SAPIENZA
UNIVERSITÀ DI ROMA

DIAG

Hybrid Control Homework

INTELLIGENCE AND HYBRID CONTROL

Professor:
Luca Benvenuti

Student:
Ludovica Cartolano

Academic Year 2022/2023

Contents

1 Introduction

In this work I will formulate, discuss and solve six different Homework on hybrid systems.
An hybrid system is a mathematical model that involve the interaction of different types of dynamics in particular it describes a phenomenon in which event-driven discrete-state dynamics and time-driven continuous state dynamics coexist and interact. By event-driven systems one means systems that instantaneously change state in response to certain asynchronous discrete events [1].

Event-driven finite-state systems can be described by Finite State Machines (FSMs) or Automata. A Finite automation M is represented as

$$M = (Q, \Sigma, \delta, \Omega) \tag{1}$$

where $Q = q_1, ..., q_n$ is the finite set of discrete states, $\Sigma = e_1, ...e_m$ is the finite set of input events, $\delta : Q \times \Sigma \to 2^Q$ is the transition function, $\delta(q, e) \subseteq Q$ is the set of all states the system can transit from general state q_i to the general event e_i and $\Omega = y_1, ..., y_p$ is the finite set of output symbols.
A discrete state is said to be safe when it has requirements that should be continuously true so that some "bad things" will never happen.

A hybrid system H is a collection

$$H(Q, X, Init, f, Inv, E, G, R) \tag{2}$$

where, as it is defined in [1],

- Q is a finite collection of discrete variables $\{q_1, q_2, ..\}$,
- $X = \mathbb{R}^n$, $\{x_1, ..., x_n\}$ is a finite collection of continuous states,
- $Init \subseteq Q \times X$ is a set of initial states,
- $f(\cdot, \cdot) : Q \times X \to \mathbb{R}^n$ is a vector field assumed to be Lipschitz,
- $Inv(\cdot) : Q \to 2^X$ assigns to each $q \in Q$ an invariant set,
- $E \subseteq Q \times Q$ is a collection of edges,
- $G(\cdot) : E \to 2^X$ assigns to each edge $e = (q, q') \in E$ a guard condition,
- $R(\cdot, \cdot) : E \times X \to 2^X$ assigns to each edge $e = (q, q') \in E$ and $x \in X$ a reset relation.

An hybrid execution of H (2) is an hybrid trajectory $\aleph = (\tau, q, x)$ such that

- Initial condition $(q_0(\tau_0), x_0(\tau_0)) \in Init$,

- Event-driven evolution: $\forall i,\ (q_i(\tau_i'), q_{i+1}(\tau_{i+1})) \in E,\ x_i(\tau_i') \in G(q_i(\tau_i'), q_{i+1}(\tau_{i+1}))$ and $x_{i+1}(\tau_{i+1}) \in R(q_i(\tau_i'), q_{i+1}(\tau_{i+1}), x_i(\tau_i'))$.

- Continuous evolution: $\forall i$ with $\tau_i < \tau_i',\ q_i(t) = q_i(\tau_i) \quad \forall t \in I_i = [\ \tau_i, \tau_i']$ and $x_i(\cdot)$ is the solution to the differential equation $f(q_i(t), x_i(t))$ and $\forall t \in [\ \tau_i, \tau_i'),\ x_i(t) \in Inv(q_i(t))$ [1].

with, $\tau = \{I_0, ..., I_n\}$ is the hybrid time set, $q_i(\cdot) : I_i \to Q$ and $x_i(\cdot) : I_i \to X$ are sequences of functions.

An execution is called *Zenonian* if it is infinite but the sum of intervals is finite $\sum_i(\tau_i' - \tau_i) < \infty$. In other words executions that show an infinite number of discrete transitions in a finite amount of time are referred to Zeno executions.
$\mathcal{H}^{\infty}_{(q_0,x_0)}$ is the set of all executions of H with initial condition $(q_0, x_0) \in Init$ and infinite executions.

An hybrid system H as defined in Eq.2 is called *Zeno* if $\exists(q_0, x_o) \in Init$ such that all infinite executions $\mathcal{H}^{\infty}_{(q_0,x_0)}$ are Zeno [2].

Typically the Zeno behaviour arises due to modeling abstractions employed to obtain models that are simpler to analyse and control. A way to solve this, is by reintroducing some of the physical considerations though the process of regularization which is a standard technique for dealing with differential equations whose solutions are not well defined.

In the next chapters I will use these concepts as a basis to solve the Homework assigned.

2 Homework 1 - Dual Game Theory Approach

Prove formally that the following algorithms computing the maximal safe set give the same result.

$$
\begin{array}{l}
W^0 := \overline{Unsafe} \\
i := -1 \\
repeat\{ \\
\qquad i := i + 1 \\
\qquad W^{i+1} := W^i \backslash UPre(W^i) \\
\}\, until\ (W^{i+1} = W^i) \\
Safe := W^i
\end{array}
\qquad\qquad
\begin{array}{l}
W^0 := \overline{Unsafe} \\
i := -1 \\
repeat\{ \\
\qquad i := i + 1 \\
\qquad W^{i+1} := W^i \cap CPre(W^i) \\
\}\, until\ (W^{i+1} = W^i) \\
Safe := W^i
\end{array}
$$

