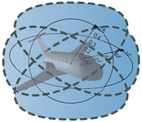
Refinements of Hybrid Dynamical Systems Logic

André Platzer

Karlsruhe Institute of Technology

Carnegie Mellon University

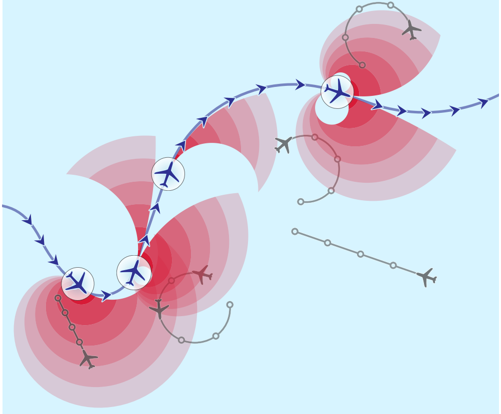
Alexander von HUMBOLDT STIFTUNG



Karlsruhe Institute of Technology

Cyber-Physical Systems Analysis: Aircraft Example

Which control decisions are safe for aircraft collision avoidance?



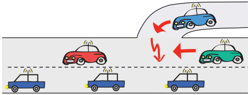
Cyber-Physical Systems

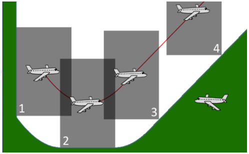
CPSs combine cyber capabilities with physical capabilities
to solve problems that neither part could solve alone.

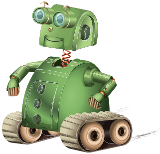
Cyber-Physical Systems Promise Transformative Impact!

Prospects: Safety & Efficiency

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







Cyber-Physical Systems

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

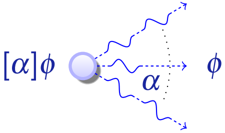
Outline

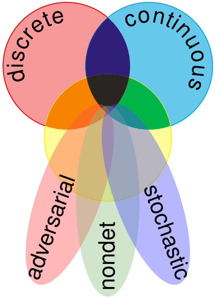
Outline

Dynamic Logics for Dynamical Systems

differential dynamic logic

dL = DL + HP



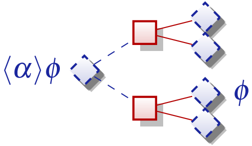


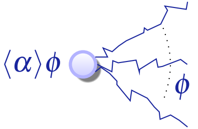
differential game logic

dGL = GL + HG

stochastic differential DL

SdL = DL + SHP





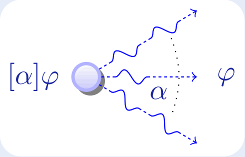
quantified differential DL

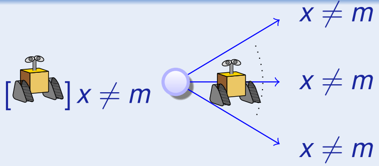
QdL = FOL + DL + QHP

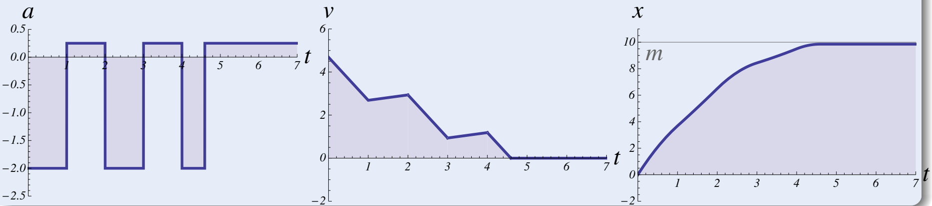
JAR'08,CADE'11,LMCS'12,LICS'12,LICS'12

TOCL'15,JAR'17,TOCL'17,JACM'20

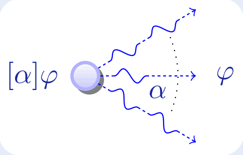
Concept (Differential Dynamic Logic) (JAR'08,LICS'12)

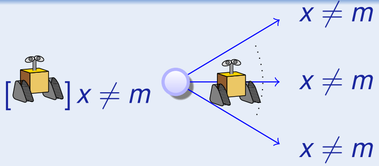






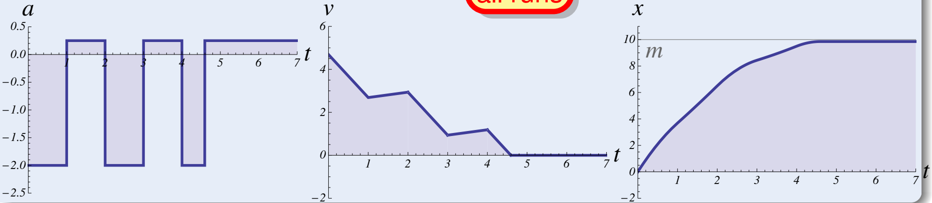
Concept (Differential Dynamic Logic) (JAR'08,LICS'12)



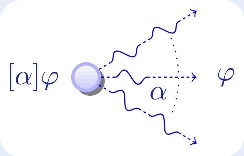


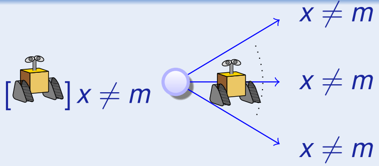
$$\left[\left(\left(\text{if}(SB(x,m))\quad a:=-b\right);\ x'=v, v'=a\right)^*\right] \underbrace{x \neq m}_{\text{post}}$$

all runs



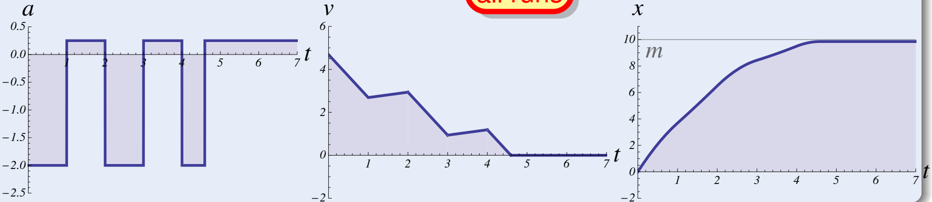
Concept (Differential Dynamic Logic) (JAR'08,LICS'12)





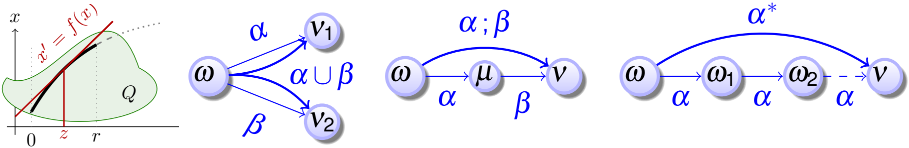
$$\underbrace{x \neq m \land b > 0}_{\text{init}} \rightarrow \Big[\Big(\big(\text{if}(\text{SB}(x,m))\quad a := -b\big)\ ;\ x' = v, v' = a\Big)^*\Big]\underbrace{x \neq m}_{\text{post}}$$

all runs



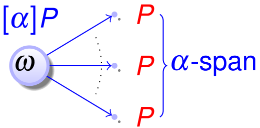
Definition (Hybrid program)

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



Definition (Differential dynamic logic) (JAR'08,LICS'12)

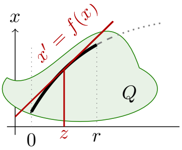
$$P,Q ::= e \geq \tilde{e} \mid \neg P \mid P \wedge Q \mid P \vee Q \mid P \rightarrow Q \mid \forall x\, P \mid \exists x\, P \mid [\alpha]P \mid \langle\alpha\rangle P$$

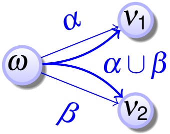


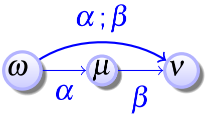
Differential Dynamic Logic dL

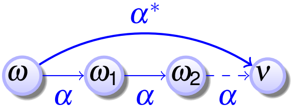
Definition (Hybrid program)

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



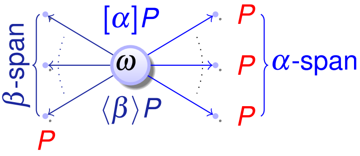






Definition (Differential dynamic logic) (JAR'08,LICS'12)

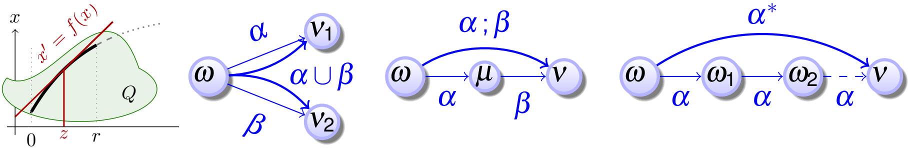
$$P, Q ::= e \geq \tilde{e} \mid \neg P \mid P \wedge Q \mid P \vee Q \mid P \rightarrow Q \mid \forall x\, P \mid \exists x\, P \mid [\alpha]P \mid \langle\alpha\rangle P$$



Differential Dynamic Logic dL

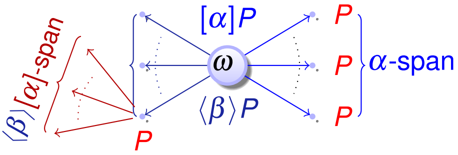
Definition (Hybrid program)

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



Definition (Differential dynamic logic) (JAR'08,LICS'12)

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



Differential Dynamic Logic dL: Semantics

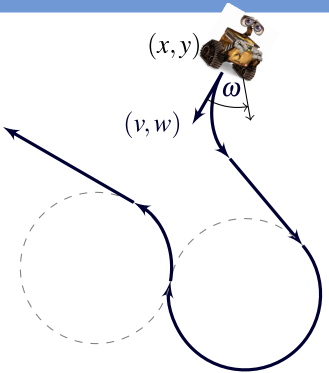
Definition (Hybrid program semantics) $([\![\cdot]\!] : \mathrm{HP} \to \wp(\mathscr{S} \times \mathscr{S}))$

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

compositional semantics

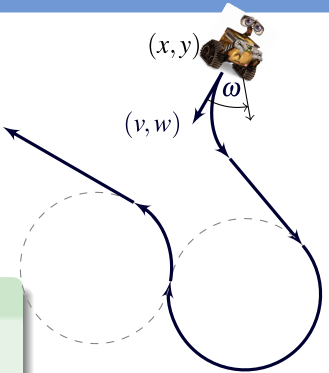
Definition (dL semantics) $([\![\cdot]\!] : \mathrm{Fml} \to \wp(\mathscr{S}))$

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



Example (Runaround Robot)

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



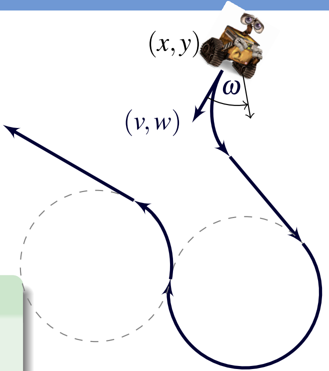
Example (Dubins Path)

