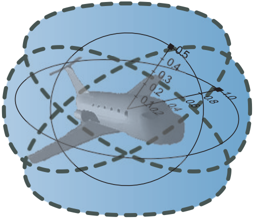
# Logic & Proofs for Cyber-Physical Systems with KeYmaera X

André Platzer

Carnegie Mellon University



# Outline

# Cyber-Physical Systems Analysis: Aircraft Example

Which control decisions are safe for aircraft collision avoidance?

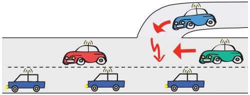
## Cyber-Physical Systems

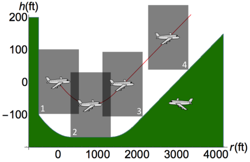
CPSs combine cyber capabilities with physical capabilities
to solve problems that neither part could solve alone.

# CPSs Promise Transformative Impact!

## Prospects: Safe & Efficient

| | | |
|---|---|---|
| Driver assistance | Pilot decision support | Train protection |
| Autonomous cars | Autopilots / UAVs | Robots near humans |







## Prerequisite: CPSs need to be safe

How do we make sure CPSs make the world a better place?

# Can you trust a computer to control physics?

# Can you trust a computer to control physics?

1. Depends on how it has been programmed
2. And on what will happen if it malfunctions

## Rationale

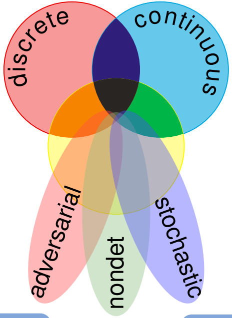
1. Safety guarantees require analytic foundations.
2. A common foundational core helps all application domains.
3. Foundations revolutionized digital computer science & our society.
4. Need even stronger foundations when software reaches out into our physical world.

# CPSs deserve proofs as safety evidence!

# CPSs are Multi-Dynamical Systems

**CPS Dynamics**

CPS are characterized by multiple facets of dynamical systems.




**CPS Compositions**

CPS combines multiple simple dynamical effects.

Descriptive simplification

**Tame Parts**

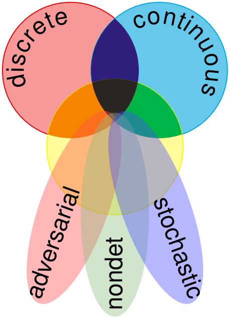
Exploiting compositionality tames CPS complexity.

Analytic simplification

# CPSs are Multi-Dynamical Systems

**hybrid systems**

HS = discrete + ODE

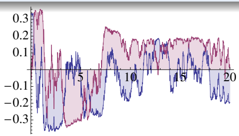



**hybrid games**

HG = HS + adversary

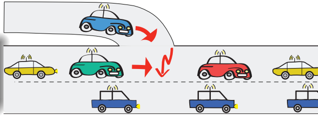
**stochastic hybrid sys.**

SHS = HS + stochastics



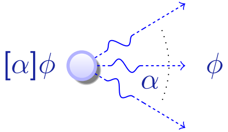

**distributed hybrid sys.**

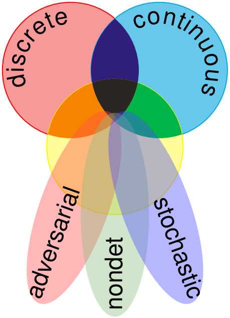
DHS = HS + distributed



**differential dynamic logic**

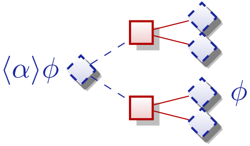
$d\mathcal{L} = DL + HP$



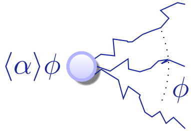

**differential game logic**

$dG\mathcal{L} = GL + HG$




**stochastic differential DL**

$Sd\mathcal{L} = DL + SHP$




**quantified differential DL**

$Qd\mathcal{L} = FOL + DL + QHP$

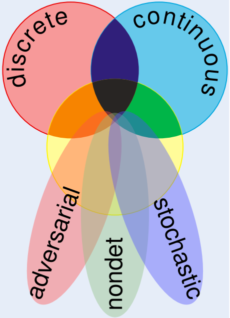
JAR'08,CADE'11,LMCS'12,LICS'12,LICS'12

TOCL'15,CADE'15,JAR'17,TOCL'17

# Dynamic Logics for Dynamical Systems

## Dynamic Logics

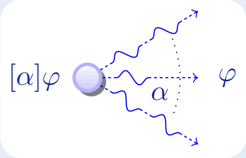
- DL has been introduced for programs Pratt'76,Harel,Kozen
- Its real calling are dynamical systems
- DL excels at providing simple+elegant logical foundations for dynamical systems
- CPSs are multi-dynamical systems
- DL for CPS are multi-dynamical

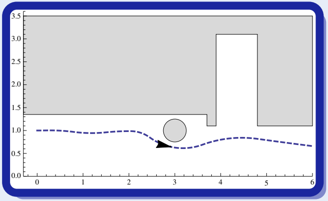


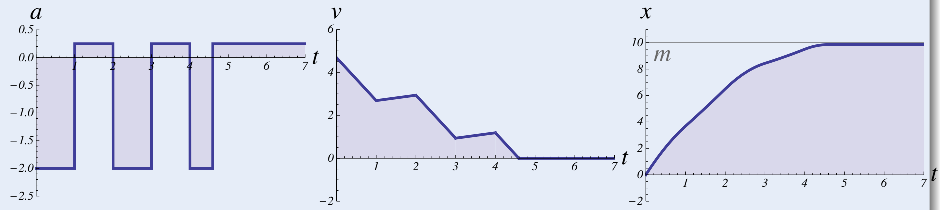

JAR'08,CADE'11,LMCS'12,LICS'12,LICS'12

TOCL'15,CADE'15,JAR'17,TOCL'17

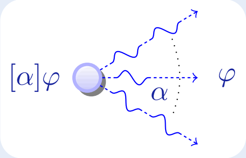
## Concept (Differential Dynamic Logic) (JAR'08,LICS'12)

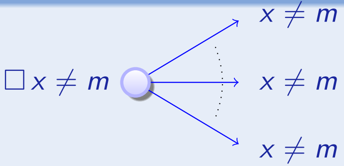


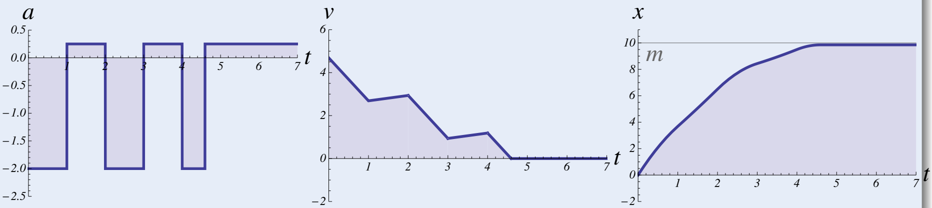





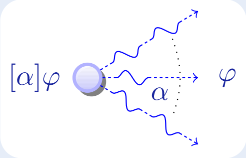
## Concept (Differential Dynamic Logic) (JAR'08,LICS'12)

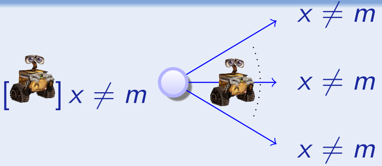


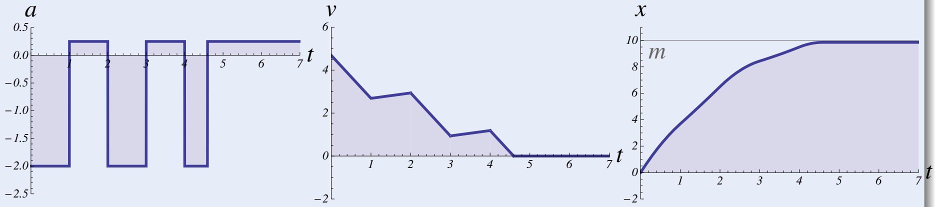



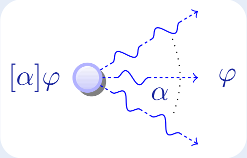
## Concept (Differential Dynamic Logic) (JAR'08,LICS'12)

