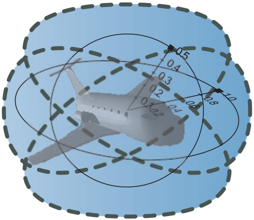
Logical Foundations of Cyber-Physical Systems

André Platzer

Computer Science Department
Carnegie Mellon University



Cyber-Physical Systems Analysis: Aircraft Example

Which control decisions are safe for aircraft collision avoidance?

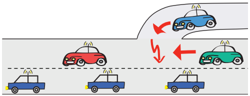
Cyber-Physical Systems

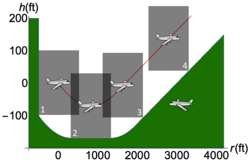
CPSs combine cyber capabilities with physical capabilities
to solve problems that neither part could solve alone.

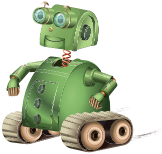
Cyber-Physical Systems Promise Transformative Impact!

Prospects: Safety & Efficiency

(Autonomous) cars | (Auto)Pilot support | Robots near humans



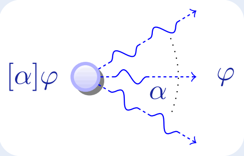


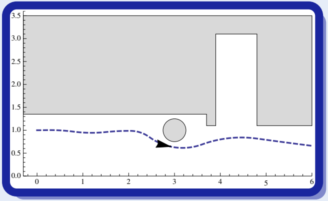


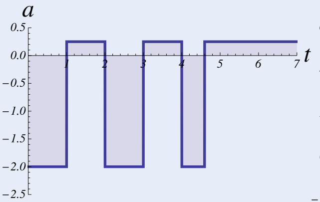
Cyber-Physical Systems

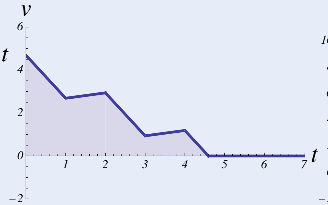
CPSs combine cyber capabilities with physical capabilities
to solve problems that neither part could solve alone.

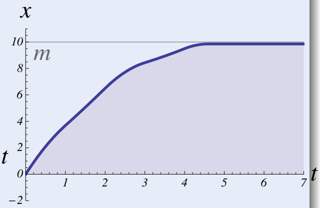
Concept (Differential Dynamic Logic) (JAR'08,LICS'12)





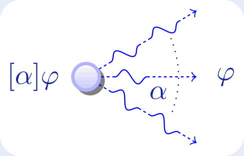


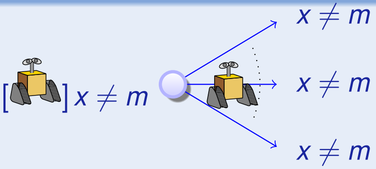


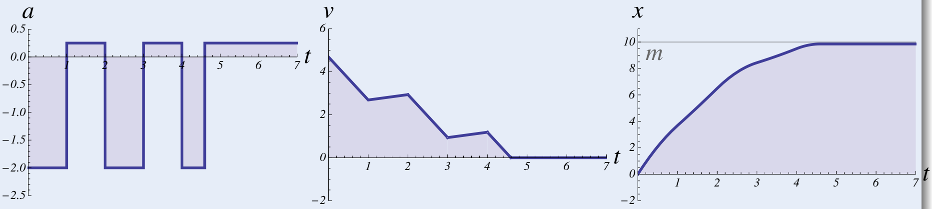


CPS Analysis

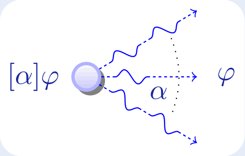
Concept (Differential Dynamic Logic) (JAR'08,LICS'12)

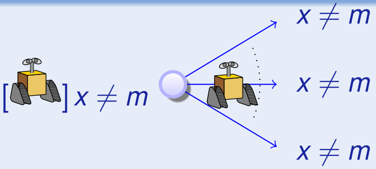




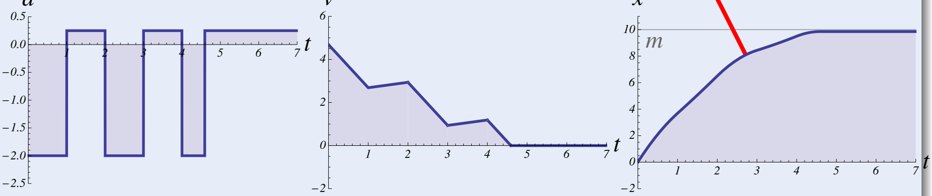


Concept (Differential Dynamic Logic) (JAR'08,LICS'12)

