# VerSiS: Verification and Simulation of embedded hybrid Systems

Alexander Dreyer, Raffaella Gentilini, Martin Proetzsch

Adaptive Systems
Fraunhofer ITWM

Reactive Systems Group
Dep. of Computer Science
University of Kaiserslautern

Robotics Research Lab
Dep. of Computer Science
University of Kaiserslautern

February 28th 2007

# Outline

# Outline

# Overview

## Heterogenous embedded hybrid systems

- Digital hardware and software components
- Analog sensor and actuator components

## Goals

- Extension of formal verification methods to analog components
- Modeling, simulation, and automated approximation of multi-physical systems
- Digital verification with algebraic methods
- Integration in work flow for design and analysis of behavior-based robotic systems as sample application

# Overview

## Heterogenous embedded hybrid systems

- Digital hardware and software components
- Analog sensor and actuator components

## Goals

- Extension of formal verification methods to analog components
- Modeling, simulation, and automated approximation of multi-physical systems
- Digital verification with algebraic methods
- Integration in work flow for design and analysis of behavior-based robotic systems as sample application

# Scientific Personnel and Experience

## Adaptive Systems (Fraunhofer ITWM)

- Dr. Patrick Lang
- Dr. Alexander Dreyer
- Dr. Jochen Broz (sponsored)

- Analog Insydes: Tools for symbolic/numeric analysis of analog circuits, based on Differential-Algebraic Equations (DAEs)
- Modeling and simulation of mechatronic systems
- Symbolic approximation methods and tolerance analysis

# Scientific Personell and Experience (cont'd)

## Reactive Systems Group (Computer Science)

- Prof. Klaus Schneider (chair)
- Dr. Raffaella Gentilini (sponsored)
- Jens Brandt (sponsored)

- Averest Tool: specification, verification, and implementation of reactive systems
- Model checking, automated theorem proving, and formal methods for systems' analysis and verification
- Temporal logics and automata theory

# Scientific Personell and Experience (cont'd)

## Robotics Research Lab (Computer Science)

- Prof. Karsten Berns
- Martin Proetzsch (sponsored)

- Realization of autonomous mobile robots, like wheel-based vehicles (indoor, outdoor), climbing robots, as well as humanoid assistance systems
- Modular Controller Architecture (MCA) for implementing complex systems featuring real-time analysis and state monitoring
- Biologically motivated behavior-based system design approach

# Scientific Personell and Experience (cont'd)

## Algebra, Geometry and Computer Algebra (Mathematics)

- Prof. Gert-Martin Greuel (chair)
- Oliver Wienand

- SINGULAR: Fast Gröbner system for polynomial systems
- POLYBORI: Gröbner approaches for Boolean equation systems for digital verification (DFG project together with ITWM-AS and MFO)

# Additional Information

Project start: 1.10.2005

## External Cooperations

- Michael Brickenstein
  (Mathematisches Forschungsinstitut Oberwolfach *MFO*)
- Dr.-Ing. Markus Wedler, Evgeny Pavlenko (EIT, TU KL)

# Outline

# Development of Robot Control Systems


RA

Robust Autonomous Vehicle for Outdoor Navigation

- Complexity of solving tasks especially in outdoor robotics
- Approach: decompose control system into *behaviors* inspired by observations in biology
- Problem: assure certain safety critical properties of the system

⇒ Need for analysis methods for behavior-based networks

# Development of Robot Control Systems

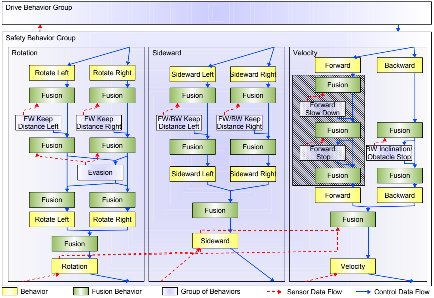

RA

Robust Autonomous Vehicle for Outdoor Navigation

- Complexity of solving tasks especially in outdoor robotics
- Approach: decompose control system into *behaviors* inspired by observations in biology
- Problem: assure certain safety critical properties of the system

