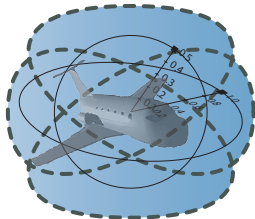


# Hybrid Logical Verification for Hybrid Systems

André Platzer

aplatzer@cs.cmu.edu  
Carnegie Mellon University, Pittsburgh, PA



## 1 Motivation

## 2 Differential Dynamic Logic $d\mathcal{L}$ for Hybrid Systems

- Design Motives
- Syntax
- Semantics

## 3 Compositional Verification using $d\mathcal{L}$

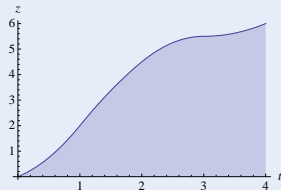
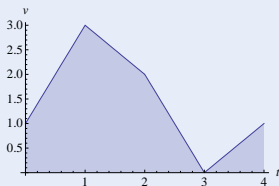
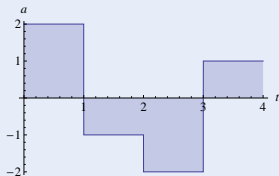
- Compositional Verification Calculus
- Deduction Modulo by Side Deduction
- Soundness and Completeness

## 4 Conclusions

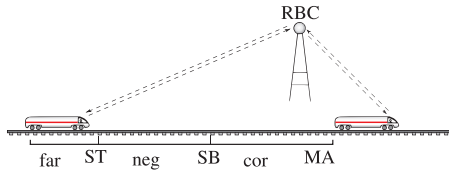
## Challenge

### Hybrid System

- Continuous evolutions (differential equations)
- Discrete jumps (control decisions)

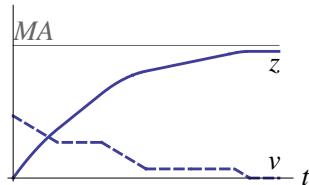


# Verifying Parametric Hybrid Systems

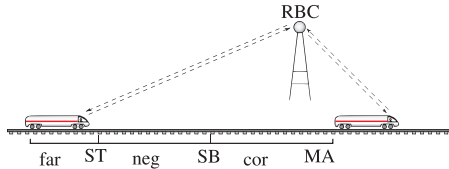


## Parametric Hybrid Systems

continuous evolution along differential equations + discrete change

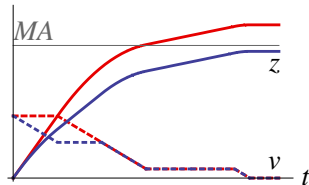


# Verifying Parametric Hybrid Systems

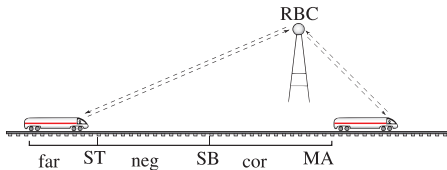


## Parametric Hybrid Systems

continuous evolution along differential equations + discrete change

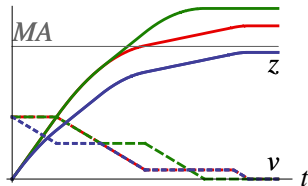


# Verifying Parametric Hybrid Systems

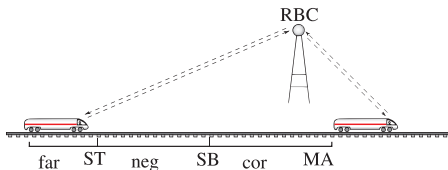


## Parametric Hybrid Systems

continuous evolution along differential equations + discrete change



# Verifying Parametric Hybrid Systems

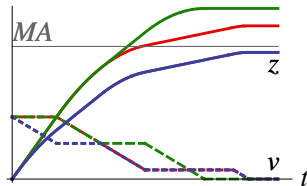


## Parametric Hybrid Systems

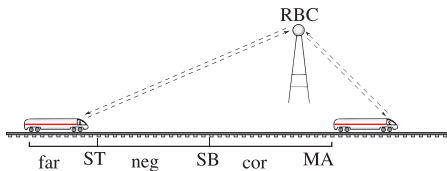
continuous evolution along differential equations + discrete change

- Parameters have nonlinear influence
- Handle  $SB$  as free symbolic parameter?
- Challenge: verification (falsifying is “easy”)
- Which constraints for  $SB$ ?

$\forall MA \exists SB$  “train always safe”

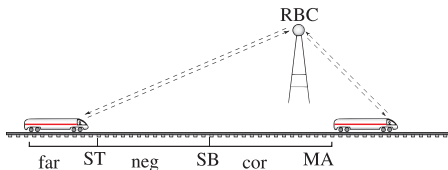


# Verification Approaches for Hybrid Systems



problem	technique	Op	Par	T	Cl	Aut
$ETCS \models z < MA$	TL-MC	✓	✗	✓	✗	✓

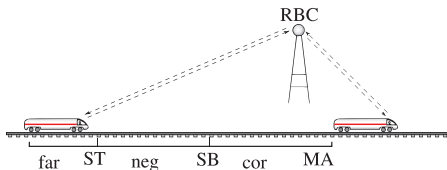
# Verification Approaches for Hybrid Systems



problem	technique	Op	Par	T	Cl	Aut
$ETCS \models z < MA$	TL-MC	✓	✗	✓	✗	✓

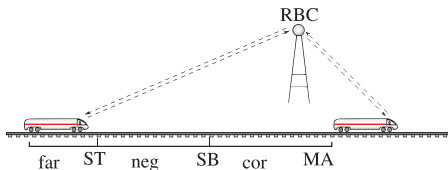
- ✗ no finite-state bisimulation for HS
- ✗ no general handling of free parameters
- ✗ with parameters, everything gets nonlinear!