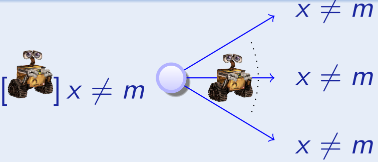


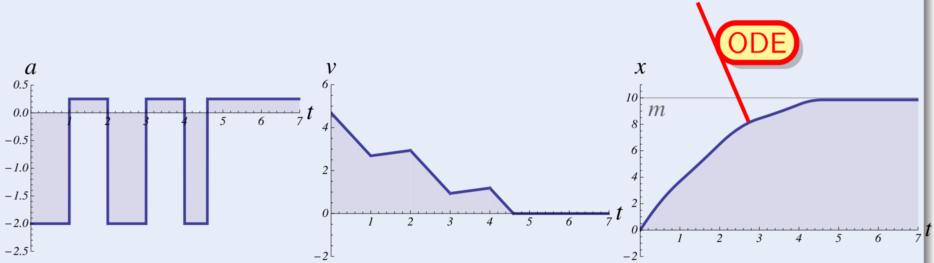



## Concept (Differential Dynamic Logic) (JAR'08,LICS'12)

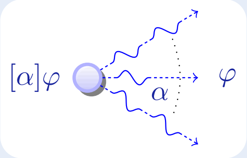


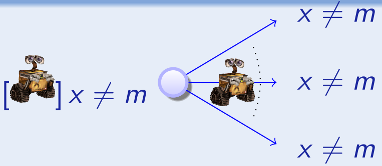




$x' = v, v' = a$

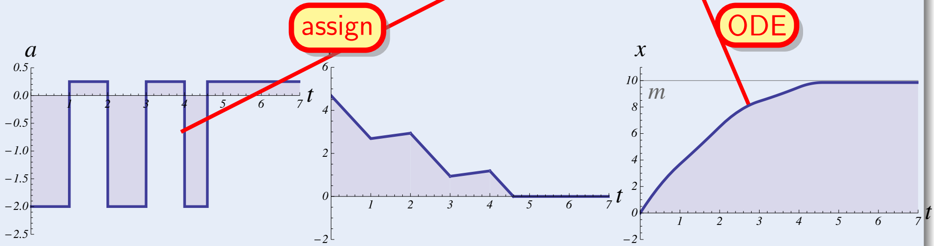


## Concept (Differential Dynamic Logic) (JAR'08,LICS'12)

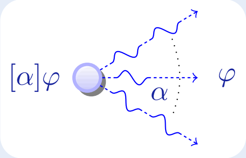


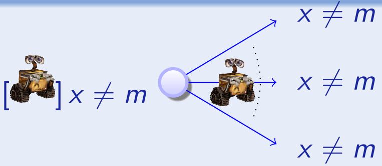




$a := -b$ $\quad x' = v, v' = a$

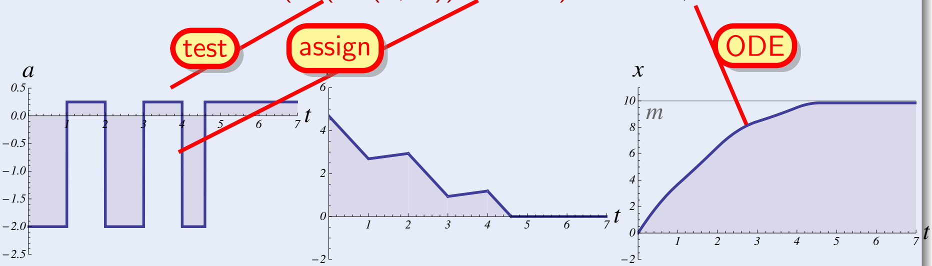


## Concept (Differential Dynamic Logic) (JAR'08,LICS'12)

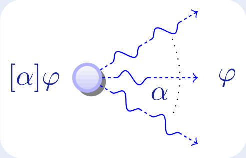



$(\texttt{if}(SB(x, m))\ a := -b)$ $\quad x' = v, v' = a$



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



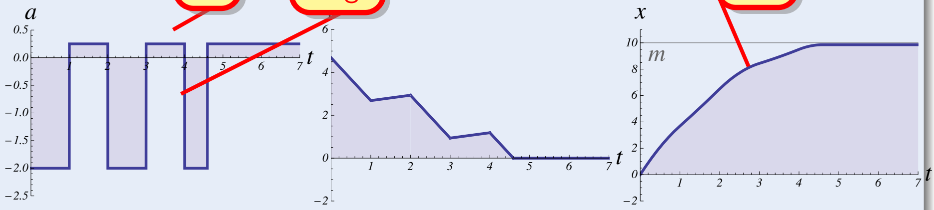

seq. compose

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

test

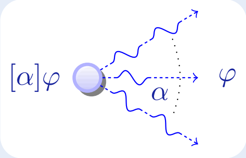
assign

ODE



# CPS Analysis

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




seq. compose

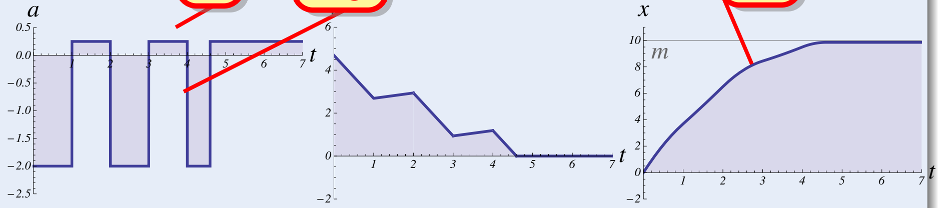
nondet. repeat

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

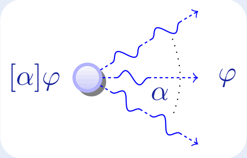
test

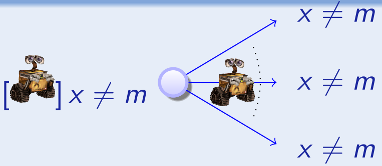
assign

ODE



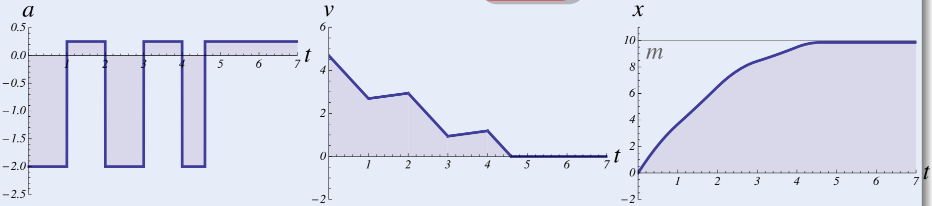
## Concept (Differential Dynamic Logic) (JAR'08,LICS'12)



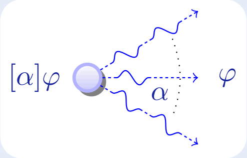



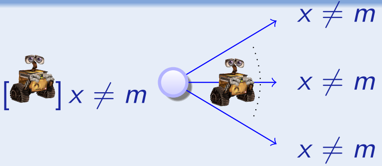

$$\left[\left(\left(\mathtt{if}(\mathsf{SB}(x,m))\ a := -b\right)\ ;\ x' = v, v' = a\right)^*\right] \underbrace{x \neq m}_{\text{post}}$$

all runs



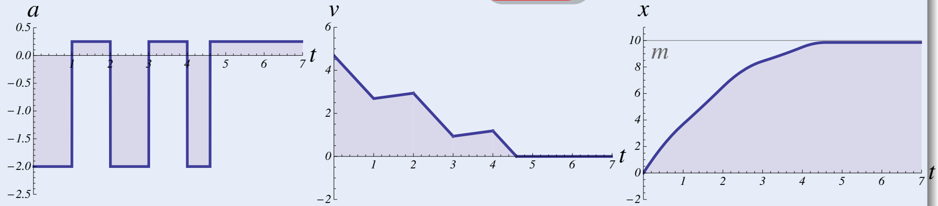
## Concept (Differential Dynamic Logic) (JAR'08,LICS'12)




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

all runs



# Differential Dynamic Logic d$\mathcal{L}$: Syntax

**Definition (Hybrid program $\alpha$)**

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

**Definition (d$\mathcal{L}$ Formula $P$)**

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

JAR'08,LICS'12,JAR'17

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

**Definition (Hybrid program $\alpha$)**

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

**Definition (d$\mathcal{L}$ 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

# Differential Dynamic Logic: Axiomatization

equations of truth

| | |
|---|---|
| [:=] | $[x := e]P(x) \leftrightarrow P(e)$ |
| [?] | $[?Q]P \leftrightarrow (Q \rightarrow P)$ |
| ['] | $[x' = f(x)]P \leftrightarrow \forall t{\geq}0\, [x := y(t)]P$ $\qquad (y'(t) = f(y))$ |
| [∪] | $[\alpha \cup \beta]P \leftrightarrow [\alpha]P \wedge [\beta]P$ |
| [;] | $[\alpha; \beta]P \leftrightarrow [\alpha][\beta]P$ |
| [*] | $[\alpha^*]P \leftrightarrow P \wedge [\alpha][\alpha^*]P$ |
| K | $[\alpha](P \rightarrow Q) \rightarrow ([\alpha]P \rightarrow [\alpha]Q)$ |
| I | $[\alpha^*]P \leftrightarrow P \wedge [\alpha^*](P \rightarrow [\alpha]P)$ |
| C | $[\alpha^*]\forall v{>}0\, (P(v) \rightarrow \langle\alpha\rangle P(v-1)) \rightarrow \forall v\, (P(v) \rightarrow \langle\alpha^*\rangle \exists v{\leq}0\, P(v))$ |

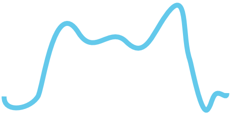
LICS'12,JAR'17

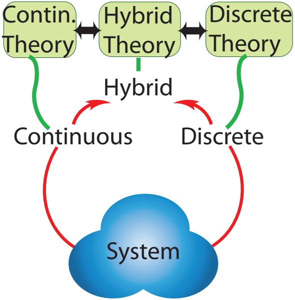
**Theorem (Sound & Complete)** (JAR'08, LICS'12, JAR'17)

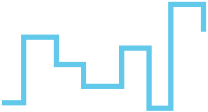
d$\mathcal{L}$ *calculus is a sound & complete axiomatization of hybrid systems relative to either differential equations* **or** *to discrete dynamics.* *Proof 25pp*

**Corollary (Complete Proof-theoretical Bridge)**

proving continuous = proving hybrid = proving discrete

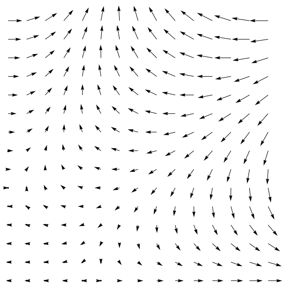




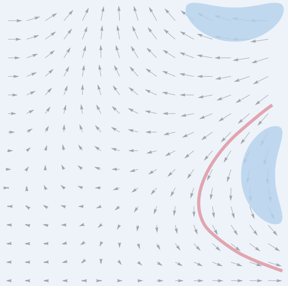



# Differential Invariants for Differential Equations

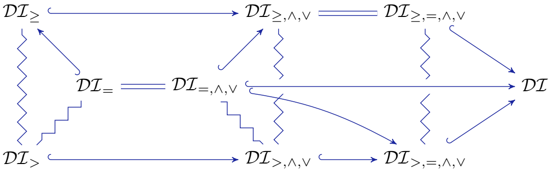
## Differential Invariant



## Differential Cut



## Differential Ghost


## Logic

Provability theory

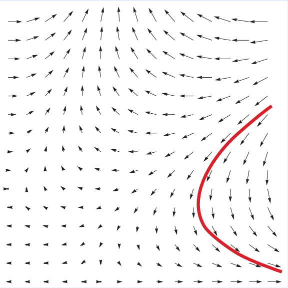
## Math

Characteristic PDE

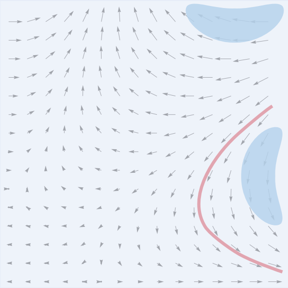
JLogComput'10,CAV'08,FMSD'09,LMCS'12,LICS'12,ITP'12,JAR'17

# Differential Invariants for Differential Equations

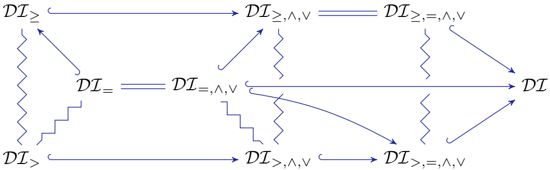
## Differential Invariant



## Differential Cut



## Differential Ghost


## Logic

Provability theory

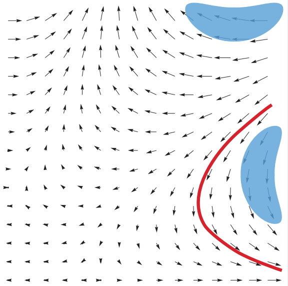
## Math

Character-istic PDE

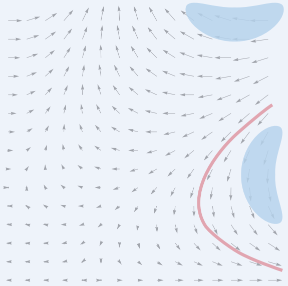
JLogComput'10,CAV'08,FMSD'09,LMCS'12,LICS'12,ITP'12,JAR'17