$\Rightarrow$ Need for analysis methods for behavior-based networks

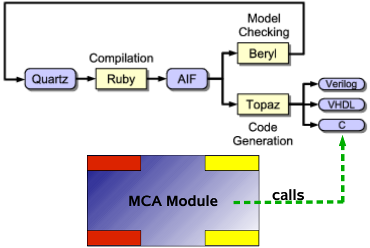
# Safety Behavior Network for RAVON




## Control Structure

- MCA: Modular Controller Architecture
- Behaviors for assuring vehicle integrity, uniform interaction
- Slope observation, obstacle proximity (several sensors)
- Choose behaviors influencing vehicle velocity for verification

# Verification of Safety Behaviors: Tools




## Averest

- Tools for verifying temporal properties of synchronous systems
- Quartz compiler: `ruby`
- Symbolic model checker: `beryl`
- Generate code in software and hardware: `topaz`

## Interaction with MCA

- MCA module calls provided C-methods

# Verification: Specifications and Performance

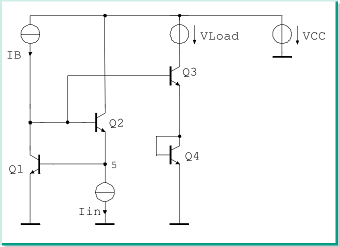
- Verified 15 specifications concerning safety properties of the vehicle
- Examples:

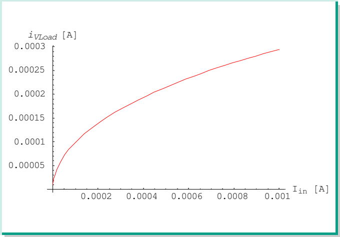
```
A G (co_velocity <= ci_velocity);
A G ((si_camera_distance >= MAX_VEL_OBSTACLE_DIST)
   & (si_scanner_distance >= MAX_VEL_OBSTACLE_DIST)
   & (si_roll <= MAX_VELOCITY_INCLINATION)
   & (si_pitch <= MAX_VELOCITY_INCLINATION)
   -> co_velocity == ci_velocity);
A G (si_roll >= STOP_INCLINATION
 -> co_velocity == 0u);
```

- Uniform modular structure of behavior-based architecture proven favorable for verification
- Verification runtime: 8.85 s for 8 bits