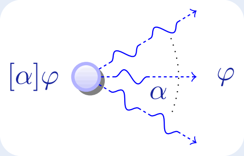




$$x' = v, v' = a$$



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



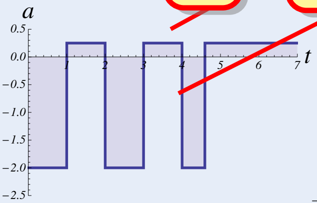
seq. compose

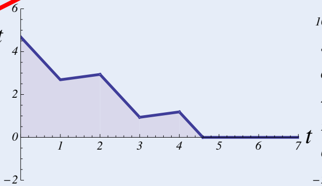
$(\texttt{if}(SB(x,m))\, a := -b)\ ;\ x' = v, v' = a$

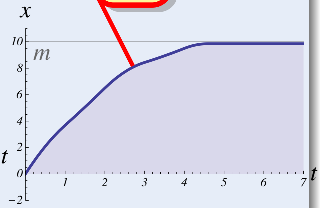
cond

assign

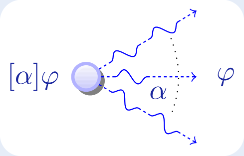
ODE







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



seq. compose

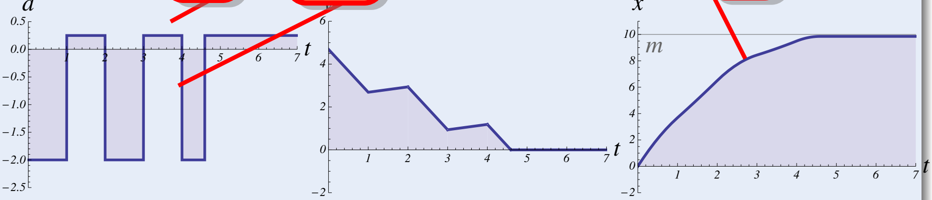
nondet. repeat

$$((\mathtt{if}(SB(x,m))\; a := -b) \; ; \; x' = v, v' = a)^*$$

cond

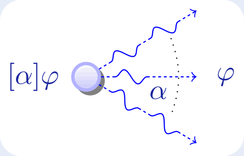
assign

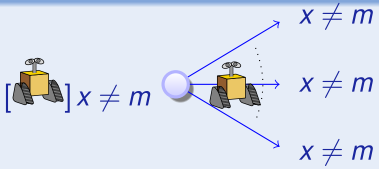
ODE



CPS Analysis

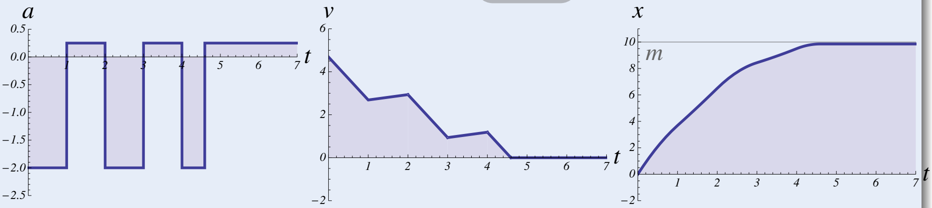
Concept (Differential Dynamic Logic) (JAR'08,LICS'12)



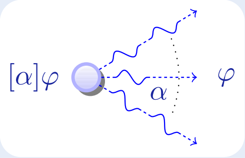


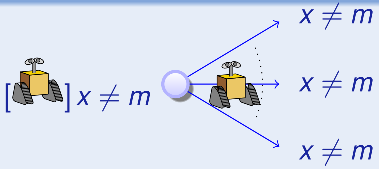
$$[((\mathtt{if}(SB(x,m))\ a := -b) ;\ x' = v, v' = a)^*]\underbrace{x \neq m}_{\text{post}}$$

all runs



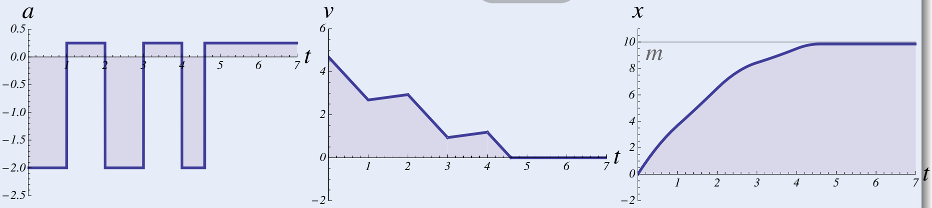
Concept (Differential Dynamic Logic) (JAR'08,LICS'12)





