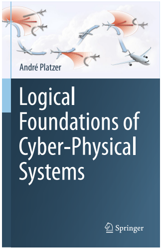
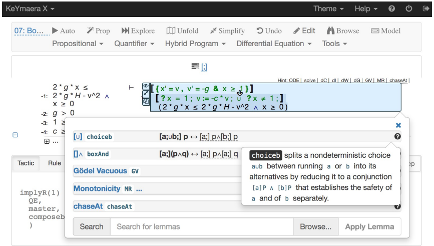
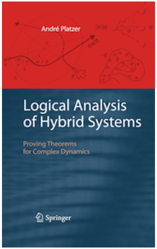
Cyber-Physical Systems Verification with KeYmaera X

André Platzer

Carnegie Mellon University







Outline

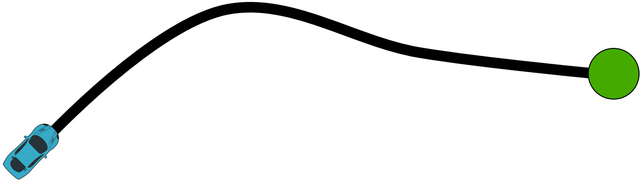
Outline

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

Cyber-Physical Systems Safety

Prospects: Safety & Efficiency

(Autonomous) cars Pilot support Robots near humans

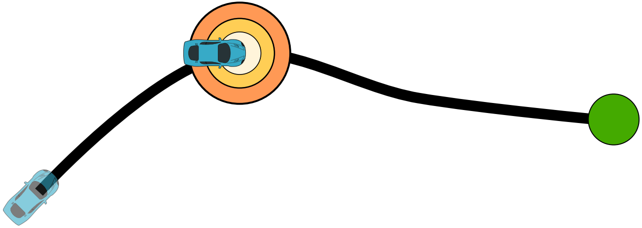


Cyber-Physical Systems

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

Prospects: Safety & Efficiency

(Autonomous) cars — Pilot support — Robots near humans

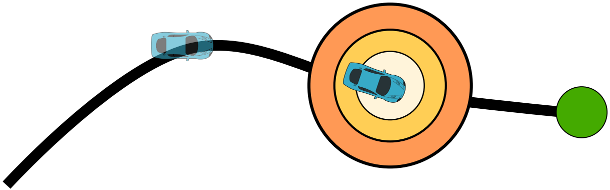


Cyber-Physical Systems

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

Prospects: Safety & Efficiency

(Autonomous) cars   Pilot support   Robots near humans



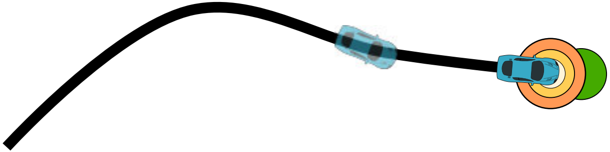
Cyber-Physical Systems

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

Cyber-Physical Systems Safety

Prospects: Safety & Efficiency

(Autonomous) cars Pilot support Robots near humans



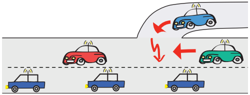
Cyber-Physical Systems

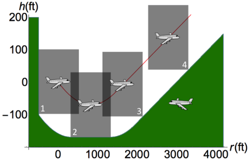
CPSs combine cyber capabilities with physical capabilities
to solve problems that neither part could solve alone.

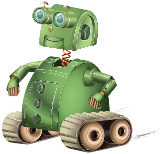
Cyber-Physical Systems Safety

Prospects: Safety & Efficiency

(Autonomous) cars | Pilot support | Robots near humans







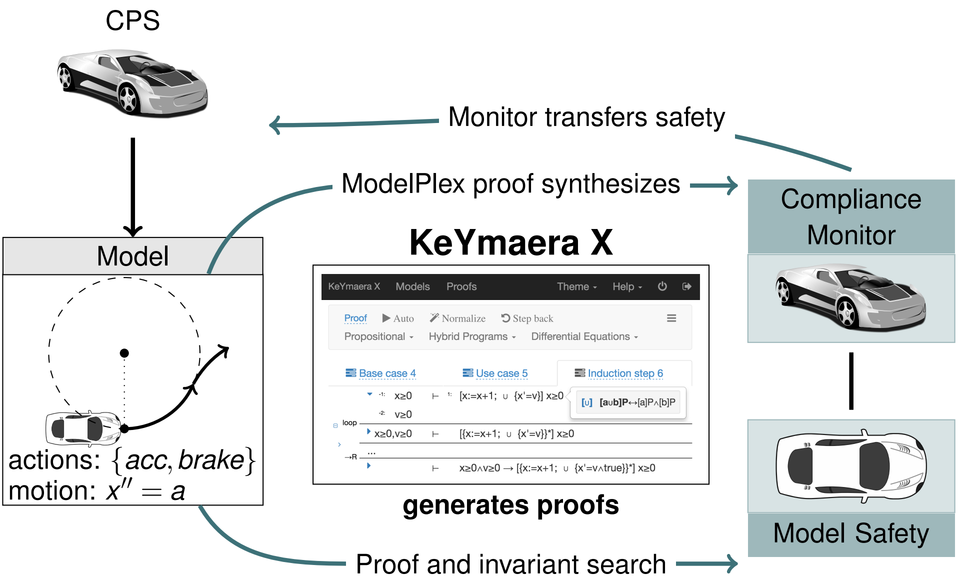
Cyber-Physical Systems

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

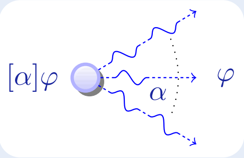
Outline

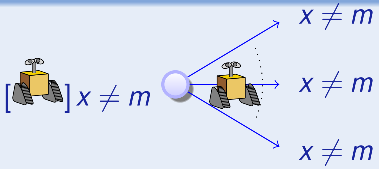
1. Cyber-Physical Systems
2. Foundation: Differential Dynamic Logic
3. ModelPlex: Model Safety Transfer
4. VeriPhy: Executable Proof Transfer
5. Applications
   - Airborne Collision Avoidance System
   - Safe Learning in CPS
6. Summary

