# Bestiarium of Hybrid Systems[*]

Tomas Krilavičius
e-mail: t.krilavicius@utwente.nl
Formal Methods and Tools Group
University of Twente
P.O.Box 217, 7500 AE Enschede

March 22, 2005

**Abstract**

*Hybrid systems* is a wide class of systems, where discrete and continuous phenomena can be encountered. Significant progress was done in this young branch of science, but even more problems are encountered. One of such problems is evaluation of different hybrid systems modelling approaches. Different approaches can be suitable for the different classes of hybrid systems. And, of course, different problems can be solved easier or faster, using different approaches. One of ways to compare them is to use benchmarks. In this paper, a collection of examples of hybrid systems is presented. These examples can be used for benchmarking and/or other purposes.

## Contents

---

[*] Eternal draft: Content is not updated from February, 2003

# 1 Introduction

A very wide class of systems is hidden under the notion of hybrid systems. But only a fraction of them are used as examples in the different papers. List of such systems can be useful at least for two purposes: to find a specific system and references, or to find, which class of systems was left aside. This list is not exhaustive, but it can be useful, as a starting point for a hybrid systems researcher.

# 2 Examples of Hybrid Systems

In this section a set of examples is presented. The list is not full and the examples are chosen by publications, which were easy to get at the time of analysis.

The examples are presented in a following way - the list of examples is given, and then in every subsection an example (a class of examples) is described, and literature, where the example is analyzed, is listed.

**Remark 2.1** (Delays). In some examples delays are introduced explicitly, but in almost in all examples delays for information from sensors and actuators can be introduced. ☐

Structure of the list: *title of an example (a class of examples), references, a subsection with description of the example.*

- Trivial systems
  - A bouncing ball Lygeros and Sastry [1999], Simic et al. [2000], 2.1
  - Newton's cradle (colliding bodies) Mosterman [1999]
  - A falling rod Mosterman [1999]
  - A cat and mouse Chaochen et al. [1991], Anderson et al. [1993], 2.2
- Small and medium systems
  - Raibert's Hopper (complex version of the bouncing ball) Back et al. [1993]
  - A thermostat Lygeros and Sastry [1999], Alur et al. [1994], Henzinger [1996], Henzinger et al. [1997], Rönkkö and Ravn [1997b], Vereijken [1995], Jacobs [2000], 2.3
  - The fluid level control 2.4
    - One tank system, a pump on/off control (very popular) Alur et al. [1994] (similar case - valve control Henzinger et al. [1993])
    - One tank system model, automatic control of leaking and filling based on level Rönkkö and Ravn [1997a]
    - One tank system, a pump and a valve on/off control, composition of automata Heymann et al. [1997]
    - One tank, several fillers, on/off control Rönkkö and Ravn [1997a],
    - One tank, three outlets (one constant, two at the specified height), a controllable pump Cuzzola and Morari [2001]
    - Two tanks - one filler Lygeros and Sastry [1999], Simic et al. [2000], Heymann et al. [1997]
    - Three tanks - one filler Labinaz et al. [1996]
    - Two interconnected tanks Kowalewski et al. [1999]
    - Three interconnected tanks, two fillers (fixed for the first and the third tanks) Raisch et al. [1999]
  - A leaking gas burner Alur et al. [1994], Henzinger and Rusu [1998], Lamport [1993], 2.5
  - A chemical reactions control 2.6
    - A chemical reactions control Anderson et al. [1993], Jacobs [2000], Williams and Newell [1997], Philippe et al. [2000], 2.6
    - An evaporator Mosterman [1999], Kowalewski and Stursberg [1998], 2.6.1
  - A steam boiler Lygeros and Sastry [1999], Bishop et al. [1993], Lygeros et al. [1999], 2.7

  - A temperature control system Alur et al. [1994], Henzinger and Ho [1995], Henzinger and Rusu [1998], 2.8
  - Computer science examples extended to hybrid systems
    - The timers Jacobs [2000]
    - A mutual-exclusion protocol Alur et al. [1994]
    - A coffee-tea automaton Maler [2001], Larsen et al. [1997]
  - A game of billiards Alur et al. [1994]
  - A railroad gate control Puri and Varaiya [1995], Henzinger et al. [1995], Henzinger [1996], Henzinger et al. [1997], 2.9
  - Pursuit games 2.10
    - A helicopter, a pursuer and an evader Alur et al. [1997]
    - Runners and bridges Maler et al. [1995]
- Large, complex systems
  - A production line control van Beek and Rooda [2000], Daws and Yovine [1995], the filling-station (complex system) D. A. van Beek and J. E. Rooda [2000], 2.11
  - Mobile vehicles 2.12
    - One-dimensional movement 2.12.1
      - Automated highway system (AHS) - movement of vehicles in the platoon, etc.
    - Two-dimensional movement 2.12.2
      - A multi-robot coordination Alur et al. [1999]
      - Automated highway system (AHS) Lygeros and Sastry [1999], Alur et al. [2001], Lygeros et al. [1997], Lynch [1995], 2.13.3
      - A multi-modal control of a planar helicopter Koo et al. [2001], 2.13.2
      - STMS (Sea Traffic Management System) Godhavn et al. [1996], 2.13.1
    - Three-dimensional movement 2.12.3
      - Air Traffic Management (ATM) Tomlin et al. [1998], 2.13.2
      - An autonomous helicopter Branicky et al. [2000], 2.13.2
      - A flight vehicle management system (FVMS) Lygeros et al. [1999], 2.13.2

