Logic of Autonomous Dynamical Systems

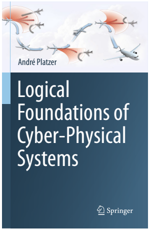
André Platzer

Karlsruhe Institute of Technology

Computer Science Department
Carnegie Mellon University

Summer School on Verification Technology, Systems & Applications 2022

http://keymaeraX.org/



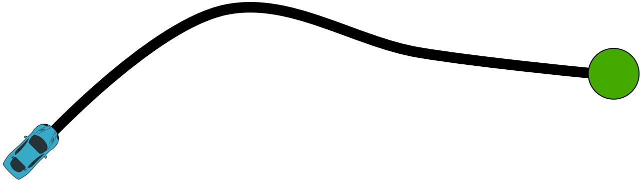
Unterstützt von / Supported by

Alexander von Humboldt
Stiftung / Foundation

Outline

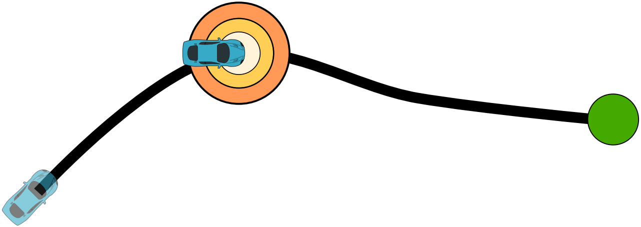
Outline

1. Autonomous Cyber-Physical Systems
2. Foundation: Differential Dynamic Logic
3. ModelPlex: Model Safety Transfer
4. VeriPhy: Executable Proof Transfer
5. Safe Learning in CPSs
6. Applications
   - Airborne Collision Avoidance System
   - Ground Robot Navigation
7. Summary



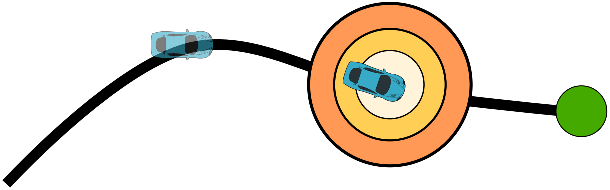
Cyber-Physical Systems

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



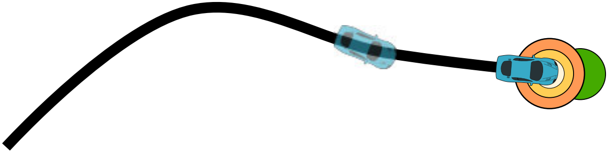
Cyber-Physical Systems

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



Cyber-Physical Systems

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



Cyber-Physical Systems

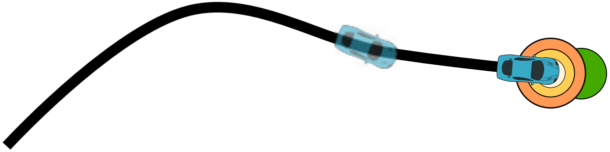
CPSs combine cyber capabilities with physical capabilities
to solve problems that neither part could solve alone.

CPS Analysis

- Simple control
- ODE model
- Strong predictions
- Nondet decisions

AI Learning

- Flexible responses
- “No” model*
- Hard to predict
- Optimal decision $(t\to\infty)$



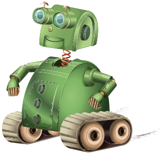
Cyber-Physical Systems

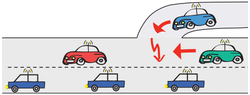
CPSs combine cyber capabilities with physical capabilities
to solve problems that neither part could solve alone.

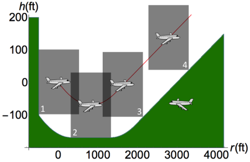
Autonomous CPSs Are on the Rise Everywhere

Prospects: Safety & Efficiency & Autonomy

Autonomous cars | Autonomous pilots | Robots near humans



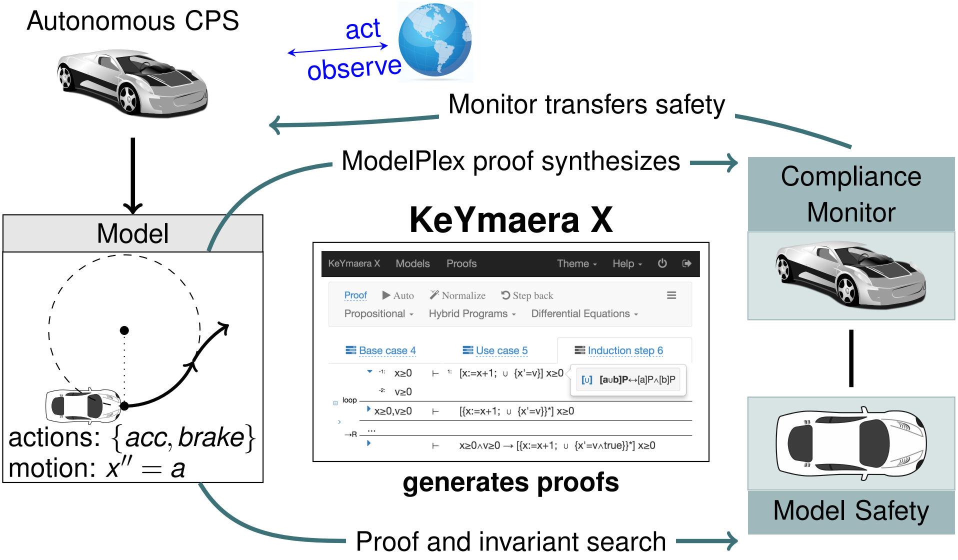




Objective

Best of both worlds: safety from CPS + flexibility from AI

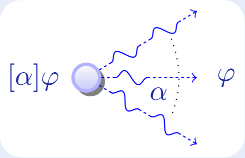
Approach: Safety Proofs for Autonomous CPS

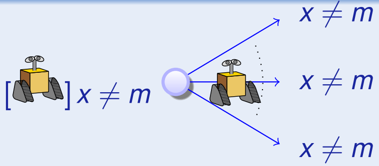


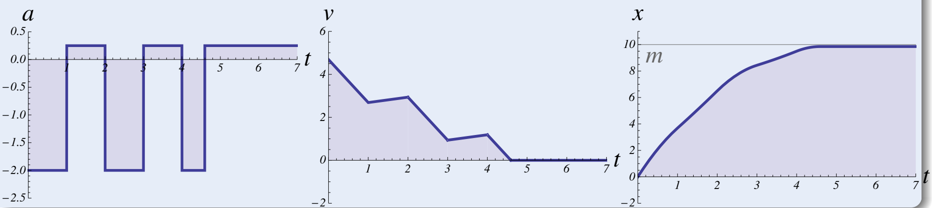
Outline

1. Autonomous Cyber-Physical Systems
2. Foundation: Differential Dynamic Logic
3. ModelPlex: Model Safety Transfer
4. VeriPhy: Executable Proof Transfer
5. Safe Learning in CPSs
6. Applications
   - Airborne Collision Avoidance System
   - Ground Robot Navigation
7. Summary

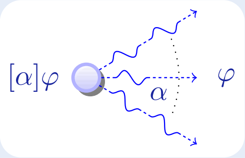
Concept (Differential Dynamic Logic) (JAR'08,LICS'12)

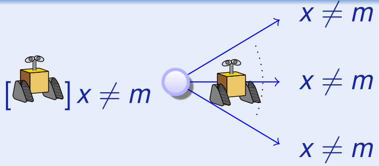




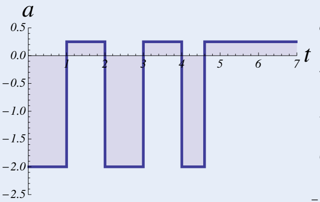


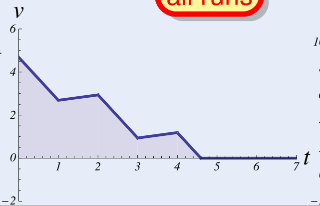
Concept (Differential Dynamic Logic) (JAR'08,LICS'12)

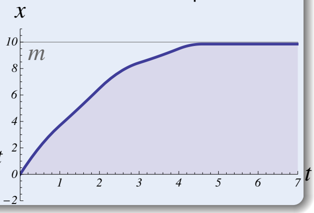




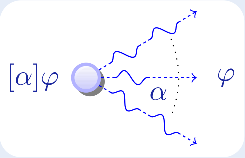
$$[\underbrace{((\text{if}(SB(x,m))\quad a := -b)\ ;\ x' = v, v' = a)^*}_{\text{all runs}}]\underbrace{x \neq m}_{\text{post}}$$

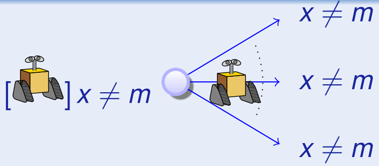




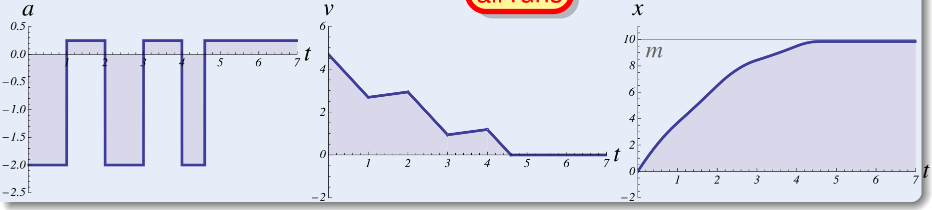


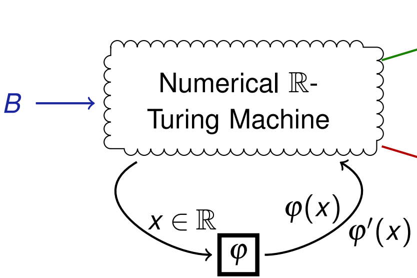
Concept (Differential Dynamic Logic) (JAR'08,LICS'12)