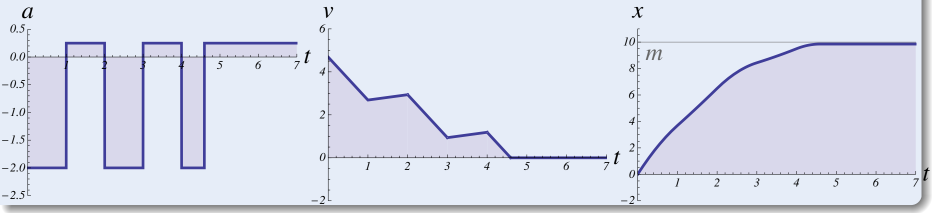
Approach: Proofs for Cyber-Physical Systems



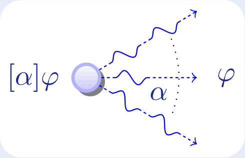
Concept (Differential Dynamic Logic) (JAR'08,LICS'12)

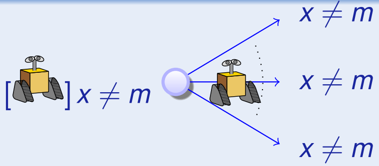




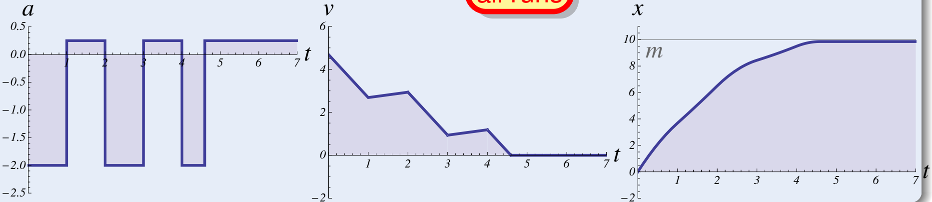


Concept (Differential Dynamic Logic) (JAR'08,LICS'12)

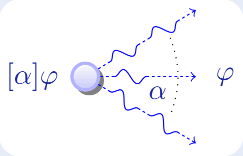


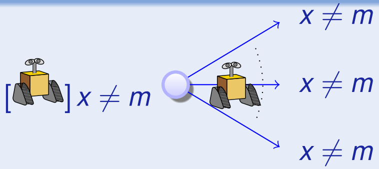


$$\underbrace{[((\text{if}(SB(x,m))\quad a:=-b)\ ;\ x'=v, v'=a)^*]}_{\text{all runs}}\ \underbrace{x \neq m}_{\text{post}}$$



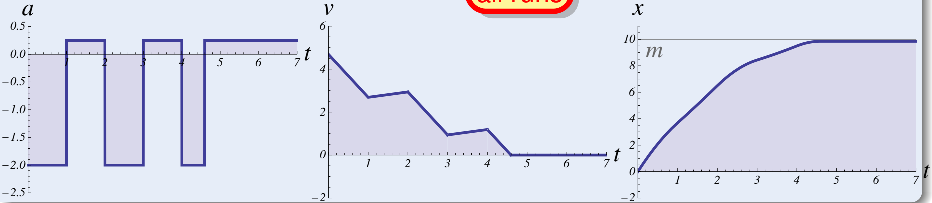
Concept (Differential Dynamic Logic) (JAR'08,LICS'12)





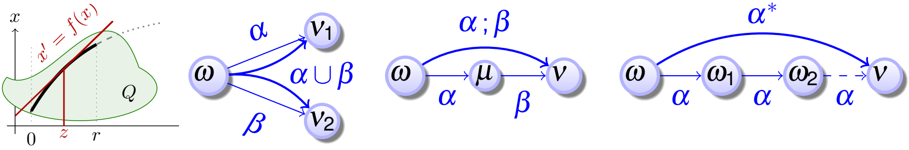
$$\underbrace{x \neq m \land b > 0}_{\text{init}} \rightarrow \Big[\Big(\big(\text{if}(\text{SB}(x,m))\quad a := -b\big)\ ;\ x' = v, v' = a\Big)^*\Big]\underbrace{x \neq m}_{\text{post}}$$

all runs



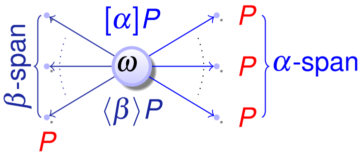
Definition (Hybrid program)

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



Definition (Differential dynamic logic)

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



equations of truth

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

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

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

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

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

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

$\mathrm{K}\ [\alpha](P \rightarrow Q) \rightarrow ([\alpha]P \rightarrow [\alpha]Q)$