## 2.1 A bouncing ball

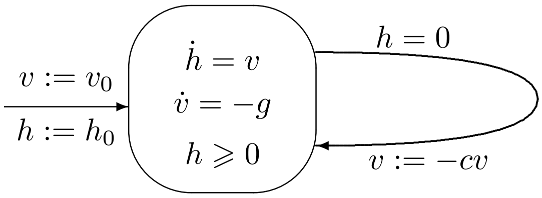

Figure 1: A Bouncing Ball

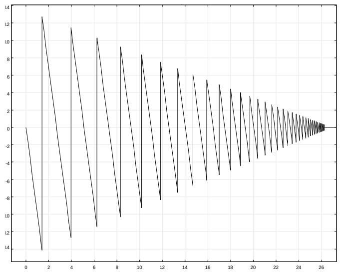

Figure 2: Trajectory of $v$

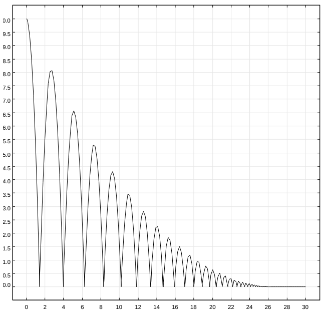

Figure 3: Trajectory of $y$

**Example 2.2** (Bouncing Ball). A bouncing ball is one of the trivial hybrid systems examples. This is a simplified model of an elastic ball that is bouncing and losing fraction of its energy with each bounce. The altitude of the ball is $y$, and $v$ is a vertical speed, $c$ is coefficient for lost energy. The ball moves according to flow conditions and at the bounce time variables are reassigned. The automaton for the bouncing ball is also very simple 1. $\square$

### 2.1.1 References

The bouncing ball example is presented in Lygeros and Sastry [1999] as an example of hybrid systems, and in Simic et al. [2000] several different versions of the bouncing ball are presented and analyzed.

## 2.2 A cat and mouse

A cat and mouse example is one of the first hybrid systems examples. It was very popular in 1991-1993, but later it lost popularity.

The idea is very simple: a cat chases a mouse (version from Manna and Pnueli [1993]). At time $T = 0$, the mouse starts running from a certain position on the floor in a straight line towards a hole in the wall, which is at a distance $X_0$ from the initial position. The mouse runs at a constant velocity $v_m$. After a delay of $\Delta$ time units, the cat is released at the same initial position and chases the mouse at velocity $v_c$ along the same path. Will the cat catch the mouse, or will the mouse find sanctuary?

This version of the example is not very interesting. But sometimes extended versions are used - the cat chases the mouse around a house in a cyclic path, and some doors could be closed to stop the cat or the mouse. But this example is quite different from the classical cat and mouse example, and may be mapped to one of the controller generations examples in Subsection 2.10.

### 2.2.1 References

One of versions of the cat and mouse example are presented in Chaochen et al. [1991], Anderson et al. [1993].

## 2.3 A thermostat

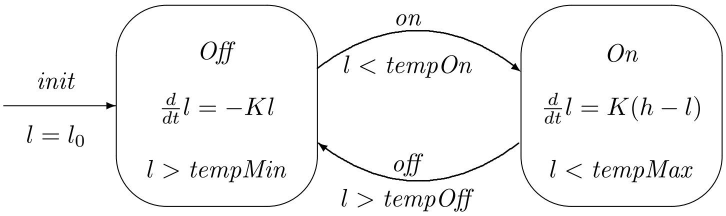

Figure 4: A Thermostat

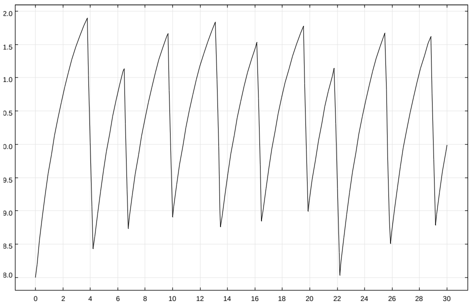

Figure 5: Change of the temperature

A thermostat is one of the main introductory examples of hybrid systems. The temperature of a room is controlled by a thermostat, which continuously senses the temperature and switches a heater on and off. The temperature changes are defined by differential equations. When the heater is off, the temperature decreases according to the exponential function $x(t) = \theta e^{Kt}$, where $t$ is time, $x$ is the temperature in the room, $\theta$ is the initial temperature, and $K$ is a constant determined by the room. When the heater is on, the temperature increases according to the function $x(t) = \theta e^{-Kt} + h(1 - e^{-Kt})$, where $h$ is a constant, that depends on the power of the heater. The temperature should be maintained between $minT$ and $maxT$. $minOn$ and $minOff$ are the minimal and temperatures, when the heater can be turned on and off.

A hybrid automaton based model of the thermostat is shown in Figure 4. The system starts with the temperature $inT$ (it should be right temperature, $minT \leqslant inT \leqslant maxT$). There are two locations

- **Off** - the heater is off, the temperature drops according to the flow equation.
- **On** - the heater is on, the temperature increases according to the flow conditions.

**Remark 2.3** (Extensions). Sometimes complexity of the system is extended - delays are added, a temperature is controlled by several heaters, in several rooms. Sometimes it is cooling, not warming. □

### 2.3.1 References

The thermostat example is very widely used in literature. A simple, classical thermostat, which is modelled by the two-states hybrid automaton is described in Lygeros and Sastry [1999], Alur et al. [1994], Henzinger [1996]. In Henzinger et al. [1997] several different versions of the thermostat are described, in deep analysis of different properties using HyTech is presented. In Rönkkö and Ravn [1997b] hybrid actions approach is used to model a simple thermostat. An ACP-style process algebra $ACP_{hs}$ (related with language $\chi$) is used to specify a simple thermostat in Vereijken [1995]. And in Jacobs [2000] a thermostat is specified using coalgebras with monoid actions.

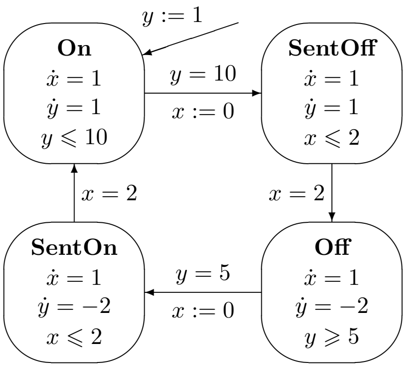

Figure 6: A water level control

## 2.4 A fluid level control

A fluid (often water) level control is a wide class of different examples - from a trivial fluid level control in a tank to a fluid level control in a complex network of tanks with plenty of valves and pumps. The main idea of the fluid level control examples is maintenance of the required fluid level in several vessels changing fluid input and drain speed, opening and closing (starting and stopping) valves and pumps (with delayed reaction of pumps, and sometimes valves). One of the most popular versions of the fluid level control examples is following (from Alur et al. [1994]).

