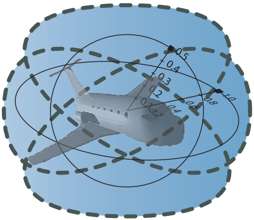
Quantified Differential Invariants

André Platzer

Carnegie Mellon University, Pittsburgh, PA



Outline

1. Motivation
2. Quantified Differential Dynamic Logic Qd$\mathcal{L}$
   - Design
   - Syntax
   - Semantics
3. Proof Calculus for Distributed Hybrid Systems
   - Compositional Verification Calculus
   - Air Traffic Control
   - Derivations and Differentiation
   - Soundness and Completeness
4. Conclusions

Complex Physical Systems:

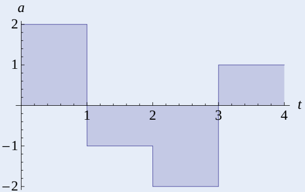
Q: Verify my plane?

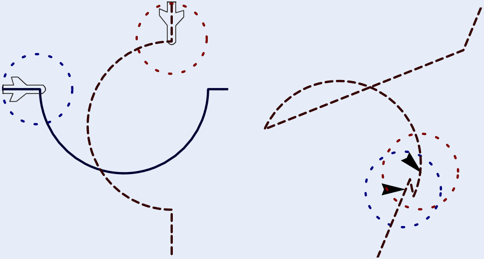
Challenge

Q: Verify my plane? A: Hybrid systems

Challenge (Hybrid Systems)

- Continuous dynamics (differential equations)
- Discrete dynamics (control decisions)

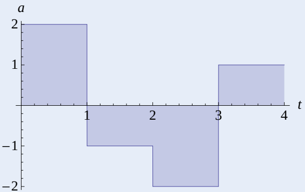


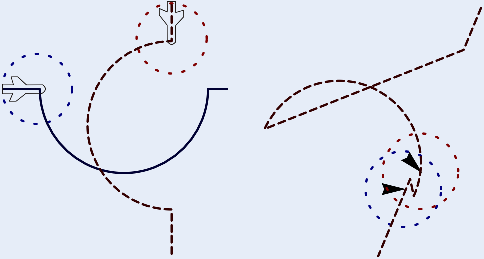


Q: Verify my plane? A: Hybrid systems Q: But there's lots of planes!

Challenge (Hybrid Systems)

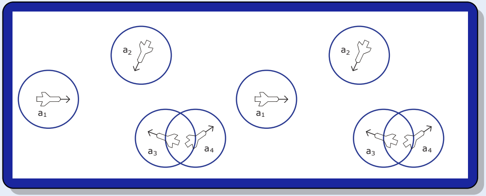
- Continuous dynamics (differential equations)
- Discrete dynamics (control decisions)





Q: Verify lots of planes?

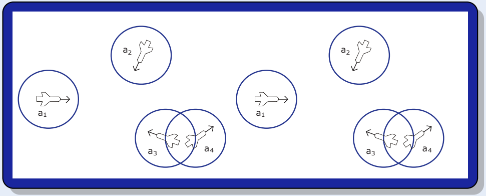
Challenge

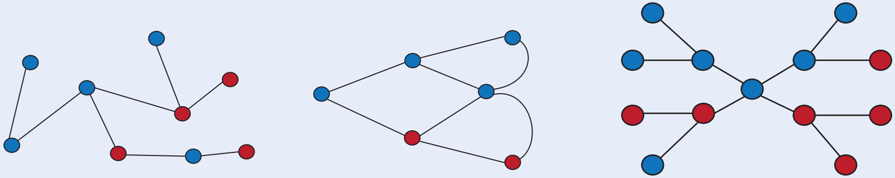


Q: Verify lots of planes? A: Distributed systems

Challenge (Distributed Systems)

- Local computation (finite state automaton)
- Remote communication (network graph)



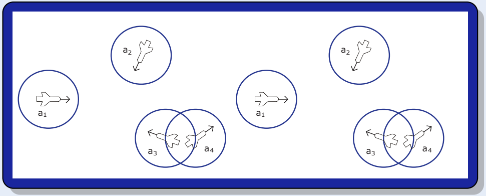


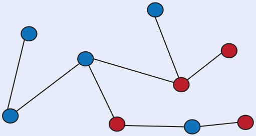
Complex Physical Systems: Distributed Systems

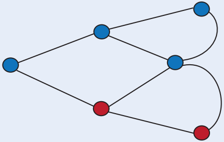
Q: Verify lots of planes? A: Distributed systems Q: But they move!

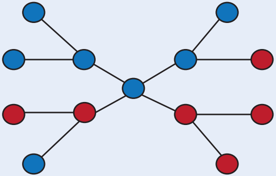
Challenge (Distributed Systems)

- Local computation
  (finite state automaton)
- Remote communication
  (network graph)



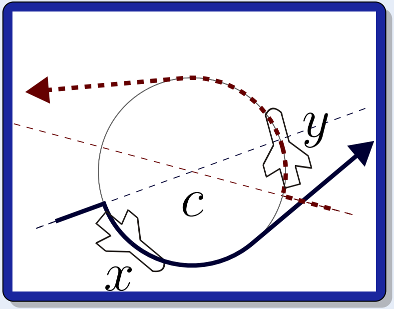






Q: Verify lots of moving planes?

Challenge

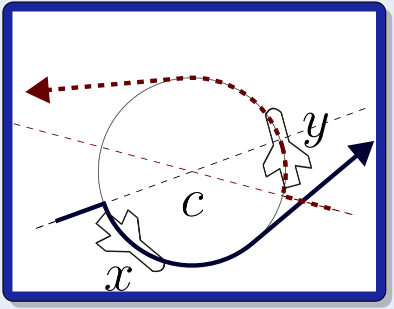


Complex Physical Systems: Distributed Hybrid Systems

Q: Verify lots of moving planes? A: Distributed hybrid systems

Challenge (Distributed Hybrid Systems)

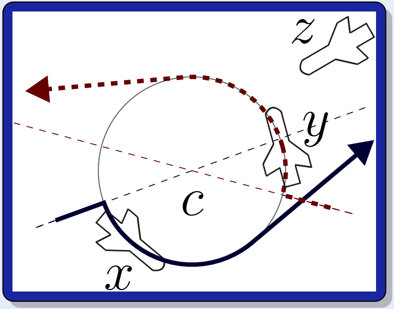
- Continuous dynamics (differential equations)
- Discrete dynamics (control decisions)
- Structural dynamics (remote communication)



Q: Verify lots of moving planes? A: Distributed hybrid systems

Challenge (Distributed Hybrid Systems)

- Continuous dynamics (differential equations)
- Discrete dynamics (control decisions)
- Structural dynamics (remote communication)
- Dimensional dynamics (appearance)

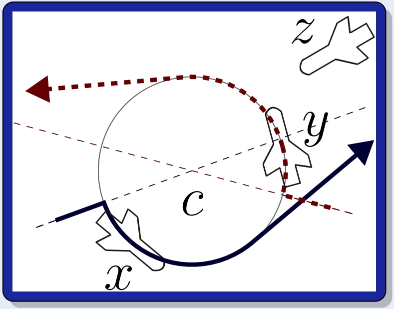


Complex Physical Systems: Distributed Hybrid Systems

Q: Verify lots of moving planes? A: Distributed hybrid systems Q: How?

Challenge (Distributed Hybrid Systems)

- Continuous dynamics (differential equations)
- Discrete dynamics (control decisions)
- Structural dynamics (remote communication)
- Dimensional dynamics (appearance)



State of the Art:

Shift [DGV96] The Hybrid System Simulation Programming Language

Hybrid CSP [CJR95] Semantics in Extended Duration Calculus

HyPA [CR05] Translate fragment into normal form.

χ process algebra [vBMR$^+$06] Simulation, translation of fragments to PHAVER, UPPAAL

R-Charon [KSPL06] Modeling Language for Reconfigurable Hybrid Systems

Φ-calculus [Rou04] Semantics in rich set theory

ACP_{hs}^{srt} [BM05] Modeling language proposal

OBSHS [MS06] Partial random simulation of objects

State of the Art: Modeling and Simulation

Shift [DGV96] The Hybrid System Simulation Programming Language

R-Charon [KSPL06] Modeling Language for Reconfigurable Hybrid Systems

Hybrid CSP [CJR95] Semantics in Extended Duration Calculus

Φ-calculus [Rou04] Semantics in rich set theory

HyPA [CR05] Translate fragment into normal form.

ACP^{srt}_{hs} [BM05] Modeling language proposal

χ process algebra [vBMR$^+$06] Simulation, translation of fragments to PHAVER, UPPAAL

OBSHS [MS06] Partial random simulation of objects

State of the Art: Modeling and Simulation

No formal verification of distributed hybrid systems

Shift [DGV96] The Hybrid System Simulation Programming Language

Hybrid CSP [CJR95] Semantics in Extended Duration Calculus

HyPA [CR05] Translate fragment into normal form.

