HYBRID SYSTEMS - I

Jean DELLA DORA
Laboratoire de Modélisation et Calcul
UJF-INPG
Grenoble

Contents

Introduction

- A rough definition of hybrid systems, or hybrid dynamical systems is the following : **systems that combine discrete event mode logic and continuous time dynamics within each node**

- The goal of hybrid system "theory" is to be formal model for large, complex and heterogeneous physical systems.

- Three components are part of the model :

  1. The model for the physical part of the system is based on set of differential equations. In general algebraic equations or inequations.
     But hudge systems of "simple" equations.

  2. The mode switching logic is expressed as a finite state machine.

3. The architecture of the sensor processing system is capture by dataflow models.

- Control systems has been, in the past, a very important part of engineering methodology. Today the standard technique is driven by **Computer control** -reduction of the cost of computing, generalisation of embedded softwares, need for real-time application .

- But, actually, most of the available softwares are based on **simulation**. The most important issue is about safety and there is a need for formal methods taking care of such issue.
  Why not algebraic methods ?
  Why not an extension of the realm of computer algebra ?
  Why not control computer algebra ?.

Some actual projects

- **The Ptolemy project**
  http://ptolemey.eecs.berkeley.edu

- **Modelica**
  Lund Sweden
  http://www.modelica.org

- **SHIFT**
  http://www.path.berkeley.edu/shift/

- **Dymola**
  http://www.dynasim.se

- **OmSim**
  http://www.control.lth.se/ cace/omsim.html/

- **Stateflow**
  http://www.mathworks.com/product/stateflow.

Some actual exemples

- **Power transmission networks**
- **Embedded systems and real-time applications**
- **Chemical plants**
- **Genetic networks**

Some recalls

How can we build a model of a "simple" system ?

- We must highlight the possible **states** of the system.

Exemple 1 *The temperature in a room* (x) *controlled by a switching heater - on and off-*

Electric circuit with switches.
Chemical plants with valves and pumps.

The result of these observations is a finite (or countable) set

$$\mathcal{Q} = \{q_1, \ldots, q_m\}$$

of states.

- The second step is related to the **dynamic** of the system.
  If the system is in state q then under the action of the outside world it will jump to another state.
  The simple way to record the dynamic is to build the list of outside events acting on the states.
  At most in the begining, let's suppose that the number of these outside events is finite.
  Let
  $$\Sigma = \{\sigma_1, \ldots, \sigma_n\}$$
  be the set of such events.

- Knowing $\mathcal{Q}$ and Σ we are able to describe the way Σ acts on $\mathcal{Q}$.

- For doing this we choose the following presentation :

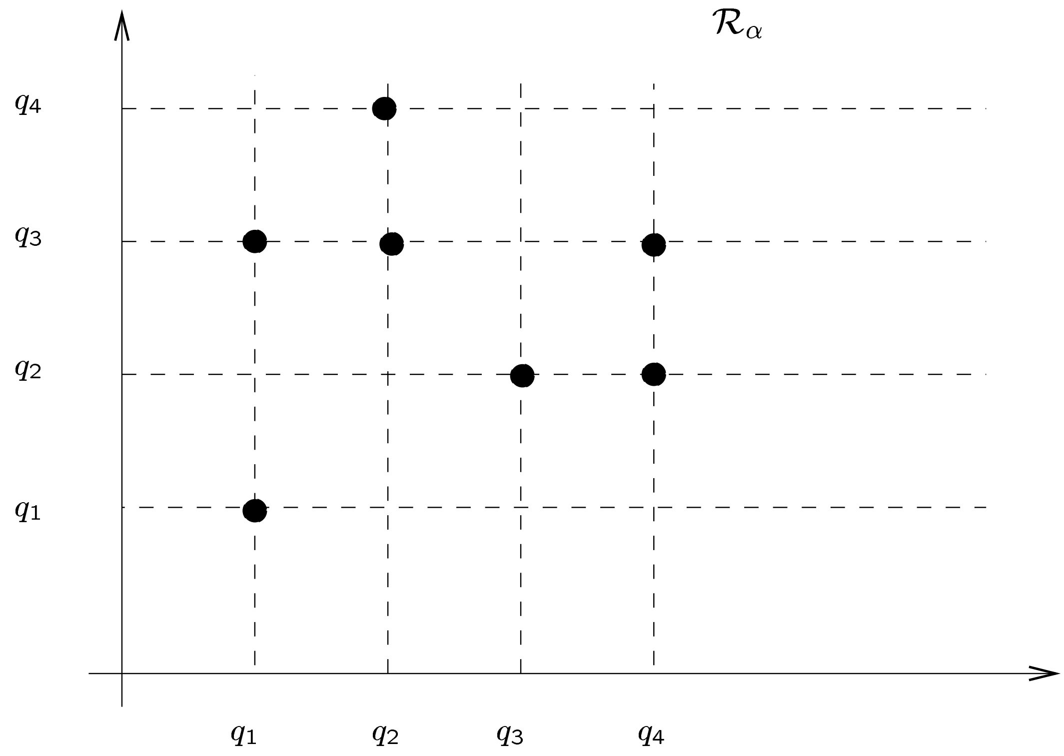
If the system is subject to an event (say $\alpha \in \Sigma$) then there is a **Relation**

$$\mathcal{R}_\alpha \subset \mathcal{Q} \times \mathcal{Q}$$

such that, if the system is in state q and if $q \in \mathcal{D}om(\mathcal{R}_\alpha)$ them the system can jump to any of the states belonging to the **set**