A water level in a tank is controlled through a monitor, which continuously senses the water level and turns a pump on and off. The water level changes as a piecewise function over time. When the pump is off, the water level, denoted by variable $y$, falls by 2 cm per second; when the pump is on, the water level rises by 1 cm per second. It is required to keep water level between 1 and 12 cm. The pump receives signal from monitor delayed by 2 seconds. Thus, signals to turn pump on and off should be sent earlier, than the threshold is reached. The hybrid automaton model of this system is given. The system has four locations

- **On** - the pump is on
- **SendOff** - the pump is on, but signal to stop the pump is sent
- **Off** - the pump is off
- **SendOn** - the pump is off, but signal to start the pump is sent

### 2.4.1 References

The fluid control examples are widely presented in different papers, but often these examples are different - tanks are connected in different way, control of volume of the fluid can be accomplished differently (opening and closing valves of outlets, switching on and off pumps), etc.

The simplest version of the fluid control is a one tank system, where water flows out all the time at the constant speed, and a pump can be switched on and off, when it is necessary, but it takes some time to start and to stop the pump (delays are fixed). It is described in Alur et al. [1994]. A very similar example is given in Henzinger et al. [1993], but water flows in at a constant speed, and an output valve can be closed and opened, when required, and valve reacts without delays. In Rönkkö and Ravn [1997a] hybrid actions are used to model a one tank system, where leaking starts only when the maximum water level is reached, but some invariants for filling are also defined. A similar system, where a pump and a outlet valve can be turned on and off is described in Heymann et al. [1997]. A one tank system with several fillers is specified in Rönkkö and Ravn [1997a] using hybrid actions. More complex example with one tank is given in Cuzzola and Morari [2001], where two outlets are activated, when fluid level reaches some limits and one outlet allows a constant output all the time. Input flow varies between 0 and $u_{max}$.

In other type of the fluid level control examples some scheduling problems are modelled. For several tanks only one filler is available, and the objective is optimal use of it. In Lygeros and Sastry [1999], Simic et al. [2000], Heymann et al. [1997] a fluid level in two tanks should be maintained between some limits. In Labinaz et al. [1996] there are three tanks. Input and output flows are constant.

To complicate it even more, tanks can be interconnected. In Kowalewski et al. [1999] two interconnected tanks, which are situated at different heights, are analyzed. A pump, the valve between tanks and an output valve can be switched on and off. The dynamics of such system are far from trivial. And in Raisch et al. [1999] required fluid level should be maintained in three interconnected tanks (from the first tank fluid flows to the second, from the second to the third, and from the third - out). The first and the third tank can filled turning the first and the third pumps on (the pumps are operated independently).

## 2.5 A leaking gas burner

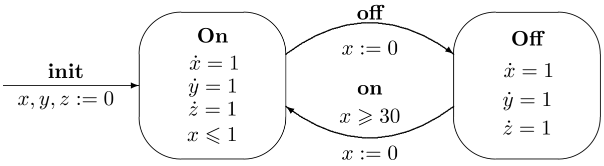

Figure 7: A leaking gas burner

A leaking gas burner is a simple example of a valve, which controls a gas supply to a burner. There are several different versions of the leaking gas burner examples: sometimes only leaking and non-leaking of gas is considered, and sometimes burning period. Very often simpler version of the leaking gas burner is considered, where two properties are analyzed

- A continuous leaking period cannot extend beyond some specified amount of time units (leaking_time).
- The accumulated time of leakage is at most some specified amount of time (accumulated_leakage) in any interval of at least 60 seconds.

Such system can be modelled by the two-states hybrid automaton (Figure 7) from Alur et al. [1994]. In this model $\text{leaking\_time} = 1, \text{accumulated\_leakage} = 3$. The clock $x$ records the time spent in the current location, the integrator $z$ records the cumulative leakage time and the clock $y$ records total elapsed time.

**Remark 2.4** (Memory). Such system can be considered as a system with implicit memory, because evolution of the system depends on previous behaviour of it, but it is not explicitly related with locations. In the fluid level control 2.4 and the thermostat 2.3 examples all necessary previous behaviour information is encoded as locations. □

**Remark 2.5** (Continuous dynamics). The above presented version of the leaking gas burner can be analyzed as timed system (timed automaton), because all continuous variables change with speed 1, threfore they can be modelled as clocks. □

### 2.5.1 References

The leaking gas burner is quite popular example, it can be encountered in many papers. In Alur et al. [1994], Henzinger and Rusu [1998] a simple leaking gas burner is analyzed. And in Lamport [1993] temporal logic of actions is used to model a simple gas burner.

## 2.6 Chemical reaction control

Chemical reactions control is a wide group of the hybrid systems examples. A scenario is following - there are several chemical substances, and they should be mixed in the specified proportions (quantities), sometimes following specified order. In addition, temperature should be controlled (all the time, or only after mixing substances). Sometimes mixing device could be used (it could be turned on and off). And, at some time moment, the resulting chemical substance should be removed. Sometimes, additional vessel for produced chemical substance is used, which should be ready to receive the product. In addition, several different sensors could be present for the temperature, the fluid level and the concentration measurements. Such systems are quite complicated, because it is necessary to follow several different objectives. In literature, examples are quite different, sometimes abstractions of real-life systems are presented.

In more complicated versions delays for the actuators and the sensors can be introduced.

### 2.6.1 An evaporator vessel

An evaporator vessel could be considered as a chemical reaction 2.6 with some additional control - a steam pressure valve could be used to control pressure in the vessel. Some versions of the evaporator vessel are similar to the steam boiler problem.

Sometimes the evaporator vessel problem is abstracted to the fluid level control problem. And the fluid level control examples are discussed in the fluid level control subsection 2.4.

### 2.6.2 References

Two types of the chemical reactions examples were separated in 2.6 and 2.6.1. And both classes of the examples are common in the scientific literature.

