Un panorama des systèmes hybrides en temps discret

Giancarlo Ferrari-Trecate

Institute National de Recherche en
Informatique et en Automatique (INRIA)
Unité de Rocquencourt

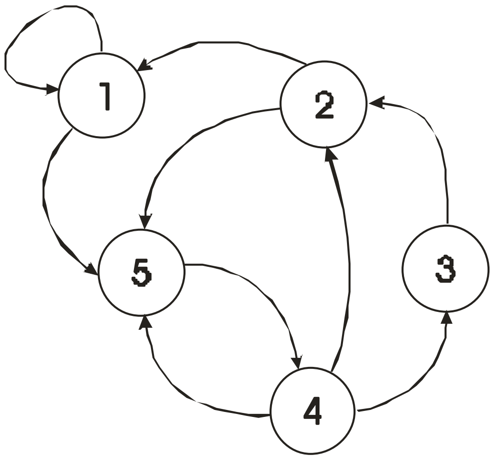
Giancarlo.Ferrari-Trecate@inria.fr

Technical reports: **http://www-rocq.inria.fr/Giancarlo.Ferrari-Trecate/**

INRIA
ROCQUENCOURT

Hybrid Systems

Computer Science



Finite state machines

$$
\begin{aligned}
S &\triangleq (X, U, \varphi),\\
X &= \{1,2,3,4,5\},\\
U &= \{A, B, C\},\\
\varphi &: X \times U \to X
\end{aligned}
$$

Control Theory

Continuous dynamical systems

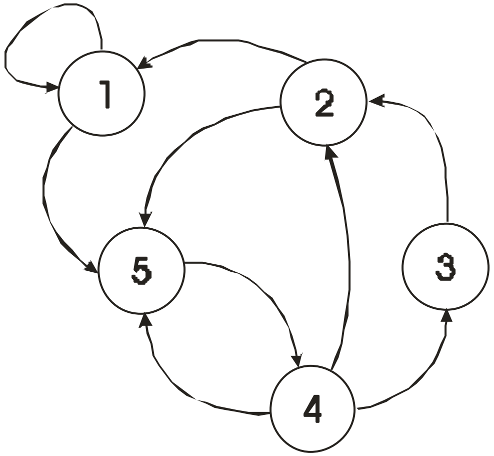
$$
\begin{aligned}
x(t+1) &= f(x(t), u(t))\\
y(t) &= g(x(t), u(t))
\end{aligned}
$$

$$x \in \mathbf{R}^n, u \in \mathbf{R}^m, y \in \mathbf{R}^p$$

$$f, g \in C^1$$

Hybrid Systems

Computer Science



Finite state machines

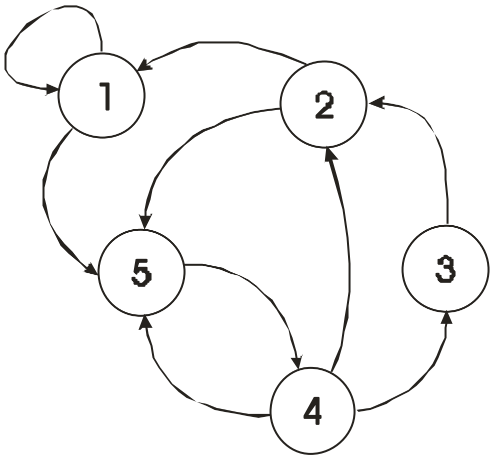
Good for modeling:
- discrete actions/decisions
- computer programs
- logic rules (e.g. if … then … else)