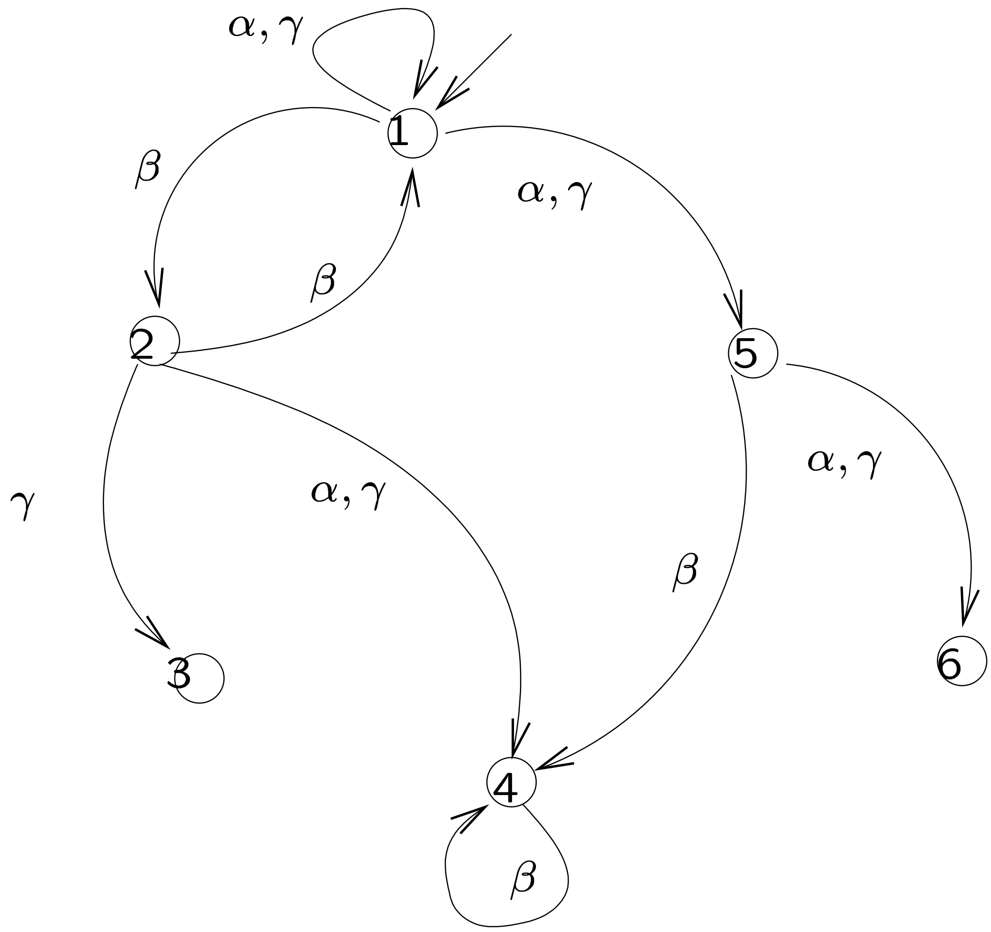
$$\mathcal{R}_\alpha(q).$$

The following picture shows the graph of the relation $\mathcal{R}_\alpha(q)$.



Then the dynamic is described (in extension) by

$$\mathcal{R} = \bigcup_{\alpha \in \Sigma} \mathcal{R}_\alpha$$



Exemple 2

$$\mathcal{Q} = \{1, \ldots, 6\}$$

$$\Sigma = \{\alpha, \beta, \gamma\}$$

$$\mathcal{R}_\alpha = \{(1, \alpha, 1), (1, \alpha, 5), (2, \alpha, 4), (5, \alpha, 6)\}.$$

Two sets are associated to the system :

$$Q^\omega$$

and

$$\Sigma^\omega$$

the first is the set of all infinite words that can be written using the alphabet $\mathcal{Q}$ (and starting with q_1 a **marked state**), the second is the set of infinite words build on Σ.
From the dynamical point of view, $\mathcal{Q}^\omega$ can be seen as the **state space** of the system.
A word $\widehat{q} \in Q^\omega$ represents a **state trajectory**.

But all the elements of $\mathcal{Q}^\omega$ are not valid :

$\widehat{q}$ is valid if

$$\begin{cases} \widehat{q}(0) = q_0 \\ \forall i \geq 0 \\ (\widehat{q}(i), \widehat{q}(i+1)) \ \in \ \mathcal{R} \end{cases}$$

In a certain sens, $\mathcal{R}$ represent **constraints** on the state space.

All this is well know. But very useful and has been used from a long time when the system are finite state (as many are) for effective constructions and decision procedures for automatically manipulating and analyzing system behavior.
But ....

Although the decision to abstract away from quantitative time has had many advantages, it is ultimately counterproductive when reasoning about systems that must interact with physical processes; the correct functioning of the control system of airplanes and toasters depends crucially upon *real-time* considerations. We would like to be able to specify and verify models of real-time systems as easily as qualitative models. Our goal is to modify finite automata for this task and develop a theory of *timed* automata, similar in spirit to the theory of ω-regular languages. We belive that this should be the first stepin building theories for the real-time verification problem.

Rajeev Alur , David L.Dill
A theory of timed automata

How to model time ?

