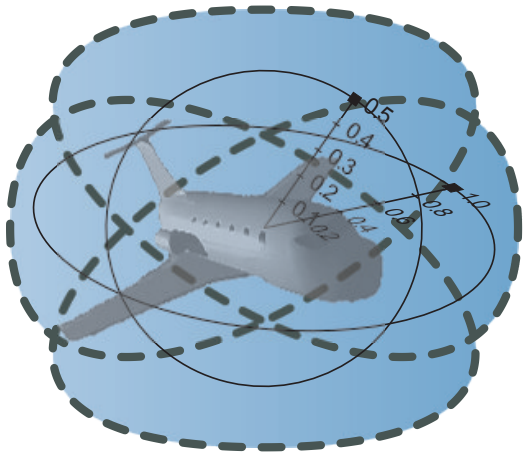
Logic of Autonomous Dynamical Systems

André Platzer

Karlsruhe Institute of Technology
Department of Informatics

Computer Science Department
Carnegie Mellon University

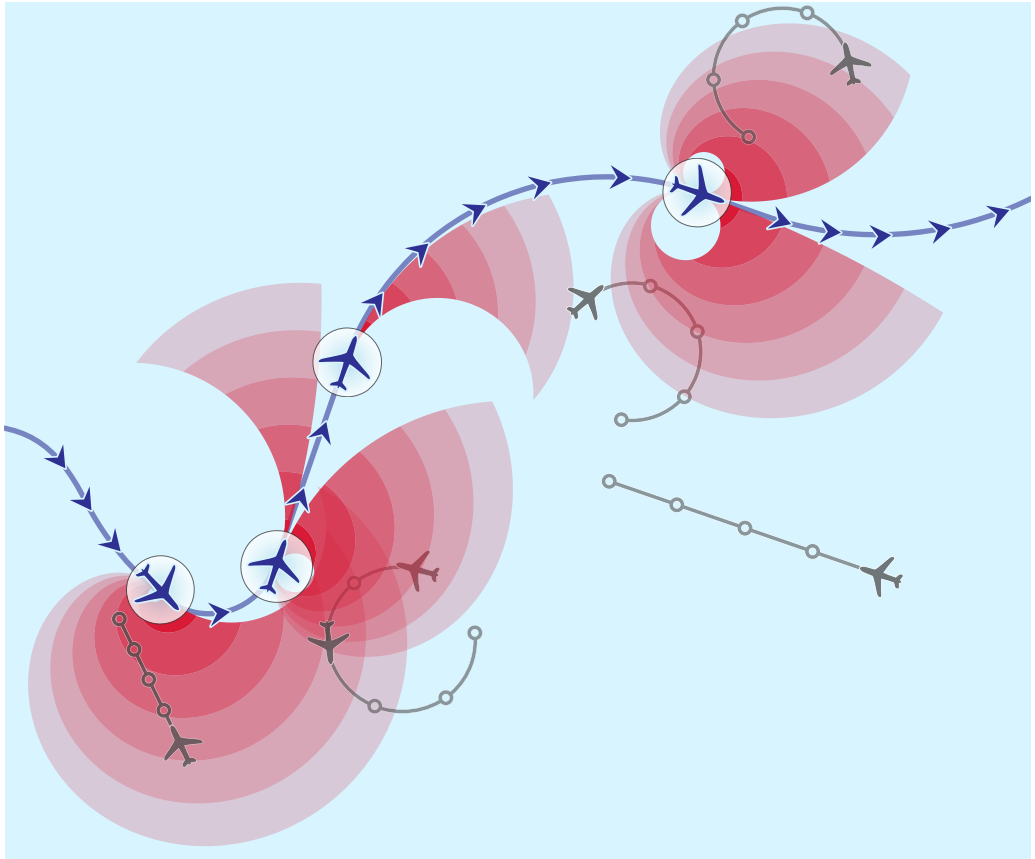


Unterstützt von / Supported by

Alexander von Humboldt
Stiftung/Foundation

Cyber-Physical Systems Analysis: Aircraft Example

Which control decisions are safe for aircraft collision avoidance?



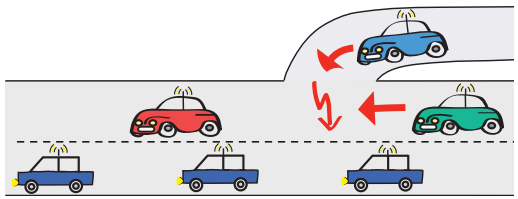
Cyber-Physical Systems

CPSs combine cyber capabilities with physical capabilities
to solve problems that neither part could solve alone.

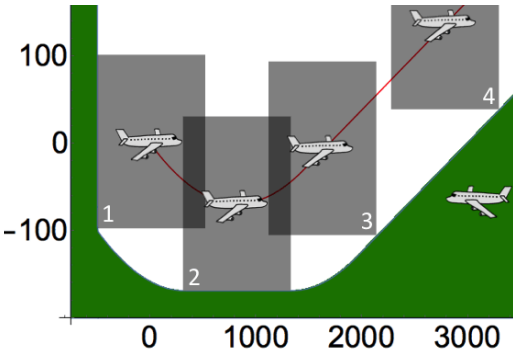
Cyber-Physical Systems Promise Transformative Impact!

Prospects: Safety & Efficiency

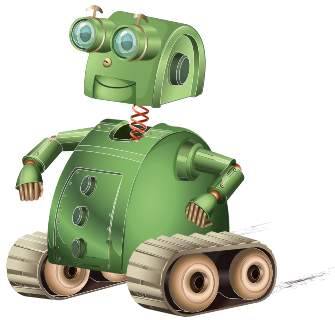
(Autonomous) cars



(Auto)Pilot support



Robots near humans



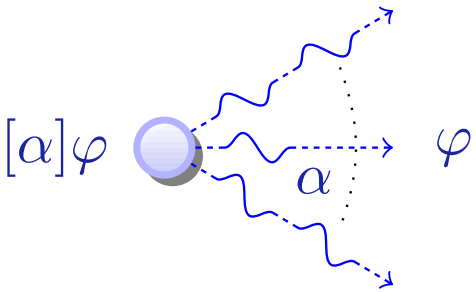
Cyber-Physical Systems

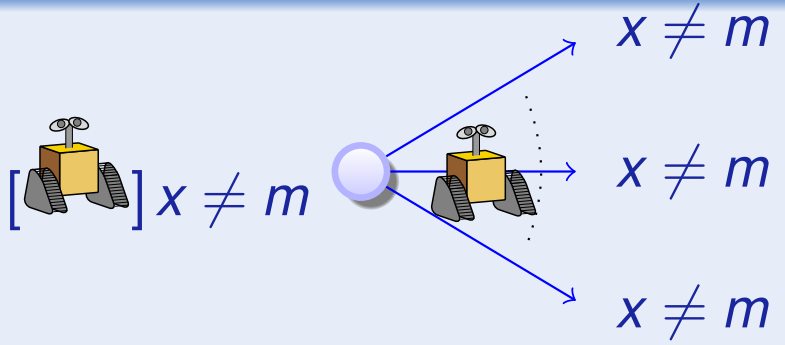
CPSs combine cyber capabilities with physical capabilities
to solve problems that neither part could solve alone.

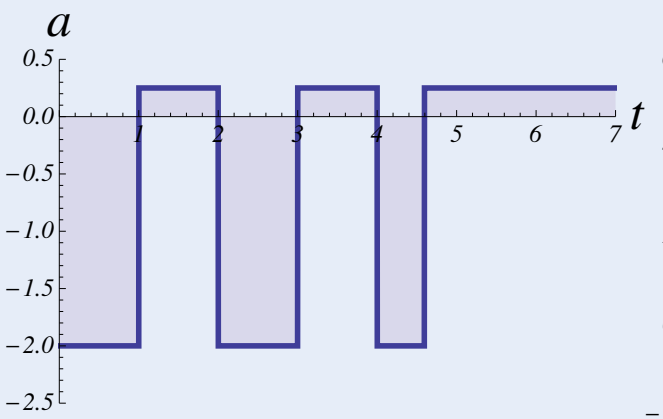
Outline

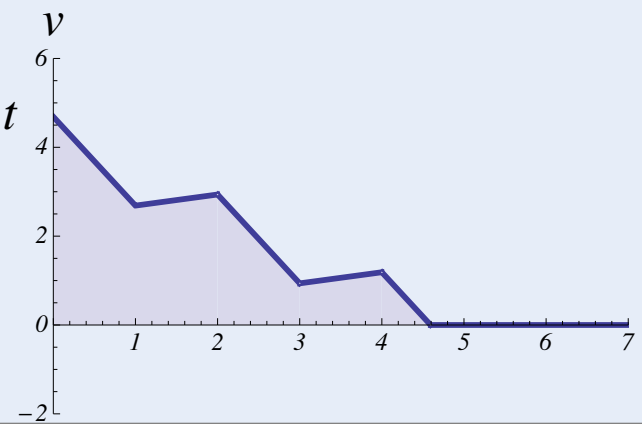
Outline

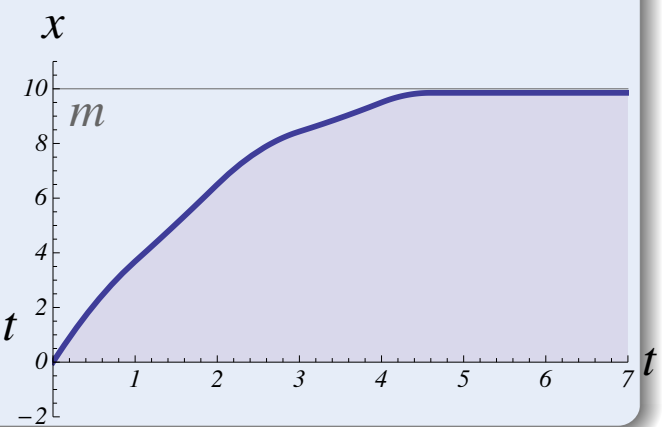
Concept (Differential Dynamic Logic) (JAR'08,LICS'12)



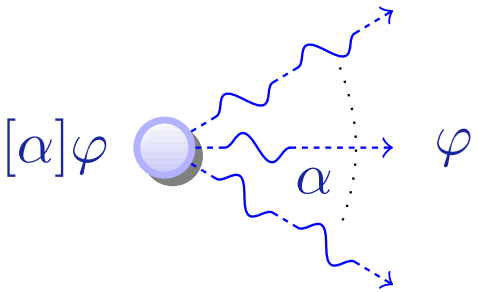


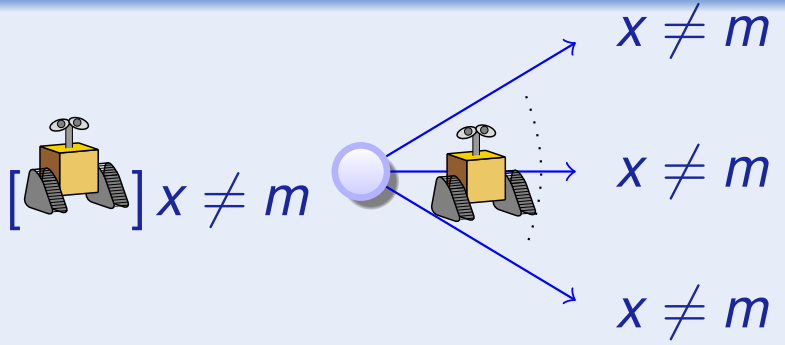






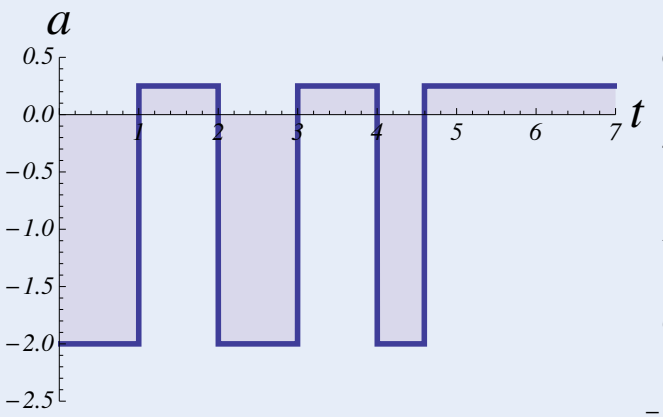
Concept (Differential Dynamic Logic) (JAR'08,LICS'12)

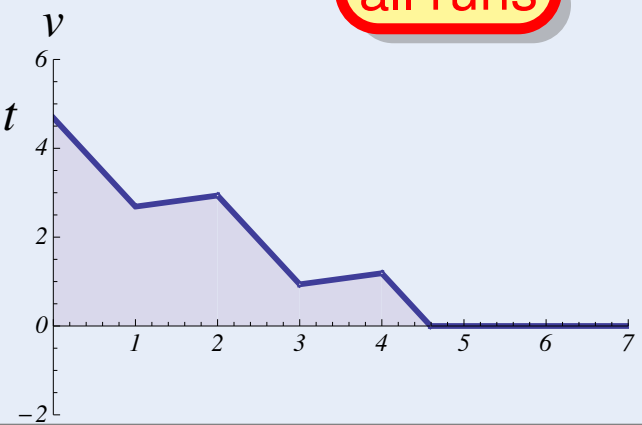


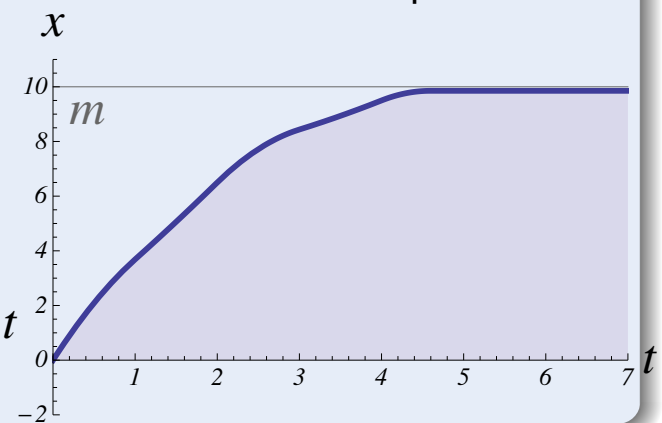


$$\big[\big((\texttt{if}(\mathrm{SB}(x,m))\; a := -b)\;;\; x' = v, v' = a\big)^*\big]\underbrace{x \neq m}_{\text{post}}$$

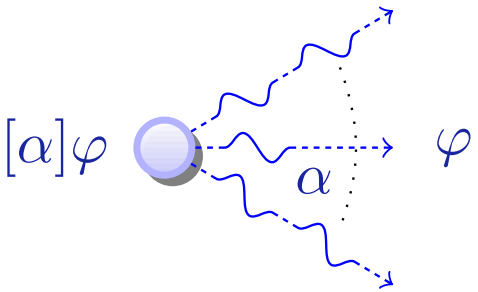
all runs

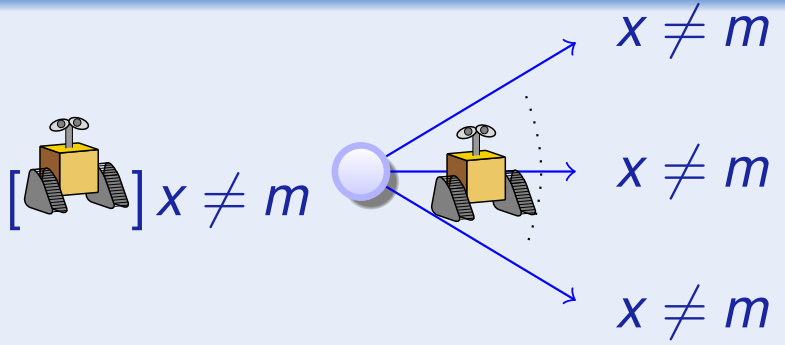




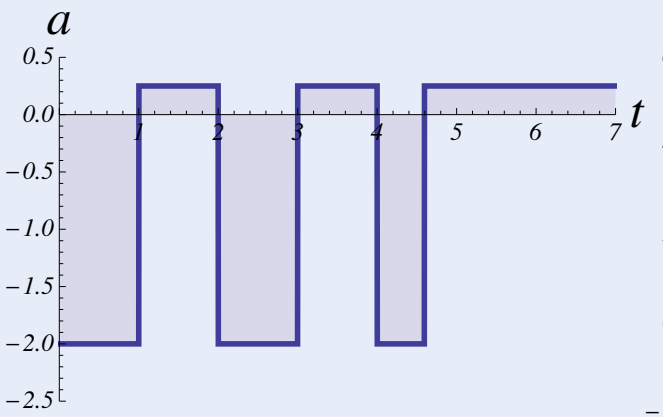


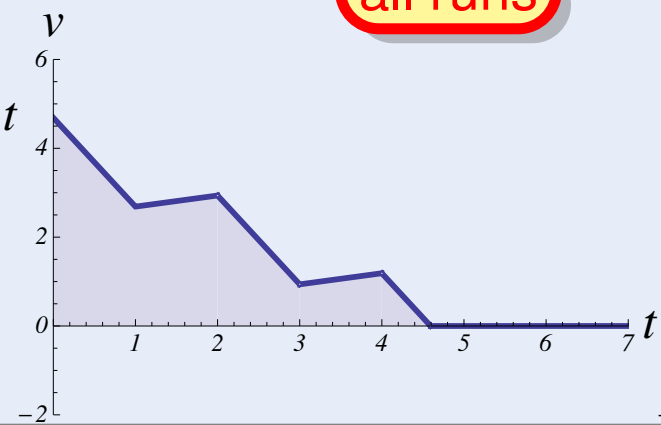
Concept (Differential Dynamic Logic) (JAR'08,LICS'12)

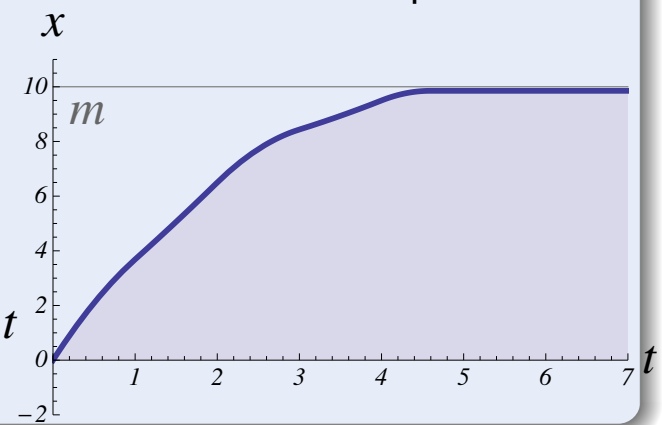




$$\underbrace{x \neq m \wedge b > 0}_{\text{init}} \rightarrow \Big[\underbrace{\big((\texttt{if}(\mathrm{SB}(x,m))\ a := -b)\ ;\ x' = v, v' = a\big)^*}_{\text{all runs}}\Big]\underbrace{x \neq m}_{\text{post}}$$



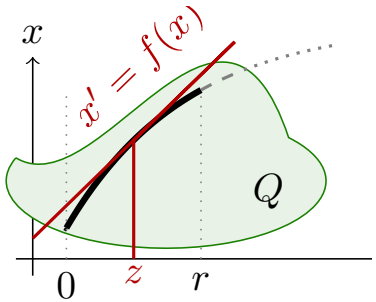


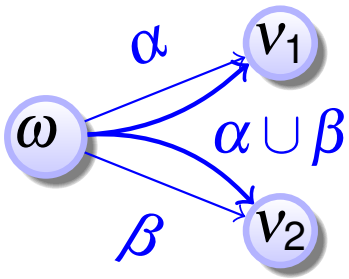


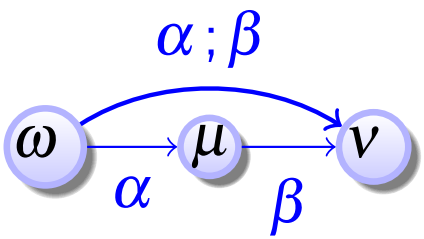
Differential Dynamic Logic dL

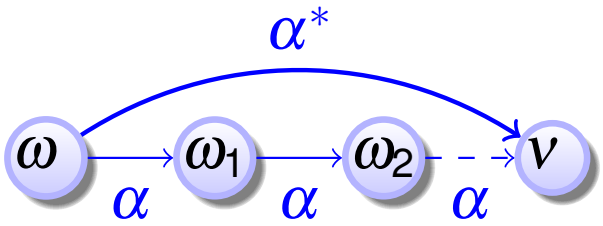
Definition (Hybrid program)

$$\alpha, \beta \ ::= \ x := e \mid ?Q \mid x' = f(x) \,\&\, Q \mid \alpha \cup \beta \mid \alpha; \beta \mid \alpha^*$$



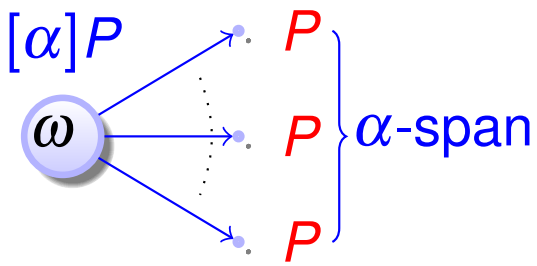






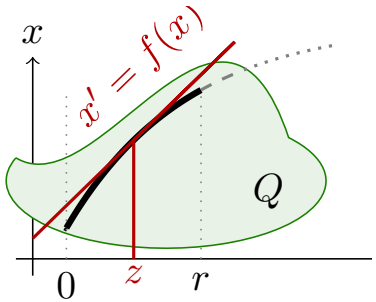
Definition (Differential dynamic logic)

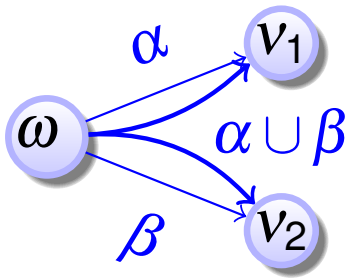
$$P, Q ::= e \geq \tilde{e} \mid \neg P \mid P \wedge Q \mid P \vee Q \mid P \rightarrow Q \mid \forall x\, P \mid \exists x\, P \mid [\alpha]P \mid \langle \alpha \rangle P$$

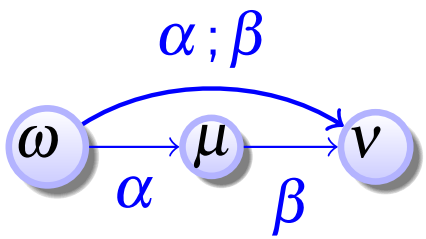


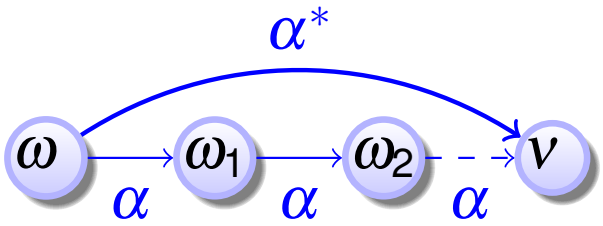
Definition (Hybrid program)

$$\alpha, \beta \;::=\; x := e \mid ?Q \mid x' = f(x)\,\&\,Q \mid \alpha \cup \beta \mid \alpha;\beta \mid \alpha^*$$



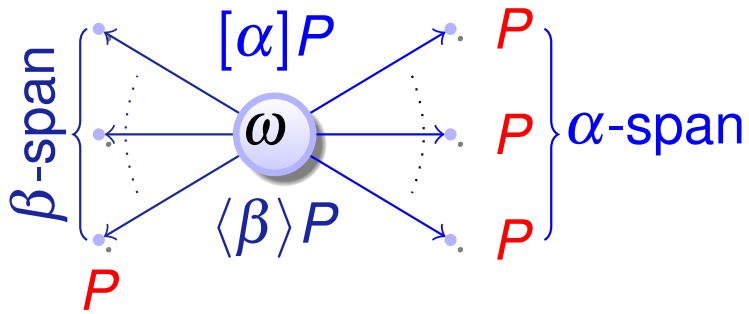






Definition (Differential dynamic logic)

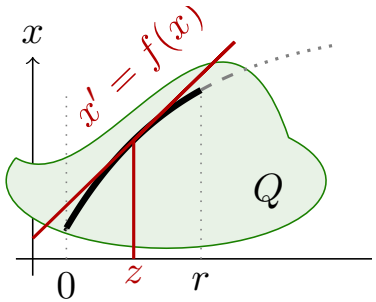
$$P, Q ::= e \geq \tilde{e} \mid \neg P \mid P \wedge Q \mid P \vee Q \mid P \rightarrow Q \mid \forall x\, P \mid \exists x\, P \mid [\alpha]P \mid \langle\alpha\rangle P$$

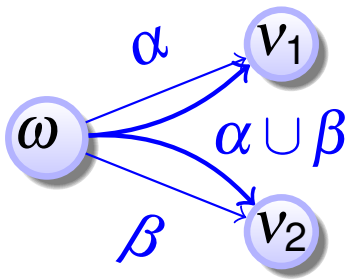


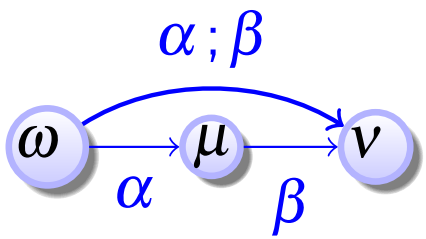
Differential Dynamic Logic dL

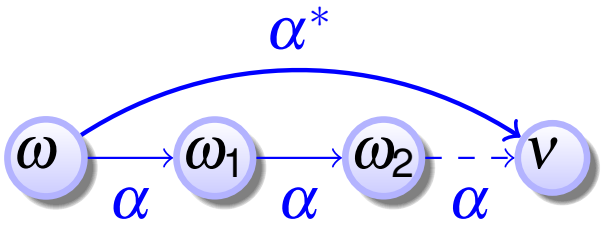
Definition (Hybrid program)

$$\alpha, \beta \;::=\; x := e \mid ?Q \mid x' = f(x)\,\&\,Q \mid \alpha \cup \beta \mid \alpha;\beta \mid \alpha^*$$



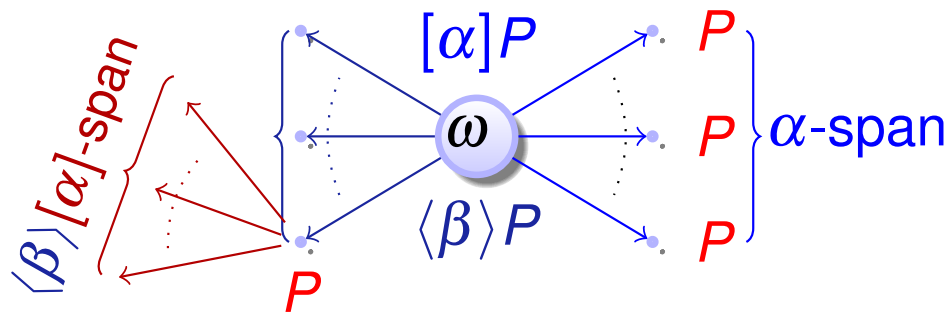






Definition (Differential dynamic logic)

$$P, Q ::= e \geq \tilde{e} \mid \neg P \mid P \wedge Q \mid P \vee Q \mid P \rightarrow Q \mid \forall x\, P \mid \exists x\, P \mid [\alpha]P \mid \langle\alpha\rangle P$$



Differential Dynamic Logic dL: Semantics

Definition (Hybrid program semantics) $([\![\cdot]\!] : \text{HP} \to \wp(\mathscr{S} \times \mathscr{S}))$

$$
\begin{aligned}
[\![x := e]\!] &= \{(\omega, \nu) : \nu = \omega \text{ except } \nu[\![x]\!] = \omega[\![e]\!]\} \\
[\![?Q]\!] &= \{(\omega, \omega) : \omega \in [\![Q]\!]\} \\
[\![x' = f(x)]\!] &= \{(\varphi(0), \varphi(r)) : \varphi \models x' = f(x) \text{ for some duration } r\} \\
[\![\alpha \cup \beta]\!] &= [\![\alpha]\!] \cup [\![\beta]\!] \\
[\![\alpha; \beta]\!] &= [\![\alpha]\!] \circ [\![\beta]\!] \\
[\![\alpha^*]\!] &= [\![\alpha]\!]^* = \bigcup_{n \in \mathbb{N}} [\![\alpha^n]\!]
\end{aligned}
$$

compositional semantics

Definition (dL semantics) $([\![\cdot]\!] : \text{Fml} \to \wp(\mathscr{S}))$

$$
\begin{aligned}
[\![e \geq \tilde{e}]\!] &= \{\omega : \omega[\![e]\!] \geq \omega[\![\tilde{e}]\!]\} \\
[\![\neg P]\!] &= [\![P]\!]^{\complement} \\
[\![P \wedge Q]\!] &= [\![P]\!] \cap [\![Q]\!] \\
[\![\langle \alpha \rangle P]\!] &= [\![\alpha]\!] \circ [\![P]\!] = \{\omega : \nu \in [\![P]\!] \text{ for some } \nu : (\omega, \nu) \in [\![\alpha]\!]\} \\
[\![[\alpha] P]\!] &= [\![\neg\langle \alpha \rangle \neg P]\!] = \{\omega : \nu \in [\![P]\!] \text{ for all } \nu : (\omega, \nu) \in [\![\alpha]\!]\} \\
[\![\exists x\, P]\!] &= \{\omega : \omega^r_x \in [\![P]\!] \text{ for some } r \in \mathbb{R}\}
\end{aligned}
$$

Outline

Differential Dynamic Logic dL: Axiomatization

[:=] $[x := e]P(x) \leftrightarrow P(e)$

[?] $[?Q]P \leftrightarrow (Q \rightarrow P)$

[′] $[x' = f(x)]P \leftrightarrow \forall t{\geq}0\,[x := x(t)]P$ $\qquad (x'(t) = f(x))$