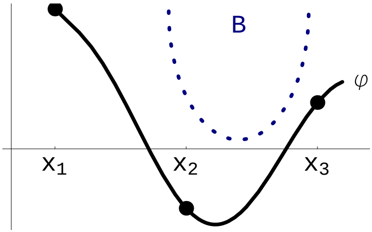


$$\underbrace{x \neq m \wedge b > 0}_{\text{init}} \rightarrow \Big[\underbrace{\big((\text{if}(SB(x,m)) \quad a := -b) \; ; \; x' = v, v' = a\big)^*}_{\text{all runs}}\Big] \underbrace{x \neq m}_{\text{post}}$$





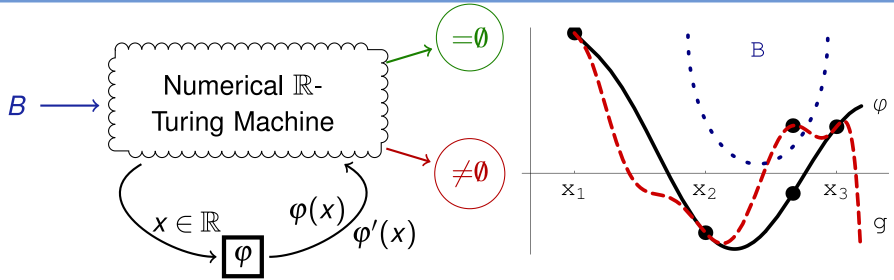
B
Numerical $\mathbb{R}$-Turing Machine
$=\emptyset$
$\neq\emptyset$
$x \in \mathbb{R}$
φ
$\varphi(x)$
$\varphi'(x)$



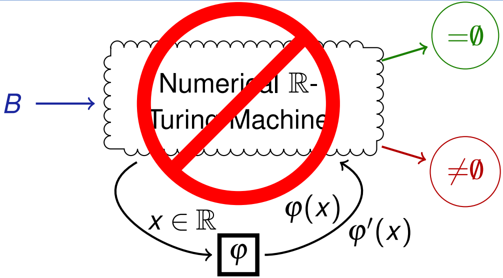
B
φ
x_1
x_2
x_3

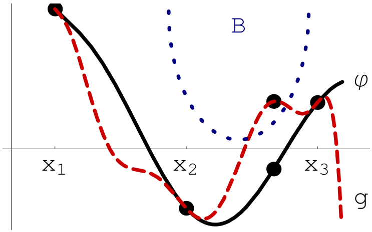
HSCC'07

Model Requirements



HSCC'07



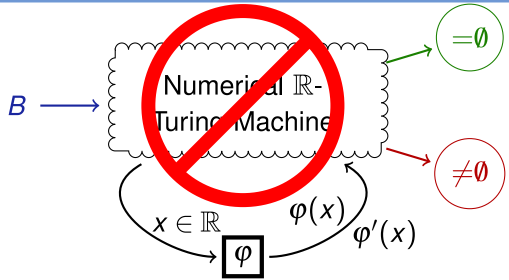


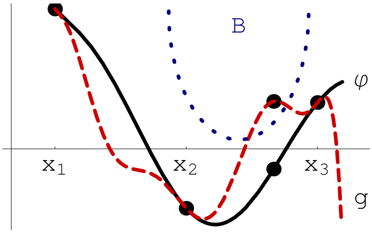
Proposition (Continuous image computation undecidable)

$\varphi(D) \cap B \overset{?}{=} \emptyset$ is undecidable by evaluating $\varphi(x)$ for

- *arbitrarily effective flow $\varphi \in C^k(D \subseteq \mathbb{R}^n, \mathbb{R}^m)$ with effective D, B*
- *even if tolerating error $\varepsilon > 0$ in decisions*

HSCC'07



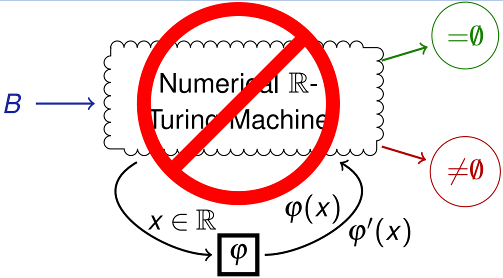


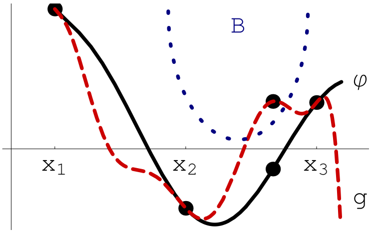
Proposition (Continuous image computation undecidable)

$\varphi(D) \cap B \stackrel{?}{=} \emptyset$ *is undecidable by evaluating* $\varphi(x)$ *for*

- *arbitrarily effective flow* $\varphi \in C^k(D \subseteq \mathbb{R}^n, \mathbb{R}^m)$ *with effective* D, B
- *even if tolerating error* $\varepsilon > 0$ *in decisions*
- *even* φ *smooth polynomial function with* $\mathbb{Q}$*-coefficients*
- *even in Blum-Shub-Smale “real Turing machines”*

HSCC’07





Proposition (Continuous image computation undecidable)

$\varphi(D) \cap B \overset{?}{=} \emptyset$ is undecidable by evaluating $\varphi(x)$ for

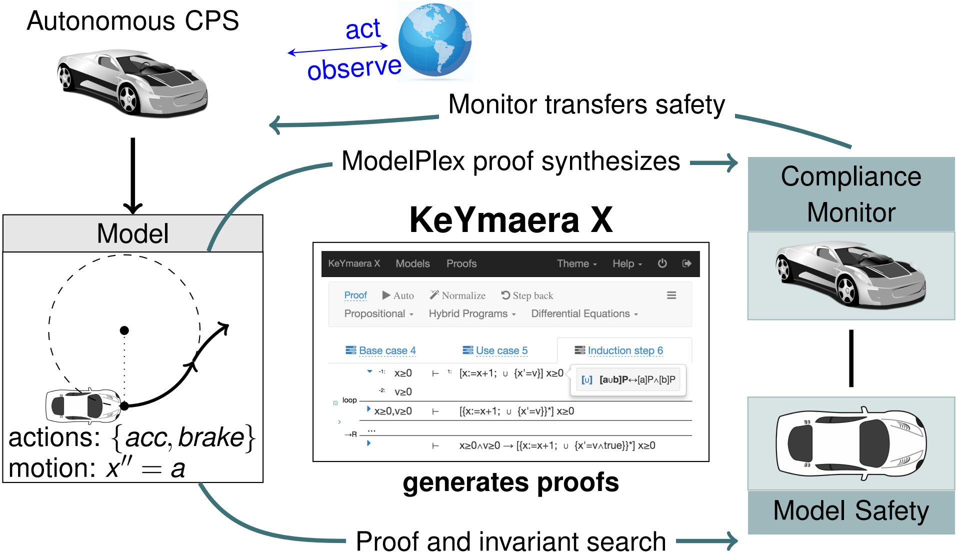
- *arbitrarily effective flow $\varphi \in C^k(D \subseteq \mathbb{R}^n, \mathbb{R}^m)$ with effective D, B*
- *even if tolerating error $\varepsilon > 0$ in decisions*
- *even φ smooth polynomial function with $\mathbb{Q}$-coefficients*
- *even in Blum-Shub-Smale "real Turing machines"*

The promise of "no model" is a myth

HSCC'07

Outline

Approach: Safety Proofs for Autonomous CPS

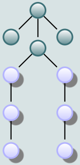


Formal Verification in CPS Development

Real CPS

safe

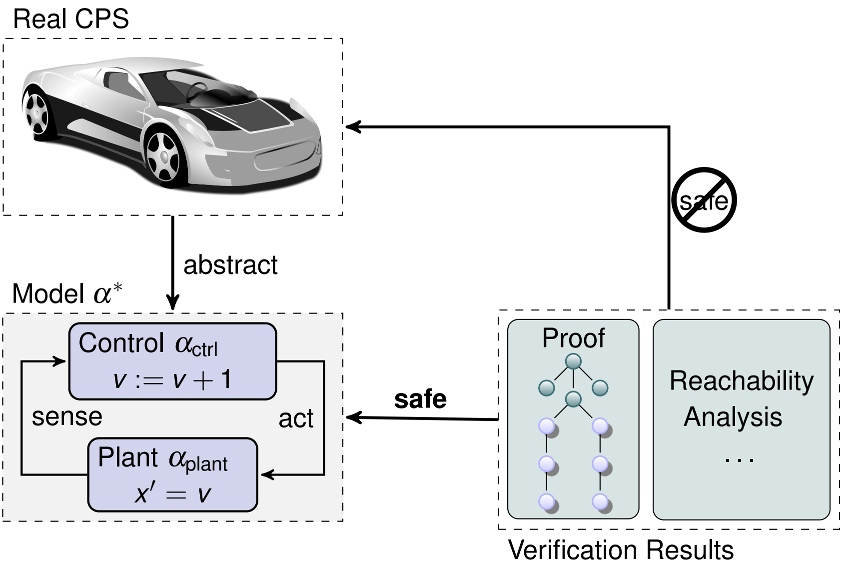
Proof



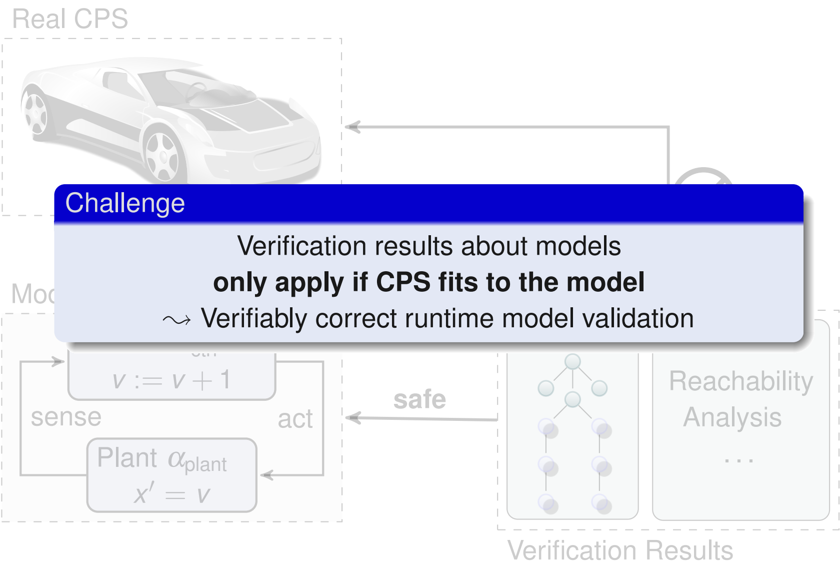
Reachability Analysis

...