# Outline

# Symbolic Approximation using Simulations



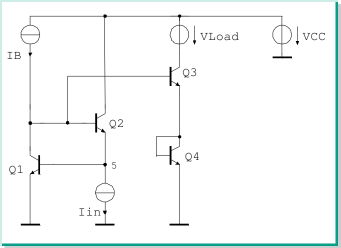



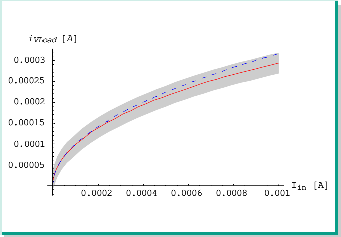

## Symbolic equations

- 27 equations
- 17 parameters

$$i_{CS,Q2} + i_{VDC} + i_{VLoad} = -I_B$$
$$i_{BS,Q2} + i_{BS,Q3} + i_{CS,Q1} = I_B$$
$$i_{BS,Q4} + i_{CS,Q4} + i_{ES,Q3} = 0$$
$$i_{BS,Q1} + i_{ES,Q2} = -I_{in}$$
$$i_{CS,Q3} - i_{VLoad} = 0$$
$$-i_{BS,Q1} - i_{CS,Q1} - i_{ES,Q1} = 0$$
$$i_{BS,Q1} = ib_{Q1}$$
$$-i_{BS,Q1} - i_{ES,Q1} = ic_{Q1}$$
$$ic_{Q1} = v_5 G_{min} + \left(-1 + e^{\frac{v_5}{Vt}}\right) Is_{Q1} - \left(1 + \frac{1}{Br_{Q1}}\right) \left((-v_3 + v_5) G_{min} + \left(-1 + e^{\frac{-v_3+v_5}{Vt}}\right) Is_{Q1}\right)$$
$$ib_{Q1} = \frac{v_5 G_{min} + \left(-1 + e^{\frac{v_5}{Vt}}\right) Is_{Q1}}{Bf_{Q1}} + \frac{(-v_3+v_5) G_{min} + \left(-1 + e^{\frac{-v_3+v_5}{Vt}}\right) Is_{Q1}}{Br_{Q1}}$$
$$-i_{BS,Q2} - i_{CS,Q2} - i_{ES,Q2} = 0$$
$$i_{BS,Q2} = ib_{Q2}$$
$$-i_{BS,Q2} - i_{ES,Q2} = ic_{Q2}$$
$$ic_{Q2} = (v_3 - v_5) G_{min} + \left(-1 + e^{\frac{v_3-v_5}{Vt}}\right) Is_{Q2} - \left(1 + \frac{1}{Br_{Q2}}\right) \left((-v_1 + v_3) G_{min} + \left(-1 + e^{\frac{-v_1+v_3}{Vt}}\right) Is_{Q2}\right)$$
$$ib_{Q2} = \frac{(-v_1+v_3) G_{min} + \left(-1 + e^{\frac{-v_1+v_3}{Vt}}\right) Is_{Q2}}{Br_{Q2}} + \frac{(v_3-v_5) G_{min} + \left(-1 + e^{\frac{v_3-v_5}{Vt}}\right) Is_{Q2}}{Bf_{Q2}}$$
$$-i_{BS,Q3} - i_{CS,Q3} - i_{ES,Q3} = 0$$
$$i_{BS,Q3} = ib_{Q3}$$
$$-i_{BS,Q3} - i_{ES,Q3} = ic_{Q3}$$
$$ic_{Q3} = (v_3 - v_4) G_{min} + \left(-1 + e^{\frac{v_3-v_4}{Vt}}\right) Is_{Q3} - \left(1 + \frac{1}{Br_{Q3}}\right) \left((v_3 - v_{out}) G_{min} + \left(-1 + e^{\frac{v_3-v_{out}}{Vt}}\right) Is_{Q3}\right)$$
$$ib_{Q3} = \frac{(v_3-v_4) G_{min} + \left(-1 + e^{\frac{v_3-v_4}{Vt}}\right) Is_{Q3}}{Bf_{Q3}} + \frac{(v_3-v_{out}) G_{min} + \left(-1 + e^{\frac{v_3-v_{out}}{Vt}}\right) Is_{Q3}}{Br_{Q3}}$$
$$-i_{BS,Q4} - i_{CS,Q4} - i_{ES,Q4} = 0$$
$$i_{BS,Q4} = ib_{Q4}$$
$$-i_{BS,Q4} - i_{ES,Q4} = ic_{Q4}$$
$$ic_{Q4} = v_4 G_{min} + \left(-1 + e^{\frac{v_4}{Vt}}\right) Is_{Q4}$$
$$ib_{Q4} = \frac{v_4 G_{min} + \left(-1 + e^{\frac{v_4}{Vt}}\right) Is_{Q4}}{Bf_{Q4}}$$
$$v_1 - v_{out} = V_{Load}$$
$$v_1 = V_{DC}$$

# Symbolic Approximation using Simulations




## Symbolic equations

- 27 equations
- 17 parameters

## Reduced equations

- 4 equations
- 7 parameters

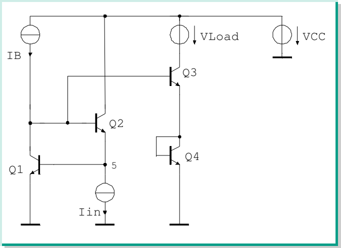
$$-I_{in} + e^{\frac{v_3 - v_5}{Vt}} \, Is_{Q2} = 0$$

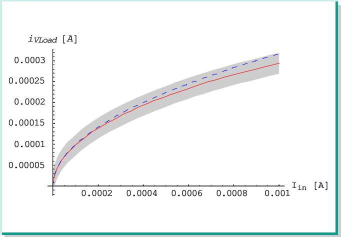
$$I_B - e^{\frac{v_5}{Vt}} \, Is_{Q1} = 0$$

$$i_{VLoad} - e^{\frac{v_3 - v_4}{Vt}} \, Is_{Q3} = 0$$

$$i_{VLoad} - e^{\frac{v_4}{Vt}} \, Is_{Q4} = 0$$

# Symbolic Approximation using Simulations




## Symbolic equations

- 27 equations
- 17 parameters

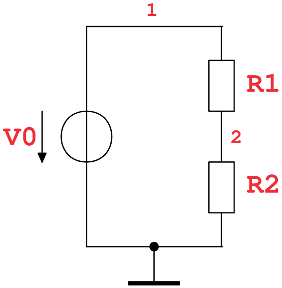
## Reduced equations

- 4 equations
- 7 parameters

## Explicit formula

$$i_{vload}^{simple} = \sqrt{\frac{Is_{Q3}\, Is_{Q4}}{Is_{Q1}\, Is_{Q2}}}\sqrt{I_B}\sqrt{I_{in}}$$