# Verification Approaches for Hybrid Systems



problem	technique	Op	Par	T	Cl	Aut
$ETCS \models z < MA$	TL-MC	✓	✗	✓	✗	✓
$\models (Ax(ETCS) \rightarrow z < MA)$	TL-calculus	✗	✗	✓	..	✗

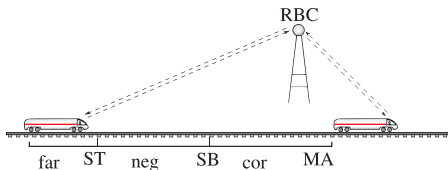
# Verification Approaches for Hybrid Systems



problem	technique	Op	Par	T	Cl	Aut
$ETCS \models z < MA$	TL-MC	✓	✗	✓	✗	✓
$\models (Ax(ETCS) \rightarrow z < MA)$	TL-calculus	✗	✗	✓	..	✗

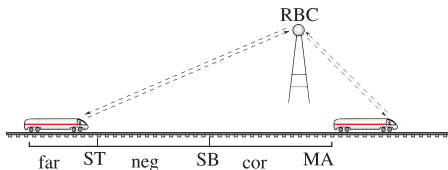
- ✗ declaratively axiomatise operational model
- ✗ expressiveness for characterisation?
- ✗ automation

# Verification Approaches for Hybrid Systems



problem	technique	Op	Par	T	Cl	Aut
$ETCS \models z < MA$	TL-MC	✓	✗	✓	✗	✓
$\models (Ax(ETCS) \rightarrow z < MA)$	TL-calculus	✗	✗	✓	..	✗
$\models [ETCS] z < MA$	DL-calculus	✓	✓	✗	✓	✗

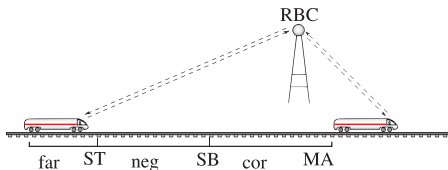
# Verification Approaches for Hybrid Systems



problem	technique	Op	Par	T	Cl	Aut
$ETCS \models z < MA$	TL-MC	✓	✗	✓	✗	✓
$\models (Ax(ETCS) \rightarrow z < MA)$	TL-calculus	✗	✗	✓	..	✗
$\models [ETCS] z < MA$	DL-calculus	✓	✓	✗	✓	✗

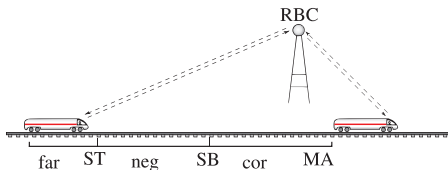
- ✓  $[RBC] \text{partitioned} \rightarrow \exists SB \langle \text{Train} \rangle [RBC] \text{safe}$
- ✗ intermediate states
- ✗ automation

# Verification Approaches for Hybrid Systems



problem	technique	Op	Par	T	Cl	Aut
$ETCS \models z < MA$	TL-MC	✓	✗	✓	✗	✓
$\models (Ax(ETCS) \rightarrow z < MA)$	TL-calculus	✗	✗	✓	..	✗
$\models [ETCS] z < MA$	DL-calculus	✓	✓	✗	✓	✗

# Verification Approaches for Hybrid Systems

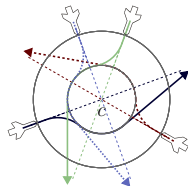
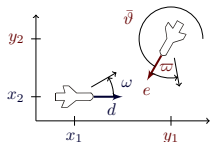
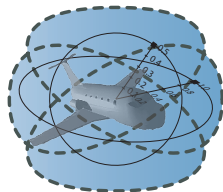
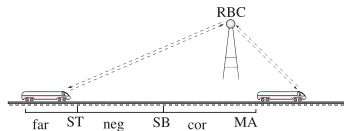


problem	technique	Op	Par	T	Cl	Aut
$ETCS \models z < MA$	TL-MC	✓	✗	✓	✗	✓
$\models (Ax(ETCS) \rightarrow z < MA)$	TL-calculus	✗	✗	✓	..	✗
$\models [ETCS] z < MA$	DL-calculus	✓	✓	✗	✓	?

differential dynamic logic

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

# Hybrid Systems Analysis is Important for ...

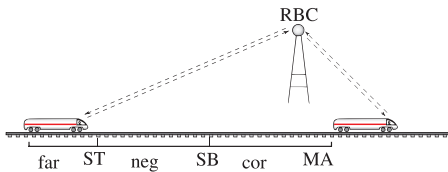


- 1 Motivation
- 2 Differential Dynamic Logic  $d\mathcal{L}$  for Hybrid Systems
  - Design Motives
  - Syntax
  - Semantics
- 3 Compositional Verification using  $d\mathcal{L}$ 
  - Compositional Verification Calculus
  - Deduction Modulo by Side Deduction
  - Soundness and Completeness
- 4 Conclusions

- 1 Motivation
- 2 Differential Dynamic Logic  $d\mathcal{L}$  for Hybrid Systems
  - Design Motives
  - Syntax
  - Semantics
- 3 Compositional Verification using  $d\mathcal{L}$ 
  - Compositional Verification Calculus
  - Deduction Modulo by Side Deduction
  - Soundness and Completeness
- 4 Conclusions

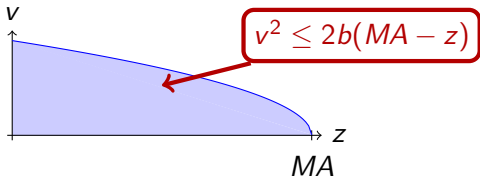
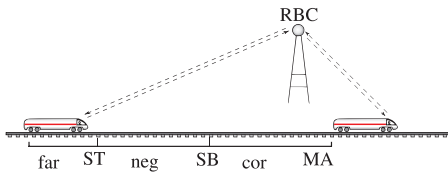
differential dynamic logic