[∪] $[\alpha \cup \beta]P \leftrightarrow [\alpha]P \wedge [\beta]P$

[;] $[\alpha;\beta]P \leftrightarrow [\alpha][\beta]P$

[*] $[\alpha^*]P \leftrightarrow P \wedge [\alpha][\alpha^*]P$

K $[\alpha](P \rightarrow Q) \rightarrow ([\alpha]P \rightarrow [\alpha]Q)$

I $[\alpha^*]P \leftrightarrow P \wedge [\alpha^*](P \rightarrow [\alpha]P)$

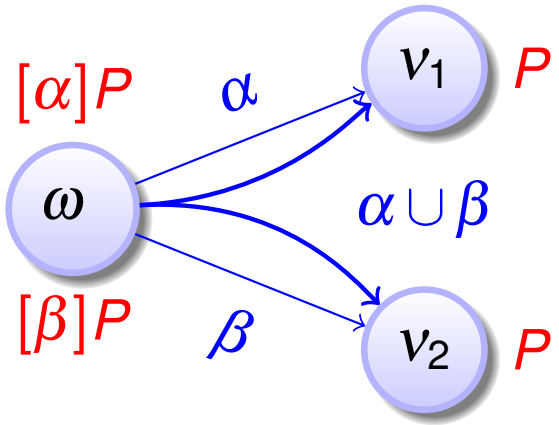
C $[\alpha^*]\forall v{>}0\,(P(v) \rightarrow \langle\alpha\rangle P(v{-}1)) \rightarrow \forall v\,(P(v) \rightarrow \langle\alpha^*\rangle\exists v{\leq}0\,P(v))$

equations of truth

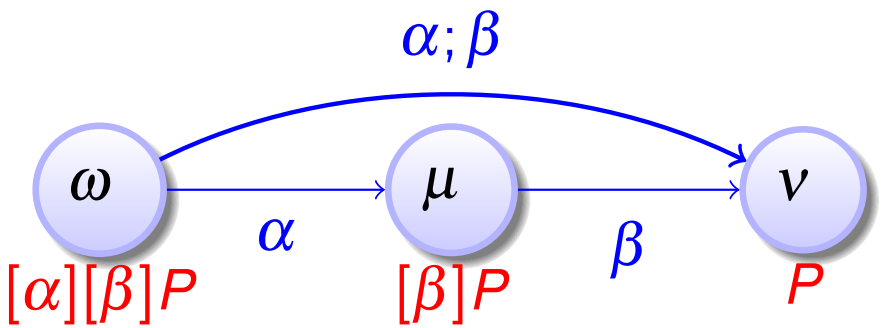
laws of logic of
laws of physics

LICS'12,JAR'17

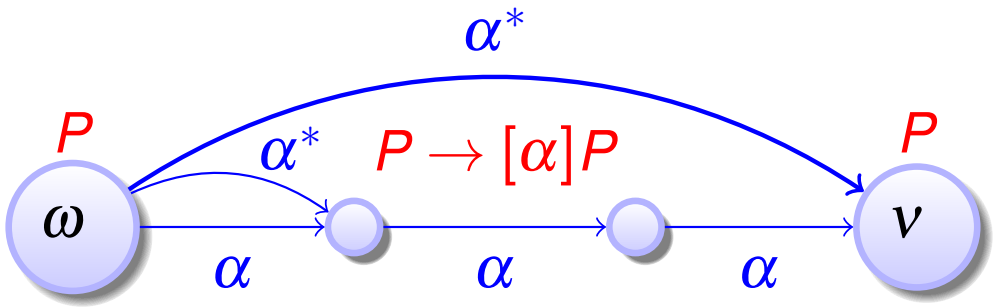
$$[\alpha \cup \beta]P \leftrightarrow [\alpha]P \wedge [\beta]P$$



$$[\alpha;\beta]P \leftrightarrow [\alpha][\beta]P$$



$$[\alpha^*]P \leftrightarrow P \wedge [\alpha^*](P \rightarrow [\alpha]P)$$



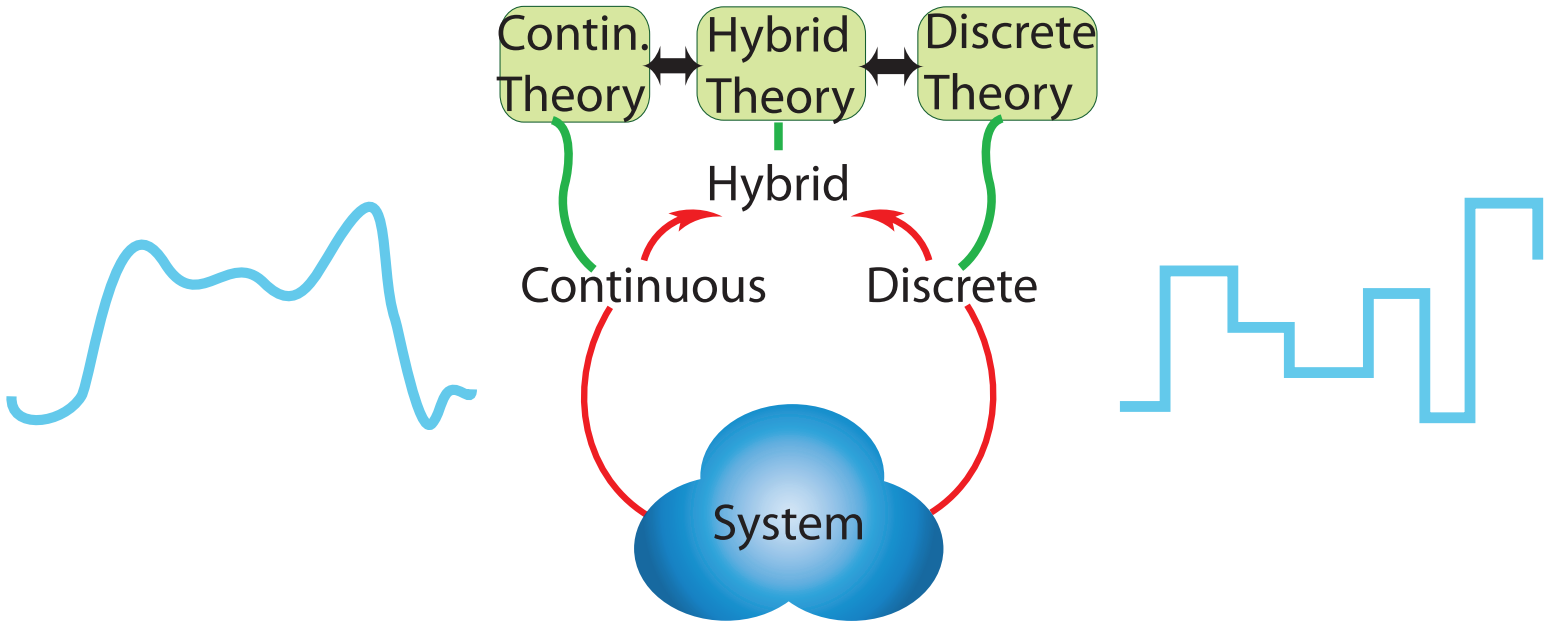
Complete Proof Theory of Hybrid Systems

Theorem (Sound & Complete) (JAR'08, LICS'12, JAR'17)

dL *calculus is a sound & complete axiomatization of hybrid systems relative to either differential equations* ***or*** *to discrete dynamics.*

Corollary (Complete Proof-theoretical Bridge)

proving continuous = proving hybrid = proving discrete

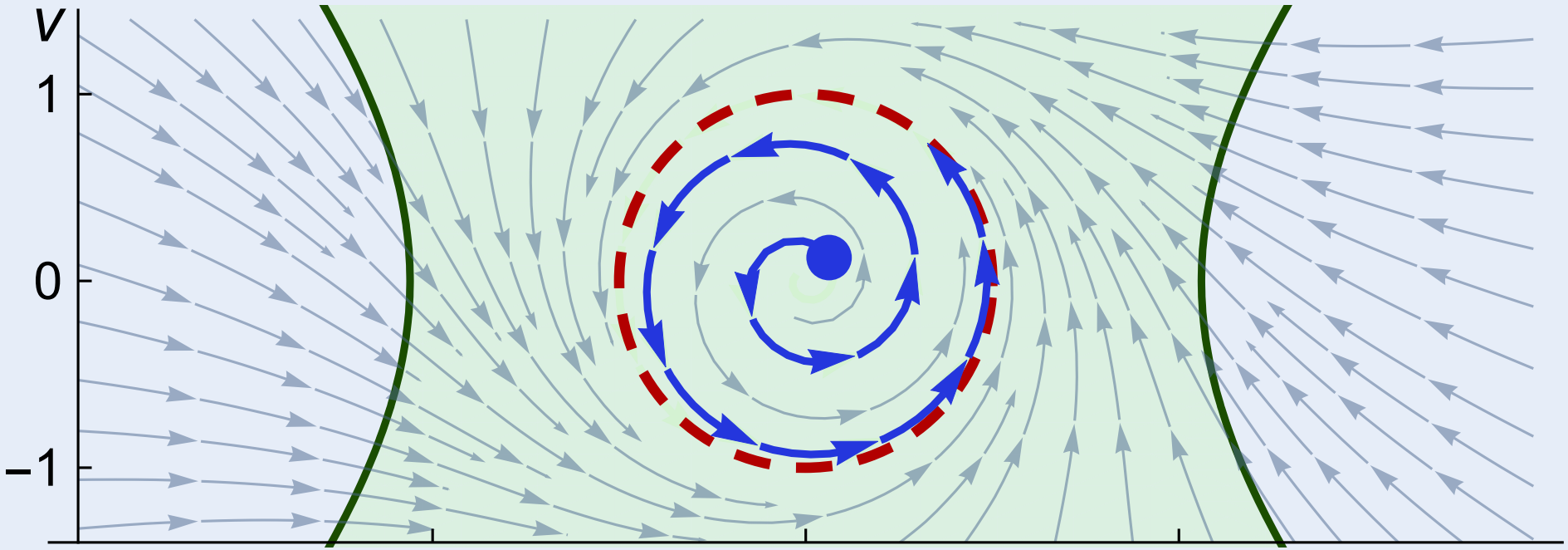


Outline

Concept (Differential Dynamic Logic) (JAR'08,LICS'12)

$$u^2 \leq v^2+\frac{9}{2} \rightarrow [u' = -v+\frac{u}{4}(1-u^2-v^2), v' = u+\frac{v}{4}(1-u^2-v^2)]\, u^2 \leq v^2+\frac{9}{2}$$

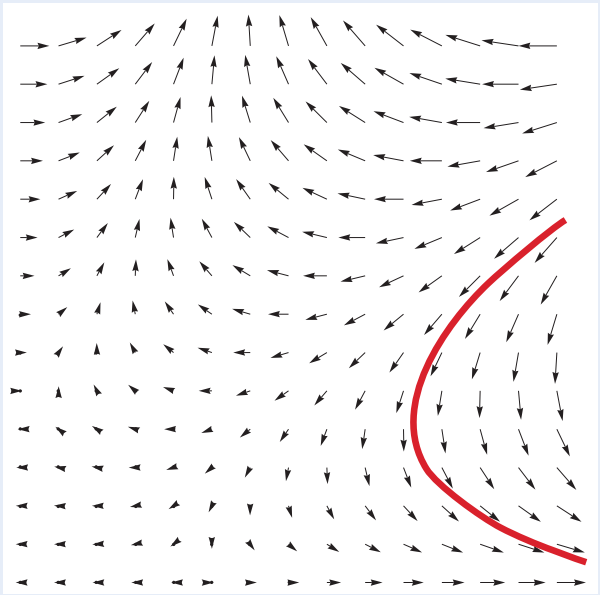
$$u^2+v^2 = 1 \rightarrow [u' = -v+\frac{u}{4}(1-u^2-v^2), v' = u+\frac{v}{4}(1-u^2-v^2)]\, u^2+v^2 = 1$$