Verification Results



Real CPS
abstract
Model α^*
Control α_{ctrl}
$v := v + 1$
sense
act
Plant α_{plant}
$x' = v$
safe
safe
Proof
Reachability Analysis
...
Verification Results



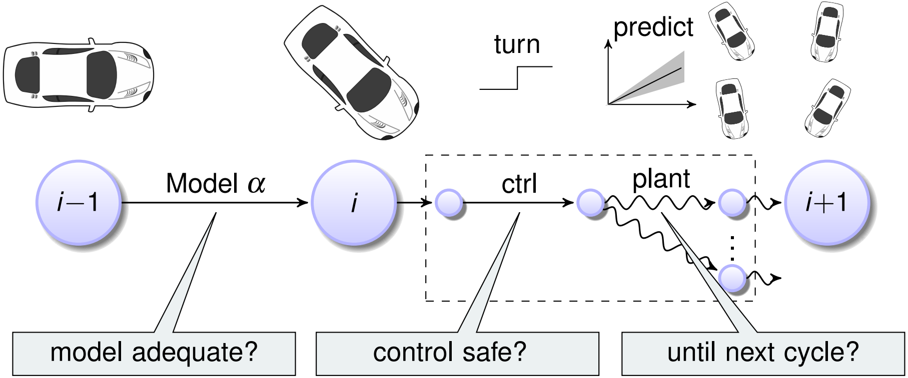
Challenge

Verification results about models

only apply if CPS fits to the model

$\rightsquigarrow$ Verifiably correct runtime model validation

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



RV'14,FMSD'16

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

Insights

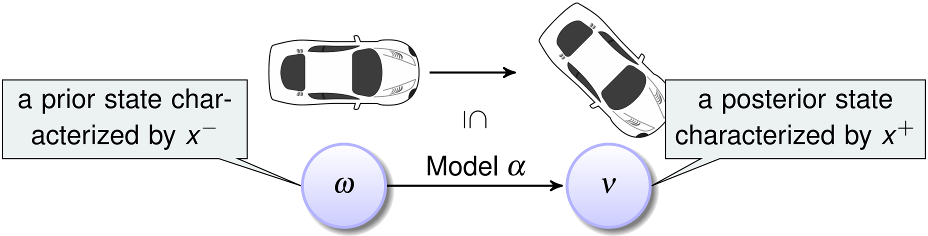
- Verification results about models transfer to the CPS when validating model compliance.
- Compliance with model is characterizable in logic dL.
- Compliance formula transformed by dL proof to monitor.
- Correct-by-construction provably correct model validation at runtime.

model adequate? | control safe? | until next cycle?

RV'14,FMSD'16

Characterizing State Relations in Logic

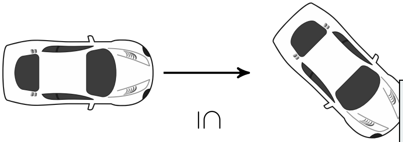
When are two states linked through a run of model α?



Semantical: $(\omega, \nu) \in [\![\alpha]\!]$ — reachability relation of α

RV'14,FMSD'16

When are two states linked through a run of model α?



a prior state characterized by x^-

$\subseteq$

$$\omega \xrightarrow{\text{Model } \alpha} \nu$$

a posterior state characterized by x^+

Offline

Semantical: $(\omega, \nu) \in [\![\alpha]\!]$

$\Updownarrow$ Lemma

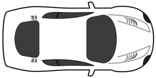
Logical dL: $(\omega, \nu) \models \langle \alpha \rangle (x = x^+)$

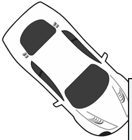
exists a run of α to a state where $x = x^+$

RV'14,FMSD'16

Characterizing State Relations in Logic

When are two states linked through a run of model α?





$\subseteq$

a prior state characterized by x^-

$$\omega \xrightarrow{\text{Model } \alpha} \nu$$

a posterior state characterized by x^+

Offline

Semantical: $(\omega, \nu) \in [\![\alpha]\!]$

$\Updownarrow$ Lemma

Logical dL: $(\omega, \nu) \models \langle\alpha\rangle(x = x^+)$

exists a run of α to a state where $x = x^+$

$\Updownarrow$ dL proof

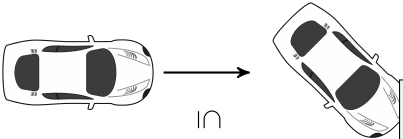
Arithmetical: $(\omega, \nu) \models F(x^-, x^+)$

check at runtime (efficient)

RV'14,FMSD'16

Characterizing State Relations in Logic

When are two states linked through a run of model α?



$\cap\!|$

a prior state characterized by x^-

a posterior state characterized by x^+

$$\omega \xrightarrow{\text{Model } \alpha} \nu$$

Offline

Semantical: $(\omega, \nu) \in [\![\alpha]\!]$

$\Updownarrow$ Lemma

Logical dL: $(\omega, \nu) \models \langle \alpha \rangle (x = x^+)$

exists a run of α to a state where $x = x^+$

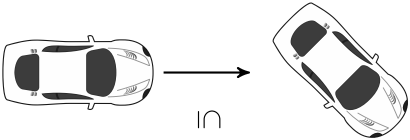
$\Uparrow$ dL proof

Arithmetical: $(\omega, \nu) \models F(x^-, x^+)$

check at runtime (efficient)

RV'14,FMSD'16

Logic reduces CPS safety to runtime monitor with offline proof



dL proof $A \to [\alpha]S$ $\quad \omega \xrightarrow{\text{Model } \alpha} \nu$

Offline

Init $\omega \in [\![A]\!]$ $\qquad$ Safe $\nu \in [\![S]\!]$

Semantical: $(\omega, \nu) \in [\![\alpha]\!]$

$\Updownarrow$ Lemma

Logical dL: $(\omega, \nu) \models \langle\alpha\rangle(x = x^+)$

$\Uparrow$ dL proof

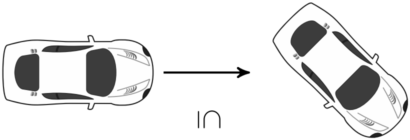
Arithmetical: $(\omega, \nu) \models F(x^-, x^+)$

check at runtime (efficient)

RV'14,FMSD'16

Logical Reductions for Model Safety Transfer

Logic reduces CPS safety to runtime monitor with offline proof



$\in$

dL proof $A \rightarrow [\alpha]S$ $\quad \omega \xrightarrow{\text{Model } \alpha} \nu$

Init $\omega \in [\![A]\!]$ $\qquad$ Safe $\nu \in [\![S]\!]$

Offline

Semantical: $(\omega, \nu) \in [\![\alpha]\!]$

$\Updownarrow$ Lemma

Logical dL: $(\omega, \nu) \models \langle \alpha \rangle (x = x^+)$

$\Uparrow$ dL proof

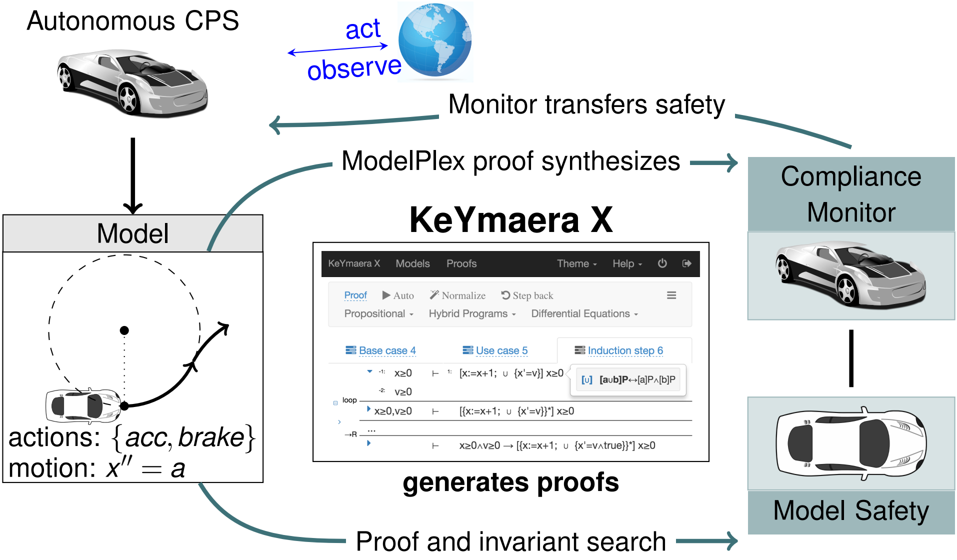
Arithmetical: $(\omega, \nu) \models F(x^-, x^+)$

check at runtime (efficient)