χ process algebra [vBMR+06] Simulation, translation of fragments to PHAVER, UPPAAL

R-Charon [KSPL06] Modeling Language for Reconfigurable Hybrid Systems

Φ-calculus [Rou04] Semantics in rich set theory

ACP^{srt}_{hs} [BM05] Modeling language proposal

OBSHS [MS06] Partial random simulation of objects

Outline

1. Motivation
2. Quantified Differential Dynamic Logic Q$d\mathcal{L}$
   - Design
   - Syntax
   - Semantics
3. Proof Calculus for Distributed Hybrid Systems
   - Compositional Verification Calculus
   - Air Traffic Control
   - Derivations and Differentiation
   - Soundness and Completeness
4. Conclusions

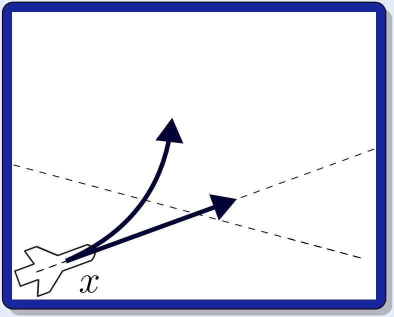
Outline (Conceptual Approach)

1. Motivation
2. Quantified Differential Dynamic Logic Qd$\mathcal{L}$
   - Design
   - Syntax
   - Semantics
3. Proof Calculus for Distributed Hybrid Systems
   - Compositional Verification Calculus
   - Air Traffic Control
   - Derivations and Differentiation
   - Soundness and Completeness
4. Conclusions

Model for Distributed Hybrid Systems

Q: How to model distributed hybrid systems

Model (Distributed Hybrid Systems)

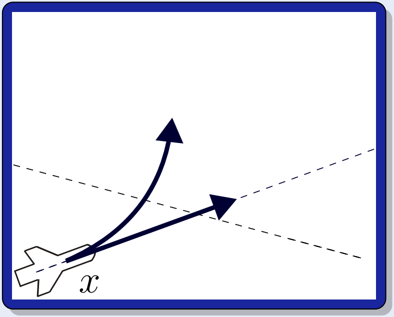


Model for Distributed Hybrid Systems

Q: How to model distributed hybrid systems

Model (Distributed Hybrid Systems)

- Continuous dynamics (differential equations)
- Discrete dynamics (control decisions)
- Structural dynamics (communication/coupling)

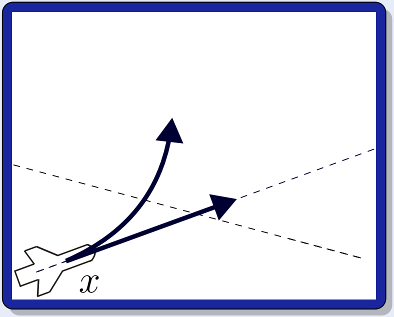


Model for Distributed Hybrid Systems

Q: How to model distributed hybrid systems

Model (Distributed Hybrid Systems)

- Continuous dynamics (differential equations)
  $x' = d, d' = f(\omega, d)$
- Discrete dynamics (control decisions)
- Structural dynamics (communication/coupling)

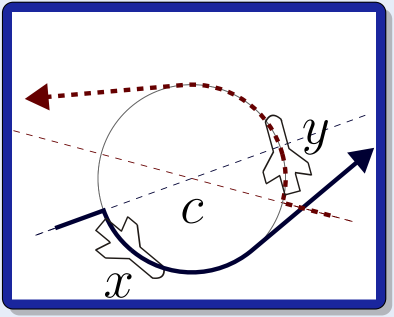


Model for Distributed Hybrid Systems

Q: How to model distributed hybrid systems

Model (Distributed Hybrid Systems)

- Continuous dynamics (differential equations)
  $$x' = d, d' = f(\omega, d)$$
- Discrete dynamics (control decisions)
  $$\omega := \texttt{if} .. \texttt{then}\, 0\, \texttt{else}\, 2$$
- Structural dynamics (communication/coupling)

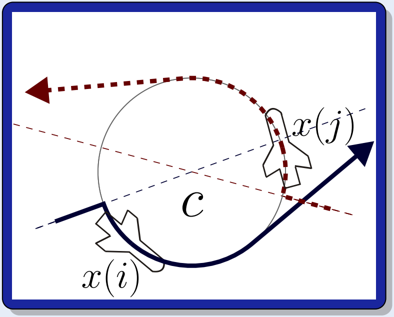


Model for Distributed Hybrid Systems

Q: How to model distributed hybrid systems

Model (Distributed Hybrid Systems)

- Continuous dynamics (differential equations)
  $$x' = d, d' = f(\omega, d)$$
- Discrete dynamics (control decisions)
  $$\omega := \texttt{if} \mathinner{..} \texttt{then}\, 0 \,\texttt{else}\, 2$$
- Structural dynamics (communication/coupling)



Model for Distributed Hybrid Systems

Q: How to model distributed hybrid systems

Model (Distributed Hybrid Systems)

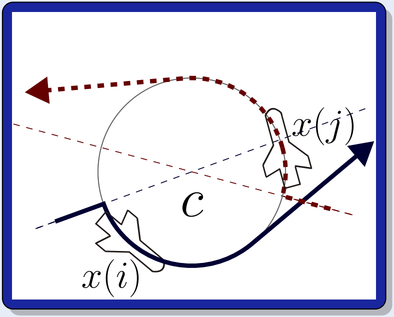
- Continuous dynamics (differential equations)

  $$x' = d, d' = f(\omega, d)$$

- Discrete dynamics (control decisions)

  $$\omega := \texttt{if} .. \texttt{then}\, 0\, \texttt{else}\, 2$$

- Structural dynamics (communication/coupling)

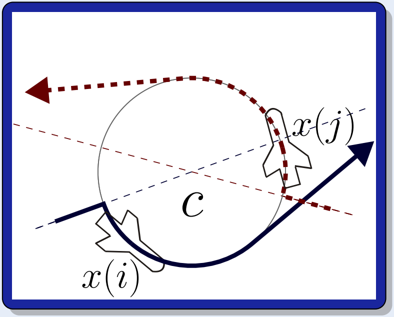


Model for Distributed Hybrid Systems

Q: How to model distributed hybrid systems

Model (Distributed Hybrid Systems)

- Continuous dynamics (differential equations)
  $x(i)' = d(i), d(i)' = f(\omega(i), d(i))$
- Discrete dynamics (control decisions)
  $\omega(i) := \texttt{if} .. \texttt{then}\, 0\, \texttt{else}\, 2$
- Structural dynamics (communication/coupling)



Model for Distributed Hybrid Systems

Q: How to model distributed hybrid systems

Model (Distributed Hybrid Systems)

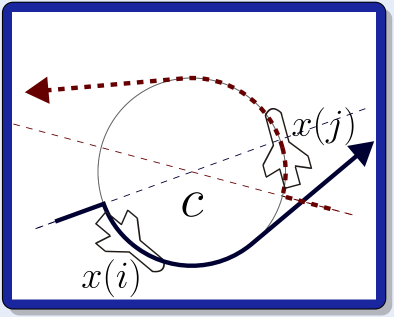
- Continuous dynamics (differential equations)

  $\forall i\, x(i)' = d(i), d(i)' = f(\omega(i), d(i))$

- Discrete dynamics (control decisions)

  $\forall i\, \omega(i) := \texttt{if} .. \texttt{then}\, 0\, \texttt{else}\, 2$

- Structural dynamics (communication/coupling)

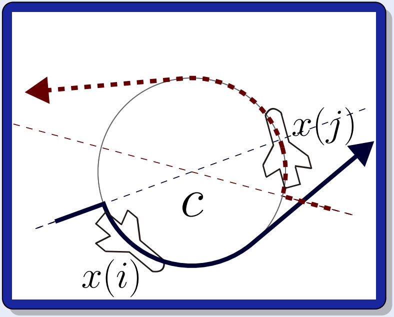


Model for Distributed Hybrid Systems

Q: How to model distributed hybrid systems

Model (Distributed Hybrid Systems)

- Continuous dynamics (differential equations)
  $\forall i\, x(i)' = d(i), d(i)' = f(\omega(i), d(i))$
- Discrete dynamics (control decisions)
  $\forall i\, \omega(i) := \texttt{if} .. \texttt{then}\, 0\, \texttt{else}\, 2$
- Structural dynamics (communication/coupling)
  $c(i) := negotiate(i,j)$

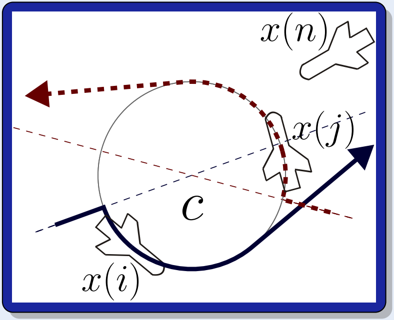


Model for Distributed Hybrid Systems

Q: How to model distributed hybrid systems A: Quantified Hybrid Programs

Model (Distributed Hybrid Systems)

- Continuous dynamics (differential equations)
  $\forall i\, x(i)' = d(i), d(i)' = f(\omega(i), d(i))$
- Discrete dynamics (control decisions)
  $\forall i\, \omega(i) := \texttt{if .. then 0 else 2}$
- Structural dynamics (communication/coupling)
  $c(i) := \mathit{negotiate(i,j)}$
- Dimensional dynamics (appearance)

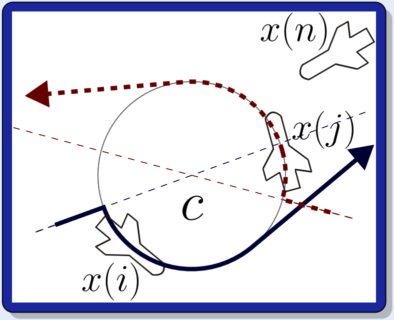


Model for Distributed Hybrid Systems

Q: How to model distributed hybrid systems A: Quantified Hybrid Programs

Model (Distributed Hybrid Systems)

- Continuous dynamics (differential equations)
  $\forall i \, x(i)' = d(i), d(i)' = f(\omega(i), d(i))$
- Discrete dynamics (control decisions)
  $\forall i \, \omega(i) := \texttt{if .. then } 0 \texttt{ else } 2$
- Structural dynamics (communication/coupling)
  $c(i) := negotiate(i,j)$
- Dimensional dynamics (appearance)
  $n := \texttt{new } Aircraft$



Quantified Differential Dynamic Logic Qd$\mathcal{L}$: Syntax

Definition (Quantified hybrid program α)

$\forall i : C\ x(i)' = \theta$	(quantified ODE)	
$\forall i : C\ x(i) := \theta$	(quantified assignment)	jump & test
$?\chi$	(conditional execution)	
$\alpha ; \beta$	(seq. composition)	Kleene algebra
$\alpha \cup \beta$	(nondet. choice)	
α^*	(nondet. repetition)	

Quantified Differential Dynamic Logic Qd$\mathcal{L}$: Syntax

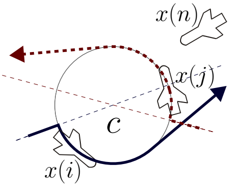
Definition (Quantified hybrid program α)