# Parameter Tolerances using Interval-arithmetic

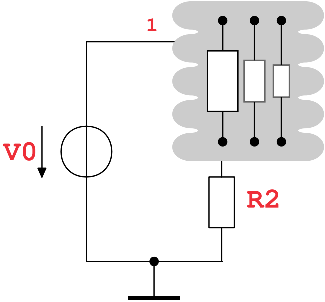



- Simulations ensure validity for discrete settings only

**V0** $= 1\,\mathrm{V}$
**R1** $= 10\Omega$
**R2** $= 100\Omega$

$$V_2 = \frac{V_0}{R_1/R_2+1} \approx 0.909\,\mathrm{V}$$

# Parameter Tolerances using Interval-arithmetic




- Simulations ensure validity for discrete settings only
- Interval-valued computations can verify properties for domains

$$\texttt{V0} = 1\,\mathrm{V} \pm 10\%$$
$$\texttt{R1} = 10\Omega \pm 10\%$$
$$\texttt{R2} = 100\Omega \pm 10\%$$

$$V_2 = \frac{V_0}{R_1/R_2+1} \approx 0.909\,\mathrm{V}$$
$$V_2 \in \frac{[0.9,1.1]}{[9,11]/[90,110]+1} = [0.80, 1.02]$$

# Validation of Symbolic Approximations

## Compare different accuracy levels

$$i_{vload} \approx i_{vload}^{simple} \ ?$$

$$i_{vload} = \frac{\sqrt{Is_{Q3}\, Is_{Q4}\, ((0.025\, Is_{Q3}\, Is_{Q4} - Is_{Q2}\, Is_{Q1})10^{-2}\, I_{in}{}^{2} + Is_{Q1}\, Is_{Q2}\, I_B\, I_{in})} - 5\cdot 10^{-3}\, I_{in}\, Is_{Q3}\, Is_{Q4}}{Is_{Q1}\, Is_{Q2}}$$

$$i_{vload}^{simple} = \sqrt{\frac{Is_{Q3}\, Is_{Q4}}{Is_{Q1}\, Is_{Q2}}}\, \sqrt{I_B}\, \sqrt{I_{in}}$$

## Example

For $Is_{Qi} = 10^{-16}\mathrm{A}$, $I_{in}/1\mathrm{mA} \in [0.2, 0.3]$, and $I_B/1\mathrm{mA} \in [0.1, 0.2]$ interval-techniques **prove** $|i_{vload} - i_{vload}^{simple}| < 2.5\%$

# Validation of Symbolic Approximations

## Compare different accuracy levels

$$i_{vload} \approx i_{vload}^{simple} \ ?$$

$$i_{vload} = \frac{\sqrt{Is_{Q3}\, Is_{Q4}\, ((0.025 Is_{Q3}\, Is_{Q4} - Is_{Q2}\, Is_{Q1})10^{-2} I_{in}{}^{2} + Is_{Q1}\, Is_{Q2}\, I_B\, I_{in})} - 5 \cdot 10^{-3} I_{in}\, Is_{Q3}\, Is_{Q4}}{Is_{Q1}\, Is_{Q2}}$$

$$i_{vload}^{simple} = \sqrt{\frac{Is_{Q3}\ Is_{Q4}}{Is_{Q1}\ Is_{Q2}}}\sqrt{I_B}\sqrt{I_{in}}$$

## Example

For $Is_{Qi} = 10^{-16}\mathrm{A}$, $I_{in}/1\mathrm{mA} \in [0.2, 0.3]$, and $I_B/1\mathrm{mA} \in [0.1, 0.2]$ interval-techniques **prove** $|i_{vload} - i_{vload}^{simple}| < 2.5\%$

# Validation of Symbolic Approximations

## Compare different accuracy levels

$$i_{vload} \approx i_{vload}^{simple} \text{ ?}$$

$$i_{vload} = \frac{\sqrt{Is_{Q3}\, Is_{Q4}\, ((0.025 Is_{Q3}\, Is_{Q4} - Is_{Q2}\, Is_{Q1}) 10^{-2} I_{in}{}^2 + Is_{Q1}\, Is_{Q2}\, I_B\, I_{in})} - 5 \cdot 10^{-3} I_{in}\, Is_{Q3}\, Is_{Q4}}{Is_{Q1}\, Is_{Q2}}$$

$$i_{vload}^{simple} = \sqrt{\frac{Is_{Q3}\ Is_{Q4}}{Is_{Q1}\ Is_{Q2}}}\sqrt{I_B}\sqrt{I_{in}}$$