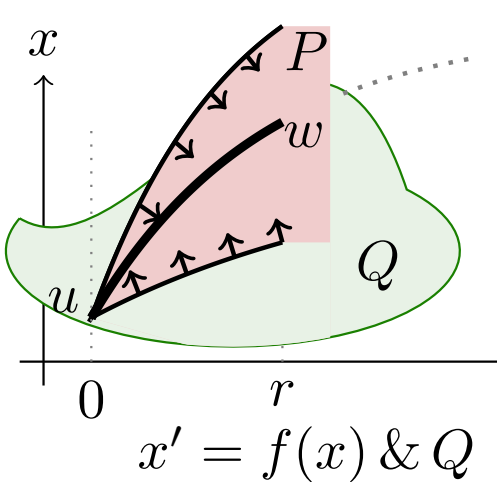


Analyzing ODEs via solutions undoes their descriptive power! Poincaré 1881

Differential Invariants for Differential Equations

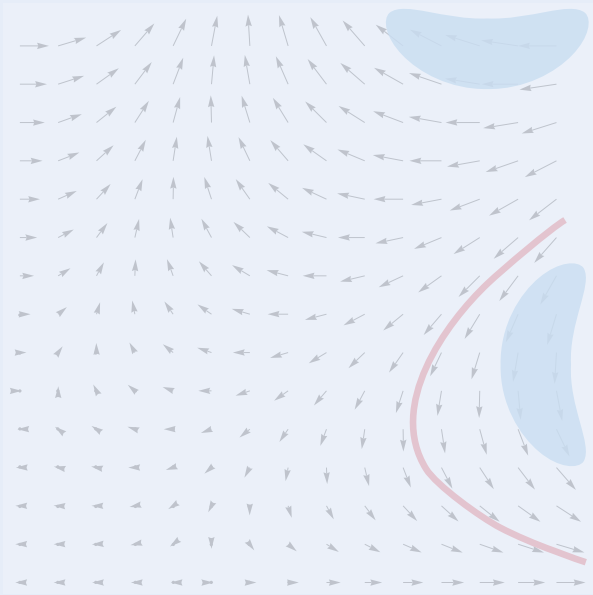
Differential Invariant

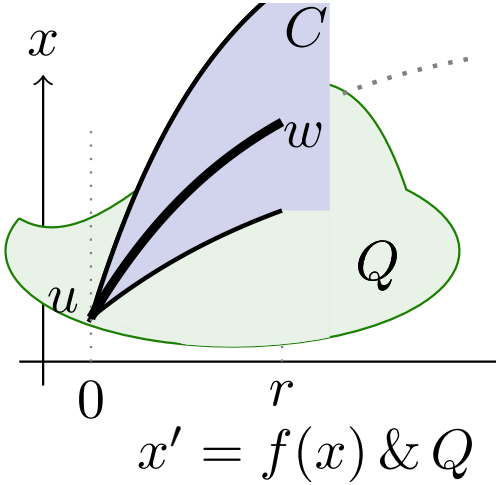




$$x' = f(x) \,\&\, Q$$

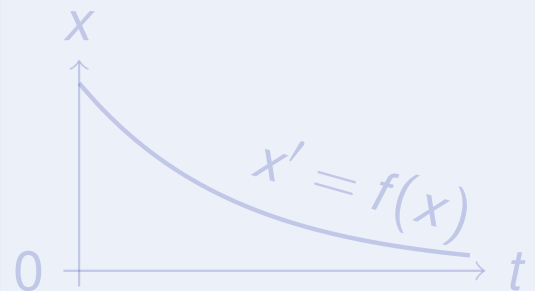
Differential Cut

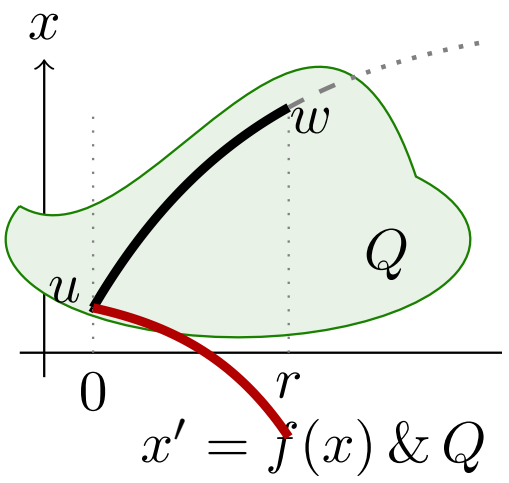




$$x' = f(x) \,\&\, Q$$

Differential Ghost

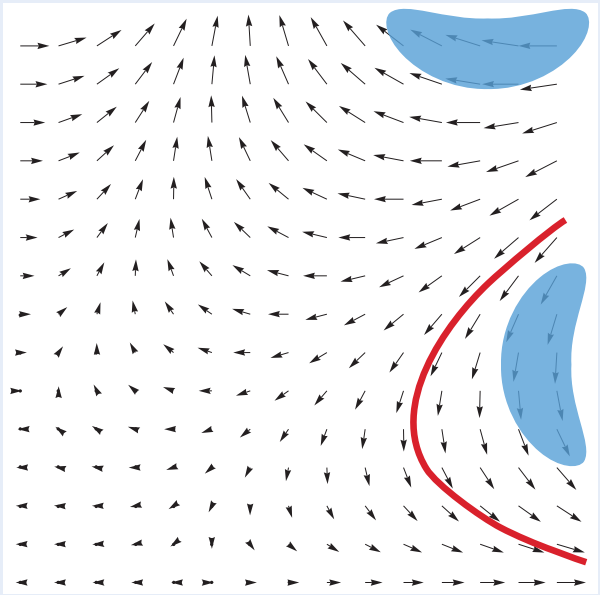


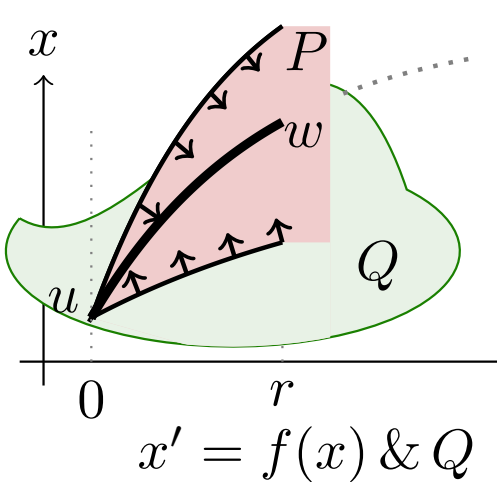


$$x' = f(x) \,\&\, Q$$

Differential Invariants for Differential Equations

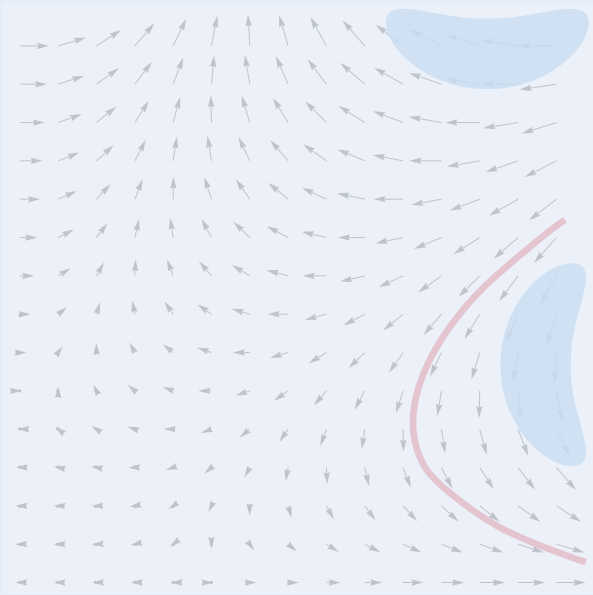
Differential Invariant

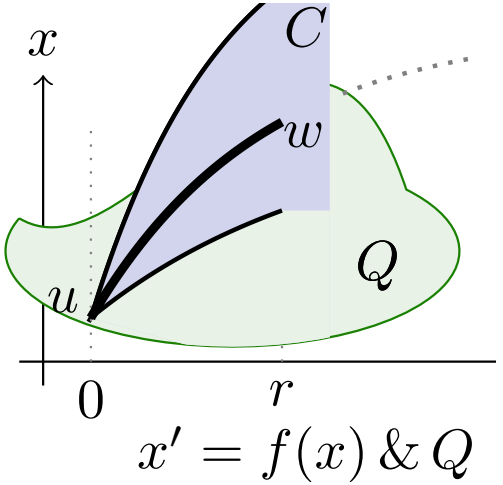




$$x' = f(x) \,\&\, Q$$

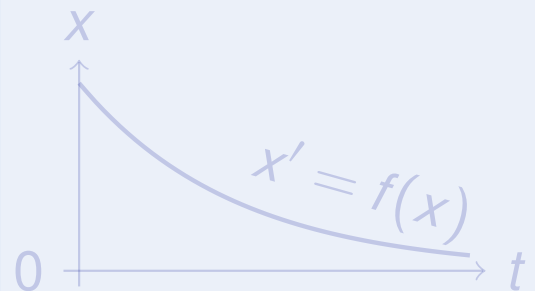
Differential Cut

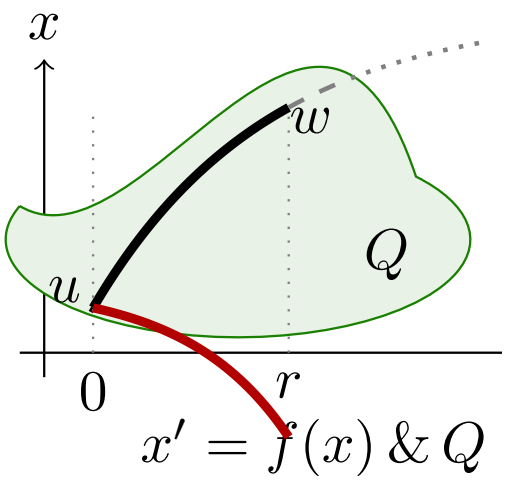




$$x' = f(x) \,\&\, Q$$

Differential Ghost

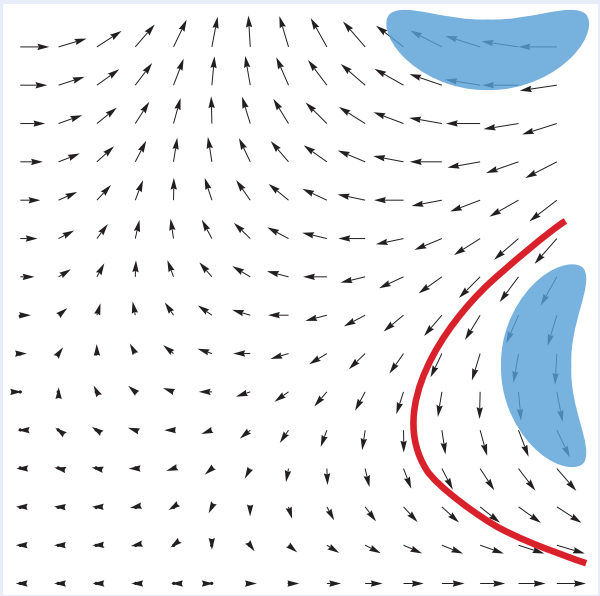




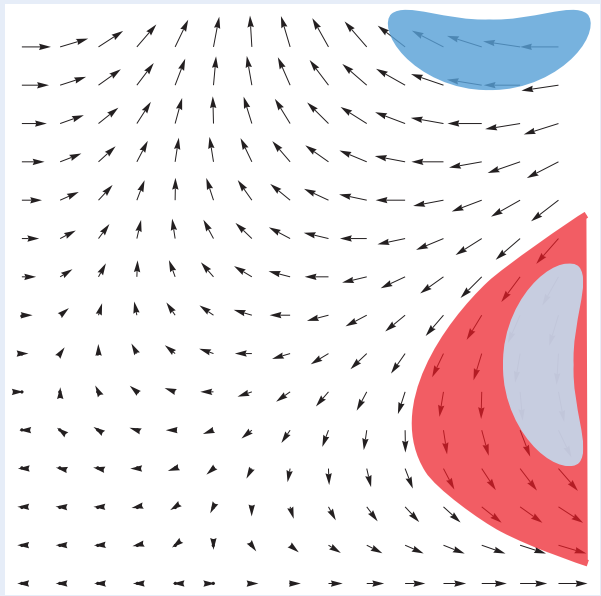
$$x' = f(x) \,\&\, Q$$

Differential Invariants for Differential Equations

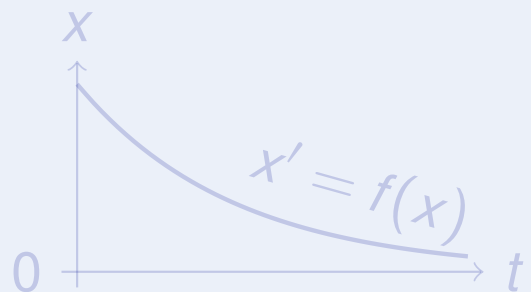
Differential Invariant

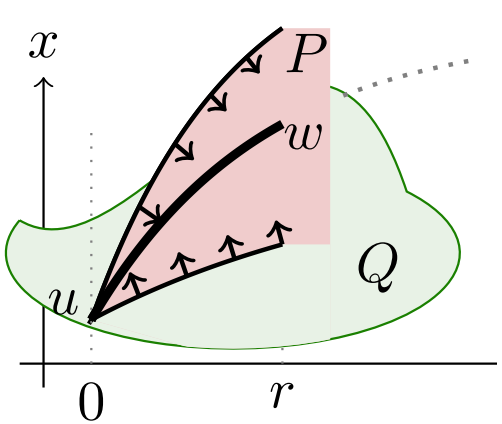


Differential Cut

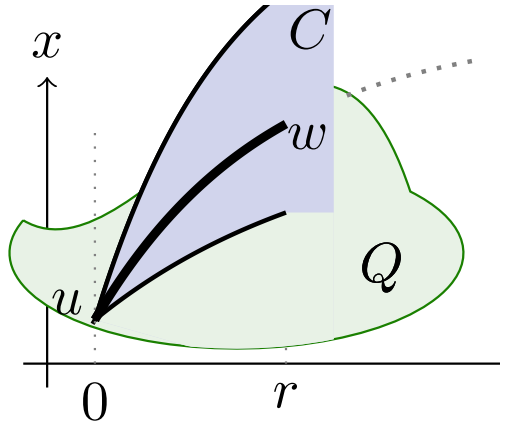


Differential Ghost

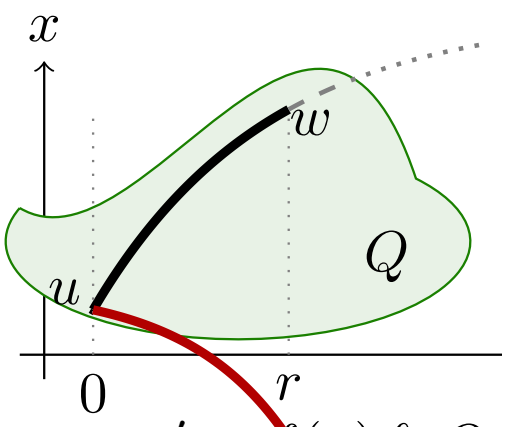




$$x' = f(x) \,\&\, Q$$



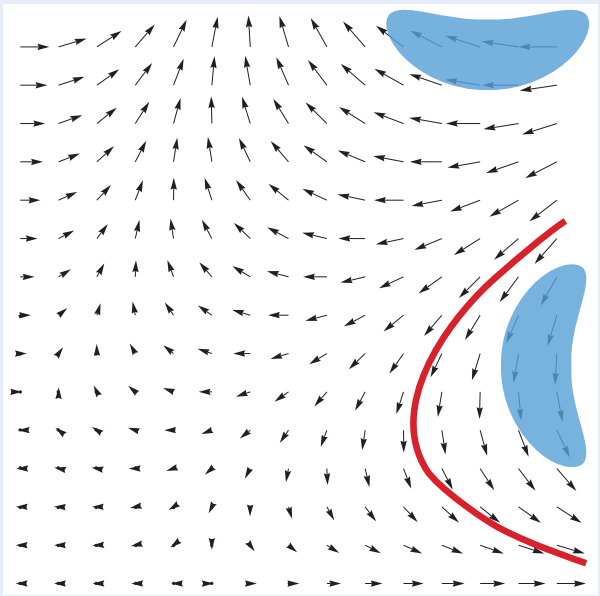
$$x' = f(x) \,\&\, Q$$



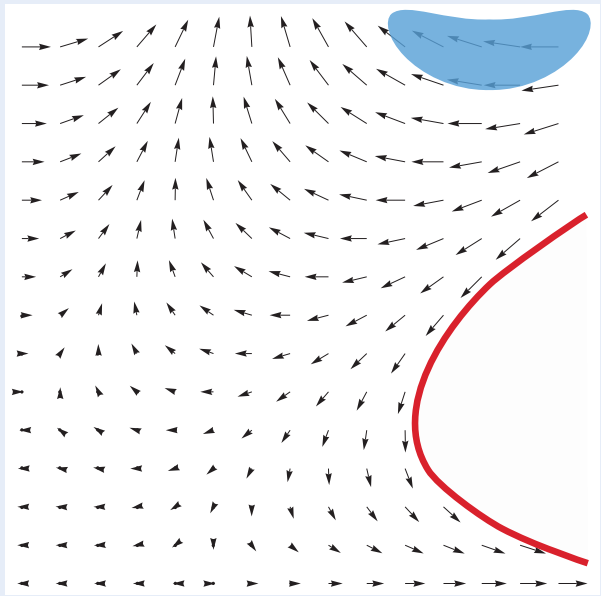
$$x' = f(x) \,\&\, Q$$

Differential Invariants for Differential Equations

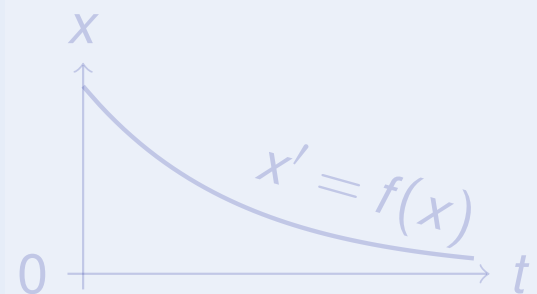
Differential Invariant

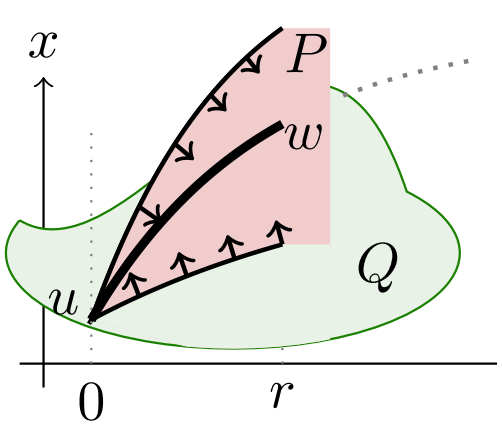


Differential Cut

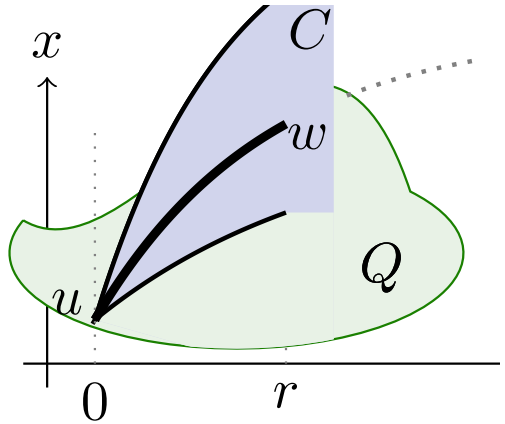


Differential Ghost

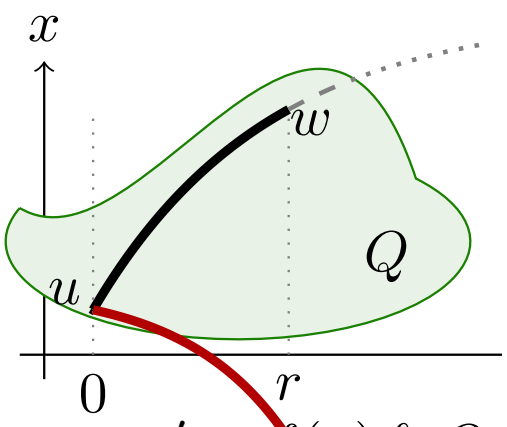




$x' = f(x) \,\&\, Q$



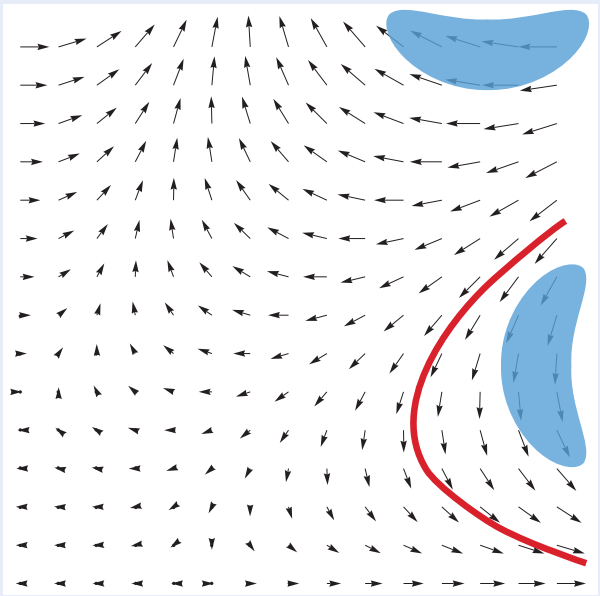
$x' = f(x) \,\&\, Q$



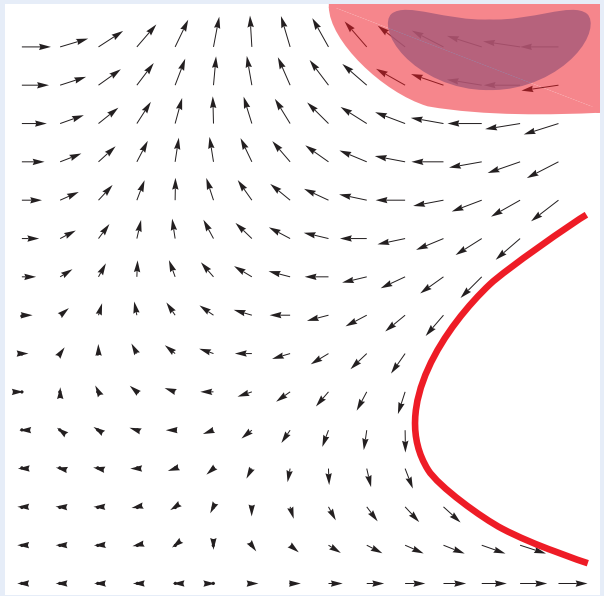
$x' = f(x) \,\&\, Q$

Differential Invariants for Differential Equations

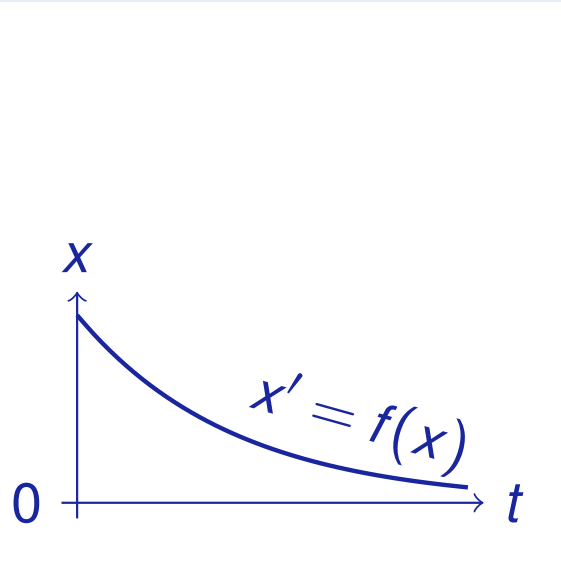
Differential Invariant

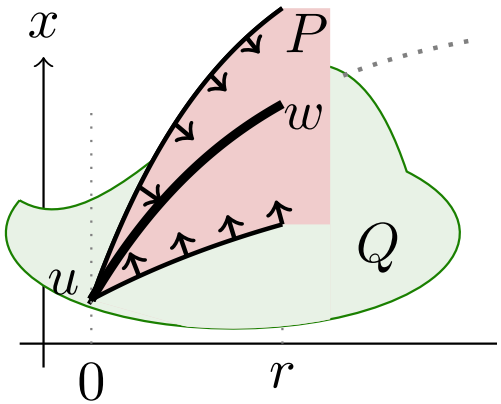


Differential Cut

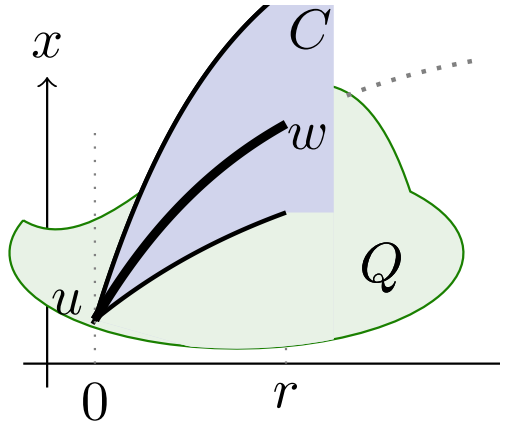


Differential Ghost

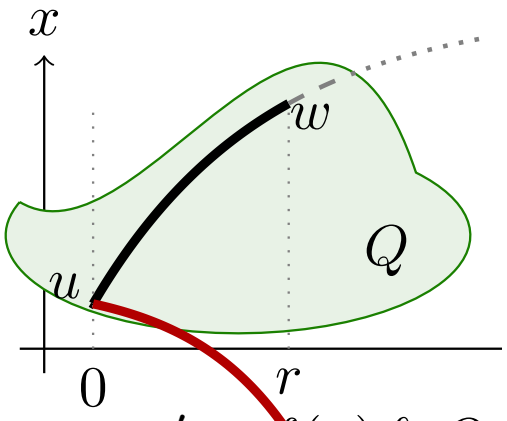




$x' = f(x) \,\&\, Q$



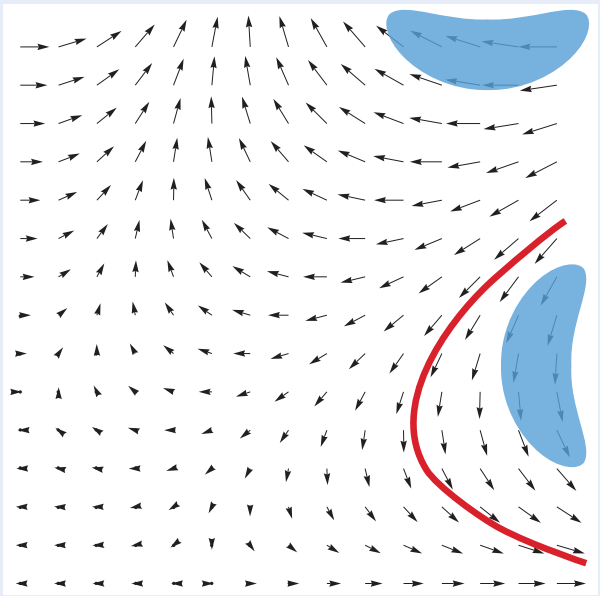
$x' = f(x) \,\&\, Q$



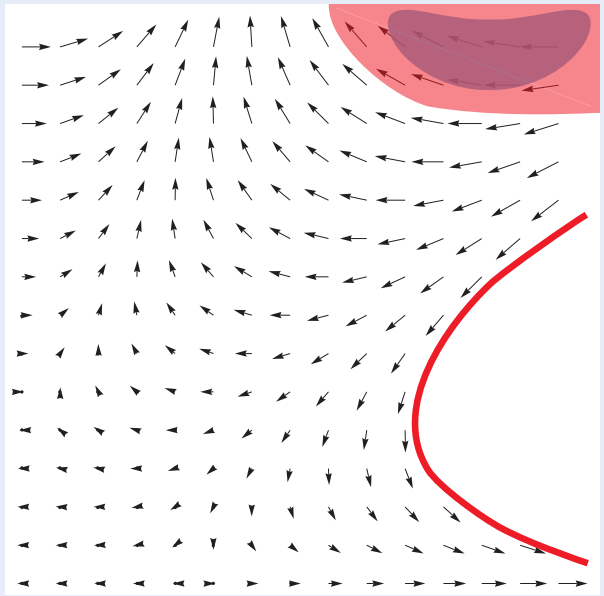
$x' = f(x) \,\&\, Q$

Differential Invariants for Differential Equations

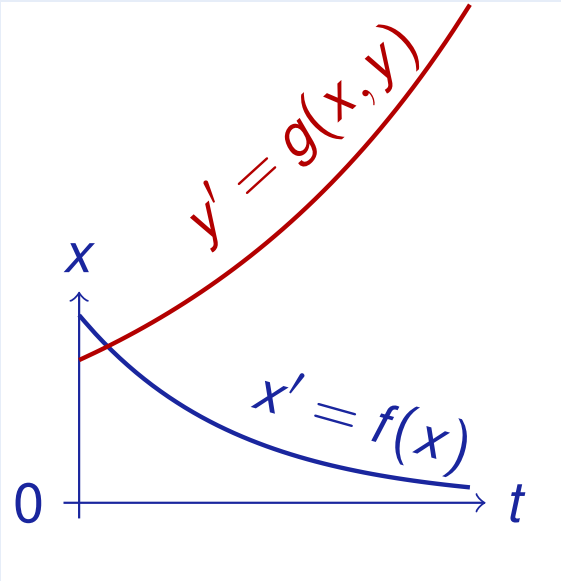
Differential Invariant

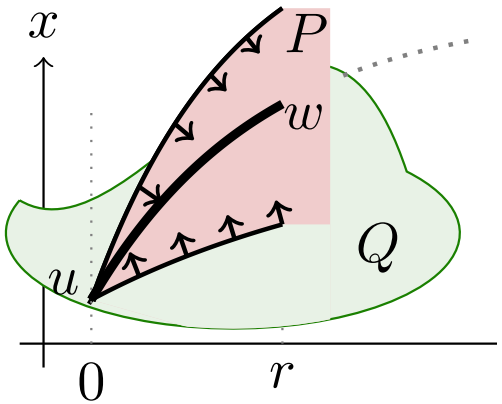


Differential Cut

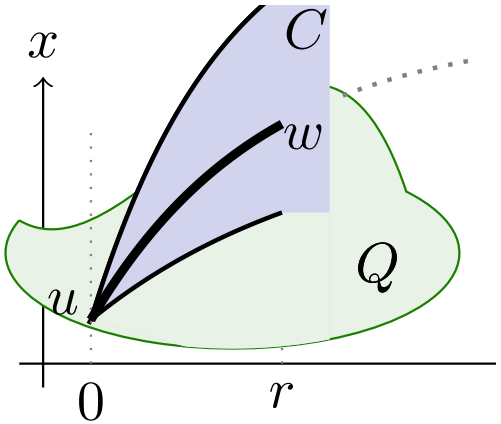


Differential Ghost

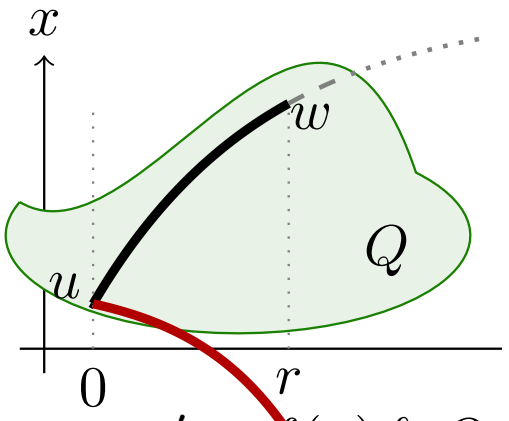




$x' = f(x) \,\&\, Q$



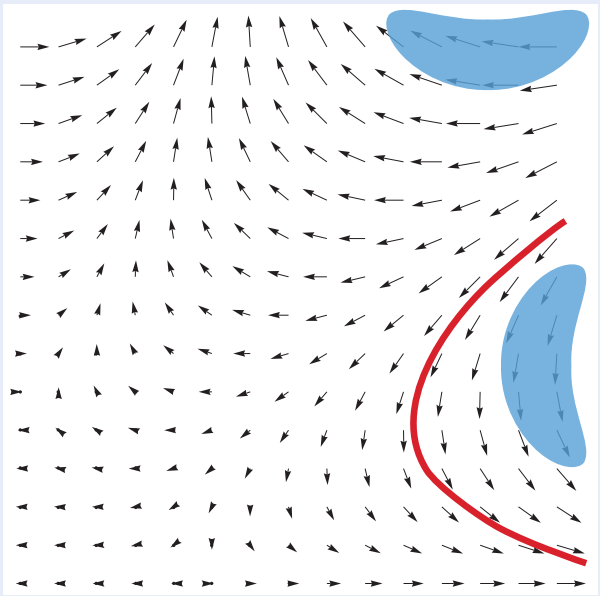
$x' = f(x) \,\&\, Q$



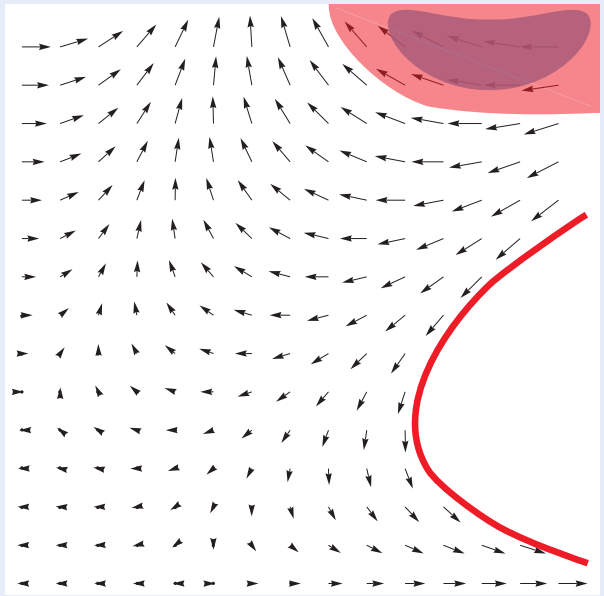
$x' = f(x) \,\&\, Q$

Differential Invariants for Differential Equations

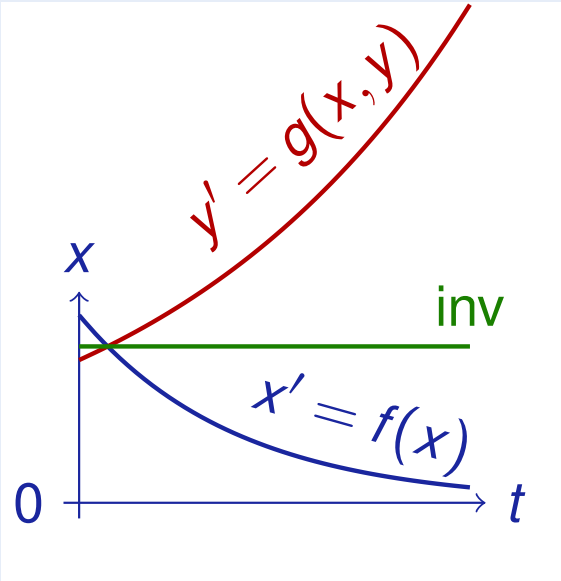
Differential Invariant

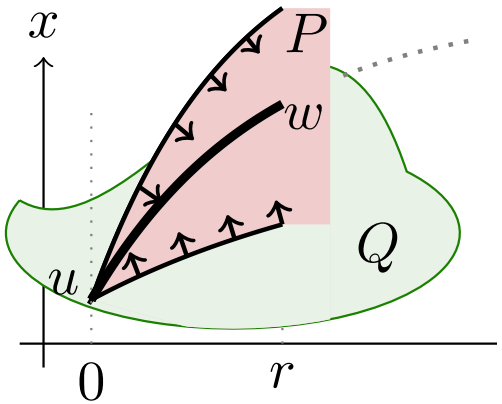


Differential Cut

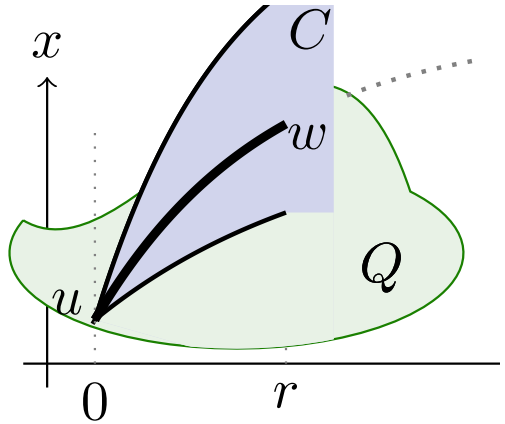


Differential Ghost

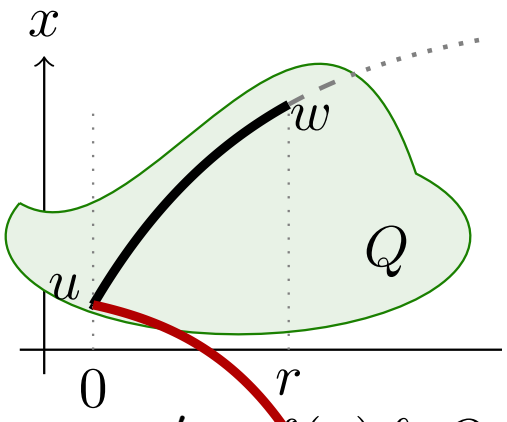




$x' = f(x) \,\&\, Q$



$x' = f(x) \,\&\, Q$



$x' = f(x) \,\&\, Q$

Differential Invariants for Differential Equations

Differential Invariant

$$\frac{Q \vdash [x' := f(x)](P)'}{P \vdash [x' = f(x) \,\&\, Q]P}$$

Differential Cut

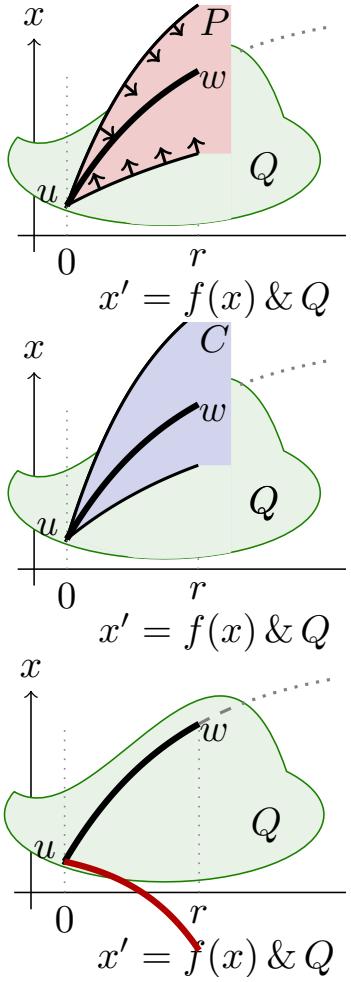
$$\frac{P \vdash [x' = f(x) \,\&\, Q]C \quad P \vdash [x' = f(x) \,\&\, Q \wedge C]P}{P \vdash [x' = f(x) \,\&\, Q]P}$$

Differential Ghost

$$\frac{P \leftrightarrow \exists y\, G \quad G \vdash [x' = f(x), y' = g(x,y) \,\&\, Q]G}{P \vdash [x' = f(x) \,\&\, Q]P}$$

deductive power added DI $\prec$ DI+DC $\prec$ DI+DC+DG

$$u[\![(e)']\!] = \sum_x u(x') \frac{\partial [\![e]\!]}{\partial x}(u)$$



Springer'10, LMCS'12, LICS'12,JAR'17,LICS'18,JACM'20

Differential Invariants for Differential Equations

Differential Invariant

$$\frac{Q \vdash [x' := f(x)](P)'}{P \vdash [x' = f(x) \,\&\, Q]P}$$

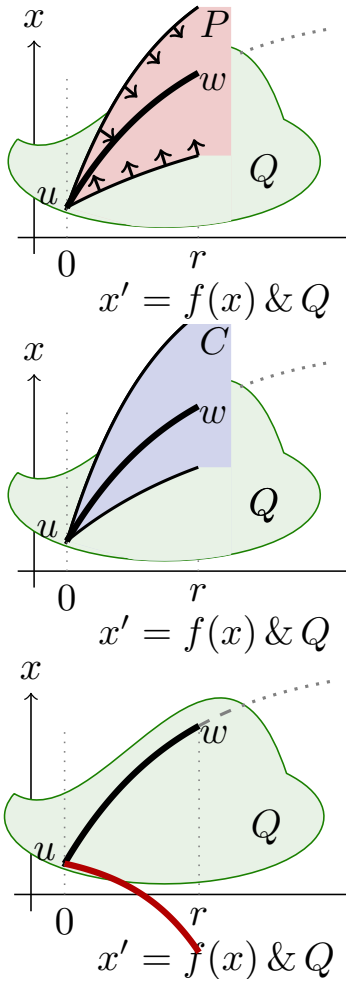
Differential Cut

$$\frac{P \vdash [x' = f(x) \,\&\, Q]C \quad P \vdash [x' = f(x) \,\&\, Q \wedge C]P}{P \vdash [x' = f(x) \,\&\, Q]P}$$

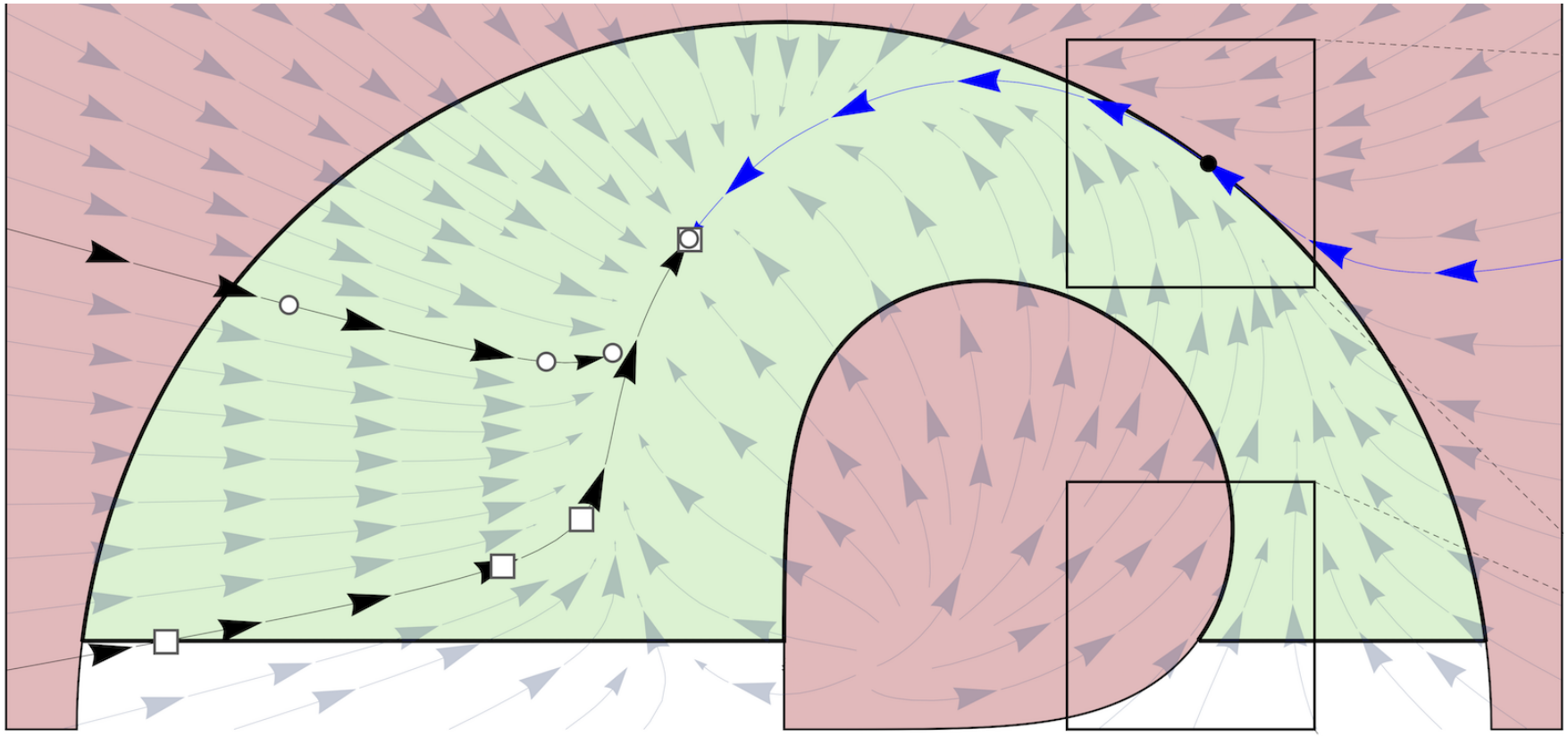
Differential Ghost

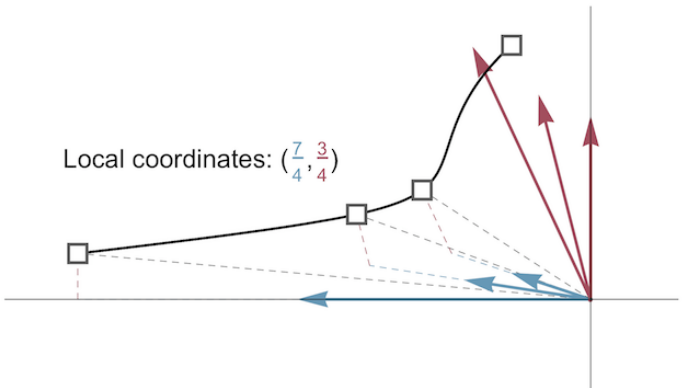
$$\frac{P \leftrightarrow \exists y\, G \quad G \vdash [x' = f(x), y' = g(x,y) \,\&\, Q]G}{P \vdash [x' = f(x) \,\&\, Q]P}$$

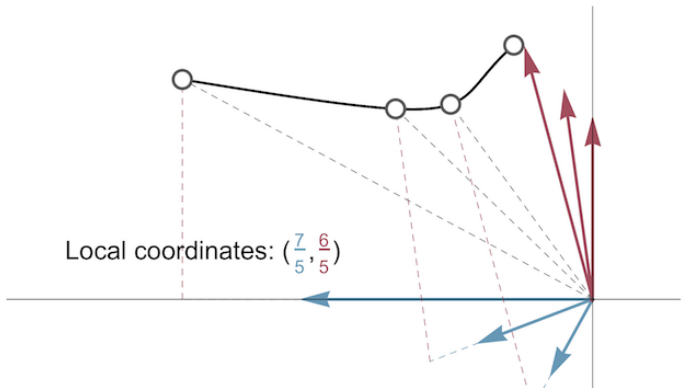
if $g(x,y) = a(x)y + b(x)$, so has long solution!



Springer'10, LMCS'12, LICS'12,JAR'17,LICS'18,JACM'20







LICS'18,JACM'20

Differential Equation Axiomatization

Theorem (Algebraic Completeness) (LICS'18,JACM'20)

dL *calculus is a sound & complete axiomatization of algebraic invariants of polynomial differential equations. They are decidable by* DI,DC,DG *in* dL*.*

Theorem (Semialgebraic Completeness) (LICS'18,JACM'20)

dL *calculus with RI is a sound & complete axiomatization of semialgebraic invariants of differential equations. They are decidable in* dL*.*

Differential Equation Axiomatization

Theorem (Algebraic Completeness) (LICS'18,JACM'20)

dL *calculus is a sound & complete axiomatization of algebraic invariants of polynomial differential equations. They are decidable*

$$\text{(DRI)}\quad [x' = f(x)\,\&\,Q]e = 0 \leftrightarrow \big(Q \rightarrow e'^{*} = 0\big) \qquad (Q \text{ open})$$

Theorem (Semialgebraic Completeness) (LICS'18,JACM'20)

dL *calculus with RI is a sound & complete axiomatization of semialgebraic invariants of differential equations. They are decidable*

$$\text{(SAI)}\quad \forall x\,(P \rightarrow [x' = f(x)]P) \leftrightarrow \forall x\,\big(P \rightarrow P'^{*}\big) \wedge \forall x\,\big(\neg P \rightarrow (\neg P)'^{*-}\big)$$

Definable e'^{*} is short for *all/significant* Lie derivative w.r.t. ODE
Definable e'^{*-} is w.r.t. backwards ODE $x' = -f(x)$. Also for P.