A classical chemical reaction is modelled and analyzed in Anderson et al. [1993]. In this example a special attention is given to the analysis of the safety requirements. In Jacobs [2000] chemical reactions are specified and analysed using coalgebras with monoid actions, which capture the influence of time on the state space. A simple reactor for waste water treatment is modelled and analysed in Williams and Newell [1997]. And in Philippe et al. [2000] Pontryagin's Maximum Principle is used to minimize the overall operating time of the system, which is modelled using hybrid automaton.

An evaporator vessel, as a part of a chemical reaction is modelled in Mosterman [1999]. And a complex batch evaporator is analyzed in Kowalewski and Stursberg [1998].

## 2.7 A steam boiler

A steam boiler is one of the oldest hybrid systems examples. It is not very frequent, but every self-respecting hybrid systems specialist at least once in his life tried to specify a steam boiler. The steam boiler consists of a vessel for a fluid. The vessel is heated, the fluid evaporates. Amount of the fluid should be maintained between the specified boundaries using a pump (several pumps), and sometimes a valve (valves). Different sensors can be used: temperature, fluid level, pressure, steam rate. Sensors and actuators could fail. Safety, and sometimes optimality of the system should be maintained. Several different operation modes could be used (at least startup, normal operation and shutdown).

### 2.7.1 References

A steam boiler is specified and analyzd in Lygeros and Sastry [1999]. In Lygeros et al. [1999] controler generation problem for a steam boiler is considered. A step-wise development and verification of the Generic Boiler System is presented in Bishop et al. [1993].

## 2.8 A temperature control

Temperature control examples can be incorporated in to the chemical reactions control 2.6 and/or the steam boiler 2.7 control frameworks. The goal of the temperature control system is to maintain a temperature of a coolant in a system (a reactor) between some specified bounds by moving independent control rods. If it is not possible to maintain necessary temperature, the system should be shut down. Sometimes, additional sensors and actuators could be used (valves, additional fluid, etc.).

### 2.8.1 References

A reactor (nuclear reactor) temperature control problem is analyzed in Alur et al. [1994], Henzinger and Ho [1995], Henzinger and Rusu [1998] using hybrid automaton.

## 2.9 Railroad gate control

The railroad gate control models a gate with a controller on a circular train track and a train. The controller should issue commands to the gate to close and open, when it is necessary (based on information about train movement). The version from Henzinger [1996] is used in this paper.

Initially the speed of the train is between 40 and 50 meters per second. At the distance (represented by variable $x$) of 1000 meters from the gate, the train issues an *approach* event and may slow down to 30 meters per second. At the distance of 100 meters past the gate it issues an *exit* event. The circular track is between 2 and 5 kilometres long. When an *approach* event is received, the controller issues a *lower* event within $u$ seconds delay, and when an *exit* event is received, the controller issues a *raise* event within $u$ seconds. The elapsed time is represented by variable $z$. And the gate initially is open (position of the gate in degrees is 90, and it is represented by variable $y$). When a *lower* event is received, the gate starts closing at the rate of 9 degrees per second, and when *raise* event is received, the gate starts opening at the same rate. The purpose of model is to find $u$ - the reaction delay. Hybrid automatons of the train, the gate and the controller are presented in Figure 8.

### 2.9.1 References

In application of HyTech and hybrid automaton theory for the rail-road gate control is presented in Henzinger [1996], Henzinger et al. [1995, 1997]. In Puri and Varaiya [1995] several different approaches are used - with a simplified differential inclusion and using a timed automaton.

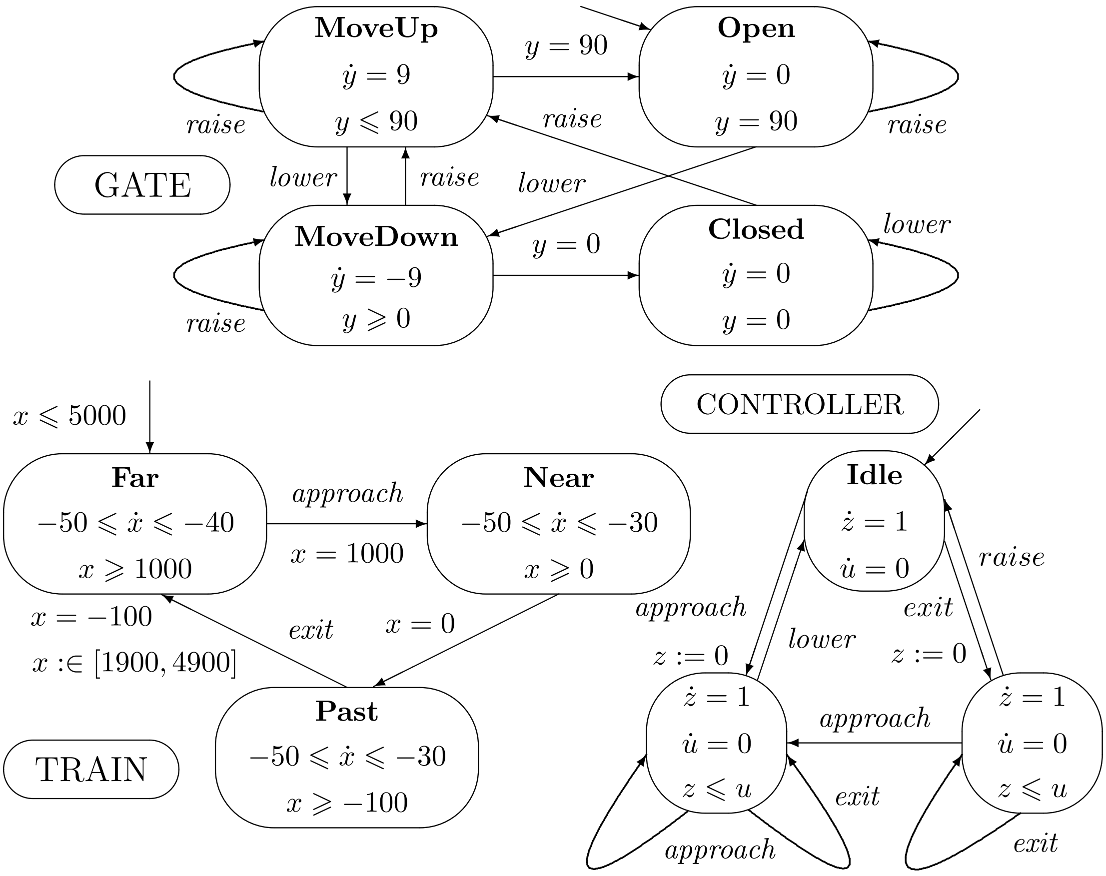

Figure 8: Train, gate and controller automata

## 2.10 Pursuit games

Pursuit games class of examples, which is often used to illustrate control problems. It is easy to guess type of examples from the title of the class - in these examples a pursuer (pursuers) tries to catch his opponent (an evader). As a simplified version of such games, the cat and mouse example can be considered (described in Subsection 2.2). But often, these examples are more complicated, and it is not easy to find winning strategies.