# Differential Invariants for Differential Equations

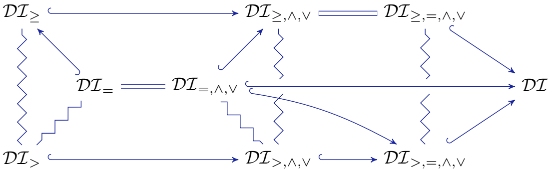
## Differential Invariant



## Differential Cut



## Differential Ghost


## Logic

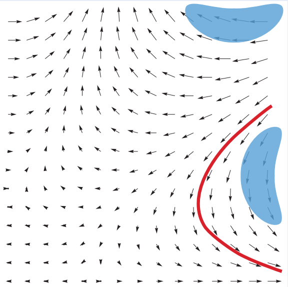
Provability theory

## Math

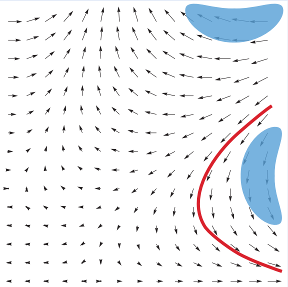
Character-istic PDE

JLogComput'10,CAV'08,FMSD'09,LMCS'12,LICS'12,ITP'12,JAR'17

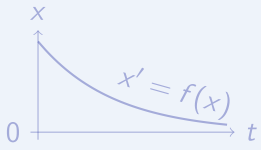
### Differential Invariant

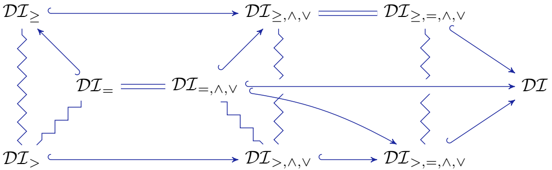


### Differential Cut



### Differential Ghost




### Logic

Provability theory

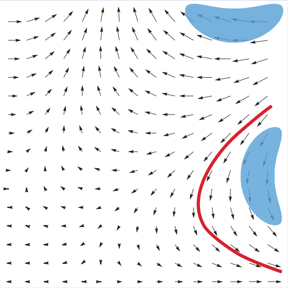
### Math

Character-istic PDE

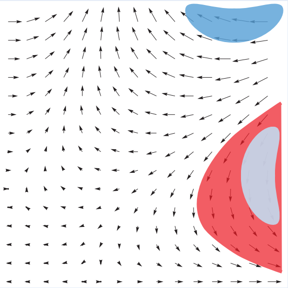
JLogComput'10,CAV'08,FMSD'09,LMCS'12,LICS'12,ITP'12,JAR'17

# Differential Invariants for Differential Equations

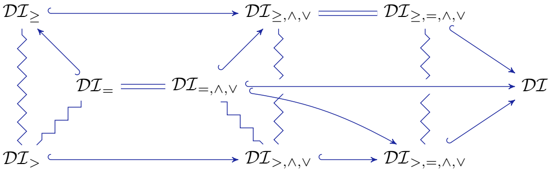
## Differential Invariant



## Differential Cut



## Differential Ghost


## Logic

Provability theory

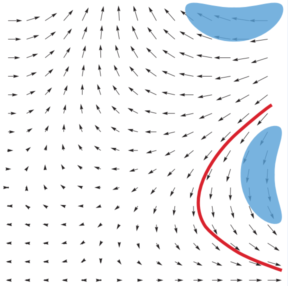
## Math

Character-istic PDE

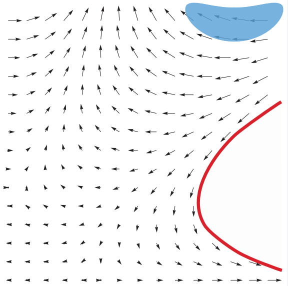
JLogComput'10,CAV'08,FMSD'09,LMCS'12,LICS'12,ITP'12,JAR'17

# Differential Invariants for Differential Equations

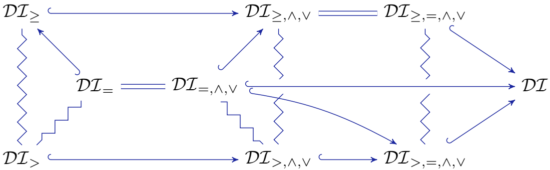
## Differential Invariant



## Differential Cut



## Differential Ghost


## Logic

Provability theory

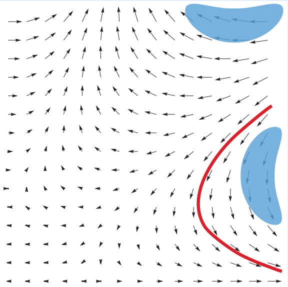
## Math

Character-istic PDE

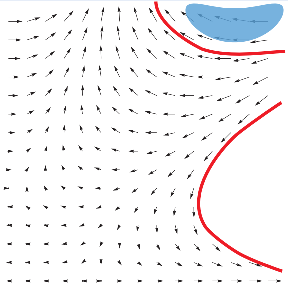
JLogComput'10,CAV'08,FMSD'09,LMCS'12,LICS'12,ITP'12,JAR'17

# Differential Invariants for Differential Equations

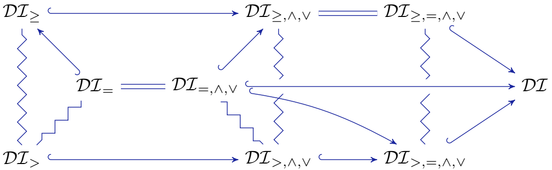
## Differential Invariant



## Differential Cut



## Differential Ghost


## Logic

Provability theory

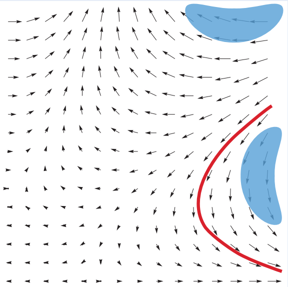
## Math

Character-istic PDE

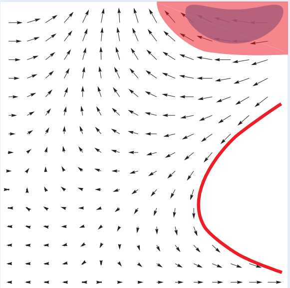
JLogComput'10,CAV'08,FMSD'09,LMCS'12,LICS'12,ITP'12,JAR'17

# Differential Invariants for Differential Equations

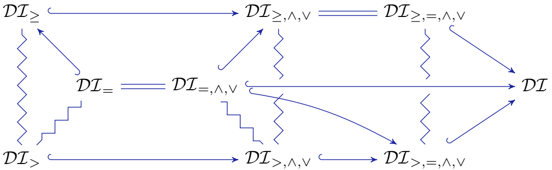
## Differential Invariant



## Differential Cut



## Differential Ghost


## Logic

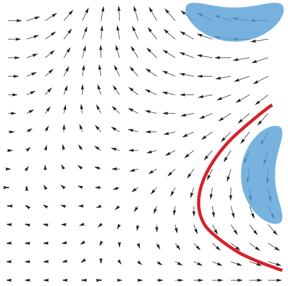
Provability theory

## Math

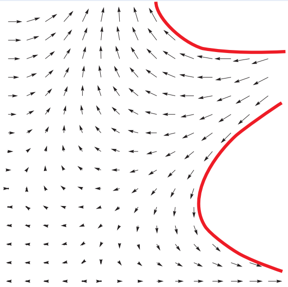
Characteristic PDE

JLogComput'10,CAV'08,FMSD'09,LMCS'12,LICS'12,ITP'12,JAR'17

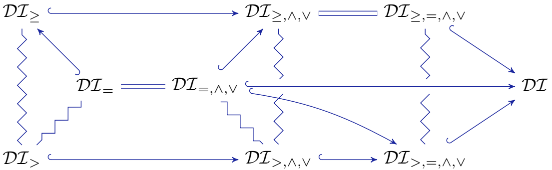
## Differential Invariant



## Differential Cut



## Differential Ghost


## Logic

Provability theory

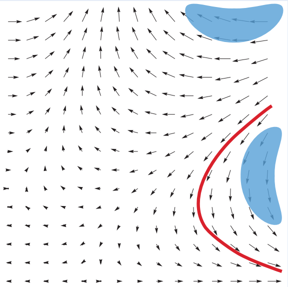
## Math

Characteristic PDE

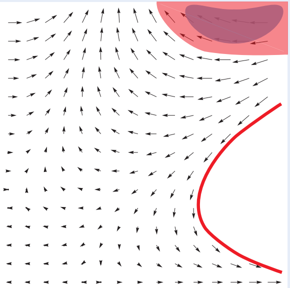
JLogComput'10,CAV'08,FMSD'09,LMCS'12,LICS'12,ITP'12,JAR'17

# Differential Invariants for Differential Equations

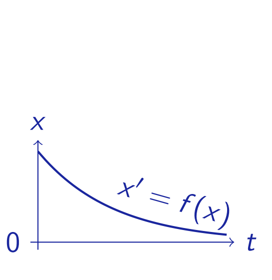
## Differential Invariant

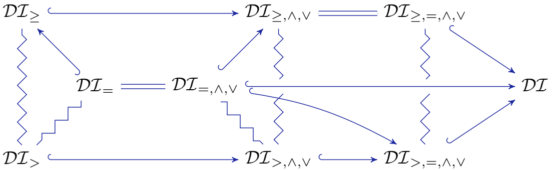


## Differential Cut



## Differential Ghost




## Logic

Provability theory

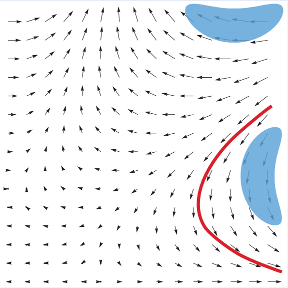
## Math

Character-istic PDE

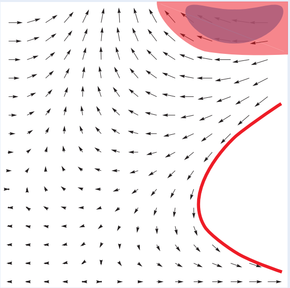
JLogComput'10,CAV'08,FMSD'09,LMCS'12,LICS'12,ITP'12,JAR'17

# Differential Invariants for Differential Equations

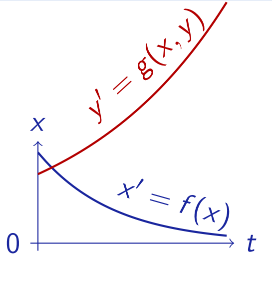
## Differential Invariant

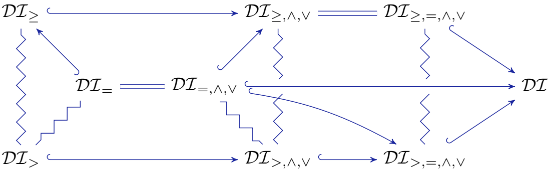


## Differential Cut



## Differential Ghost




## Logic

Provability theory

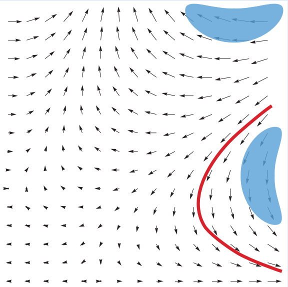
## Math

Character-istic PDE

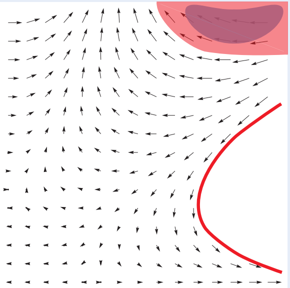
JLogComput'10,CAV'08,FMSD'09,LMCS'12,LICS'12,ITP'12,JAR'17