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

Example (Runaround Robot)

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

Ex: Runaround Robot



Example (Dubins Path)

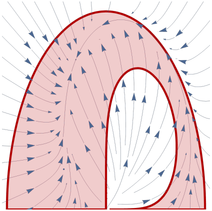
$$v^2 + w^2 \neq 0 \rightarrow \langle ((\omega := -1 \cup \omega := 1 \cup \omega := 0)$$
$$\{x' = v, y' = w, v' = \omega w, w' = -\omega v\})^* \rangle (x, y) = o$$

Example (Runaround Robot)

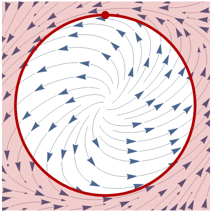
$$(x, y) \neq o \rightarrow \left[((?Q_{-1}; \omega := -1 \cup ?Q_1; \omega := 1 \cup ?Q_0; \omega := 0); \{x' = v, y' = w, v' = \omega w, w' = -\omega v\})^* \right] (x, y) \neq o$$

Differential Dynamic Logic: Example Properties

Safety $Q \to [\alpha]P$

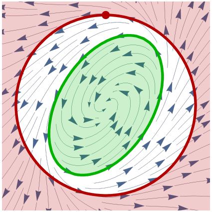


Liveness $Q \to \langle\alpha\rangle P$



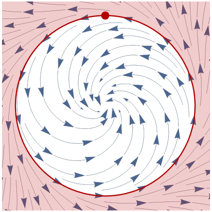
Stability

$$\forall \varepsilon{>}0\, \exists \delta{>}0\, \forall x\, \big(\mathscr{U}_\delta(x=0) \to [x'=f(x)]\mathscr{U}_\varepsilon(x=0)\big)$$



Attractivity

$$\exists \delta{>}0\, \forall x\, \big(\mathscr{U}_\delta(x=0) \to \forall \varepsilon{>}0\, \langle x'=f(x)\rangle[x'=f(x)]\mathscr{U}_\varepsilon(x{=}0)\big)$$



Outline

equations of truth

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

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

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

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

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

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

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

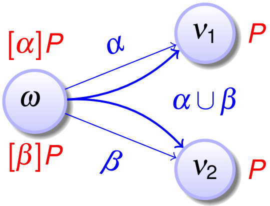
I $[\alpha^*]P \leftrightarrow P \land [\alpha^*](P \rightarrow [\alpha]P)$

laws of logic of
laws of physics

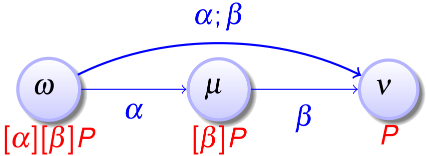
C $[\alpha^*]\forall v>0\, (P(v) \rightarrow \langle\alpha\rangle P(v-1)) \rightarrow \forall v\, (P(v) \rightarrow \langle\alpha^*\rangle \exists v \leq 0\, P(v))$

LICS'12,JAR'17

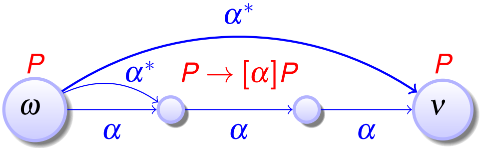
$[\alpha \cup \beta]P \leftrightarrow [\alpha]P \wedge [\beta]P$



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



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



LICS'12,JAR'17

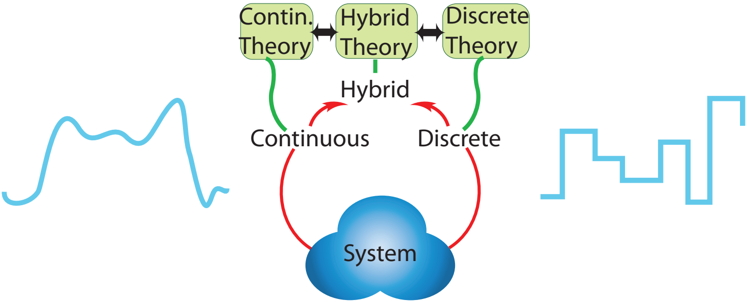
Complete Proof Theory of Hybrid Systems

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

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

Corollary (Complete Proof-theoretical Bridge)

proving continuous = proving hybrid = proving discrete

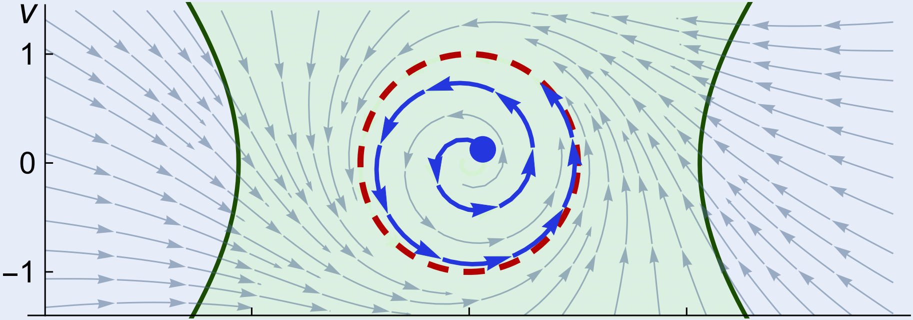


Outline

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

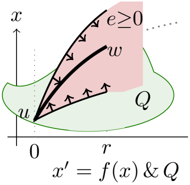
$$u^2 \leq v^2+\frac{9}{2} \rightarrow [u' = -v+\frac{u}{4}(1-u^2-v^2), v' = u+\frac{v}{4}(1-u^2-v^2)]\, u^2 \leq v^2+\frac{9}{2}$$

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

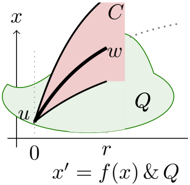


Analyzing ODEs via solutions undoes their descriptive power! Poincaré 1881

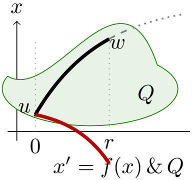
DI $[x' = f(x)]e \geq 0 \leftarrow e \geq 0 \land [x' = f(x)](e)' \geq 0$



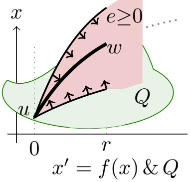
DC $\big([x' = f(x) \,\&\, Q]P \leftrightarrow [x' = f(x) \,\&\, Q \land C]P\big)$
$\leftarrow [x' = f(x) \,\&\, Q]C$



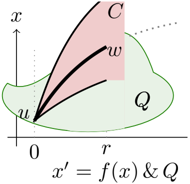
DG $[x' = f(x) \,\&\, Q]P$
$\leftrightarrow \exists y\, [x' = f(x), y' = a(x)y + b(x) \,\&\, Q]P$



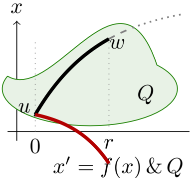
DI $[x'=f(x)]e\geq 0 \leftarrow e\geq 0 \land [x'=f(x)](e)'\geq 0$



DC $$\big([x'=f(x)\,\&\,Q]P \leftrightarrow [x'=f(x)\,\&\,Q\land C]P\big) \leftarrow [x'=f(x)\,\&\,Q]C$$



DG $$[x'=f(x)\,\&\,Q]P \leftrightarrow \exists y\,[x'=f(x), y'=a(x)y+b(x)\,\&\,Q]P$$



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

Differential Equation Axiomatization

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

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

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

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

Differential Equation Axiomatization

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

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

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

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

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

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

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

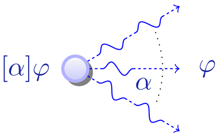
$$e'^{*} = 0 \equiv e{=}0 \land (e')'^{*}{=}0 \qquad (P \land Q)'^{*} \equiv P'^{*} \land Q'^{*}$$

$$e'^{*} \geq 0 \equiv e{\geq}0 \land (e{=}0{\rightarrow}(e')'^{*}{\geq}0) \quad (P \lor Q)'^{*} \equiv P'^{*} \lor Q'^{*}$$

Takeaway: Hybrid Systems Logic

Differential dynamic logic

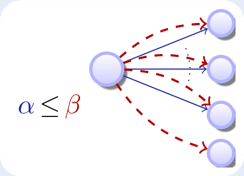
- Logical lingua franca for control systems
- Safety, liveneness, controllability, stability are definable by $[\cdot], \langle\cdot\rangle, \forall, \exists$
- Specification and verification interlinked
- Compositional verification helps scale for well-engineered systems
- Small-core complete axiomatization (2000 LOC)
- Differential equation invariants decidable by dL proof
- Significant applications in KeYmaera X theorem prover



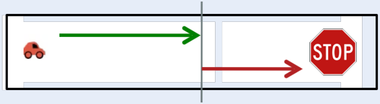
JAR'08,LICS'12,JAR'17

Outline

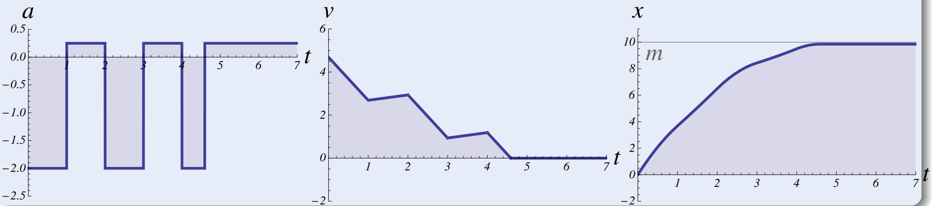
Concept (Differential Refinement Logic) (LICS'16)



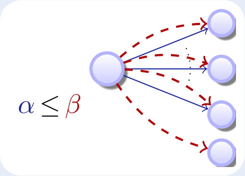
event-triggered



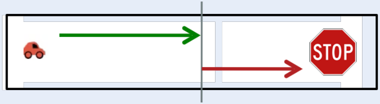
$$(u :\in G(x); x' = f(x) \,\&\, Q(x))^*$$



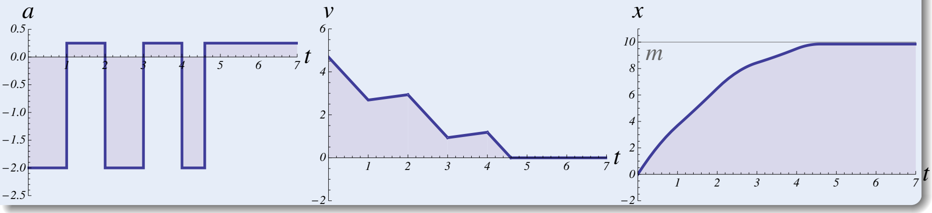
Concept (Differential Refinement Logic) (LICS'16)



