

Formal Verification of Chi Models Using PHAVer

K.L. Man

Dept. Math. and Computer Science

R.R.H. Schiffelers

Dept. Mechanical Engineering

Eindhoven University of Technology

The Netherlands

Email: {k.l.man,r.r.h.schiffelers}@tue.nl

Table of contents

- Introduction of our Ph.D project
- Theoretical background
- Introduction of hybrid Chi (χ)
- Example for modeling
- Translations between hybrid χ and other formalism
- Tool support
- Verification example: water tank
- Conclusions

Introduction of our Ph.D project

- Project :
 - Formal specification and analysis of hybrid systems.
- People :
 - Prof. J.C.M. Baeten, Prof. J.E. Rooda, Dr. M.A. Reniers, Dr. D.A. van Beek, R.R.H. Schiffelers and K.L. Man (Ph.D candidates).
- Goal :
 - Development of a process algebra that is suited to modeling, simulation and verification of hybrid systems.

Theoretical background

- Hybrid systems
 - exhibit both discrete and continuous behavior,
 - application areas: air-traffic control, automated manufacturing, chemical process control, etc.
- Process algebras for the specification and analysis of hybrid systems
 - HyPA and ACP_{hs}^{srt} (based on ACP).

Introduction of hybrid Chi (χ)

- the hybrid χ formalism integrates concepts from dynamics and control theory with concepts from computer science;
- the semantics of hybrid χ is defined by means of deduction rules in structured operational semantics (SOS) style that associates a hybrid transition system with a hybrid χ process;
- hybrid χ is closely related to hybrid automata.

Hybrid χ process

A hybrid χ process is a triple $\langle p, \sigma, E \rangle$, where p denotes a process term, σ denotes a valuation and E denotes an environment. An environment E is a tuple (C, J, L, H, R) :

- C denotes a set of *continuous* variables, the values of continuous variables change according to an absolutely continuous function of time while delaying,
- J denotes a set of *jumping* variables, the values of jumping variables can jump to arbitrary values during action transitions,
- L denotes the set of *algebraic* variables, algebraic variables behave in a similar way as continuous variables, but algebraic variables may also change according to a discontinuous function of time,
- H denotes the set of *channels* (for communication between parallel components),
- R denotes a recursion definition.

Furthermore, the values of discrete variables remain constant while delaying, and the predefined variable 'time' denotes the current model time.

Syntax of hybrid χ

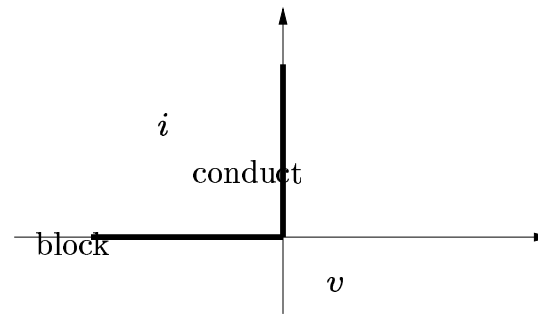
The set of process terms P is defined by the following grammar for the process terms $p \in P$:

$$\begin{aligned}
 p ::= & W : r \gg l_a \mid u \mid \delta \mid \perp \\
 & \mid [p] \mid u \curvearrowright p \mid p; p \mid b \rightarrow p \mid p \parallel p \\
 & \mid p \parallel p \mid h !! \mathbf{e}_n \mid h ?? \mathbf{x}_n \mid \partial_A(p) \mid v_H(p) \\
 & \mid X \mid \iota_{J^+}(p) \mid \llbracket \mathbb{V} \sigma_{\perp}, C, L \text{ '}' p \rrbracket \mid \llbracket \mathbb{H} H \text{ '}' p \rrbracket \mid \llbracket \mathbb{R} R \text{ '}' p \rrbracket
 \end{aligned}$$

Example

An ideal diode can either block or conduct the current:

- block: $v \leq 0 \wedge i = 0$
- conduct: $i \geq 0 \wedge v = 0$



In hybrid χ :

$\langle (v \leq 0 \wedge i = 0) \vee (i \geq 0 \wedge v = 0), \{v \mapsto -1, i \mapsto 0\}, E \rangle,$

where $E = (\{v, i\}, \emptyset, \emptyset, \emptyset, \emptyset)$

Translations between hybrid χ and other formalisms

- translations that aim to show that the hybrid χ formalism is at least as expressive as
 - * discrete-time piecewise affine systems,
 - * continuous-time piecewise affine systems,
 - * hybrid automata.
- translation from hybrid χ to hybrid automata:
 - * enables verification of hybrid χ specifications using existing hybrid automata based verification tools:
 - HYTECH,
 - PHAVER.