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

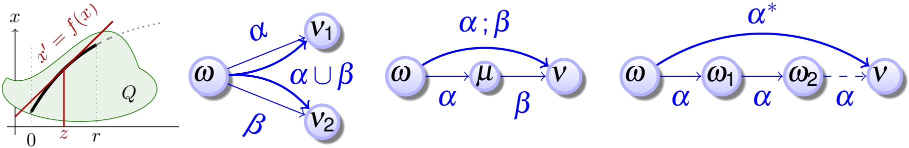
all runs



Differential Dynamic Logic dL

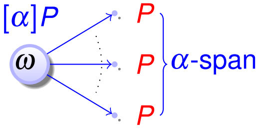
Definition (Hybrid program)

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



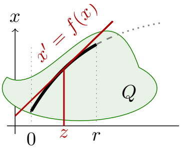
Definition (Differential dynamic logic)

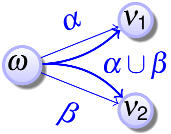
$$P, Q ::= e \geq \tilde{e} \mid \neg P \mid P \wedge Q \mid P \vee Q \mid P \rightarrow Q \mid \forall x\, P \mid \exists x\, P \mid [\alpha]P \mid \langle \alpha \rangle P$$

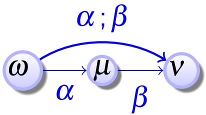


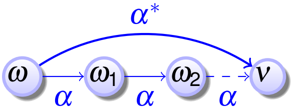
Definition (Hybrid program)

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



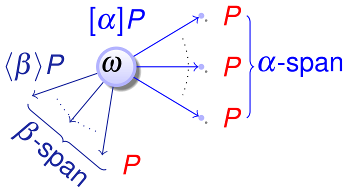






Definition (Differential dynamic logic)

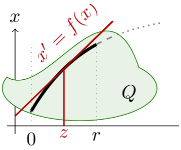
$$P, Q ::= e \geq \tilde{e} \mid \neg P \mid P \wedge Q \mid P \vee Q \mid P \rightarrow Q \mid \forall x\, P \mid \exists x\, P \mid [\alpha]P \mid \langle \alpha \rangle P$$

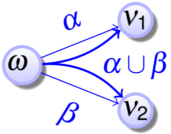


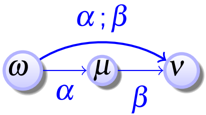
Differential Dynamic Logic dL

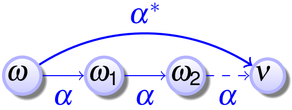
Definition (Hybrid program)

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



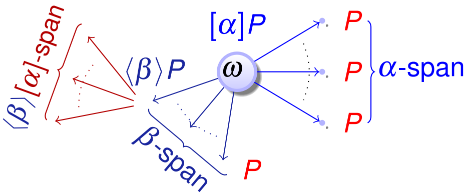






Definition (Differential dynamic logic)

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



Differential Dynamic Logic dL: Axiomatization

equations of truth

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

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

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

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

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

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

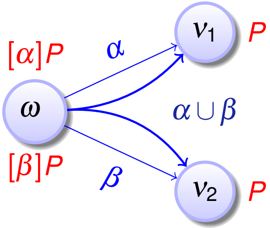
K $[\alpha](P \rightarrow Q) \rightarrow ([\alpha]P \rightarrow [\alpha]Q)$

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

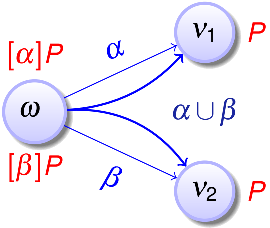
C $[\alpha^*]\forall v{>}0\,(P(v) \rightarrow \langle\alpha\rangle P(v-1)) \rightarrow \forall v\,(P(v) \rightarrow \langle\alpha^*\rangle \exists v{\leq}0\,P(v))$

LICS'12,JAR'17

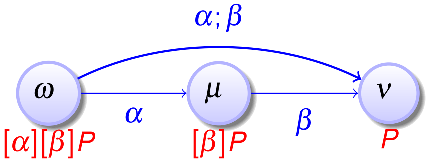
$$[\alpha \cup \beta]P \leftrightarrow [\alpha]P \wedge [\beta]P$$