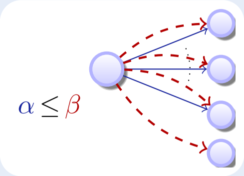
event-triggered



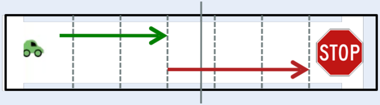
$$[(u :\in G(x); x' = f(x) \,\&\, Q(x))^*]\,\mathrm{safe}$$



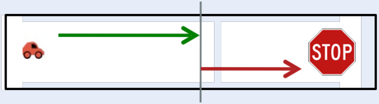
Concept (Differential Refinement Logic) (LICS'16)



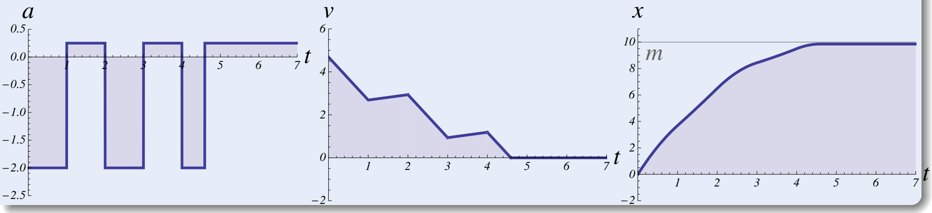
time-triggered



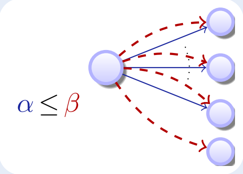
event-triggered



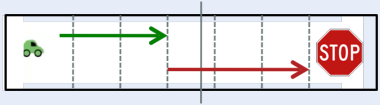
$[(u := g(x); x' = f(x) \,\&\, t \leq T)^*]\text{safe}$ $\quad$ $[(u :\in G(x); x' = f(x) \,\&\, Q(x))^*]\text{safe}$



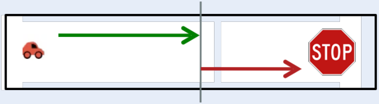
Concept (Differential Refinement Logic) (LICS'16)



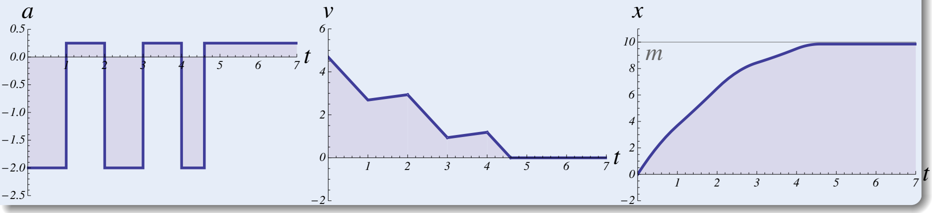
time-triggered
implementable



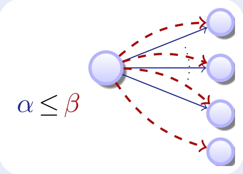
event-triggered
verifiable



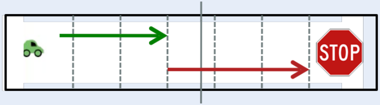
$[(u := g(x); x' = f(x) \,\&\, t \leq T)^*]\text{safe}$ $\quad [(u :\in G(x); x' = f(x) \,\&\, Q(x))^*]\text{safe}$



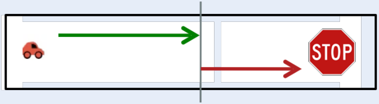
Concept (Differential Refinement Logic) (LICS'16)



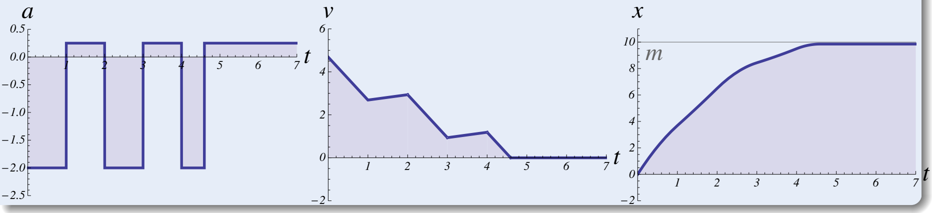
time-triggered
implementable



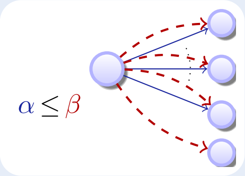
event-triggered
verifiable



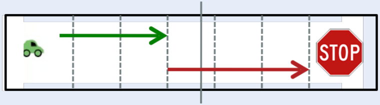
$$[(u := g(x); x' = f(x) \,\&\, t \leq T)^*]\,\text{safe} \leftarrow [(u :\in G(x); x' = f(x) \,\&\, Q(x))^*]\,\text{safe}$$



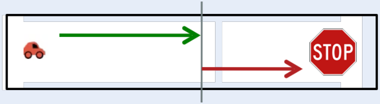
Concept (Differential Refinement Logic) (LICS'16)



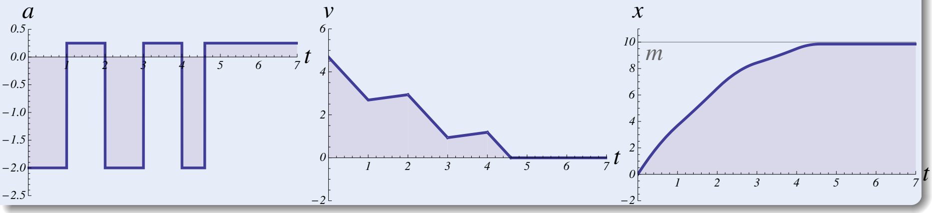
time-triggered
implementable



event-triggered
verifiable



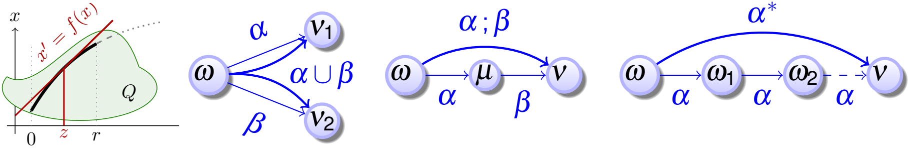
$$(u := g(x); x' = f(x) \,\&\, t \leq T)^* \quad \leq \quad (u :\in G(x); x' = f(x) \,\&\, Q(x))^*$$



Differential Refinement Logic dRL

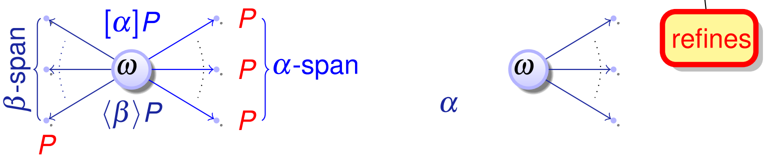
Definition (Hybrid program)

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



Definition (Differential refinement logic) (LICS'16)

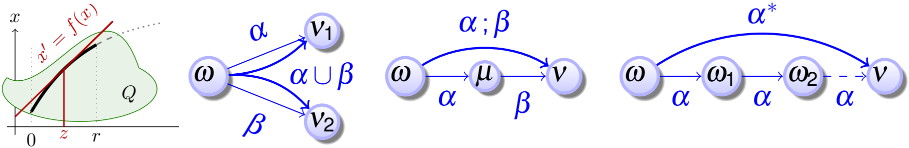
$$P, Q ::= e \geq \tilde{e} \mid \neg P \mid P \wedge Q \mid \ \mid P \to Q \mid \forall x\, P \mid \exists x\, P \mid [\alpha]P \mid \langle\alpha\rangle P \mid \alpha \leq \beta$$



Differential Refinement Logic dRL

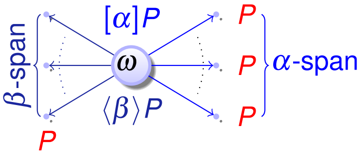
Definition (Hybrid program)

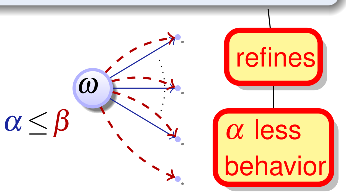
$$\alpha, \beta \ ::= \ x := e \mid ?Q \mid x' = f(x) \,\&\, Q \mid \alpha \cup \beta \mid \alpha ; \beta \mid \alpha^*$$



Definition (Differential refinement logic) (LICS'16)

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





Differential Refinement Logic dRL: Semantics

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

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

compositional semantics

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

$$\begin{aligned}
[\![\alpha \leq \beta]\!] &= \{\omega : \{\nu : (\omega, \nu) \in [\![\alpha]\!]\} \subseteq \{\nu : (\omega, \nu) \in [\![\beta]\!]\}\} \\
[\![e \geq \tilde{e}]\!] &= \{\omega : \omega[\![e]\!] \geq \omega[\![\tilde{e}]\!]\} \\
[\![\neg P]\!] &= [\![P]\!]^\complement \\
[\![P \land Q]\!] &= [\![P]\!] \cap [\![Q]\!] \\
[\![\langle\alpha\rangle P]\!] &= [\![\alpha]\!] \circ [\![P]\!] = \{\omega : \nu \in [\![P]\!] \text{ for some } \nu : (\omega, \nu) \in [\![\alpha]\!]\} \\
[\![[\alpha] P]\!] &= [\![\neg\langle\alpha\rangle\neg P]\!] = \{\omega : \nu \in [\![P]\!] \text{ for all } \nu : (\omega, \nu) \in [\![\alpha]\!]\}
\end{aligned}$$

Differential Refinement Logic: Axiomatization

$$[\leq]\quad \frac{P \to [\alpha]Q \qquad P \to \gamma \leq \alpha}{P \to [\gamma]Q}$$

$$\langle\leq\rangle\quad \frac{P \to \langle\alpha\rangle Q \qquad P \to \alpha \leq \gamma}{P \to \langle\gamma\rangle Q}$$

$$(;)\quad \frac{P \to \alpha_1 \leq \alpha_2 \qquad P \to [\alpha_1](\beta_1 \leq \beta_2)}{P \to (\alpha_1;\beta_1) \leq (\alpha_2;\beta_2)}$$

$$(\cup)_l\quad \frac{P \to \alpha_1 \leq \beta \land \alpha_2 \leq \beta}{P \to \alpha_1 \cup \alpha_2 \leq \beta}$$

$$({}^*)_l\quad \frac{P \to [\alpha^*](\alpha;\gamma \leq \gamma) \qquad P \to [\alpha^*](\beta \leq \gamma)}{P \to \alpha^*;\beta \leq \gamma}$$

