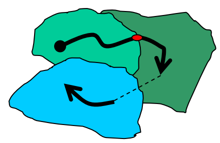
Models for Hybrid Systems



MoDES Meeting
DTU October 2006

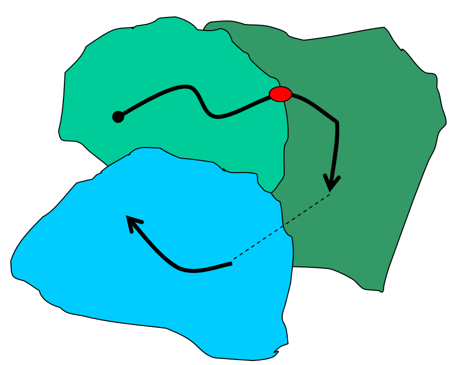
Anders P. Ravn
Aalborg University, Denmark

Hybrid System

A dynamical system with a *non-trivial* interaction of discrete and continuous dynamics

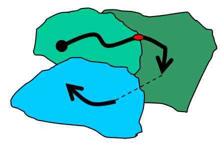
- autonomous
  - switches
  - jumps
- controlled
  - switches
  - jump

between manifolds
(Branicky 1995)



Examples

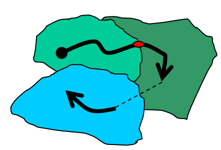
- Autonomous Systems (Tomlin)
- Biological Systems
- Power Grid
- Power Converters
- Automotive Power Trains
- Refrigeration (Larsen)
- …



Application Characteristics

Intelligent Control:

- – Modelling of plant modes and zones
- – Switched control
- – Model based control
- – Bounded optimal control
- – Distributed control
- – …



Hybrid Automaton – Syntax (Henzinger, …)

$X = \{x_1, \dots x_n\}$ - variables

(V, E) – control graph

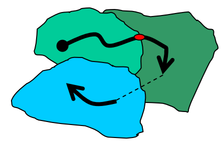
$init: V \to pred(X)$

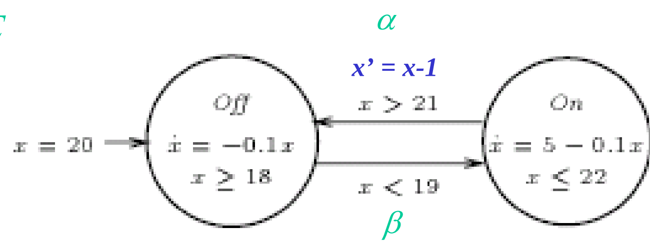
$inv: V \to pred(X)$

$flow: V \to pred(X \cup \dot{X})$

$jump: E \to pred(X \cup X')$

$event: E \to \Sigma$





Transition Semantics of HA

$X = \{x_1, \dots x_n\}$ - variables

(V, E) – control graph

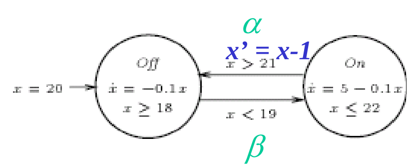
$init: V \to pred(X)$

$inv: V \to pred(X)$

$flow: V \to pred(X \cup \dot{X})$

$jump: E \to pred(X \cup X')$

$event: E \to \Sigma$



Q - states – $\{(v,\mathbf{x}) \mid v \in V \text{ and } inv(v)[X := \mathbf{x}]\}$

Q^0 – initial states - $\{(v,\mathbf{x}) \in Q \mid init(v)[X := \mathbf{x}]\}$

A - labels - $\Sigma \cup R_{\geq 0}$

$\{ (v,\mathbf{x}) - \alpha\!\to (v',\mathbf{x}') \mid e \in E(v,v') \text{ and } event(e) = \alpha \text{ and } jump(e)\ [X := \mathbf{x}]\}$