$\forall i : C\ x(i)' = \theta$	(quantified ODE)	
$\forall i : C\ x(i) := \theta$	(quantified assignment)	jump & test
$?\chi$	(conditional execution)	
$\alpha; \beta$	(seq. composition)	
$\alpha \cup \beta$	(nondet. choice)	Kleene algebra
α^*	(nondet. repetition)	

$$DATC \equiv (ctrl; fly)^*$$

$$ctrl \equiv \forall i : A\ \omega(i) := \texttt{if}\ \forall j : A\ far(i,j)\ \texttt{then}\ 0\ \texttt{else}\ 2$$

$$fly \equiv \forall i : A\ x(i)'' = d(i), d(i)' = f(\omega(i), d(i))$$



Quantified Differential Dynamic Logic $Qd\mathcal{L}$: Syntax

Definition (Quantified hybrid program α)

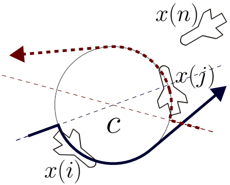
$\forall i : C\ x(i)' = \theta$	(quantified ODE)	
$\forall i : C\ x(i) := \theta$	(quantified assignment)	jump & test
$?\chi$	(conditional execution)	jump & test
$\alpha; \beta$	(seq. composition)	Kleene algebra
$\alpha \cup \beta$	(nondet. choice)	Kleene algebra
α^*	(nondet. repetition)	Kleene algebra

$$DATC \equiv (appear; ctrl; fly)^*$$

$$appear \equiv n := \mathtt{new}\ A;\quad ?(\forall j : A\ far(j, n))$$

$$ctrl \equiv \forall i : A\ \omega(i) := \mathtt{if}\ \forall j : A\ far(i, j)\ \mathtt{then}\ 0\ \mathtt{else}\ 2$$

$$fly \equiv \forall i : A\ x(i)'' = d(i), d(i)' = f(\omega(i), d(i))$$



Quantified Differential Dynamic Logic Qd$\mathcal{L}$: Syntax

Definition (Quantified hybrid program α)

$\forall i : C\ x(i)' = \theta$	(quantified ODE)	
$\forall i : C\ x(i) := \theta$	(quantified assignment)	jump & test
$?\chi$	(conditional execution)	jump & test
$\alpha ; \beta$	(seq. composition)	Kleene algebra
$\alpha \cup \beta$	(nondet. choice)	Kleene algebra
α^*	(nondet. repetition)	Kleene algebra

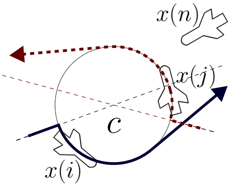
$$DATC \equiv (appear\,;\,ctrl\,;\,fly)^*$$

$$appear \equiv n := \mathtt{new}\,A; \quad ?(\forall j : A\ far(j, n))$$

$$ctrl \equiv \forall i : A\ \omega(i) := \mathtt{if}\ \forall j : A\ far(i, j)\ \mathtt{then}\ 0\ \mathtt{else}\ 2$$

$$fly \equiv \forall i : A\ x(i)'' = d(i), d(i)' = f(\omega(i), d(i))$$

$\mathtt{new}\ A$ is definable!



Quantified Differential Dynamic Logic Qd$\mathcal{L}$: Syntax

Definition (Quantified hybrid program α)

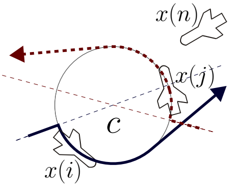
$\forall i : C\ x(i)' = \theta$	(quantified ODE)	
$\forall i : C\ x(i) := \theta$	(quantified assignment)	jump & test
$?\chi$	(conditional execution)	
$\alpha; \beta$	(seq. composition)	
$\alpha \cup \beta$	(nondet. choice)	Kleene algebra
α^*	(nondet. repetition)	

$$DATC \equiv (appear; ctrl; fly)^*$$

$$appear \equiv n := \mathtt{new}\ A; \quad ?(\forall j : A\ far(j, n))$$

$$ctrl \equiv \forall i : A\ \omega(i) := \mathtt{if}\ \forall j : A\ far(i, j)\ \mathtt{then}\ 0\ \mathtt{else}\ 2$$

$$fly \equiv \forall i : A\ x(i)'' = d(i), d(i)' = f(\omega(i), d(i))$$



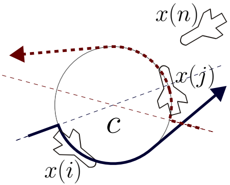
Quantified Differential Dynamic Logic Qd$\mathcal{L}$: Syntax

Definition (Qd$\mathcal{L}$ Formula ϕ)

$\neg, \wedge, \vee, \rightarrow,\ \forall x, \exists x,\ =, \leq,\ +, \cdot$ (ℝ-first-order part)

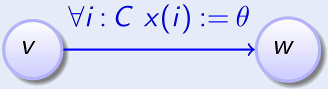
$[\alpha]\phi,\quad \langle\alpha\rangle\phi$ (dynamic part)

$$\forall i, j : A\ far(i,j) \ \rightarrow$$
$$[(appear; ctrl; fly)^*]\ \forall i, j : A\ (i = j \vee (x_1(i) - x_1(j))^2 + (x_2(i) - x_2(j))^2 \geq p^2)$$

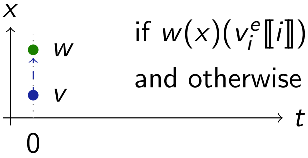


Quantified Differential Dynamic Logic Qd$\mathcal{L}$: Semantics

Definition (Quantified hybrid program α: transition semantics)



Details

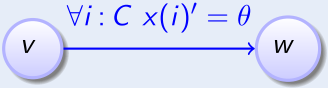


if $w(x)(v_i^e[\![i]\!]) = v_i^e[\![\theta]\!]$ (for all e)

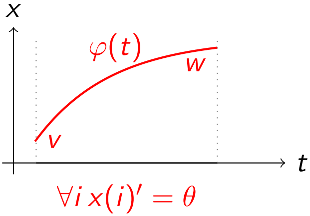
and otherwise unchanged

Quantified Differential Dynamic Logic Qd$\mathcal{L}$: Semantics

Definition (Quantified hybrid program α: transition semantics)



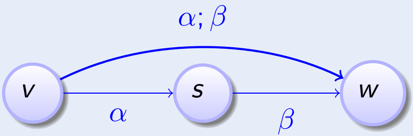
Details



$$\frac{d\,\varphi(t)_i^e[\![x(i)]\!]}{dt}(\zeta) = \varphi(\zeta)_i^e[\![\theta]\!] \quad \text{(for all } e\text{)}$$

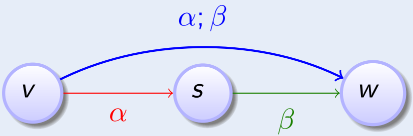
Quantified Differential Dynamic Logic Qd$\mathcal{L}$: Semantics

Definition (Quantified hybrid program α: transition semantics)

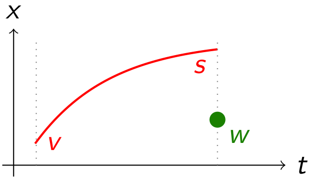


Details

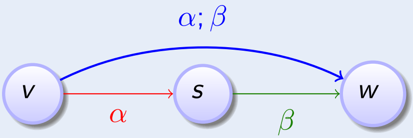
Definition (Quantified hybrid program α: transition semantics)



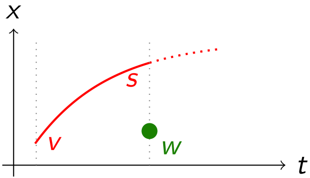
Details



Definition (Quantified hybrid program α: transition semantics)

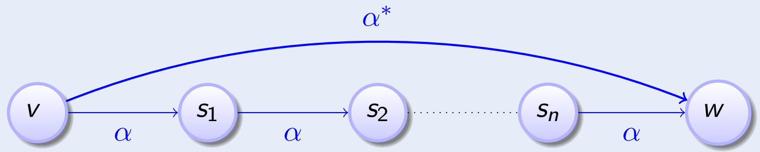


Details



Quantified Differential Dynamic Logic Qd$\mathcal{L}$: Semantics

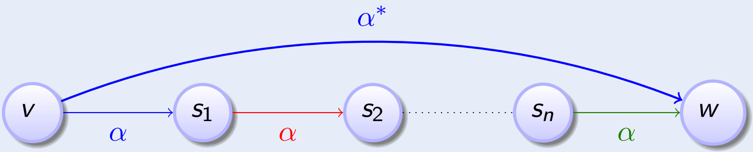
Definition (Quantified hybrid program α: transition semantics)



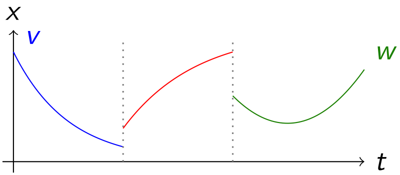
Details

Quantified Differential Dynamic Logic Qd$\mathcal{L}$: Semantics

Definition (Quantified hybrid program α: transition semantics)

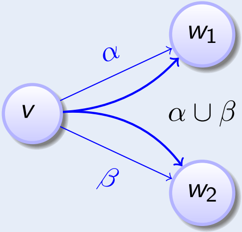


Details



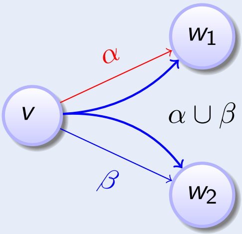
Quantified Differential Dynamic Logic Qd$\mathcal{L}$: Semantics

Definition (Quantified hybrid program α: transition semantics)

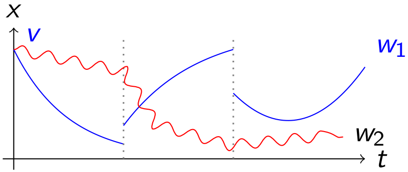


Details

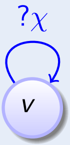
Definition (Quantified hybrid program α: transition semantics)



Details

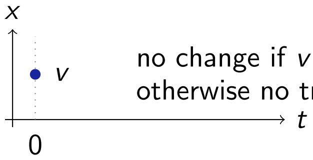


Definition (Quantified hybrid program α: transition semantics)



if $v \models \chi$

Details

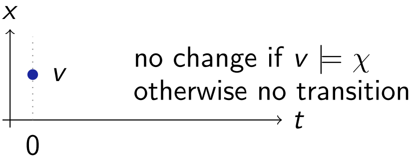


no change if $v \models \chi$
otherwise no transition

Definition (Quantified hybrid program α: transition semantics)