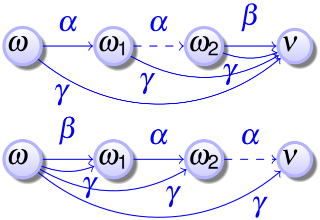
$$({}^*)_r\quad \frac{P \to \beta \leq \gamma \qquad P \to \gamma;\alpha \leq \gamma}{P \to \beta;\alpha^* \leq \gamma}$$

$$({}^*)\quad \frac{P \to [\alpha^*](\alpha \leq \beta)}{P \to \alpha^* \leq \beta^*}$$

$$\leq\quad \alpha \leq \beta \leftrightarrow \forall y\,(\langle\alpha\rangle x = y \to \langle\beta\rangle x = y)$$

$$\leq'\quad [\alpha]P \leftrightarrow \alpha \leq (x := *; ?P)$$

$$(;)_s\quad \frac{P \to \alpha_1 \leq \alpha_2 \qquad \beta_1 \leq \beta_2}{P \to (\alpha_1;\beta_1) \leq (\alpha_2;\beta_2)}$$



LICS'16

$$[\leq]\quad \frac{P \to [\alpha]Q \quad P \to \gamma \leq \alpha}{P \to [\gamma]Q}$$

Property via refine

$$\langle\leq\rangle\quad \frac{P \to \langle\alpha\rangle Q \quad P \to \alpha \leq \gamma}{P \to \langle\gamma\rangle Q}$$

$$(;)\quad \frac{P \to \alpha_1 \leq \alpha_2 \quad P \to [\alpha_1](\beta_1 \leq \beta_2)}{P \to (\alpha_1;\beta_1) \leq (\alpha_2;\beta_2)}$$

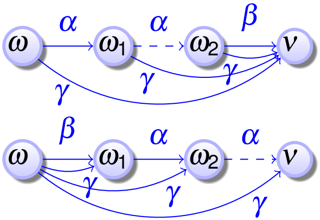
Refine via property

$$(\cup)_l\quad \frac{P \to \alpha_1 \leq \beta \wedge \alpha_2 \leq \beta}{P \to \alpha_1 \cup \alpha_2 \leq \beta}$$

$$(^*)_l\quad \frac{P \to [\alpha^*](\alpha;\gamma \leq \gamma) \quad P \to [\alpha^*](\beta \leq \gamma)}{P \to \alpha^*;\beta \leq \gamma}$$

$$(^*)_r\quad \frac{P \to \beta \leq \gamma \quad P \to \gamma;\alpha \leq \gamma}{P \to \beta;\alpha^* \leq \gamma}$$

$$(^*)\quad \frac{P \to [\alpha^*](\alpha \leq \beta)}{P \to \alpha^* \leq \beta^*}$$



LICS'16

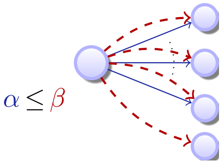
Takeaway: Hybrid System Refinements

Differential refinement logic

- Event-triggered control: Easy to verify but hard to implement
- Time-triggered control: Easy to implement but hard to verify
- Best of both worlds: verify event-triggered, implement time-triggered
- dRL proofs identify required conditions (e.g., event invariance)
- Implementation model $\neq$ verification model
- Iterative design reduces risk, increases repeated effort
- Hierarchical proof structuring by refinement

Relations $\alpha \leq \beta$ between hybrid systems models are just as useful as properties $[\alpha]\varphi$ of hybrid systems models.

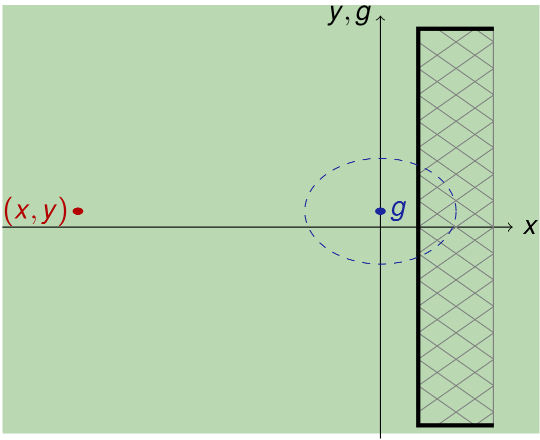
Simultaneous logical language integration is best.



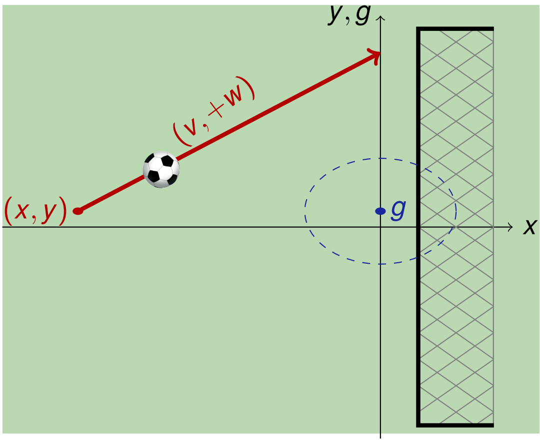
LICS'16

Outline

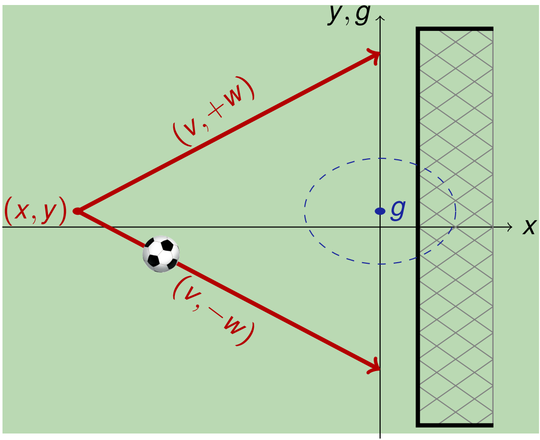
Example: Goalie in Robot Soccer



$$x < 0 \land v > 0 \land y = g \rightarrow$$
$$\big\langle (w := +w \cap w := -w);$$
$$\big((u := +u \cup u := -u); \{x' = v, y' = w, g' = u\}\big)^* \big\rangle\, x^2 + (y-g)^2 \le 1$$

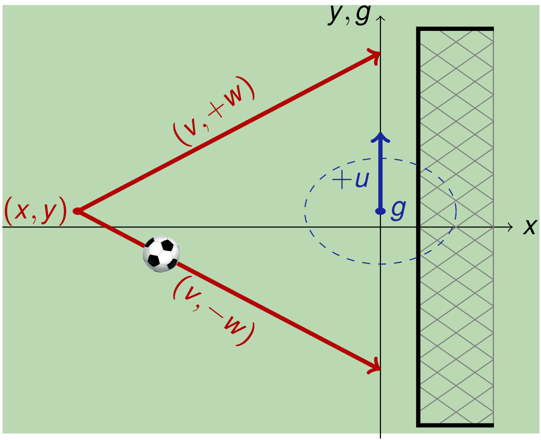


$$x < 0 \land v > 0 \land y = g \rightarrow$$
$$\Big\langle (w := +w \cap w := -w);$$
$$\Big((u := +u \cup u := -u); \{x' = v, y' = w, g' = u\} \Big)^* \Big\rangle\, x^2 + (y-g)^2 \leq 1$$



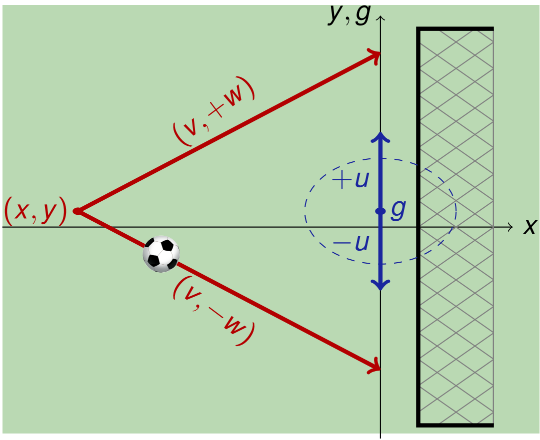
$$x < 0 \land v > 0 \land y = g \rightarrow$$
$$\big\langle (w := +w \cap w := -w);$$
$$\big((u := +u \cup u := -u); \{x' = v, y' = w, g' = u\}\big)^* \big\rangle\, x^2 + (y-g)^2 \le 1$$

Example: Goalie in Robot Soccer



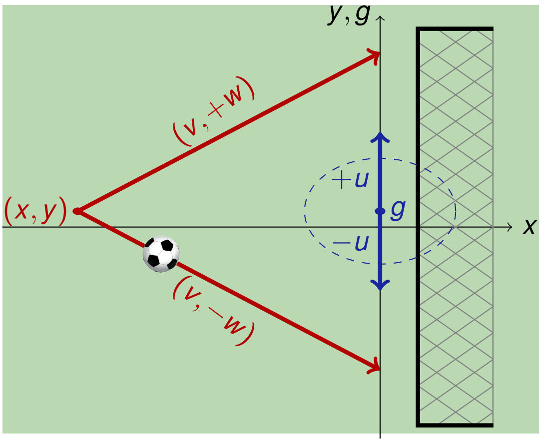
$$x < 0 \land v > 0 \land y = g \rightarrow \\ \langle (w := +w \cap w := -w); \\ \big((u := +u \cup u := -u); \{x' = v, y' = w, g' = u\}\big)^* \rangle\, x^2 + (y-g)^2 \le 1$$

Example: Goalie in Robot Soccer



$$x < 0 \land v > 0 \land y = g \rightarrow$$
$$\langle (w := +w \cap w := -w);$$
$$\big((u := +u \cup u := -u); \{x' = v, y' = w, g' = u\}\big)^* \rangle\, x^2 + (y-g)^2 \le 1$$

Example: Goalie in Robot Soccer



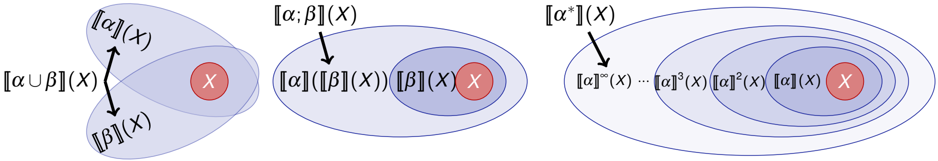
Goalie's Secret

$$\left(\frac{x}{v}\right)^2 (u-w)^2 \leq 1 \land$$
$$x < 0 \land v > 0 \land y = g \rightarrow$$
$$\big\langle (w := +w \cap w := -w);$$
$$\big((u := +u \cup u := -u); \{x' = v, y' = w, g' = u\}\big)^*\big\rangle\, x^2 + (y-g)^2 \leq 1$$

Differential Game Logic dGL

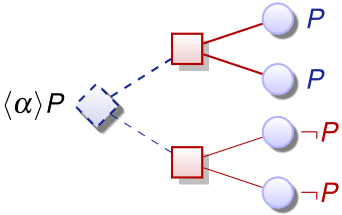
Definition (Hybrid game)