## Example

For $Is_{Qi} = 10^{-16}\mathrm{A}$, $I_{in}/1\mathrm{mA} \in [0.2, 0.3]$, and $I_B/1\mathrm{mA} \in [0.1, 0.2]$ interval-techniques **prove** $|i_{vload} - i_{vload}^{simple}| < 2.5\%$

# Formal Verification of Analog Circuits

**Motivation:**
**Simulation vs. Formal Verification**

Contrary to simulation, formal verification techniques aim at:

- Verifying system properties before manufacturing
- Making the verification process automatic
- Coping with the variability of parameters and input signals

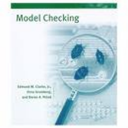
# Formal Verification of Analog Circuits

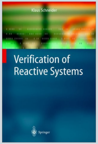
## Motivation: Simulation vs. Formal Verification

Contrary to simulation, formal verification techniques aim at:

- Verifying system properties before manufacturing
- Making the verification process automatic
- Coping with the variability of parameters and input signals

## The Model Checking Verification Technique

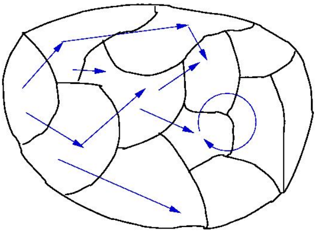



A well established technique for the verification of digital finite systems. Properties stated via a temporal logic language verified:

- completely automatically
- on any computational path of the graph-modeled system

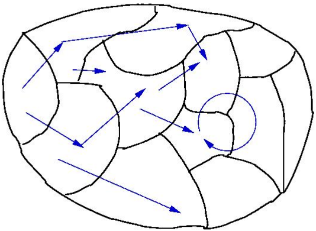
# Formal Verification of Analog Circuits



## The Model Checking Verification Technique

A natural way of extending the model checking technique to analog circuits involves the extraction of some discrete approximation of the infinite graph underlying the evolution of an analog circuit.

# Formal Verification of Analog Circuits

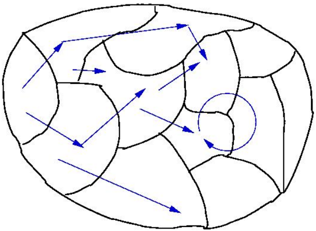


## The Model Checking Verification Technique

However, two major issues have to be considered to obtain an effective final verification technique:

- Need of adequate languages to state analog properties.
- Need of formal tools to ensure preservation results from the abstraction to the analog model.

# Formal Verification of Analog Circuits



## The Model Checking Verification Technique

However, two major issues have to be considered to obtain an effective final verification technique:

- Need of adequate languages to state analog properties.
- Need of formal tools to ensure preservation results from the abstraction to the analog model.

# $CTL_f$: a Language for the Modeling of Analog Prop.

$CTL_f$ is an enrichment of classical CTL language, where we allow 'basic formulæ' to state the membership in boxes of values given by arbitrary elementary functional relations.

## The $CTL_f$ Syntax

Let $X = \{x_1, \ldots, x_n\}$ be a finite set of real valued variables. The set of $CTL_f$ formulæ is defined according the following grammar:

| | | |
|---|---|---|
| $\phi ::=$ | $f(x_{i_1}, \ldots, x_{i_m}) \triangleright I$ | (basic formulæ testing membership in box $I$) |
| | $\neg\phi \mid \phi \vee \phi$ | (boolean combinators) |
| | $\mathrm{E}\phi\mathrm{U}\phi \mid \mathrm{A}\phi\mathrm{U}\phi$ | (temporal combinators & path quantifiers) |

where $1 \leq i_1 \leq \cdots \leq i_m \leq n$, $f : \mathbb{R}^m \mapsto \mathbb{R}^p$ is an arbitrary composition of elementary functions, $I$ is a box in $\mathbb{IR}^p$.