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

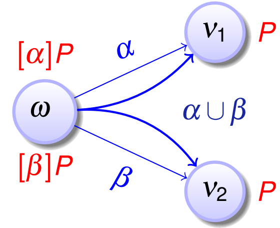


$$[\alpha;\beta]P \leftrightarrow [\alpha][\beta]P$$

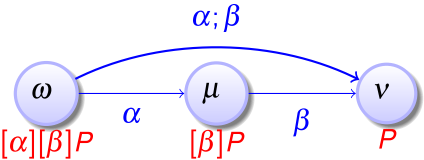


Proofs for CPS

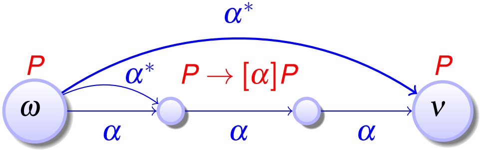
$[\alpha \cup \beta]P \leftrightarrow [\alpha]P \wedge [\beta]P$



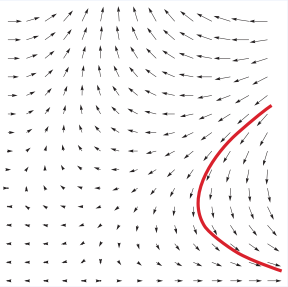
$[\alpha;\beta]P \leftrightarrow [\alpha][\beta]P$



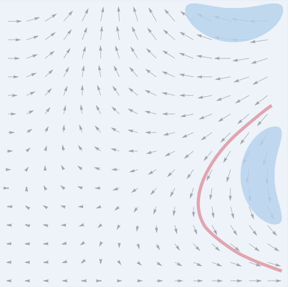
$[\alpha^*]P \leftrightarrow P \wedge [\alpha^*](P \rightarrow [\alpha]P)$



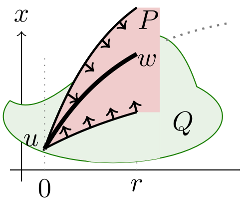
Differential Invariant



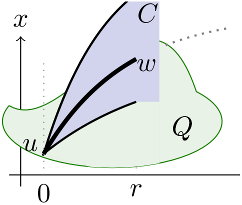
Differential Cut



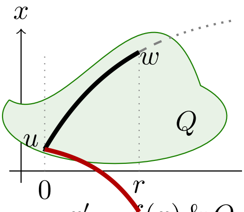
Differential Ghost



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



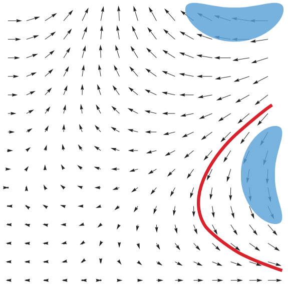
$x' = f(x) \,\&\, Q$

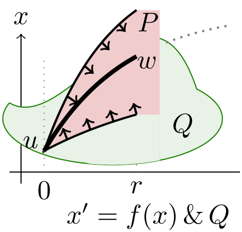


$x' = f(x) \,\&\, Q$

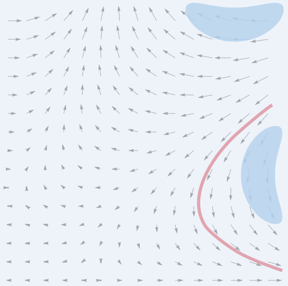
Differential Invariants for Differential Equations