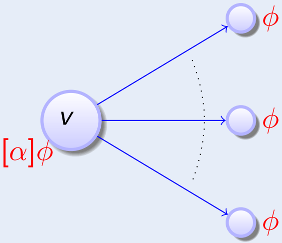
if $v \not\models \chi$

Details



Quantified Differential Dynamic Logic Qd$\mathcal{L}$: Semantics

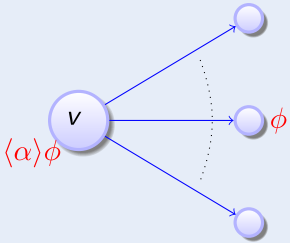
Definition (Qd$\mathcal{L}$ Formula ϕ)



Details

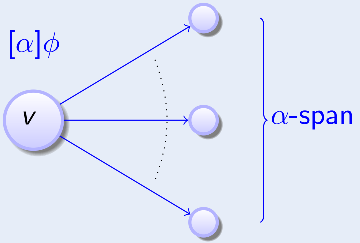
Quantified Differential Dynamic Logic Qd$\mathcal{L}$: Semantics

Definition (Qd$\mathcal{L}$ Formula ϕ)



Details

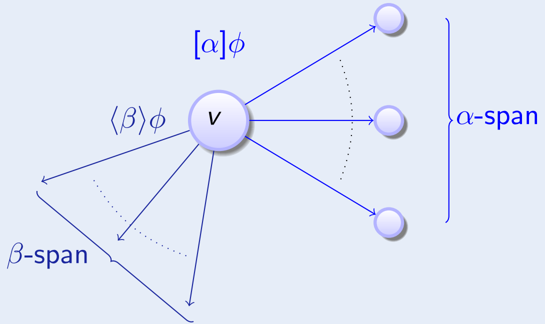
Definition (QdL Formula ϕ)



Details

Quantified Differential Dynamic Logic Qd$\mathcal{L}$: Semantics

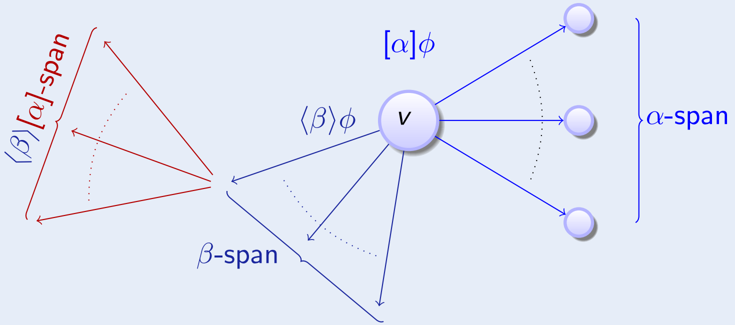
Definition (Qd$\mathcal{L}$ Formula ϕ)



Details

Quantified Differential Dynamic Logic QdL: Semantics

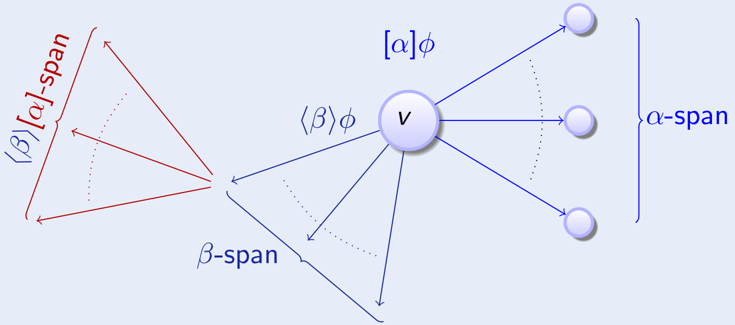
Definition (QdL Formula ϕ)



Details

Quantified Differential Dynamic Logic Qd$\mathcal{L}$: Semantics

Definition (Qd$\mathcal{L}$ Formula ϕ)



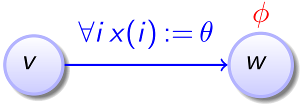
Details

compositional semantics $\Rightarrow$ compositional calculus!

Outline (Verification Approach)

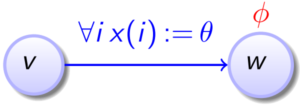
1. Motivation
2. Quantified Differential Dynamic Logic Q$d\mathcal{L}$
   - Design
   - Syntax
   - Semantics
3. Proof Calculus for Distributed Hybrid Systems
   - Compositional Verification Calculus
   - Air Traffic Control
   - Derivations and Differentiation
   - Soundness and Completeness
4. Conclusions

$$\frac{\forall i\,(i = u \to \phi(\theta))}{\phi([\forall i\, x(i) := \theta]x(u))}$$

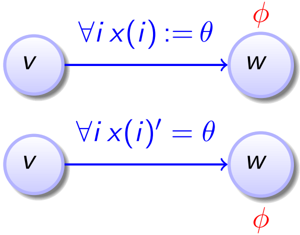


Proof Calculus for Quantified Differential Dynamic Logic

$$\frac{\forall i\,(i = [\forall i\, x(i) := \theta]u \rightarrow \phi(\theta))}{\phi([\forall i\, x(i) := \theta]x(u))}$$

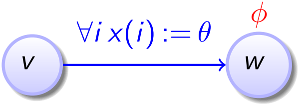


$$\frac{\forall i\,(i = [\forall i\, x(i) := \theta]u \rightarrow \phi(\theta))}{\phi([\forall i\, x(i) := \theta]x(u))}$$

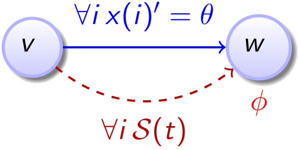


$$\frac{\exists t{\geq}0\ \langle \forall i\, \mathcal{S}(t)\rangle \phi}{\langle \forall i\, x(i)' = \theta \rangle \phi}$$

$$\frac{\forall i\,(i = [\forall i\, x(i) := \theta]u \to \phi(\theta))}{\phi([\forall i\, x(i) := \theta]x(u))}$$



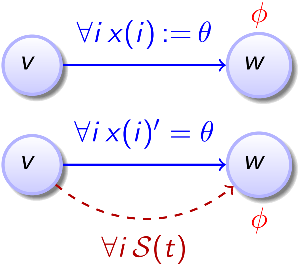
$$\frac{\exists t{\geq}0\, \langle \forall i\, \mathcal{S}(t)\rangle \phi}{\langle \forall i\, x(i)' = \theta\rangle \phi}$$



Proof Calculus for Quantified Differential Dynamic Logic

$$\frac{\forall i\, (i = [\forall i\, x(i) := \theta]u \rightarrow \phi(\theta))}{\phi([\forall i\, x(i) := \theta]x(u))}$$

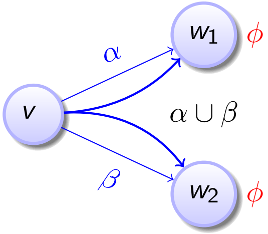
$$\frac{\exists t{\geq}0\, \langle \forall i\, \mathcal{S}(t)\rangle \phi}{\langle \forall i\, x(i)' = \theta\rangle \phi}$$



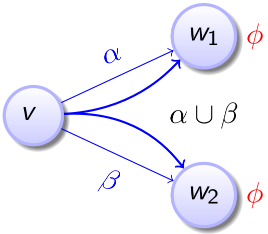
solve infinite-dimensional diff. eqn.?

compositional semantics $\Rightarrow$ compositional rules!

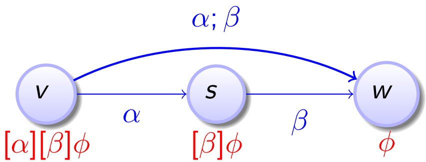
$$\frac{[\alpha]\phi \land [\beta]\phi}{[\alpha \cup \beta]\phi}$$



$$\frac{[\alpha]\phi \wedge [\beta]\phi}{[\alpha \cup \beta]\phi}$$

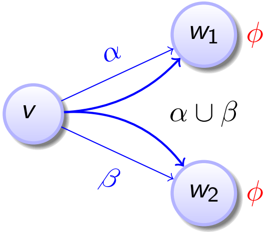


$$\frac{[\alpha][\beta]\phi}{[\alpha; \beta]\phi}$$

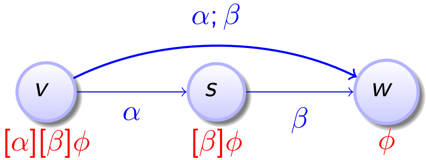


Proof Calculus for Quantified Differential Dynamic Logic

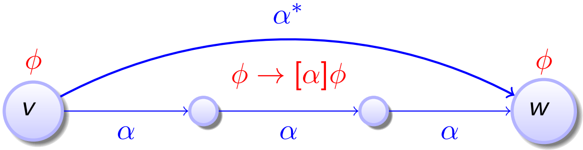
$$\frac{[\alpha]\phi \land [\beta]\phi}{[\alpha \cup \beta]\phi}$$



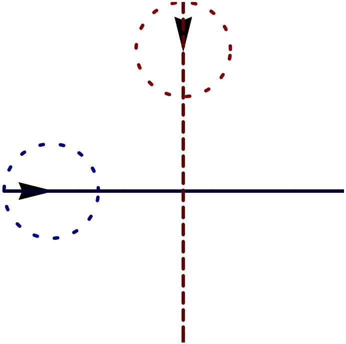
$$\frac{[\alpha][\beta]\phi}{[\alpha;\beta]\phi}$$



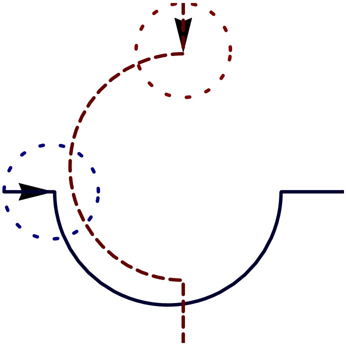
$$\frac{\phi \quad (\phi \to [\alpha]\phi)}{[\alpha^*]\phi}$$



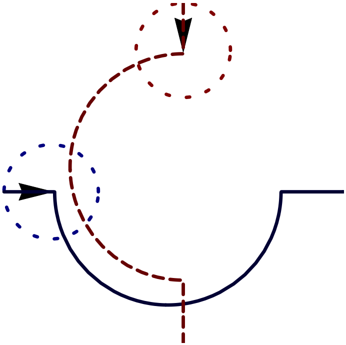
Air Traffic Control



Air Traffic Control



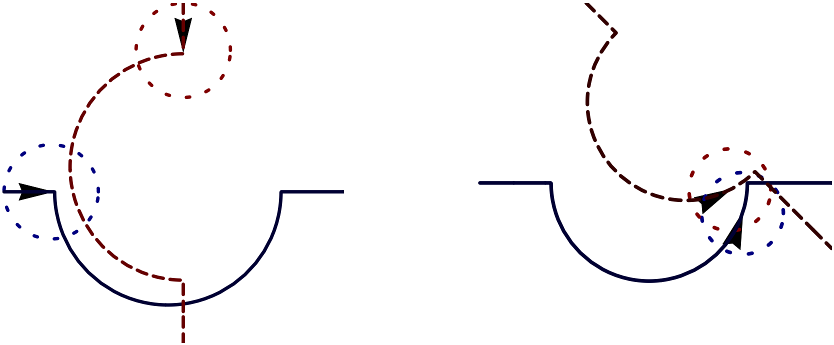
Air Traffic Control



Verification?