# Differential Invariants for Differential Equations

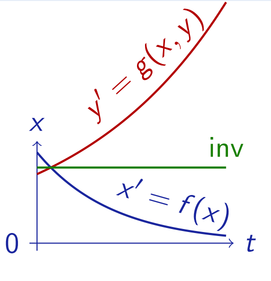
## Differential Invariant

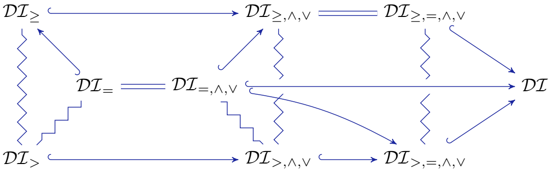


## Differential Cut



## Differential Ghost




## Logic

Provability theory

## Math

Character-istic PDE

JLogComput'10,CAV'08,FMSD'09,LMCS'12,LICS'12,ITP'12,JAR'17

# Differential Invariants for Differential Equations

**Differential Invariant**

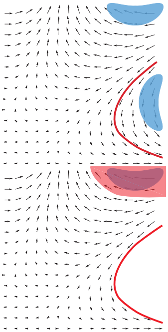
$$\frac{Q \vdash [x' := f(x)](P)'}{P \vdash [x' = f(x) \,\&\, Q]P}$$

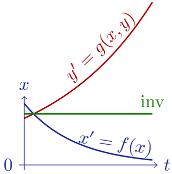
**Differential Cut**

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

**Differential Ghost**

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






JLogComput'10,LMCS'12, LICS'12,JAR'17

# Differential Invariants for Differential Equations

**Differential Invariant**

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

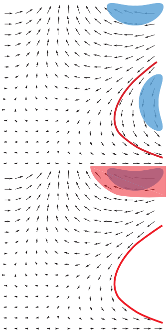
**Differential Cut**

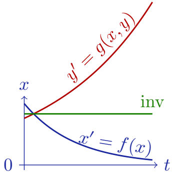
$$\frac{P \vdash [x' = f(x) \,\&\, Q]C \quad P \vdash [x' = f(x) \,\&\, Q \wedge C]P}{P \vdash [x' = f(x) \,\&\, Q]P}$$

**Differential Ghost**

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

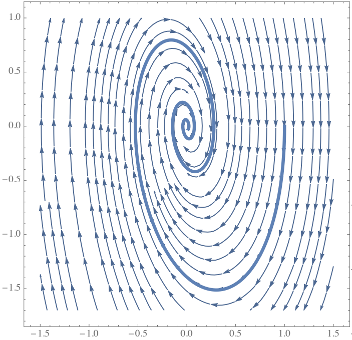
if new $y' = g(x,y)$ has a global solution

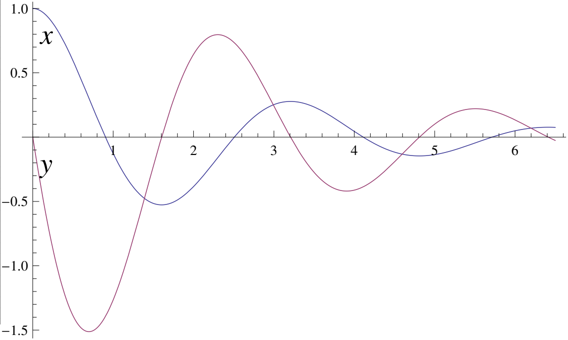




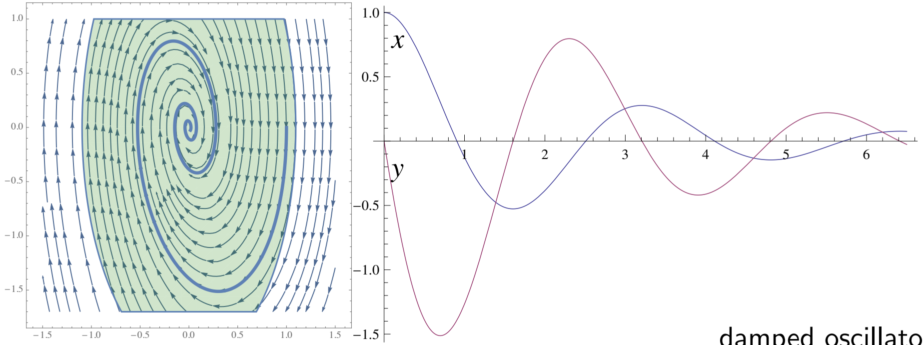

JLogComput'10,LMCS'12, LICS'12,JAR'17

$$\frac{}{\omega^2x^2+y^2\le c^2 \vdash [x' = y, y' = -\omega^2 x - 2d\omega y \,\&\, \omega\ge 0 \wedge d\ge 0]\, \omega^2x^2+y^2\le c^2}$$



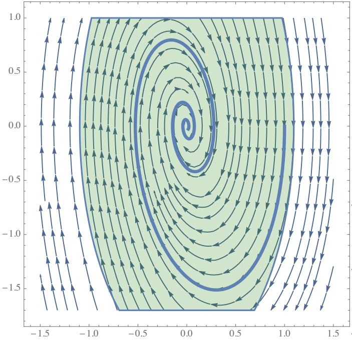



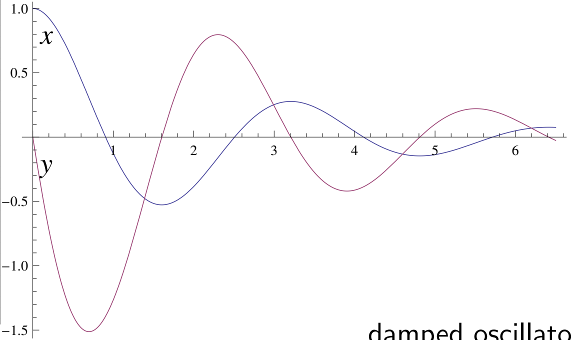
$$\frac{}{\omega^2x^2+y^2\le c^2 \vdash [x' = y, y' = -\omega^2 x - 2d\omega y \,\&\, \omega\ge 0 \land d\ge 0]\, \omega^2x^2+y^2\le c^2}$$




damped oscillator

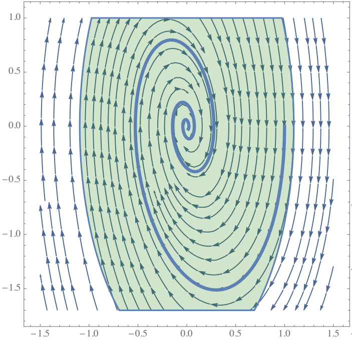
$$\dfrac{\omega\geq 0 \land d\geq 0 \vdash [x':=y][y':=-\omega^2 x - 2d\omega y]\, 2\omega^2 x x' + 2 y y' \leq 0}{\omega^2x^2+y^2\leq c^2 \vdash [x' = y, y' = -\omega^2 x - 2d\omega y \,\&\, \omega\geq 0 \land d\geq 0]\, \omega^2x^2+y^2\leq c^2}$$

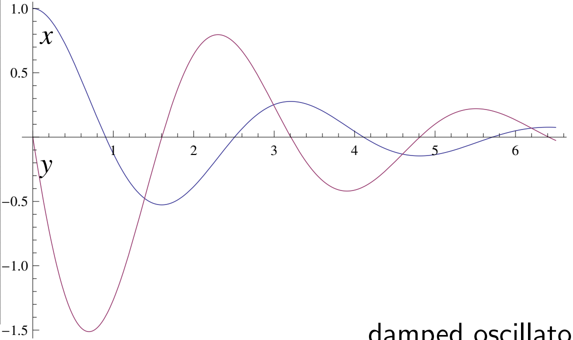



damped oscillator

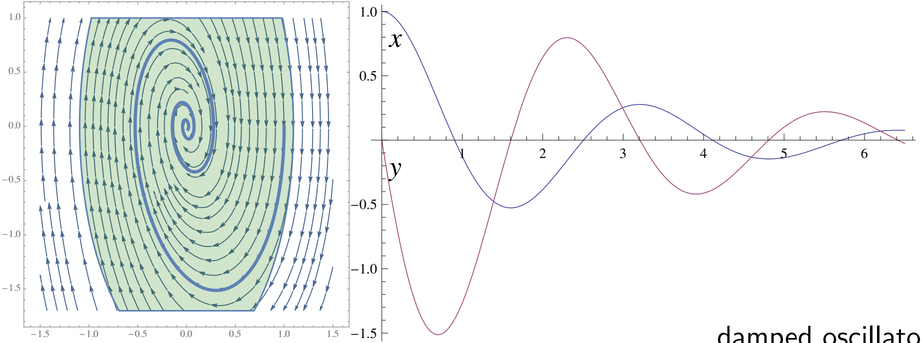
$$\dfrac{\dfrac{\omega{\geq}0 \land d{\geq}0 \vdash 2\omega^2 x y + 2y(-\omega^2 x - 2d\omega y) \leq 0}{\omega{\geq}0 \land d{\geq}0 \vdash [x'{:=}y][y'{:=}-\omega^2 x - 2d\omega y]\, 2\omega^2 x x' + 2y y' \leq 0}}{\omega^2 x^2 + y^2 \leq c^2 \vdash [x' = y, y' = -\omega^2 x - 2d\omega y \,\&\, \omega{\geq}0 \land d{\geq}0]\, \omega^2 x^2 + y^2 \leq c^2}$$




damped oscillator

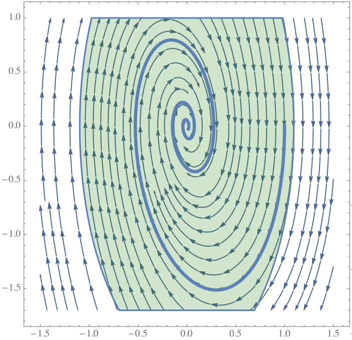
$$\dfrac{\dfrac{\dfrac{*}{\omega\geq 0 \land d\geq 0 \vdash 2\omega^2 xy + 2y(-\omega^2 x - 2d\omega y) \leq 0}}{\omega\geq 0 \land d\geq 0 \vdash [x':=y][y':=-\omega^2 x - 2d\omega y]\, 2\omega^2 x x' + 2yy' \leq 0}}{\omega^2x^2+y^2\leq c^2 \vdash [x' = y, y' = -\omega^2 x - 2d\omega y \,\&\, \omega\geq 0 \land d\geq 0]\, \omega^2x^2+y^2\leq c^2}$$