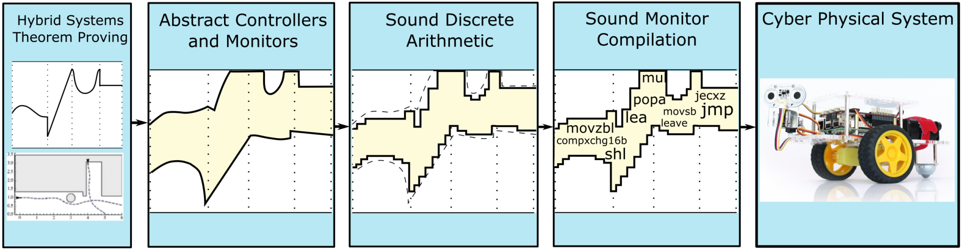
RV'14,FMSD'16

Outline

Approach: Safety Proofs for Autonomous CPS

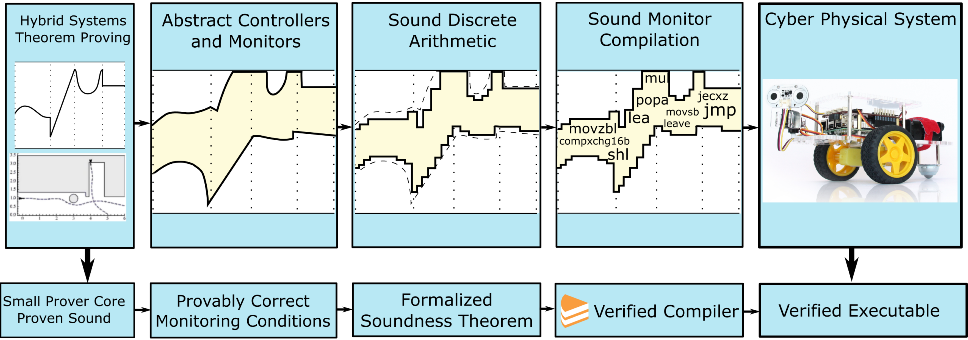


VeriPhy: Automatic, Verified EXEs from Controllers



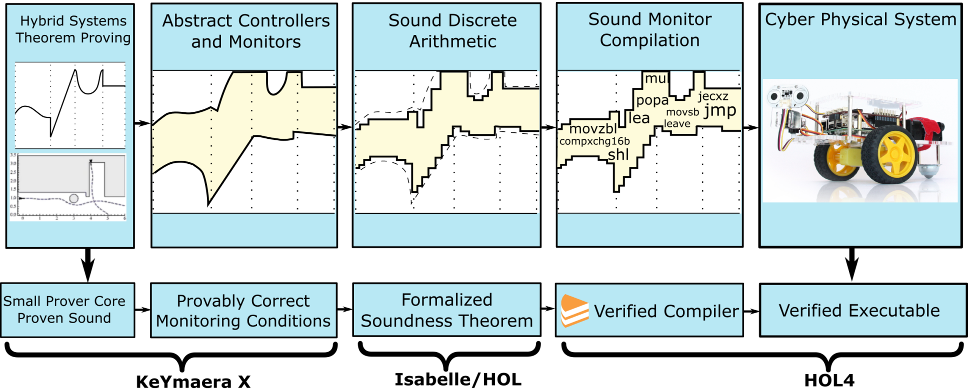
PLDI'18

VeriPhy: Automatic, Verified EXEs from Controllers



PLDI'18

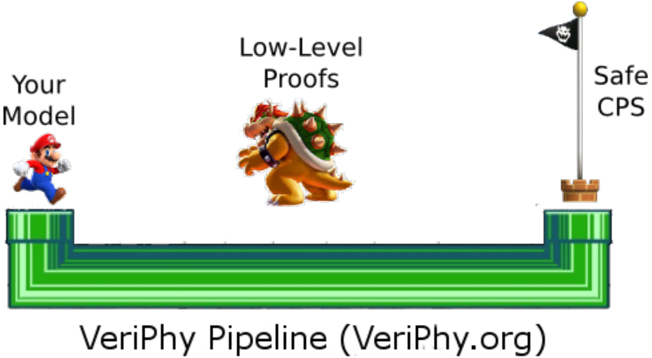
VeriPhy: Automatic, Verified EXEs from Controllers



PLDI'18

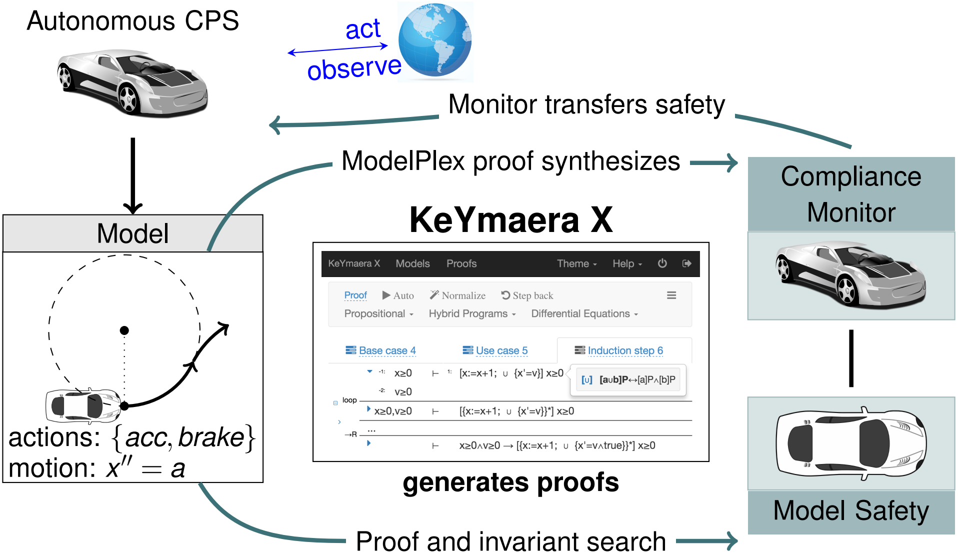
PLDI'18

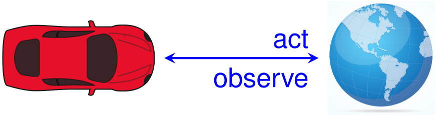
VeriPhy: Takeaway Metaphor



PLDI'18

Outline

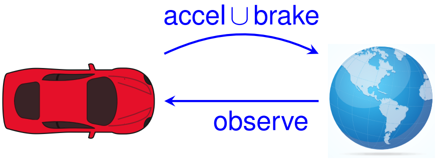

Autonomous CPS
act
observe
Monitor transfers safety
ModelPlex proof synthesizes
Compliance Monitor
Model
KeYmaera X
KeYmaera X Models Proofs Theme Help
Proof Auto Normalize Step back
Propositional Hybrid Programs Differential Equations
Base case 4 Use case 5 Induction step 6
-1: x≥0 ⊢ 1: [x:=x+1; ∪ {x'=v}] x≥0
-2: v≥0
[∪] [a∪b]P↔[a]P∧[b]P
loop
x≥0,v≥0 ⊢ [{x:=x+1; ∪ {x'=v}}*] x≥0
→R ...
⊢ x≥0∧v≥0 → [{x:=x+1; ∪ {x'=v∧true}}*] x≥0
generates proofs
actions: $\{acc, brake\}$
motion: $x'' = a$
Model Safety
Proof and invariant search



Reinforcement Learning learns from experience of trying actions

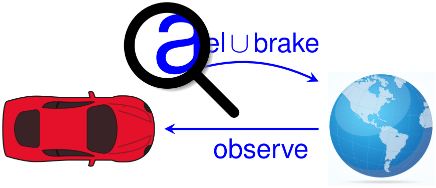
AAAI'18,ITC'18,TACAS'19,QEST'19

Learning to Act in a CPS



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

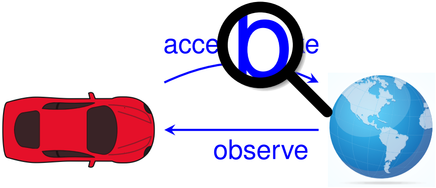
AAAI'18,ITC'18,TACAS'19,QEST'19



ModelPlex monitor inspects each decision, vetoes if unsafe

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

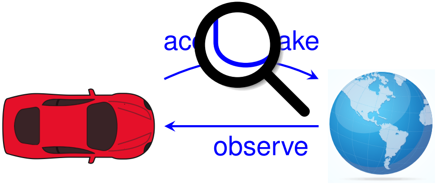
Learning to Act Safely in a CPS



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

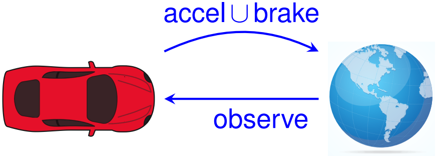
Learning to Act Safely in a CPS



dL benefits from RL optimization.

RL benefits from dL safety signal.

