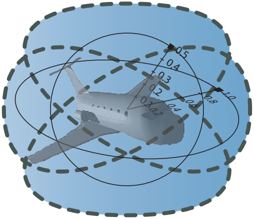
Logic for Distributed Hybrid Systems

André Platzer

Carnegie Mellon University, Pittsburgh, PA



Outline

1. Motivation
2. Quantified Differential Dynamic Logic Qd$\mathcal{L}$
   - Design
   - Syntax
3. Verification of Distributed Hybrid Systems
   - Soundness and Completeness
4. Conclusions

Complex Physical Systems:

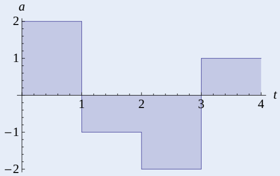
Q: I want to verify my car

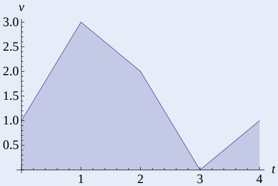
Challenge

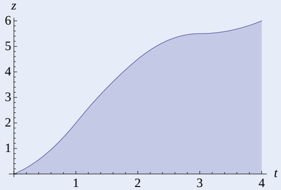
Q: I want to verify my car A: Hybrid systems

Challenge (Hybrid Systems)

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



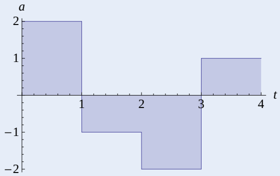


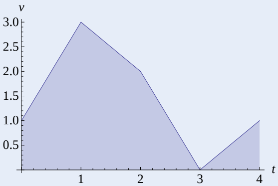


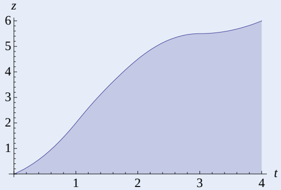
Q: I want to verify my car A: Hybrid systems Q: But there's a lot of cars!

Challenge (Hybrid Systems)

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



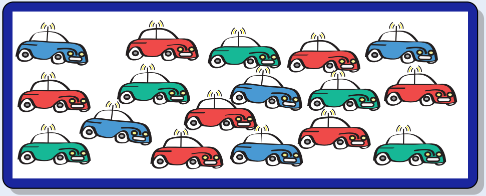




Complex Physical Systems:

Q: I want to verify a lot of cars

Challenge

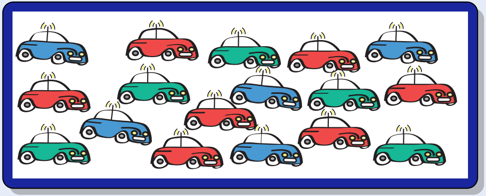


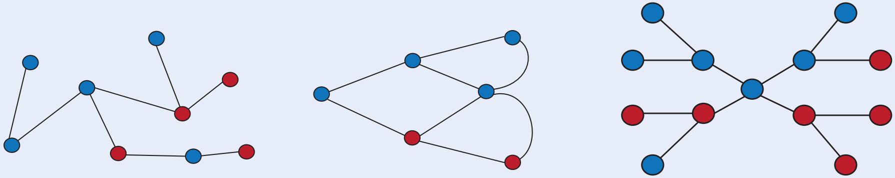
Complex Physical Systems: Distributed Systems

Q: I want to verify a lot of cars A: Distributed systems

Challenge (Distributed Systems)

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

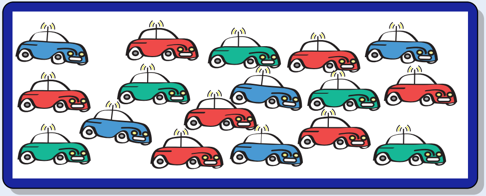


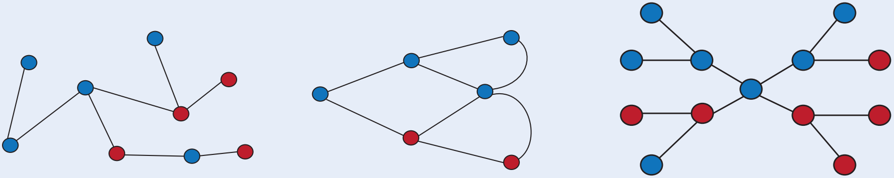


Q: I want to verify a lot of cars A: Distributed systems Q: But they move!