Tool support

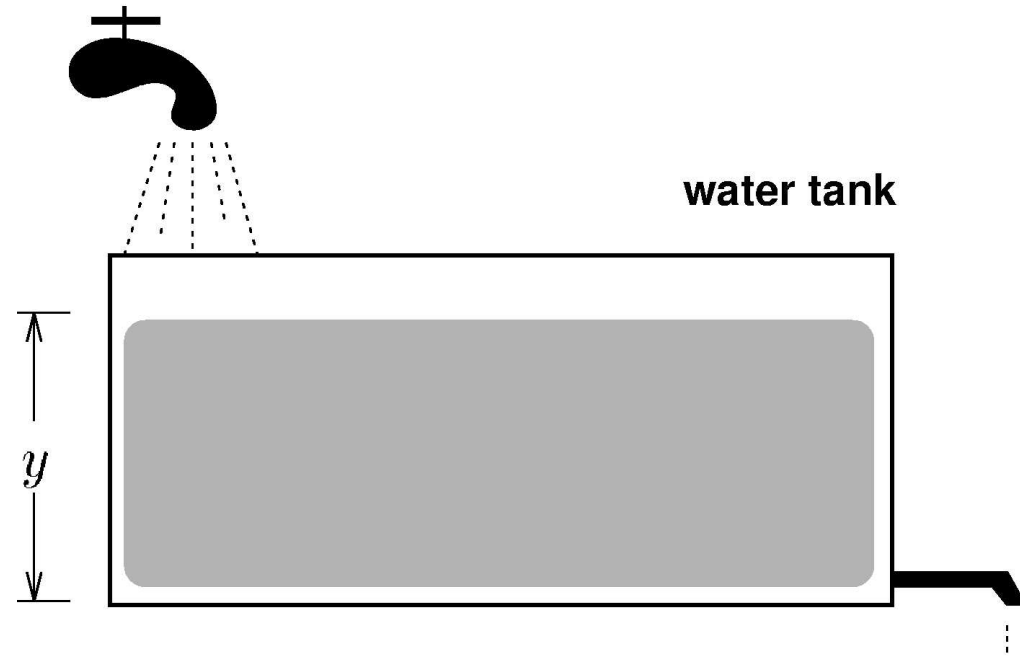
– **Symbolic Simulator**

- * simulates all possible transitions for a given χ process,
- * uses the symbolic solver from MAPLE,
- * implemented in Python.

– **Chi2HA translator**

- * translates hybrid Chi specifications to hybrid automata,
- * implemented in Python.

Verification example: Water Tank (1/2)