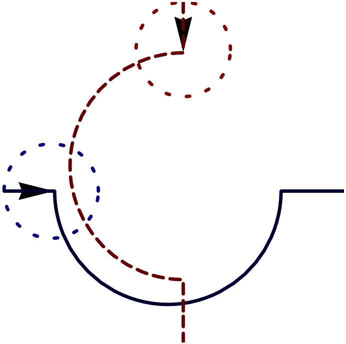
looks correct

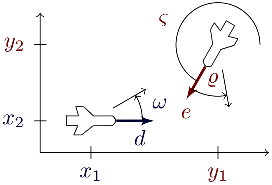
Air Traffic Control



Verification?

looks correct NO!

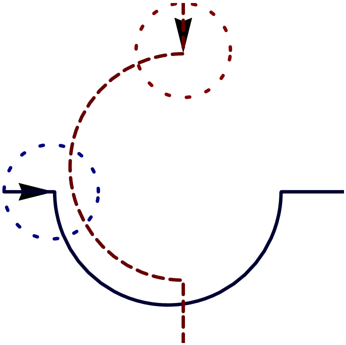


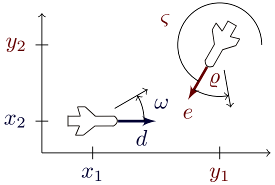


$$\begin{bmatrix} x_1' = -v_1 + v_2 \cos\vartheta + \omega x_2 \\ x_2' = v_2 \sin\vartheta - \omega x_1 \\ \vartheta' = \varpi - \omega \end{bmatrix}$$

Verification?

looks correct NO!





$$\begin{bmatrix} x_1' = -v_1 + v_2 \cos\vartheta + \omega x_2 \\ x_2' = v_2 \sin\vartheta - \omega x_1 \\ \vartheta' = \varpi - \omega \end{bmatrix}$$

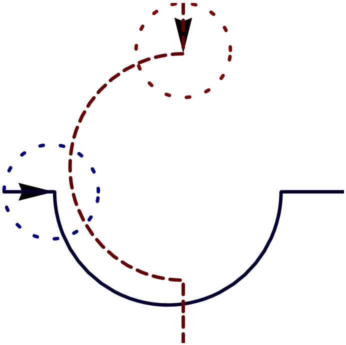
Example ("Solving" differential equations)

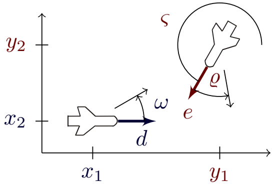
$$x_1(t) = \frac{1}{\omega\varpi}\big(x_1\omega\varpi \cos t\omega - v_2\omega \cos t\omega \sin\vartheta + v_2\omega \cos t\omega \cos t\varpi \sin\vartheta - v_1\varpi \sin t\omega$$

$$+ x_2\omega\varpi \sin t\omega - v_2\omega \cos\vartheta \cos t\varpi \sin t\omega - v_2\omega\sqrt{1 - \sin\vartheta^2} \sin t\omega$$

$$+ v_2\omega \cos\vartheta \cos t\omega \sin t\varpi + v_2\omega \sin\vartheta \sin t\omega \sin t\varpi\big) \ldots$$

Air Traffic Control





$$\begin{bmatrix} x_1' = -v_1 + v_2 \cos\vartheta + \omega x_2 \\ x_2' = v_2 \sin\vartheta - \omega x_1 \\ \vartheta' = \varpi - \omega \end{bmatrix}$$

Example ("Solving" differential equations)

$$\forall t{\geq}0 \quad \frac{1}{\omega\varpi}\big(x_1\omega\varpi \cos t\omega - v_2\omega \cos t\omega \sin\vartheta + v_2\omega \cos t\omega \cos t\varpi \sin\vartheta - v_1\varpi \sin t\omega$$

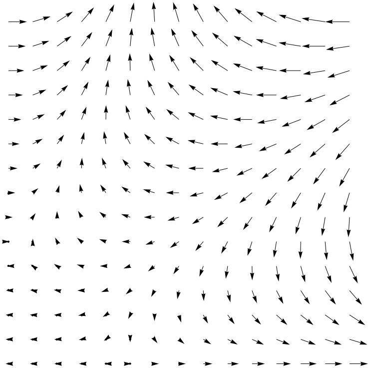
$$+ x_2\omega\varpi \sin t\omega - v_2\omega \cos\vartheta \cos t\varpi \sin t\omega - v_2\omega\sqrt{1 - \sin\vartheta^2}\sin t\omega$$

$$+ v_2\omega \cos\vartheta \cos t\omega \sin t\varpi + v_2\omega \sin\vartheta \sin t\omega \sin t\varpi\big) \ldots$$

Differential Invariants for Differential Equations

Idea (Differential Invariant)

Formula that remains true in the direction of the dynamics

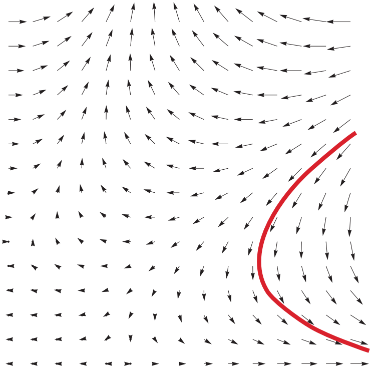


André Platzer.
Differential-algebraic dynamic logic for differential-algebraic programs.
J. Log. Comput., 35(1): 309–352, 2010.

Differential Invariants for Differential Equations

Idea (Differential Invariant)

Formula that remains true in the direction of the dynamics

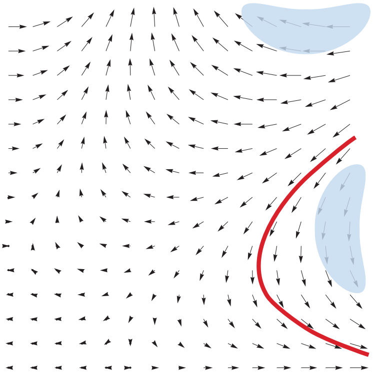


André Platzer.
Differential-algebraic dynamic logic for differential-algebraic programs.
J. Log. Comput., 35(1): 309–352, 2010.

Differential Invariants for Differential Equations

Idea (Differential Invariant)

Formula that remains true in the direction of the dynamics

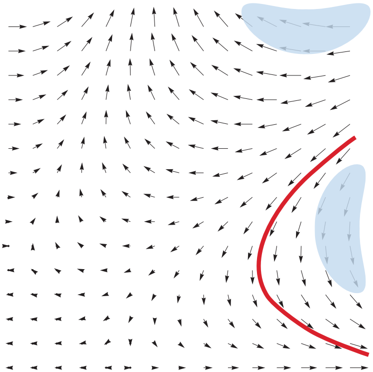


André Platzer.
Differential-algebraic dynamic logic for differential-algebraic programs.
J. Log. Comput., 35(1): 309–352, 2010.

Differential Invariants for Differential Equations

Idea (Differential Invariant)

Formula that remains true in the direction of the dynamics



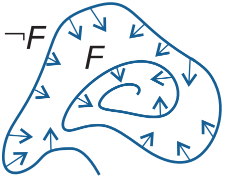
$\mathbb{R}^2$ but $\mathbb{R}^\infty$??

André Platzer.
Differential-algebraic dynamic logic for differential-algebraic programs.
J. Log. Comput., 35(1): 309–352, 2010.

Differential Induction: Local Dynamics w/o Solutions

Definition (Differential Invariant)

F closed under total differentiation with respect to differential constraints



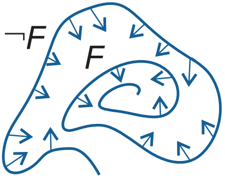
Details

$$\frac{(\chi \to F')}{\chi \to F \to [x' = \theta \wedge \chi]F}$$

Differential Induction: Local Dynamics w/o Solutions

Definition (Differential Invariant)

F closed under total differentiation with respect to differential constraints



$$\frac{(\chi \rightarrow F')}{\chi \rightarrow F \rightarrow [x' = \theta \land \chi]F}$$

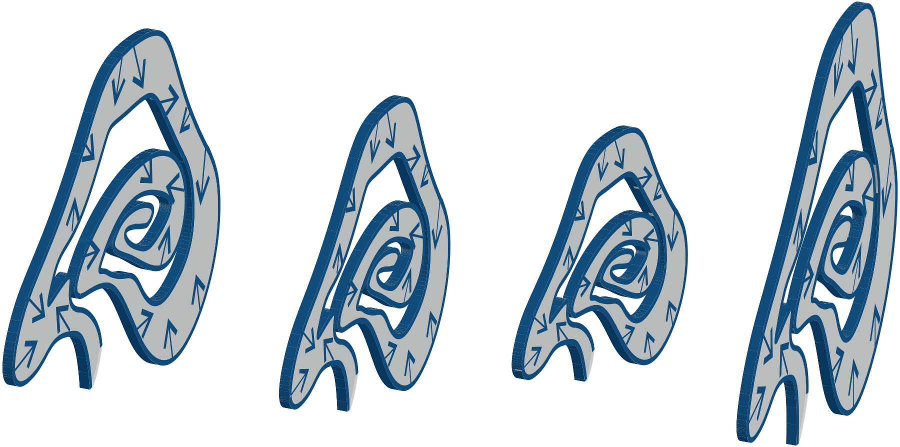
Details

Total differential F' of *formulas*?