Challenge (Distributed Systems)

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

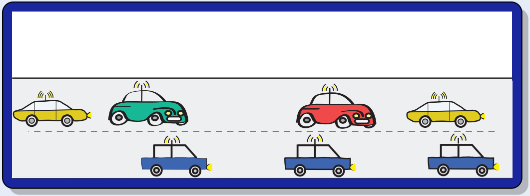




Complex Physical Systems:

Q: I want to verify lots of moving cars

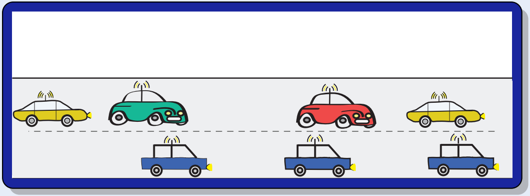
Challenge



Q: I want to verify lots of moving cars A: Distributed hybrid systems

Challenge (Distributed Hybrid Systems)

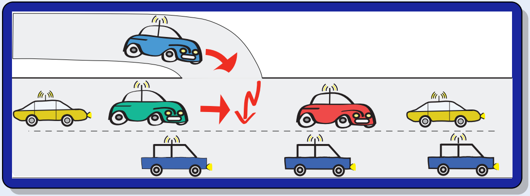
- Continuous dynamics (differential equations)
- Discrete dynamics (control decisions)
- Structural dynamics (remote communication)



Q: I want to verify lots of moving cars A: Distributed hybrid systems

Challenge (Distributed Hybrid Systems)

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

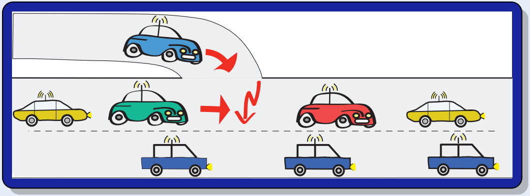


Complex Physical Systems: Distributed Hybrid Systems

Q: I want to verify lots of moving cars 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

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

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

Contributions

1. System model and semantics for distributed hybrid systems: QHP
2. Specification and verification logic: Qd$\mathcal{L}$
3. Compositional verification for Qd$\mathcal{L}$
4. First verification approach for distributed hybrid systems
5. Sound and complete relative to differential equations
6. Verify collision freedom in a (simple) distributed car control system, where new cars may appear dynamically on the road
7. Logical foundation for analysis of distributed hybrid systems
8. Fundamental extension: first-order $x(i)$ versus primitive x

Outline

1. Motivation
2. Quantified Differential Dynamic Logic Qd$\mathcal{L}$
   - Design
   - Syntax
3. Verification of Distributed Hybrid Systems
   - Soundness and Completeness
4. Conclusions

Outline (Conceptual Approach)

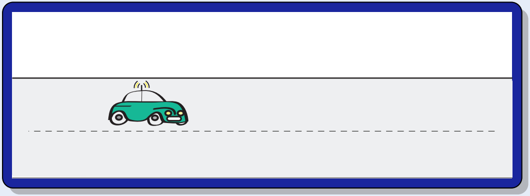
1. Motivation
2. Quantified Differential Dynamic Logic Qd$\mathcal{L}$
   - Design
   - Syntax
3. Verification of Distributed Hybrid Systems
   - Soundness and Completeness
4. Conclusions

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 (remote communication)