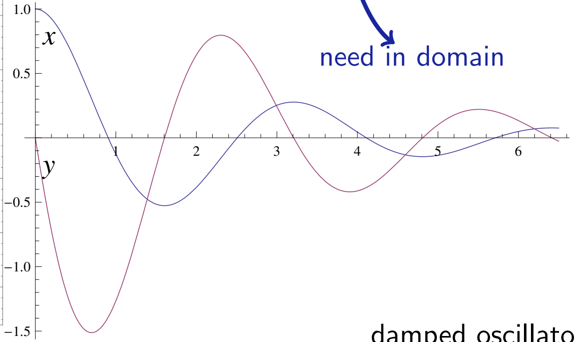



damped oscillator

$$\frac{\dfrac{\dfrac{*}{\omega\geq 0 \land d\geq 0 \vdash 2\omega^2xy + 2y(-\omega^2x - 2d\omega y) \leq 0}}{\omega\geq 0 \land d\geq 0 \vdash [x':=y][y':=-\omega^2x - 2d\omega y]\,2\omega^2xx' + 2yy' \leq 0}}{\omega^2x^2+y^2\leq c^2 \vdash [x' = y, y' = -\omega^2x - 2d\omega y \,\&\, \omega\geq 0 \land d\geq 0]\,\omega^2x^2+y^2\leq c^2}$$

need in domain



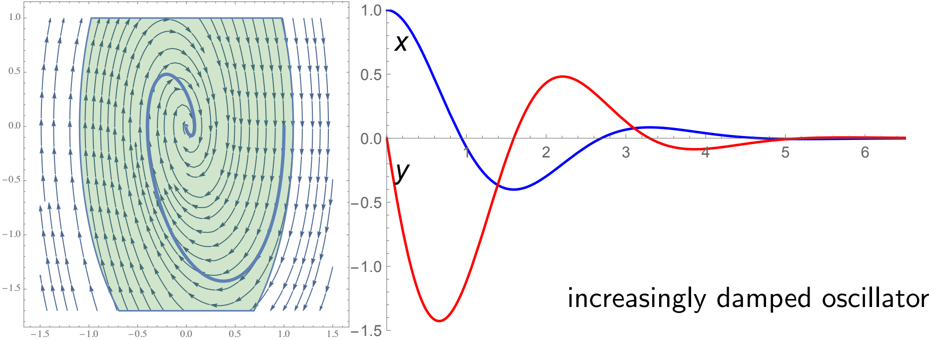




damped oscillator

# Differential Cuts for Differential Equations

$$\overline{\omega^2x^2+y^2\leq c^2 \vdash [x' = y, y' = -\omega^2 x - 2d\omega y, d'=7 \,\&\, \omega\geq 0]\, \omega^2x^2+y^2\leq c^2}$$

$$\omega^2x^2+y^2\le c^2 \vdash [x' = y, y' = -\omega^2 x - 2d\omega y, d'=7 \,\&\, \omega\ge 0]\, \omega^2x^2+y^2\le c^2$$



$$\dfrac{\omega^2x^2+y^2\leq c^2 \vdash [x'=y, y'=-\omega^2x-2d\omega y, d'=7 \,\&\, \omega\geq0\land d\geq0]\,\omega^2x^2+y^2\leq c^2}{\omega^2x^2+y^2\leq c^2 \vdash [x'=y, y'=-\omega^2x-2d\omega y, d'=7 \,\&\, \omega\geq0]\,\omega^2x^2+y^2\leq c^2}$$

increasingly damped oscillator

# Differential Cuts for Differential Equations

$$\dfrac{\omega^2x^2+y^2\le c^2 \vdash [x'=y, y'=-\omega^2x-2d\omega y, d'=7 \,\&\, \omega\ge0\land d\ge0]\,\omega^2x^2+y^2\le c^2}{\omega^2x^2+y^2\le c^2 \vdash [x'=y, y'=-\omega^2x-2d\omega y, d'=7 \,\&\, \omega\ge0]\,\omega^2x^2+y^2\le c^2}$$

ask

$$\dfrac{}{d\ge0 \vdash [x'=y, y'=-\omega^2x-2d\omega y, d'=7 \,\&\, \omega\ge0]\, d\ge0}$$

increasingly damped oscillator

$$\dfrac{\overline{\omega^2x^2+y^2\leq c^2 \vdash [x' = y, y' = -\omega^2x-2d\omega y, d'=7 \,\&\, \omega\geq 0 \land d\geq 0]\, \omega^2x^2+y^2\leq c^2}}{\omega^2x^2+y^2\leq c^2 \vdash [x' = y, y' = -\omega^2x-2d\omega y, d'=7 \,\&\, \omega\geq 0]\, \omega^2x^2+y^2\leq c^2}$$

$$\dfrac{\overline{\omega\geq 0 \vdash [d':=7]\, d'\geq 0}}{d\geq 0 \vdash [x' = y, y' = -\omega^2x - 2d\omega y, d'=7 \,\&\, \omega\geq 0]\, d\geq 0}$$

increasingly damped oscillator

$$\dfrac{\overline{\omega^2x^2+y^2\le c^2 \vdash [x' = y, y' = -\omega^2x-2d\omega y, d'=7 \,\&\, \omega\ge0\land d\ge0]\,\omega^2x^2+y^2\le c^2}}{\omega^2x^2+y^2\le c^2 \vdash [x' = y, y' = -\omega^2x-2d\omega y, d'=7 \,\&\, \omega\ge0]\,\omega^2x^2+y^2\le c^2}$$

$$\dfrac{\dfrac{\overline{\omega\ge0 \vdash 7\ge0}}{\omega\ge0 \vdash [d':=7]\, d'\ge0}}{d\ge0 \vdash [x' = y, y' = -\omega^2x - 2d\omega y, d'=7 \,\&\, \omega\ge0]\, d\ge0}$$

increasingly damped oscillator

$$\dfrac{\omega^2x^2+y^2\le c^2 \vdash [x'=y, y'=-\omega^2x-2d\omega y, d'=7\,\&\,\omega\ge0\land d\ge0]\,\omega^2x^2+y^2\le c^2}{\omega^2x^2+y^2\le c^2 \vdash [x'=y, y'=-\omega^2x-2d\omega y, d'=7\,\&\,\omega\ge0]\,\omega^2x^2+y^2\le c^2}$$

DC

$$\dfrac{\dfrac{\dfrac{*}{\omega\ge0 \vdash 7\ge0}}{\omega\ge0 \vdash [d':=7]\,d'\ge0}}{d\ge0 \vdash [x'=y, y'=-\omega^2x-2d\omega y, d'=7\,\&\,\omega\ge0]\,d\ge0}$$

increasingly damped oscillator

# Differential Cuts for Differential Equations

$$\frac{\dfrac{\omega{\ge}0\wedge d{\ge}0 \vdash [x':=y][y':=-\omega^2 x - 2d\omega y]\,2\omega^2 x x' + 2 y y' \le 0}{\omega^2x^2{+}y^2{\le}c^2 \vdash [x' = y, y' = -\omega^2 x{-}2d\omega y, d'{=}7\,\&\,\omega{\ge}0\wedge d{\ge}0]\,\omega^2x^2{+}y^2{\le}c^2}}{\omega^2x^2{+}y^2{\le}c^2 \vdash [x' = y, y' = -\omega^2 x{-}2d\omega y, d'{=}7\,\&\,\omega{\ge}0]\,\omega^2x^2{+}y^2{\le}c^2}$$

$$\cfrac{\cfrac{\cfrac{*}{\omega{\ge}0 \vdash 7{\ge}0}}{\omega{\ge}0 \vdash [d':=7]\,d'{\ge}0}}{d{\ge}0 \vdash [x' = y, y' = -\omega^2 x - 2d\omega y, d'{=}7\,\&\,\omega{\ge}0]\,d{\ge}0}$$

increasingly damped oscillator

$$\dfrac{\dfrac{\dfrac{\omega\ge 0\land d\ge 0 \vdash 2\omega^2 xy + 2y(-\omega^2 x - 2d\omega y) \le 0}{\omega\ge 0\land d\ge 0 \vdash [x':=y][y':=-\omega^2 x - 2d\omega y]\, 2\omega^2 x x' + 2y y' \le 0}}{\omega^2x^2+y^2\le c^2 \vdash [x' = y, y' = -\omega^2 x - 2d\omega y, d'=7 \,\&\, \omega\ge 0\land d\ge 0]\, \omega^2x^2+y^2\le c^2}}{\omega^2x^2+y^2\le c^2 \vdash [x' = y, y' = -\omega^2 x - 2d\omega y, d'=7 \,\&\, \omega\ge 0]\, \omega^2x^2+y^2\le c^2}$$

$$\dfrac{\dfrac{\dfrac{*}{\omega\ge 0 \vdash 7\ge 0}}{\omega\ge 0 \vdash [d':=7]\, d'\ge 0}}{d\ge 0 \vdash [x' = y, y' = -\omega^2 x - 2d\omega y, d'=7 \,\&\, \omega\ge 0]\, d\ge 0}$$

increasingly damped oscillator

$$\dfrac{\dfrac{\dfrac{\dfrac{*}{\omega\geq 0\land d\geq 0 \vdash 2\omega^2xy + 2y(-\omega^2x - 2d\omega y) \leq 0}}{\omega\geq 0\land d\geq 0 \vdash [x':=y][y':=-\omega^2x - 2d\omega y]\, 2\omega^2xx' + 2yy' \leq 0}}{\omega^2x^2+y^2\leq c^2 \vdash [x' = y, y' = -\omega^2x-2d\omega y, d'=7 \,\&\, \omega\geq 0\land d\geq 0]\, \omega^2x^2+y^2\leq c^2}}{\omega^2x^2+y^2\leq c^2 \vdash [x' = y, y' = -\omega^2x-2d\omega y, d'=7 \,\&\, \omega\geq 0]\, \omega^2x^2+y^2\leq c^2}$$

$$\dfrac{\dfrac{\dfrac{*}{\omega\geq 0 \vdash 7\geq 0}}{\omega\geq 0 \vdash [d':=7]\, d'\geq 0}}{d\geq 0 \vdash [x' = y, y' = -\omega^2x - 2d\omega y, d'=7 \,\&\, \omega\geq 0]\, d\geq 0}$$

increasingly damped oscillator

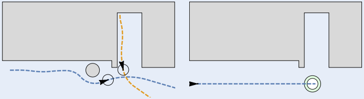
# Differential Cuts for Differential Equations

$$\dfrac{\dfrac{\dfrac{\dfrac{*}{\omega\geq 0\land d\geq 0 \vdash 2\omega^2xy + 2y(-\omega^2x - 2d\omega y) \leq 0}}{\omega\geq 0\land d\geq 0 \vdash [x':=y][y':=-\omega^2x - 2d\omega y]\, 2\omega^2xx' + 2yy' \leq 0}}{\omega^2x^2+y^2\leq c^2 \vdash [x' = y, y' = -\omega^2x-2d\omega y, d'=7\,\&\,\omega\geq 0\land d\geq 0]\,\omega^2x^2+y^2\leq c^2}}{\omega^2x^2+y^2\leq c^2 \vdash [x' = y, y' = -\omega^2x-2d\omega y, d'=7\,\&\,\omega\geq 0]\,\omega^2x^2+y^2\leq c^2}$$

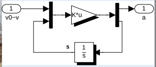
$$\dfrac{\dfrac{\dfrac{*}{\omega\geq 0 \vdash 7\geq 0}}{\omega\geq 0 \vdash [d':=7]\, d'\geq 0}}{d\geq 0 \vdash [x' = y, y' = -\omega^2x - 2d\omega y, d'=7\,\&\,\omega\geq 0]\, d\geq 0}$$

Could repeatedly diffcut in formulas to help the proof

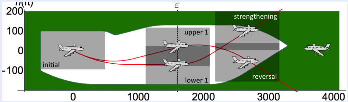
## Obstacle Avoidance + Ground Navigation