### 2.10.1 References

In Alur et al. [1997] a simple two-person pursuit game is modelled. There is a pursuer in a golf cart, which chases an evader on a circular track. Hybrid automaton is used to model it, and controler generation problem for such system is analyzed.

In Maler et al. [1995] a two-oponents game is analyzed. There are several roads (two) and one player should choose free road, another should try to block the road. The objective of the first player is to reach final point within some fixed time, and the objective of the second player is opposite. A real-time controler generation algorithms for the system are analyzed.

## 2.11 A production line control

A production line control covers a wide area of hybrid systems. It could be a control of a conveyor and robots, or it could be a control of filling station with restrictions on a filling speed, a filling time, etc. Modelling, analysis and controllers generation for such systems has several objectives:

- to check, is it possible to fulfil the requirements with given constraints;
- to find optimal schedule;
- to generate controller, which would dynamically react to changes in the system and steer it in the "right" direction;
- to prepare scenarios for recovery from different failures;
- etc.

In different examples, different objectives are analyzed. Some of examples are abstractions of real-life production systems.

### 2.11.1 References

In different papers different types of production lines are analyzed, but objectives are similar. In Daws and Yovine [1995] only a part of production line - one D-Robot is analyzed using multirate timed automaton and Kronos tool. Model of a transport system is presented in van Beek and Rooda [2000]. And in D. A. van Beek and J. E. Rooda [2000] a filling station model is analysed with the emphasis on the discrete-event aspects.

## 2.12 Mobile vehicles

Mobile vehicles control is very wide and vague class of examples. Different types of vehicles are considered, they move in different environments (with different dimensions), they have different degrees of freedom. Control objectives also are different. Often such systems are modelled in a hierarchical way - there are several central control centres, but in addition, agents can interact with each other without the centres of mediation. Often agents behaviour is classified to different modes, and abstracted to lesser dimensionality. Several types of systems fit into the mobile vehicles framework:

- Automated Highway Systems (AHS)
- Air Traffic Management Systems (ATM), Flight Vehicle Management Systems (FVMS), Autonomous Flight Vehicles
- Sea Traffic Management Systems (STMS), Autonomous Underwater Vehicles
- Mobile Robots movement coordination systems

Most of such systems have following configuration and behaviour

- An autonomous or semi-autonomous vehicle, with following responsibilities
  - Movement of single vehicle, according to the comfort and physical requirements
  - Movement of single vehicle, following an optimal route, avoiding obstacles
  - Movement in a group of other vehicles, avoiding collisions
  - Communication with other vehicles (negotiations, etc.)
- A movement coordination centre
  - A traffic control
  - An interaction with vehicles

### 2.12.1 One-dimensional movement

In one-dimensional movement analysis of the following problems is considered

- Movement of single vehicle: acceleration, cruising and braking.
- Joining a group of vehicles, safe movement with and in the group of vehicles, splitting from the group of vehicles.

Often, such analysis is a part of bigger systems analysis, or some introductory research of mobile systems Alur et al. [2001], Lygeros et al. [1997].

### 2.12.2 Two-dimensional movement

Two-dimensional movement analysis is used for Sea Traffic Management, Automated Highway Systems, abstraction of Flying Vehicles. In addition to one-dimensional case, following objectives can be added

- Avoiding obstacles
- Joining and splitting from the platoons (groups) in two dimensional plane, other manoeuvers (safe, comfortable and efficient)
- Negotiations with other vehicles in order to avoid collisions
- Corresponding reaction to the control centre recommendations and demands

Wider description of such systems can be found in Alur et al. [1999], Koo et al. [2001], Lynch [1995], Lygeros and Sastry [1999]

### 2.12.3 Three-dimensional movement

The most complicated cases of movement, in a meanwhile, is a movement in a three dimensional space. It is Air Traffic Control and Flight Vehicle Management, but sometimes movement of the Underwater Vehicles considered too. Objectives are the same, as for one- and two-dimensional cases, but dynamics are more complicated. Some of papers, where these problems are described in Lygeros et al. [1999], Tomlin et al. [1998], Branicky et al. [2000].

## 2.13 Special cases of mobile vehicles

Additional section - mobile vehicles examples description.

### 2.13.1 Sea traffic management

A sea traffic management and sea vehicles management is very closely related with air traffic management and flight vehicles management. Several different classes of examples can be distinguished - underwater vehicles and above water vehicles. Underwater vehicles operate in 3-dimensional space, like flight vehicles 2.13.2, and above water vehicles operation is similar to Automated Highway System operation 2.13.3.

### 2.13.2 Air traffic and flight vehicles management

An air traffic management and flight vehicles management are very closely connected problems. It can be regarded, as a three-dimensional AHS 2.13.3 problem with stricter requirements and more complicated dynamics. In general, two different types of examples belong to this class of hybrid systems - flight vehicle control and air traffic control. These two types are closely related - air traffic control is often hierarchically higher part of such systems.