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



differential dynamic logic

$$d\mathcal{L} = \text{FOL}_{\mathbb{R}}$$

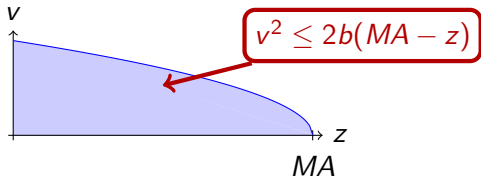
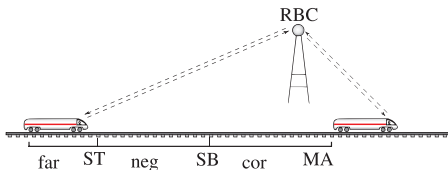


differential dynamic logic

$$d\mathcal{L} = \text{FOL}_{\mathbb{R}}$$

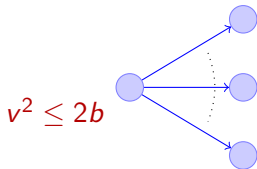
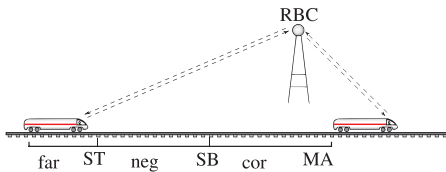
$$\forall MA \exists SB \dots$$

$$\forall t \geq 0 \dots$$



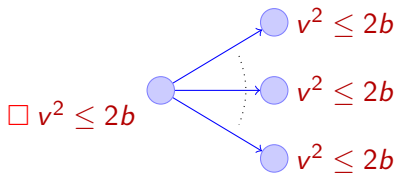
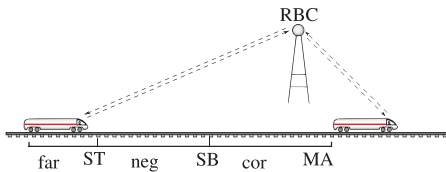
differential dynamic logic

$$d\mathcal{L} = \text{FOL}_{\mathbb{R}} +$$



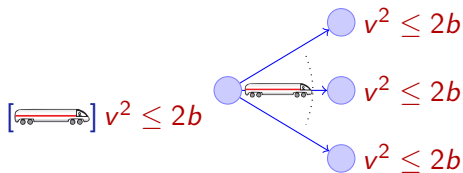
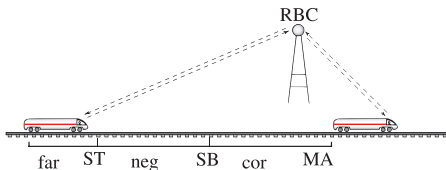
differential dynamic logic

$$d\mathcal{L} = \text{FOL}_{\mathbb{R}} + \text{ML}$$



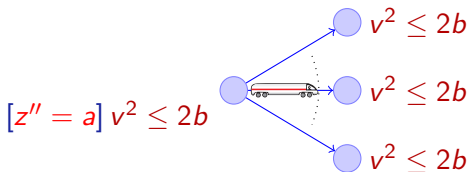
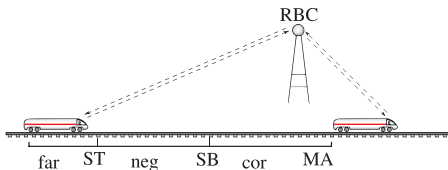
differential dynamic logic

$$d\mathcal{L} = \text{FOL}_{\mathbb{R}} + \text{DL}$$



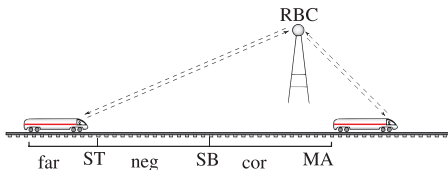
differential dynamic logic

$$d\mathcal{L} = \text{FOL}_{\mathbb{R}} + \text{DL} + \text{HP}$$

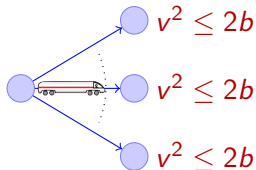


differential dynamic logic

$$d\mathcal{L} = \text{FOL}_{\mathbb{R}} + \text{DL} + \text{HP}$$

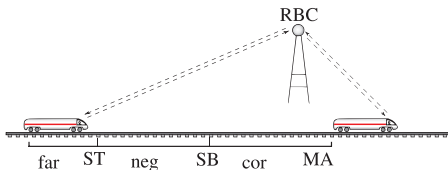


$[\text{if}(z > SB) a := -b; z'' = a] v^2 \leq 2b$

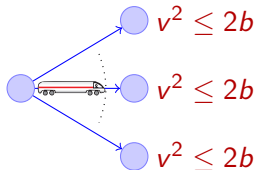


differential dynamic logic

$$d\mathcal{L} = \text{FOL}_{\mathbb{R}} + \text{DL} + \text{HP}$$



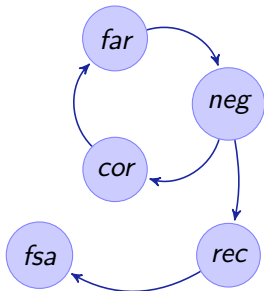
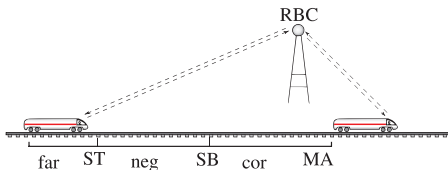
$$\underbrace{[\text{if}(z > SB) a := -b; z'' = a]}_{\text{hybrid program}} v^2 \leq 2b$$