Quantified Differential Invariants

Definition (Quantified Differential Invariant)

Quantified formula F closed under total differentiation with respect to quantified differential constraints



Derivations and Differentiation

Definition (Syntactic total derivation D)

$$
\begin{aligned}
D(r) &= 0 && \text{if } r \text{ a number symbol}\\
D(x(i)) &= x(i)' && \text{if } x : C \to \mathbb{R},\ C \neq \mathbb{R}\\
D(a+b) &= D(a) + D(b)\\
D(a \cdot b) &= D(a) \cdot b + a \cdot D(b)\\
D(a/b) &= (D(a) \cdot b - a \cdot D(b))/b^2
\end{aligned}
$$

Derivations and Differentiation

Definition (Syntactic total derivation D)

$$
\begin{aligned}
D(r) &= 0 && \text{if } r \text{ a number symbol} \\
D(x(i)) &= x(i)' && \text{if } x : C \to \mathbb{R},\ C \neq \mathbb{R} \\
D(a+b) &= D(a) + D(b) \\
D(a \cdot b) &= D(a) \cdot b + a \cdot D(b) \\
D(a/b) &= (D(a) \cdot b - a \cdot D(b))/b^2 \\
D(a \geq b) &\equiv D(a) \geq D(b) && \text{accordingly for } >, = \\
D(F \wedge G) &\equiv D(F) \wedge D(G) \\
D(\forall i\, F) &\equiv \forall i\, D(F)
\end{aligned}
$$

Derivations and Differentiation

Definition (Syntactic total derivation D)

$$
\begin{aligned}
D(r) &= 0 && \text{if } r \text{ a number symbol}\\
D(x(i)) &= x(i)' && \text{if } x : C \to \mathbb{R},\ C \neq \mathbb{R}\\
D(a+b) &= D(a) + D(b)\\
D(a\cdot b) &= D(a)\cdot b + a \cdot D(b)\\
D(a/b) &= (D(a)\cdot b - a\cdot D(b))/b^2\\
D(a \geq b) &\equiv D(a) \geq D(b) && \text{accordingly for } >, =\\
D(F \wedge G) &\equiv D(F) \wedge D(G)\\
D(\forall i\, F) &\equiv \forall i\, D(F)
\end{aligned}
$$

$$
\begin{aligned}
\mathcal{P} &\equiv \forall i,j : A\ \left(i = j \vee (x_1(i) - x_1(j))^2 + (x_2(i) - x_2(j))^2 \geq p^2\right)\\
\Rightarrow D(\mathcal{P}) &\equiv \forall i,j : A\ \big(i' = j' \wedge 2(x_1(i) - x_1(j))(x_1(i)' - x_1(j)')\\
&\qquad + 2(x_2(i) - x_2(j))(x_2(i)' - x_2(j)') \geq 0\big)
\end{aligned}
$$

Derivations and Differentiation

Definition (Syntactic total derivation D)

$$
\begin{aligned}
D(r) &= 0 && \text{if } r \text{ a number symbol}\\
D(x(i)) &= x(i)' && \text{if } x : C \to \mathbb{R},\ C \neq \mathbb{R}\\
D(a+b) &= D(a) + D(b)\\
D(a \cdot b) &= D(a)\cdot b + a \cdot D(b)\\
D(a/b) &= (D(a)\cdot b - a\cdot D(b))/b^2\\
D(a \geq b) &\equiv D(a) \geq D(b) && \text{accordingly for } >, =\\
D(F \land G) &\equiv D(F) \land D(G)\\
D(\forall i\, F) &\equiv \forall i\, D(F)
\end{aligned}
$$

$$
\begin{aligned}
\mathcal{P} &\equiv \forall i,j\,{:}\,A\ \left(i = j \lor (x_1(i) - x_1(j))^2 + (x_2(i) - x_2(j))^2 \geq p^2\right)\\
\Rightarrow D(\mathcal{P}) &\equiv \forall i,j\,{:}\,A\ \Big(i' = j' \land 2(x_1(i) - x_1(j))(x_1(i)' - x_1(j)')\\
&\qquad + 2(x_2(i) - x_2(j))(x_2(i)' - x_2(j)') \geq 0\Big)
\end{aligned}
$$

Derivations and Differentiation

Syntactic derivation $D(\cdot)$ coincides with analytic differentiation:

Lemma (Derivation lemma)

Valuation is a differential homomorphism: for all flows φ all $\zeta \in [0, r]$

$$\frac{\mathrm{d}\,\varphi(t)[\![\theta]\!]}{\mathrm{d}t}(\zeta) = \bar{\varphi}(\zeta)[\![D(\theta)]\!]$$

Derivations and Differentiation

Syntactic derivation $D(\cdot)$ coincides with analytic differentiation:

Lemma (Derivation lemma)

Valuation is a differential homomorphism: for all flows φ *all* $\zeta \in [0, r]$

$$\frac{\mathrm{d}\,\varphi(t)[\![\theta]\!]}{\mathrm{d}t}(\zeta) = \bar{\varphi}(\zeta)[\![D(\theta)]\!]$$

Locally understand QDE as quantified assignments:

Lemma (Quantified differential substitution principle)

If $\varphi \models \forall i : C\ f(i)' = \theta \wedge \chi$, *then* $\varphi \models \upsilon = [\forall i : C\ f(i)' := \theta]\upsilon$ *for all* υ.

Derivations and Differentiation

Syntactic derivation $D(\cdot)$ coincides with analytic differentiation:

Lemma (Derivation lemma)

Valuation is a differential homomorphism: for all flows φ *all* $\zeta \in [0, r]$

$$\frac{\mathrm{d}\,\varphi(t)[\![\theta]\!]}{\mathrm{d}t}(\zeta) = \bar{\varphi}(\zeta)[\![D(\theta)]\!]$$

Locally understand QDE as quantified assignments:

Lemma (Quantified differential substitution principle)

If $\varphi \models \forall i : C\ f(i)' = \theta \wedge \chi$*, then* $\varphi \models \upsilon = [\forall i : C\ f(i)' := \theta]\upsilon$ *for all* υ*.*

Theorem (Quantified Differential Invariant)

$$(QDI)\quad \frac{\chi \rightarrow [\forall i : C\ f(i)' := \theta]D(F)}{F \rightarrow [\forall i : C\ f(i)' = \theta \wedge \chi]F} \quad \textit{is sound}$$

A Simple Proof with Quantified Differential Invariants

$$\dfrac{}{\forall i : C\ 2x(i)^3 \geq 1 \rightarrow [\forall i : C\ x(i)' = x(i)^2 + x(i)^4 + 2]\forall i : C\ 2x(i)^3 \geq 1}$$

A Simple Proof with Quantified Differential Invariants

$$\dfrac{[\forall i : C\ x(i)' := x(i)^2 + x(i)^4 + 2]\forall i : C\ 2(x(i)^3)' \geq 0}{\forall i : C\ 2x(i)^3 \geq 1 \rightarrow [\forall i : C\ x(i)' = x(i)^2 + x(i)^4 + 2]\forall i : C\ 2x(i)^3 \geq 1}$$

A Simple Proof with Quantified Differential Invariants

$$\dfrac{\dfrac{[\forall i : C\ x(i)' := x(i)^2 + x(i)^4 + 2]\forall i : C\ 6x(i)^2 x(i)' \geq 0}{[\forall i : C\ x(i)' := x(i)^2 + x(i)^4 + 2]\forall i : C\ 2(x(i)^3)' \geq 0}}{\forall i : C\ 2x(i)^3 \geq 1 \rightarrow [\forall i : C\ x(i)' = x(i)^2 + x(i)^4 + 2]\forall i : C\ 2x(i)^3 \geq 1}$$

A Simple Proof with Quantified Differential Invariants

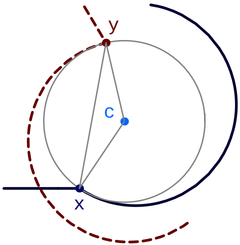
$$\dfrac{\dfrac{\dfrac{\forall i : C\ 6x(i)^2(x(i)^2 + x(i)^4 + 2) \geq 0}{[\forall i : C\ x(i)' := x(i)^2 + x(i)^4 + 2]\forall i : C\ 6x(i)^2 x(i)' \geq 0}}{[\forall i : C\ x(i)' := x(i)^2 + x(i)^4 + 2]\forall i : C\ 2(x(i)^3)' \geq 0}}{\forall i : C\ 2x(i)^3 \geq 1 \rightarrow [\forall i : C\ x(i)' = x(i)^2 + x(i)^4 + 2]\forall i : C\ 2x(i)^3 \geq 1}$$

A Simple Proof with Quantified Differential Invariants

$$\dfrac{\dfrac{\dfrac{\dfrac{true}{\forall i : C\ 6x(i)^2(x(i)^2 + x(i)^4 + 2) \geq 0}}{[\forall i : C\ x(i)' := x(i)^2 + x(i)^4 + 2]\forall i : C\ 6x(i)^2 x(i)' \geq 0}}{[\forall i : C\ x(i)' := x(i)^2 + x(i)^4 + 2]\forall i : C\ 2(x(i)^3)' \geq 0}}{\forall i : C\ 2x(i)^3 \geq 1 \rightarrow [\forall i : C\ x(i)' = x(i)^2 + x(i)^4 + 2]\forall i : C\ 2x(i)^3 \geq 1}$$