$\{ (v,\mathbf{x}) - \delta\!\to (v,\mathbf{x}') \mid \delta \in R{\geq}0 \text{ and } f: (0,\delta) \to R^n \text{ s.t. } f \text{ is diff. and } f(0) = \mathbf{x} \text{ and } f(\delta) = \mathbf{x}' \text{ and } flow(v)[X := f(t), \dot{X} := \dot{f}(t)],\ t \in (0,\delta) \}$

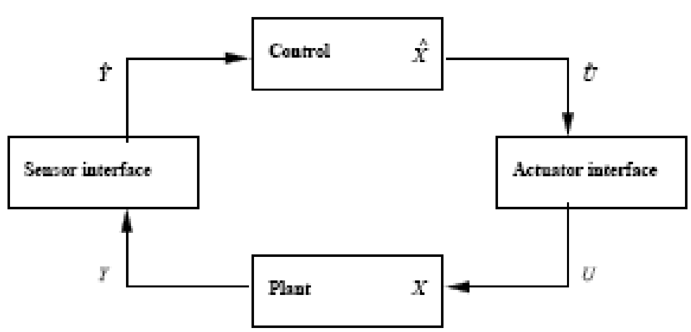
Hybrid Systems in Control

(take up of CS ideas 1990 - …)

- Hybrid Automata is the Spec. Language
- Tools for simulation and model checking
  (Henzinger,Alur,Maler,Dang)
- Bisimulation as abstraction technique
  (Pappas,Neruda,Koo)
- Industrial Applications

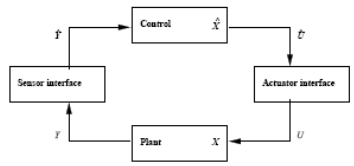
… so all is well …

When analyzing ”the control system” model:



Unnecessary CS Concepts ?

- Concurrency
- Communication
- Modularity
- Property Languages



… and yet:

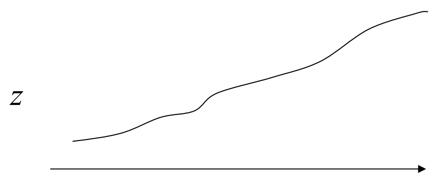
"Control Engineers will have to master computer and software technologies to be able to build the systems of the future, and software engineers need to use control concepts to master ever-increasing complexity of computing systems."

(IFAC Newsletter December 2005 No.6)

Proposition

The digital computer is feasible

IS



a REFINEMENT of

$$z < 1 \rightarrow z := 1$$

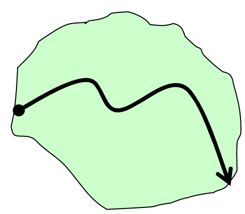
M. Rönnkö, A. P. Ravn and K. Sere: Hybrid Action Systems, *Theoretical Computer Science* **290,** pp. 937-973, January 2003

A Programming Language: Actions

A ::= $x := e$
| $p \rightarrow$ A
| A1 [] A2
| ***do*** A ***od***

The Differential Action (syntax)

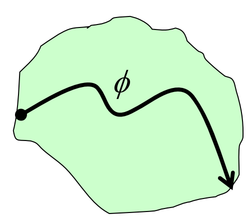
$$p :\rightarrow \Psi(\underline{x}, \underline{\dot{x}})$$



The Differential Action (semantics)

$$p :\rightarrow \Psi(\underline{x}, \underline{\dot{x}})$$

$\{ (v,\mathbf{x}) \rightarrow (v,\mathbf{x'}) \mid \textit{there exists } \phi \in SF(p, \Psi)$
$\textit{s.t.} \quad 0 < \Delta(\phi,p) < \infty$
$\textit{and} \quad \phi(0) = \mathbf{x}$
$\textit{and} \quad \phi(\Delta(\phi,p)) = \mathbf{x'} \}$



Conclusion

- HS is a key ingredient for Intelligent Control
- Support for analysis and synthesis is a rich research area
- TA a specialization and abstraction