# $\text{CTL}_f$: a Language for the Modeling of Analog Prop.

$\text{CTL}_f$ is an enrichment of classical CTL language, where we allow 'basic formulæ' to state the membership in boxes of values given by arbitrary elementary functional relations.

## The $\text{CTL}_f$ Syntax

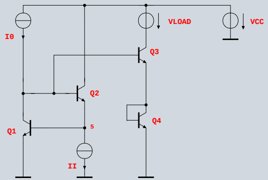
Let $X = \{x_1, \dots, x_n\}$ be a finite set of real valued variables. The set of $\text{CTL}_f$ formulæ is defined according the following grammar:

| | | |
|---|---|---|
| $\phi ::=$ | $f(x_{i_1}, \dots, x_{i_m}) \triangleright I$ | (basic formulæ testing membership in box $I$) |
| | $\neg\phi \mid \phi \vee \phi$ | (boolean combinators) |
| | $\mathsf{E}\phi\mathsf{U}\phi \mid \mathsf{A}\phi\mathsf{U}\phi$ | (temporal combinators & path quantifiers) |

where $1 \leq i_1 \leq \cdots \leq i_m \leq n$, $f : \mathbb{R}^m \mapsto \mathbb{R}^p$ is an arbitrary composition of elementary functions, $I$ is a box in $\mathbb{IR}^p$.

# CTL$_f$: a Language for the Modeling of Analog Prop.

## Example




- In the *square root function block* property (1) holds:

$$I_0 = \beta \sqrt{I_I} \sqrt{I_B} \ , \qquad (1)$$

- CTL$_f$ formula encoding property (1):

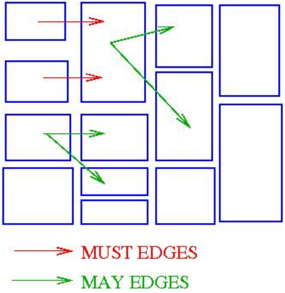
$$\mathrm{AG}((I_0 - \beta \sqrt{I_I} \sqrt{I_B}\ ) \triangleright [-\delta, +\delta]) \qquad (2)$$

# 3-Valued $CTL_f$ Model Checking on Analog Circuits

We use Interval Arithmetic to provide a three-valued semantics to our $CTL_f$ logic on suitable abstraction of the states-space of analog circuits (box-modal abstractions).

**The Three-Valued Semantics**

A natural choice since the value of our $CTL_f$ formulae could be indefinite on abstract states (boxes)

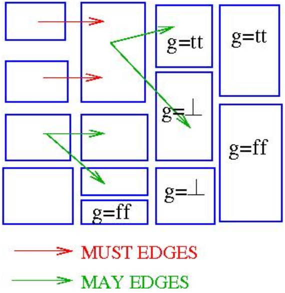


# 3-Valued $CTL_f$ Model Checking on Analog Circuits

We use Interval Arithmetic to provide a three-valued semantics to our $CTL_f$ logic on suitable abstraction of the states-space of analog circuits (box-modal abstractions).

## The Three-Valued Semantics

A natural choice since the value of our $CTL_f$ formulae could be indefinite on abstract states (boxes)

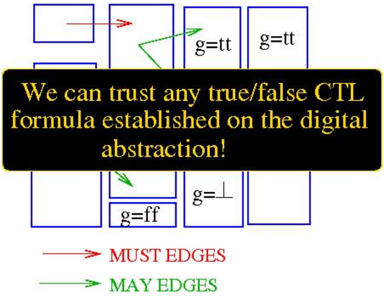


# 3-Valued $CTL_f$ Model Checking on Analog Circuits

We use Interval Arithmetic to provide a three-valued semantics to our $CTL_f$ logic on suitable abstraction of the states-space of analog circuits (box-modal abstractions).

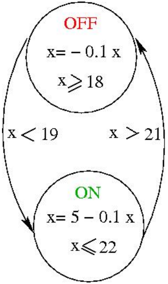
## The Three-Valued Semantics

Allow to apply 3-valued model checking techniques and to state a fundamental preservation result for $CTL_f$ verified formulae:



# Outline

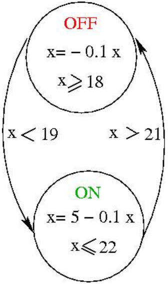
# The Hybrid Automata (HA) Model




## Hybrid Automata

- State-of-the-art formalism for modeling hybrid systems (interacting continuous/discrete dynamics) in CS.
- Combine the automata tool (for modelling the discrete component) with systems of differential equations (for representing continuous dynamics).
- Inverse relation between decidability and modeling capability.