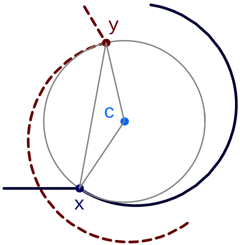
Differential Induction for Aircraft Roundabouts

$$[\forall i\, x_1(i)' = d_1(i), d_1(i)' = -\omega d_2(i), x_2(i)' = d_2(i), d_2(i)' = \omega d_1(i)](x_1(i) - x_1(j))^2 + (x_2(i)$$

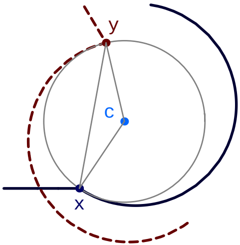


Differential Induction for Aircraft Roundabouts

$$\frac{i' = j' \land 2(x_1(i) - x_1(j))(x_1(i)' - x_1(j)') + 2(x_2(i) - x_2(j))(x_2(i)' - x_2(j)') \geq 0}{[\forall i\, x_1(i)' = d_1(i), d_1(i)' = -\omega d_2(i), x_2(i)' = d_2(i), d_2(i)' = \omega d_1(i)](x_1(i) - x_1(j))^2 + (x_2(i)}$$



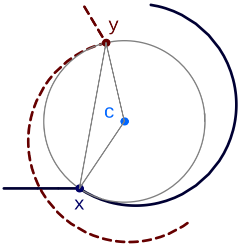
$$\frac{i' = j' \land 2(x_1(i) - x_1(j))(x_1(i)' - x_1(j)') + 2(x_2(i) - x_2(j))(x_2(i)' - x_2(j)') \geq 0}{[\forall i\, x_1(i)' = d_1(i), d_1(i)' = -\omega d_2(i), x_2(i)' = d_2(i), d_2(i)' = \omega d_1(i)](x_1(i) - x_1(j))^2 + (x_2(i)}$$



Differential Induction for Aircraft Roundabouts

$$0 = 0 \land 2(x_1(i) - x_1(j))(d_1(i) - d_1(j)) + 2(x_2(i) - x_2(j))(d_2(i) - d_2(j)) \geq 0$$

$$[\forall i\, x_1(i)' = d_1(i), d_1(i)' = -\omega d_2(i), x_2(i)' = d_2(i), d_2(i)' = \omega d_1(i)](x_1(i) - x_1(j))^2 + (x_2(i)$$

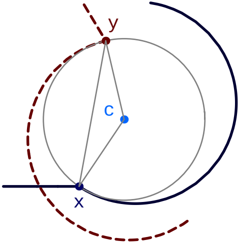


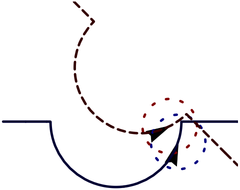
Differential Induction for Aircraft Roundabouts

$$2(x_1(i) - x_1(j))(d_1(i) - d_1(j)) + 2(x_2(i) - x_2(j))(d_2(i) - d_2(j)) \geq 0$$

$$0 = 0 \land 2(x_1(i) - x_1(j))(d_1(i) - d_1(j)) + 2(x_2(i) - x_2(j))(d_2(i) - d_2(j)) \geq 0$$

$$[\forall i\, x_1(i)' = d_1(i), d_1(i)' = -\omega d_2(i), x_2(i)' = d_2(i), d_2(i)' = \omega d_1(i)](x_1(i) - x_1(j))^2 + (x_2(i)$$



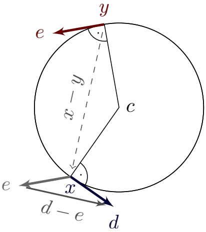


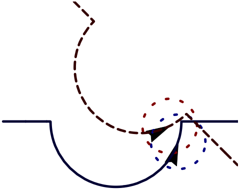
Differential Induction for Aircraft Roundabouts

$$2(x_1(i) - x_1(j))(d_1(i) - d_1(j)) + 2(x_2(i) - x_2(j))(d_2(i) - d_2(j)) \geq 0$$

$$0 = 0 \land 2(x_1(i) - x_1(j))(d_1(i) - d_1(j)) + 2(x_2(i) - x_2(j))(d_2(i) - d_2(j)) \geq 0$$

$$[\forall i\, x_1(i)' = d_1(i), d_1(i)' = -\omega d_2(i), x_2(i)' = d_2(i), d_2(i)' = \omega d_1(i)](x_1(i) - x_1(j))^2 + (x_2(i)$$



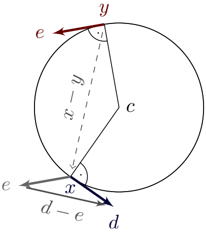


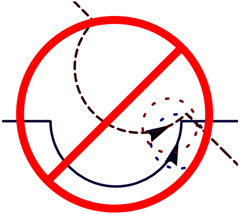
Differential Induction for Aircraft Roundabouts

$$2(x_1(i) - x_1(j))(d_1(i) - d_1(j)) + 2(x_2(i) - x_2(j))(d_2(i) - d_2(j)) \geq 0$$

$$0 = 0 \land 2(x_1(i) - x_1(j))(d_1(i) - d_1(j)) + 2(x_2(i) - x_2(j))(d_2(i) - d_2(j)) \geq 0$$

$$[\forall i\, x_1(i)' = d_1(i), d_1(i)' = -\omega d_2(i), x_2(i)' = d_2(i), d_2(i)' = \omega d_1(i)](x_1(i) - x_1(j))^2 + (x_2(i)$$





$$[\forall i\, x_1(i)' = d_1(i), d_1(i)' = -\omega d_2(i), x_2(i)' = d_2(i), d_2(i)' = \omega d_1(i)]d_1(i) - d_1(j) = -\omega(x_2(i$$

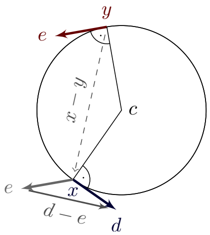
Differential Induction for Aircraft Roundabouts

$$2(x_1(i) - x_1(j))(-\omega(x_2(i) - x_2(j))) + 2(x_2(i) - x_2(j))\omega(x_1(i) - x_1(j)) \geq 0$$

$$2(x_1(i) - x_1(j))(d_1(i) - d_1(j)) + 2(x_2(i) - x_2(j))(d_2(i) - d_2(j)) \geq 0$$

$$0 = 0 \land 2(x_1(i) - x_1(j))(d_1(i) - d_1(j)) + 2(x_2(i) - x_2(j))(d_2(i) - d_2(j)) \geq 0$$

$$[\forall i\, x_1(i)' = d_1(i), d_1(i)' = -\omega d_2(i), x_2(i)' = d_2(i), d_2(i)' = \omega d_1(i)](x_1(i) - x_1(j))^2 + (x_2(i)$$



$$[\forall i\, x_1(i)' = d_1(i), d_1(i)' = -\omega d_2(i), x_2(i)' = d_2(i), d_2(i)' = \omega d_1(i)]d_1(i) - d_1(j) = -\omega(x_2(i$$

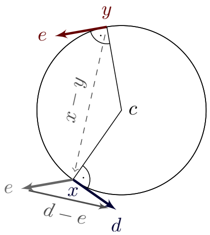
Differential Induction for Aircraft Roundabouts

$$2(x_1(i) - x_1(j))(-\omega(x_2(i) - x_2(j))) + 2(x_2(i) - x_2(j))\omega(x_1(i) - x_1(j)) \geq 0$$

$$2(x_1(i) - x_1(j))(d_1(i) - d_1(j)) + 2(x_2(i) - x_2(j))(d_2(i) - d_2(j)) \geq 0$$

$$0 = 0 \land 2(x_1(i) - x_1(j))(d_1(i) - d_1(j)) + 2(x_2(i) - x_2(j))(d_2(i) - d_2(j)) \geq 0$$

$$[\forall i\, x_1(i)' = d_1(i), d_1(i)' = -\omega d_2(i), x_2(i)' = d_2(i), d_2(i)' = \omega d_1(i)](x_1(i) - x_1(j))^2 + (x_2(i)$$



$$d_1(i)' - d_1(j)' = -\omega(x_2(i)' - x_2(j)')$$

$$[\forall i\, x_1(i)' = d_1(i), d_1(i)' = -\omega d_2(i), x_2(i)' = d_2(i), d_2(i)' = \omega d_1(i)]\,d_1(i) - d_1(j) = -\omega(x_2(i$$

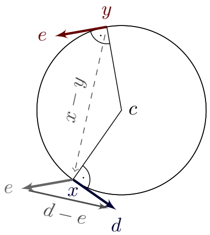
Differential Induction for Aircraft Roundabouts

$$2(x_1(i)-x_1(j))(-\omega(x_2(i)-x_2(j)))+2(x_2(i)-x_2(j))\omega(x_1(i)-x_1(j))\geq 0$$

$$2(x_1(i)-x_1(j))(d_1(i)-d_1(j))+2(x_2(i)-x_2(j))(d_2(i)-d_2(j))\geq 0$$

$$0=0\wedge 2(x_1(i)-x_1(j))(d_1(i)-d_1(j))+2(x_2(i)-x_2(j))(d_2(i)-d_2(j))\geq 0$$

$$[\forall i\, x_1(i)'=d_1(i), d_1(i)'=-\omega d_2(i), x_2(i)'=d_2(i), d_2(i)'=\omega d_1(i)](x_1(i)-x_1(j))^2+(x_2(i)$$



$$d_1(i)'-d_1(j)'=-\omega(x_2(i)'-x_2(j)')$$

$$[\forall i\, x_1(i)'=d_1(i), d_1(i)'=-\omega d_2(i), x_2(i)'=d_2(i), d_2(i)'=\omega d_1(i)]d_1(i)-d_1(j)=-\omega(x_2(i$$