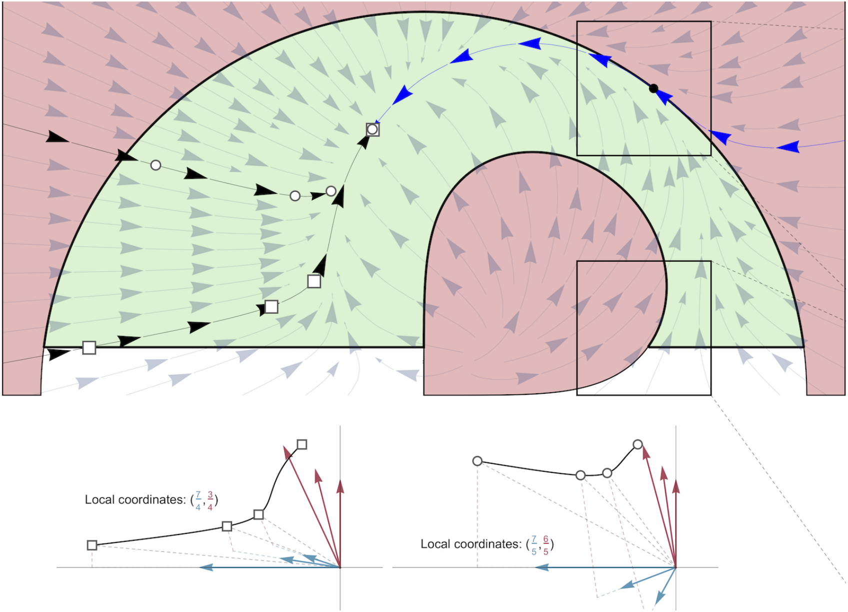
$\mathrm{I}\ [\alpha^*]P \leftrightarrow P \land [\alpha^*](P \rightarrow [\alpha]P)$

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

laws of logic of
laws of physics

LICS'12,JAR'17

Completeness for Differential Equation Invariants



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.

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

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

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

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*

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

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.

$$\begin{array}{ll} e'^{*} = 0 \equiv e{=}0 \wedge (e')'^{*}{=}0 & (P \wedge Q)'^{*} \equiv P'^{*} \wedge Q'^{*} \\ e'^{*} \geq 0 \equiv e{\geq}0 \wedge (e{=}0{\rightarrow}(e')'^{*}{\geq}0) & (P \vee Q)'^{*} \equiv P'^{*} \vee Q'^{*} \end{array}$$

Differential Invariants for Differential Equations

Differential Invariant

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

Differential Cut

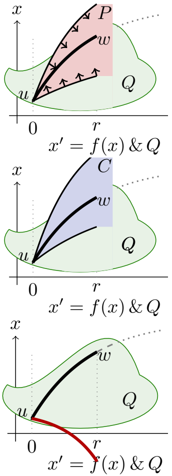
$$\frac{P \vdash [x' = f(x) \,\&\, Q]C \quad P \vdash [x' = f(x) \,\&\, Q \wedge C]P}{P \vdash [x' = f(x) \,\&\, Q]P}$$

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

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



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

Differential Invariant

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

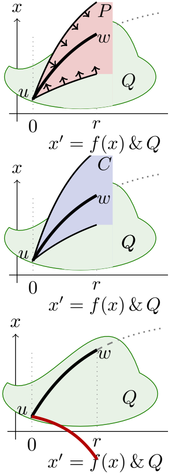
Differential Cut

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

Differential Ghost

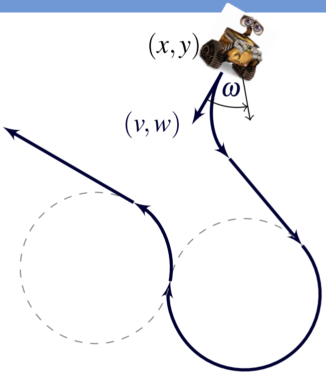
$$\frac{P \leftrightarrow \exists y\, G \quad G \vdash [x' = f(x), y' = g(x,y) \,\&\, Q]G}{P \vdash [x' = f(x) \,\&\, Q]P}$$

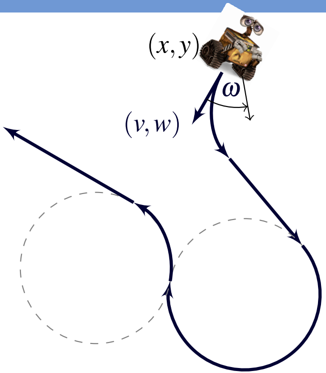
if $g(x,y) = a(x)y + b(x)$, so has long solution!



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

Ex: Runaround Robot

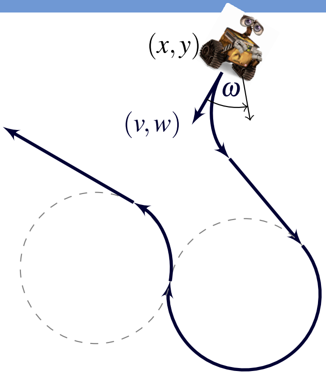




Example (Runaround Robot)

$$\big((\omega := -1 \cup \omega := 1 \cup \omega := 0);\\ \{x' = v, y' = w, v' = \omega w, w' = -\omega v\}\big)^*$$

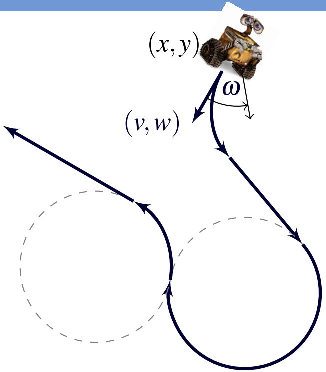
Ex: Runaround Robot



Example (Runaround Robot)

$$(x,y) \neq o \rightarrow \Big[\Big((\omega := -1 \cup \omega := 1 \cup \omega := 0);$$
$$\{x' = v, y' = w, v' = \omega w, w' = -\omega v\}\Big)^*\Big]\,(x,y) \neq o$$