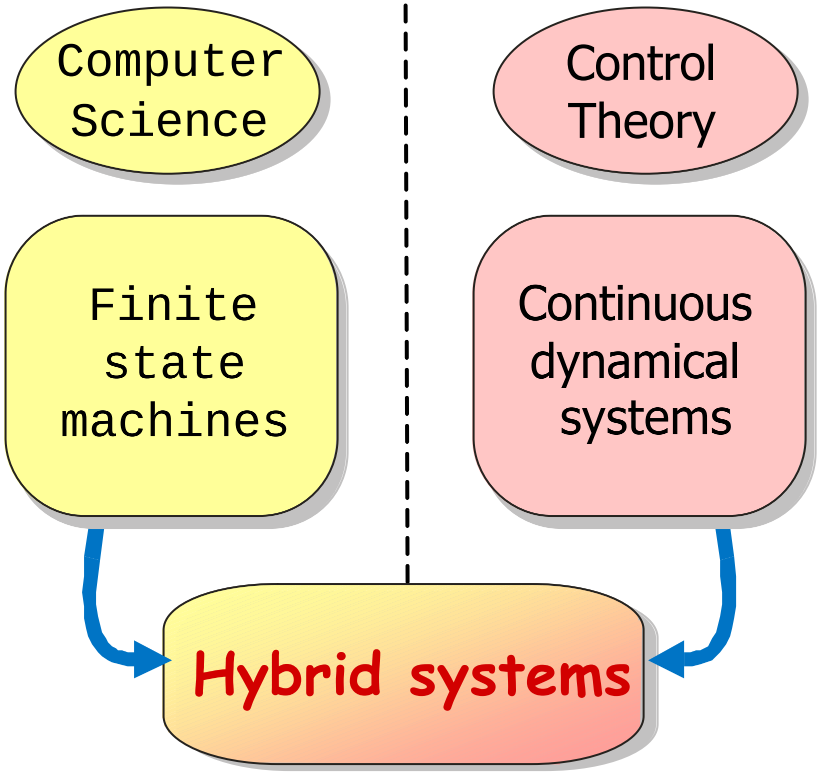
Control Theory

Continuous dynamical systems

Good for modeling:
- continuous physical quantities (concentrations, velocity, energy, …)

Hybrid Systems





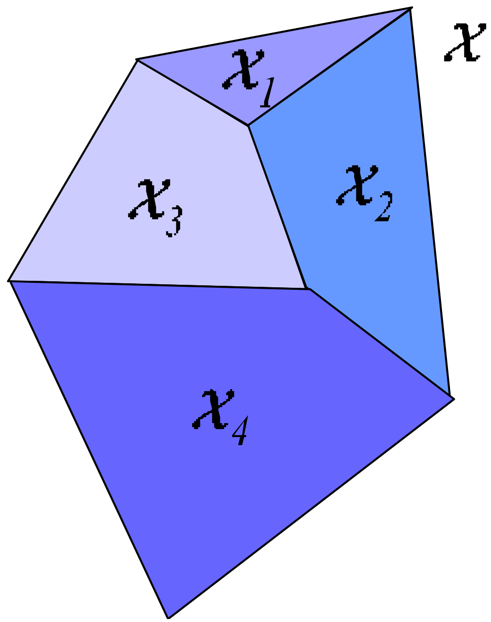
Good for modeling the interaction between discrete/logic components and continuous components

This is the standard scenario of modern technology
(e.g. embedded systems = real-time programs
interacting with physical processes)

Piece-Wise Affine (PWA) systems

$$\begin{aligned} x(t+1) &= A_i x(t) + B_i u(t) + f_i \\ y(t) &= C_i x(t) + g_i \end{aligned} \qquad \text{if } [\, x'(t) \;\; u'(t) \,]' \in \mathcal{X}_i$$

- s submodels
- $\mathcal{X}$ is a polytope, $\{\mathcal{X}_i\}_{i=1}^{s}$ is a polyhedral partition of $\mathcal{X}$
- Affine dynamics on each region



- States, inputs and outputs are partitioned in Boolean and continuous components

$$x = \begin{bmatrix} x_c \\ x_\ell \end{bmatrix}$$

Physical quantities: $x_c \in \mathbf{R}^{n_c}$

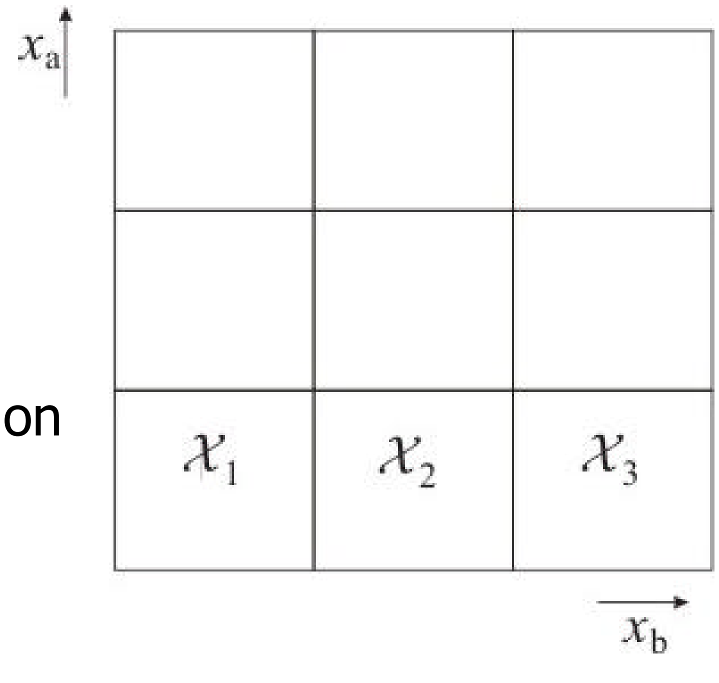
Logic states: $x_\ell \in \{0,1\}^{n_\ell}$