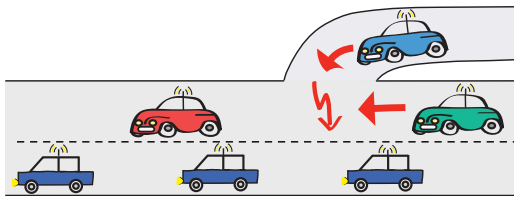
$$e'^{*} = 0 \equiv e{=}0 \wedge (e')'^{*}{=}0 \qquad (P \wedge Q)'^{*} \equiv P'^{*} \wedge Q'^{*}$$
$$e'^{*} \geq 0 \equiv e{\geq}0 \wedge (e{=}0 \rightarrow (e')'^{*}{\geq}0) \qquad (P \vee Q)'^{*} \equiv P'^{*} \vee Q'^{*}$$

Outline

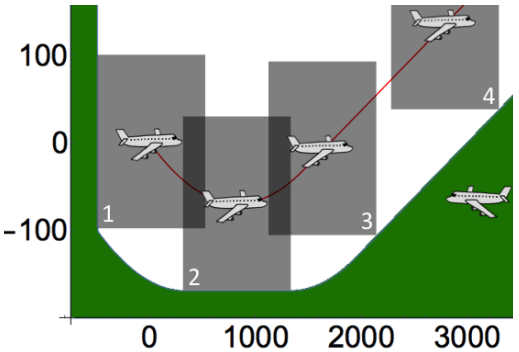
Applications with Impact

Prospects: Safety & Efficiency

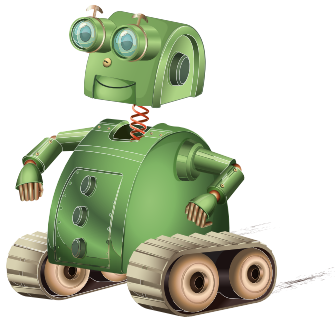
(Autonomous) cars



(Auto)Pilot support



Robots near humans

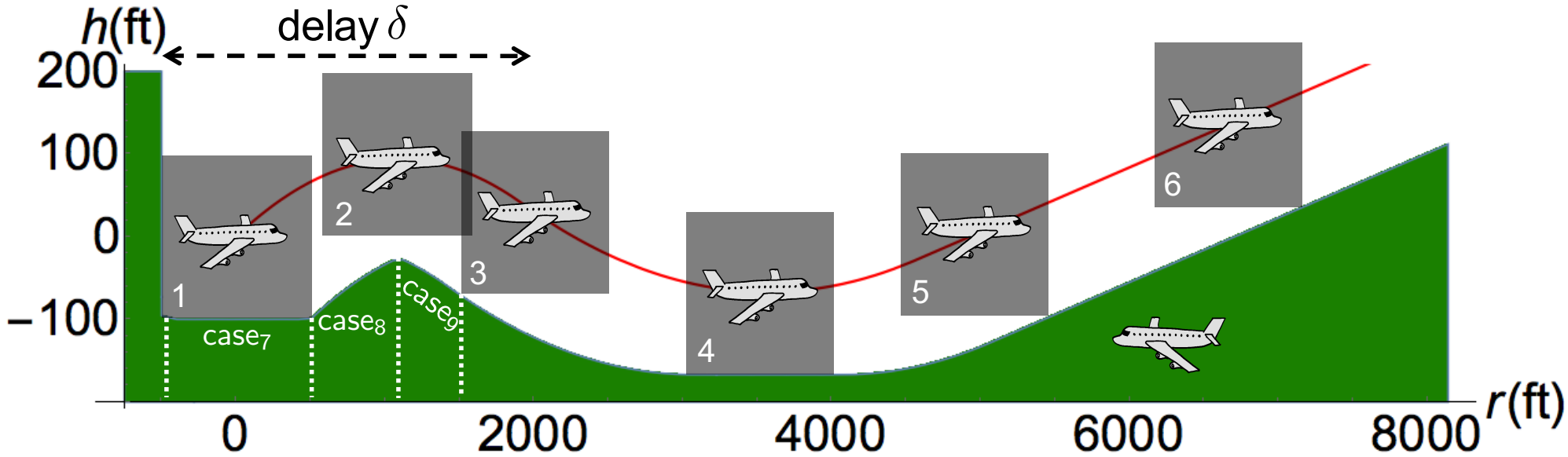


Cyber-Physical Systems

CPSs combine cyber capabilities with physical capabilities
to solve problems that neither part could solve alone.

Airborne Collision Avoidance System ACAS X: Verify

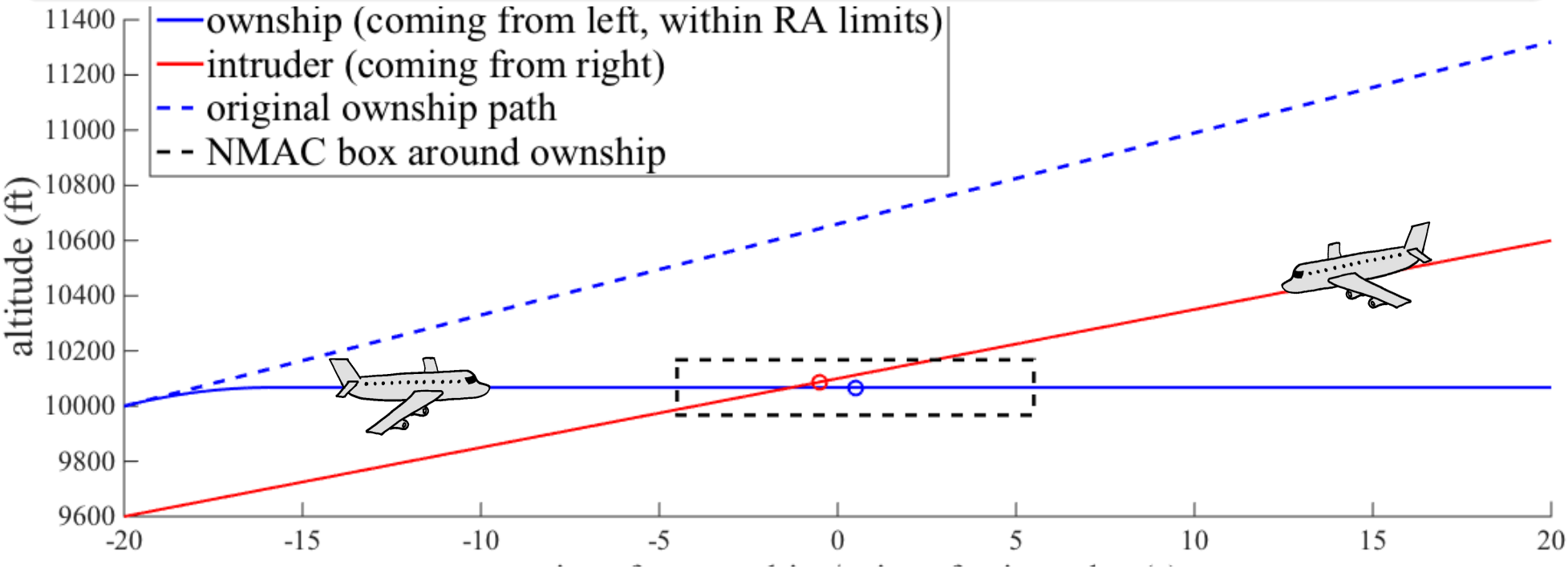
- Developed by the FAA to replace current TCAS in aircraft
- Approximately optimizes Markov Decision Process on a grid
- Advisory from lookup tables with numerous 5D interpolation regions



1. Identified safe region for each advisory symbolically
2. Proved safety for hybrid systems flight model in KeYmaera X

STTT'17

ACAS X table comparison shows safe advisory in 97.7% of the 648,591,384,375 states compared (15,160,434,734 counterexamples).

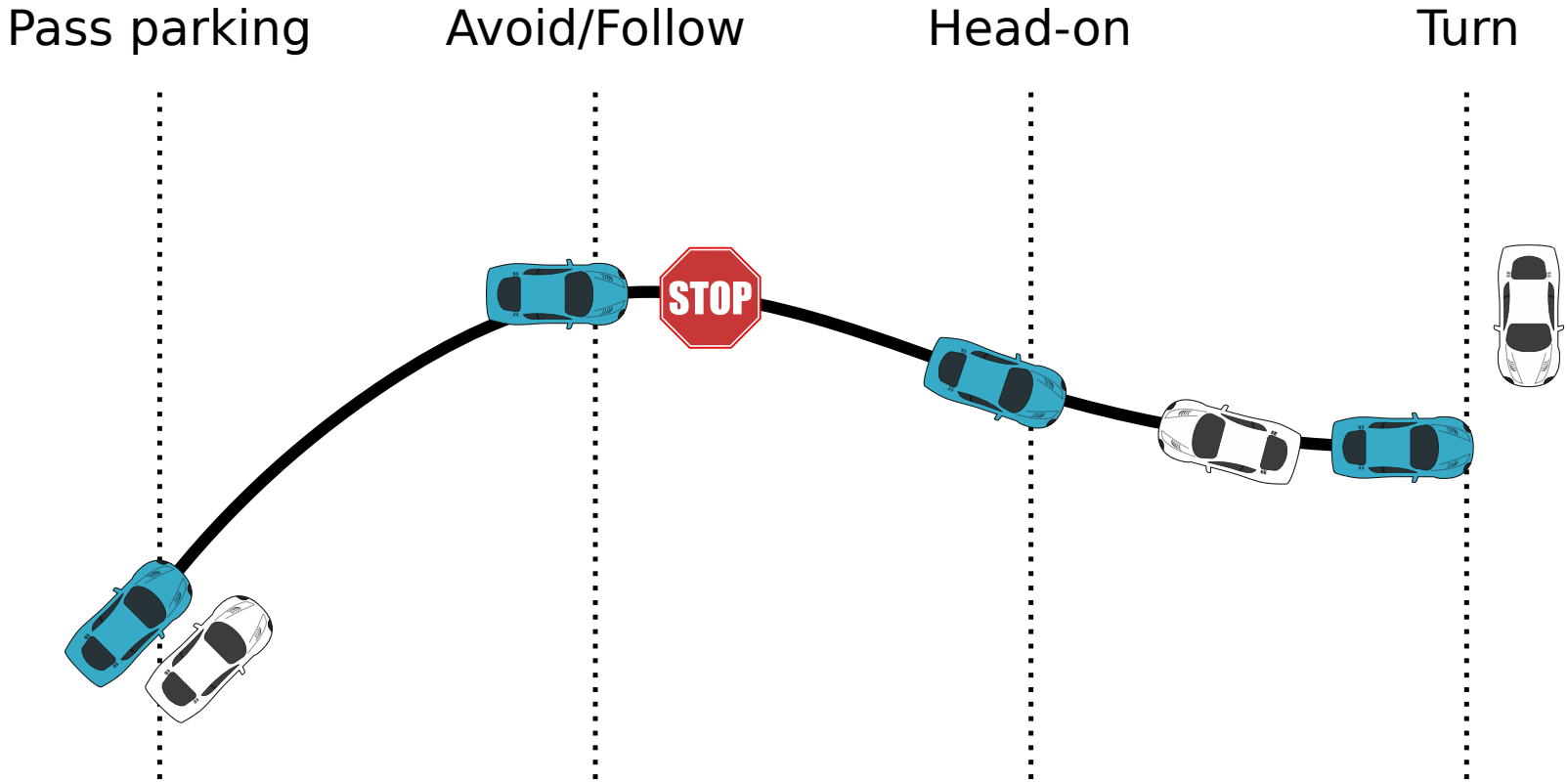


ACAS X issues DNC advisory, which induces collision unless corrected

STTT'17

Ground Robot Obstacle Avoidance: Verify

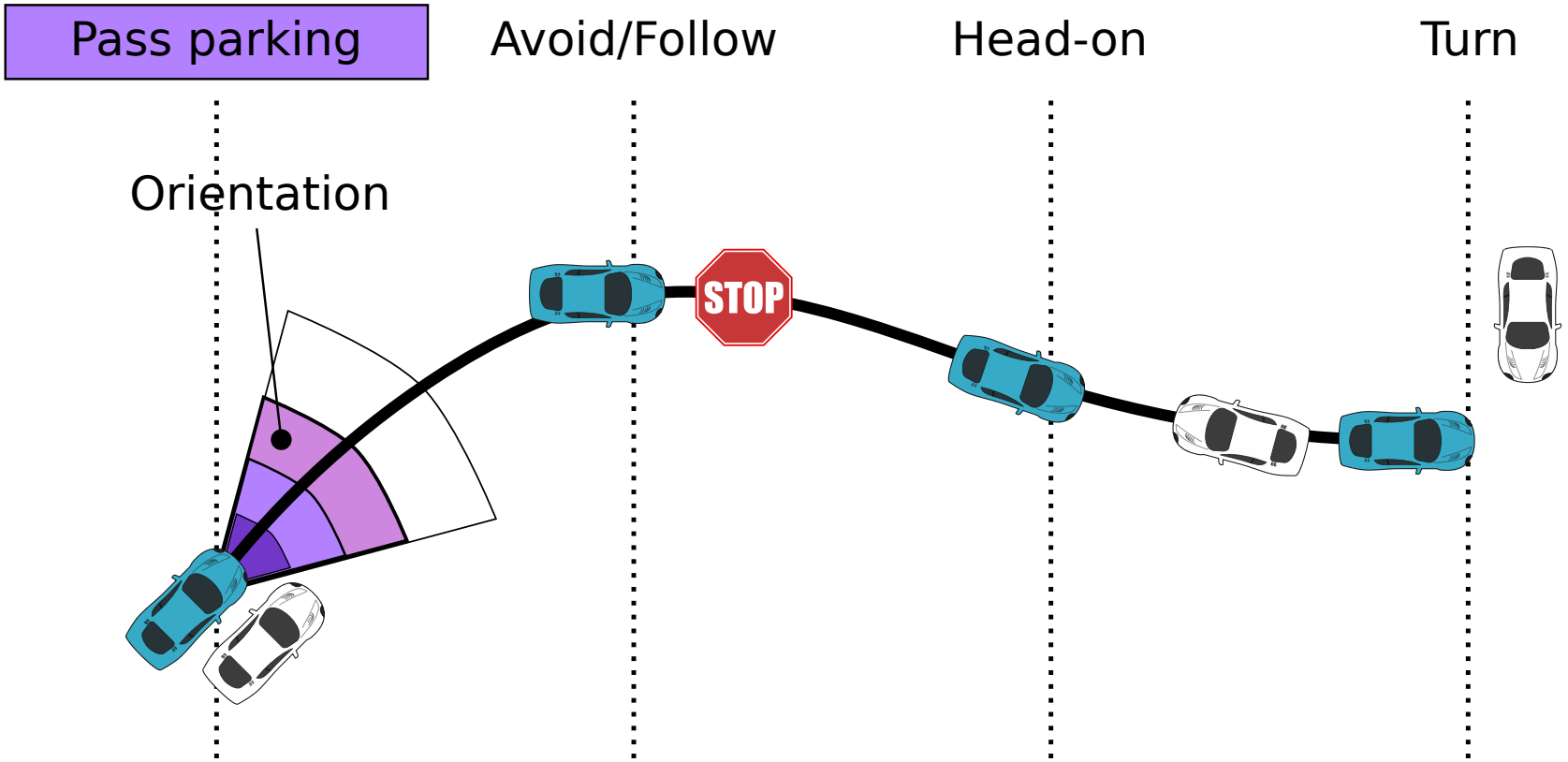
- Fundamental safety question for ground robot navigation IJRR'17
- When will which control decision avoid obstacles?
- Depends on safety objective, physical capabilities of robot + obstacle



1. Identified safe region for each safety notion symbolically
2. Proved safety for hybrid systems ground robot model in KeYmaera X

Ground Robot Obstacle Avoidance: Verify

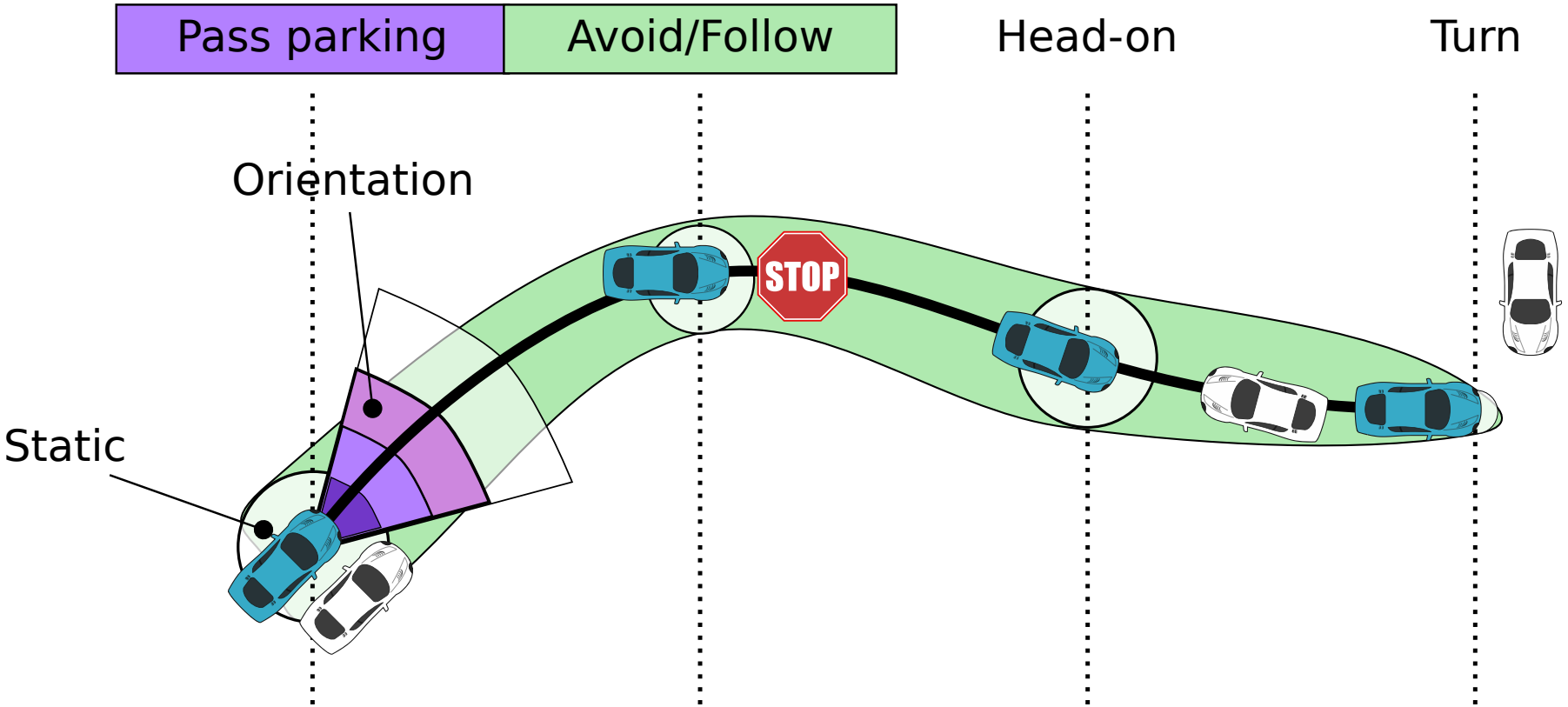
- Fundamental safety question for ground robot navigation IJRR'17
- When will which control decision avoid obstacles?
- Depends on safety objective, physical capabilities of robot + obstacle



1. Identified safe region for each safety notion symbolically
2. Proved safety for hybrid systems ground robot model in KeYmaera X

Ground Robot Obstacle Avoidance: Verify

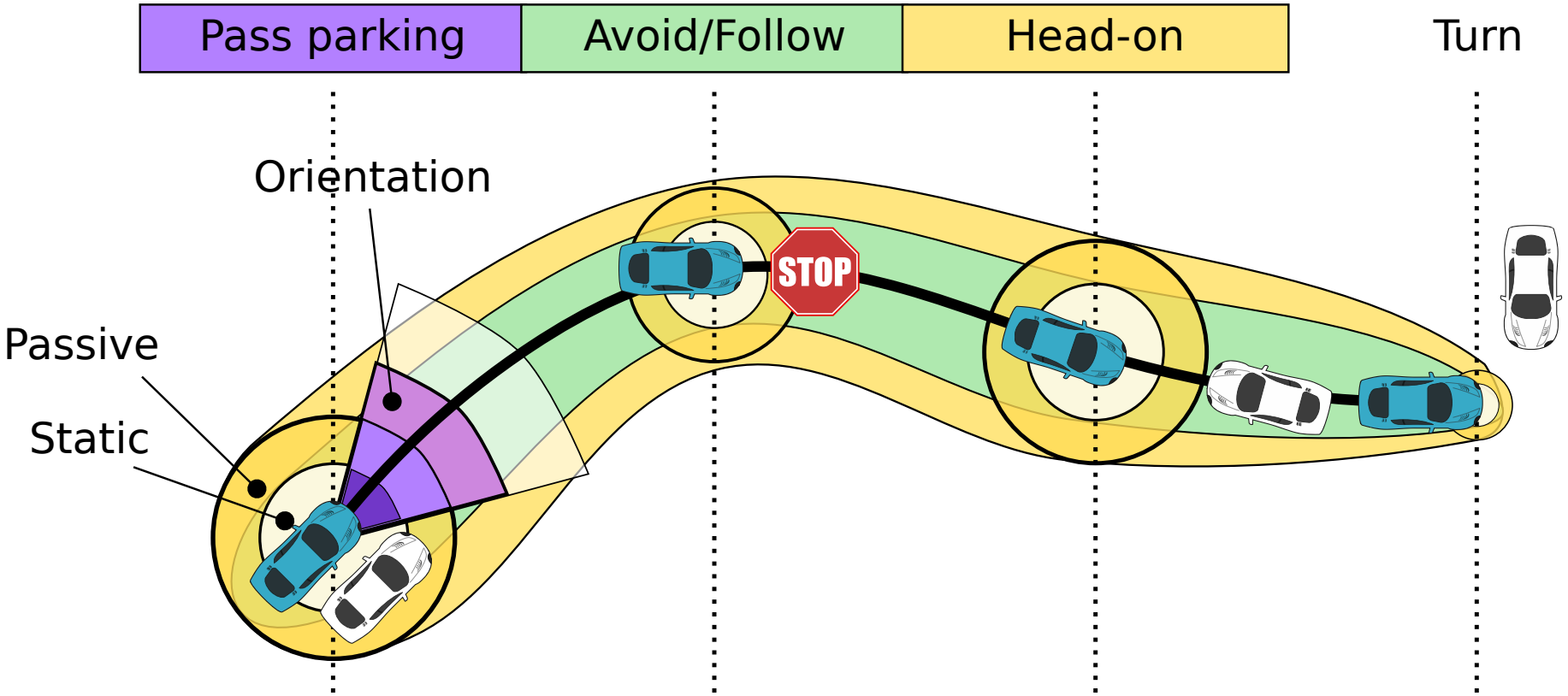
- Fundamental safety question for ground robot navigation IJRR'17
- When will which control decision avoid obstacles?
- Depends on safety objective, physical capabilities of robot + obstacle



1. Identified safe region for each safety notion symbolically
2. Proved safety for hybrid systems ground robot model in KeYmaera X

Ground Robot Obstacle Avoidance: Verify

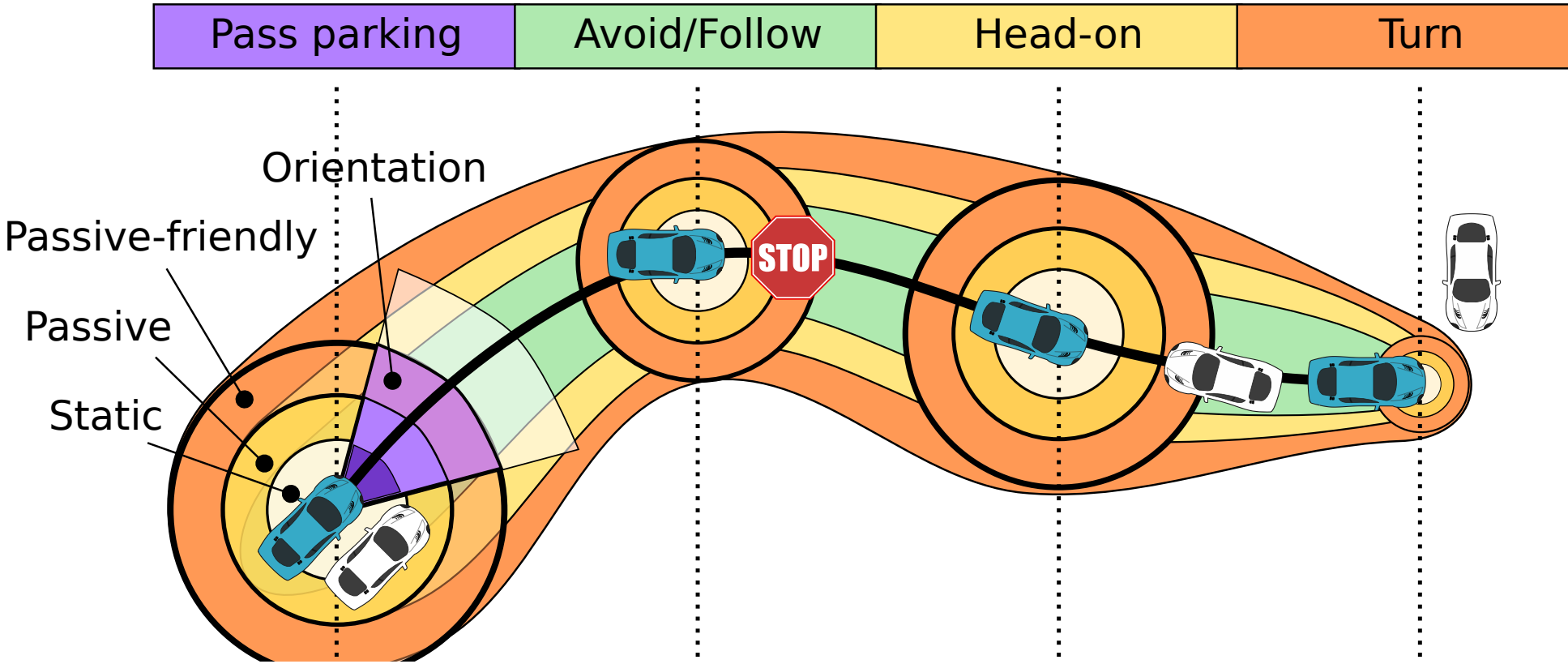
- Fundamental safety question for ground robot navigation IJRR'17
- When will which control decision avoid obstacles?
- Depends on safety objective, physical capabilities of robot + obstacle



1. Identified safe region for each safety notion symbolically
2. Proved safety for hybrid systems ground robot model in KeYmaera X

Ground Robot Obstacle Avoidance: Verify

- Fundamental safety question for ground robot navigation IJRR'17
- When will which control decision avoid obstacles?
- Depends on safety objective, physical capabilities of robot + obstacle



1. Identified safe region for each safety notion symbolically
2. Proved safety for hybrid systems ground robot model in KeYmaera X

Ground Robot Invariants and Safe Control Constraints

Safety	Invariant + Safe Control
static	$\|p-o\|_\infty > \frac{s^2}{2b} + \left(\frac{A}{b}+1\right)\left(\frac{A}{2}\varepsilon^2+\varepsilon s\right)$
passive	$s \neq 0 \to \|p-o\|_\infty > \frac{s^2}{2b} + V\frac{s}{b} + \left(\frac{A}{b}+1\right)\left(\frac{A}{2}\varepsilon^2+\varepsilon(s+V)\right)$
+ sensor	$\|\hat{p}-o\|_\infty > \frac{s^2}{2b} + V\frac{s}{b} + \left(\frac{A}{b}+1\right)\left(\frac{A}{2}\varepsilon^2+\varepsilon(s+V)\right) + \Delta_p$
+ disturb.	$\|p-o\|_\infty > \frac{s^2}{2b\Delta_a} + V\frac{s}{b\Delta_a} + \left(\frac{A}{b\Delta_a}+1\right)\left(\frac{A}{2}\varepsilon^2+\varepsilon(s+V)\right)$
+ failure	$\|\hat{p}-o\|_\infty > \frac{s^2}{2b} + V\frac{s}{b} + \left(\frac{A}{b}+1\right)\left(\frac{A}{2}\varepsilon^2+\varepsilon(v+V)\right) + \Delta_p + g\Delta$
friendly	$\|p-o\|_\infty > \frac{s^2}{2b} + \frac{V^2}{2b_o} + V\left(\frac{s}{b}+\tau\right) + \left(\frac{A}{b}+1\right)\left(\frac{A}{2}\varepsilon^2+\varepsilon(s+V)\right)$

IJRR'17

Ground Robot Invariants and Safe Control Constraints

Safety	Invariant + Safe Control
static	$\|p-o\|_\infty > \frac{s^2}{2b} + \left(\frac{A}{b}+1\right)\left(\frac{A}{2}\varepsilon^2+\varepsilon s\right)$
passive	$s \neq 0 \rightarrow \|p-o\|_\infty > \frac{s^2}{2b} + V\frac{s}{b} + \left(\frac{A}{b}+1\right)\left(\frac{A}{2}\varepsilon^2+\varepsilon(s+V)\right)$
+ sensor	$\|p-o\|_\infty > \frac{s^2}{2b} + V\frac{s}{b} + \left(\frac{A}{b}+1\right)\left(\frac{A}{2}\varepsilon^2+\varepsilon(s+V)\right) + \Delta_p$
+ disturb.	$\|p-o\|_\infty > \frac{s^2}{2b\Delta_a} + V\frac{s}{b\Delta_a} + \left(\frac{A}{b\Delta_a}+1\right)\left(\frac{A}{2}\varepsilon^2+\varepsilon(s+V)\right)$
+ failure	$\|\hat{p}-o\|_\infty > \frac{s^2}{2b} + V\frac{s}{b} + \left(\frac{A}{b}+1\right)\left(\frac{A}{2}\varepsilon^2+\varepsilon(v+V)\right) + \Delta_p + g\Delta$
friendly	$\|p-o\|_\infty > \frac{s^2}{2b} + \frac{V^2}{2b_o} + V\left(\frac{s}{b}+\tau\right) + \left(\frac{A}{b}+1\right)\left(\frac{A}{2}\varepsilon^2+\varepsilon(s+V)\right)$

Question

How to find and justify constraints? Proof!