The best way is to be rich enough in order to buy some clocks......
And of course it's the solution !
We add to our previous sytem a family (finite) of clocks.
A **clock** is a mapping :

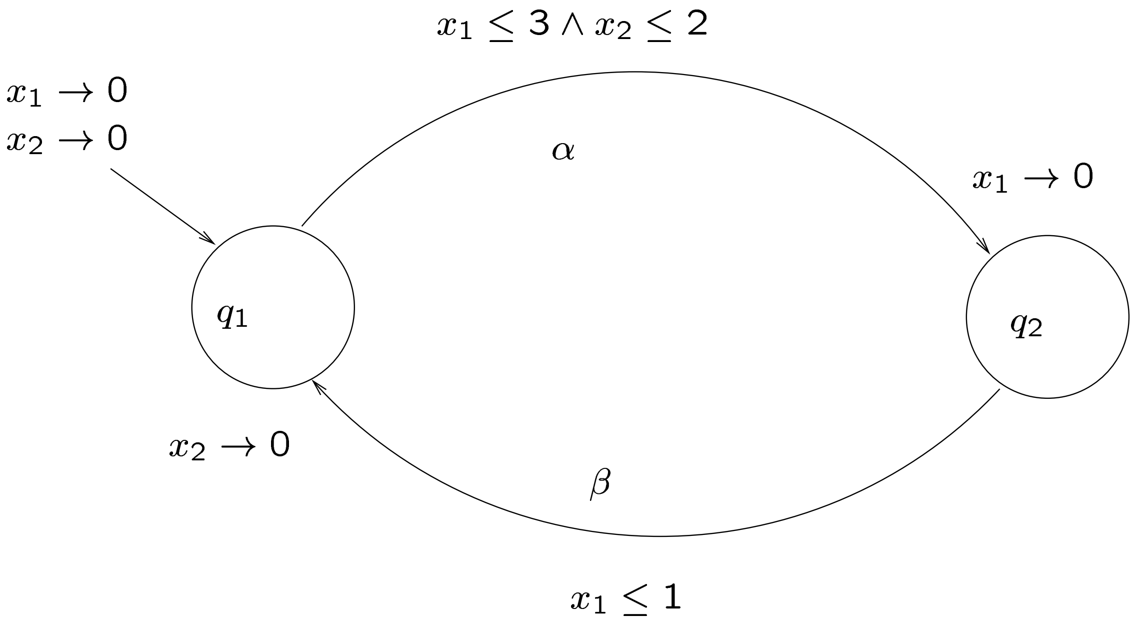
$$x : \mathbb{R} \longrightarrow \mathbb{R}$$

on such a clock we suppose that we are able to put...some **constraints**

Définition 1 *(Alur, Diel) For a set X of clock variables, the set $\Phi(X)$ of clock constraints δ is defined inductively by*

$$\delta := x \ \leq c \mid c \leq x \mid \delta_1 \wedge \delta_2, \ \mid \neg\delta$$

Let's start with an exemple :



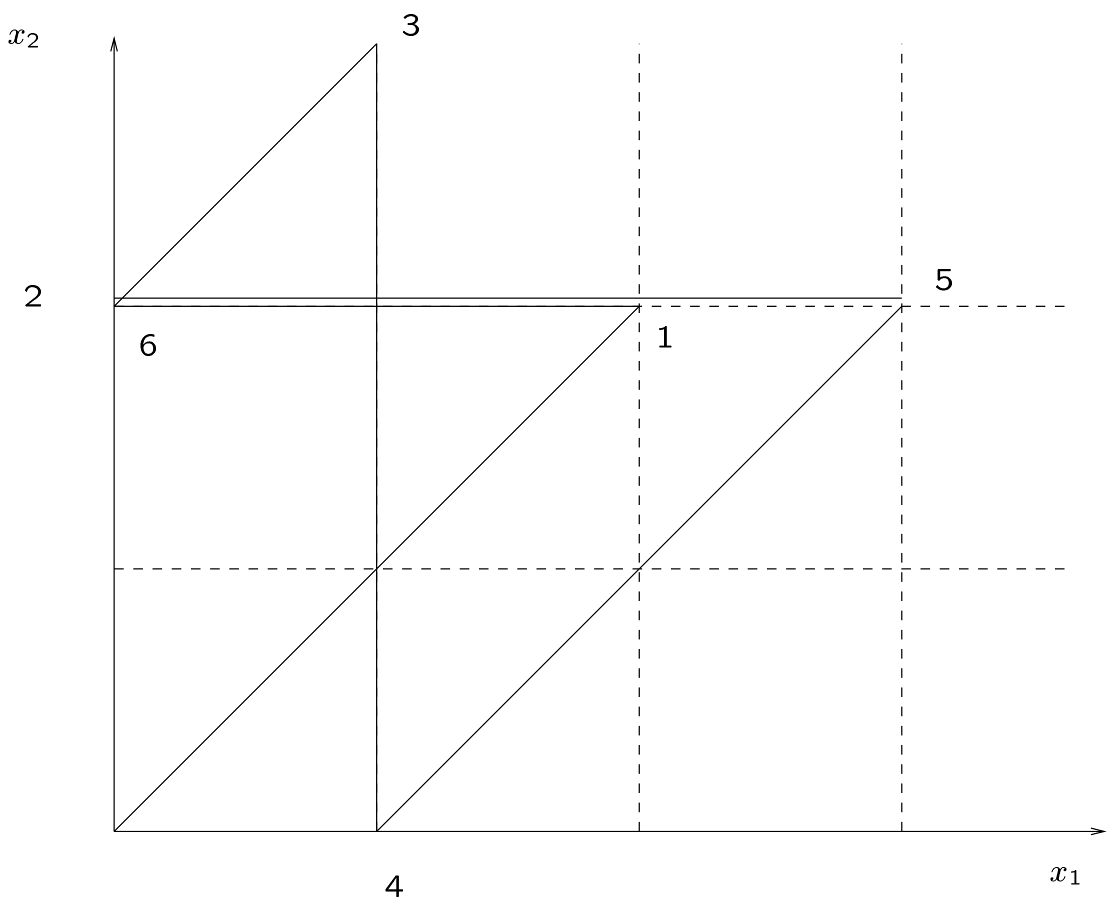
The edges of the graphic are labelled with the constraints :