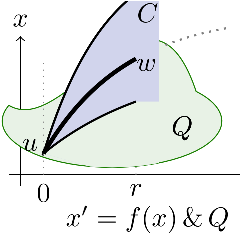
Differential Invariant



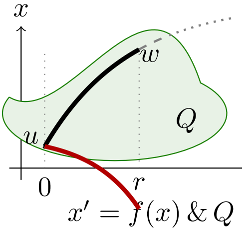


Differential Cut



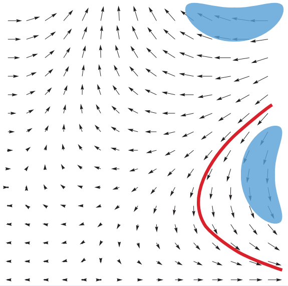


Differential Ghost

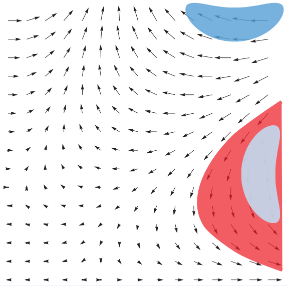


Differential Invariants for Differential Equations

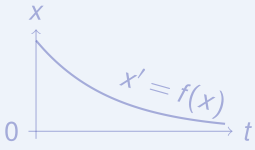
Differential Invariant

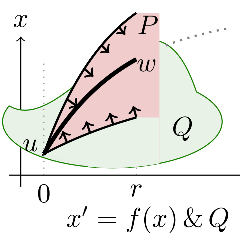


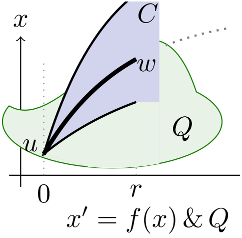
Differential Cut

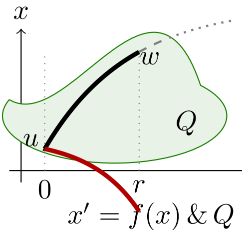


Differential Ghost



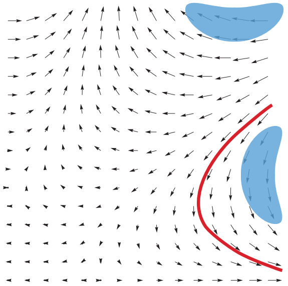




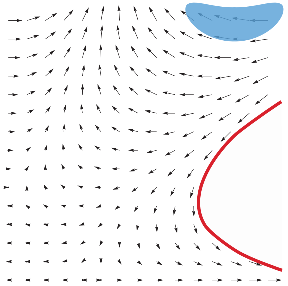


Differential Invariants for Differential Equations

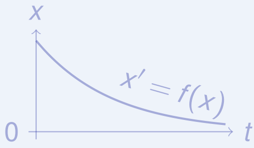
Differential Invariant

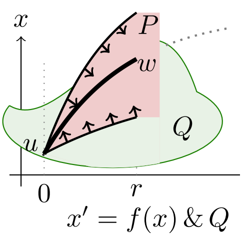


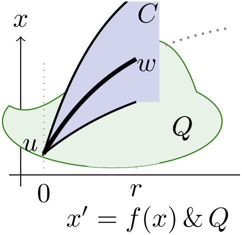
Differential Cut

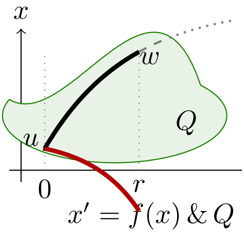


Differential Ghost



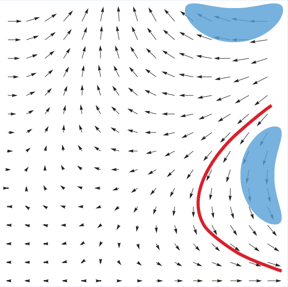


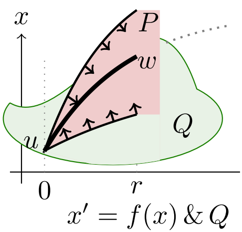




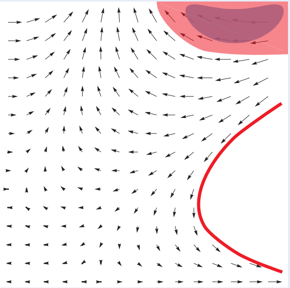
Differential Invariants for Differential Equations