IJRR'17

Approach: Safety Proofs for Autonomous CPS

Autonomous CPS

act

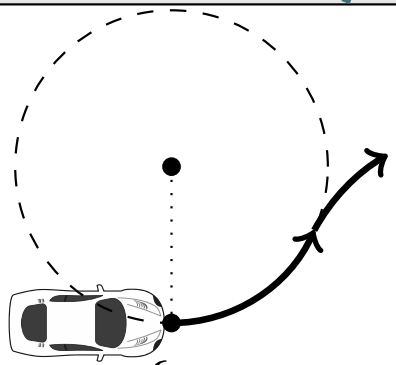
observe

Monitor transfers safety

ModelPlex proof synthesizes

Compliance Monitor

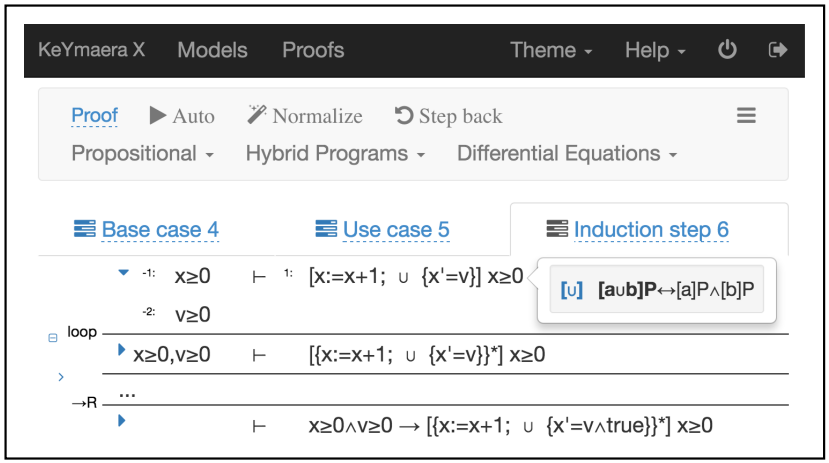
Model



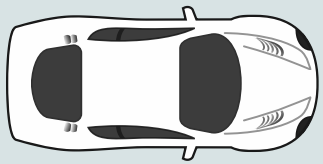
actions: $\{acc, brake\}$

motion: $x'' = a$

KeYmaera X

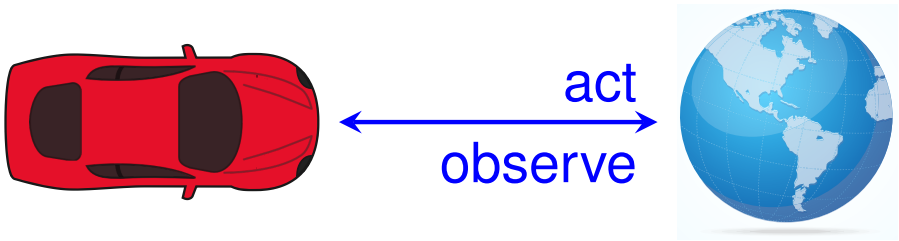


generates proofs



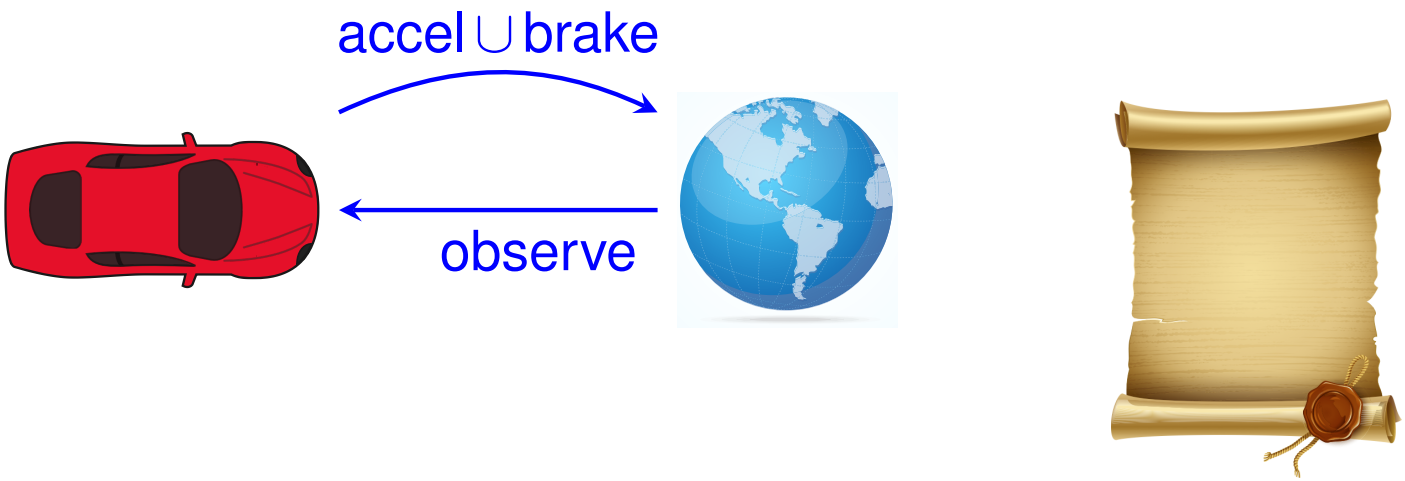
Model Safety

Proof and invariant search



Reinforcement Learning learns from experience of trying actions

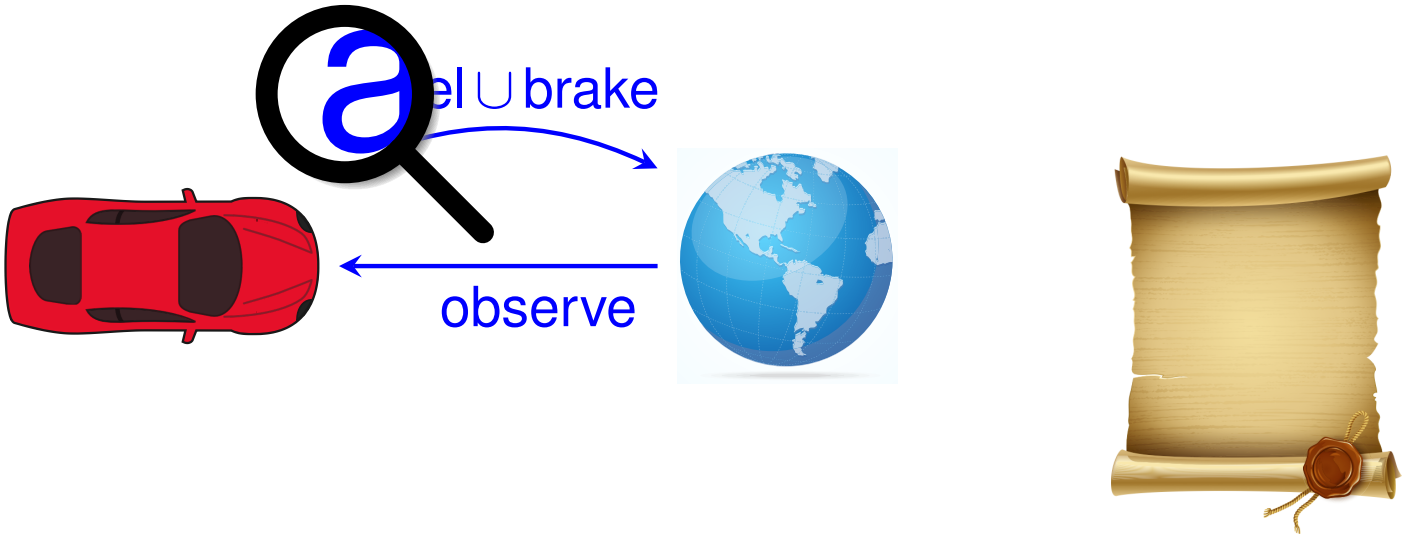
AAAI'18,ITC'18,TACAS'19,QEST'19



RL chooses an action, observes outcome, reinforces in policy if successful

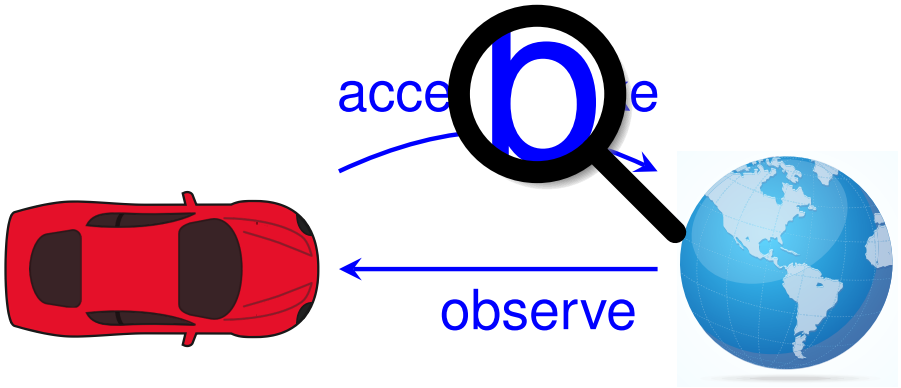
AAAI'18,ITC'18,TACAS'19,QEST'19

Safe AI for Autonomous CPS



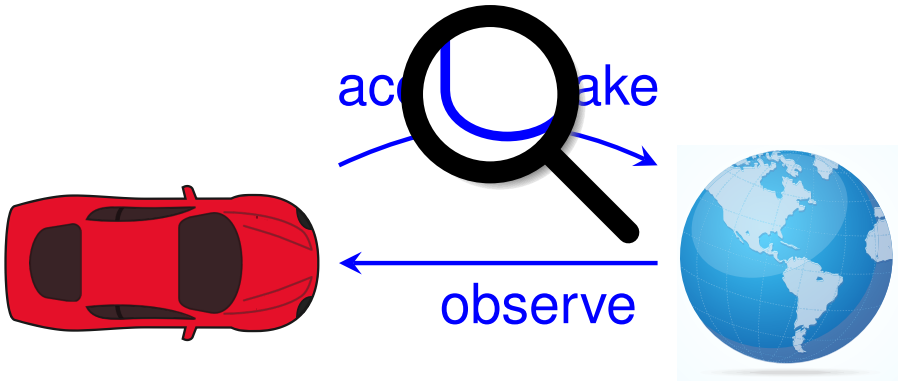
ModelPlex monitor inspects each decision, vetoes if unsafe

AAAI'18,ITC'18,TACAS'19,QEST'19



ModelPlex monitor gives early feedback about possible future problems.
No need to wait till disaster strikes and propagate back.

AAAI'18,ITC'18,TACAS'19,QEST'19

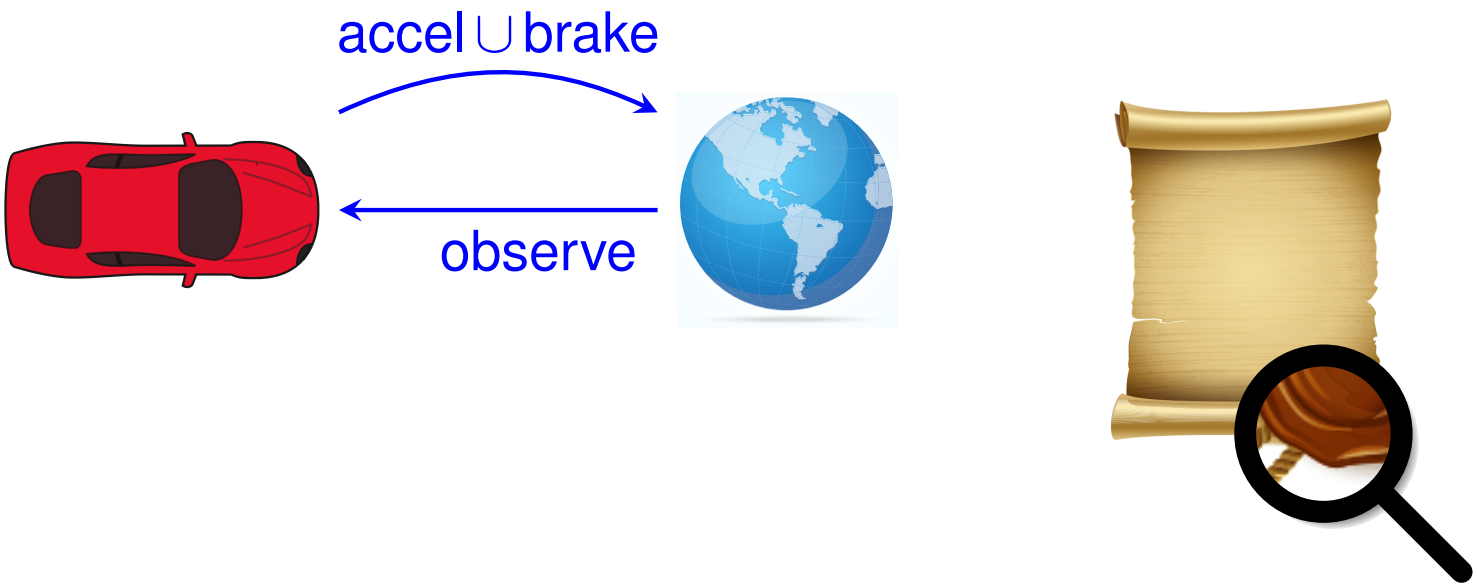


dL benefits from RL optimization.

RL benefits from dL safety signal.

AAAI'18,ITC'18,TACAS'19,QEST'19

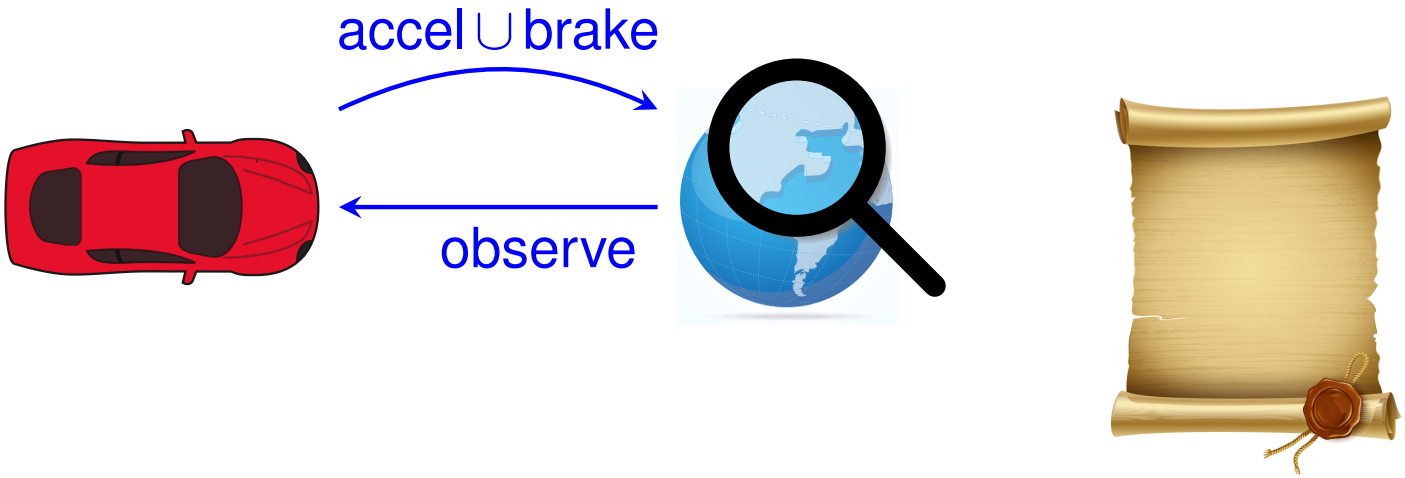
Safe AI for Autonomous CPS



Theorem	Safe policy if ODE accurate
Experiment	Graceful recovery outside ODE ⇝ quantitative ModelPlex
	Detect modeled versus unmodeled state space ⇝ ModelPlex

AAAI'18,ITC'18,TACAS'19,QEST'19

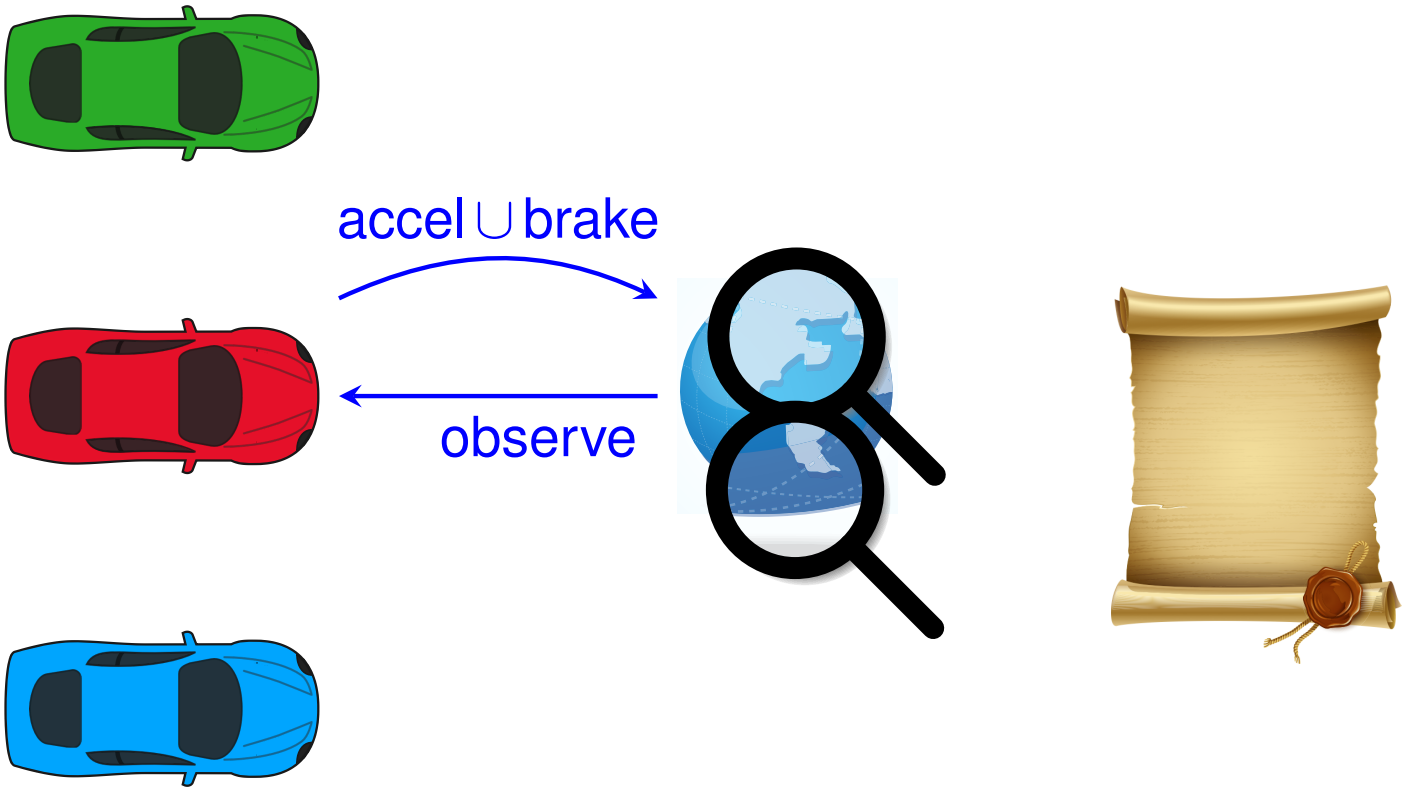
Safe AI for Autonomous CPS



What's safe when off model?

AAAI'18,ITC'18,TACAS'19,QEST'19

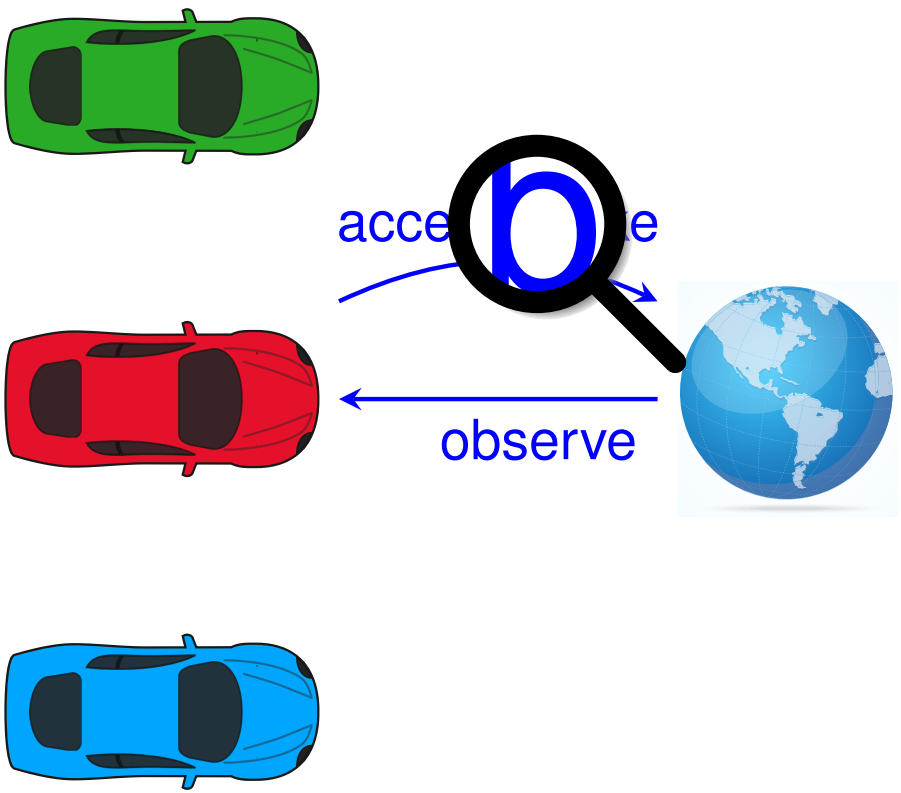
Safe AI for Autonomous CPS



What's safe with multiple possible models?

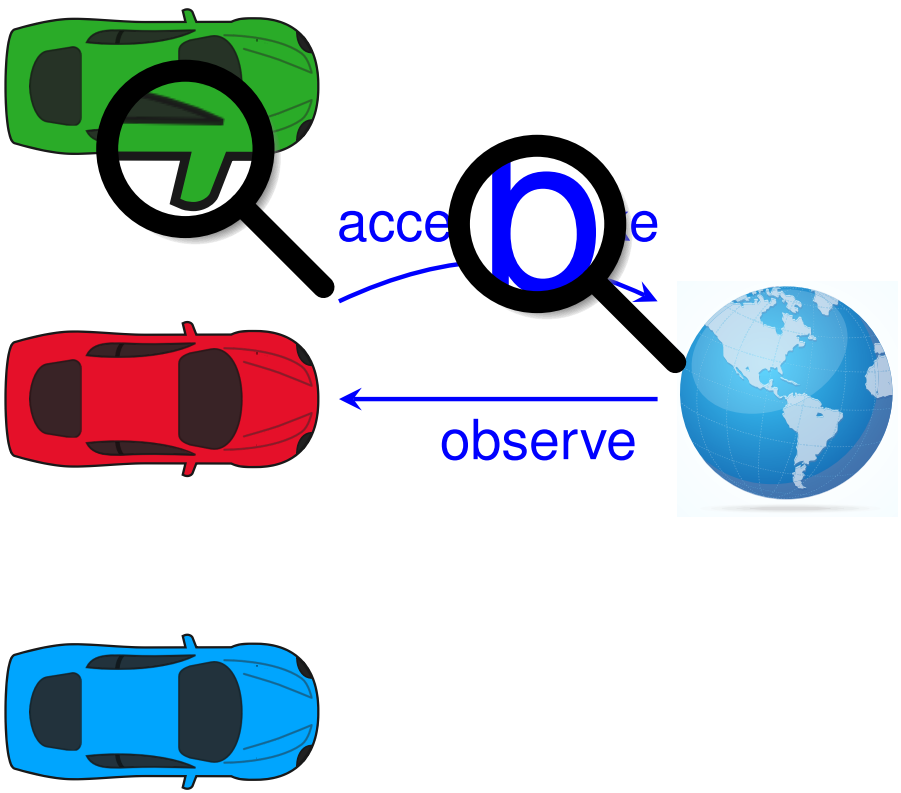
AAAI'18,ITC'18,TACAS'19,QEST'19

Safe AI for Autonomous CPS



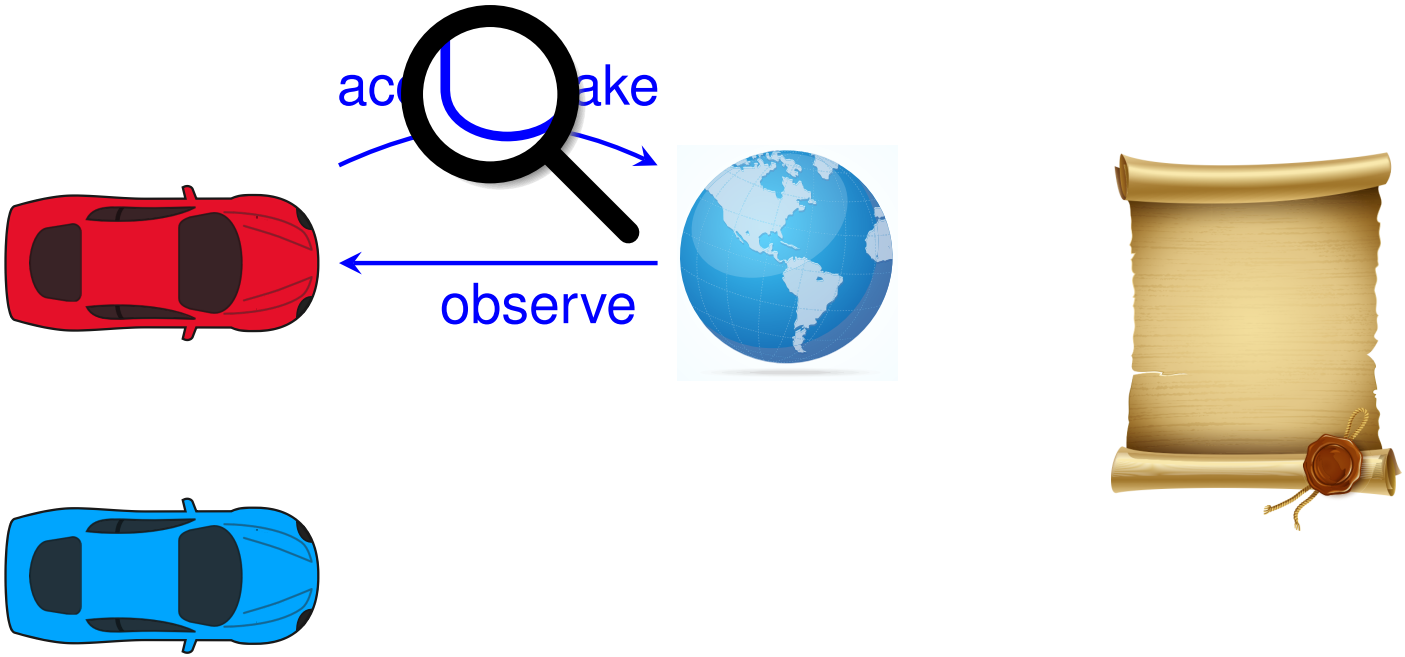
ModelPlex monitors conjunction of all plausible models

AAAI'18,ITC'18,TACAS'19,QEST'19



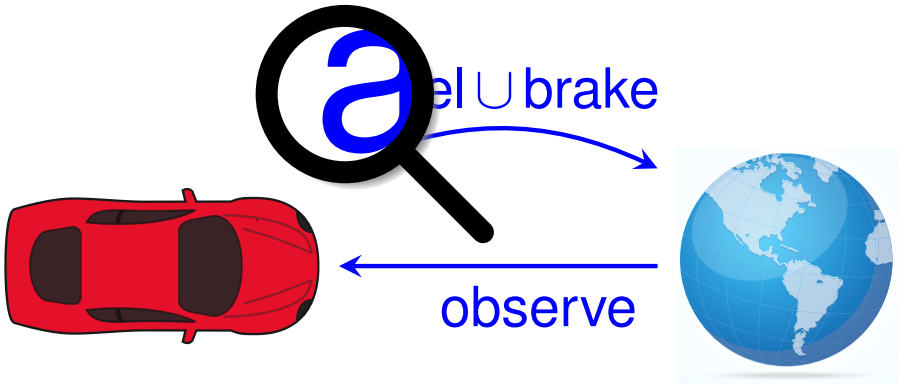
Remove incompatible models after contradictory observation

AAAI'18,ITC'18,TACAS'19,QEST'19



Plan differentiating experiment $\rightsquigarrow$ predictive monitor distinctions

AAAI'18,ITC'18,TACAS'19,QEST'19



Convergence Plausible models converge to true model a.s., if possible

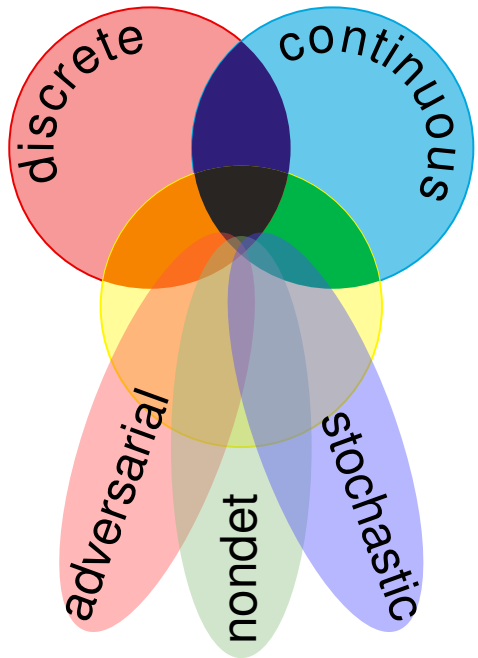
AAAI'18,ITC'18,TACAS'19,QEST'19

Outline

Further Dynamical Systems Challenges

CPSs deserve proofs as safety evidence!