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



Definition (Differential game logic) (TOCL'15)

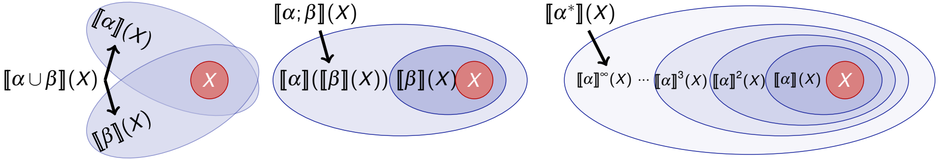
$$P, Q ::= e \geq \tilde{e} \mid \neg P \mid P \wedge Q \mid P \vee Q \mid P \rightarrow Q \mid \forall x\, P \mid \exists x\, P \mid [\alpha]P \mid \langle\alpha\rangle P$$



Differential Game Logic dGL

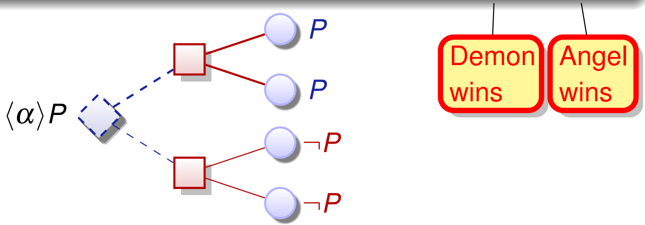
Definition (Hybrid game)

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



Definition (Differential game logic) (TOCL'15)

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



Definition (Hybrid game α) $[\![\cdot]\!] : \mathrm{HG} \to (\wp(\mathscr{S}) \to \wp(\mathscr{S}))$

$$
\begin{aligned}
\varsigma_{x:=e}(X) &= \{\omega \in \mathscr{S} : \omega_x^{\omega[\![e]\!]} \in X\} \\
\varsigma_{x'=f(x)}(X) &= \{\varphi(0) \in \mathscr{S} : \varphi(r) \in X, \frac{\mathrm{d}\,\varphi(t)(x)}{\mathrm{d}t}(\zeta) = \varphi(\zeta)[\![f(x)]\!] \text{ for all } \zeta\} \\
\varsigma_{?Q}(X) &= [\![Q]\!] \cap X \\
\varsigma_{\alpha\cup\beta}(X) &= \varsigma_\alpha(X) \cup \varsigma_\beta(X) \\
\varsigma_{\alpha;\beta}(X) &= \varsigma_\alpha(\varsigma_\beta(X)) \\
\varsigma_{\alpha^*}(X) &= \bigcap\{Z \subseteq \mathscr{S} : X \cup \varsigma_\alpha(Z) \subseteq Z\} \\
\varsigma_{\alpha^d}(X) &= (\varsigma_\alpha(X^\complement))^\complement
\end{aligned}
$$

Definition (dGL Formula P) $[\![\cdot]\!] : \mathrm{Fml} \to \wp(\mathscr{S})$

$$
\begin{aligned}
[\![e \geq \tilde{e}]\!] &= \{\omega \in \mathscr{S} : \omega[\![e]\!] \geq \omega[\![\tilde{e}]\!]\} \\
[\![\neg P]\!] &= ([\![P]\!])^\complement \\
[\![P \wedge Q]\!] &= [\![P]\!] \cap [\![Q]\!] \\
[\![\langle\alpha\rangle P]\!] &= \varsigma_\alpha([\![P]\!]) \\
[\![[\alpha]P]\!] &= \delta_\alpha([\![P]\!])
\end{aligned}
$$

compositional semantics

Differential Game Logic: Axiomatization

$[\cdot]$ $[\alpha]P \leftrightarrow \neg\langle\alpha\rangle\neg P$

$\langle:=\rangle$ $\langle x:=e\rangle p(x) \leftrightarrow p(e)$

$\langle'\rangle$ $\langle x' = f(x)\rangle P \leftrightarrow \exists t{\geq}0\, \langle x:=y(t)\rangle P$

$\langle?\rangle$ $\langle ?Q\rangle P \leftrightarrow (Q \land P)$

$\langle\cup\rangle$ $\langle\alpha\cup\beta\rangle P \leftrightarrow \langle\alpha\rangle P \lor \langle\beta\rangle P$

$\langle;\rangle$ $\langle\alpha;\beta\rangle P \leftrightarrow \langle\alpha\rangle\langle\beta\rangle P$

$\langle^*\rangle$ $\langle\alpha^*\rangle P \leftrightarrow P \lor \langle\alpha\rangle\langle\alpha^*\rangle P$

$\langle^d\rangle$ $\langle\alpha^d\rangle P \leftrightarrow \neg\langle\alpha\rangle\neg P$

M $\dfrac{P \rightarrow Q}{\langle\alpha\rangle P \rightarrow \langle\alpha\rangle Q}$

FP $\dfrac{P \lor \langle\alpha\rangle Q \rightarrow Q}{\langle\alpha^*\rangle P \rightarrow Q}$

MP $\dfrac{P \quad P \rightarrow Q}{Q}$

$\forall$ $\dfrac{p \rightarrow Q}{p \rightarrow \forall x\, Q}$ $\quad (x \notin \mathrm{FV}(p))$

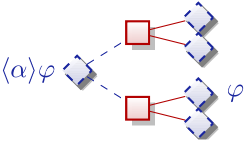
US $\dfrac{\varphi}{\varphi_{p(\cdot)}^{\psi(\cdot)}}$

TOCL'15

Takeaway: Hybrid Games

Differential game logic

- True adversarial competition
- Analytic competition: different agents reach decisions independently
- Cause: misunderstandings, interference, disturbance, different goals
- More general semantics, tame axiomatics
- Compositional verification
- Small-core complete axiomatization in KeYmaera X theorem prover
- Differential game invariants for differential hybrid games
- Almost everything is characterizable via hybrid games
- Arbitrarily nested inductive / coinductive concepts over augmented $\mathbb{R}$



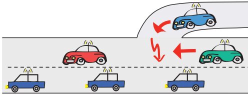
TOCL'15,TOCL'17

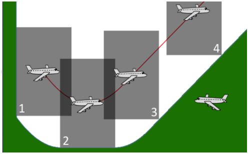
Outline

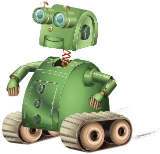
Applications with Impact

Prospects: Safety & Efficiency

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



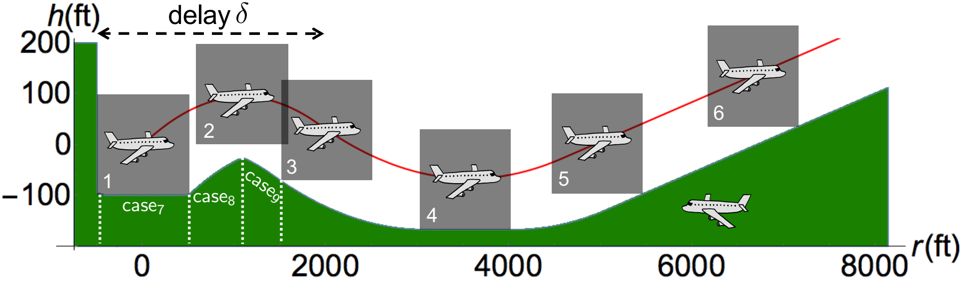




Cyber-Physical Systems

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

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

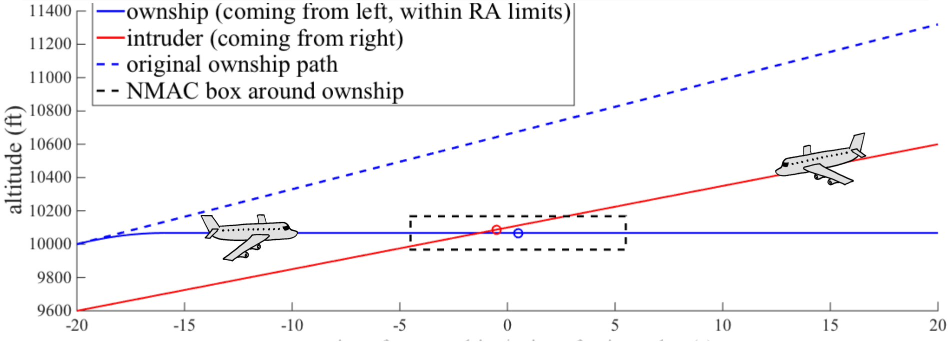


1. Identified safe region for each advisory symbolically
2. Proved safety for hybrid systems flight model in KeYmaera X

STTT'17

Airborne Collision Avoidance System ACAS X: Compare

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

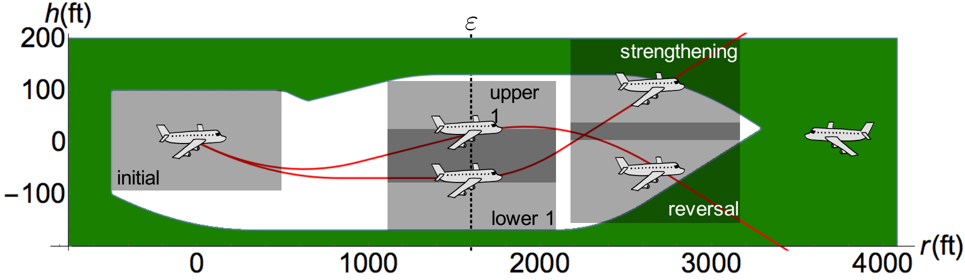


ACAS X issues DNC advisory, which induces collision unless corrected

STTT'17, TECS'22

Airborne Collision Avoidance System ACAS X: Refine

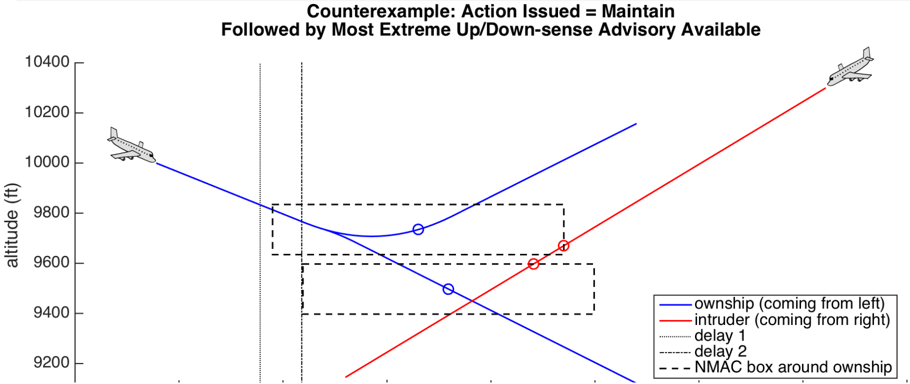
- Conservative, so too many counterexamples
- Settle for: safe for a little while, with safe future advisory possibility
- Safeable advisory: a subsequent advisory can safely avoid collision