Ex: Runaround Robot

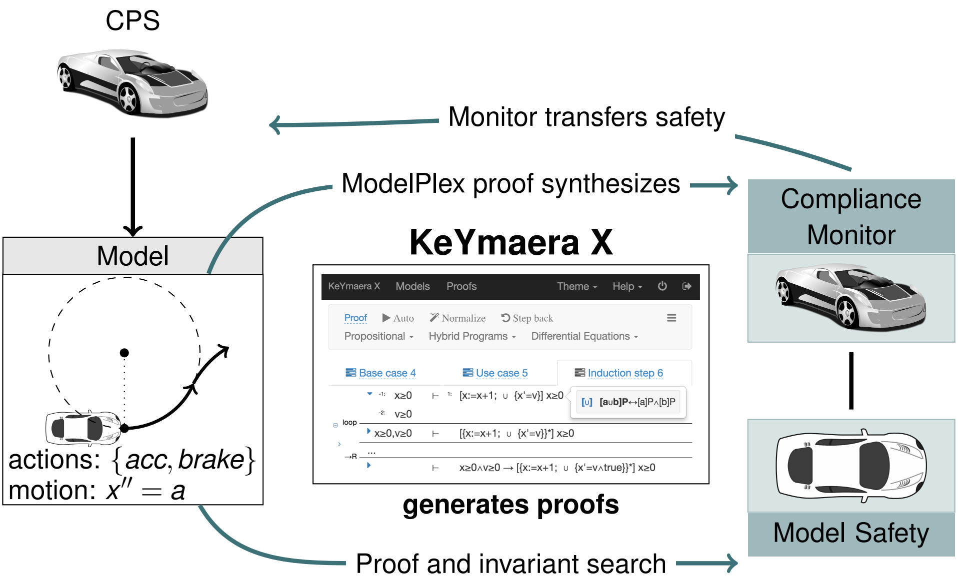


Example (Runaround Robot)

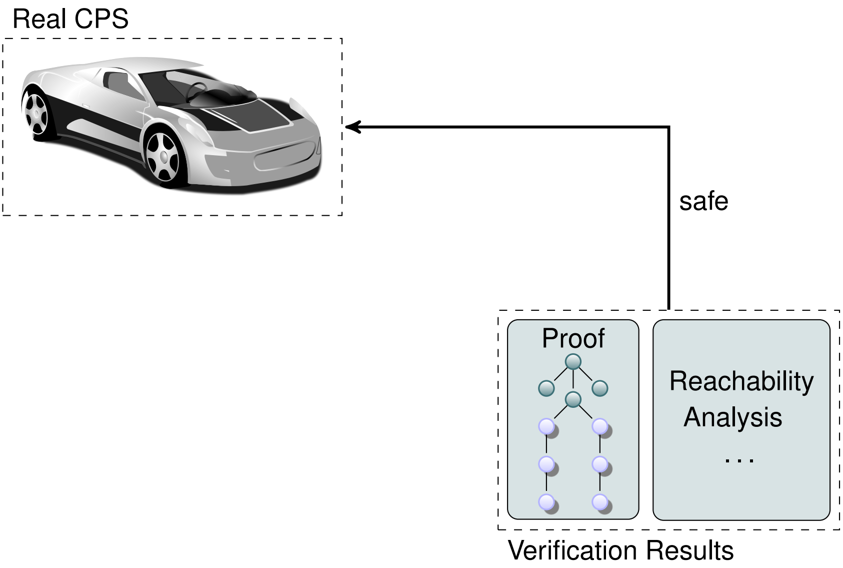
$$(x,y) \neq o \rightarrow \Big[\big((?Q_{-1};\omega := -1 \cup ?Q_1;\omega := 1 \cup ?Q_0;\omega := 0);$$
$$\{x' = v, y' = w, v' = \omega w, w' = -\omega v\}\big)^*\Big]\,(x,y) \neq o$$

Outline

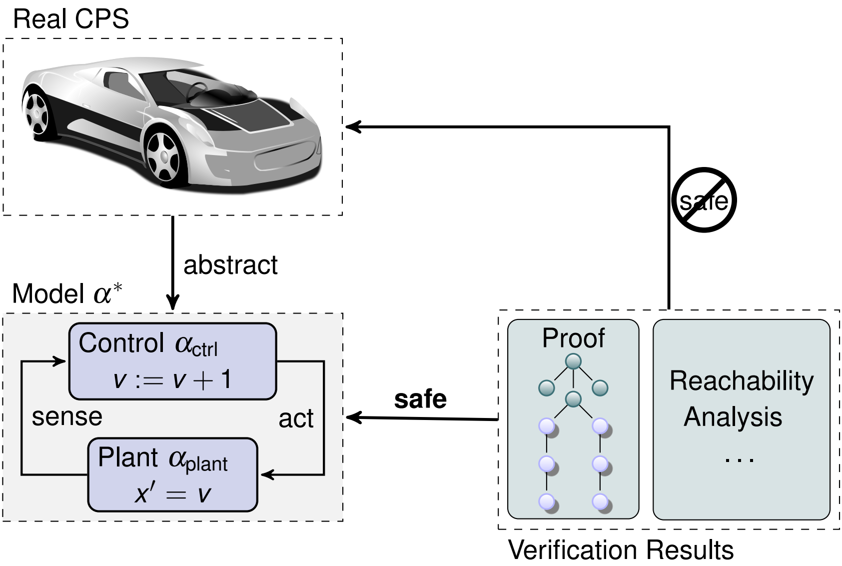
Approach: Proofs for Cyber-Physical Systems