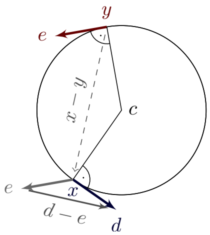
Differential Induction for Aircraft Roundabouts

$$2(x_1(i) - x_1(j))(-\omega(x_2(i) - x_2(j))) + 2(x_2(i) - x_2(j))\omega(x_1(i) - x_1(j)) \geq 0$$

$$2(x_1(i) - x_1(j))(d_1(i) - d_1(j)) + 2(x_2(i) - x_2(j))(d_2(i) - d_2(j)) \geq 0$$

$$0 = 0 \land 2(x_1(i) - x_1(j))(d_1(i) - d_1(j)) + 2(x_2(i) - x_2(j))(d_2(i) - d_2(j)) \geq 0$$

$$[\forall i\, x_1(i)' = d_1(i), d_1(i)' = -\omega d_2(i), x_2(i)' = d_2(i), d_2(i)' = \omega d_1(i)](x_1(i) - x_1(j))^2 + (x_2(i)$$



$$-\omega d_2(i) - -\omega d_2(j) = -\omega(d_2(i) - d_2(j))$$

$$[\forall i\, x_1(i)' = d_1(i), d_1(i)' = -\omega d_2(i), x_2(i)' = d_2(i), d_2(i)' = \omega d_1(i)]d_1(i) - d_1(j) = -\omega(x_2(i$$

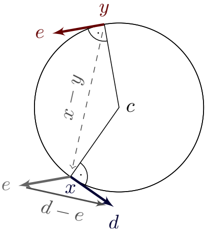
Differential Induction for Aircraft Roundabouts

$$2(x_1(i) - x_1(j))(-\omega(x_2(i) - x_2(j))) + 2(x_2(i) - x_2(j))\omega(x_1(i) - x_1(j)) \geq 0$$

$$2(x_1(i) - x_1(j))(d_1(i) - d_1(j)) + 2(x_2(i) - x_2(j))(d_2(i) - d_2(j)) \geq 0$$

$$0 = 0 \land 2(x_1(i) - x_1(j))(d_1(i) - d_1(j)) + 2(x_2(i) - x_2(j))(d_2(i) - d_2(j)) \geq 0$$

$$[\forall i\, x_1(i)' = d_1(i), d_1(i)' = -\omega d_2(i), x_2(i)' = d_2(i), d_2(i)' = \omega d_1(i)](x_1(i) - x_1(j))^2 + (x_2(i)$$



$$-\omega d_2(i) + \omega d_2(j) = -\omega(d_2(i) - d_2(j))$$

$$-\omega d_2(i) - -\omega d_2(j) = -\omega(d_2(i) - d_2(j))$$

$$[\forall i\, x_1(i)' = d_1(i), d_1(i)' = -\omega d_2(i), x_2(i)' = d_2(i), d_2(i)' = \omega d_1(i)]d_1(i) - d_1(j) = -\omega(x_2(i$$

Differential Induction & Differential Cuts

$$2(x_1(i) - x_1(j))(-\omega(x_2(i) - x_2(j))) + 2(x_2(i) - x_2(j))\omega(x_1(i) - x_1(j)) \geq 0$$

$$2(x_1(i) - x_1(j))(d_1(i) - d_1(j)) + 2(x_2(i) - x_2(j))(d_2(i) - d_2(j)) \geq 0$$

$$0 = 0 \wedge 2(x_1(i) - x_1(j))(d_1(i) - d_1(j)) + 2(x_2(i) - x_2(j))(d_2(i) - d_2(j)) \geq 0$$

$$[\forall i x_1(i)' = d_1(i), d_1(i)' = -\omega d_2(i), x_2(i)' = d_2(i), d_2(i)' = \omega d_1(i)](x_1(i) - x_1(j))^2 + (x_2(i)$$

Proposition (Differential cut)

F differential invariant of $[\forall i\, x(i)' = \theta \wedge H]\phi$, then

$$[\forall i\, x(i)' = \theta \wedge H]\phi \quad \text{iff} \quad [\forall i\, x(i)' = \theta \wedge H \wedge F]\phi$$

$$-\omega d_2(i) + \omega d_2(j) = -\omega(d_2(i) - d_2(j))$$

$$-\omega d_2(i) - -\omega d_2(j) = -\omega(d_2(i) - d_2(j))$$

$$[\forall i x_1(i)' = d_1(i), d_1(i)' = -\omega d_2(i), x_2(i)' = d_2(i), d_2(i)' = \omega d_1(i)]d_1(i) - d_1(j) = -\omega(x_2(i$$

Differential Induction & Differential Cuts

$$\frac{\frac{\frac{2(x_1(i)-x_1(j))(-\omega(x_2(i)-x_2(j)))+2(x_2(i)-x_2(j))\omega(x_1(i)-x_1(j))\geq 0}{2(x_1(i)-x_1(j))(d_1(i)-d_1(j))+2(x_2(i)-x_2(j))(d_2(i)-d_2(j))\geq 0}}{0=0\land 2(x_1(i)-x_1(j))(d_1(i)-d_1(j))+2(x_2(i)-x_2(j))(d_2(i)-d_2(j))\geq 0}}{[\forall i\, x_1(i)'=d_1(i), d_1(i)'=-\omega d_2(i), x_2(i)'=d_2(i), d_2(i)'=\omega d_1(i)](x_1(i)-x_1(j))^2+(x_2(i)}$$

refine dynamics

by differential cut

$$\frac{\frac{-\omega d_2(i)+\omega d_2(j)=-\omega(d_2(i)-d_2(j))}{-\omega d_2(i)--\omega d_2(j)=-\omega(d_2(i)-d_2(j))}}{[\forall i\, x_1(i)'=d_1(i), d_1(i)'=-\omega d_2(i), x_2(i)'=d_2(i), d_2(i)'=\omega d_1(i)]d_1(i)-d_1(j)=-\omega(x_2(i}$$

Soundness and Completeness

Theorem (Relative Completeness)

Qd$\mathcal{L}$ *calculus is a sound & complete axiomatisation of distributed hybrid systems relative to quantified differential equations.*

▸ Proof 16p.

Soundness and Completeness

Theorem (Relative Completeness)

QdL *calculus is a sound & complete axiomatisation of distributed hybrid systems relative to quantified differential equations.*

▸ Proof 16p.

Corollary (Proof-theoretical Alignment)

proving distributed hybrid systems = proving dynamical systems!

Soundness and Completeness

Theorem (Relative Completeness)

Qd$\mathcal{L}$ *calculus is a sound & complete axiomatisation of distributed hybrid systems relative to quantified differential equations.*

Proof 16p.

Corollary (Proof-theoretical Alignment)

proving distributed hybrid systems = proving dynamical systems!

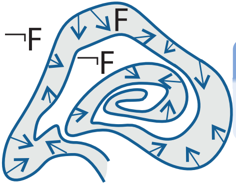
Corollary (Yes, we can!)

distributed hybrid systems can be verified by recursive decomposition

Outline

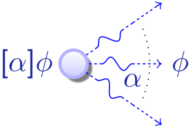
1. Motivation
2. Quantified Differential Dynamic Logic Qd$\mathcal{L}$
   - Design
   - Syntax
   - Semantics
3. Proof Calculus for Distributed Hybrid Systems
   - Compositional Verification Calculus
   - Air Traffic Control
   - Derivations and Differentiation
   - Soundness and Completeness
4. Conclusions

Conclusions

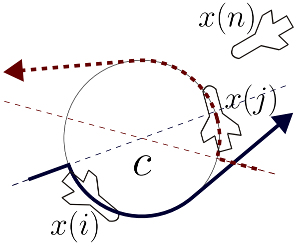


quantified differential dynamic logic

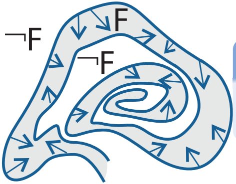
$$\mathrm{Qd}\mathcal{L} = \mathrm{FOL} + \mathrm{DL} + \mathrm{QHP}$$



- Quantified differential invariants
- Verify quantified differential equations
- Logic for distributed hybrid systems
- Compositional proof calculus
- Sound & complete / diff. eqn.
- First verification approach
- Verified appearance of aircraft

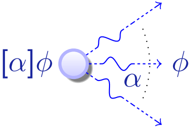


Conclusions

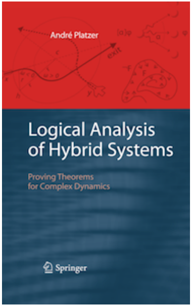


quantified differential dynamic logic

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



- Quantified differential invariants
- Verify quantified differential equations
- Logic for distributed hybrid systems
- Compositional proof calculus
- Sound & complete / diff. eqn.
- First verification approach
- Verified appearance of aircraft



Jan A. Bergstra and C. A. Middelburg.
Process algebra for hybrid systems.
Theor. Comput. Sci., 335(2-3):215–280, 2005.

Zhou Chaochen, Wang Ji, and Anders P. Ravn.
A formal description of hybrid systems.
In Rajeev Alur, Thomas A. Henzinger, and Eduardo D. Sontag, editors, *Hybrid Systems*, volume 1066 of *LNCS*, pages 511–530. Springer, 1995.

Pieter J. L. Cuijpers and Michel A. Reniers.
Hybrid process algebra.
J. Log. Algebr. Program., 62(2):191–245, 2005.

Akash Deshpande, Aleks Göllü, and Pravin Varaiya.
SHIFT: A formalism and a programming language for dynamic networks of hybrid automata.
In Panos J. Antsaklis, Wolf Kohn, Anil Nerode, and Shankar Sastry, editors, *Hybrid Systems*, volume 1273 of *LNCS*, pages 113–133. Springer, 1996.

João P. Hespanha and Ashish Tiwari, editors.
Hybrid Systems: Computation and Control, 9th International Workshop, HSCC 2006, Santa Barbara, CA, USA, March 29-31, 2006, Proceedings, volume 3927 of *LNCS*. Springer, 2006.

Fabian Kratz, Oleg Sokolsky, George J. Pappas, and Insup Lee.
R-Charon, a modeling language for reconfigurable hybrid systems.
In Hespanha and Tiwari [HT06], pages 392–406.

José Meseguer and Raman Sharykin.
Specification and analysis of distributed object-based stochastic hybrid systems.
In Hespanha and Tiwari [HT06], pages 460–475.

William C. Rounds.
A spatial logic for the hybrid π-calculus.
In Rajeev Alur and George J. Pappas, editors, *HSCC*, volume 2993 of *LNCS*, pages 508–522. Springer, 2004.

D. A. van Beek, Ka L. Man, Michel A. Reniers, J. E. Rooda, and Ramon R. H. Schiffelers.

Syntax and consistent equation semantics of hybrid Chi.
J. Log. Algebr. Program., 68(1-2):129–210, 2006.