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

STTT'17, TECS'22

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

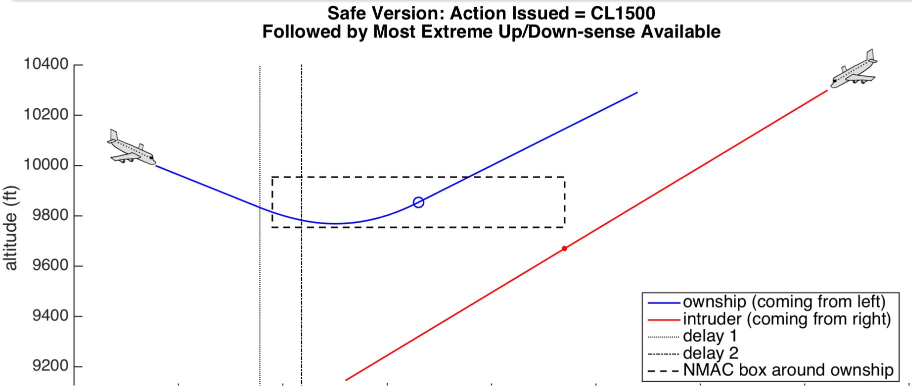


ACAS X issues Maintain advisory instead of CL1500

STTT'17, TECS'22

Airborne Collision Avoidance System ACAS X: Compare

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

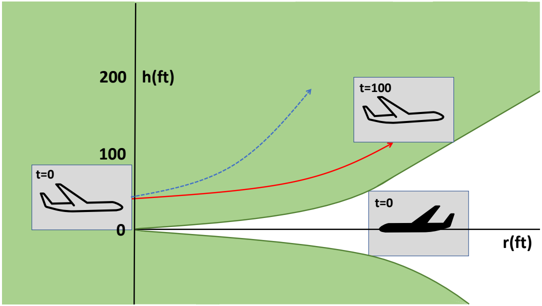


ACAS X issues Maintain advisory instead of CL1500

STTT'17, TECS'22

Airborne Collision Avoidance Games in ACAS X

- Ownship and intruder aircraft both maneuver
- Intruder aircraft chooses actions independently
- ACAS X is a hybrid game

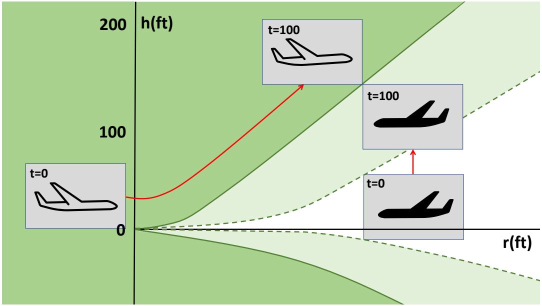


1. Identified safe region for each advisory symbolically
2. Proved safety for hybrid **games** flight model in KeYmaera X

TECS'22

Airborne Collision Avoidance Games in ACAS X

- Ownship and intruder aircraft both maneuver
- Intruder aircraft chooses actions independently
- ACAS X is a hybrid game

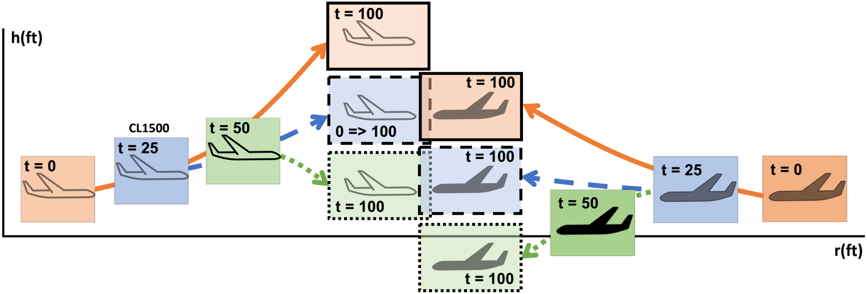


1. Identified safe region for each advisory symbolically
2. Proved safety for hybrid **games** flight model in KeYmaera X

TECS'22

Airborne Collision Avoidance Games in ACAS X

- Ownship and intruder aircraft both maneuver
- Intruder aircraft chooses actions independently
- ACAS X is a hybrid game

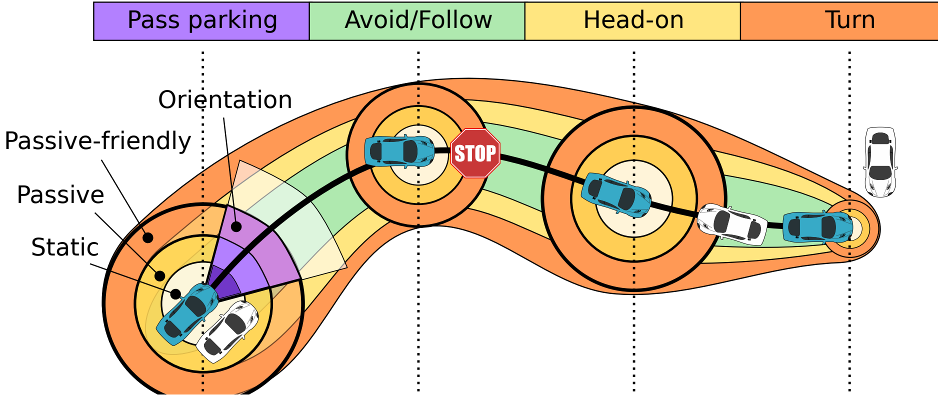


1. Identified safe region for each advisory symbolically
2. Proved safety for hybrid **games** flight model in KeYmaera X

TECS'22

Ground Robot Obstacle Avoidance: Verify

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



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

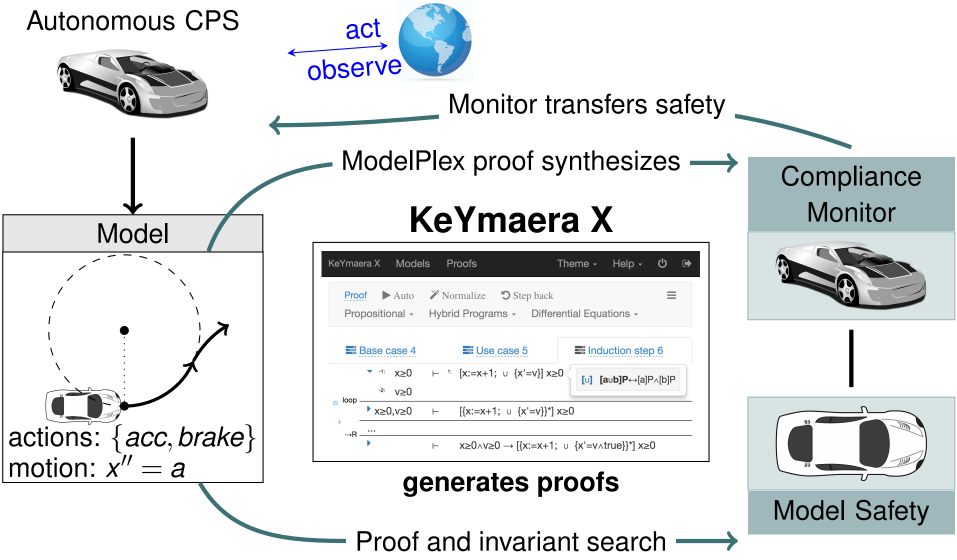
Ground Robot Invariants and Safe Control Constraints

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

⋮

IJRR'17

Approach: Safety Proofs for Autonomous CPS

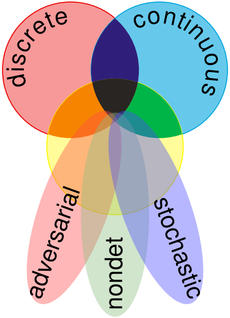


Outline

Further Dynamical Systems Challenges

CPSs deserve proofs as safety evidence!

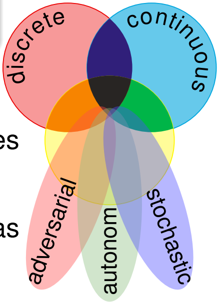
- Verified CPS implementations by ModelPlex — FMSD'16
- Correct CPS execution — PLDI'18
- CPS proof and tactic languages+libraries — ITP'17
- Big CPS built from safe components — STTT'18
- ODE invariance — JACM'20
- ODE liveness — FAC'21
- ODE stability — TACAS'21
- Invariant generation — FMSD'21
- Safe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

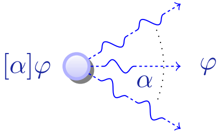


Refinements of Hybrid Dynamical Systems Logic

differential dynamic logic

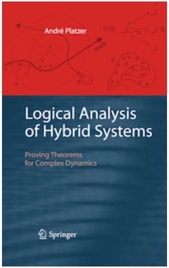
dL = DL + HP



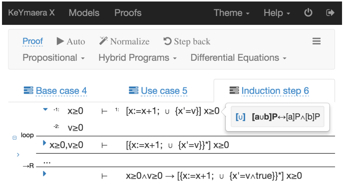


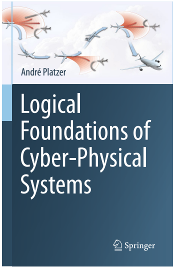
- Strong analytic foundations
- Practical reasoning advances
- Significant applications
- Catalyze many science areas

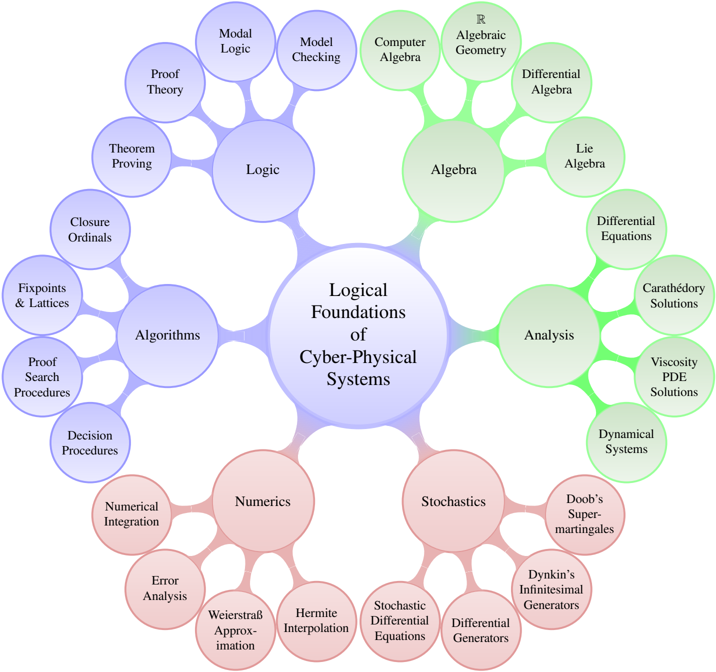
- Logic & Proofs for CPS
- Programming languages
- Theorem proving
- Multi-dynamical systems