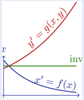
## Train Control Brakes


## Airborne Collision Avoidance (ACAS X)


## Ship Cooling

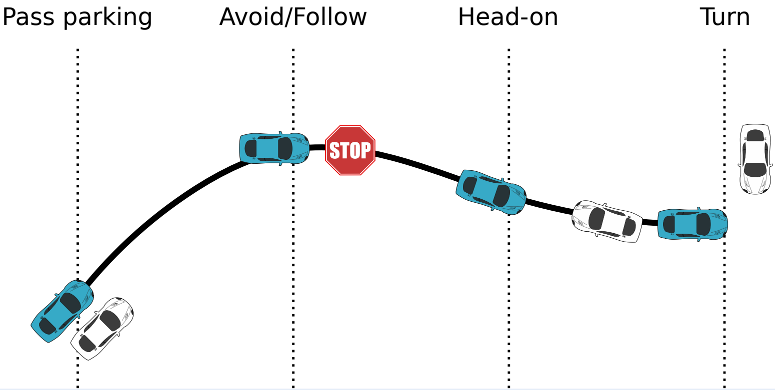




NSF DARPA BOSCH SIEMENS

JOHNS HOPKINS APPLIED PHYSICS LABORATORY

# Ground Robot Obstacle Avoidance: Verify

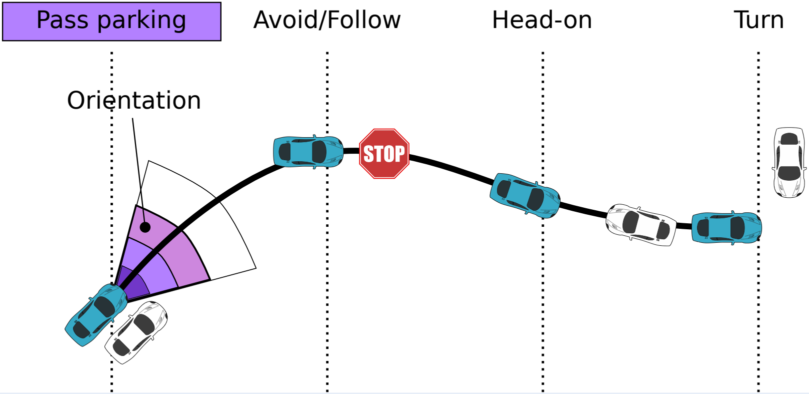
- Fundamental safety question for ground robot navigation
- When will which control decision avoid obstacles?
- Depends on safety objective, physical capabilities of robot + obstacle




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

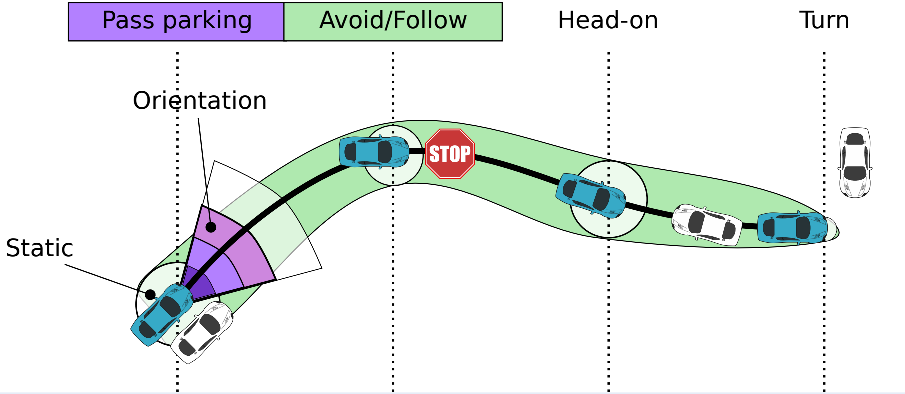
- Fundamental safety question for ground robot navigation
- When will which control decision avoid obstacles?
- Depends on safety objective, physical capabilities of robot + obstacle




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

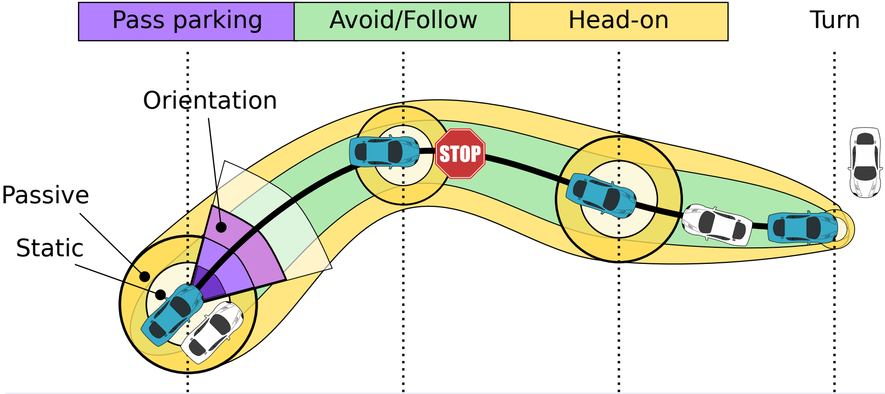
- Fundamental safety question for ground robot navigation
- When will which control decision avoid obstacles?
- Depends on safety objective, physical capabilities of robot + obstacle




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

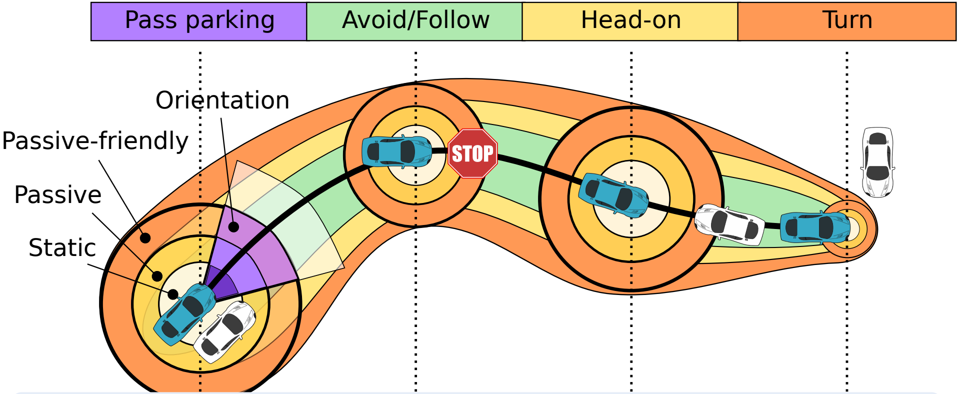
- Fundamental safety question for ground robot navigation
- When will which control decision avoid obstacles?
- Depends on safety objective, physical capabilities of robot + obstacle




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
- When will which control decision avoid obstacles?
- Depends on safety objective, physical capabilities of robot + obstacle




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

# Ground Robot Invariants and Safe Control Constraints

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

RSS'13,IJRR'17

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

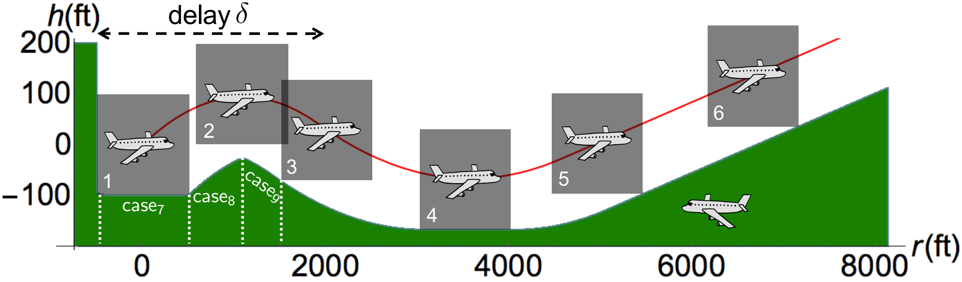
**Question**

How to find and justify constraints? Proof!

RSS'13,IJRR'17

# 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

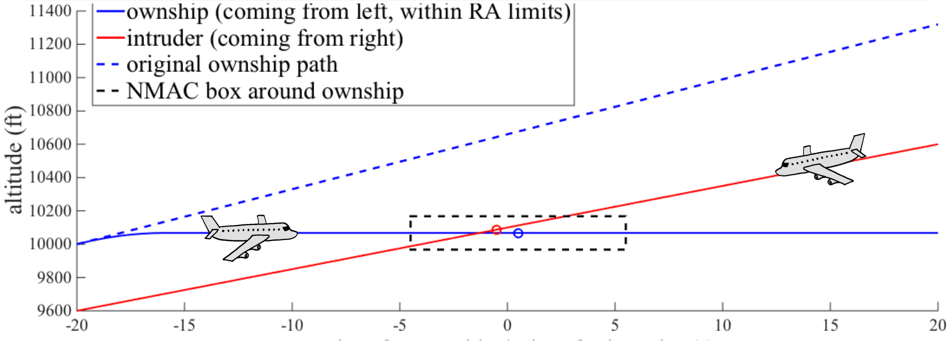



1. Identified safe region for each advisory symbolically
2. Proved safety for hybrid systems flight model in KeYmaera X

TACAS'15,EMSOFT'15,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).

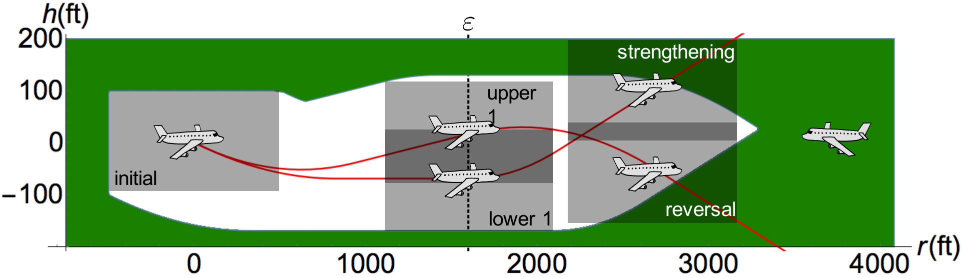



ACAS X issues DNC advisory, which induces collision unless corrected

TACAS'15,EMSOFT'15,STTT'17

# Airborne Collision Avoidance System ACAS X: Refine

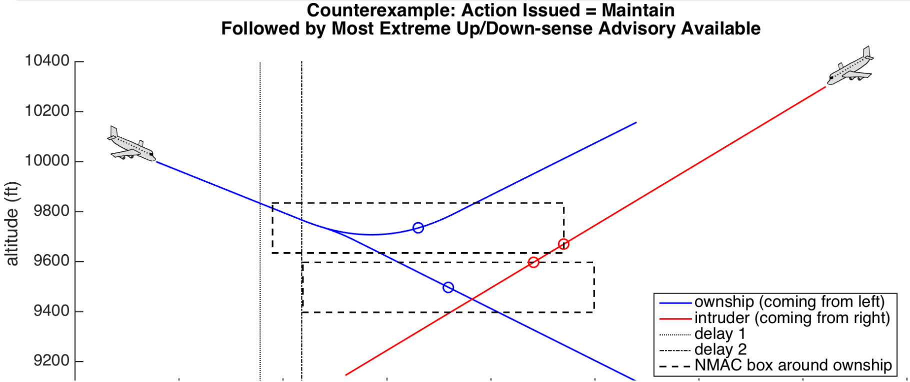
- Conservative, so too many counterexamples
- Settle for: safe for a little while, with safe future advisory possibility
- Safeable advisory: a subsequent advisory can safely avoid collision




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

STTT'17

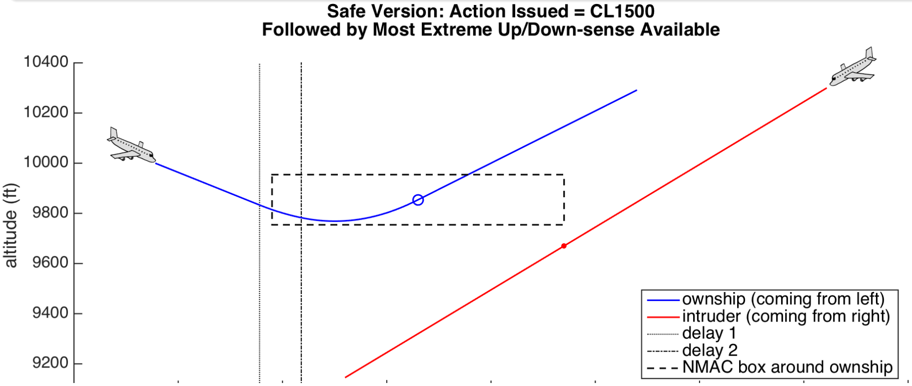
ACAS X table comparison shows safeable advisory in more of the 648,591,384,375 states compared (≈899 $10^6$ counterexamples).