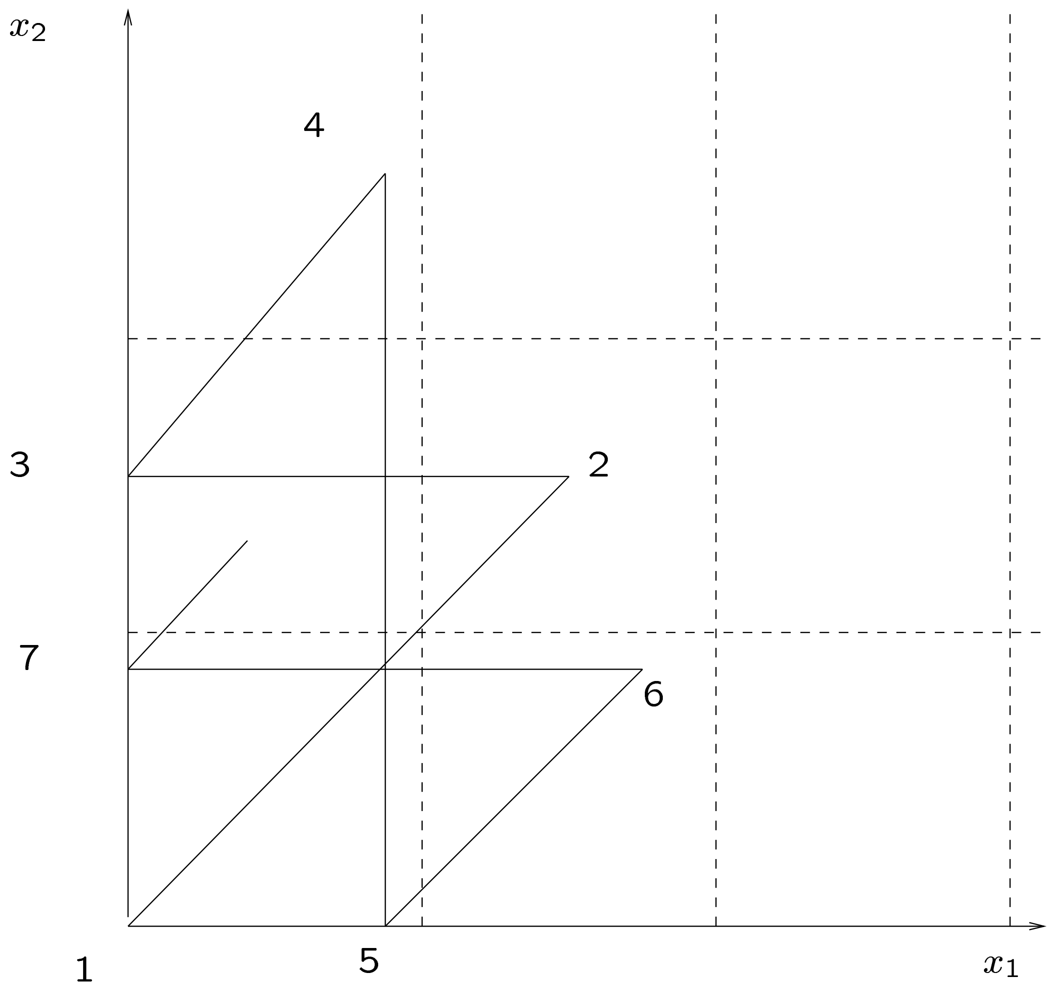
- **State constraints** as before.

- **Time constraints**

We see that when the system starts he has a first thing to do : he start the two clocks. Then in order to jump to state q_2 he must wait until the event α appends. Then he can jump (but he is not forced to do so) if he decides to jump he has to verify that the constraints on the clocks are valid.
If so, he can jump and lands on the q_2 state. Then he has to restart clock x_1 to zero and he does the same job in the q_2 state.

There is a simple way to represent such a behavior :




x_2
4
3
2
7
6
1
5
x_1

Extension of the previous theory

The problem is to describe a run of a timed system.

Définition 2 *A* **time sequence**

$$\tau = \tau_1 \tau_2 \ldots$$

is an infinite sequence of time values ($\leftrightarrow \tau_i \in \mathbb{R}$) with $\tau_i > 0$, satisfying the following constraints :

1. *Monotonicity : $\tau_i < \tau_{i+1}$ $\forall i$.*

2. *Progress : $\forall t \in \mathbb{R}$, there is some $i \geq 1$ such that $\tau_i > t$.*

Définition 3 Alur-Dill *A* **timed word** *over an alphabet* Σ *is a pair*

$$(\sigma, \tau)$$

where $\sigma \in \Sigma^\omega$ *and* τ *is a timed sequence.*
A **timed language** *over* Σ *is a set of timed words over* Σ

Timed automaton

Définition 4 *A timed automaton $\mathcal{A}$ is describe by*

- Σ *a finite alphabet.*
- $\mathcal{Q}$ *a finite set of states.*
- $q_0 \in \mathcal{Q}$.
- $\mathcal{C}$ *a finite set of clocks.*
- $\mathcal{E}$ *a relation on*

$$\mathcal{Q} \times \mathcal{Q} \times \Sigma \times 2^{\mathcal{C}} \times \Phi(\mathcal{C})$$

How does $\mathcal{A}$ work ?