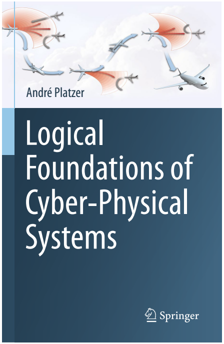


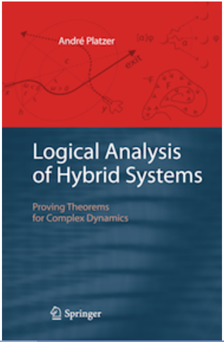
KeYmaera X



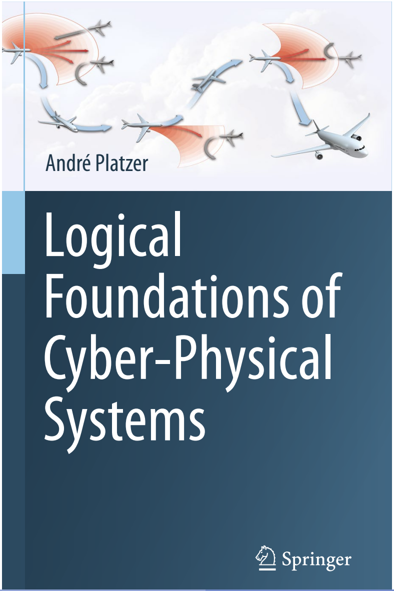



Logical Foundations of Cyber-Physical Systems
Logic
Modal Logic
Model Checking
Proof Theory
Theorem Proving
Algebra
Computer Algebra
ℝ Algebraic Geometry
Differential Algebra
Lie Algebra
Analysis
Differential Equations
Carathéodory Solutions
Viscosity PDE Solutions
Dynamical Systems
Stochastics
Doob's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


André Platzer
Logical Foundations of Cyber-Physical Systems
Springer


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

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



Outline

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

9 Appendix

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

Complete Proof Theory of Hybrid Systems

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

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

Corollary (Complete Proof-theoretical Bridge)

proving continuous = proving hybrid = proving discrete

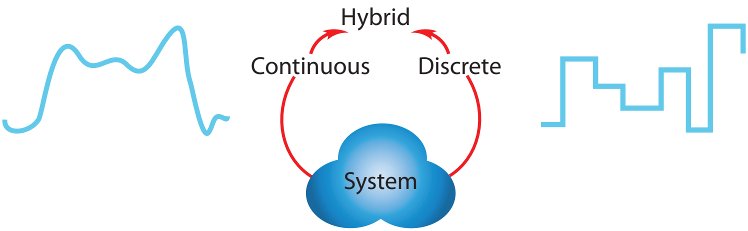
$$\vDash P \text{ iff } \mathrm{FOD} \vdash_{\mathrm{dL}} P$$

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

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

Corollary (Complete Proof-theoretical Bridge)

proving continuous = proving hybrid = proving discrete



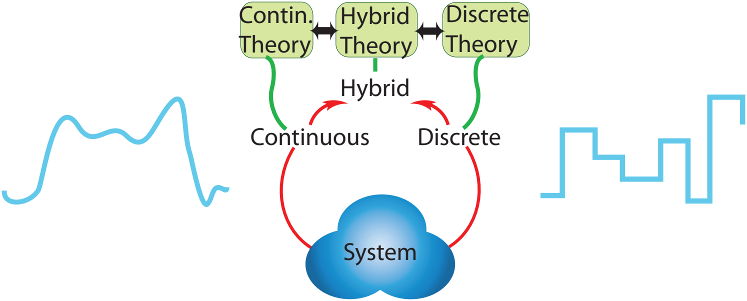
Complete Proof Theory of Hybrid Systems

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

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

Corollary (Complete Proof-theoretical Bridge)

proving continuous = proving hybrid = proving discrete



Uniform Substitution

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

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

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

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

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

JAR'17

Uniform Substitution

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

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

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

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

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

JAR'17

Uniform Substitution

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

Modular interface: Prover vs. Logic

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

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

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

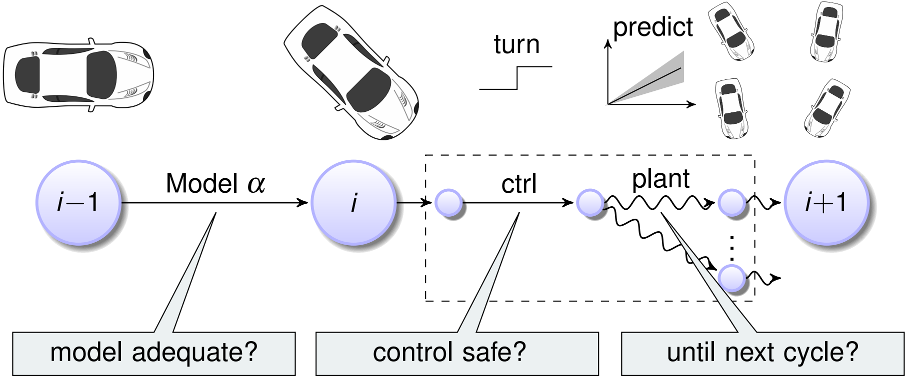
If you bind a free variable, you go to logic jail!

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

Clash

JAR'17

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



RV'14,FMSD'16

ModelPlex Runtime Model Validation

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

Insights

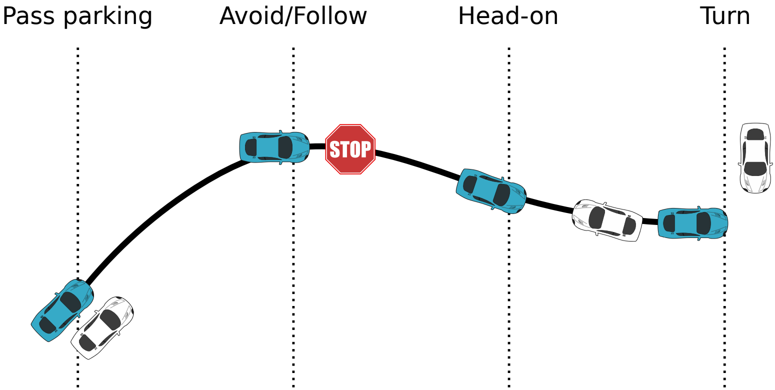
- Verification results about models transfer to the CPS when validating model compliance.
- Compliance with model is characterizable in logic dL.
- Compliance formula transformed by dL proof to monitor.
- Correct-by-construction provably correct model validation at runtime.

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

RV'14,FMSD'16

Ground Robot Obstacle Avoidance: Verify

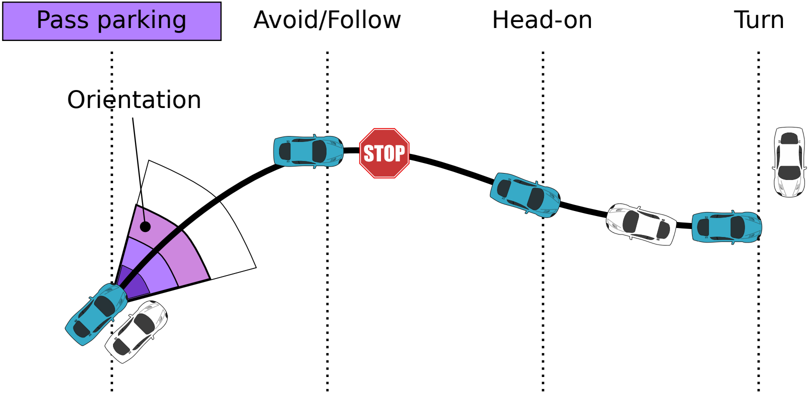
- Fundamental safety question for ground robot navigation IJRR'17
- When will which control decision avoid obstacles?
- Depends on safety objective, physical capabilities of robot + obstacle



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

Ground Robot Obstacle Avoidance: Verify

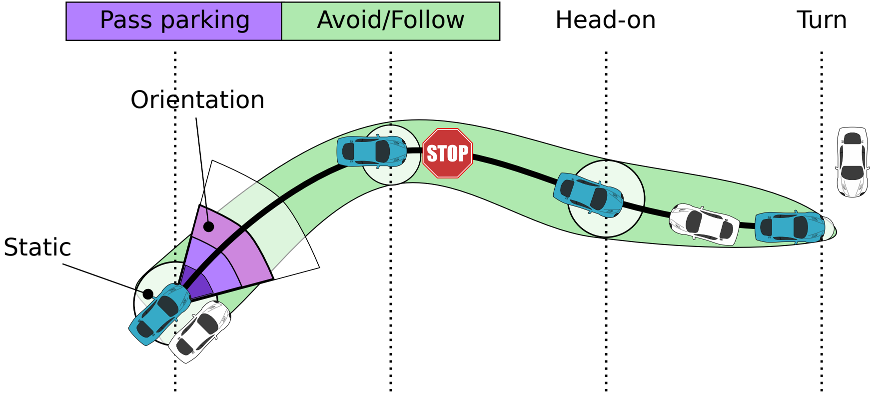
- Fundamental safety question for ground robot navigation — IJRR'17
- When will which control decision avoid obstacles?
- Depends on safety objective, physical capabilities of robot + obstacle



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

Ground Robot Obstacle Avoidance: Verify

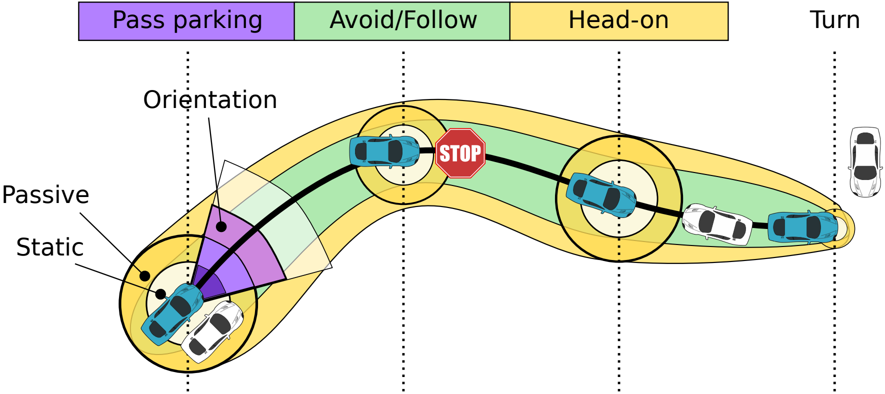
- Fundamental safety question for ground robot navigation IJRR'17
- When will which control decision avoid obstacles?
- Depends on safety objective, physical capabilities of robot + obstacle