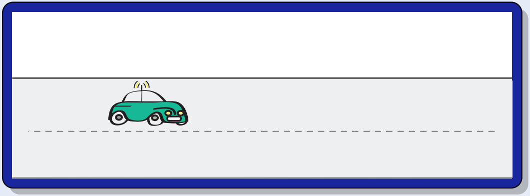


Model for Distributed Hybrid Systems

Q: How to model distributed hybrid systems

Model (Distributed Hybrid Systems)

- Continuous dynamics (differential equations)
  $x'' = a$
- Discrete dynamics (control decisions)
- Structural dynamics (remote communication)



Model for Distributed Hybrid Systems

Q: How to model distributed hybrid systems

Model (Distributed Hybrid Systems)

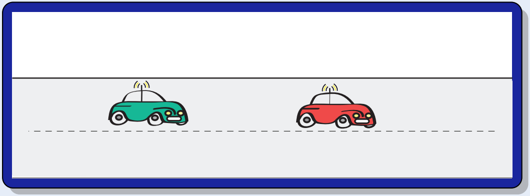
- Continuous dynamics (differential equations)

  $$x'' = a$$

- Discrete dynamics (control decisions)

  $a := \texttt{if} .. \texttt{then} \, A \, \texttt{else} \, -b$

- Structural dynamics (remote communication)



Model for Distributed Hybrid Systems

Q: How to model distributed hybrid systems

Model (Distributed Hybrid Systems)

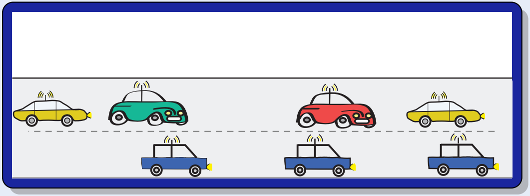
- Continuous dynamics (differential equations)

  $$x'' = a$$

- Discrete dynamics (control decisions)

  $a := \texttt{if} .. \texttt{then}\, A\, \texttt{else} -b$

- Structural dynamics (remote communication)



Model for Distributed Hybrid Systems

Q: How to model distributed hybrid systems

Model (Distributed Hybrid Systems)

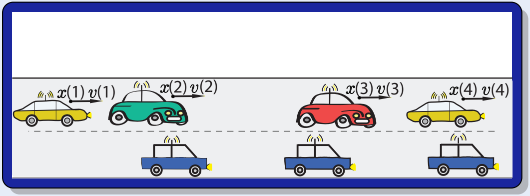
- Continuous dynamics (differential equations)

  $$x'' = a$$

- Discrete dynamics (control decisions)

  $a := \texttt{if} .. \texttt{then}\, A\, \texttt{else} -b$

- Structural dynamics (remote communication)

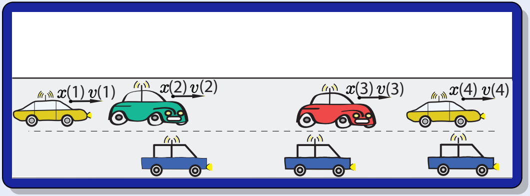


Model for Distributed Hybrid Systems

Q: How to model distributed hybrid systems

Model (Distributed Hybrid Systems)

- Continuous dynamics (differential equations)
  $$x(i)'' = a(i)$$
- Discrete dynamics (control decisions)
  $$a(i) := \texttt{if} .. \texttt{then}\, A\, \texttt{else} -b$$
- Structural dynamics (remote communication)

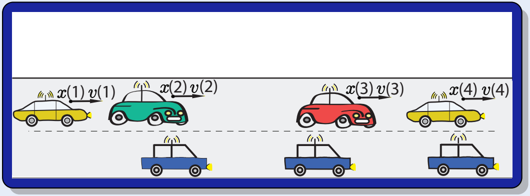


Model for Distributed Hybrid Systems

Q: How to model distributed hybrid systems

Model (Distributed Hybrid Systems)