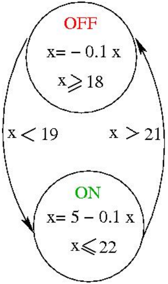
# The Hybrid Automata (HA) Model




## Hybrid Automata

- State-of-the-art formalism for modeling hybrid systems (interacting continuous/discrete dynamics) in CS.
- Combine the automata tool (for modelling the discrete component) with systems of differential equations (for representing continuous dynamics).
- Inverse relation between decidability and modeling capability.

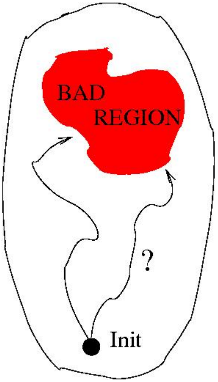
# The Hybrid Automata (HA) Model




## Hybrid Automata

- State-of-the-art formalism for modeling hybrid systems (interacting continuous/discrete dynamics) in CS.
- Combine the automata tool (for modelling the discrete component) with systems of differential equations (for representing continuous dynamics).
- Inverse relation between decidability and modeling capability.

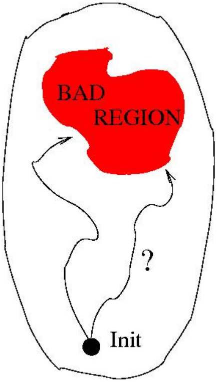
# Hybrid Automata Verification




## The Reachability Problem

- Is the region *B* reachable from any initial state?
- Safety analysis can be reduced to reachability.
- Most (interesting) classes of HA known undecidable with respect to reachability.

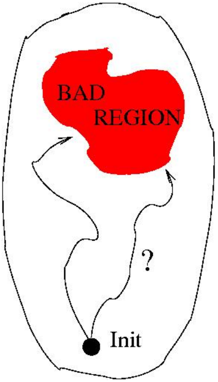
# Hybrid Automata Verification




## Reachablity Analysis over Undecidable Hybrid Automata

- Known symbolic techniques force to choose between reach-sets over-/under-approx. (i.e between trying to prove/disprove safety).
- Mostly based on the simulation preorder that in general allows for the preservation of only true prop. in the universal fragment of CTL.

# Hybrid Automata Verification




## Reachablity Analysis over Undecidable Hybrid Automata

- In a recent joint work, we introduce the first framework for combining over-/under approximated reach-set analysis on (undecidable) HA.
- Our method is based on a succession of abstractions, where general CTL stated reactive system properties can be monotonically proved/disproved.

# Outline

# More on VerSiS Project: Ongoing and Future Work

- Extension of verified behavior region in robot verification
- Optimization of the behavior architecture using results from model checking and robot verification
- Expand combined interval and three-valued logic approach to real-world applications
- Ongoing work on the use of computer algebra methods (in particular Gröbner Bases methods) for the analysis of hybrid automata.
- Ongoing work related to the issue of modeling hybrid systems by means of synchronous languages, in order to integrate them into the *Averest Tool*.

# Further Reading


- Raffaella Gentilini, Klaus Schneider, and Alexander Dreyer.
Three-valued automated reasoning on analog properties.
In *17th ACM Great Lakes Symposium on VLSI (GLSVLSI '07). Stresa-Lago Maggiore, Italy*. ACM Press, March 2007.

- Raffaella Gentilini, Klaus Schneider, and Bud Mishra.
Series of abstractions of hybrid automata for monotonic ctl model checking.
In *International Symposium on Logical Foundations of Computer Science (LFCS '07). New York, U.S.A.*, LNCS. Springer, 2007.

- Alexander Dreyer.
Interval methods for analog circuits.
Technical report, Fraunhofer ITWM, 2006.

# Further Reading (cont'd)

Michael Brickenstein and Alexander Dreyer.
Polybori: A framework for Gröbner basis computations with Boolean polynomials.
Submitted to: *Effective Methods in Algebraic Geometry MEGA 2007*, June 2007.

Martin Proetzsch, Tobias Schüle, Karsten Berns, and Klaus Schneider.
Formal Verification of Safety Behaviours of the Outdoor Robot RAVON.
Submitted to: *4th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2007). Angers, France*. May 2007.