AAAI’18,ITC’18,TACAS’19,QEST’19

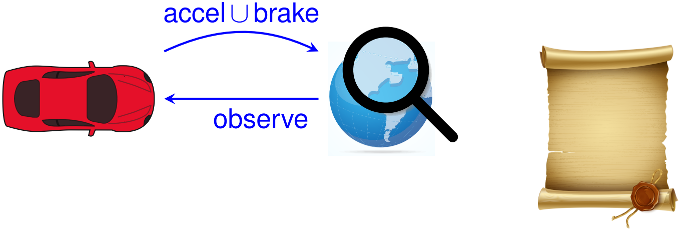


Theorem Safe policy if ODE accurate

Experiment Graceful recovery outside ODE ⇜ quantitative ModelPlex

Detect modeled versus unmodeled state space ⇜ ModelPlex

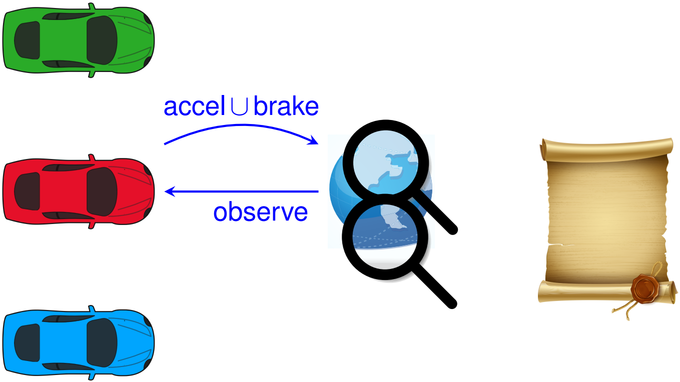
AAAI'18,ITC'18,TACAS'19,QEST'19



What's safe when off model?

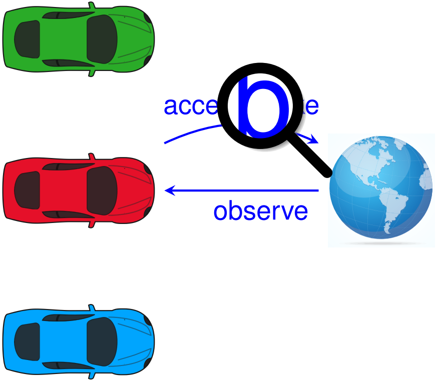
AAAI'18,ITC'18,TACAS'19,QEST'19

Learning to Act Safely in a CPS with Multiple Models



What's safe with multiple possible models?

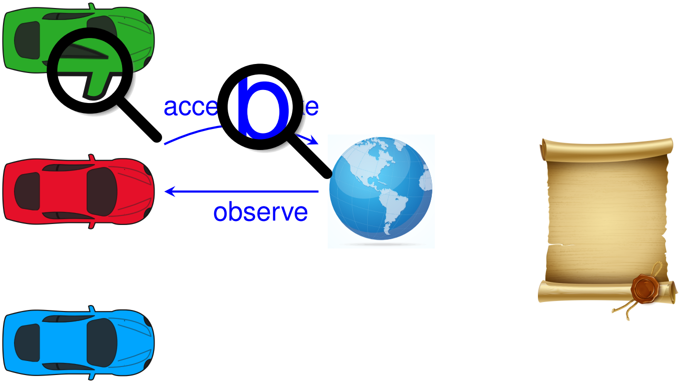
AAAI'18,ITC'18,TACAS'19,QEST'19



ModelPlex monitors conjunction of all plausible models

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

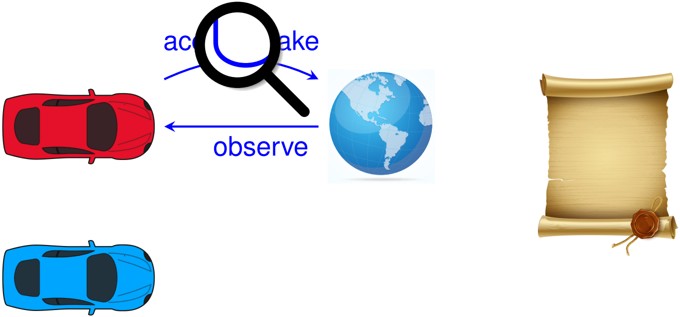
Learning to Act Safely in a CPS with Multiple Models



Remove incompatible models after contradictory observation

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

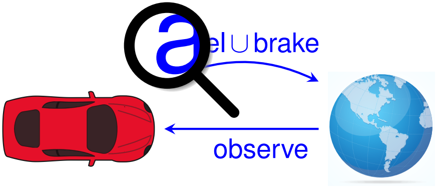
Learning to Act Safely in a CPS with Multiple Models



Plan differentiating experiment $\leadsto$ predictive monitor distinctions

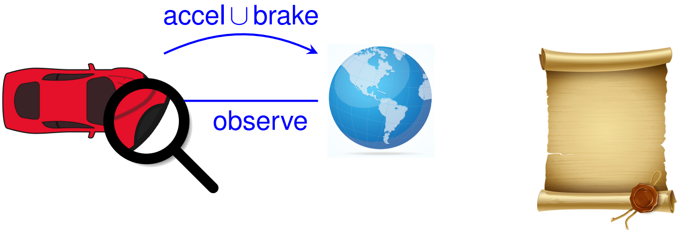
AAAI'18,ITC'18,TACAS'19,QEST'19

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



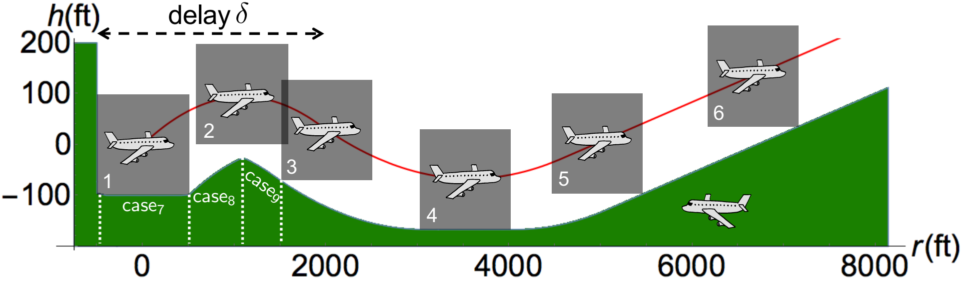
Modify model to fit observations by verification-preserving model update.
Safety proofs reified: modify model + proof tactic to preserve fit + safety

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

Outline

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

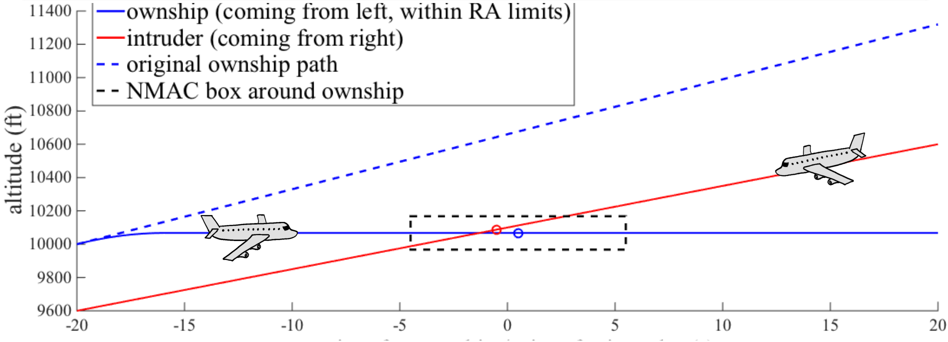


1. Identified safe region for each advisory symbolically
2. Proved safety for hybrid systems flight model in KeYmaera X

STTT'17

Airborne Collision Avoidance System ACAS X: Compare

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

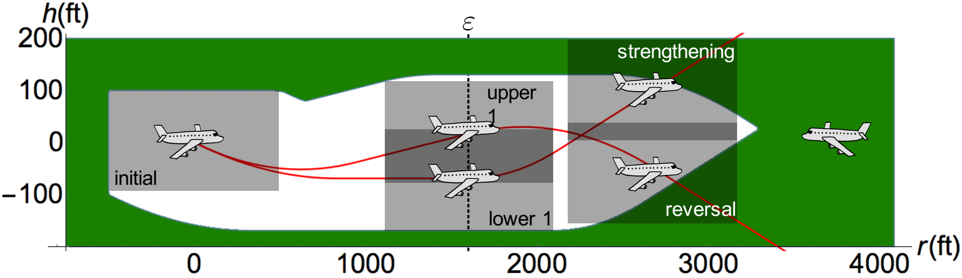


ACAS X issues DNC advisory, which induces collision unless corrected

STTT'17

Airborne Collision Avoidance System ACAS X: Refine

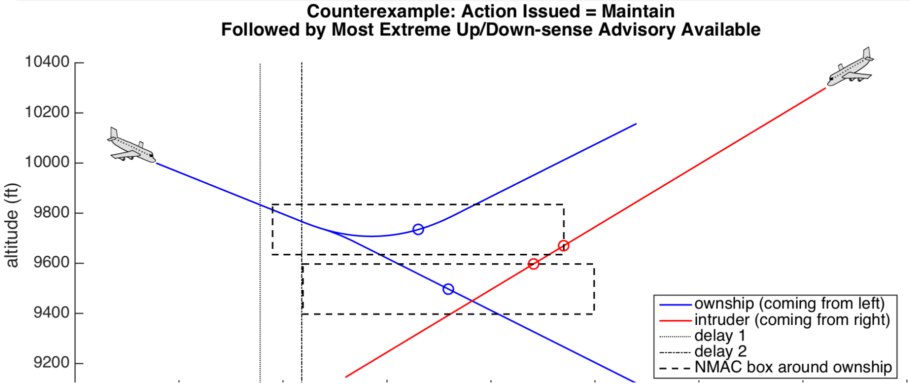
- Conservative, so too many counterexamples
- Settle for: safe for a little while, with safe future advisory possibility
- Safeable advisory: a subsequent advisory can safely avoid collision



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

STTT'17

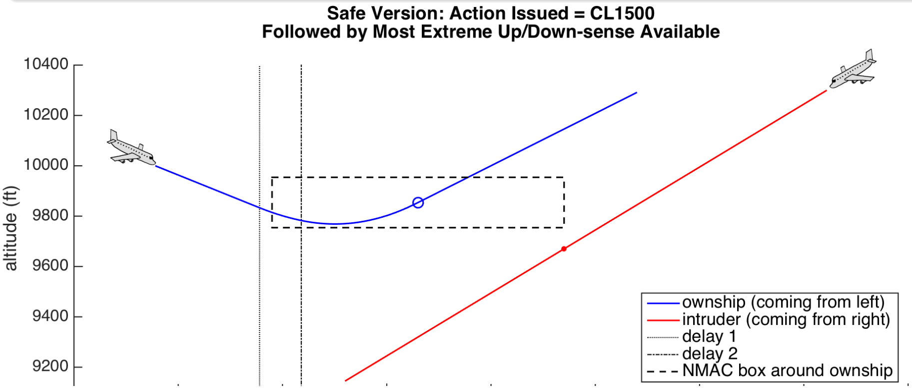
ACAS X table comparison shows safeable advisory in more of the 648,591,384,375 states compared (≈899 10^6 counterexamples).