Let (σ, τ) be a timed word.

- At time 0 : $\mathcal{A}$ initializes all it's clocks at 0. As time flows, the value of all clocks change. At time τ_1 $\mathcal{A}$ is confronted to the event σ_1 and the time constraints δ.

- If he can accept these two constraints then he jump to another state q'. In that case $(q, q', \sigma_1, \lambda, \delta)$ is a legal transition belonging to the set of constraints.

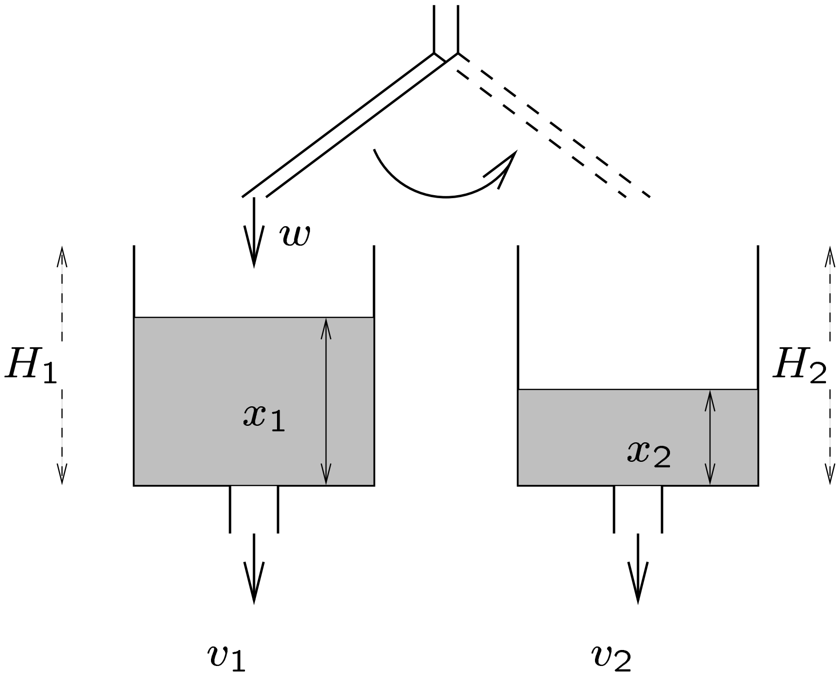
- If not he can't accept the timed word.

The two tanks system

The two tanks system

The system is composed of two different tanks (of section unity) filled of water. Each has liquid draining from its bottom with a fixed flow and there is an overhead spoud that can be positioned above any one tank at a given time and pours liquid into the tank at some constant rate.

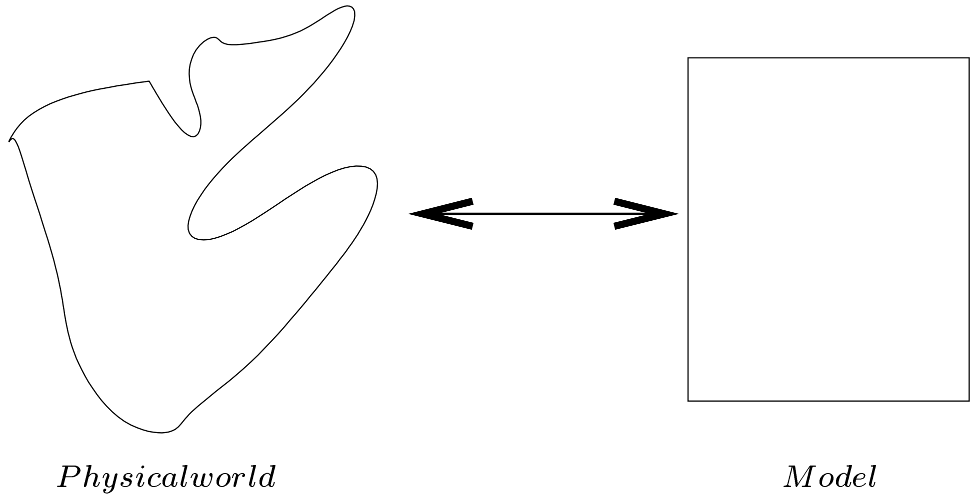
The **goal** is to maintain the level of each tank above 0 for exemple.



w
H_1
x_1
H_2
x_2
v_1
v_2

Hybrid modelling of the system

First we must be very careful about the reality of things ! We will build a **model** of a physical situation and we only will prove things (eventually . . .) about the model.



The system presents two **states** (or **modes**) :

- The (q_1) mode is when the pipe is filling the first tank (and of course the second one is emptying).

- The second (q_2) is when the pipe is on the second tank (and the first one.....).

There are two legal transitions :

- The first one

  $$q_1 \longrightarrow q_2$$

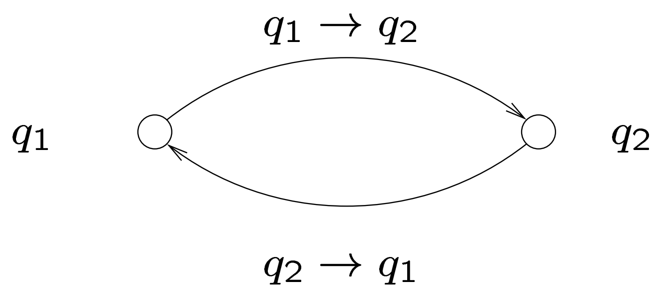
  describing the jump from q_1 state to q_2 state.