# $\mathcal{A}$ dL Motives: What about Hybrid Automata?

differential dynamic logic

$$d\mathcal{L} = \text{FOL}_{\mathbb{R}} + \text{DL} + \text{HP}$$

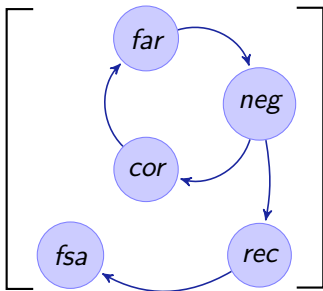
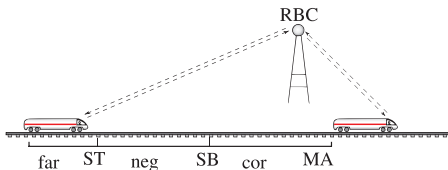


How about hybrid automata?

# $\mathcal{A}$ dL Motives: What about Hybrid Automata?

differential dynamic logic

$$d\mathcal{L} = \text{FOL}_{\mathbb{R}} + \text{DL} + \text{HP}$$

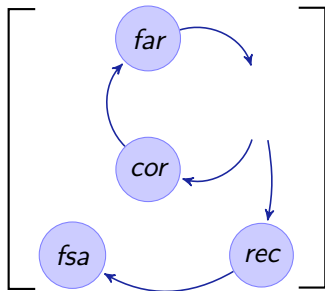
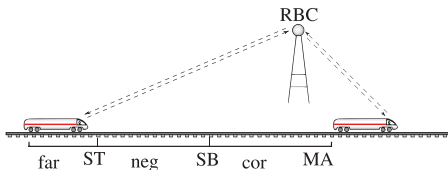


$$v^2 \leq 2b \dots$$

# $\mathcal{A}$ dL Motives: What about Hybrid Automata?

differential dynamic logic

$$d\mathcal{L} = \text{FOL}_{\mathbb{R}} + \text{DL} + \text{HP}$$

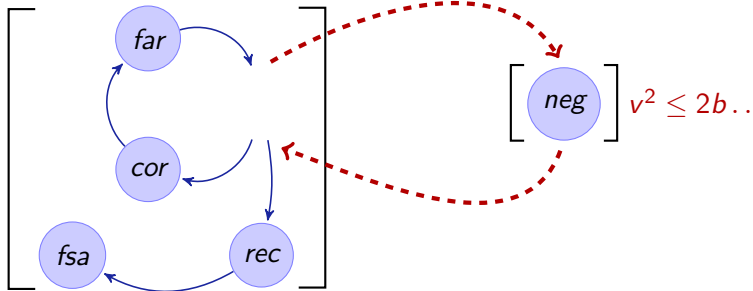
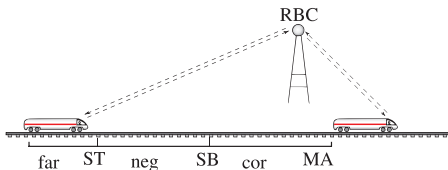


$$\left[ \text{neg} \right] v^2 \leq 2b..$$

# $\mathcal{A}$ dL Motives: What about Hybrid Automata?

differential dynamic logic

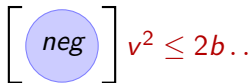
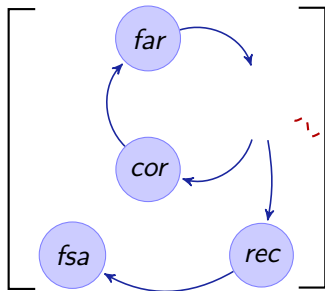
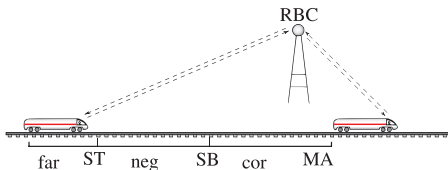
$$d\mathcal{L} = \text{FOL}_{\mathbb{R}} + \text{DL} + \text{HP}$$



# $\mathcal{A}$ dL Motives: What about Hybrid Automata?

differential dynamic logic

$$d\mathcal{L} = \text{FOL}_{\mathbb{R}} + \text{DL} + \text{HP}$$



not compositional

## Definition (Hybrid program $\alpha$ )

$x' = f(x)$	(continuous evolution)	
$x := f(x)$	(discrete jump)	} jump & test
$? \chi$	(conditional execution)	
$\alpha; \beta$	(seq. composition)	} Kleene algebra
$\alpha \cup \beta$	(nondet. choice)	
$\alpha^*$	(nondet. repetition)	

## Definition (Hybrid program $\alpha$ )

$x' = f(x)$	(continuous evolution)	} jump & test
$x := f(x)$	(discrete jump)	
$? \chi$	(conditional execution)	
$\alpha; \beta$	(seq. composition)	} Kleene algebra
$\alpha \cup \beta$	(nondet. choice)	
$\alpha^*$	(nondet. repetition)	

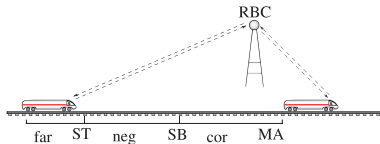
$$ETCS \equiv (ctrl; drive)^*$$

$$ctrl \equiv (?MA - z \leq SB; a := -b)$$

$$\cup (?MA - z \geq SB; a := \dots)$$

$$drive \equiv \quad \quad \quad z'' = a$$

$$\wedge v \geq 0 \wedge \tau \leq \varepsilon$$



## Definition (Hybrid program $\alpha$ )

$x' = f(x)$	(continuous evolution)	} jump & test
$x := f(x)$	(discrete jump)	
$? \chi$	(conditional execution)	
$\alpha; \beta$	(seq. composition)	} Kleene algebra
$\alpha \cup \beta$	(nondet. choice)	
$\alpha^*$	(nondet. repetition)	