ACAS X issues Maintain advisory instead of CL1500

STTT'17

ACAS X table comparison shows safeable advisory in more of the 648,591,384,375 states compared ($\approx$899 10^6 counterexamples).

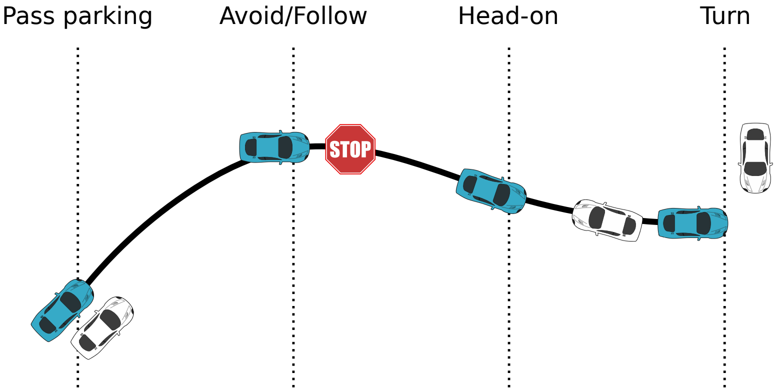


ACAS X issues Maintain advisory instead of CL1500

STTT'17

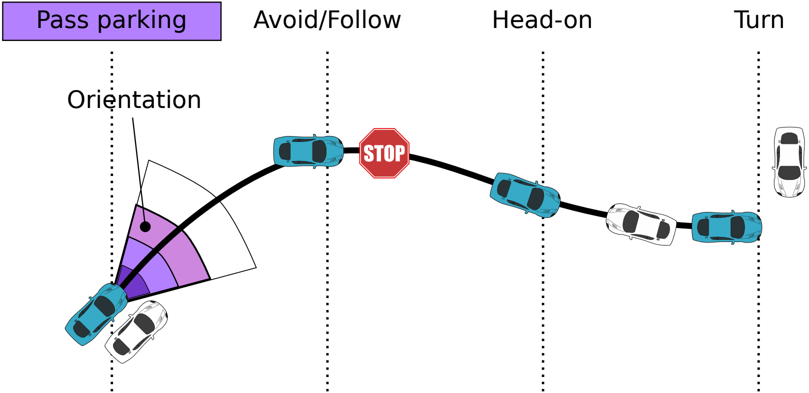
Ground Robot Obstacle Avoidance: Verify

- 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

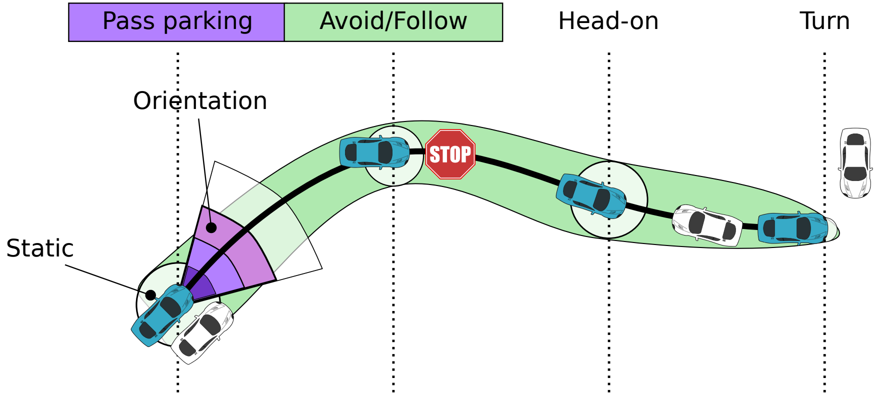
- Fundamental safety question for ground robot navigation IJRR'17
- When will which control decision avoid obstacles?
- Depends on safety objective, physical capabilities of robot + obstacle



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

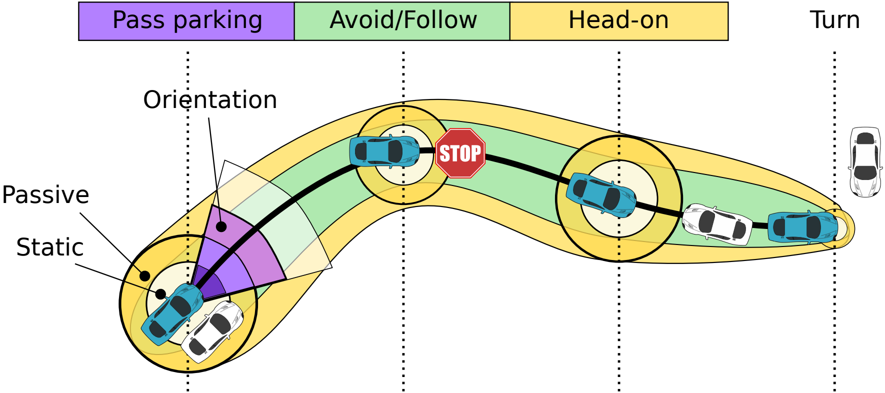
- Fundamental safety question for ground robot navigation IJRR'17
- When will which control decision avoid obstacles?
- Depends on safety objective, physical capabilities of robot + obstacle



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

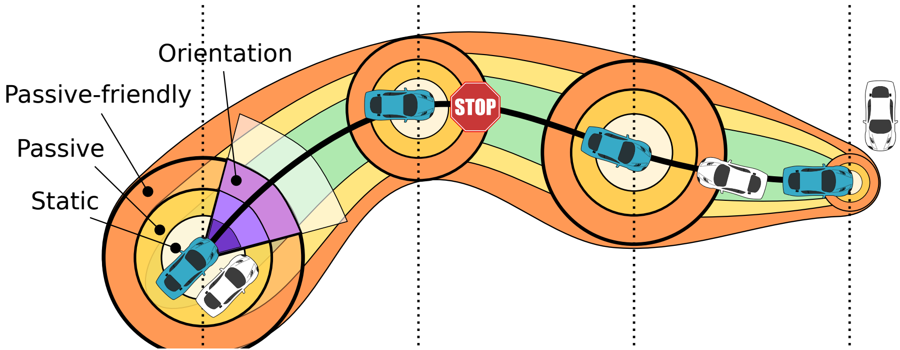


1. Identified safe region for each safety notion symbolically
2. Proved safety for hybrid systems ground robot model in KeYmaera X

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

Pass parking	Avoid/Follow	Head-on	Turn



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	$\|p-o\|_\infty > \frac{s^2}{2b} + \frac{V^2}{2b_o} + V\left(\frac{s}{b}+\tau\right) + \left(\frac{A}{b}+1\right)\left(\frac{A}{2}\varepsilon^2+\varepsilon(s+V)\right)$

$\vdots$

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	[illegible] $V)\Big) + \Delta_p$
+ disturb.	$\|p-o\|_\infty >$ [illegible] $2b\Delta_a$ $+ V$ [illegible] $b\Delta_a$ $+ \Big($ [illegible] $b\Delta_a$ $+1\Big)\Big($ [illegible] 2 $\varepsilon^2+\varepsilon(s+V)\Big)$
+ 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

Outline

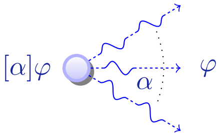
Acknowledgments

Logic of Autonomous Dynamical Systems, Karlsruhe Institute of Technology
Logical Systems lab, Carnegie Mellon University, Computer Science
Yong Kiam Tan, Brandon Bohrer, Nathan Fulton, Sarah Loos, Katherine Cordwell
Stefan Mitsch, Khalil Ghorbal, Jean-Baptiste Jeannin, Andrew Sogokon

Logic & Proofs for Autonomous Cyber-Physical Systems

differential dynamic logic

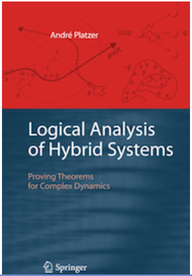
$\mathsf{dL} = \mathsf{DL} + \mathsf{HP}$



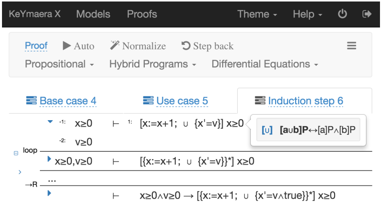
Logical Triumvirate of Technologies for Transitioning Trustworthiness

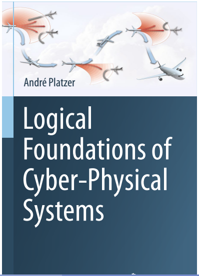
1. KeYmaera X: safe action in CPS model
2. ModelPlex: safe model $\rightsquigarrow$ safe impl
3. VeriPhy: sandbox $\rightsquigarrow$ safe executable