PWA systems in GDyn

$$\begin{aligned} x(t+1) &= A_i x(t) + B_i u(t) + f_i \\ y(t) &= C_i x(t) + g_i \end{aligned} \quad \text{if } [\, x'(t) \;\; u'(t) \,]' \in \mathcal{X}_i$$

- States: Protein concentrations
- Inputs: Concentrations of proteins regulated outside the network
- Outputs: Measurable quantities

- Regions: Regulatory domains
- Dynamics: Rates of expression/degradation in each domain

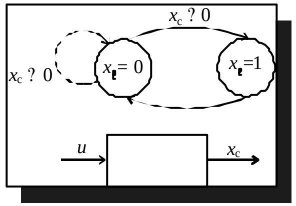


Modeling capabilities

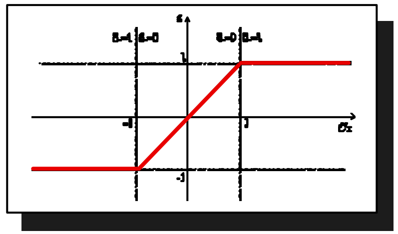
Real-world domains → Abstraction → Hybrid phenomena

- Automotive
- Biology
- Electrical Engineering
- Mechanical engineering
- Medicine
- Physics
- Economy
- …

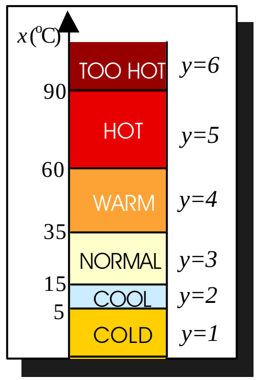
- Automata dependent on dynamical systems



- Nonlinearities (by piecewise affine approximation)



- Qualitative behaviours

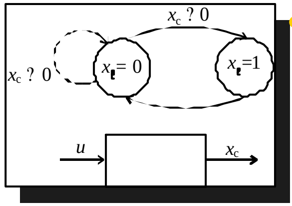


- Constrained dynamics
- …

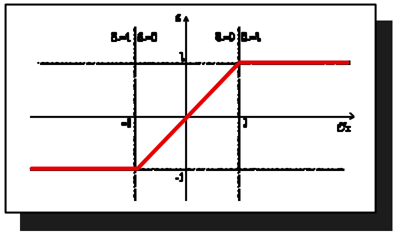
Modeling capabilities

Hybrid phenomena → Abstraction → **PWA form of Hybrid Systems**

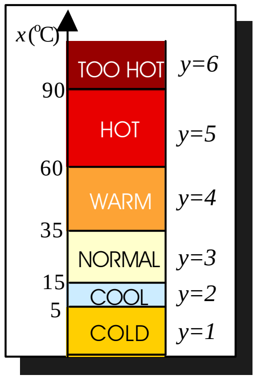
- Automata dependent on dynamical systems



- Nonlinearities
  (by piecewise affine approximation)