- The second transition

  $$q_2 \longrightarrow q_1$$

The set of legal transitions will be :

$$\mathcal{E} = \{(q_1, q_2), (q_2, q_1)\}.$$



At this level the system is a **discrete system** :that is to say a finite automata

$$\mathcal{M} = (\mathcal{Q}, \mathcal{E}, \Delta)$$

where

- $\mathcal{Q} = \{q_1, q_2\}$
- $\mathcal{E} = (e_1, e_2)$
- δ is a transition relation $\delta \ \subset \mathcal{Q} \times \mathcal{E} \times \mathcal{Q}$

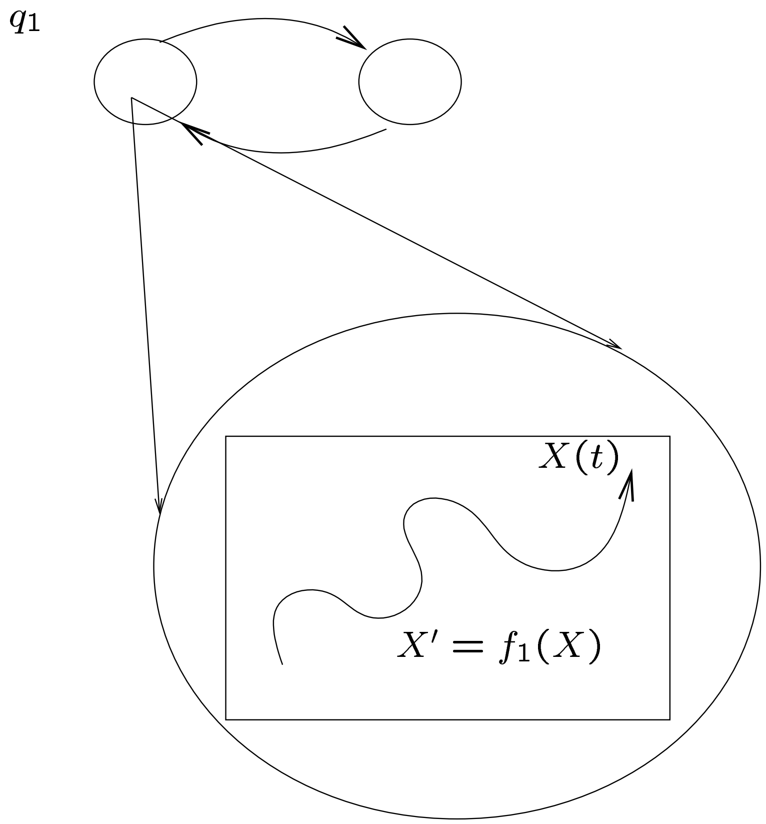
We can also specify an initial state.

Let

$$\mathcal{E}^*$$

be the set of strings of finite length of elements of $\mathcal{E}$. The string of length 0 will be defined by ϵ.

Now we want to see what happened inside a state :



To describe the dynamical part of the state we must introduce a certain number of variables :

- x_i $i = 1, 2$ the height of water in tank i or equivalently the volum of water in tank i.

- $v_i > 0,\ i = 1, 2$ the constant flow of water out of tank i.

- We will suppose that $x_1(0) > 0$ and $x_2(0) > 0$.

- w is the ingoing flow through the pipe.

The state space of the system is $\mathbb{R}^2$ and the representative point of the state of the system is $X = (x_1, x_2)\ \in \mathbb{R}^2$.

At time t, if the system is in the q_1-state, then the dynamic is described by the following set of ODE :

$$\left\{ \begin{array}{ll} x_1' = & w - v_1 \\ x_2' = & -v_2 \end{array} \right.$$

If we are in the q_2 state

$$\left\{ \begin{array}{ll} x_1' = & -v_1 \\ x_2' = & w - v_2 \end{array} \right.$$

We will denote by

$$X' = f_{q_i}(X), \;\; i = 1, 2$$

these two vector fields.

Invariants of the system

In fact (for the moment) an invarant is a subset of the state space where it's legal to use one of the two vector fields.
$I(q_1)$ is the invariant subset of $\mathbb{R}^2$ associated to the first v.f :

$$I(q_1) = \{X = (x_1, x_2), x_2 \leq H_2,\ x_2 \geq 0\}$$

and

$$I(q_2) = \{X = (x_1, x_2), x_1 \leq H_1,\ x_1 \geq 0\}$$

The guards of the system

It's the places in the state space where we must jump to another mode :

$$G(q_1, q_2) = \{X = (x_1, x_2), x_2 \leq 0 \text{or} x_1 \geq H_1\}$$

and symmetrically

$$G(q_2, q_1) = \{X = (x_1, x_2), x_1 \leq 0 or H_2 \leq x_2\}$$

For exemple, if, at time t, the system intersect $G(q_1, q_2)$ he is in state q_1 and he must jump to state q_2.
Now a *strong hypothesis of the hybrid system model* is that such a jump take place in **null** time.
We will see in the future the consequences of such an hypothesis.