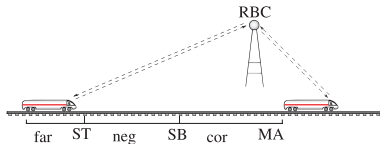
$ETCS \equiv (ctrl; drive)^*$

$ctrl \equiv (?MA - z \leq SB; a := -b)$

$\cup (?MA - z \geq SB; a := \dots)$

$drive \equiv \tau := 0; z' = v, v' = a, \tau' = 1$

$\wedge v \geq 0 \wedge \tau \leq \varepsilon$



## Definition (Hybrid program $\alpha$ )

$x' = f(x) \wedge \chi$	(continuous evolution)	} jump & test
$x := f(x)$	(discrete jump)	
$? \chi$	(conditional execution)	
$\alpha; \beta$	(seq. composition)	} Kleene algebra
$\alpha \cup \beta$	(nondet. choice)	
$\alpha^*$	(nondet. repetition)	

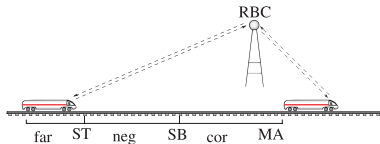
$$ETCS \equiv (ctrl; drive)^*$$

$$ctrl \equiv (?MA - z \leq SB; a := -b)$$

$$\cup (?MA - z \geq SB; a := \dots)$$

$$drive \equiv \tau := 0; z' = v, v' = a, \tau' = 1$$

$$\wedge v \geq 0 \wedge \tau \leq \varepsilon$$



## Definition (Formulas $\phi$ )

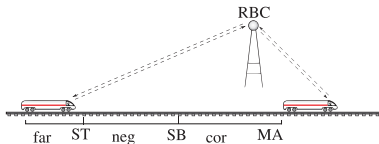
$\neg, \wedge, \vee, \rightarrow, \forall x, \exists x, =, \leq, +, \cdot$     ( $\mathbb{R}$ -first-order part)  
 $[\alpha]\phi, \langle \alpha \rangle \phi$     (dynamic part)

$SB \geq \dots \rightarrow [(ctrl; drive)^*] z \leq MA$

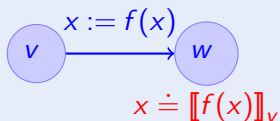
All trains respect  $MA$

$RBC$  partitions  $MA$

$\Rightarrow$  system collision free



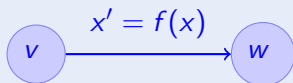
Definition (Hybrid programs  $\alpha$ : transition semantics)



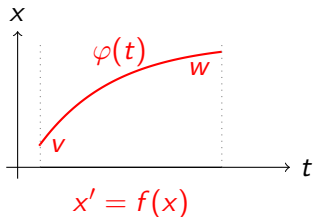
► Details



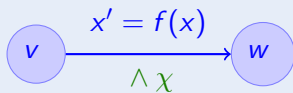
Definition (Hybrid programs  $\alpha$ : transition semantics)



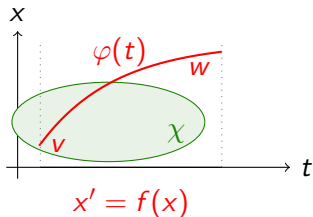
► Details



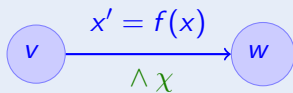
Definition (Hybrid programs  $\alpha$ : transition semantics)



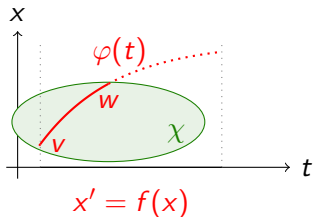
► Details



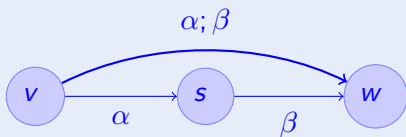
Definition (Hybrid programs  $\alpha$ : transition semantics)



► Details



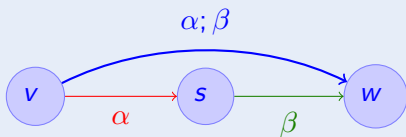
Definition (Hybrid programs  $\alpha$ : transition semantics)



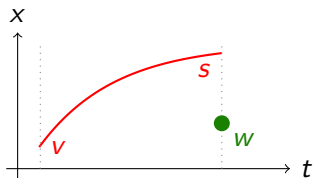
► Details



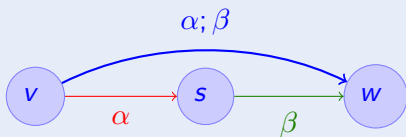
Definition (Hybrid programs  $\alpha; \beta$ : transition semantics)



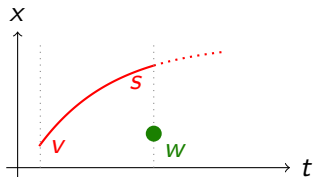
► Details



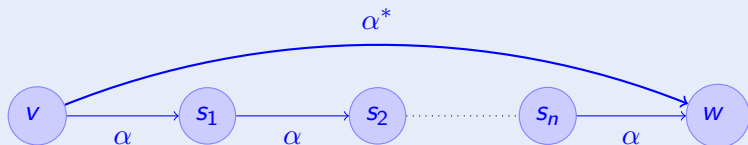
Definition (Hybrid programs  $\alpha$ : transition semantics)