- Qualitative behaviours



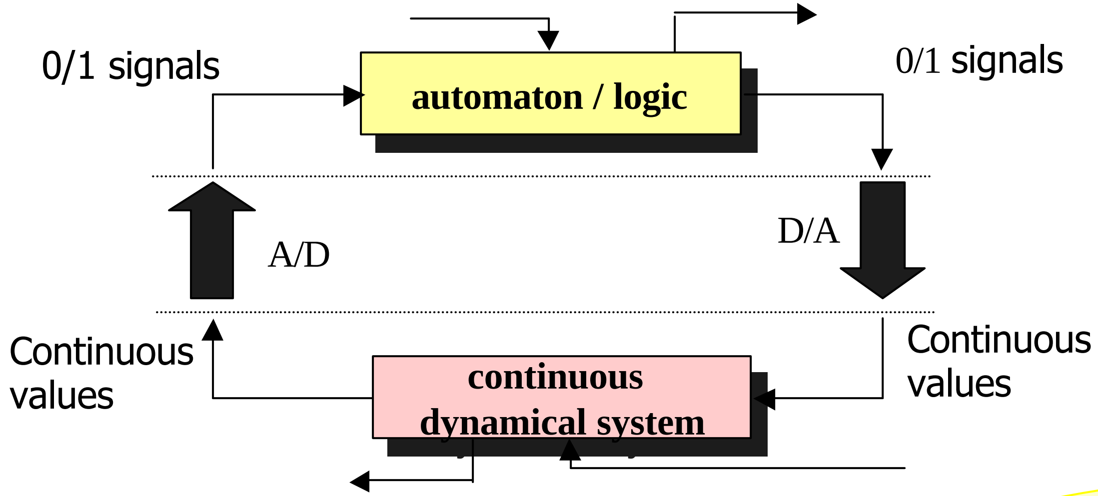
- Constrained dynamics
- …

PWA systems

Modeling language

Scientists do not think in terms of PWA models …

High-level system description:



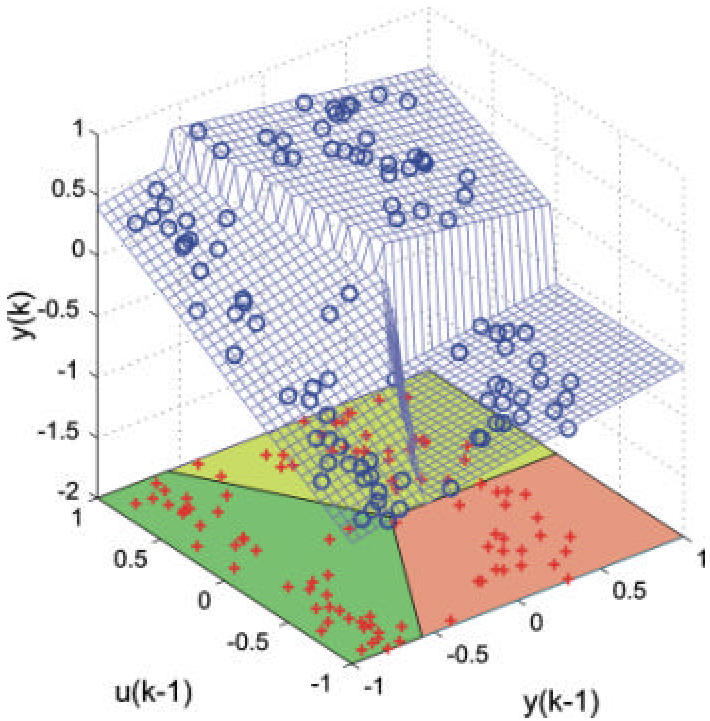
HYSDEL compiler
(HYbrid System DEscription Language)
(Torrisi, Bemporad 2003)