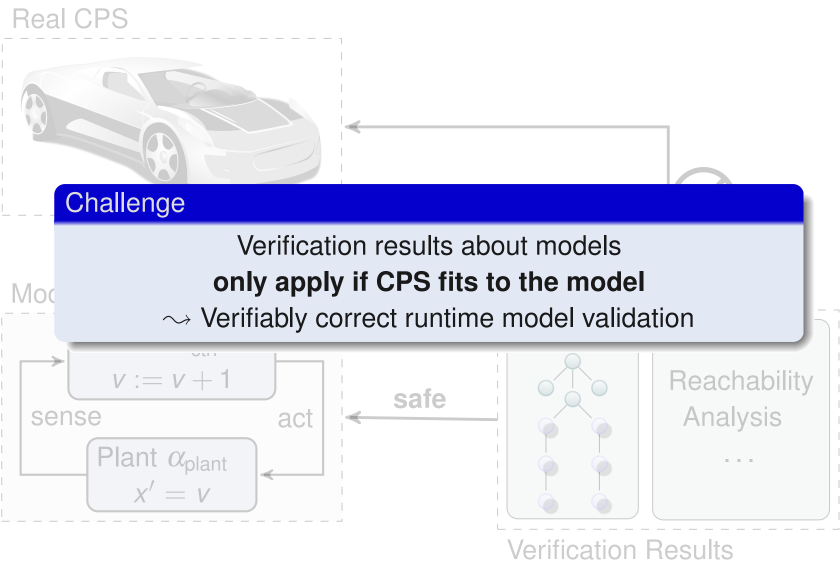


Formal Verification in CPS Development



Formal Verification in CPS Development





Challenge

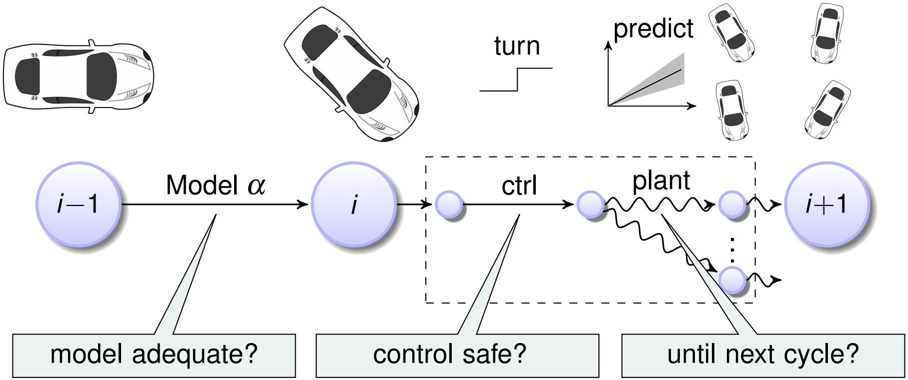
Verification results about models

only apply if CPS fits to the model

$\rightsquigarrow$ Verifiably correct runtime model validation

ModelPlex Runtime Model Validation

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



FMSD'16

ModelPlex Runtime Model Validation

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

Insights

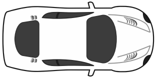
- Verification results about models transfer to the CPS when validating model compliance.
- Compliance with model is characterizable in logic dL.
- Compliance formula transformed by dL proof to monitor.
- Correct-by-construction provably correct model validation at runtime.

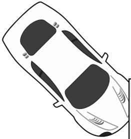
model adequate? | control safe? | until next cycle?

FMSD'16

Characterizing State Relations in Logic

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





a prior state characterized by x^-

a posterior state characterized by x^+

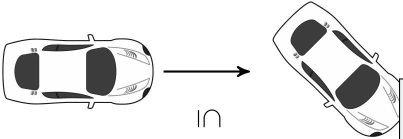
$$\omega \xrightarrow{\text{Model } \alpha} \nu$$

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

FMSD'16

Characterizing State Relations in Logic

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



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

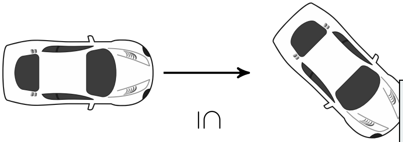
Logical dL: $(\omega, \nu) \models \langle\alpha\rangle(x = x^+)$

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

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^-

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

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

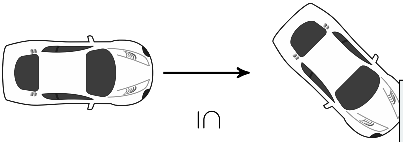
Arithmetical: $(\omega, \nu) \models F(x^-, x^+)$

check at runtime (efficient)

FMSD'16

Characterizing State Relations in Logic

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



a prior state characterized by x^-

$\cap\!|$

a posterior state characterized by x^+

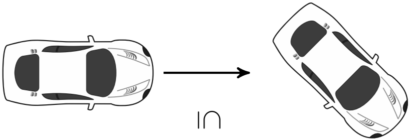
$$\omega \xrightarrow{\text{Model } \alpha} \nu$$

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)

FMSD'16

Logic reduces CPS safety to runtime monitor with offline proof



dL proof $A \rightarrow [\alpha]S$

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

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

Offline

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

$\Updownarrow$ Lemma

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

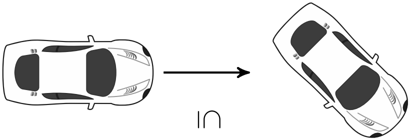
$\Uparrow$ dL proof

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

check at runtime (efficient)