The reset functions

When we meet a guard the problem is to know where we must jump. In the tanks problem the answer is simple : we jump where we are. So we introduce the following notation :

$$Res_{q_1 \longrightarrow q_2}(X) = \{X\}.$$

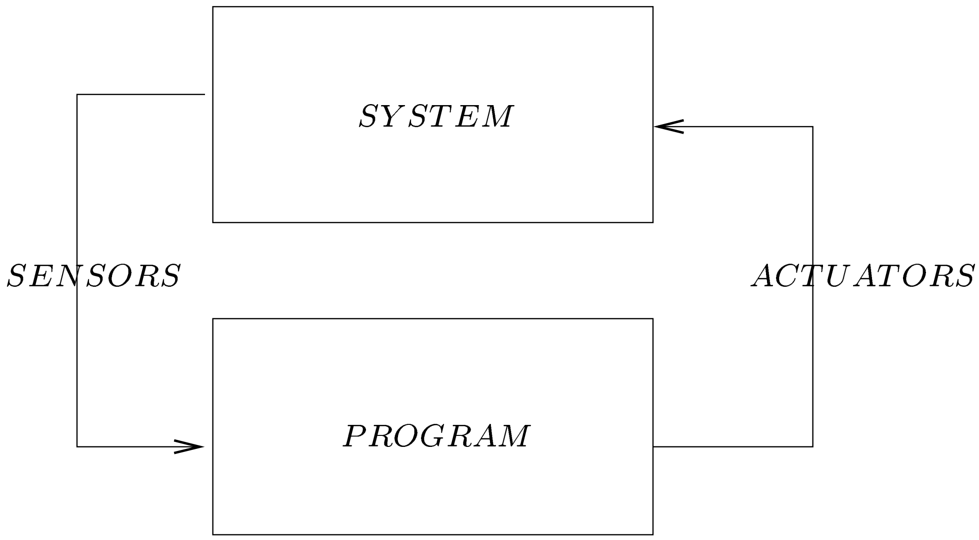
and in the same way

$$Res_{q_2 \longrightarrow q_1}(X) = \{X\}.$$

Now we we have all the ingredients of the "physical part" of the system

What about the program ?

It's a key point :
The program is in charge of the **objective** of the system.

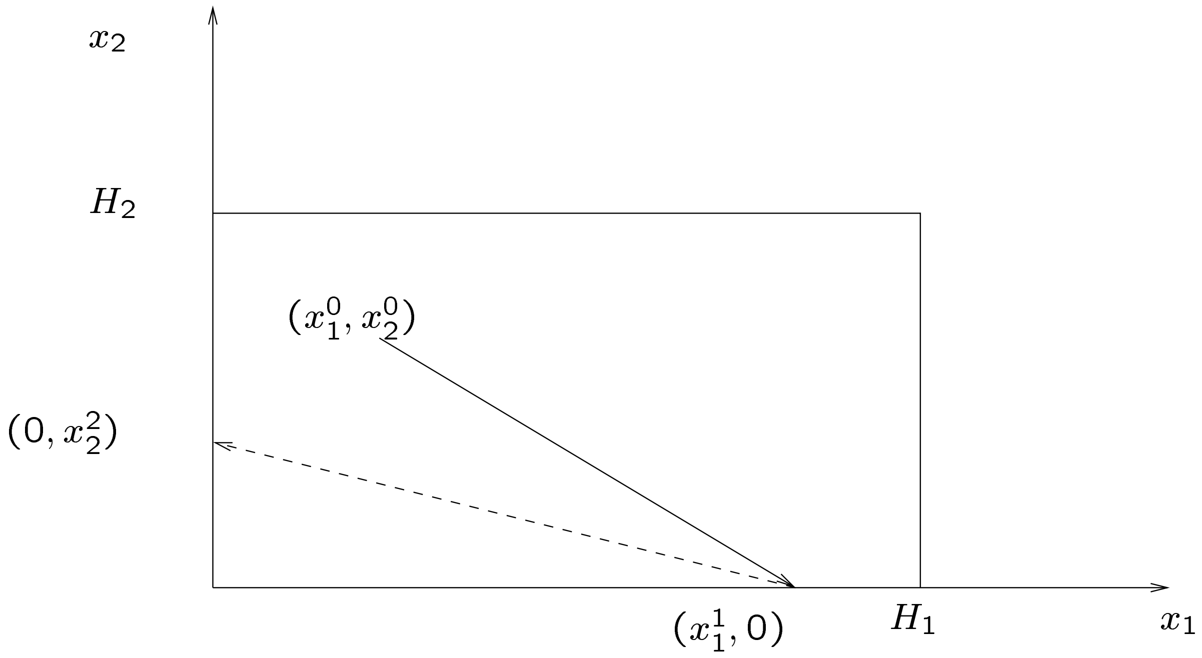


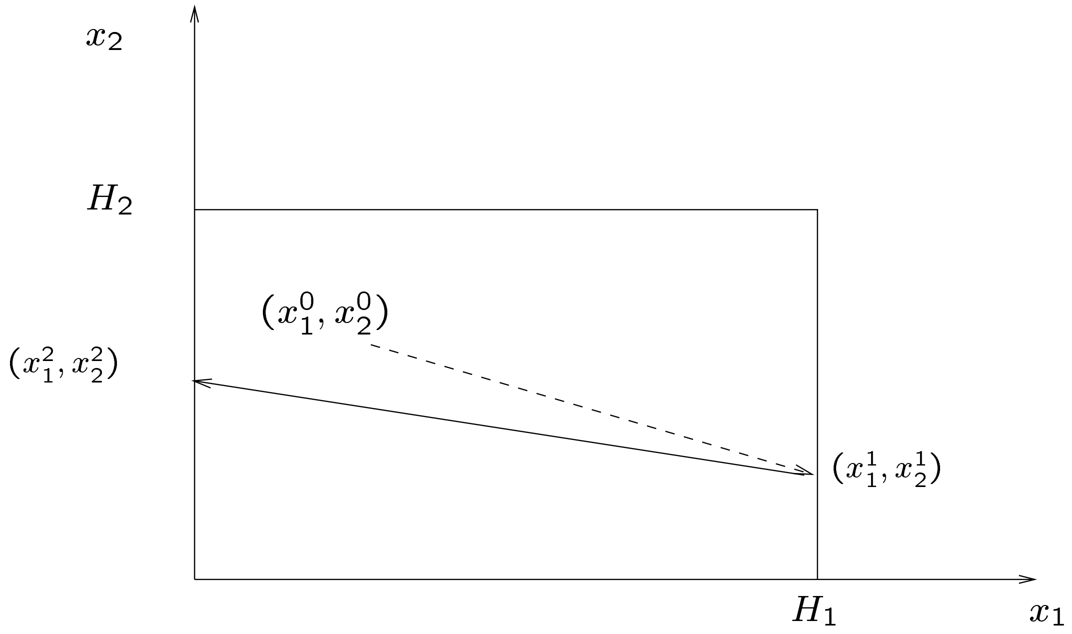
A kind of program could be :