► Details



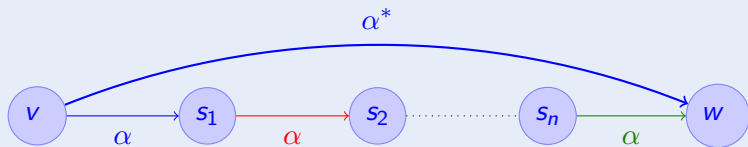
Definition (Hybrid programs  $\alpha$ : transition semantics)



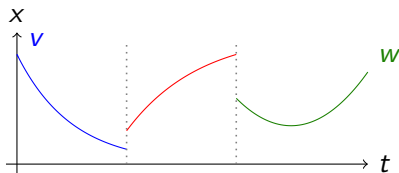
► Details



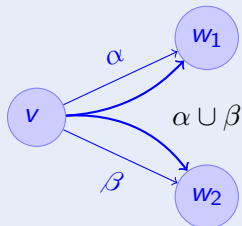
Definition (Hybrid programs  $\alpha$ : transition semantics)



► Details



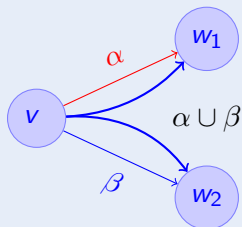
Definition (Hybrid programs  $\alpha$ : transition semantics)



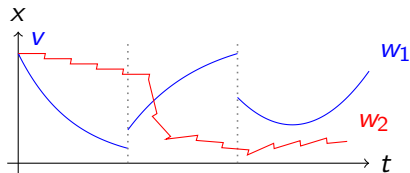
► Details



Definition (Hybrid programs  $\alpha$ : transition semantics)



► Details



Definition (Hybrid programs  $\alpha$ : transition semantics)



if  $v \models \chi$

► Details



## Definition (Hybrid programs $\alpha$ : transition semantics)

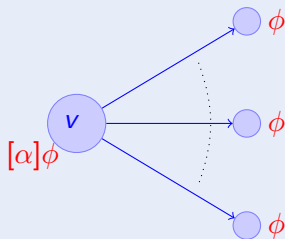


if  $v \neq \chi$

► Details



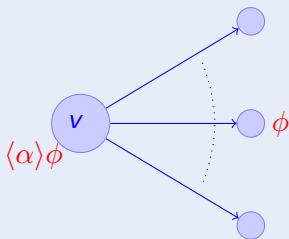
## Definition (Formulas $\phi$ )



► Details



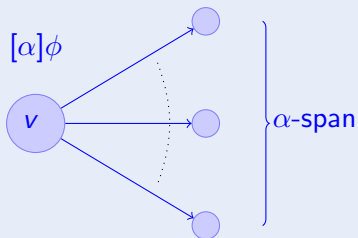
## Definition (Formulas $\phi$ )



► Details



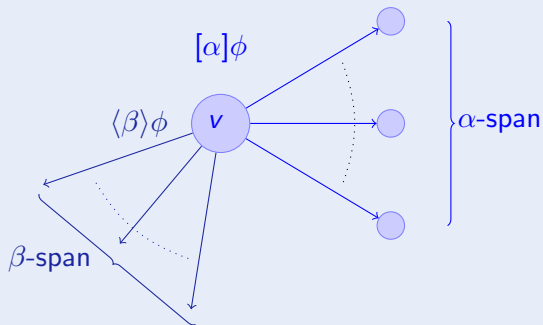
## Definition (Formulas $\phi$ )



► Details



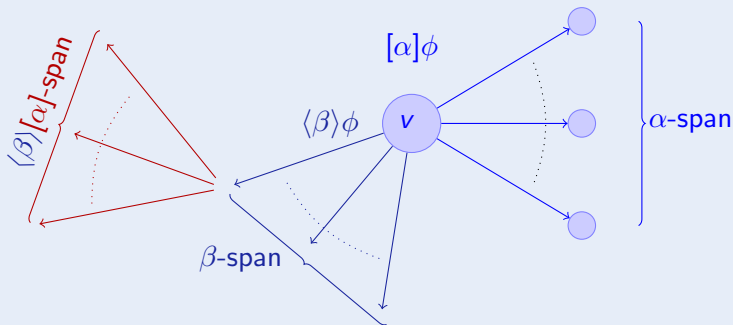
## Definition (Formulas $\phi$ )



► Details



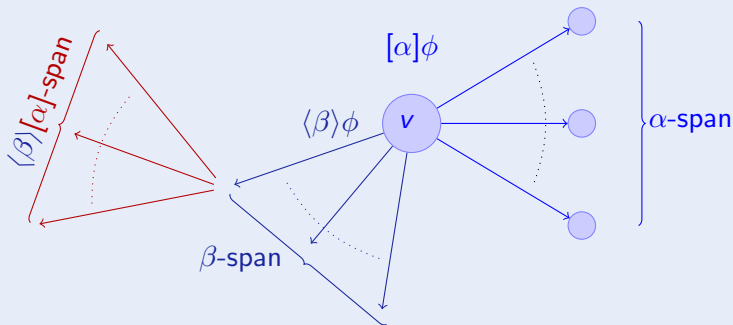
## Definition (Formulas $\phi$ )



► Details



## Definition (Formulas $\phi$ )



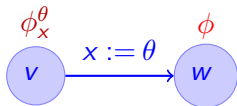
► Details



compositional semantics  $\Rightarrow$  compositional calculus!