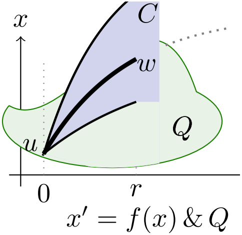
Differential Invariant



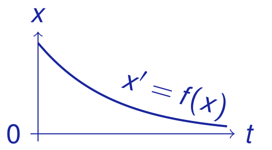


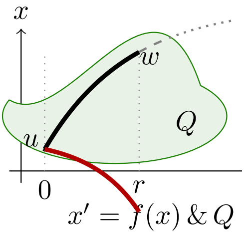
Differential Cut





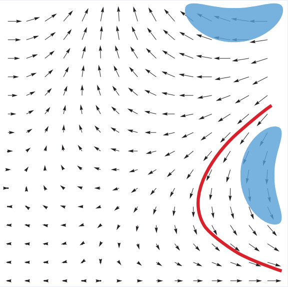
Differential Ghost

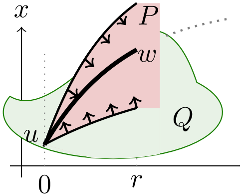




Differential Invariants for Differential Equations

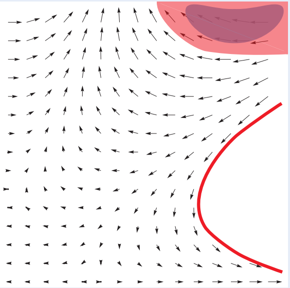
Differential Invariant

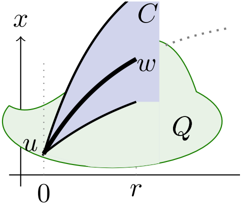




$x' = f(x) \,\&\, Q$

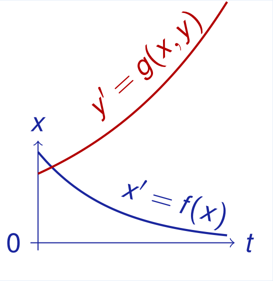
Differential Cut

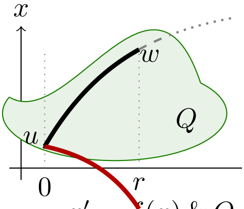




$x' = f(x) \,\&\, Q$

Differential Ghost

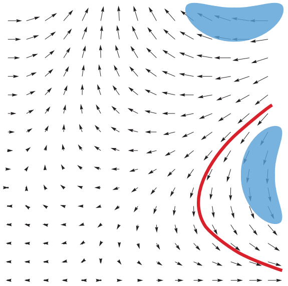




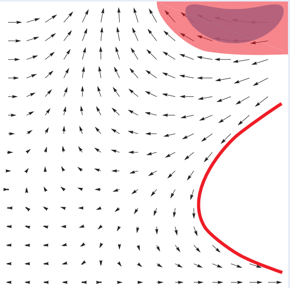
$x' = f(x) \,\&\, Q$

Differential Invariants for Differential Equations

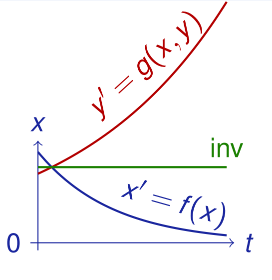
Differential Invariant

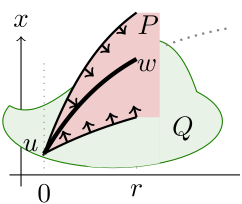


Differential Cut

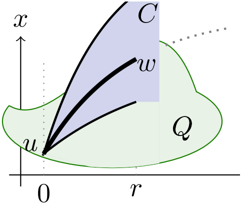


Differential Ghost

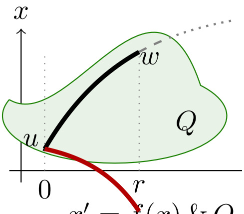




$x' = f(x) \,\&\, Q$



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



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

Differential Invariants for Differential Equations

Differential Invariant

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

Differential Cut

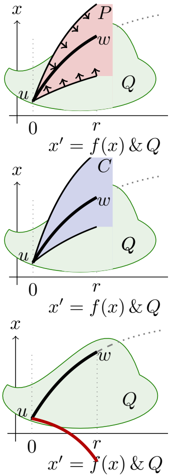
$$\frac{P \to [x' = f(x) \,\&\, Q]C \quad P \to [x' = f(x) \,\&\, Q \land C]P}{P \to [x' = f(x) \,\&\, Q]P}$$

Differential Ghost

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

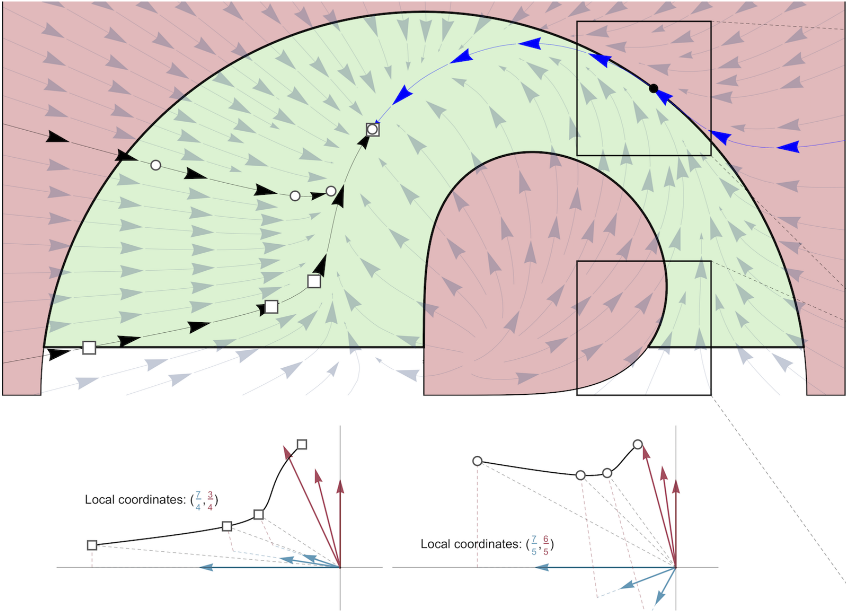
deductive power added DI $\prec$ DI+DC $\prec$ DI+DC+DG

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



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

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.