- If $X(t) \ \in G(q_1 \longrightarrow q_2)$ then
  - if $X(t) \ \in G(q_2 \longrightarrow q_1)$ then STOP
  - else
    * $Res_{q_1 \longrightarrow q_2}(X(t))$
    * INTEGRATE f_{q_1}
- else
  - if $X(t) \ \in G(q_2 \longrightarrow q_1)$ then $Res_{q_2 \longrightarrow q_1}(X(t))$ else INTEGRATE f_{q_1}.
  - else INTEGRATE

The program is the implementation of a **strategy**.
So the next question is to analyze the behaviour of the system.
Probably one of the main question is the **reachability** question. If our goal is to keep the system working (without crash) it's important to know what is the set of states that the system can reach during it's evolution.
(In our context we have only deterministic evolution so things are simple).

To do such analysis, one can visualized the situation using a copie of the state space and two situations are possible :




x_2
H_2
(x_1^0, x_2^0)
(x_1^2, x_2^2)
(x_1^1, x_2^1)
H_1
x_1

Analysis of the behavior of the system

We see that

$$x_1' + x_2' = w - v_1 - v_2 = a$$

where a is some constant. So

$$x_1 + x_2 = at + b$$

b is an integration constant.
We have supposed that at $t = 0$ the system was in the q_1 state (and it's always possible to suppose that $x_1(0) = 0$ and $x_2(0) = h$ where $0 < h < H_2$) so

$$b = h$$

.

We now have three possibilities :

- $a < 0$ after a lapse of time $t^* = -\frac{h}{a}$ the two tanks will be empty.

- $a = 0$ The total amount of liquid in the two tanks will be constant and we have a periodic solution.

- $a > 0$ then the two tanks will overflow after a period $t^* = \frac{H_1+H_2-h}{a}$.

But this is the easy part of the analysis. Now we need to be more careful about what is really happening.

Exectution

The notion of execution is basic in HS : here we only want to give an intuitive idea of it.

- The system starts at time $t_0 = 0$ in the q_1 mode and the associated point in the state space is

$$x_1^0 = 0 \text{ and } x_2^0 = h$$

(here we suppose that $0 < h < H_2$)
The dynamic of the system is describe by

$$\left\{ \begin{array}{lcl} x_1(t) & = & (w - v_1)t \\ \\ x_2(t) & = & -v_2 t + h \end{array} \right.$$

- after a certain time the system reach a guard. I must jump to the q_2 mode.

  We have two possibilities

– Either $x_1 = H_1 \Leftrightarrow t_0^+ = \frac{H_1}{w-v-1}$

– Or $x_2 = 0 \Leftrightarrow t_0^+ = \frac{h}{v_2}$.

- Now starting from time $t_1^- = t_0^+$ at the point $X(t_1^-)$ equal to $(H_1, h - v_2\frac{H_1}{w-v-1})$ or $((w - v_1)\frac{h}{v_2}, 0)$, we will performed the integration of the f_2 vector field until we reach a novel guard.

The analysis of the complete process run in the following manner (for simplicity we will only study a particular case, theother cases are left as a wonderful exercice for a rainy WE . . .) : Suppose that we switch from q_1 to q_2 in time $t_0^+ = \frac{h}{v_2}$, we have

$$x_1(t_0^+) = (w - v_1)\frac{h}{v_2},\ x_2(t_0^+) = 0.$$

Once again, we have two possibilities to switch into the state q_1, either when $x_2 = H_2$, or when $x_1 = 0$. We choose the second possibility and we calculate the time of switching :

$$x_1 = 0 \Rightarrow t_1^+ = \frac{wh}{v_1 v_2},$$

or

$$t_1^+ = t_0^+ + \frac{w - v_1}{v_1 v_2}h = t_0^+ + \frac{w - v_1}{v_1}t_0^+$$

and the new initial condition for the system of the q_1 is

$$x_1(t_1^+) = 0, x_2(t_1^+) = \frac{(w - v_1)(w - v_2)}{v_1 v_2}h.$$

After a **certain** number of iterations we see that the system jumps from one state to another state with a duration whose limit is zero. And we can calculate

$$\tau_\infty = (t_0^+ - t_0) + \sum_{i=0}^{i=\infty} (t_{i+1}^+ - t_i^+).$$

It's possible to compute the sum, and, of course the value of the serie is

$$\tau_\infty = \frac{-h}{a}.$$

(don't panic $a < 0$)
One can do a similar computation for the other case and it's possible to find that

$$\tau_\infty = \frac{H_1 + H_2 - h}{a}.$$

The **moral** of the story is that :
We cut time during the transition and we introduce in the modelling process the "artifac" called here **ZENO** behavior. An infinite number of discrete transitions in **finite** time.