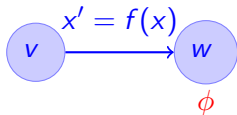
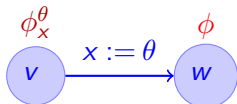
- 1 Motivation
- 2 Differential Dynamic Logic  $d\mathcal{L}$  for Hybrid Systems
  - Design Motives
  - Syntax
  - Semantics
- 3 Compositional Verification using  $d\mathcal{L}$ 
  - Compositional Verification Calculus
  - Deduction Modulo by Side Deduction
  - Soundness and Completeness
- 4 Conclusions

$$\frac{\phi_x^\theta}{[x := \theta]\phi}$$



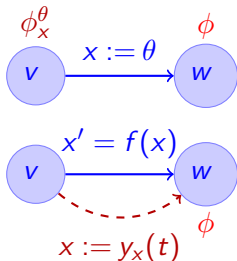
$$\frac{\phi_x^\theta}{[x := \theta]\phi}$$

$$\frac{\exists t \geq 0 \langle x := y_x(t) \rangle \phi}{\langle x' = f(x) \rangle \phi}$$



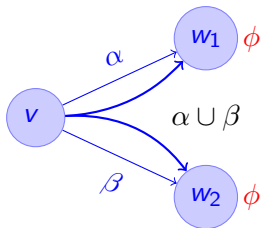
$$\frac{\phi_x^\theta}{[x := \theta]\phi}$$

$$\frac{\exists t \geq 0 \langle x := y_x(t) \rangle \phi}{\langle x' = f(x) \rangle \phi}$$

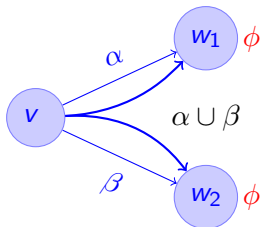


compositional semantics  $\Rightarrow$  compositional rules!

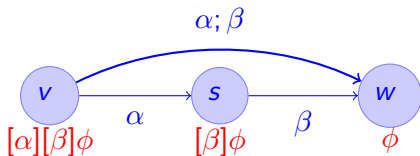
$$\frac{[\alpha]\phi \wedge [\beta]\phi}{[\alpha \cup \beta]\phi}$$



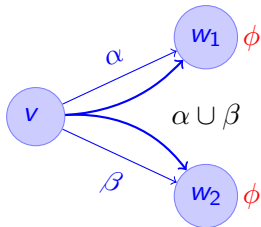
$$\frac{[\alpha]\phi \wedge [\beta]\phi}{[\alpha \cup \beta]\phi}$$



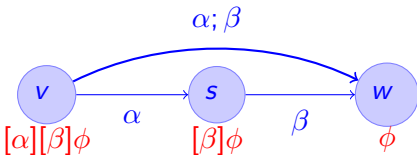
$$\frac{[\alpha][\beta]\phi}{[\alpha; \beta]\phi}$$



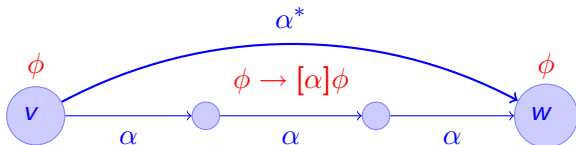
$$\frac{[\alpha]\phi \wedge [\beta]\phi}{[\alpha \cup \beta]\phi}$$

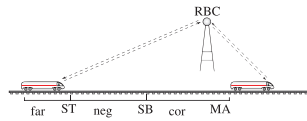


$$\frac{[\alpha][\beta]\phi}{[\alpha; \beta]\phi}$$



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



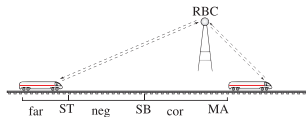


---

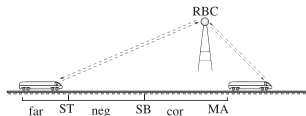
---

---

$$\vdash v \geq 0 \wedge z < MA \rightarrow \langle z' = v, v' = -b \rangle z > MA$$



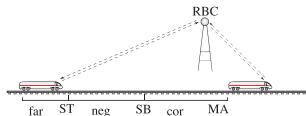
$$\frac{\frac{v \geq 0, z < MA \vdash \exists t \geq 0 \langle z := -\frac{b}{2}t^2 + vt + z \rangle z > MA}{v \geq 0, z < MA \vdash \langle z' = v, v' = -b \rangle z > MA}}{\vdash v \geq 0 \wedge z < MA \rightarrow \langle z' = v, v' = -b \rangle z > MA}$$



Collins/Tarski QE not applicable!

$$\frac{\frac{v \geq 0, z < MA \vdash \exists t \geq 0 \langle z := -\frac{b}{2}t^2 + vt + z \rangle z > MA}{v \geq 0, z < MA \vdash \langle z' = v, v' = -b \rangle z > MA}}{\vdash v \geq 0 \wedge z < MA \rightarrow \langle z' = v, v' = -b \rangle z > MA}$$

# $\mathcal{A}$ Deduction Modulo (Side Deduction)



$$\frac{}{v \geq 0, z < MA \vdash t \geq 0 \wedge \langle z := -\frac{b}{2}t^2 + vt + z \rangle z > MA}$$

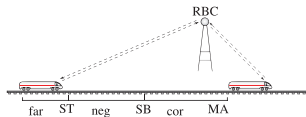
$$\frac{}{v \geq 0, z < MA \vdash \exists t \geq 0 \langle z := -\frac{b}{2}t^2 + vt + z \rangle z > MA}$$

$$\frac{}{v \geq 0, z < MA \vdash \langle z' = v, v' = -b \rangle z > MA}$$

$$\vdash v \geq 0 \wedge z < MA \rightarrow \langle z' = v, v' = -b \rangle z > MA$$

start  
side

# $\mathcal{A}$ Deduction Modulo (Side Deduction)

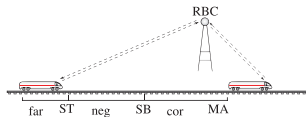


$$\frac{v \geq 0, z < MA \vdash t \geq 0 \quad \frac{v \geq 0, z < MA \vdash -\frac{b}{2}t^2 + vt + z > MA}{v \geq 0, z < MA \vdash \langle z := -\frac{b}{2}t^2 + vt + z \rangle z > MA}}{v \geq 0, z < MA \vdash t \geq 0 \wedge \langle z := -\frac{b}{2}t^2 + vt + z \rangle z > MA}$$