- Continuous dynamics (differential equations)
  $\forall i \; x(i)'' = a(i)$
- Discrete dynamics (control decisions)
  $\forall i \; a(i) := \texttt{if} .. \texttt{then} \; A \; \texttt{else} \; -b$
- Structural dynamics (remote communication)

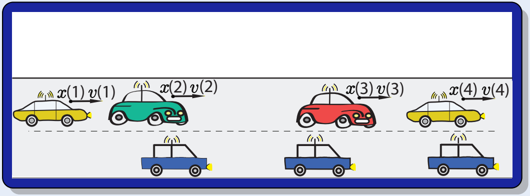


Model for Distributed Hybrid Systems

Q: How to model distributed hybrid systems

Model (Distributed Hybrid Systems)

- Continuous dynamics (differential equations)
  $\forall i\, x(i)'' = a(i)$
- Discrete dynamics (control decisions)
  $\forall i\, a(i) := \texttt{if} .. \texttt{then}\, A\, \texttt{else} -b$
- Structural dynamics (remote communication)
  $\ell(i) := carInFrontOf(i)$

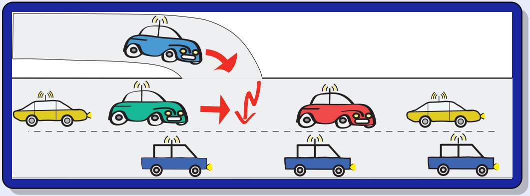


Model for Distributed Hybrid Systems

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

Model (Distributed Hybrid Systems)

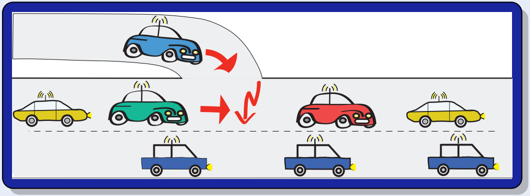
- Continuous dynamics (differential equations)
  $\forall i\, x(i)'' = a(i)$
- Discrete dynamics (control decisions)
  $\forall i\, a(i) := \texttt{if} .. \texttt{then}\, A\, \texttt{else} -b$
- Structural dynamics (remote communication)
  $\ell(i) := carInFrontOf(i)$
- Dimensional dynamics (appearance)



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

Model (Distributed Hybrid Systems)

- Continuous dynamics (differential equations)
  $\forall i\, x(i)'' = a(i)$
- Discrete dynamics (control decisions)
  $\forall i\, a(i) := \mathtt{if}\ ..\ \mathtt{then}\ A\ \mathtt{else}\ -b$
- Structural dynamics (remote communication)
  $\ell(i) := carInFrontOf(i)$
- Dimensional dynamics (appearance)
  $n := \mathtt{new}\ Car$



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

Definition (Quantified hybrid program α)

$\forall i : C\ x(s)' = \theta$	(quantified ODE)	
$\forall i : C\ x(s) := \theta$	(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

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

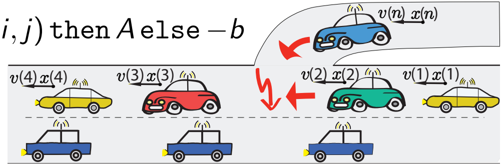
Definition (Quantified hybrid program α)

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

$$DCCS \equiv (ctrl; drive)^*$$

$$ctrl \equiv \forall i : C \; a(i) := \texttt{if} \; \forall j : C \; far(i,j) \; \texttt{then} \; A \; \texttt{else} \; -b$$

$$drive \equiv \forall i : C \; x(i)'' = a(i)$$



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

Definition (Quantified hybrid program α)