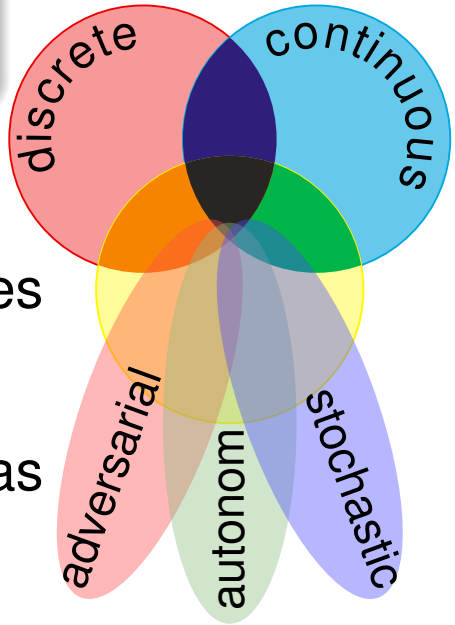
- Verified CPS implementations by ModelPlex — FMSD'16
- Correct CPS execution — PLDI'18
- CPS proof and tactic languages+libraries — ITP'17
- Big CPS built from safe components — STTT'18
- ODE invariance — JACM'20
- ODE liveness — FAC'21
- ODE stability — TACAS'21
- Invariant generation — FMSD'21
- Safe AI autonomy in CPS — AAAI'18
- Refinement + system property proofs — LICS'16
- CPS information flow — LICS'18
- Hybrid games — TOCL'15
- Constructive hybrid games — IJCAR'20

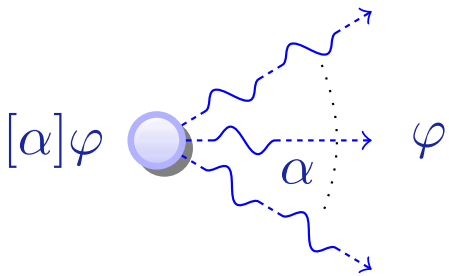


Logic of Autonomous Dynamical Systems

differential dynamic logic

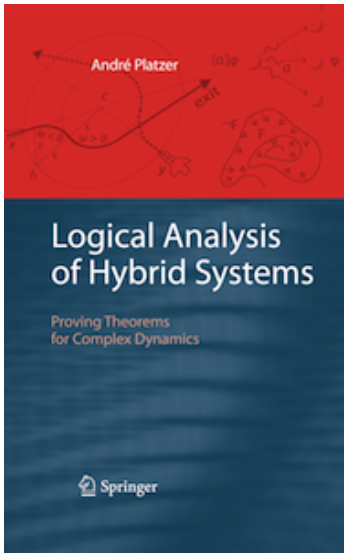
$$\mathsf{dL} = \mathsf{DL} + \mathsf{HP}$$



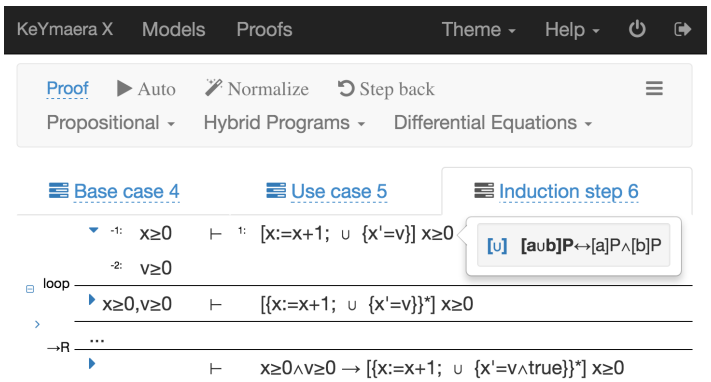


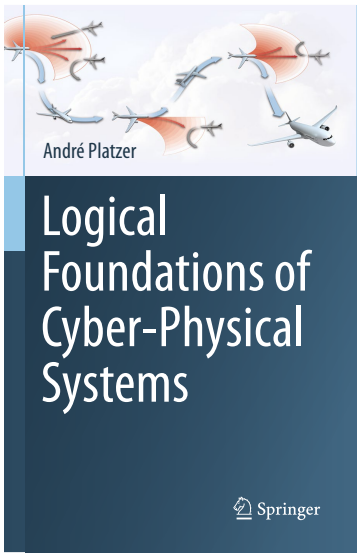
- Strong analytic foundations
- Practical reasoning advances
- Significant applications
- Catalyze many science areas

- Logic & Proofs for CPS
- Programming languages
- Theorem proving
- Multi-dynamical systems



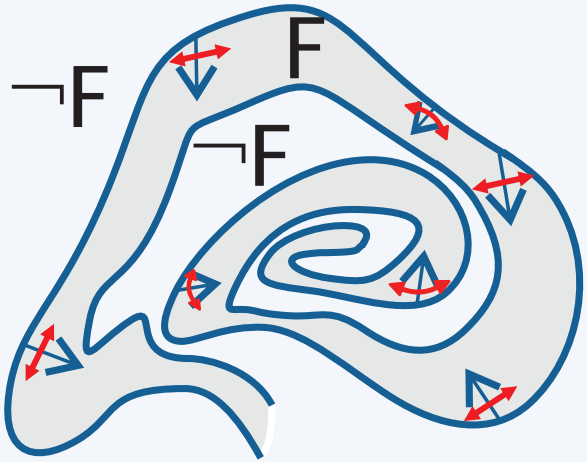
KeYmaera X



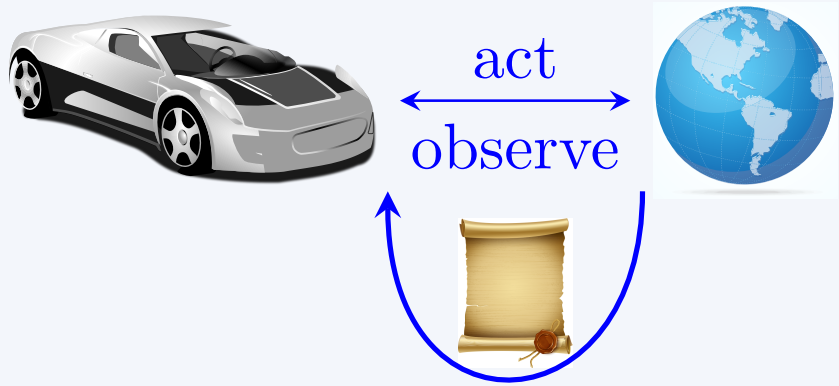


Future Research

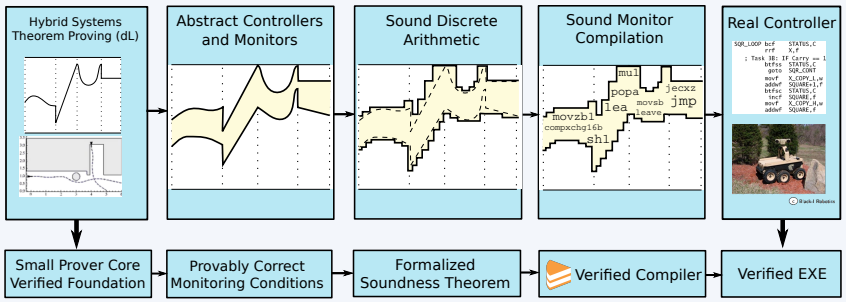
Foundations



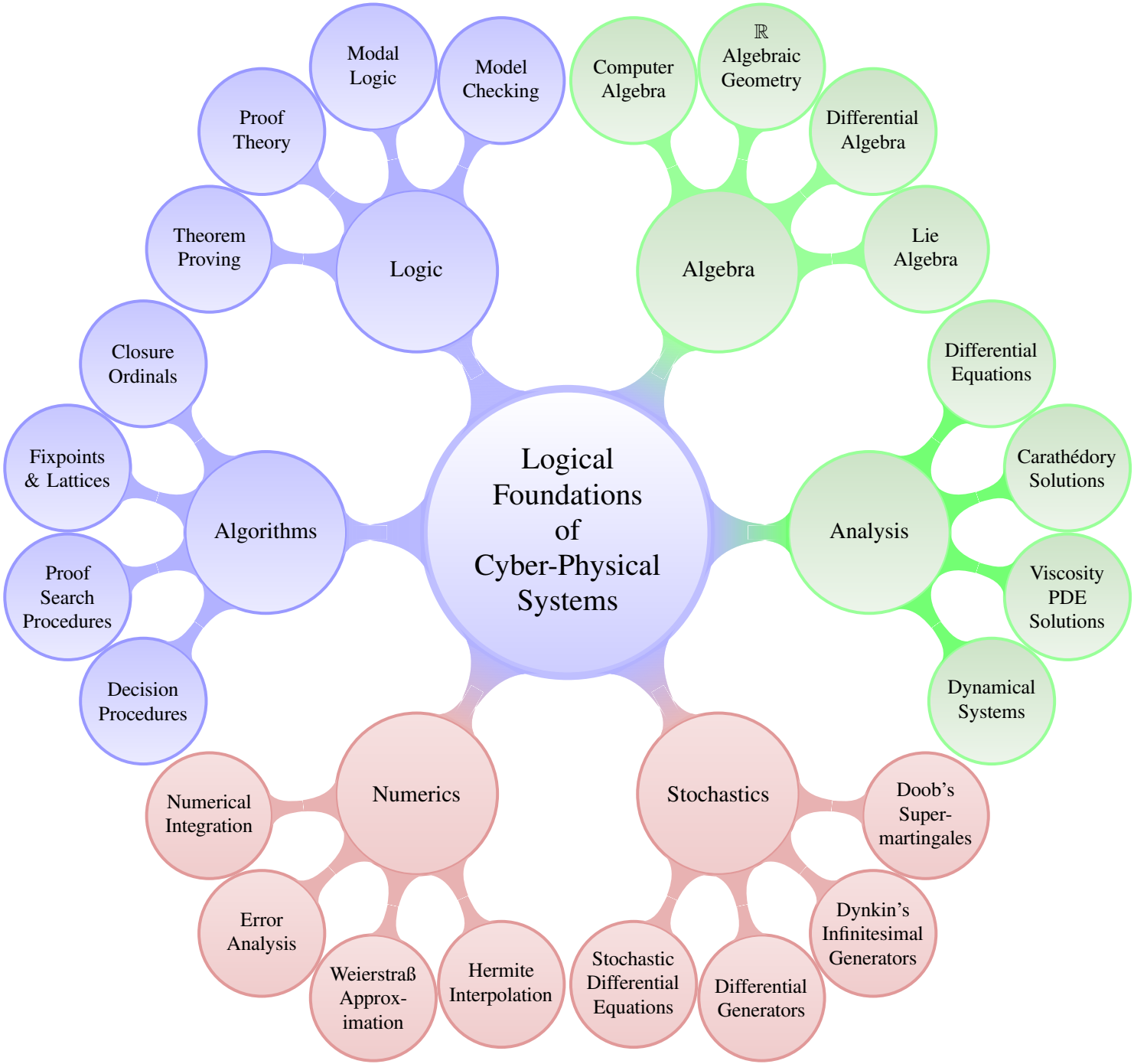
Safe AI Autonomy

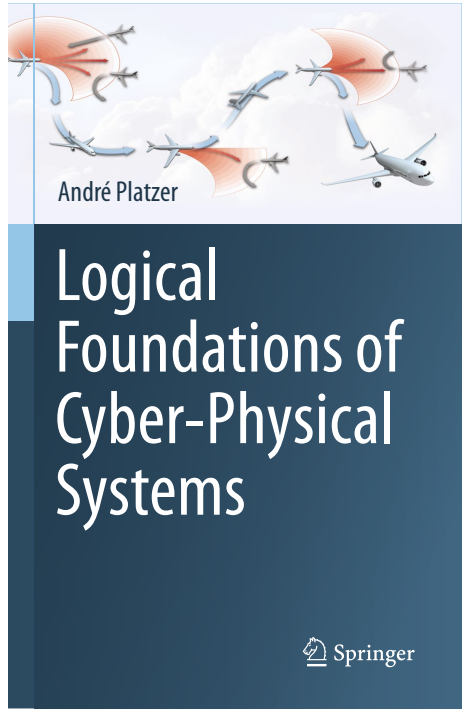


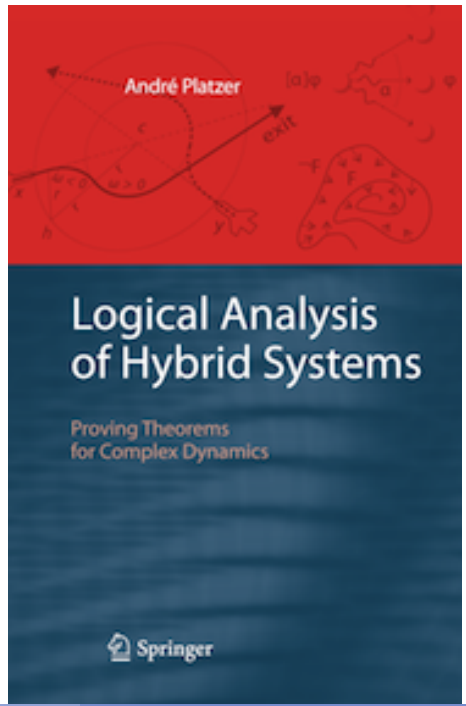
Safe CPS Stacks



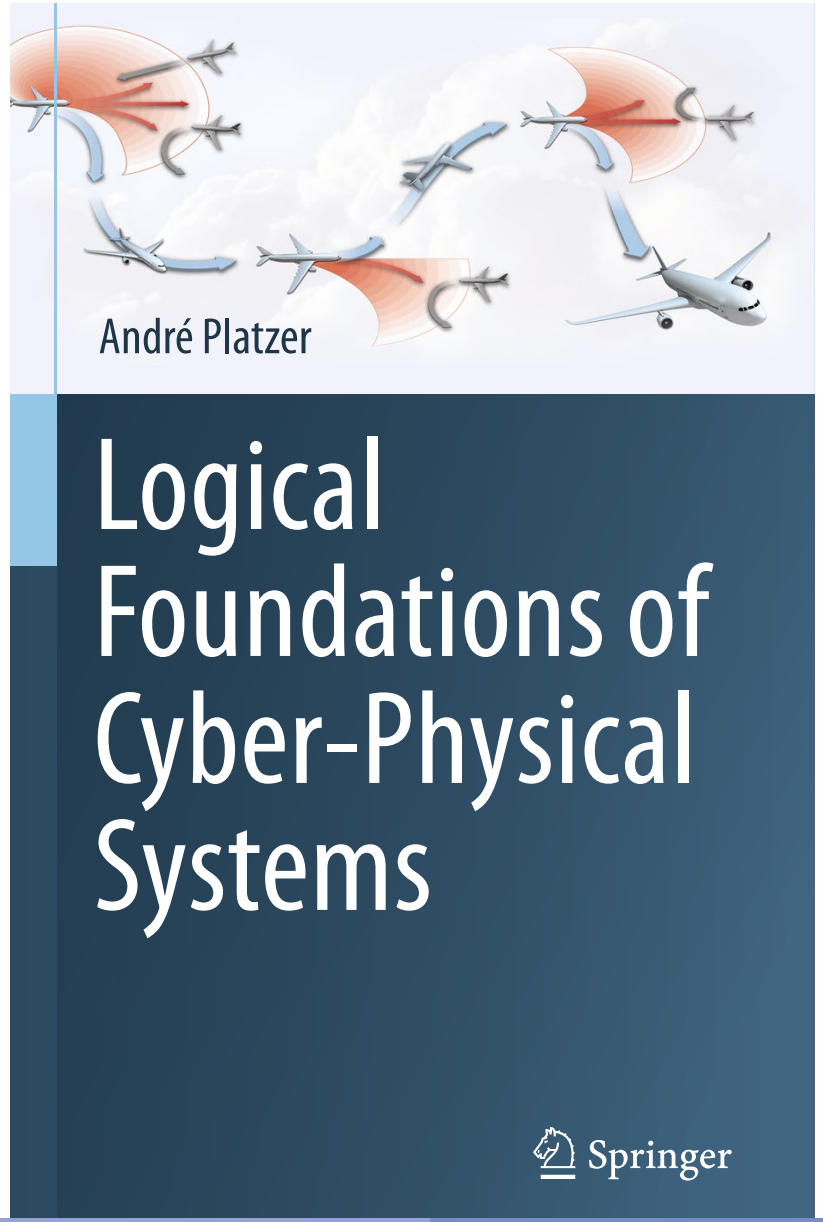
CPS Applications


Logical Foundations of Cyber-Physical Systems
Logic
Modal Logic
Model Checking
Proof Theory
Theorem Proving
Algebra
Computer Algebra
ℝ Algebraic Geometry
Differential Algebra
Lie Algebra
Analysis
Differential Equations
Carathédory Solutions
Viscosity PDE Solutions
Dynamical Systems
Algorithms
Closure Ordinals
Fixpoints & Lattices
Proof Search Procedures
Decision Procedures
Numerics
Numerical Integration
Error Analysis
Weierstraß Approx-imation
Hermite Interpolation
Stochastics
Doob's Super-martingales
Dynkin's Infinitesimal Generators
Stochastic Differential Equations
Differential Generators


André Platzer
Logical Foundations of Cyber-Physical Systems
Springer


André Platzer
Logical Analysis of Hybrid Systems
Proving Theorems for Complex Dynamics
Springer

A. Platzer. *Logical Foundations of Cyber-Physical Systems*. Springer 2018



Outline

7 Appendix

- Soundness and Completeness
- Uniform Substitution
- ModelPlex Runtime Model Validation
- Robot Applications
- Safe AI in CPS

7 Appendix

- Soundness and Completeness
- Uniform Substitution
- ModelPlex Runtime Model Validation
- Robot Applications
- Safe AI in CPS

Complete Proof Theory of Hybrid Systems

Theorem (Sound & Complete) (JAR'08, LICS'12, JAR'17)

dL *calculus is a sound & complete axiomatization of hybrid systems relative to either differential equations* ***or*** *to discrete dynamics.*

Corollary (Complete Proof-theoretical Bridge)

proving continuous = proving hybrid = proving discrete

$$\models P \text{ iff } \mathsf{FOD} \vdash_{\mathsf{dL}} P$$

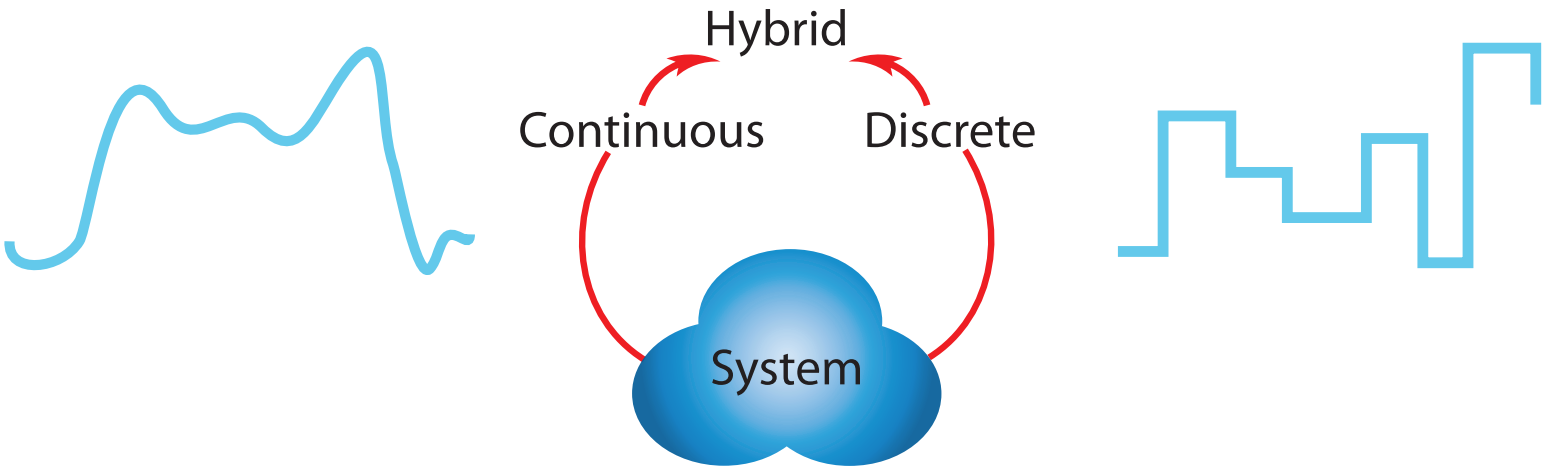
Complete Proof Theory of Hybrid Systems

Theorem (Sound & Complete) (JAR'08, LICS'12, JAR'17)

dL *calculus is a sound & complete axiomatization of hybrid systems relative to either differential equations* ***or*** *to discrete dynamics.*

Corollary (Complete Proof-theoretical Bridge)

proving continuous = proving hybrid = proving discrete



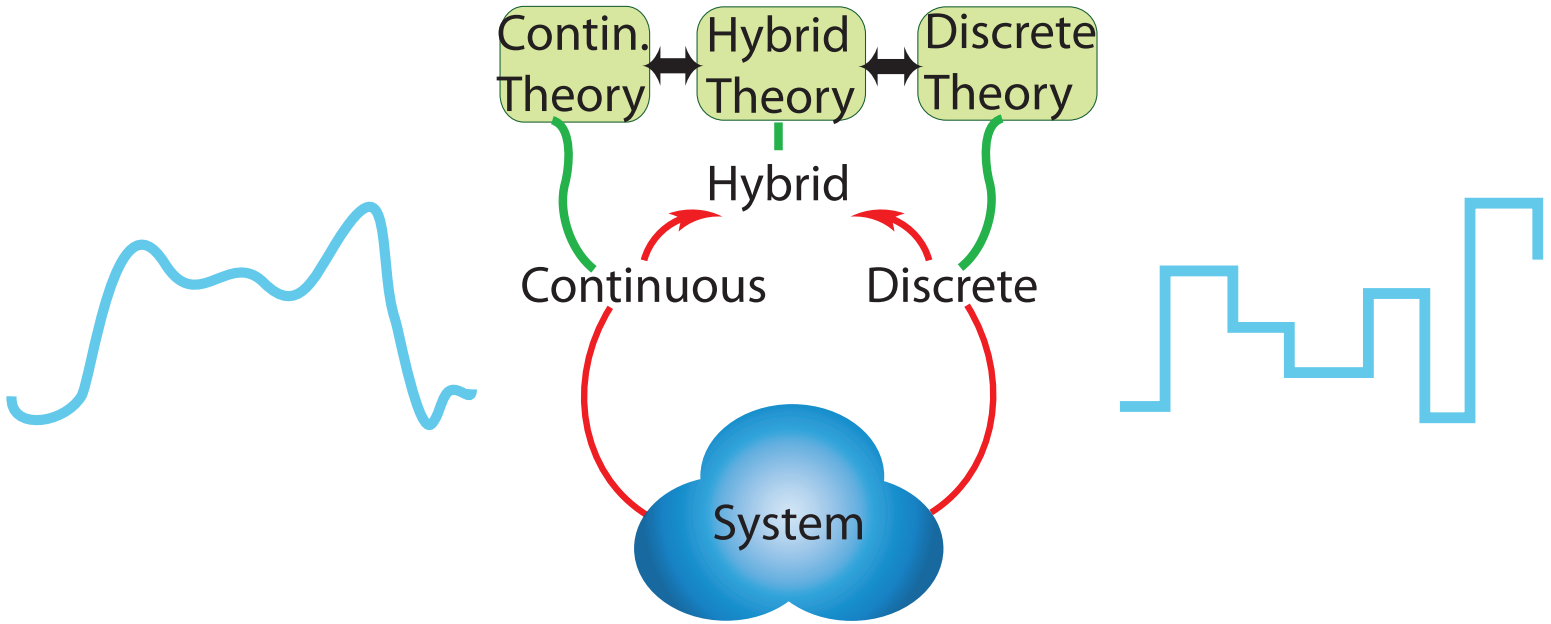
Complete Proof Theory of Hybrid Systems

Theorem (Sound & Complete) (JAR'08, LICS'12, JAR'17)

dL *calculus is a sound & complete axiomatization of hybrid systems relative to either differential equations* ***or*** *to discrete dynamics.*

Corollary (Complete Proof-theoretical Bridge)

proving continuous = proving hybrid = proving discrete



Theorem (Soundness) replace all occurrences of $p(\cdot)$

$$(US)\quad \frac{\phi}{\sigma(\phi)}$$

provided $FV(\sigma|_{\Sigma(\theta)}) \cap BV(\otimes(\cdot)) = \emptyset$ *for each operation* $\otimes(\theta)$ *in* ϕ

i.e. bound variables $U = \mathrm{BV}(\otimes(\cdot))$ of **no** operator $\otimes$
are free in the substitution on its argument θ (U-admissible)

$$\mathrm{US}\frac{[a \cup b]p(\bar{x}) \leftrightarrow [a]p(\bar{x}) \wedge [b]p(\bar{x})}{[x := x+1 \cup x' = 1]x \geq 0 \leftrightarrow [x := x+1]x \geq 0 \wedge [x' = 1]x \geq 0}$$

JAR'17

Uniform Substitution

Theorem (Soundness) replace all occurrences of $p(\cdot)$

$$(US)\quad \frac{\phi}{\sigma(\phi)}$$

provided $FV(\sigma|_{\Sigma(\theta)}) \cap BV(\otimes(\cdot)) = \emptyset$ *for each operation* $\otimes(\theta)$ *in* ϕ

i.e. bound variables $U = \mathrm{BV}(\otimes(\cdot))$ of **no** operator $\otimes$
are free in the substitution on its argument θ (U-admissible)

$$\frac{[v := f]p(v) \leftrightarrow p(f)}{[v := -x][x' = v]\, x \geq 0 \leftrightarrow [x' = -x]\, x \geq 0}$$

JAR'17

Uniform Substitution

Theorem (Soundness) replace all occurrences of $p(\cdot)$

Modular interface: Prover vs. Logic

$$(US)\quad \frac{\phi}{\sigma(\phi)}$$

provided $FV(\sigma|_{\Sigma(\theta)}) \cap BV(\otimes(\cdot)) = \emptyset$ *for each operation* $\otimes(\theta)$ *in* ϕ

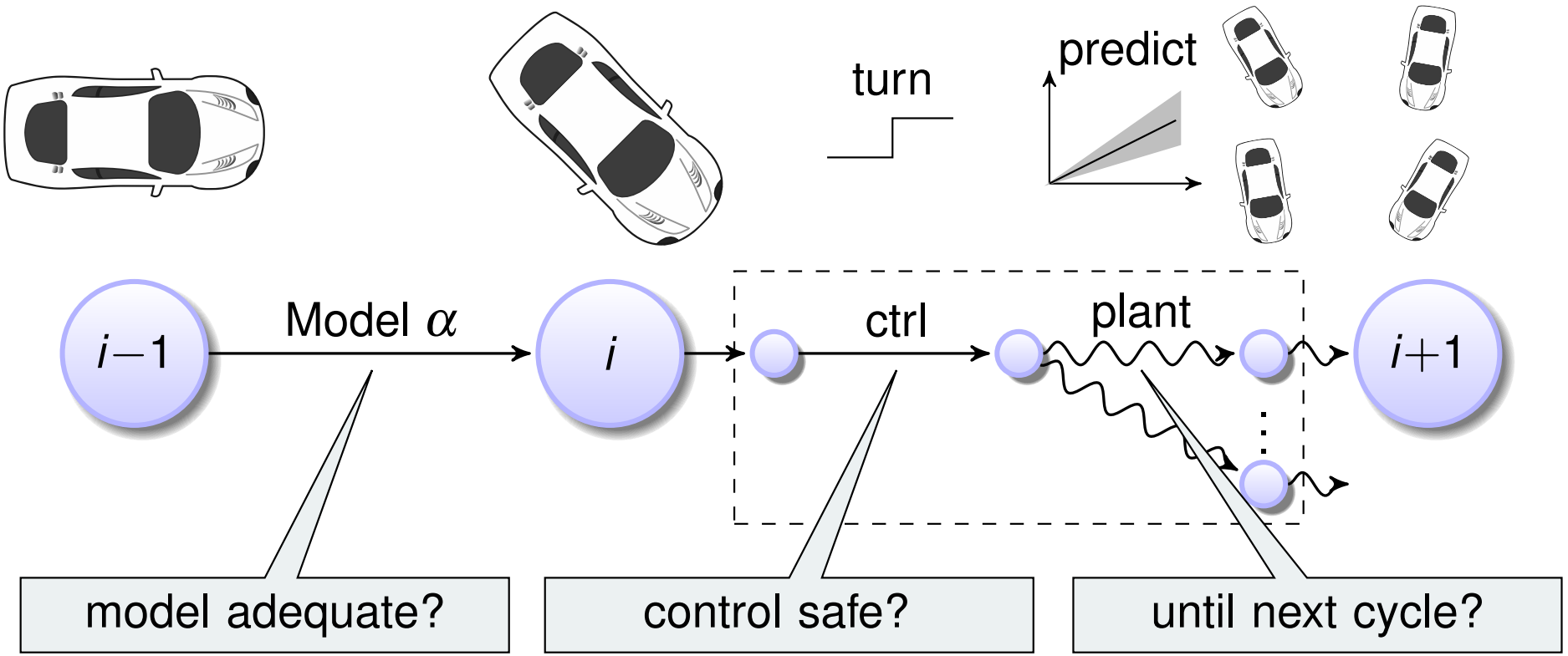
i.e. bound variables $U = \mathrm{BV}(\otimes(\cdot))$ of **no** operator $\otimes$
are free in the substitution on its argument θ (U-admissible)