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 with a derived axiom:*

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

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 with derived axiom:*

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

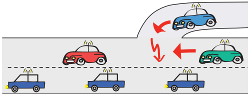
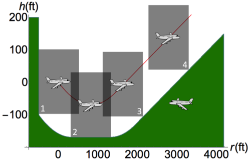
Definable $p^{\dot{}(*)}$ is short for *all/significant* Lie derivative w.r.t. ODE
Definable $p^{\dot{}(-*)}$ is w.r.t. backwards ODE $x' = -f(x)$.

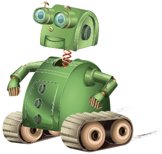
Applications with Impact

Prospects: Safety & Efficiency

(Autonomous) cars | (Auto)Pilot support | Robots near humans



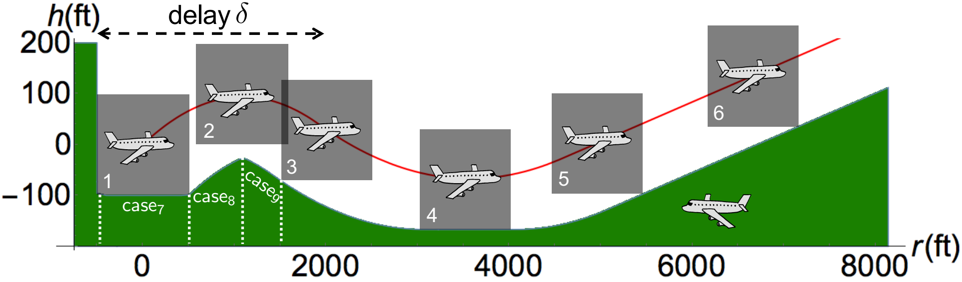





Cyber-Physical Systems

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

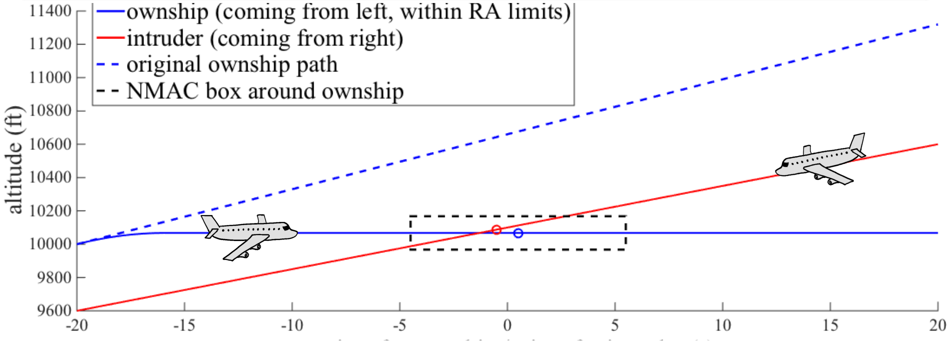
- Developed by the FAA to replace current TCAS in aircraft
- Approximately optimizes Markov Decision Process on a grid
- Advisory from lookup tables with numerous 5D interpolation regions



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

STTT'17

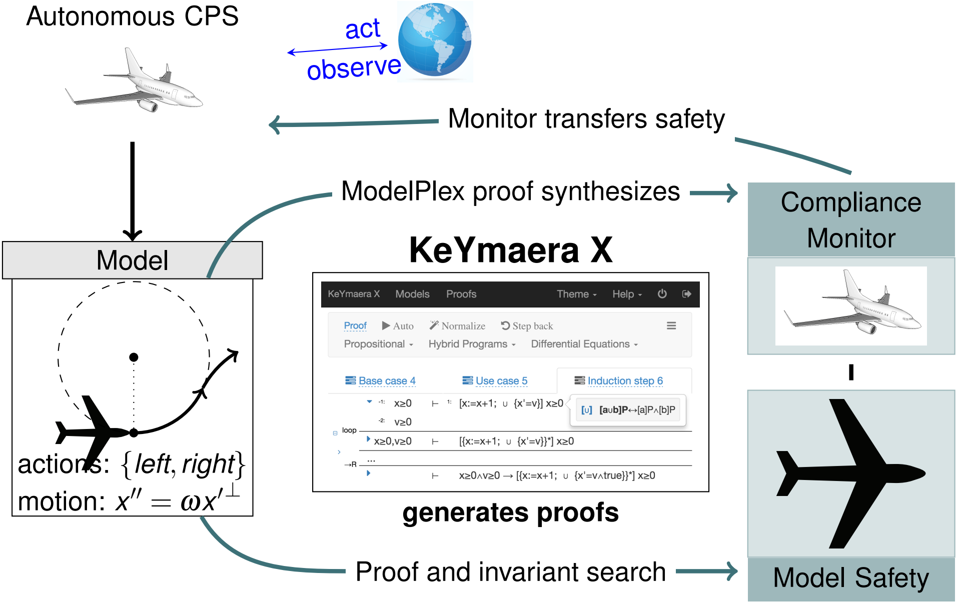
ACAS X table comparison shows safe advisory in 97.7% of the 648,591,384,375 states compared (15,160,434,734 counterexamples).