1. RL optimizes action choice
2. ModelPlex: safe reward for RL
3. VeriPhy: CPS sandbox for RL



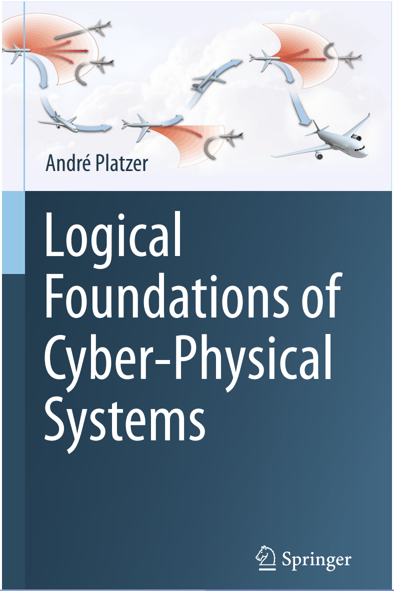
KeYmaera X

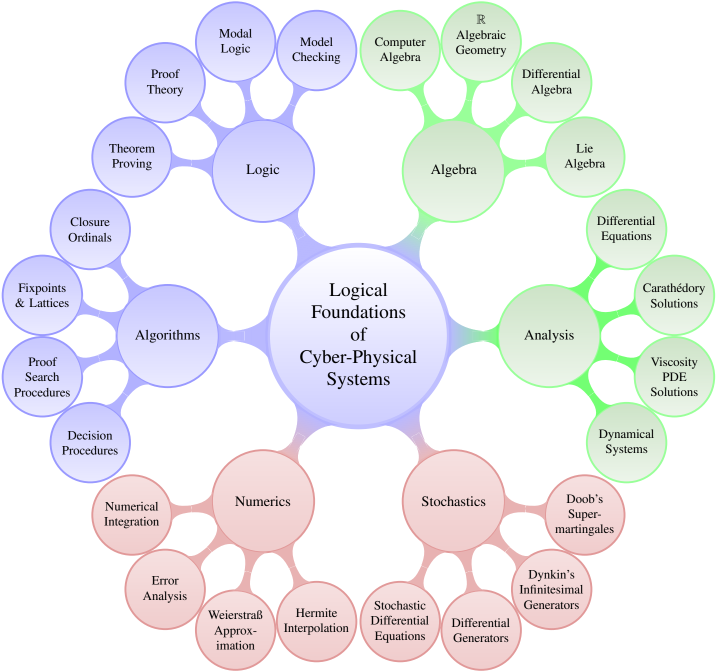


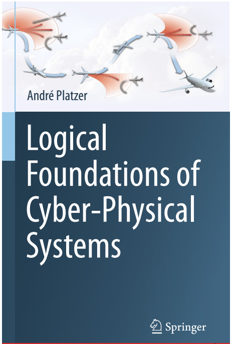


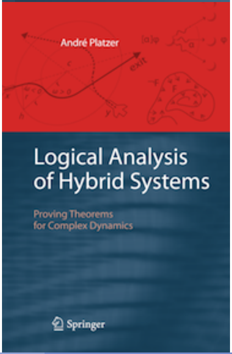
A. Platzer. *Logical Foundations of Cyber-Physical Systems*. Springer 2018

- **I Part: Elementary Cyber-Physical Systems**
- 2. Differential Equations & Domains
- 3. Choice & Control
- 4. Safety & Contracts
- 5. Dynamical Systems & Dynamic Axioms
- 6. Truth & Proof
- 7. Control Loops & Invariants
- 8. Events & Responses
- 9. Reactions & Delays
- **II Part: Differential Equations Analysis**
- 10. Differential Equations & Differential Invariants
- 11. Differential Equations & Proofs
- 12. Ghosts & Differential Ghosts
- 13. Differential Invariants & Proof Theory
- **III Part: Adversarial Cyber-Physical Systems**
- 14-17. Hybrid Systems & Hybrid Games
- **IV Part: Comprehensive CPS Correctness**




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
Carathéodory 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
Differential Generators
Stochastic Differential Equations


André Platzer
Logical Foundations of Cyber-Physical Systems
Springer


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

Differential Dynamic Logic dL: Syntax

Definition (Hybrid program α)

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

Definition (dL Formula P)

$$e \geq \tilde{e} \mid \neg P \mid P \land Q \mid \forall x\, P \mid \exists x\, P \mid [\alpha]P \mid \langle \alpha \rangle P$$

JAR'08,LICS'12,JAR'17

Differential Dynamic Logic dL: Syntax

Discrete Assign | Test Condition | Differential Equation | Nondet. Choice | Seq. Compose | Nondet. Repeat

Definition (Hybrid program α)

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

Definition (dL Formula P)

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

All Reals | Some Reals | All Runs | Some Runs

JAR'08,LICS'12,JAR'17

Definition (Hybrid program semantics) $(\llbracket\cdot\rrbracket : \mathrm{HP} \to \wp(\mathscr{S} \times \mathscr{S}))$

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

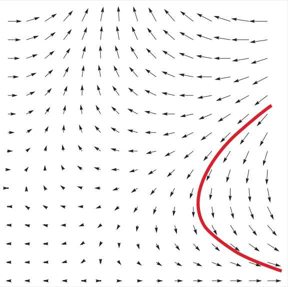
compositional semantics

Definition (dL semantics) $(\llbracket\cdot\rrbracket : \mathrm{Fml} \to \wp(\mathscr{S}))$

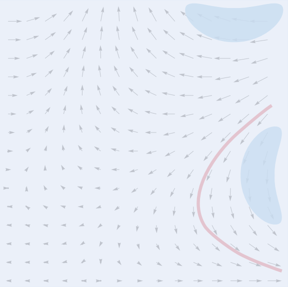
$$
\begin{aligned}
\llbracket e \geq \tilde{e} \rrbracket &= \{\omega : \omega\llbracket e \rrbracket \geq \omega\llbracket \tilde{e} \rrbracket\} \\
\llbracket \neg P \rrbracket &= \llbracket P \rrbracket^\complement \\
\llbracket P \wedge Q \rrbracket &= \llbracket P \rrbracket \cap \llbracket Q \rrbracket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