FMSD'16

Logic reduces CPS safety to runtime monitor with offline proof



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

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

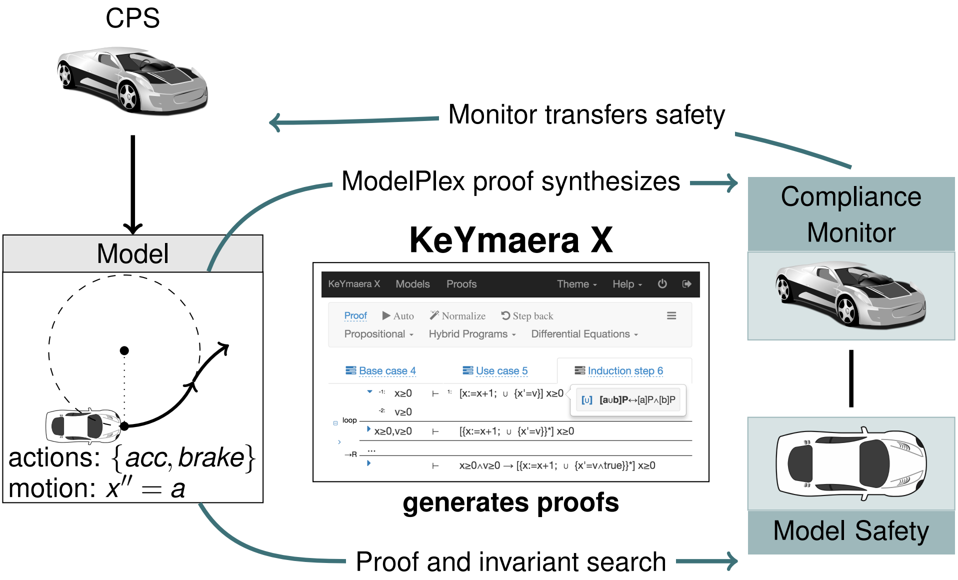
Arithmetical: $(\omega, \nu) \models F(x^-, x^+)$

check at runtime (efficient)

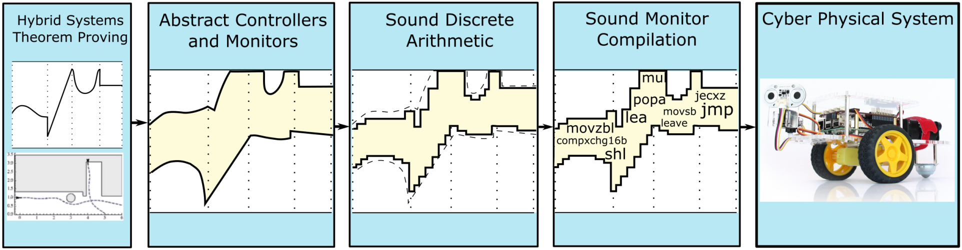
FMSD'16

Outline

Approach: Proofs for Cyber-Physical Systems

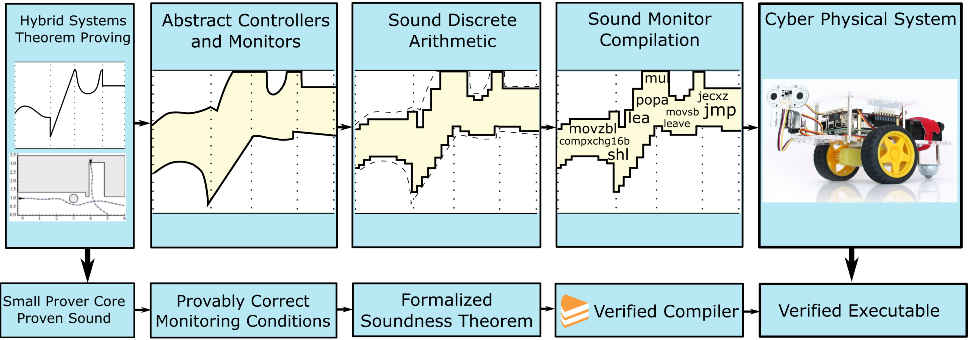


VeriPhy: Automatic, Verified EXEs from Controllers



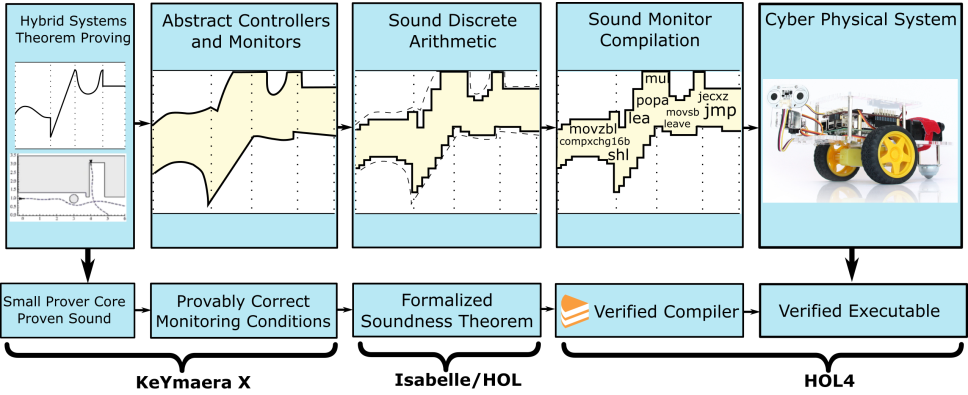
PLDI'18

VeriPhy: Automatic, Verified EXEs from Controllers



PLDI'18

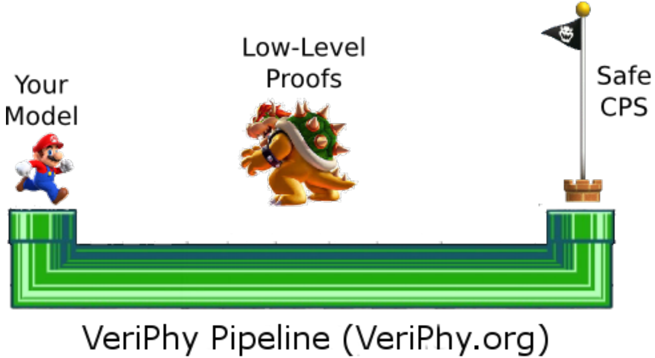
VeriPhy: Automatic, Verified EXEs from Controllers