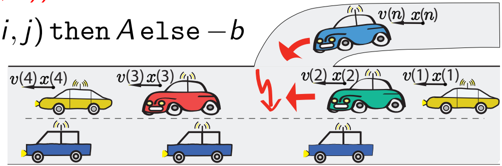
$\forall i : C\ x(s)' = \theta$	(quantified ODE)	
$\forall i : C\ x(s) := \theta$	(quantified assignment)	jump & test
$?\chi$	(conditional execution)	
$\alpha; \beta$	(seq. composition)	Kleene algebra
$\alpha \cup \beta$	(nondet. choice)	
α^*	(nondet. repetition)	

$$DCCS \equiv (appear; ctrl; drive)^*$$

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

$$ctrl \equiv \forall i : C\ a(i) := \mathtt{if}\ \forall j : C\ far(i, j)\ \mathtt{then}\ A\ \mathtt{else}\ -b$$

$$drive \equiv \forall i : C\ x(i)'' = a(i)$$



Definition (Quantified hybrid program α)

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

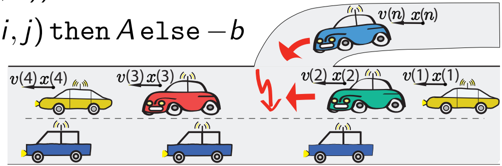
$$DCCS \equiv (appear\,;\, ctrl\,;\, drive)^*$$

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

$$ctrl \equiv \forall i : C\ a(i) := \texttt{if}\ \forall j : C\ far(i, j)\ \texttt{then}\ A\ \texttt{else}\ -b$$

$$drive \equiv \forall i : C\ x(i)'' = a(i)$$

new C is definable!



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

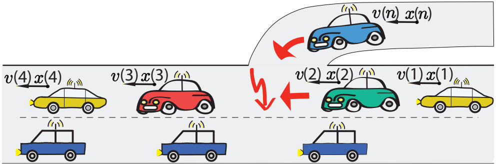
Definition (Qd$\mathcal{L}$ Formula ϕ)

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

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

$$\forall i, j : C \ \mathit{far}(i,j) \rightarrow [(\mathit{appear}; \mathit{ctrl}; \mathit{drive})^*] \ \forall i{\neq}j : C \ x(i) \neq x(j)$$

$$\mathit{far}(i,j) \equiv i \neq j \rightarrow x(i) < x(j) \wedge v(i) \leq v(j) \wedge a(i) \leq a(j) \\ \vee x(i) > x(j) \wedge v(i) \geq v(j) \wedge a(i) \geq a(j) \ldots$$



Outline (Verification Approach)

1. Motivation
2. Quantified Differential Dynamic Logic Qd$\mathcal{L}$
   - Design
   - Syntax
3. Verification of Distributed Hybrid Systems
   - Soundness and Completeness
4. Conclusions

Soundness and Completeness

Theorem (Relative Completeness)

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

▸ Proof 16p.

Soundness and Completeness

Theorem (Relative Completeness)

QdL *verification 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$\mathcal{L}$ *verification 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

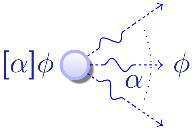
Outline

1. Motivation
2. Quantified Differential Dynamic Logic Qd$\mathcal{L}$
   - Design
   - Syntax
3. Verification of Distributed Hybrid Systems
   - Soundness and Completeness
4. Conclusions

Conclusions

quantified differential dynamic logic

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

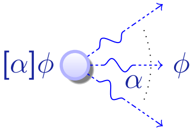


- Distributed hybrid systems everywhere
- System model and semantics
- Logic for distributed hybrid systems
- Compositional verification
- First verification approach
- Sound & complete / diff. eqn.
- Simple distributed car control verified

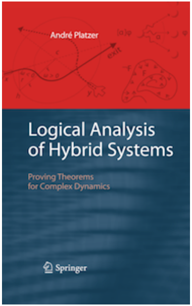
Conclusions

quantified differential dynamic logic

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



- Distributed hybrid systems everywhere
- System model and semantics
- Logic for distributed hybrid systems
- Compositional verification
- First verification approach
- Sound & complete / diff. eqn.
- Simple distributed car control verified



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.