Differential Invariants for Differential Equations

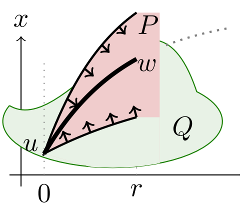
Differential Invariant



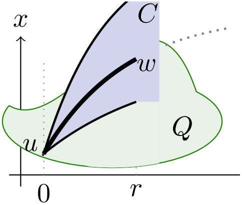
Differential Cut



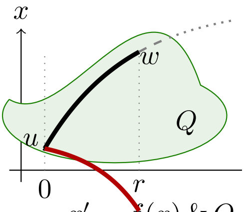
Differential Ghost



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



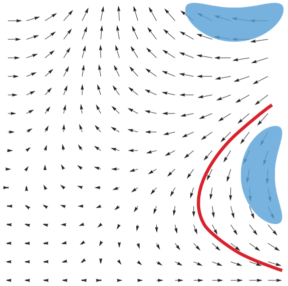
$x' = f(x) \,\&\, Q$

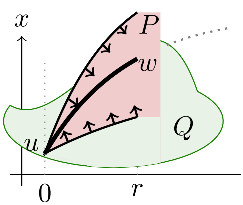


$x' = f(x) \,\&\, Q$

Differential Invariants for Differential Equations

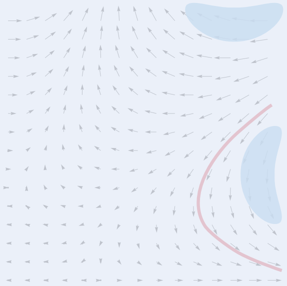
Differential Invariant

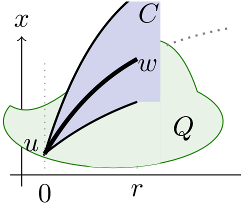




$$x' = f(x) \,\&\, Q$$

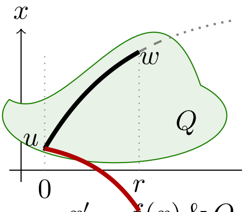
Differential Cut





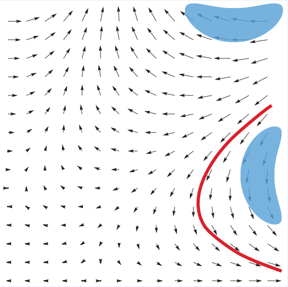
$$x' = f(x) \,\&\, Q$$

Differential Ghost

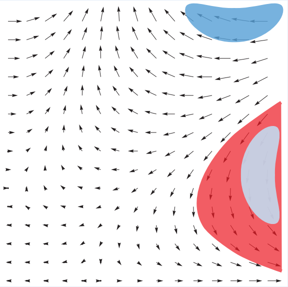


$$x' = f(x) \,\&\, Q$$

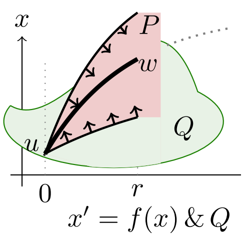
Differential Invariant

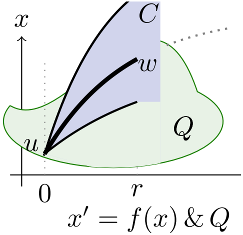


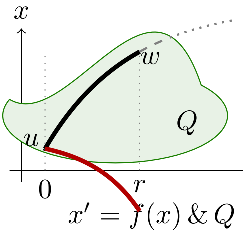
Differential Cut



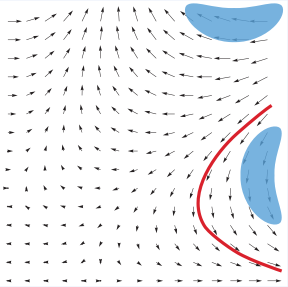
Differential Ghost



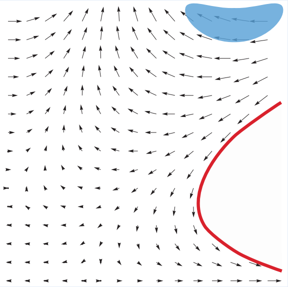




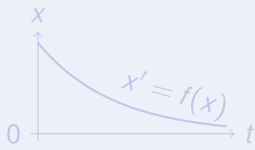
Differential Invariant

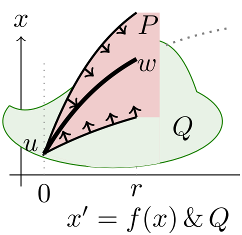


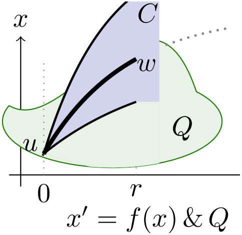
Differential Cut

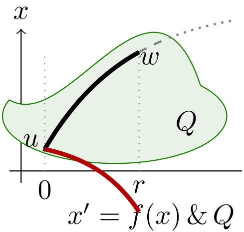


Differential Ghost

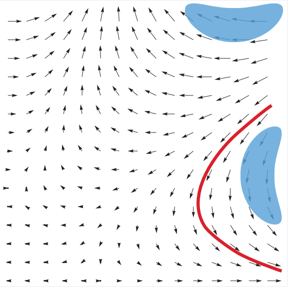




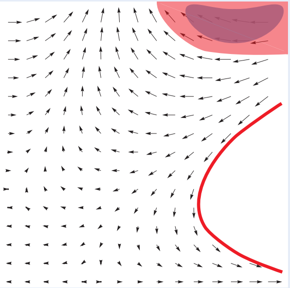




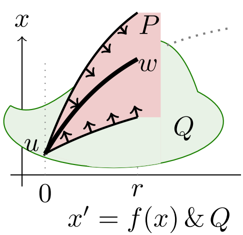
Differential Invariant

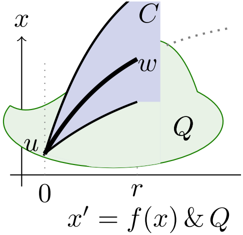


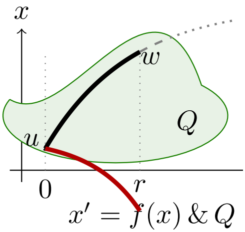
Differential Cut



Differential Ghost

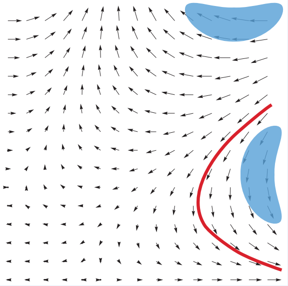




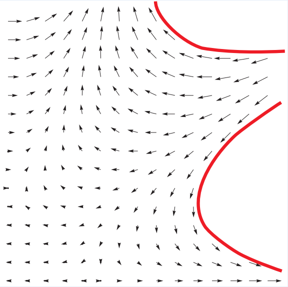


Differential Invariants for Differential Equations

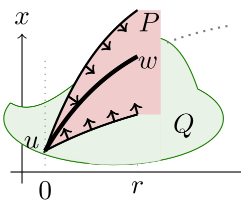
Differential Invariant

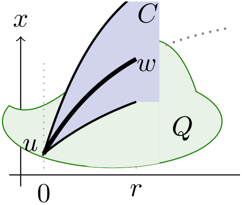


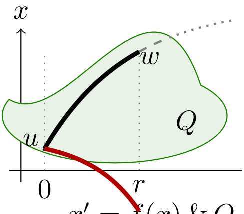
Differential Cut



Differential Ghost

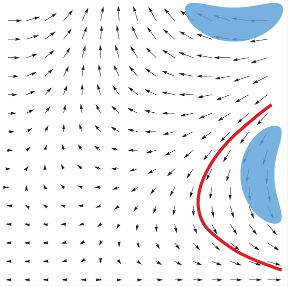




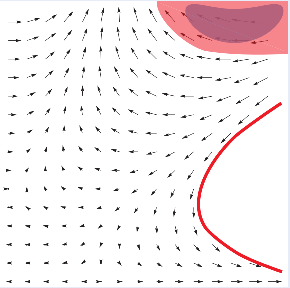


Differential Invariants for Differential Equations

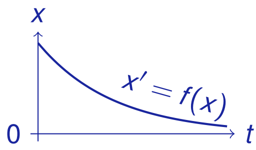
Differential Invariant

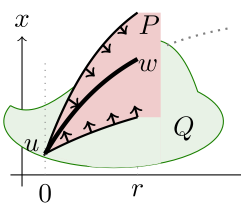


Differential Cut

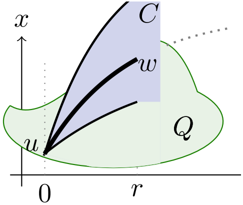


Differential Ghost

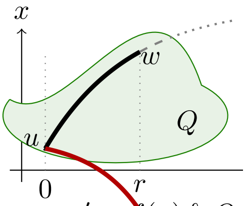




$x' = f(x) \,\&\, Q$



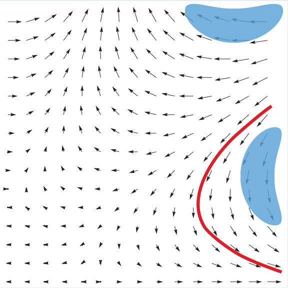
$x' = f(x) \,\&\, Q$



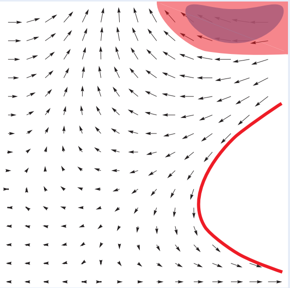
$x' = f(x) \,\&\, Q$

Differential Invariants for Differential Equations

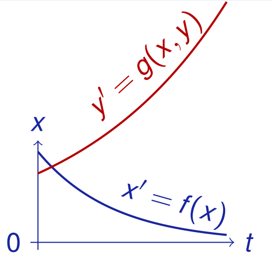
Differential Invariant

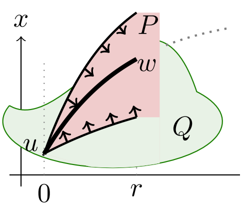


Differential Cut

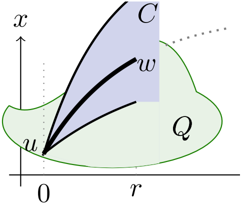


Differential Ghost

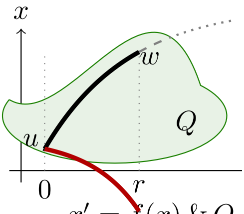




$x' = f(x) \,\&\, Q$

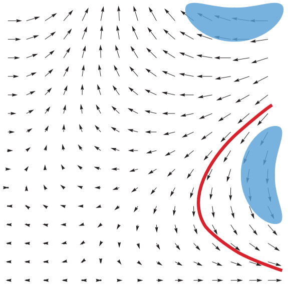


$x' = f(x) \,\&\, Q$

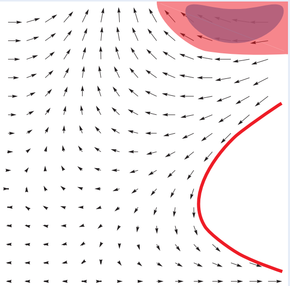


$x' = f(x) \,\&\, Q$

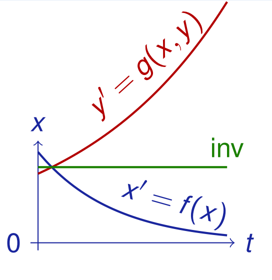
Differential Invariant

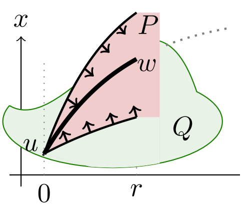


Differential Cut

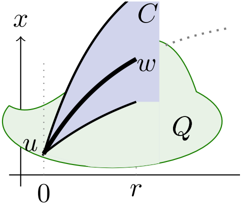


Differential Ghost

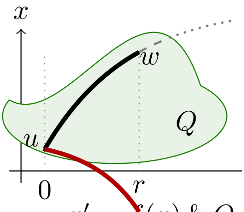




$x' = f(x) \,\&\, Q$



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



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

Differential Invariants for Differential Equations

Differential Invariant

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

Differential Cut

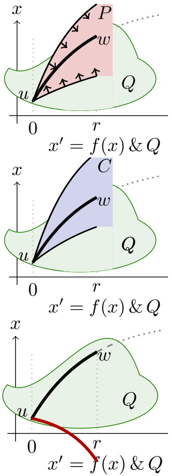
$$\frac{P \vdash [x' = f(x) \,\&\, Q]\textcolor{red}{C} \quad P \vdash [x' = f(x) \,\&\, Q \land \textcolor{red}{C}]P}{P \vdash [x' = f(x) \,\&\, Q]P}$$

Differential Ghost

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

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

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

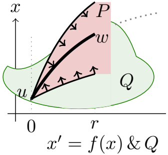


Springer'10, LMCS'12, LICS'12, JAR'17, LICS'18, JACM'20

Differential Invariants for Differential Equations

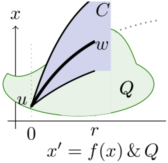
Differential Invariant

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



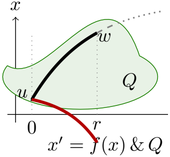
Differential Cut

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



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

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

André Platzer.
The logical path to autonomous cyber-physical systems.
In David Parker and Verena Wolf, editors, *QEST*, volume 11785 of *LNCS*, pages 25–33. Springer, 2019.
doi:10.1007/978-3-030-30281-8_2.

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.
Logics of dynamical systems.
In *LICS*, pages 13–24, Los Alamitos, 2012. IEEE.
doi:10.1109/LICS.2012.13.

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.

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 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.

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.

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-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.