PLDI'18

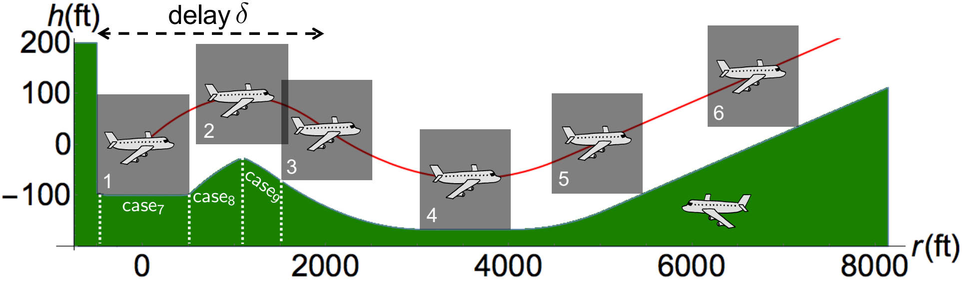
VeriPhy: Takeaway Metaphor

VeriPhy: Takeaway Metaphor



Outline

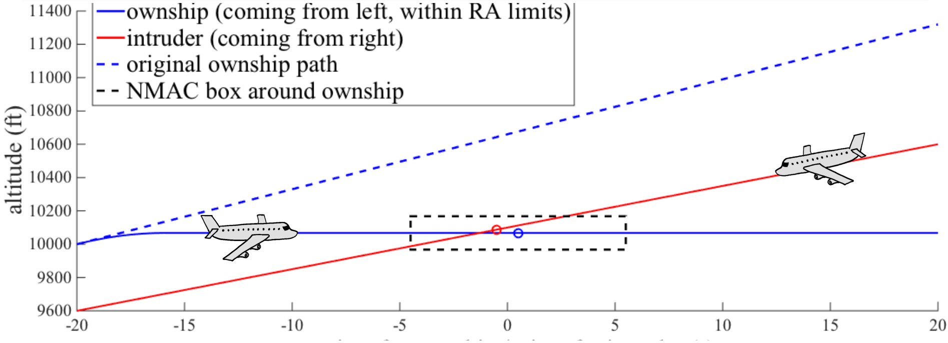
- Developed by the FAA to replace current TCAS in aircraft
- Approximately optimizes Markov Decision Process on a grid
- Advisory from lookup tables with numerous 5D interpolation regions



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

STTT'17

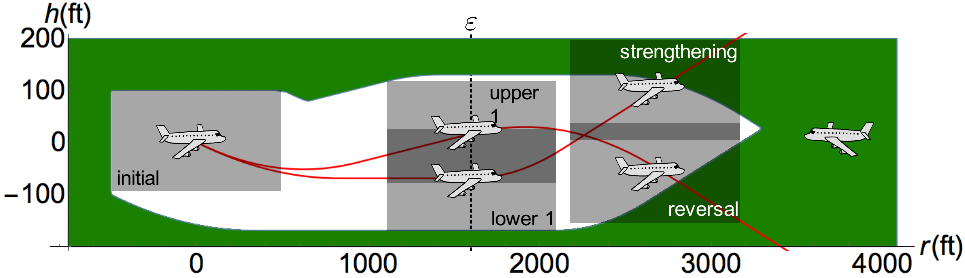
ACAS X table comparison shows safe advisory in 97.7% of the 648,591,384,375 states compared (15,160,434,734 counterexamples).



ACAS X issues DNC advisory, which induces collision unless corrected

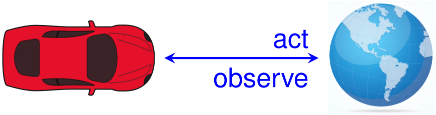
STTT'17

- 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

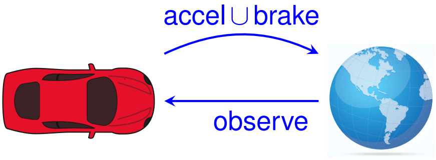
STTT'17



Reinforcement Learning learns from experience of trying actions

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

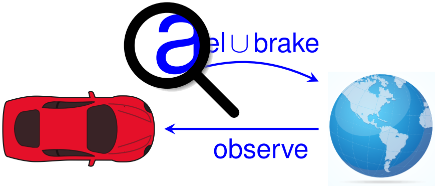
Learning to Act in a CPS



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

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

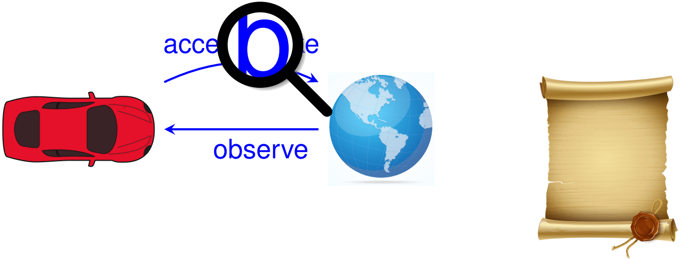
Learning to Act Safely in a CPS



ModelPlex monitor inspects each decision, vetoes if unsafe

AAAI’18,ITC’18,TACAS’19,QEST’19

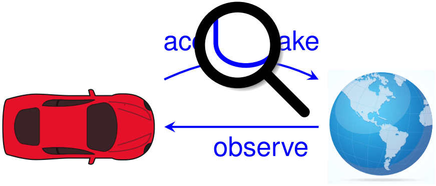
Learning to Act Safely in a CPS



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

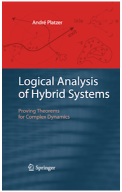
Outline

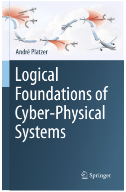
Logical Systems Lab at Carnegie Mellon University, Computer Science
Yong Kiam Tan, Brandon Bohrer, Nathan Fulton, Sarah Loos, Katherine Cordwell
Stefan Mitsch, Khalil Ghorbal, Jean-Baptiste Jeannin, Andrew Sogokon