### 2.13.3 AHS (Automated Highway System)

Automated Highway System is very commong example of large scale hybrid systems. Sometimes it is used as a simplification of flight control system (one or two dimensions against three). In different papers slightly different versions of the system are presented, but general objectives consist of two layers - vehicle control and traffic control, where vehicle control defines a control of single vehicle according to environment and "big brother" demands, and a traffic control defines interaction between single vehicle and a control centre, which provides routeing and other information. Some of problems, which should be solved by AHS, are,

- Getting and maintaining safe speed.
- Getting and maintaining safe distances between vehicles.
- Typical manoeuvers modes support - lane changes, merging and diverging traffic flow, etc.
- Conflicts resolution between different requirements, as manoeuvers, safe speeds, safe distances.
- Vehicles routes (trajectories) tracking.
- *Hand-off* of vehicles (interaction between control stations, when vehicle changes geographical location).
- Accident prevention.
- Routeing.

Every imaginable requirement is a part of AHS - safety requirements, robustness requirements, optimality requirements, etc.

One of the popular approaches to analyze this problem is a hierarchical model, where vehicles are presented as *agents*. Each agent has a well-defined interface, which is used for communication with other agents and the traffic management system. An agent (a vehicle) behaviour is decomposed to the different modes and submodes. Different continuous dynamics are defined according to the different requirements (safety, comfort, optimality, etc.) for modes and submodes. As a higher hierarchical behaviour level - groups of vehicles can be formed (*platoons*). And in addition, control centre can issue different demands for vehicles and platoons.

# 3 Classification of Hybrid Systems

Hybrid systems can be classified in many different ways. Classification can be based on several different approaches: possible customers and internal properties. Classification by possible customers is listed.

- An industry (chemical, automotive etc.).
- An organisation, which can provide financing for research (government, military, industry/industries etc.).

But such type of classification is not is not considered here.

Classification by internal properties of hybrid systems is more interesting. Systems can be classified by many different aspects.

- Complexity (dynamics, continuous dynamics, discrete dynamics).
- Discrete dynamics properties (switching types).
- Continuous dynamics properties (flow types).

| **Category** | **Explanation** |
|---|---|
| Autonomous switching | Continuous flow changes on hitting specified region border |
| Autonomous jump (reset) | Continuous component changes on hitting specified region border |
| Controlled switching | Continuous flow changes changes in response to a control command |
| Controlled jump (reset) | Continuous component changes in response to a control command |

Table 1: Classification of hybrid systems by M.S. Branicky, V.S. Borkar and S.K. Mitter

Different properties of hybrid systems can be interesting in different cases. Several different classification categories are proposed in literature. In Table 2 presented one of possible classifications of hybrid systems from Labinaz et al. [1996].

A different classification of hybrid system is given in Branicky et al. [1994] by M. S. Branicky, V. S. Borkar and S. K. Mitter. In this classification, two different aspects of hybrid systems are considered - causality of discrete events (autonomous or controlled) and continuous dynamics change. Classification categories presented in Table 1.

Pieter J. Mosterman also separates several hybrid simulation phenomena in Mosterman [1999]. It is presented in Table 3.

| **Category** | **Sub-category** | **Explanation** |
|---|---|---|
| Sampling | Regular | Measurements of states are assumed to be known (and available all the time) |
| | Continuous | Measurements are taken based on some predetermined, fixed sampling period |
| Continuous Dynamics | Linear | Dynamics are defined by linear equations |
| | Nonlinear | Ddynamics are defined by nonlinear equations |
| Determinism (Cont.dyn.) | Deterministic | The evolution of systems is single valued |
| | Nondeterministic | The evolution of system is multi-valued |
| Determinism (Discr.dyn.) | Deterministic | Each state is mapped to single next state |
| | Nondeterministic | State is mapped to a set of states |
| Control action | Continuous | Continuous-valued control functions with continuous domain and range |
| | Discrete | Function with discrete range and continuous or discrete domain |
| | Combined | Combination of discrete and continuous control |
| Specifications | Continuous | Specifications based on continuous time behaviour and/or variables |
| | Discrete | Specifications based on discrete event behaviour and/or variables |
| | Combined | Combination of continuous and discrete |

Table 2: Classification of hybrid systems by G. Labinaz, M. M. Bayoumi and K. Rudie

| Category | Type | Explanation |
|---|---|---|
| Events | Time | events are generated at predetermined times |
| | State | events occur because system crosses some thresholds, the time of their occurrence are not known a priori - they need to be detected |
| simulation model | Dynamical blocks | Blocks of sorted and solved equation may simply appear or disappear (vehicle entering or leaving highway), and, therefore, can be dynamically added/removed |
| | Changeable cont. dynamics | In some cases equations can be replaced by others, changing computational causality, and the system of equations may have to be sorted again |
| | Constraint changes | In other cases, algebraic constraints between state variables may become active and the system of equations needs to be solved again (the rod making contact to the floor) |
| Reinitialization | Explicit | Change explicitly specified by user by a new initial state |
| | Integrated | The system of equations may have to be integrated to derive physically consistent initial values for a new mode. This ensures conservation of the thermodynamic extensity holds |
| Event iteration | Invariant | State vector is invariant across the entire iteration |
| | Updated | State vector is updated after each iteration step |
| Chattering | | Fast switching between several modes |
| Dirac pulses | | Non-continuous changes for continuous variables; if their magnitudes are numerically approximated, comparison maybe affected by non-Dirac type variables |

Table 3: Classification of hybrid simulation phenomena by Pieter J. Mosterman

# References

Rajeev Alur, Costas Courcoubetis, Thomas A. Henzinger, Pei-Hsin Ho, Xavier Nicollin, Alfredo Olivero, Joseph Sifakis, and Sergio Yovine. The algorithmic analysis of hybrid systems. In G. Cohen and J.-P. Quadrat, editors, *Proceedings of the 11th International Conference on Analysis and Optimization of Systems: Discrete-event Systems*, volume 199 of *LNCIS*, pages 331–351. Springer, 1994.

Rajeev Alur, Thao Dang, Joel M. Esposito, Rafael B. Fierro, Yerang Hur, Franjo Ivancic, Vijay Kumar, Insup Lee, Pradyumna Mishra, George J. Pappas, and Oleg Sokolsky. Hierarchical hybrid modeling of embedded systems. In *EMSOFT*, pages 14–31, 2001. URL `citeseer.nj.nec.com/451283.html`.