x : stop watch

y : water level

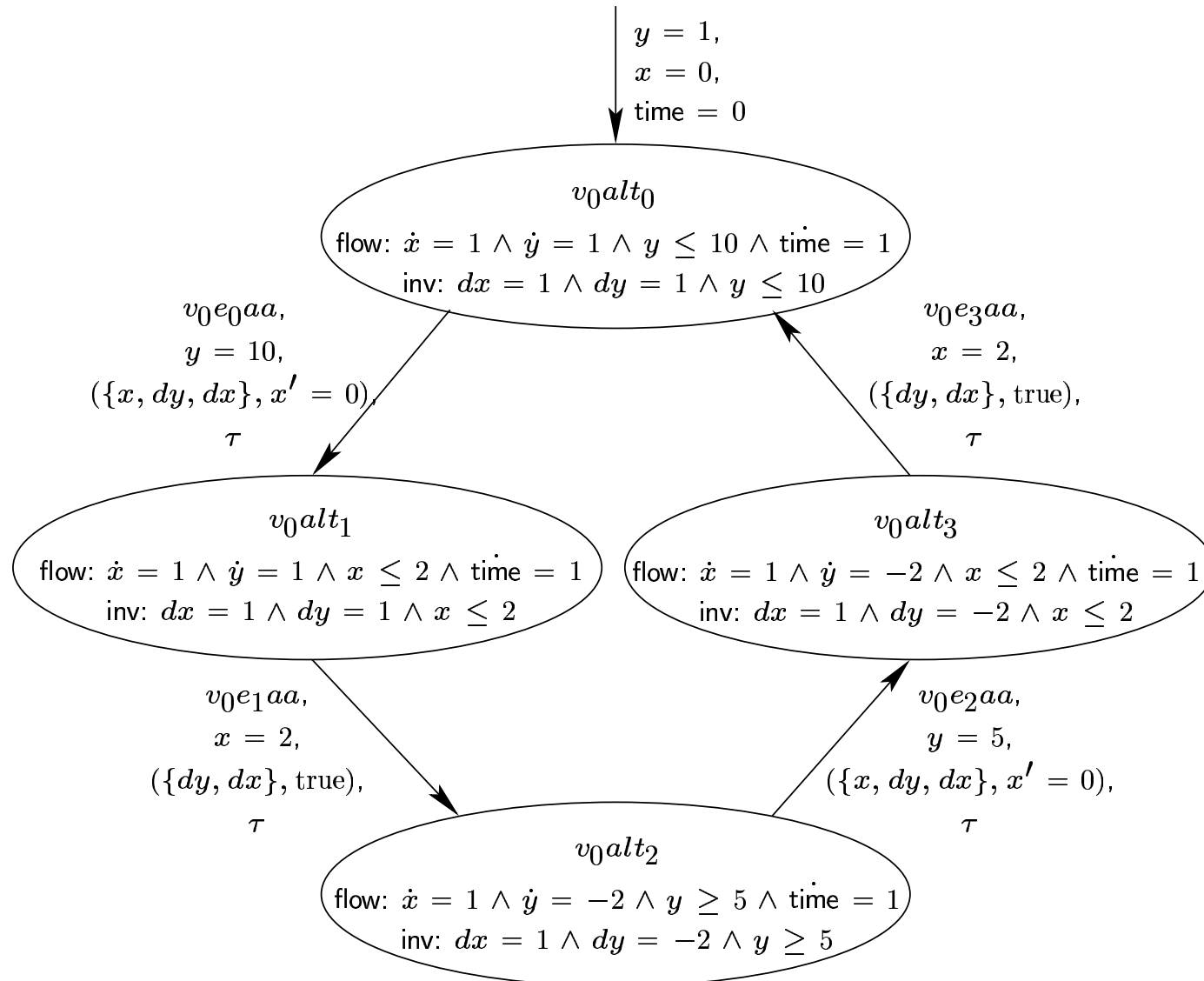
Verification example: Water Tank (2/2)

The variable x denotes a stop watch, and the variable y denotes the water level, and the water level is 1 initially and the pump is turned on :

- when the pump is turned on: the water level rises by 1 per second,
- as soon as the water level is 10, and then the pump is turned off with a delay of 2 seconds;
- when the pump is turned off: the water level drops by 2 per second,
- as soon as the water level reaches 5, and then the pump is turned on again with a delay of 2 seconds.

Hybrid Chi specification of the water tank

$$\begin{aligned}
 & \langle \dot{x} = 1 \\
 & \parallel * ((\dot{y} = 1 \quad \wedge \quad y \leq 10 \quad \parallel \quad [y \geq 10 \rightarrow \{x\} : x = 0 \gg \tau]) \\
 & \quad ; (\dot{y} = 1 \quad \wedge \quad x \leq 2 \quad \parallel \quad [x \geq 2 \rightarrow \{\emptyset\} : \text{true} \gg \tau]) \\
 & \quad ; (\dot{y} = -2 \quad \wedge \quad y \geq 5 \quad \parallel \quad [y \leq 5 \rightarrow \{x\} : x = 0 \gg \tau]) \\
 & \quad ; (\dot{y} = -2 \quad \wedge \quad x \leq 2 \quad \parallel \quad [x \geq 2 \rightarrow \{\emptyset\} : \text{true} \gg \tau]) \\
 &) \\
 & , \{x \mapsto 0, y \mapsto 1\}, (\{x, y\}, \emptyset, \emptyset, \emptyset, \emptyset) \\
 & \rangle
 \end{aligned}$$



Water tank automaton generated by the translation tool.

Verification result of the water tank automaton using PHAVer

Verified property	locations	CPU time
The water level is always between 1 and 12	4	0.2 sec.

The property states that there is no overflow in the tank, and the tank is never empty (i.e., water level is never below 1 and above 12).

Conclusions

- hybrid χ can be used to give elegant specifications of hybrid systems.
- validation and verification of hybrid systems described in hybrid χ can be done by:
 - * simulating the specifications using the hybrid χ simulator,
 - * translating hybrid χ specifications to hybrid automata, and verifying them using hybrid automata based verification tool.