Unterstützt von / Supported by
Alexander von Humboldt
Stiftung / Foundation

differential dynamic logic

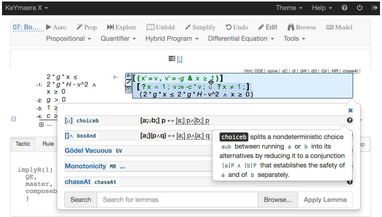
$$\mathsf{dL} = \mathsf{DL} + \mathsf{HP}$$





- Compositional formal verification
- Logic & proofs for CPS
- Small soundness core
- Proof by pointing
- Interactive proof clicking
- Tactical proof programming
- Proof search automation
- Flexible + modular API

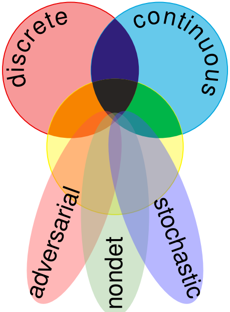
KeYmaera X



http://keymaeraX.org/

Further CPS Topics

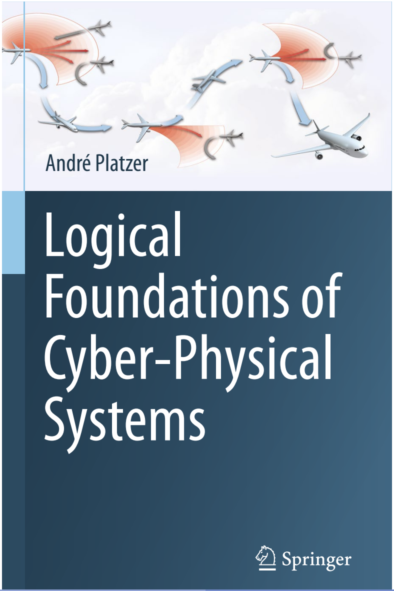
- Verified CPS systems by ModelPlex FMSD'16
- Verified CPS execution by VeriPhy PLDI'18
- CPS proof and tactic languages+libraries ITP'17
- Big CPS built from safe components STTT'18
- Stochastic hybrid systems CADE'11
- Invariant generation FM'19
- Safe AI autonomy in CPS AAAI'18 TACAS'19
- Correct model transformation FM'14
- Refinement + system property proofs LICS'16
- Automatic ODE proofs LICS'18
- CPS information flow LICS'18
- Hybrid games TOCL'15

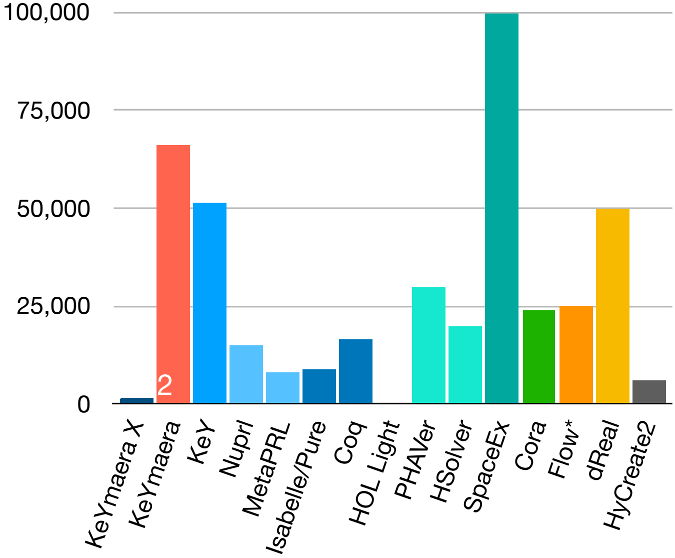


CPSs deserve proofs as safety evidence!

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**





Disclaimer: Self-reported estimates of the soundness-critical lines of code + rules

Uniform Substitution

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

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

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)

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

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

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

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)

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

Uniform Substitution

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

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

Modular interface:
Prover vs. Logic

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

Differential Dynamic Logic dL: Semantics

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

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

compositional semantics

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

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

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

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 [20], pages 13–24.
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.

André Platzer.

The complete proof theory of hybrid systems.
In LICS [20], pages 541–550.
doi:10.1109/LICS.2012.64.

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.

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

André Platzer and Edmund M. Clarke.

Computing differential invariants of hybrid systems as fixedpoints.
Form. Methods Syst. Des., 35(1):98–120, 2009.
Special issue for selected papers from CAV'08.
doi:10.1007/s10703-009-0079-8.

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.
A differential operator approach to equational differential invariants.
In Lennart Beringer and Amy Felty, editors, *ITP*, volume 7406 of *LNCS*, pages 28–48, Berlin, 2012. Springer.
doi:10.1007/978-3-642-32347-8_3.

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.

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.

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.

André Platzer.

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

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.
Stochastic differential dynamic logic for stochastic hybrid programs.
In Nikolaj Bjørner and Viorica Sofronie-Stokkermans, editors, *CADE*, volume 6803 of *LNCS*, pages 446–460, Berlin, 2011. Springer.
doi:10.1007/978-3-642-22438-6_34.

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.