Rajeev Alur, Joel M. Esposito, Moon Joo Kim, Vijay Kumar, and Insup Lee. Formal modeling and analysis of hybrid systems: A case study in multi-robot coordination. In *World Congress on Formal Methods (1)*, pages 212–232, 1999. URL `citeseer.nj.nec.com/alur99formal.html`.

Rajeev Alur, Thomas A. Henzinger, and Howard Wong-Toi. Symbolic analysis of hybrid systems. In *Proceedings of 36thIEEE Conference on Decision and Control*, pages 702–707, San Diego, CA, December 1997. IEEE Computer Society Press.

Tom Anderson, Rogé de Lemos, John S. Fitzgerald, and Amer Saeed. On formal support for industrial-scale requirements. In Grossman et al. [1993], pages 426–451. ISBN 3-540-57318-6.

Panos J. Antsaklis, Wolf Kohn, Michael D. Lemmon, Anil Nerode, and Shankar Sastry, editors. *Hybrid Systems V*, volume 1567 of *LNCS*, 1999. Springer. ISBN 3-540-65643-X.

Panos J. Antsaklis, Wolf Kohn, Anil Nerode, and Shankar Sastry, editors. *Hybrid Systems II*, volume 999 of *LNCS*, 1995. Springer. ISBN 3-540-60472-3.

Allen Back, John Guckenheimer, and Mark Myers. A dynamical simulation facility for hybrid systems. In Grossman et al. [1993], pages 255–267. ISBN 3-540-57318-6.

Maria Domenica Di Benedetto and Alberto L. Sangiovanni-Vincentelli, editors. *Hybrid Systems: Computation and Control, 4th International Workshop, HSCC 2001, Rome, Italy, March 28-30, 2001, Proceedings*, volume 2034 of *LNCS*, 2001. Springer. ISBN 3-540-41866-0.

Peter Bishop, Glenn Bruns, and Stuart Anderson. Stepwise development and verification of a boiler system specification. In *International Workshop on the Design and Review of Software Controlled Safety-related Systems*, Ottawa, Canada, June 28-29 1993.

Michael S. Branicky, Vivek S. Borkar, and Sanjoy K. Mitter. A unified framework for hybrid control: Background, model, and theory. In *Proceedings of the 33rd IEEE Conference of Desicion and Control*, Lake Buena Vista, FL, December 14-16 1994.

Michael S. Branicky, Tor A. Johansen, Idar Petersen, and Emilio Frazzoli. On-line techniques for behavioral programming. In *Proceedings of 39thIEEE Conference on Decision and Control*, pages 1840–1845, Sydney, AUSTRALIA, December 2000. IEEE Computer Society Press. URL `citeseer.nj.nec.com/420838.html`.

Zhou Chaochen, C. A. R. Hoare, and Anders P. Ravn. A calculus of durations. *Information Proc. Letters*, 40(5):269–276, Dec. 1991.

Francesco Alessandro Cuzzola and Manfred Morari. A generalized approach for analysis and control of discrete-time piecewise affine and hybrid systems. In Benedetto and Sangiovanni-Vincentelli [2001], pages 189–203. ISBN 3-540-41866-0.

D. A. van Beek and J. E. Rooda. Languages and applications in hybrid modelling and simulation: Positioning of chi. *Control Engineering Practice*, 8(1):81–91, 2000.

Conrado Daws and Sergio Yovine. Two examples of verification of multirate timed automata with KRONOS. In *Proceedings of the 16th IEEE Real-Time Systems Symposium (RTSS'95),* Pisa, Italy, pages 66–75, 1995. URL `citeseer.nj.nec.com/daws95two.html`.

John-Morten Godhavn, Trygve Lauvdal, and Olav Egeland. Hybrid control in sea traffic management systems. In Rajeev Alur, Thomas A. Henzinger, and Eduardo D. Sontag, editors, *Hybrid Systems*, volume 1066 of *LNCS*, pages 149–160. Springer, 1996. ISBN 3-540-61155-X.

Robert L. Grossman, Anil Nerode, Anders P. Ravn, and Hans Rischel, editors. *Hybrid Systems*, volume 736 of *LNCS*, 1993. Springer. ISBN 3-540-57318-6.

Thomas A. Henzinger. The theory of hybrid automata. In *LICS*, pages 278–292. IEEE Computer Society Press, 1996.

Thomas A. Henzinger and Pei-Hsin Ho. A note on abstract interpretation strategies for hybrid automaton. In Antsaklis et al. [1995], pages 265–293. ISBN 3-540-60472-3.

Thomas A. Henzinger, Pei-Hsin Ho, and Howard Wong-Toi. HyTech: the next generation. In *RTSS*, pages 56–65. IEEE Computer Society Press, 1995. ISBN 0-8186-7337-0.

Thomas A. Henzinger, Pei-Hsin Ho, and Howard Wong-Toi. HyTech: a model checker for hybrid systems. In Orna Grumberg, editor, *CAV*, volume 1254 of *LNCS*, pages 460–463. Springer, 1997. ISBN 3-540-63166-6.

Thomas A. Henzinger, Zohar Manna, and Amir Pnueli. Towards refining temporal specifications into hybrid systems. In Grossman et al. [1993], pages 60–67. ISBN 3-540-57318-6.

Thomas A. Henzinger and Vlad Rusu. Reachability verification for hybrid automata. In Thomas A. Henzinger and Shankar Sastry, editors, *HSCC*, volume 1386 of *LNCS*, pages 190–204. Springer, 1998. ISBN 3-540-64358-3. URL `citeseer.nj.nec.com/henzinger98reachability.html`.

Michael Heymann, Feng Lin, and George Meyer. Synthesis of minimally restrictive legal controllers for a class of hybrid systems. In Panos J. Antsaklis, Wolf Kohn, Anil Nerode, and Shankar Sastry, editors, *Hybrid Systems*, volume 1273 of *LNCS*, pages 134–159. Springer, 1997. ISBN 3-540-63358-8.

Bart Jacobs. Object-oriented hybrid systems of coalgebras plus monoid actions. *Texts in TCS. An EATCS Series*, 239(1):41–95, 2000. URL `citeseer.nj.nec.com/40461.html`.

