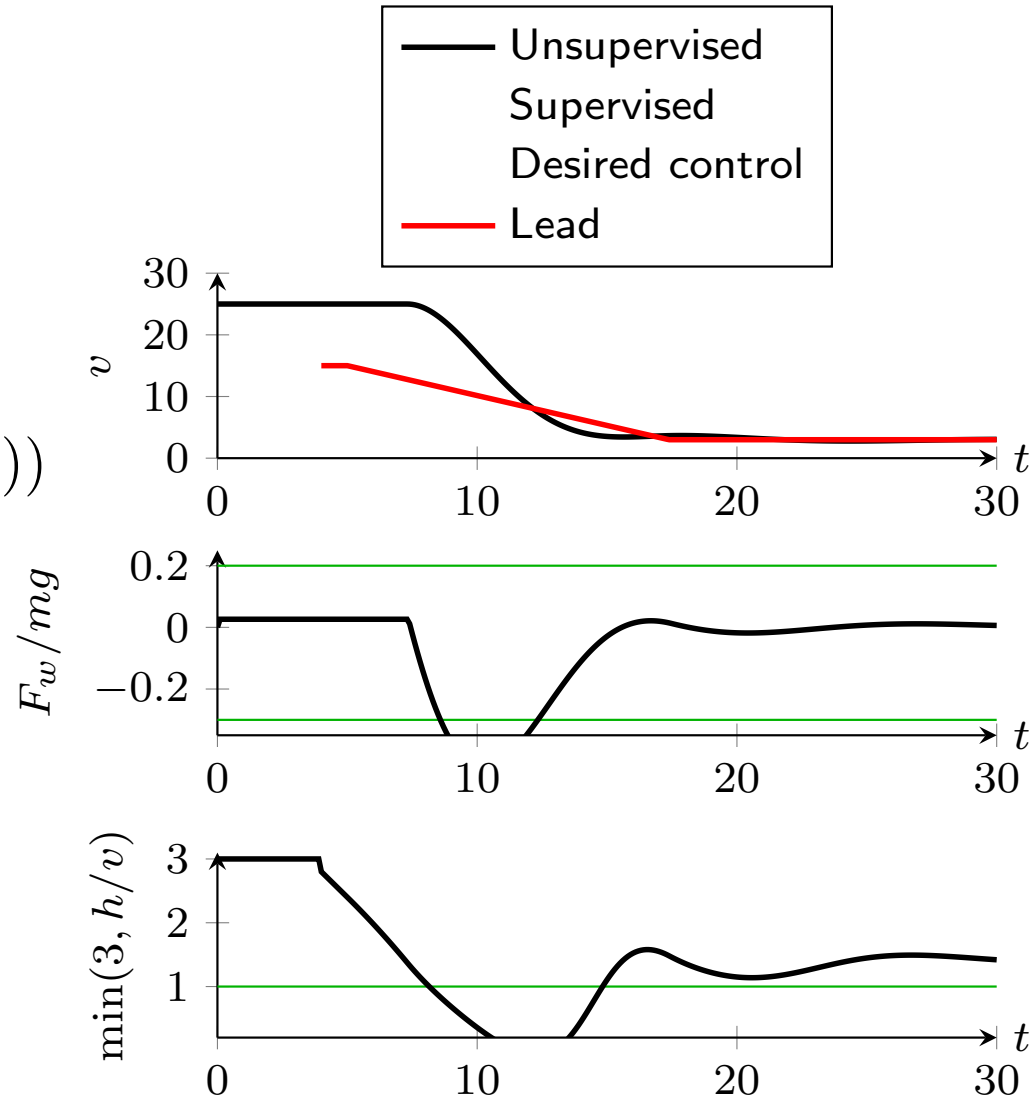
Simulations: plots



Controller supervision

Naive controller:

$$F_r(v) - k(v - \min(v^{des}, h/\omega_{des}))$$



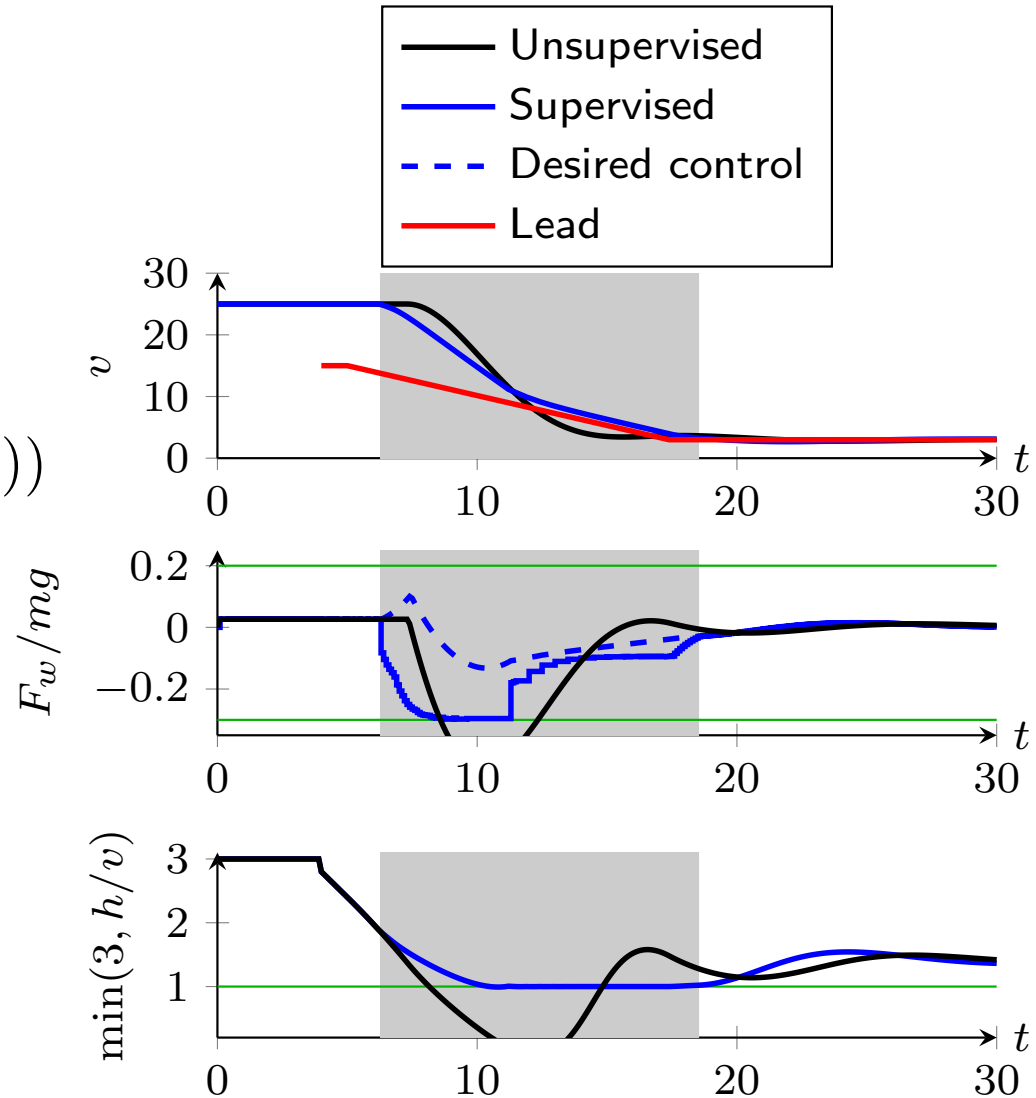
Controller supervision

Naive controller:

$$F_r(v) - k(v - \min(v^{des}, h/\omega_{des}))$$

Supervision:

- Guarantees safety
- Minimally intrusive



Summary

- Partially automated formal synthesis

Summary

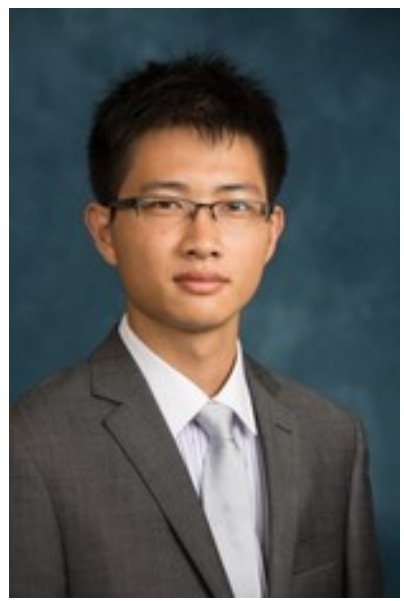
- Partially automated formal synthesis
- Explicitly computed "safe set": enables supervision/alerts

Summary

- Partially automated formal synthesis
- Explicitly computed "safe set": enables supervision/alerts
- Test reasonableness of specification and model instead of performance
  - May be unrealizable
  - May result in unintended behavior (e.g., satisfy $A \implies B$ by making A false)

Thank you for your attention!

Next speaker: Yuxiao Chen



Extra: Method comparison

	PCIS	Pessoa	Barrier fun.
Complex Specifications	Not yet	Yes	TBD
Automation	For LTI	Yes	Not yet
Approximation bounds	For LTI	For IS[6]	Maybe
Parameter tuning	Yes	Some	Yes
Non-linear	Not now	Yes	Yes
High-dimensional	Not now	Not now	Yes
Termination guarantees	Approximate	Yes	N/A

[6] Incrementally Stable Systems