ACAS X issues DNC advisory, which induces collision unless corrected

STTT'17

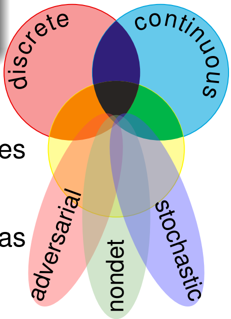
Approach: Safety Proofs for Autonomous CPS

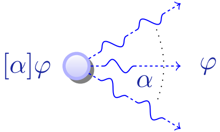


Logical Foundations of Cyber-Physical Systems

differential dynamic logic

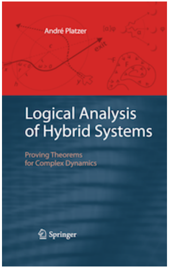
$\text{dL} = \text{DL} + \text{HP}$



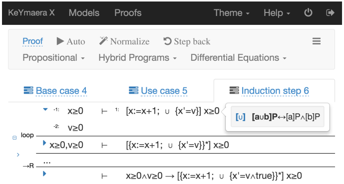


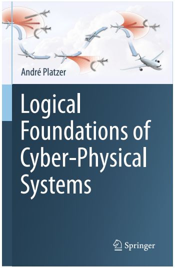
- Strong analytic foundations
- Practical reasoning advances
- Significant applications
- Catalyze many science areas

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



KeYmaera X

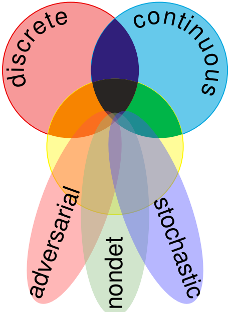




Future CPS Challenges

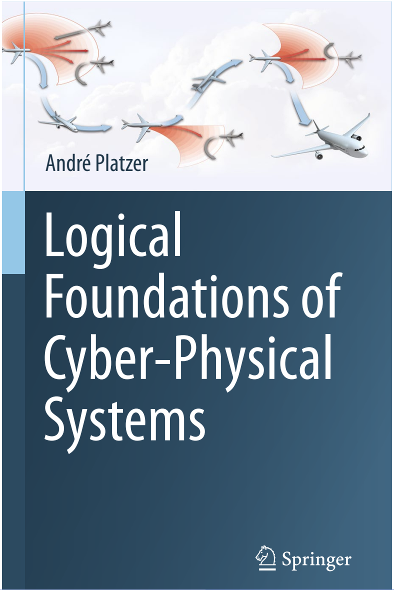
Numerous wonders remain to be discovered

- 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
- Real arithmetic: Scalable and verified CADE'09
- Scalable stochastic dynamics reasoning CADE'11
- Concurrent CPS
- Invariant generation FM'19
- Safe AI autonomy in CPS AAAI'18
- Correct model transformation FM'14
- Refinement + system property proofs LICS'16
- 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



André Platzer.
Logics of dynamical systems.
In LICS [16], pages 13–24.
doi:10.1109/LICS.2012.13.

André Platzer.
Logical Foundations of Cyber-Physical Systems.
Springer, Cham, 2018.
URL: http://www.springer.com/978-3-319-63587-3,
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 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.
The complete proof theory of hybrid systems.
In LICS [16], pages 541–550.
doi:10.1109/LICS.2012.64.

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 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, Jan-David Quesel, and Philipp Rümmer.
Real world verification.
In Renate A. Schmidt, editor, *CADE*, volume 5663 of *LNCS*, pages 485–501, Berlin, 2009. Springer.
doi:10.1007/978-3-642-02959-2_35.

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.

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