If you bind a free variable, you go to logic jail!

$$\frac{[v := f]p(v) \leftrightarrow p(f)}{[v := -x][x' = v]\, x \geq 0 \leftrightarrow [x' = -x]\, x \geq 0}$$

Clash

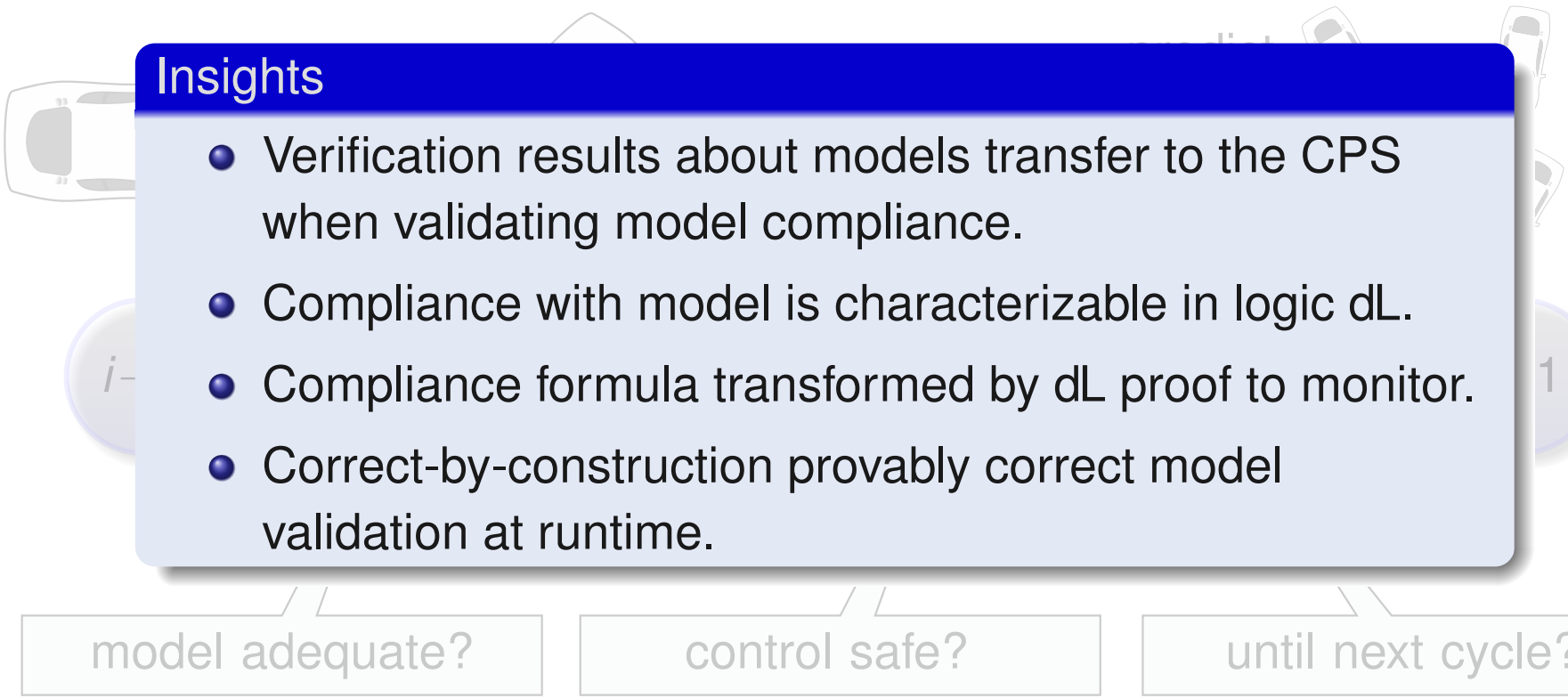
ModelPlex **ensures that verification results** about models
apply to CPS implementations



RV'14,FMSD'16

ModelPlex Runtime Model Validation

ModelPlex **ensures that verification results** about models **apply to CPS** implementations



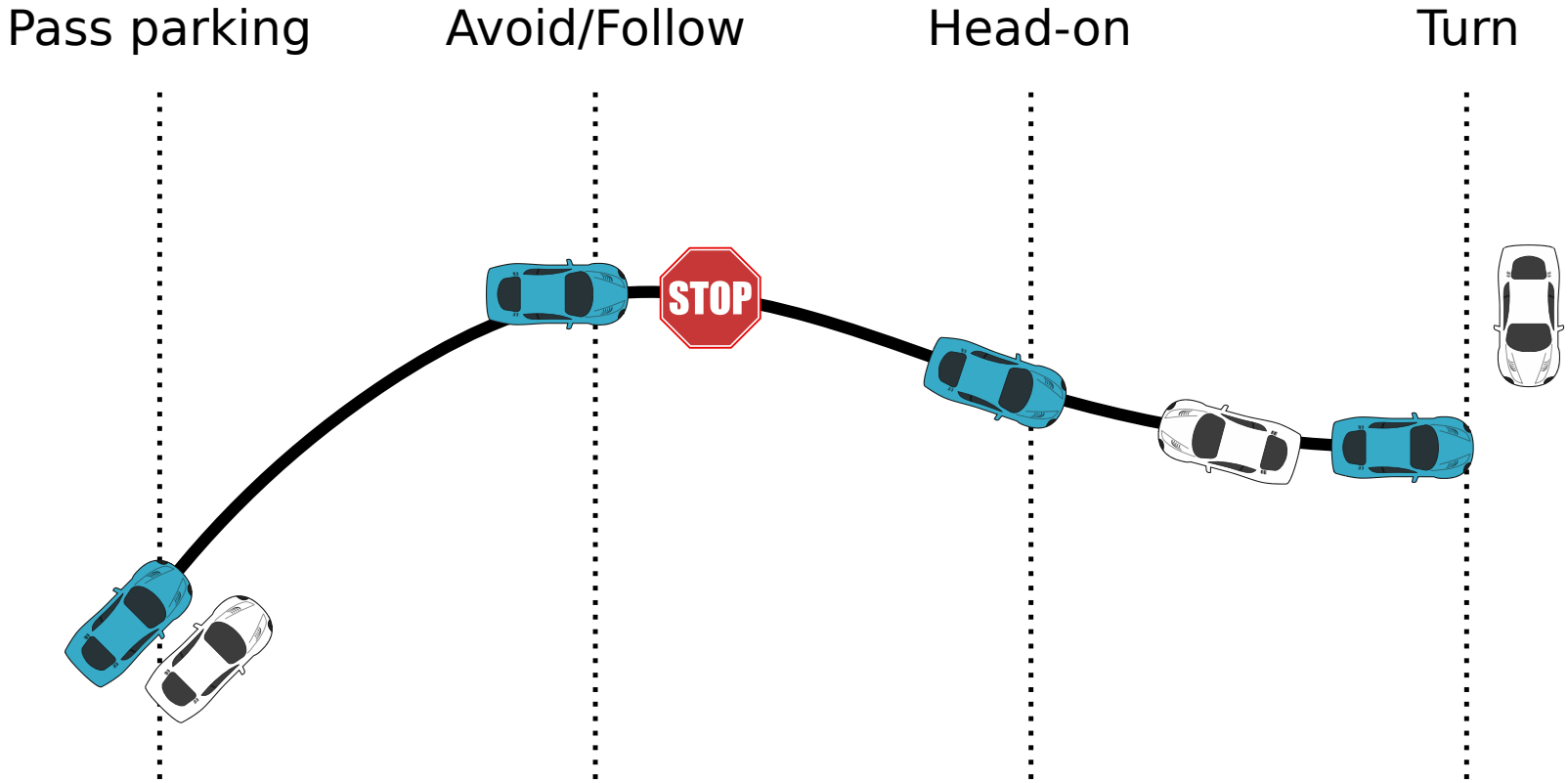
Insights

- Verification results about models transfer to the CPS when validating model compliance.
- Compliance with model is characterizable in logic dL.
- Compliance formula transformed by dL proof to monitor.
- Correct-by-construction provably correct model validation at runtime.

RV'14,FMSD'16

Ground Robot Obstacle Avoidance: Verify

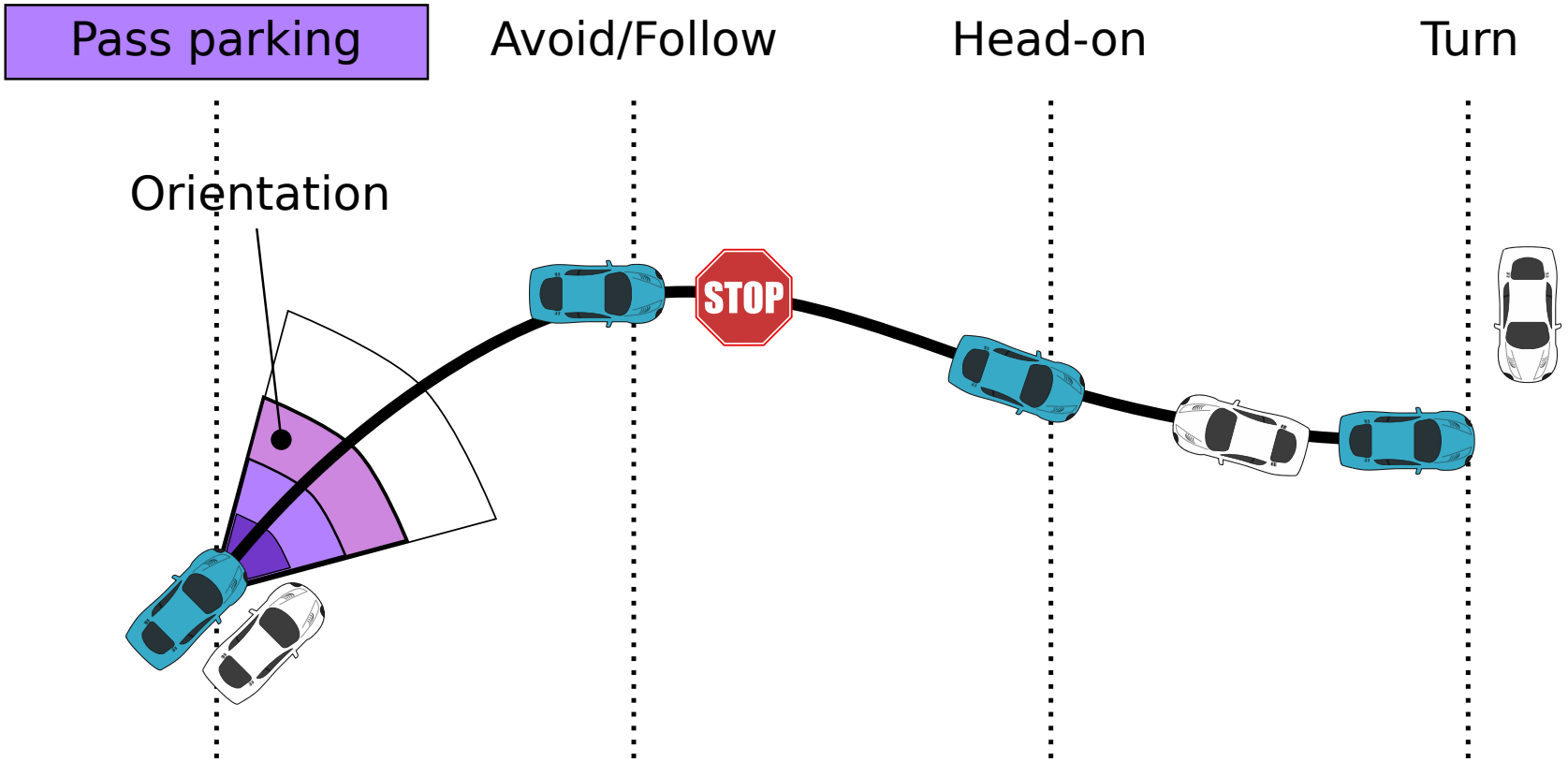
- Fundamental safety question for ground robot navigation IJRR'17
- When will which control decision avoid obstacles?
- Depends on safety objective, physical capabilities of robot + obstacle



1. Identified safe region for each safety notion symbolically
2. Proved safety for hybrid systems ground robot model in KeYmaera X

Ground Robot Obstacle Avoidance: Verify

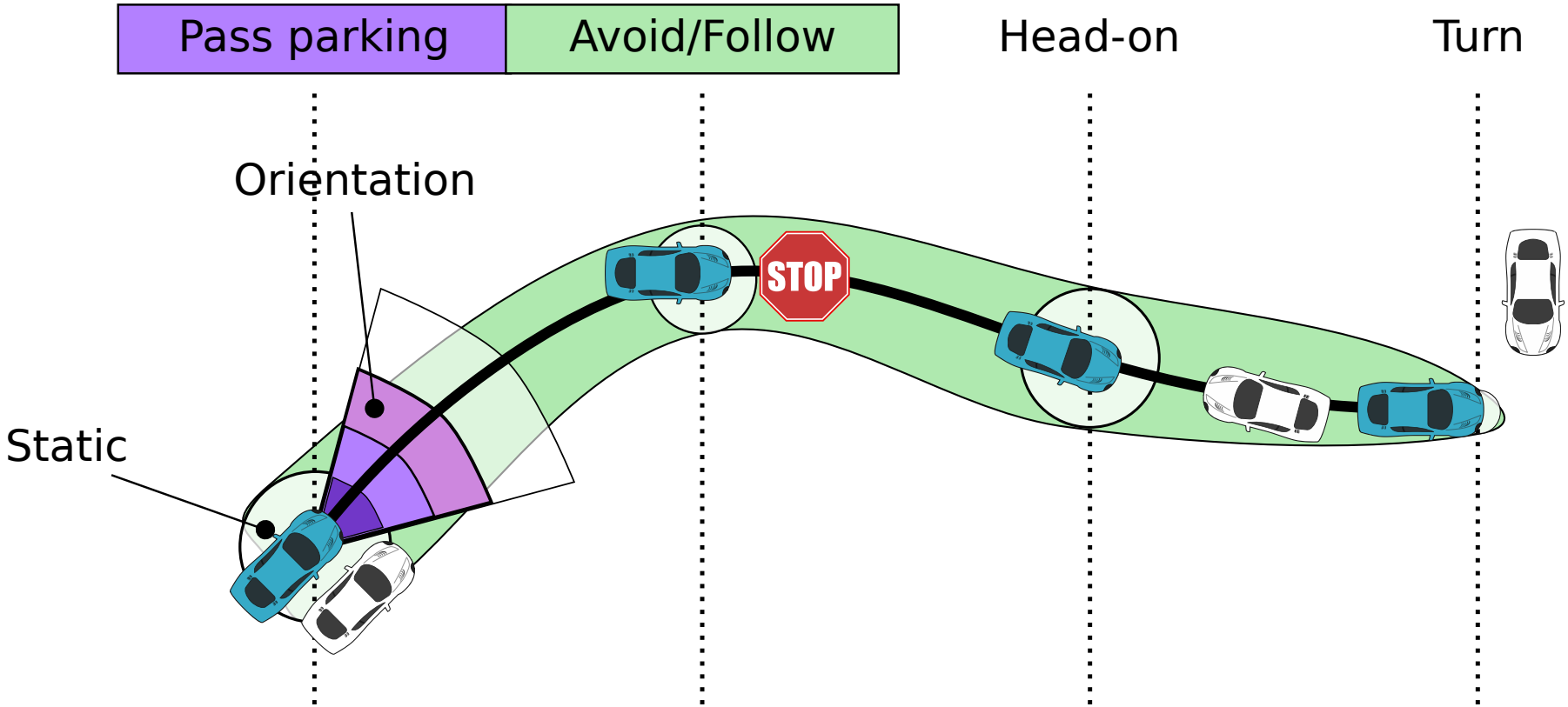
- Fundamental safety question for ground robot navigation IJRR'17
- When will which control decision avoid obstacles?
- Depends on safety objective, physical capabilities of robot + obstacle



1. Identified safe region for each safety notion symbolically
2. Proved safety for hybrid systems ground robot model in KeYmaera X

Ground Robot Obstacle Avoidance: Verify

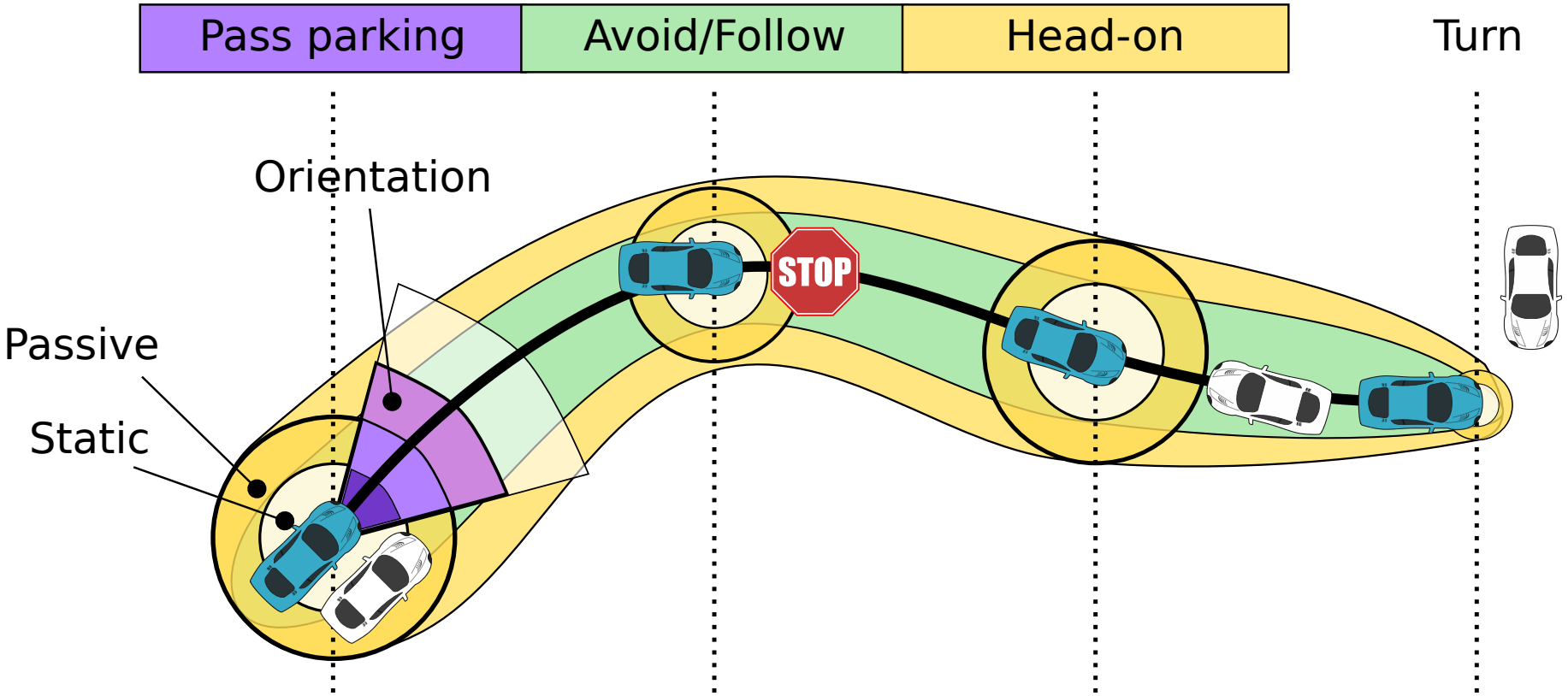
- Fundamental safety question for ground robot navigation IJRR'17
- When will which control decision avoid obstacles?
- Depends on safety objective, physical capabilities of robot + obstacle



1. Identified safe region for each safety notion symbolically
2. Proved safety for hybrid systems ground robot model in KeYmaera X

Ground Robot Obstacle Avoidance: Verify

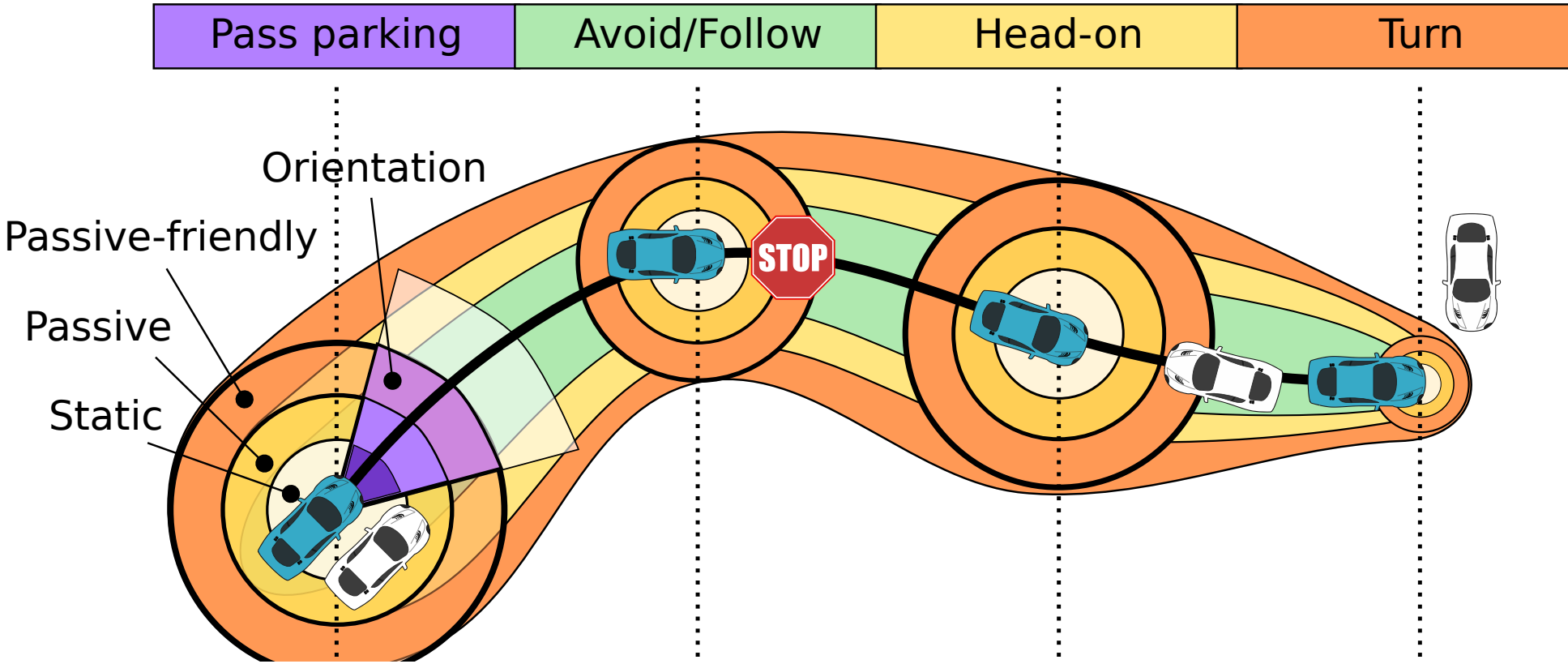
- Fundamental safety question for ground robot navigation IJRR'17
- When will which control decision avoid obstacles?
- Depends on safety objective, physical capabilities of robot + obstacle



1. Identified safe region for each safety notion symbolically
2. Proved safety for hybrid systems ground robot model in KeYmaera X

Ground Robot Obstacle Avoidance: Verify

- Fundamental safety question for ground robot navigation IJRR'17
- When will which control decision avoid obstacles?
- Depends on safety objective, physical capabilities of robot + obstacle



1. Identified safe region for each safety notion symbolically
2. Proved safety for hybrid systems ground robot model in KeYmaera X

Ground Robot Invariants and Safe Control Constraints

Safety	Invariant + Safe Control
static	$\|p-o\|_\infty > \frac{s^2}{2b} + \left(\frac{A}{b}+1\right)\left(\frac{A}{2}\varepsilon^2+\varepsilon s\right)$
passive	$s \neq 0 \rightarrow \|p-o\|_\infty > \frac{s^2}{2b} + V\frac{s}{b} + \left(\frac{A}{b}+1\right)\left(\frac{A}{2}\varepsilon^2+\varepsilon(s+V)\right)$
+ sensor	$\|\hat{p}-o\|_\infty > \frac{s^2}{2b} + V\frac{s}{b} + \left(\frac{A}{b}+1\right)\left(\frac{A}{2}\varepsilon^2+\varepsilon(s+V)\right) + \Delta_p$
+ disturb.	$\|p-o\|_\infty > \frac{s^2}{2b\Delta_a} + V\frac{s}{b\Delta_a} + \left(\frac{A}{b\Delta_a}+1\right)\left(\frac{A}{2}\varepsilon^2+\varepsilon(s+V)\right)$
+ failure	$\|\hat{p}-o\|_\infty > \frac{s^2}{2b} + V\frac{s}{b} + \left(\frac{A}{b}+1\right)\left(\frac{A}{2}\varepsilon^2+\varepsilon(v+V)\right) + \Delta_p + g\Delta$
friendly	$\|p-o\|_\infty > \frac{s^2}{2b} + \frac{V^2}{2b_o} + V\left(\frac{s}{b}+\tau\right) + \left(\frac{A}{b}+1\right)\left(\frac{A}{2}\varepsilon^2+\varepsilon(s+V)\right)$

IJRR'17

Ground Robot Invariants and Safe Control Constraints

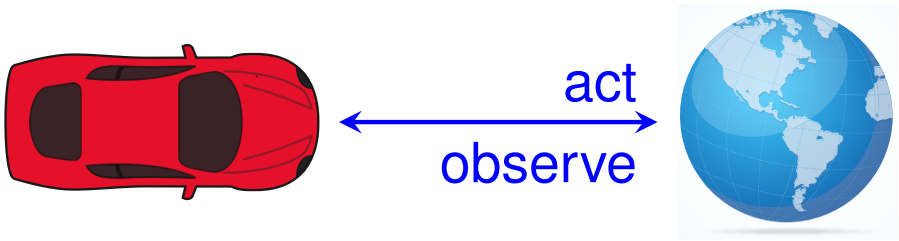
Safety	Invariant + Safe Control
static	$\|p-o\|_\infty > \frac{s^2}{2b} + \left(\frac{A}{b}+1\right)\left(\frac{A}{2}\varepsilon^2+\varepsilon s\right)$
passive	$s \neq 0 \rightarrow \|p-o\|_\infty > \frac{s^2}{2b} + V\frac{s}{b} + \left(\frac{A}{b}+1\right)\left(\frac{A}{2}\varepsilon^2+\varepsilon(s+V)\right)$
+ sensor	$\|p-o\|_\infty > \frac{s^2}{2b} + V\frac{s}{b} + \left(\frac{A}{b}+1\right)\left(\frac{A}{2}\varepsilon^2+\varepsilon(s+V)\right) + \Delta_p$
+ disturb.	$\|p-o\|_\infty > \frac{s^2}{2b\Delta_a} + V\frac{s}{b\Delta_a} + \left(\frac{A}{b\Delta_a}+1\right)\left(\frac{A}{2}\varepsilon^2+\varepsilon(s+V)\right)$
+ failure	$\|\hat{p}-o\|_\infty > \frac{s^2}{2b} + V\frac{s}{b} + \left(\frac{A}{b}+1\right)\left(\frac{A}{2}\varepsilon^2+\varepsilon(v+V)\right) + \Delta_p + g\Delta$
friendly	$\|p-o\|_\infty > \frac{s^2}{2b} + \frac{V^2}{2b_o} + V\left(\frac{s}{b}+\tau\right) + \left(\frac{A}{b}+1\right)\left(\frac{A}{2}\varepsilon^2+\varepsilon(s+V)\right)$

Question

How to find and justify constraints? Proof!