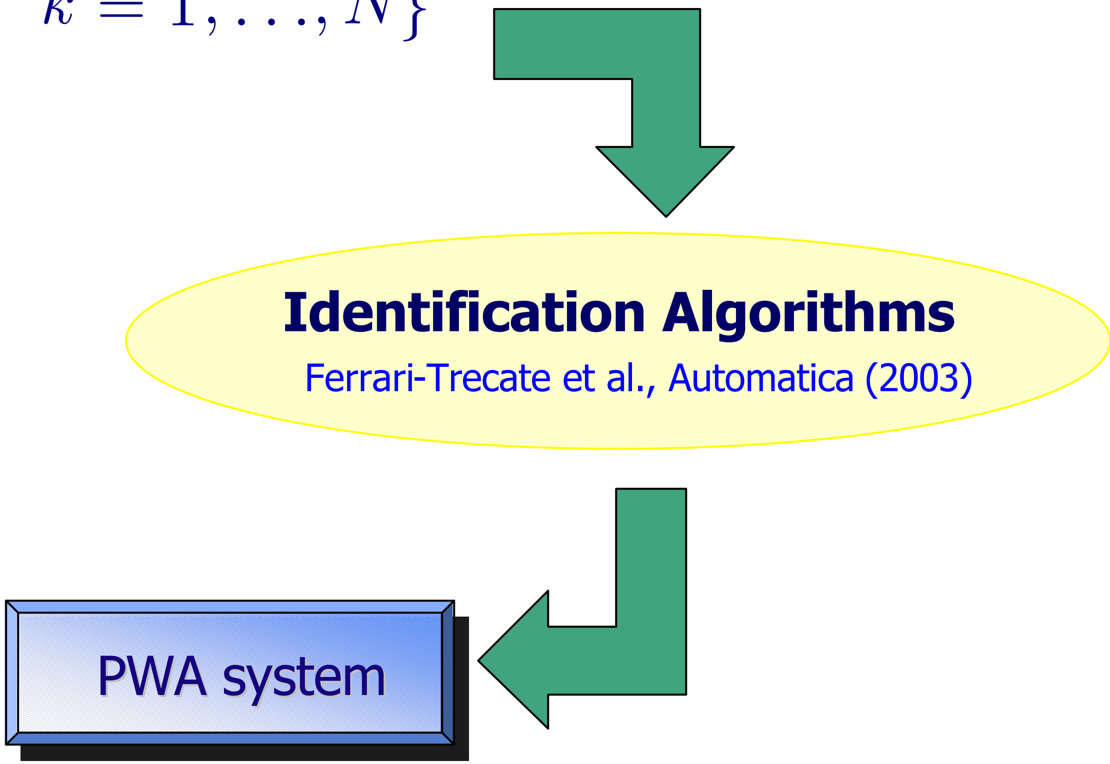
PWA system

Modeling: Identification

Is it possible to get PWA models form input/output experimental data ?

Dataset: $\mathcal{S} = \{(u(k), y(k)),\ \ k = 1, \ldots, N\}$





OK, I have a PWA model … and then ?

Hybrid Systems in PWA form

=

modeling framework for simulations ?

OK, I have a PWA model … and then ?

Hybrid Systems in PWA form

=

modeling framework for simulations ?

NOT ONLY !

Hybrid systems in PWA form

=

Mathematical framework for solving analysis and synthesis problems both theoretically and computationally

Tools for PWA systems

PWA Systems

Theory

Analysis

- Reachability/Verification
- Stability
- Observability

Synthesis

- Control (MPC)
- Control with performance (LMI)
- State estimation (MHE)/fault detection

Computation

Computational Tools

- Mixed Integer Programming
- Polyhedral computation
- Multiparametric Programming
- Linear Matrix Inequalities

Example: State estimation

Theory

PWA system: $$\begin{aligned} x(t+1) &= A_i x(t) + B_i u(t) + f_i \\ y(t) &= C_i x(t) + g_i \end{aligned} \quad \text{if} \quad [\, x'(t) \;\; u'(t) \,]' \in \mathcal{X}_i$$

Problem: When the state is not measurable it has to be reconstructed from the measurements $u(t), y(t), t = 0, \ldots, T$

Solution: Moving horizon observers can guarantee asymptotic convergence of the estimates to the true state Ferrari-Trecate et al., IEEE Trans. AC, (2002)

Computation

Moving horizon observers can be implemented by solving mixed-integer linear (or quadratic) optimization problems

⇩

Solvers: Cplex, Xpress, Baron, …

PWA systems: The universal tool ?

PWA system provide a very flexible framework for modeling, analysis and synthesis …

BUT

… the computations scale, in general, in an NP (Non Polynomial) fashion with respect the number of integer variables + number of regions. The situation is even worse in presence of large model uncertainties.

Small scale problems: PWA systems are OK !

Large scale problems: Still a lot of work to do …

- PWA systems are too flexible: The key problem is:

How to exploit special structures in analysis and synthesis ?

More in detail: exploit hierarchical structures that allow to abstract from the time/space details → Multi-scale representations

PWA system theory in GDyn

PWA system theory could be useful in GDyn for:

- providing insights about the behaviors of genetic sub-networks (e.g. minmax optimization for evaluating the possibility of isolating sub-networks)
- parameter identification (rates and thresholds)
- state estimation (reconstruction of unmeasured protein concentrations)

GDyn could be useful for large-scale PWA system theory in:

- providing good examples
- gaining insights about how to discover hierarchical structures.