ACAS X issues Maintain advisory instead of CL1500

STTT'17

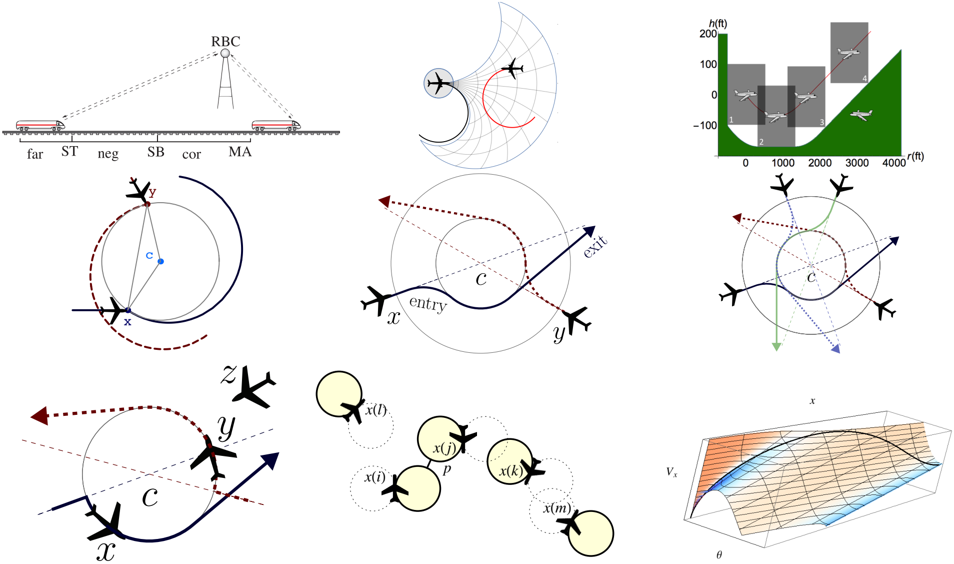
ACAS X table comparison shows safeable advisory in more of the 648,591,384,375 states compared (≈899 $10^6$ counterexamples).




ACAS X issues Maintain advisory instead of CL1500

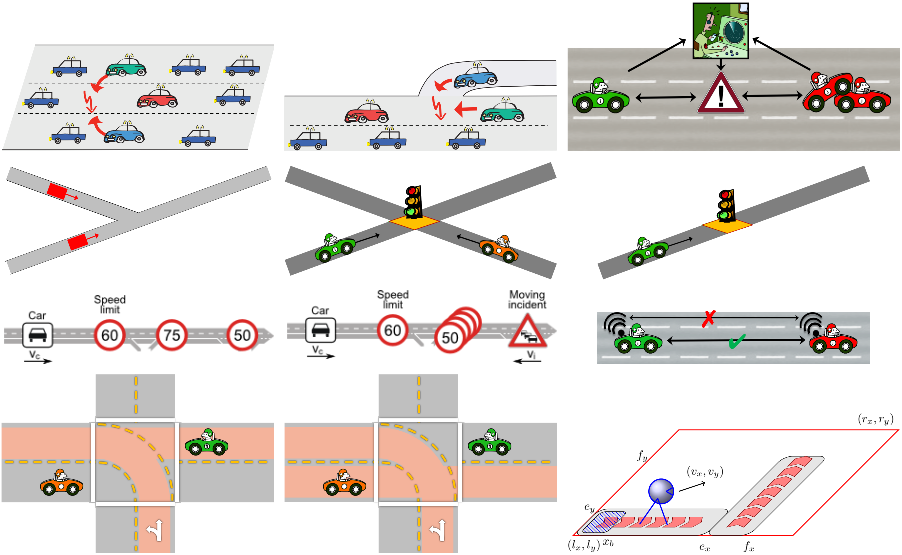
STTT'17

# Verified CPS Applications: Trains & Airplanes



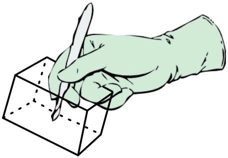

'09,JAIS'14,TACAS'15,EMSOFT'15,FM'09,HSCC'11,HSCC'13,TACAS'14, RSSRail'17

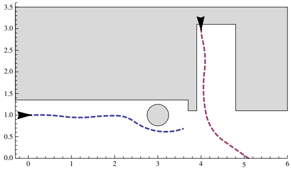
# Verified CPS Applications: Cars

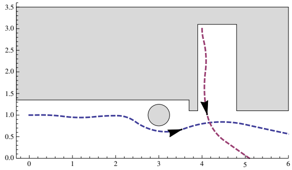


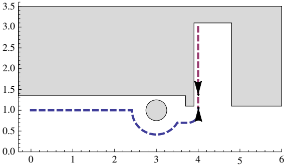

FM'11,LMCS'12,ICCPS'12,ITSC'11,ITSC'13,IJCAR'12

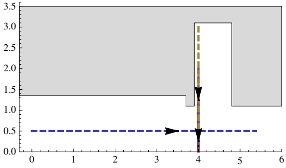
# Verified CPS Applications: Robots

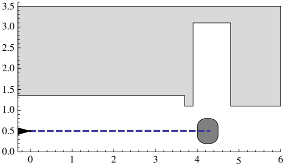


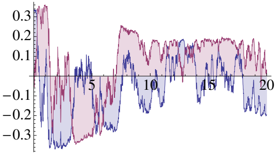


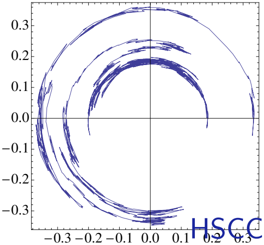


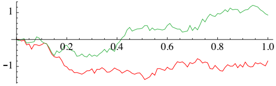




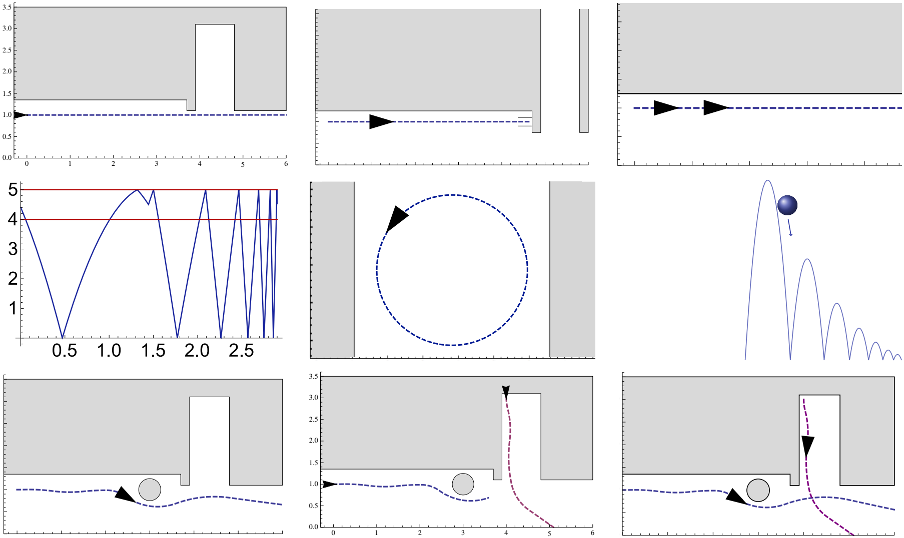








HSCC'13,RSS'13,CADE'12, IJRR'17



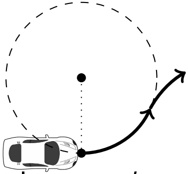

undergrads in *Foundations of Cyber-Physical Systems* course

# KeYmaera X aXiomatic Tactical Theorem Prover for CPS

Proof search →

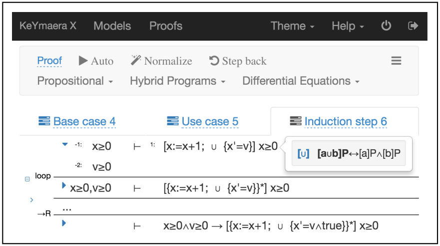
## KeYmaera X

Model



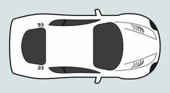
ctrl: $a := -b;$

plant: $x'' = a$




**generates proofs**

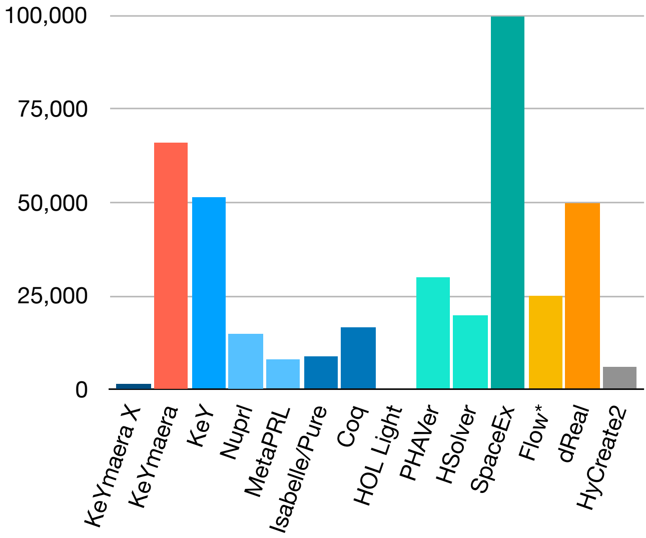
ModelPlex proof →



Model Safety

Compliance Monitor

| Trustworthy | Flexible | Customizable |
|---|---|---|
| Uniform substitution | Proof automation | Scala+Java API |
| Sound & complete | Interactive UI | Command line |
| Small core: 1700 LOC | Programmable | REST API |




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

# Acknowledgments

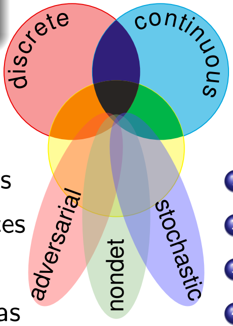
Students and postdocs of the Logical Systems Lab at Carnegie Mellon
Brandon Bohrer, Nathan Fulton, Sarah Loos, João Martins, Yong Kiam Tan
Khalil Ghorbal, Jean-Baptiste Jeannin, Stefan Mitsch

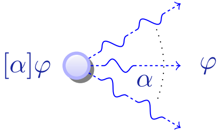
# Logic & Proofs for Cyber-Physical Systems