IJRR'17

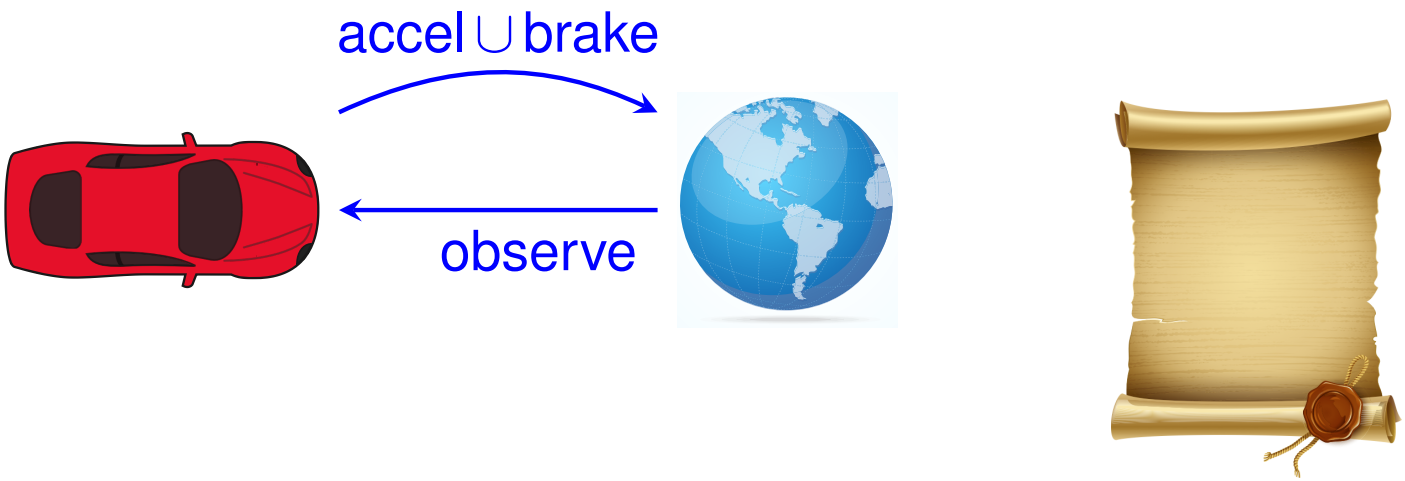
Learning to Act in a CPS



Reinforcement Learning learns from experience of trying actions

AAAI'18,ITC'18,TACAS'19,QEST'19

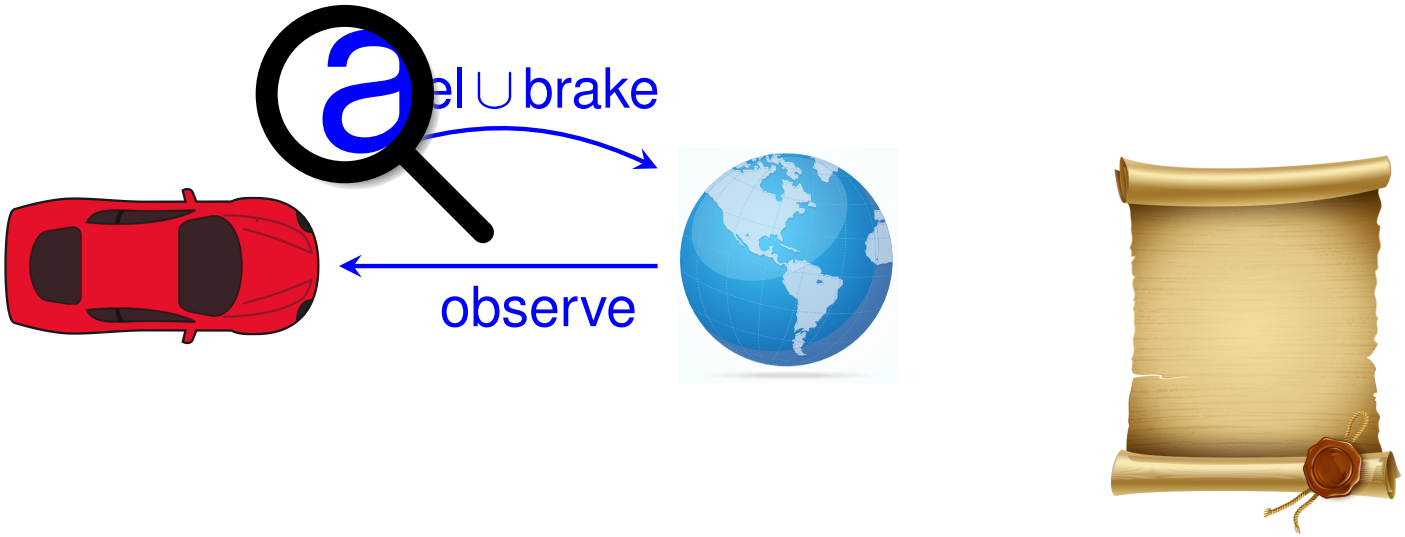
Learning to Act in a CPS



RL chooses an action, observes outcome, reinforces in policy if successful

AAAI'18,ITC'18,TACAS'19,QEST'19

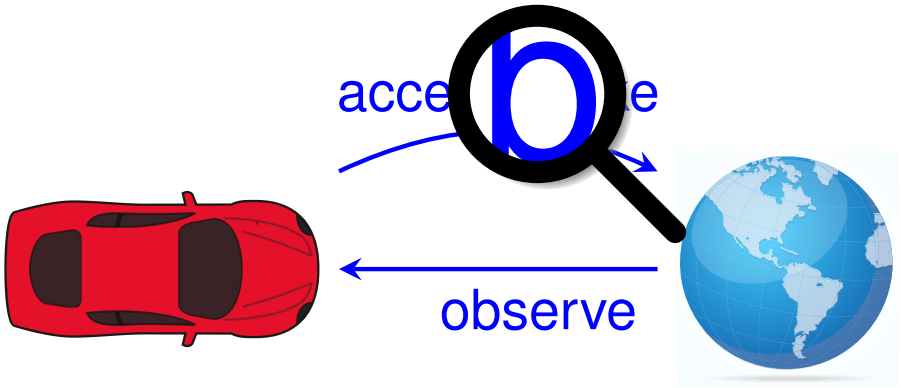
Learning to Act Safely in a CPS



ModelPlex monitor inspects each decision, vetoes if unsafe

AAAI'18,ITC'18,TACAS'19,QEST'19

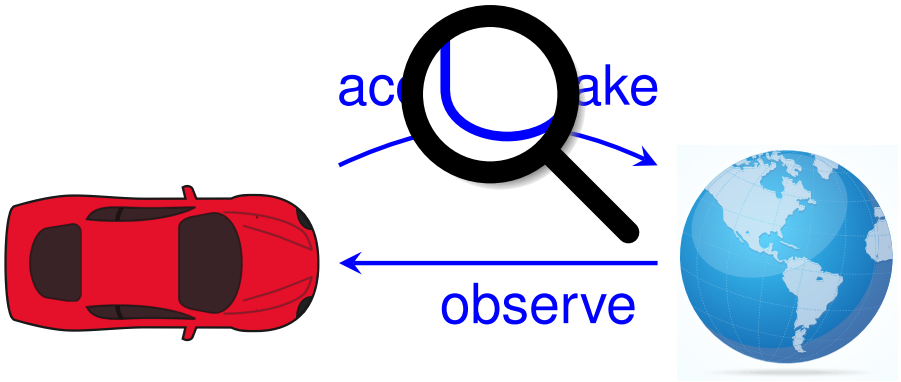
Learning to Act Safely in a CPS



ModelPlex monitor gives early feedback about possible future problems.
No need to wait till disaster strikes and propagate back.

AAAI'18,ITC'18,TACAS'19,QEST'19

Learning to Act Safely in a CPS

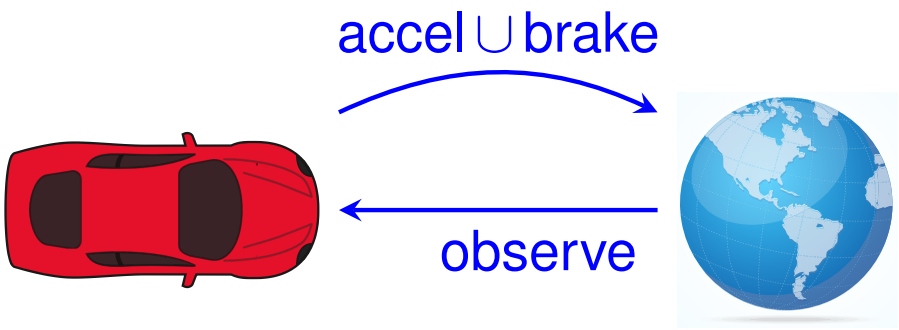


dL benefits from RL optimization.

RL benefits from dL safety signal.

AAAI'18,ITC'18,TACAS'19,QEST'19

Learning to Act Safely in a CPS



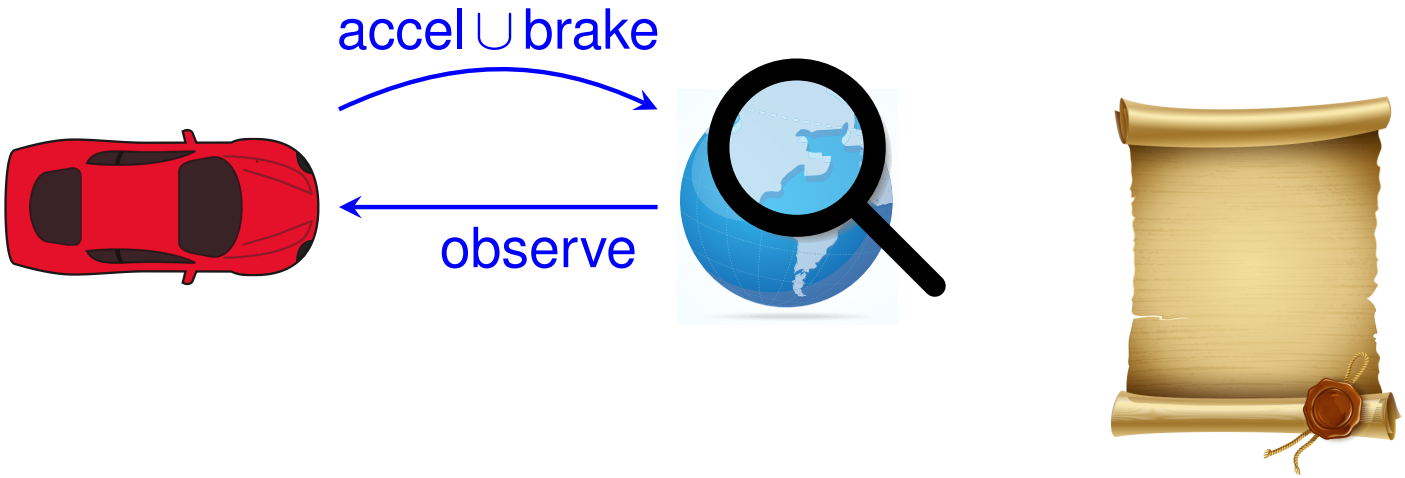
Theorem Safe policy if ODE accurate

Experiment Graceful recovery outside ODE ⇝ quantitative ModelPlex

Detect modeled versus unmodeled state space ⇝ ModelPlex

AAAI'18,ITC'18,TACAS'19,QEST'19

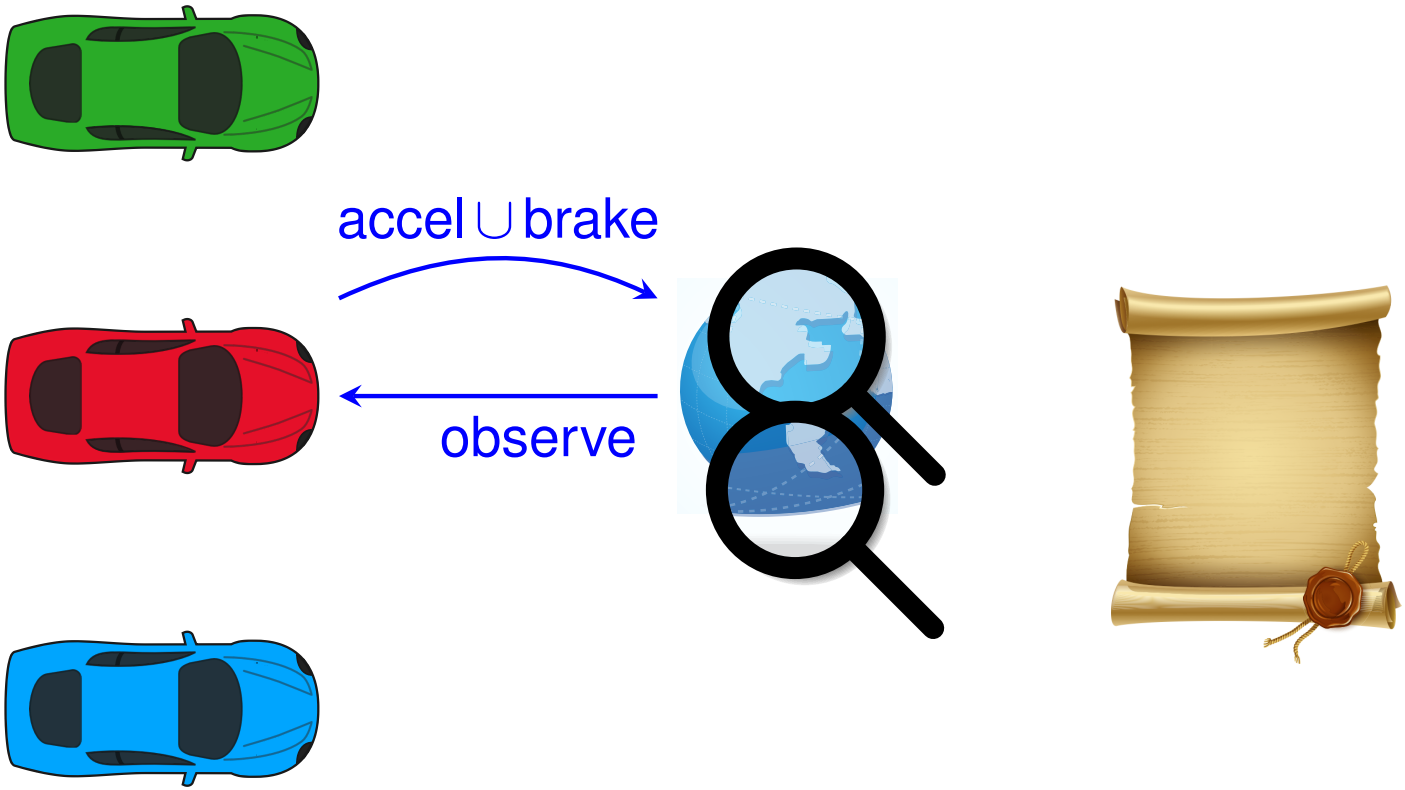
Learning to Act Safely in a CPS



What's safe when off model?

AAAI'18,ITC'18,TACAS'19,QEST'19

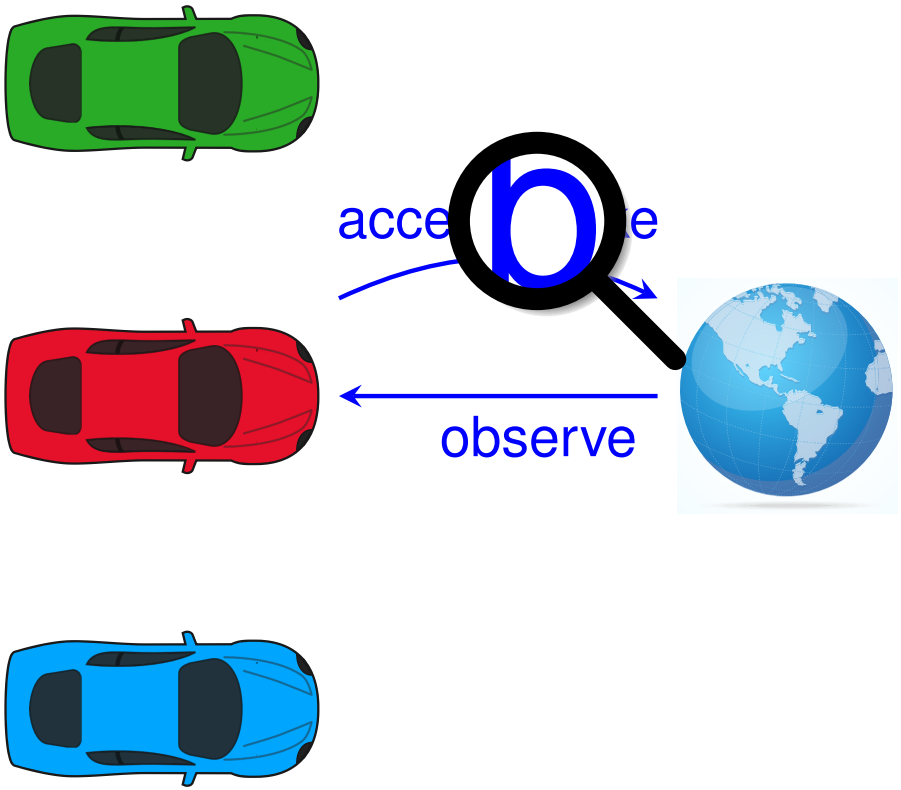
Learning to Act Safely in a CPS with Multiple Models



What's safe with multiple possible models?

AAAI'18,ITC'18,TACAS'19,QEST'19

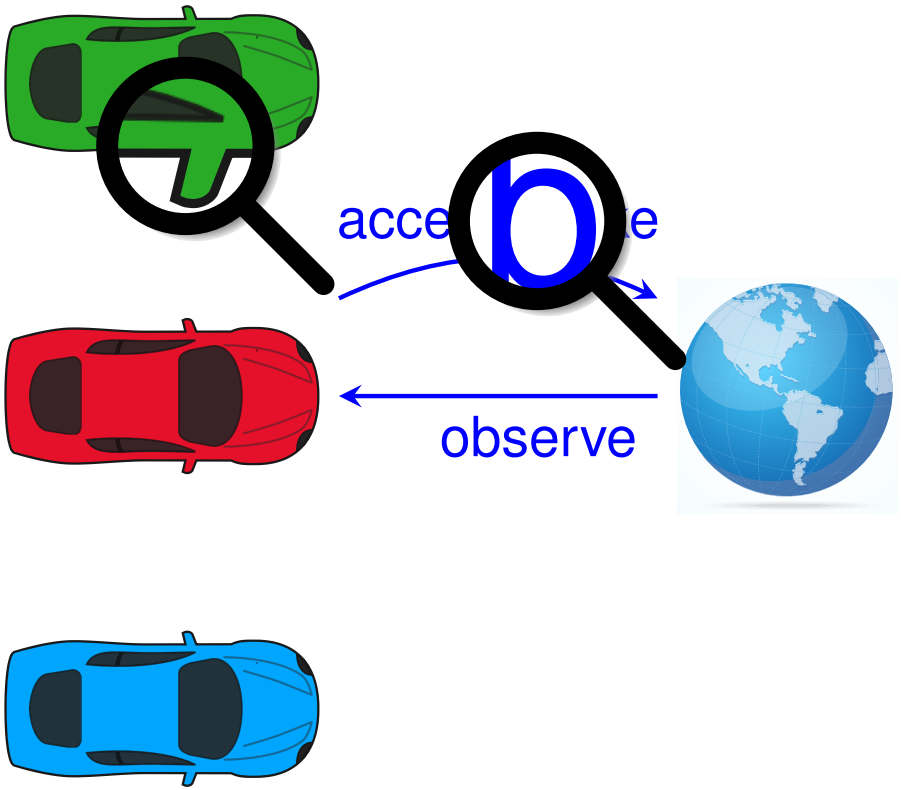
Learning to Act Safely in a CPS with Multiple Models



ModelPlex monitors conjunction of all plausible models

AAAI'18,ITC'18,TACAS'19,QEST'19

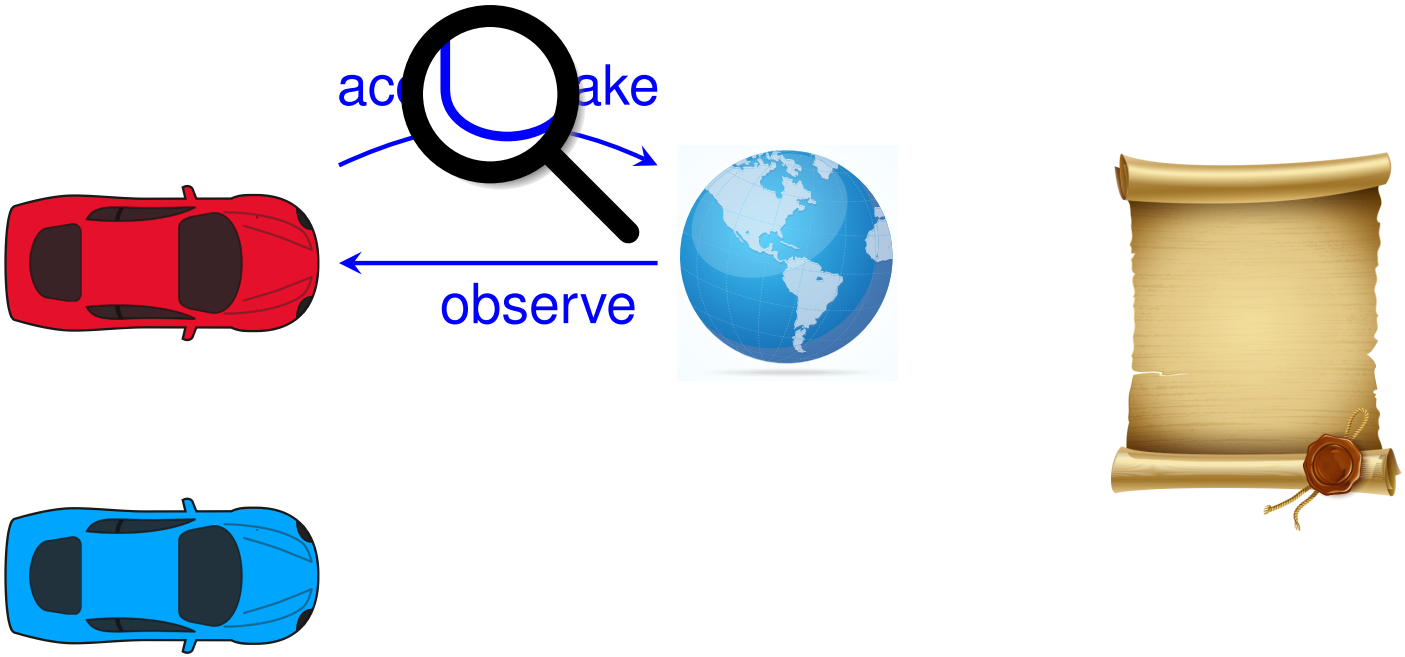
Learning to Act Safely in a CPS with Multiple Models



Remove incompatible models after contradictory observation

AAAI'18,ITC'18,TACAS'19,QEST'19

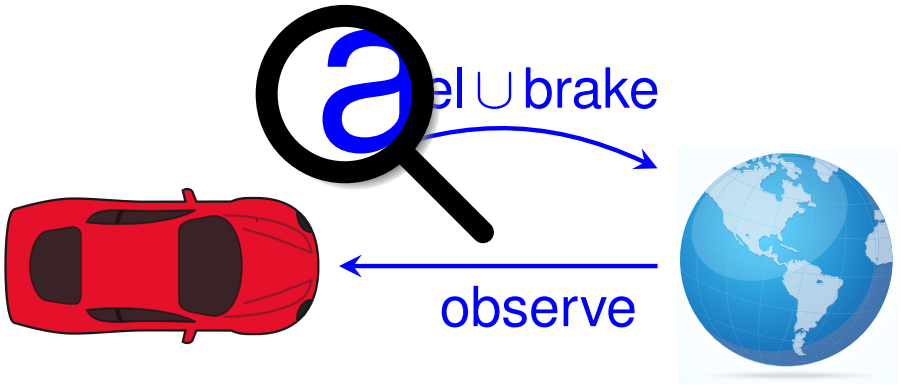
Learning to Act Safely in a CPS with Multiple Models



Plan differentiating experiment $\rightsquigarrow$ predictive monitor distinctions

AAAI'18,ITC'18,TACAS'19,QEST'19

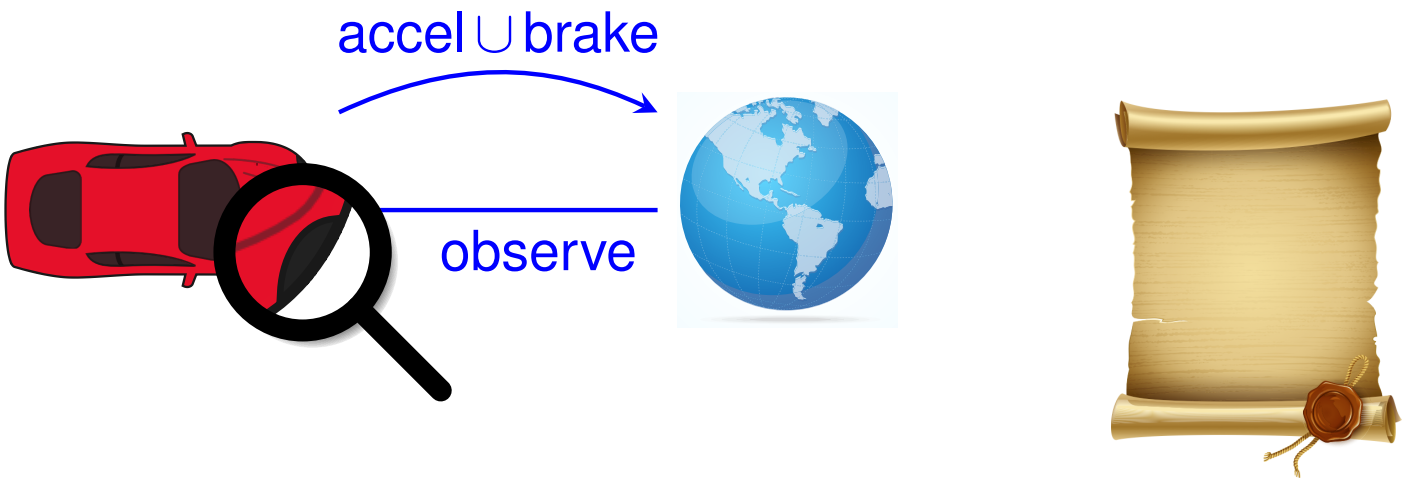
Learning to Act Safely in a CPS with Multiple Models



Convergence Plausible models converge to true model a.s., if possible

AAAI'18,ITC'18,TACAS'19,QEST'19

Learning to Act Safely in a CPS with Multiple Models



Modify model to fit observations by verification-preserving model update.
Safety proofs reified: modify model + proof tactic to preserve fit + safety

AAAI'18,ITC'18,TACAS'19,QEST'19

André Platzer.
Logics of dynamical systems.
In LICS [19], pages 13–24.
doi:10.1109/LICS.2012.13.

André Platzer.
Logical Foundations of Cyber-Physical Systems.
Springer, Cham, 2018.
doi:10.1007/978-3-319-63588-0.

André Platzer.
A complete uniform substitution calculus for differential dynamic logic.
J. Autom. Reas., 59(2):219–265, 2017.
doi:10.1007/s10817-016-9385-1.

André Platzer.
The complete proof theory of hybrid systems.
In LICS [19], pages 541–550.
doi:10.1109/LICS.2012.64.

André Platzer and Yong Kiam Tan.

Differential equation invariance axiomatization.
J. ACM, 67(1):6:1–6:66, 2020.
doi:10.1145/3380825.

André Platzer.
Logic & proofs for cyber-physical systems.
In Nicola Olivetti and Ashish Tiwari, editors, *IJCAR*, volume 9706 of *LNCS*, pages 15–21, Cham, 2016. Springer.
doi:10.1007/978-3-319-40229-1_3.

André Platzer.
Differential dynamic logic for hybrid systems.
J. Autom. Reas., 41(2):143–189, 2008.
doi:10.1007/s10817-008-9103-8.

André Platzer.
Differential-algebraic dynamic logic for differential-algebraic programs.
J. Log. Comput., 20(1):309–352, 2010.
doi:10.1093/logcom/exn070.

André Platzer.
The structure of differential invariants and differential cut elimination.

Log. Meth. Comput. Sci., 8(4:16):1–38, 2012.
doi:10.2168/LMCS-8(4:16)2012.

André Platzer and Yong Kiam Tan.
Differential equation axiomatization: The impressive power of differential ghosts.
In Anuj Dawar and Erich Grädel, editors, *LICS*, pages 819–828, New York, 2018. ACM.
doi:10.1145/3209108.3209147.

Jean-Baptiste Jeannin, Khalil Ghorbal, Yanni Kouskoulas, Aurora Schmidt, Ryan Gardner, Stefan Mitsch, and André Platzer.
A formally verified hybrid system for safe advisories in the next-generation airborne collision avoidance system.
STTT, 19(6):717–741, 2017.
doi:10.1007/s10009-016-0434-1.

Stefan Mitsch, Khalil Ghorbal, David Vogelbacher, and André Platzer.
Formal verification of obstacle avoidance and navigation of ground robots.
I. J. Robotics Res., 36(12):1312–1340, 2017.

doi:10.1177/0278364917733549.


Nathan Fulton and André Platzer.
Safe reinforcement learning via formal methods: Toward safe control through proof and learning.
In Sheila A. McIlraith and Kilian Q. Weinberger, editors, *AAAI*, pages 6485–6492. AAAI Press, 2018.


Nathan Fulton and André Platzer.
Verifiably safe off-model reinforcement learning.
In Tomas Vojnar and Lijun Zhang, editors, *TACAS, Part I*, volume 11427 of *LNCS*, pages 413–430. Springer, 2019.
doi:10.1007/978-3-030-17462-0_28.


Stefan Mitsch and André Platzer.
ModelPlex: Verified runtime validation of verified cyber-physical system models.
Form. Methods Syst. Des., 49(1-2):33–74, 2016.
Special issue of selected papers from RV'14.
doi:10.1007/s10703-016-0241-z.


Nathan Fulton, Stefan Mitsch, Brandon Bohrer, and André Platzer.

Bellerophon: Tactical theorem proving for hybrid systems.
In Mauricio Ayala-Rincón and César A. Muñoz, editors, *ITP*, volume 10499 of *LNCS*, pages 207–224. Springer, 2017.
doi:10.1007/978-3-319-66107-0_14.


André Platzer.
Logical Analysis of Hybrid Systems: Proving Theorems for Complex Dynamics.
Springer, Heidelberg, 2010.
doi:10.1007/978-3-642-14509-4.


André Platzer.
Differential game logic.
ACM Trans. Comput. Log., 17(1):1:1–1:51, 2015.
doi:10.1145/2817824.


Logic in Computer Science (LICS), 2012 27th Annual IEEE Symposium on, Los Alamitos, 2012. IEEE.