$$\frac{\frac{v \geq 0, z < MA \vdash \exists t \geq 0 \langle z := -\frac{b}{2}t^2 + vt + z \rangle z > MA}{v \geq 0, z < MA \vdash \langle z' = v, v' = -b \rangle z > MA}}{\vdash v \geq 0 \wedge z < MA \rightarrow \langle z' = v, v' = -b \rangle z > MA}$$

start  
side

# $\mathcal{A}$ Deduction Modulo (Side Deduction)



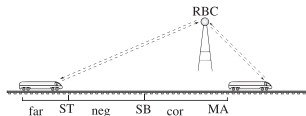
$$\text{QE} \frac{v \geq 0, z < MA \vdash t \geq 0 \quad \frac{v \geq 0, z < MA \vdash -\frac{b}{2}t^2 + vt + z > MA}{v \geq 0, z < MA \vdash \langle z := -\frac{b}{2}t^2 + vt + z \rangle z > MA}}{v \geq 0, z < MA \vdash t \geq 0 \wedge \langle z := -\frac{b}{2}t^2 + vt + z \rangle z > MA}$$

$$\frac{v \geq 0, z < MA \vdash \text{QE}(\exists t (\dots t \geq 0 \wedge -\frac{b}{2}t^2 + vt + z > MA))}{v \geq 0, z < MA \vdash \exists t \geq 0 \langle z := -\frac{b}{2}t^2 + vt + z \rangle z > MA}$$

$$\frac{v \geq 0, z < MA \vdash \langle z' = v, v' = -b \rangle z > MA}{\vdash v \geq 0 \wedge z < MA \rightarrow \langle z' = v, v' = -b \rangle z > MA}$$

start  
side

# $\mathcal{A}$ Deduction Modulo (Side Deduction)



$$\frac{v \geq 0, z < MA \vdash t \geq 0 \quad \frac{v \geq 0, z < MA \vdash -\frac{b}{2}t^2 + vt + z > MA}{v \geq 0, z < MA \vdash \langle z := -\frac{b}{2}t^2 + vt + z \rangle z > MA}}{v \geq 0, z < MA \vdash t \geq 0 \wedge \langle z := -\frac{b}{2}t^2 + vt + z \rangle z > MA}$$

$$v \geq 0, z < MA \vdash v^2 > 2b(MA - z)$$

$$\frac{v \geq 0, z < MA \vdash v^2 > 2b(MA - z)}{v \geq 0, z < MA \vdash \exists t \geq 0 \langle z := -\frac{b}{2}t^2 + vt + z \rangle z > MA}$$

$$\frac{v \geq 0, z < MA \vdash \exists t \geq 0 \langle z := -\frac{b}{2}t^2 + vt + z \rangle z > MA}{v \geq 0, z < MA \vdash \langle z' = v, v' = -b \rangle z > MA}$$

$$\vdash v \geq 0 \wedge z < MA \rightarrow \langle z' = v, v' = -b \rangle z > MA$$

start  
side

## Theorem (Relative Completeness)

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

▶ [Proof Outline 15p](#)

## Theorem (Relative Completeness)

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

▶ [Proof Outline 15p](#)

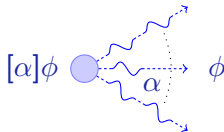
## Corollary (Proof-theoretical Alignment)

verification of hybrid systems = verification of dynamical systems!

- 1 Motivation
- 2 Differential Dynamic Logic  $d\mathcal{L}$  for Hybrid Systems
  - Design Motives
  - Syntax
  - Semantics
- 3 Compositional Verification using  $d\mathcal{L}$ 
  - Compositional Verification Calculus
  - Deduction Modulo by Side Deduction
  - Soundness and Completeness
- 4 Conclusions

differential dynamic logic

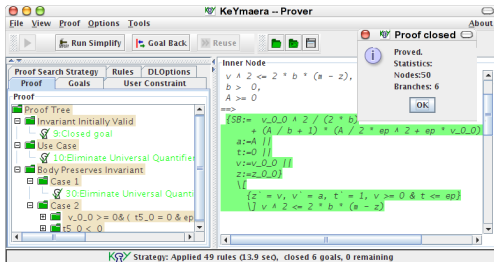
$$d\mathcal{L} = DL + HP$$





Verifying parametric hybrid systems:


- Logics for hybrid systems
- Compositional calculi
- $\mathbb{R}$ -Skolem Tree for automation
- Sound & complete / diff. eqn.
- Differential invariants
- Verification algorithms
- Challenging case studies


KeYmaera




 André Platzer.  
Differential dynamic logic for hybrid systems.  
*J. Autom. Reas.*, 41(2):143–189, 2008.

 André Platzer.  
Differential-algebraic dynamic logic for differential-algebraic programs.  
*J. Log. Comput.*, 2008.

 André Platzer and Edmund M. Clarke.  
Computing differential invariants of hybrid systems as fixedpoints.  
*Form. Methods Syst. Des.*, 35(1):98–120, 2009. Special CAV'08 issue.

 André Platzer and Jan-David Quesel.  
KeYmaera: A hybrid theorem prover for hybrid systems.  
In Alessandro Armando, Peter Baumgartner, and Gilles Dowek,  
editors, *IJCAR*, volume 5195 of *LNCS*, pages 171–178. Springer, 2008.

 André Platzer and Edmund M. Clarke.  
The image computation problem in hybrid systems model checking.  
In A. Bemporad, A. Bicchi, and G. Buttazzo, editors, *HSCC*, volume  
4416 of *LNCS*, pages 473–486. Springer, 2007.