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

Ground Robot Obstacle Avoidance: Verify

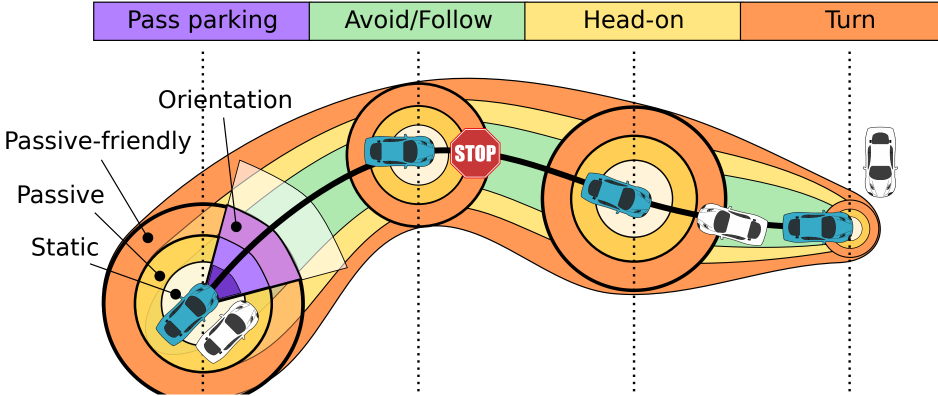
- Fundamental safety question for ground robot navigation — IJRR'17
- When will which control decision avoid obstacles?
- Depends on safety objective, physical capabilities of robot + obstacle



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

Ground Robot Obstacle Avoidance: Verify

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



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

Ground Robot Invariants and Safe Control Constraints

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

⋮

IJRR'17

Ground Robot Invariants and Safe Control Constraints

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

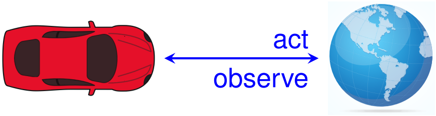
⋮

Question

How to find and justify constraints? Proof!

IJRR'17

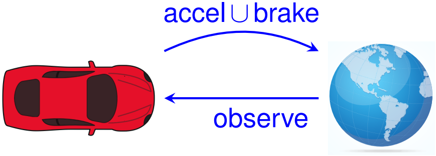
Learning to Act in a CPS



Reinforcement Learning learns from experience of trying actions

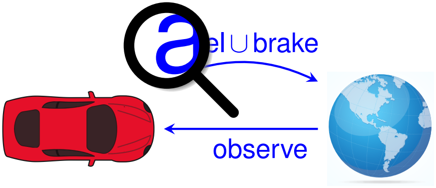
AAAI'18,ITC'18,TACAS'19,QEST'19

Learning to Act in a CPS



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

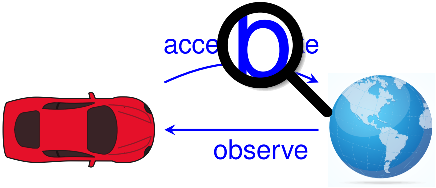
AAAI'18,ITC'18,TACAS'19,QEST'19



ModelPlex monitor inspects each decision, vetoes if unsafe

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

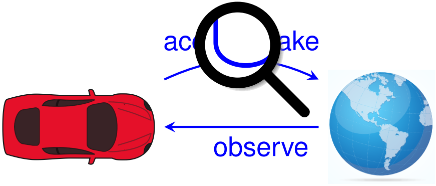
Learning to Act Safely in a CPS



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

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

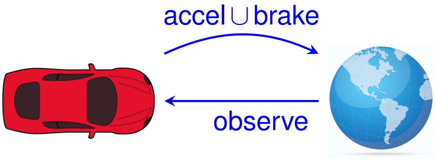
Learning to Act Safely in a CPS



dL benefits from RL optimization.

RL benefits from dL safety signal.

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

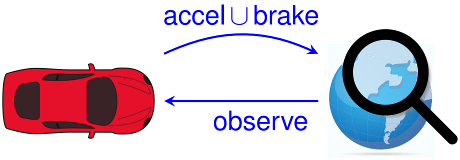


Theorem: Safe policy if ODE accurate

Experiment: Graceful recovery outside ODE ⇜ quantitative ModelPlex

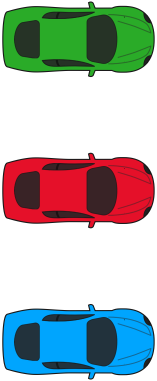
Detect modeled versus unmodeled state space ⇜ ModelPlex

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



What’s safe when off model?

AAAI’18,ITC’18,TACAS’19,QEST’19



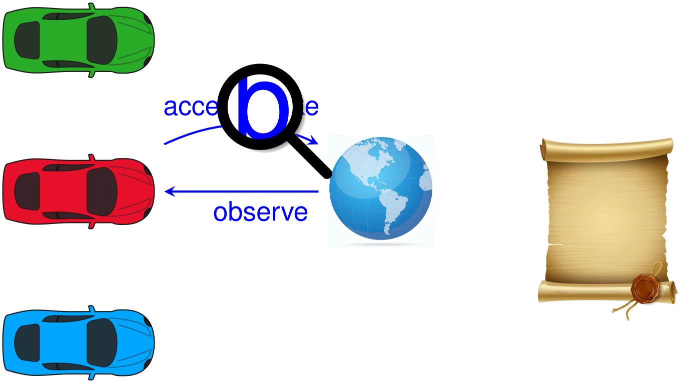
accel $\cup$ brake

observe

What's safe with multiple possible models?

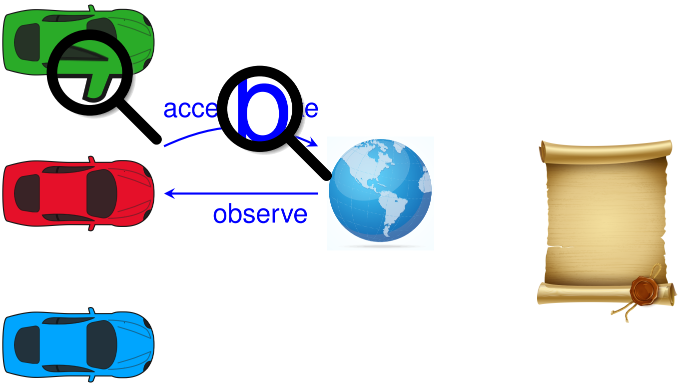
AAAI'18,ITC'18,TACAS'19,QEST'19

Learning to Act Safely in a CPS with Multiple Models



ModelPlex monitors conjunction of all plausible models

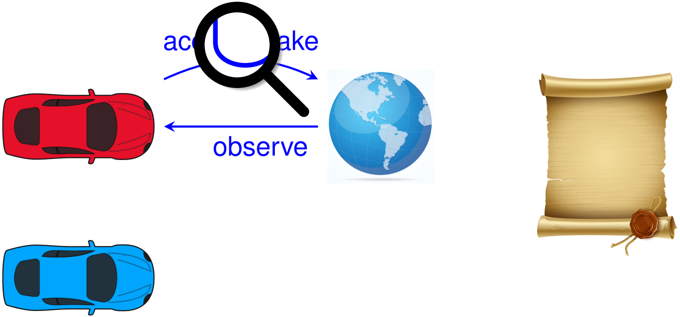
AAAI'18,ITC'18,TACAS'19,QEST'19



Remove incompatible models after contradictory observation

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

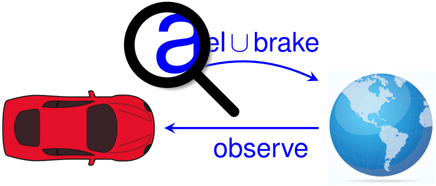
Learning to Act Safely in a CPS with Multiple Models



Plan differentiating experiment $\leftsquigarrow$ predictive monitor distinctions

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

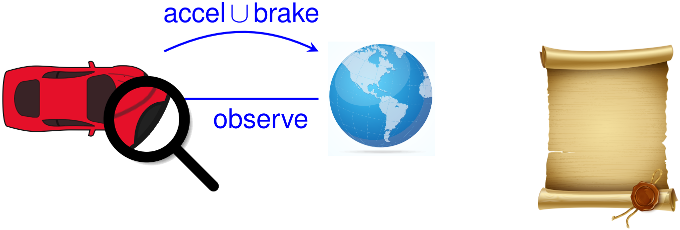
Learning to Act Safely in a CPS with Multiple Models



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

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

Learning to Act Safely in a CPS with Multiple Models



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

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

André Platzer.
Logics of dynamical systems.
In LICS [23], pages 13–24.
doi:10.1109/LICS.2012.13.

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

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

André Platzer.
The complete proof theory of hybrid systems.
In LICS [23], pages 541–550.
doi:10.1109/LICS.2012.64.

André Platzer and Yong Kiam Tan.

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

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

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

André Platzer.
A complete axiomatization of quantified differential dynamic logic for distributed hybrid systems.
Log. Meth. Comput. Sci., 8(4:17):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 446–460, Berlin, 2011. Springer.
doi:10.1007/978-3-642-22438-6_34.

André Platzer.
A uniform substitution calculus for differential dynamic logic.
In Amy Felty and Aart Middeldorp, editors, *CADE*, volume 9195 of *LNCS*, pages 467–481, Berlin, 2015. Springer.
doi:10.1007/978-3-319-21401-6_32.

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.

Yong Kiam Tan and André Platzer.
An axiomatic approach to existence and liveness for differential equations.
Formal Aspects Comput., 33(4):461–518, 2021.
doi:10.1007/s00165-020-00525-0.

Yong Kiam Tan and André Platzer.
Deductive stability proofs for ordinary differential equations.
In Jan Friso Groote and Kim Guldstrand Larsen, editors, *Tools and Algorithms for the Construction and Analysis of Systems - 27th International Conference, TACAS 2021, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2021, Proceedings, Part II*, volume 12652 of *LNCS*, pages 181–199. Springer, 2021.
doi:10.1007/978-3-030-72013-1_10.

Sarah M. Loos and André Platzer.
Differential refinement logic.
In Martin Grohe, Eric Koskinen, and Natarajan Shankar, editors, *LICS*, pages 505–514, New York, 2016. ACM.

doi:10.1145/2933575.2934555.

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

Stefan Mitsch, Khalil Ghorbal, David Vogelbacher, and André Platzer.
Formal verification of obstacle avoidance and navigation of ground robots.
I. J. Robotics Res., 36(12):1312–1340, 2017.
doi:10.1177/0278364917733549.

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

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

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

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

Nathan Fulton and André Platzer.
Verifiably safe off-model reinforcement learning.

In Tomas Vojnar and Lijun Zhang, editors, *TACAS, Part I*, volume 11427 of *LNCS*, pages 413–430. Springer, 2019.
doi:10.1007/978-3-030-17462-0_28.

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