Logical foundations make a big difference for CPS, and vice versa

**differential dynamic logic**

$$d\mathcal{L} = DL + HP$$




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

1. Multi-dynamical systems
2. Combine simple dynamics
3. Tame complexity
4. www.keymaeraX.org

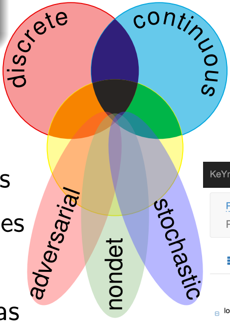
Numerous wonders remain to be discovered

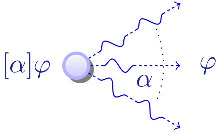
# Logic & Proofs for Cyber-Physical Systems

## Logical foundations make a big difference for CPS, and vice versa

**differential dynamic logic**

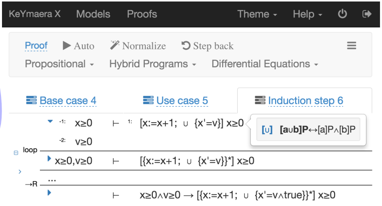
$$d\mathcal{L} = DL + HP$$




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

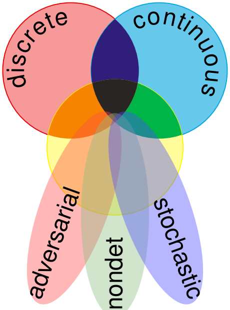
## KeYmaera X



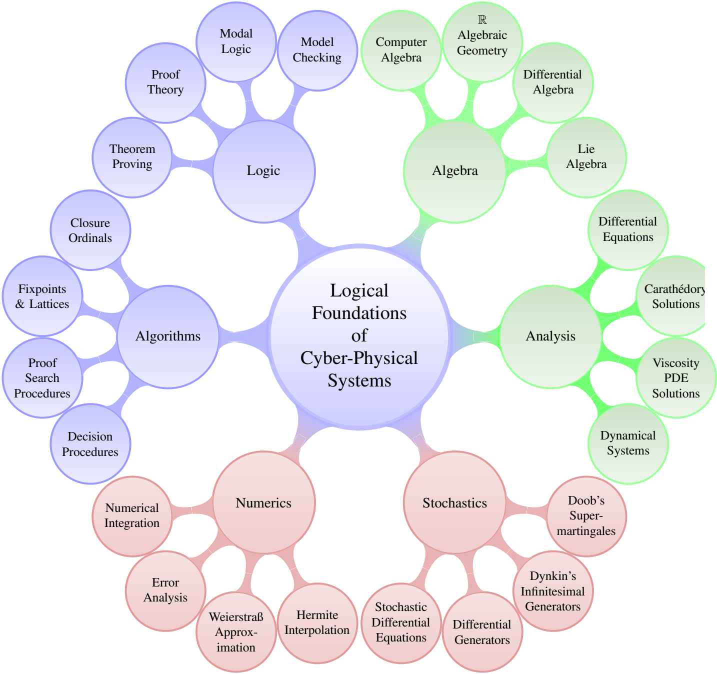

Numerous wonders remain to be discovered

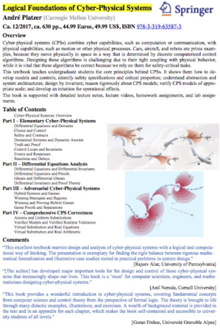
Numerous wonders remain to be discovered

- Scalable continuous stochastics CADE'11
- Concurrent CPS
- Real arithmetic: Scalable and verified CADE'09
- Verified CPS implementations, ModelPlex FMSD'16
- Correct CPS execution
- CPS-conducive tactic languages+libraries ITP'17
- Tactics exploiting CPS structure/linearity/...
- Invariant generation FMSD'09 TACAS'14
- Tactics & proofs for reachable set computations
- Parallel proof search & disprovers
- Correct model transformation FM'14
- Inspiring applications




CPSs deserve proofs as safety evidence!



**Logical Foundations of Cyber-Physical Systems** Springer

**André Platzer** [Carnegie Mellon University]

**Ca. 12/2017, ca. 630 pp., 44.99 Euros, 49.99 US$, ISBN 978-3-319-63587-3**

**Overview**

Cyber-physical systems (CPSs) combine cyber capabilities, such as computation or communication, with physical capabilities, such as motion or other physical processes. Cars, aircraft, and robots are prime examples, because they move physically in space in a way that is determined by discrete computerized control algorithms. Designing these algorithms is challenging due to their tight coupling with physical behavior, while it is vital that these algorithms be correct because we rely on them for safety-critical tasks.

This textbook teaches undergraduate students the core principles behind CPSs. It shows them how to develop models and controls; identify safety specifications and critical properties; understand abstraction and system architectures; design by invariant; reason rigorously about CPS models; verify CPS models of appropriate scale; and develop an intuition for operational effects.

The book is supported with detailed lecture notes, lecture videos, homework assignments, and lab assignments.

**Table of Contents**



**Comments**

"This excellent textbook marries design and analysis of cyber-physical systems with a logical and computational way of thinking. The presentation is exemplary for finding the right balance between rigorous mathematical formalization and illustrative case studies rooted in practical problems in system design."

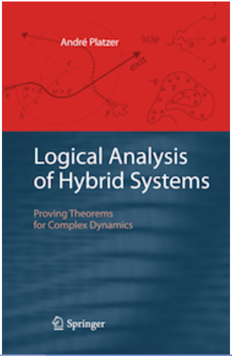
[Rajeev Alur, University of Pennsylvania]

"[The author] has developed major important tools for the design and control of those cyber-physical systems that increasingly shape our lives. This book is a 'must' for computer scientists, engineers, and mathematicians designing cyber-physical systems."

[Anil Nerode, Cornell University]

"This book provides a wonderful introduction to cyber-physical systems, covering fundamental concepts from computer science and control theory from the perspective of formal logic. The theory is brought to life through many didactic examples, illustrations, and exercises. A wealth of background material is provided in the text and in an appendix for each chapter, which makes the book self-contained and accessible to university students of all levels."

[Goran Frehse, Université Grenoble Alpes]



# Differential Dynamic Logic d$\mathcal{L}$: Semantics

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

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

compositional semantics

**Definition (d$\mathcal{L}$ semantics)** ($[\![\cdot]\!] : \mathrm{Fml} \to \wp(\mathcal{S})$)

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

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

André Platzer.
Logics of dynamical systems.
In LICS [34], pages 13–24.
doi:10.1109/LICS.2012.13.

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

André Platzer.
Differential hybrid games.
*ACM Trans. Comput. Log.*, 18(3):19:1–19:44, 2017.
doi:10.1145/3091123.

André Platzer.
The complete proof theory of hybrid systems.
In LICS [34], pages 541–550.
doi:10.1109/LICS.2012.64.

André Platzer.
A complete axiomatization of quantified differential dynamic logic for distributed hybrid systems.
*Log. Meth. Comput. Sci.*, 8(4):1–44, 2012.

Special issue for selected papers from CSL'10.
doi:10.2168/LMCS-8(4:17)2012.

André Platzer.
Stochastic differential dynamic logic for stochastic hybrid programs.
In Nikolaj Bjørner and Viorica Sofronie-Stokkermans, editors, *CADE*, volume 6803 of *LNCS*, pages 431–445. Springer, 2011.
doi:10.1007/978-3-642-22438-6_34.

André Platzer.
A uniform substitution calculus for differential dynamic logic.
In Felty and Middeldorp [35], pages 467–481.
doi:10.1007/978-3-319-21401-6_32.

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):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. Springer, 2012.
doi:10.1007/978-3-642-32347-8_3.

Stefan Mitsch, Khalil Ghorbal, David Vogelbacher, and André Platzer.
Formal verification of obstacle avoidance and navigation of ground robots.
*I. J. Robotics Res.*, 2017.

André Platzer and Jan-David Quesel.

European Train Control System: A case study in formal verification.
In Karin Breitman and Ana Cavalcanti, editors, *ICFEM*, volume 5885 of *LNCS*, pages 246–265. Springer, 2009.
doi:10.1007/978-3-642-10373-5_13.

Stefan Mitsch, Marco Gario, Christof J. Budnik, Michael Golm, and André Platzer.
Formal verification of train control with air pressure brakes.
In Alessandro Fantechi, Thierry Lecomte, and Alexander Romanovsky, editors, *RSSRail*, LNCS. Springer, 2017.

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*, 2016.
doi:10.1007/s10009-016-0434-1.

Jean-Baptiste Jeannin, Khalil Ghorbal, Yanni Kouskoulas, Ryan Gardner, Aurora Schmidt, Erik Zawadzki, and André Platzer.

A formally verified hybrid system for the next-generation airborne collision avoidance system.
In Christel Baier and Cesare Tinelli, editors, *TACAS*, volume 9035 of *LNCS*, pages 21–36. Springer, 2015.
doi:10.1007/978-3-662-46681-0_2.

Jean-Baptiste Jeannin, Khalil Ghorbal, Yanni Kouskoulas, Ryan Gardner, Aurora Schmidt, Erik Zawadzki, and André Platzer.
Formal verification of ACAS X, an industrial airborne collision avoidance system.
In Alain Girault and Nan Guan, editors, *EMSOFT*, pages 127–136. IEEE, 2015.
doi:10.1109/EMSOFT.2015.7318268.

Nathan Fulton, Stefan Mitsch, Jan-David Quesel, Marcus Völp, and André Platzer.
KeYmaera X: An axiomatic tactical theorem prover for hybrid systems.
In Felty and Middeldorp [35], pages 527–538.
doi:10.1007/978-3-319-21401-6_36.

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

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

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

André Platzer.

*Logical Foundations of Cyber-Physical Systems*.
Springer, Switzerland, 2017.
URL: http://www.springer.com/978-3-319-63587-3.

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

Thomas A. Henzinger.
The theory of hybrid automata.
In *LICS*, pages 278–292, Los Alamitos, 1996. IEEE Computer Society.
doi:10.1109/LICS.1996.561342.

Jennifer M. Davoren and Anil Nerode.
Logics for hybrid systems.
*IEEE*, 88(7):985–1010, 2000.

Ashish Tiwari.
Abstractions for hybrid systems.

*Form. Methods Syst. Des.*, 32(1):57–83, 2008.
doi:10.1007/s10703-007-0044-3.

Jan Lunze and Françoise Lamnabhi-Lagarrigue, editors.
*Handbook of Hybrid Systems Control: Theory, Tools, Applications*.
Cambridge Univ. Press, 2009.

Paulo Tabuada.
*Verification and Control of Hybrid Systems: A Symbolic Approach*.
Springer, 2009.

Rajeev Alur.
*Principles of Cyber-Physical Systems*.
MIT Press, 2015.

Laurent Doyen, Goran Frehse, George J. Pappas, and André Platzer.
Verification of hybrid systems.
In Edmund M. Clarke, Thomas A. Henzinger, Helmut Veith, and Roderick Bloem, editors, *Handbook of Model Checking*, chapter 30.
Springer, 2017.
doi:10.1007/978-3-319-10575-8_30.

*Proceedings of the 27th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2012, Dubrovnik, Croatia, June 25–28, 2012*. IEEE, 2012.

Amy Felty and Aart Middeldorp, editors.
*International Conference on Automated Deduction, CADE'15, Berlin, Germany, Proceedings*, volume 9195 of *LNCS*. Springer, 2015.