Tak-John Koo, George J. Pappas, and Shankar Sastry. Mode switching synthesis for reachability specifications. In Benedetto and Sangiovanni-Vincentelli [2001]. ISBN 3-540-41866-0. URL `citeseer.nj.nec.com/koo01mode.html`.

Stefan Kowalewski and Olaf Stursberg. The batch evaporator: A benchmark example for safety analysis of processing systems under logic control. In *Proceedings 4th Workshop on Discrete Event Systems (WODES)*, pages 302–307. IEE, London, 1998. URL `http://www-verimag.imag.fr/VHS/year1/cs11d.ps`.

Stefan Kowalewski, Olaf Stursberg, Martin Fritz, Holger Graf, Ingo Hoffmann, Jörg Preußig, Silke Simon, Heinz Treseler, and Manuel Remelhe. A case study in tool-aided analysis of discretely controlled continuous systems: the two tanks problem. In Antsaklis et al. [1999], pages 163–187. ISBN 3-540-65643-X. URL `citeseer.nj.nec.com/stursberg97case.html`.

Gino Labinaz, Mohamed M. Bayoumi, and Karren Rudie. Modeling and control of hybrid systems: A survey. In *IFAC 13th Triennial World Congress*, San Francisco, USA, 1996. URL `citeseer.nj.nec.com/labinaz96modeling.html`.

Leslie Lamport. Hybrid Systems in TLA+. In Grossman et al. [1993], pages 77–102. ISBN 3-540-57318-6. URL `citeseer.nj.nec.com/lamport93hybrid.html`.

Kim G. Larsen, Bernhard Steffen, and Carsten Weise. Continuous modelling of real time and hybrid systems: From concepts to tools. *International Journal on Software Tools for Technology Transfer*, 1(1): 64–85, December 1997.

John Lygeros, Datta Godbole, and Shankar Sastry. A design framework for hierarchical, hybrid control. Technical report, Intelligent Machines and Robotic Laboratory, University of California, Berkeley, 1997.

John Lygeros and Shankar Sastry. Hybrid systems: Modeling, analysis & control, ee291, 1999. URL `paleale.eecs.berkeley.edu/~lygeros/Teaching/ee291E.html`.

John Lygeros, Claire Tomlin, and Shankar Sastry. Controllers for reachability specifications for hybrid systems. *Automatica*, 35(3), March 1999. URL `citeseer.nj.nec.com/lygeros99controller.html`.

Nancy A. Lynch. Modelling and verification of automated transit systems, using timed automata, invariants and simulations. Technical Report MIT/LCS/TM-545, MIT Laboratory for Computer Science, 1995. URL `citeseer.nj.nec.com/156049.html`.

Oded Maler. Control from computer science. In *IFAC Symposium Nonlinear Control (NOLCOS'01)*, 2001. URL `citeseer.nj.nec.com/443949.html`.

Oded Maler, Amir Pnueli, and Joseph Sifakis. On the synthesis of discrete controllers for timed systems (extended abstract). In Ernst W. Mayr and Claude Puech, editors, *STACS*, volume 900 of *LNCS*, pages 229–242. Springer, 1995. ISBN 3-540-59042-0. URL `citeseer.nj.nec.com/302111.html`.

Zohar Manna and Amir Pnueli. Verifying hybrid systems. In Grossman et al. [1993], pages 4–35. ISBN 3-540-57318-6.

Pieter Mosterman. An overview of hybrid simulation phenomena and their support by simulation packages. In Frits W. Vaandrager and Jan H. van Schuppen, editors, *HSCC*, volume 1569 of *LNCS*, pages 165–177. Springer, 1999. ISBN 3-540-65734-7.

Manon Philippe, Valentin-Roubinet Claire, and Gilles Gérard. Optimal control of hybrid dynamical systems with the maximum principle: Application to a non linearchemical process. In *Proceedings of 39thIEEE Conference on Decision and Control*, Sydney, AUSTRALIA, December 2000. IEEE Computer Society Press.

Anuj Puri and Pravin Varaiya. Hybrid systems ii. In Antsaklis et al. [1995], pages 359–369. ISBN 3-540-60472-3. URL `citeseer.nj.nec.com/puri94verification.html`.

Jörg Raisch, Eberhard Klein, Siu O'Young, Christian Meder, and Alexander Itigin. Approximating automata and discrete control for continuous systems - two examples from process control. In Antsaklis et al. [1999], pages 279–303. ISBN 3-540-65643-X.

Mauno Rönkkö and Anders P. Ravn. Hybrid action systems. Technical Report TUCS TR-110, Turku Centre for Computer Science, May 1997a. URL `citeseer.nj.nec.com/123739.html`.

Mauno Rönkkö and Anders P. Ravn. Switches and jumps in hybrid action systems. Technical Report TUCS-TR-152, Turku Centre for Computer Science, 1997b. URL `citeseer.nj.nec.com/151567.html`.

Slobodan N. Simic, Karl Henrik Johansson, Shankar Sastry, and John Lygeros. Towards a geometric theory of hybrid systems. In Nancy A. Lynch and Bruce H. Krogh, editors, *HSCC*, volume 1790 of *LNCS*, pages 421–436. Springer, 2000. ISBN 3-540-67259-1. URL `citeseer.nj.nec.com/simic00towards.html`.

Claire J. Tomlin, George J. Pappas, and Shankar Sastry. Conflict resolution for air traffic management : A study in multi-agent hybrid systems. In *IEEE Transactions on Automatic Control*, pages 509–521, April 1998. URL `citeseer.nj.nec.com/article/tomlin98conflict.html`.

D. A. van Beek and J. E. Rooda. Semantics of state-events in hybrid languages. In *Third IMACS Symposium on Mathematical Modelling*, pages 421–424, Viena, Austria, 2000. URL `citeseer.nj.nec.com/vanbeek00semantics.html`.

Jan J. Vereijken. A process algebra for hybrid systems. In *The Second European Workshop on Real-Time and Hybrid Systems*, Grenoble, France, May 1995.

Russell Williams and Robert Newell. Hybrid analysis as a batch process controller design tool. In *Proceedings of Control'97*, Sydney, Australia, October 1997.