Figure 1: Algorithms to be proven

2.1 Discussion

Since the two algorithms in Fig.1 differ from each other only in $W \backslash UPre(W)$ and $W \cap CPre(W)$, it is needed to prove that those are the same set. Consider the Finite State Machine M (1) described in the Introduction (1).
Game theory is related to optimal control, with the difference that there are two player (the control variables are divided into two classes) with possibly conflicting objectives [1]. The set Σ of input symbols (events) will be divided in:

- $\Sigma_c = u_1, ..., u_m$ controllable input events with $u \in \Sigma_c \cup \epsilon$,
- $\Sigma_d = d_1, ..., d_s$ uncontrollable input events (environment) with $d \in \Sigma_d \cup \epsilon$

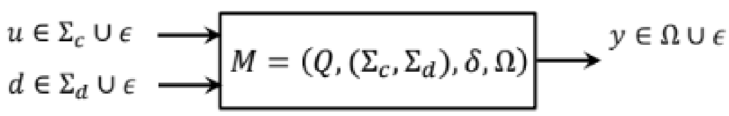

Figure 2: The FSM

The first algorithm considered in Fig. 1 is the **Maximal safe set computation** algorithm.
The controller tries to prevent the state from entering the unsafe set while the environment tries to make the state entering the unsafe set. $UPre(W)$ is the set such that there is an environment input that forced the state to go outside W in one step, whatever the controller input is

$$UPre(W) = \{q \in Q : \forall u \in \Sigma_c, \exists d \in \Sigma_d | \delta(q, (u, d)) \nsubseteq W\}. \tag{3}$$

At each step, the temporary safe set is reduced by subtracting to it the states that may be forced to go outside it by an environment input, whatever the controller input is.

To proceed it is necessary to define the set $CPre(W)$ which is the set of states such that there is a controller input that forces the state to go inside W in one step, whatever the environment's input is

$$CPre(W) = \{q \in Q : \forall u \in \Sigma_c, \exists d \in \Sigma_d | \delta(q,(u,d)) \subseteq W\} \tag{4}$$

where the environment tries to prevent the state from entering the goal set and the control tries to make the state entering the goal set.

From the definitions it is possible to empirically say that $UPre(W)$ and $CPre(W)$ are in fact complementary and $W \backslash UPre(W)$ and $W \cap CPre(W)$ are the the same set.

Using set theory [3] follows the formal proof.
By definition of complementary set [3], the set $\overline{UPre(W)}$ is the complementary set of $UPre(W)$ such that
$\overline{UPre(W)} = CPre(W) = \{q \in Q : \forall u \in \Sigma_c, \exists d \in \Sigma_d | \delta(q,(u,d)) \subseteq W\}$.

It is intuitive that each discrete state $q \in Q$ has to belong to $CPre(W)$ or to $UPre(W)$ on the bases of the set definitions (3) and (4). This is because a discrete state $q \in Q$ cannot belong to both sets since if $q \in CPre$ then $\forall u \in \Sigma_c, \exists d \in \Sigma_d | \delta(q,(u,d)) \subseteq W$ and if $q \in UPre$ then $\forall u \in \Sigma_c, \exists d \in \Sigma_d | \delta(q,(u,d)) \nsubseteq W$ and this is logically not possible. This means that

$$UPre(W) \cup CPre(W) = W. \tag{5}$$

Consider the sets A and B. Defining $A \cap B$ as the intersection between sets which is the set of all the elements that belong to both sets A and B and $A \backslash B$ as the difference between sets which is the set of all the elements of A that do not belong to B, follows quite naturally that

$$A \backslash B = A \cap \overline{B}. \tag{6}$$

From Eq.5 and Eq.6 follows that

$$W \cap CPre(W) = W \cap \overline{UPre(W)} = W \backslash UPre(W).$$

QED (Quod Erat Demonstrandum).

3 Homework 2

Define a hybrid model for the following phenomenon: Frictionless movement of a particle in a bounded interval subject to elastic collisions at the end points of the interval itself.
Provide the automaton H and define its elements Q, X, Init, f, Inv, E, G, R.

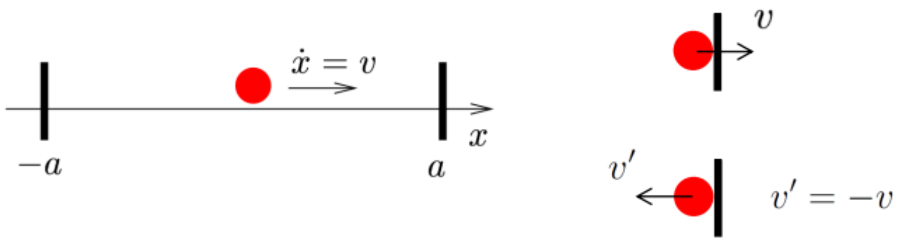

Figure 3: The Problem

3.1 Discussion and Formulation

Consider the problem showed in Fig.3, denote with

- $x_1(t)$ the position of the particle such that $x_1(t) \in [\,-a, a]$
- $x_2(t)$ the velocity of the particle such that $x_2 \in [\,-v, v]$

I define two scenarios in which the particle might find itself as time goes on:

- the particle travels between the boundaries such that $(x_1 > -a) \vee [\,(x_1 = -a),\ (x_2 > 0)]$ and $(x_1 < a) \vee [\,(x_1 = a),\ (x_2 < 0)]$, to which corresponds the discrete state q_1 that follows the dynamics $\dot{x}_1 = x_2,\ \dot{x}_2 = 0$;
- the particle elastically collides to the boundaries such that $[\ (x_1 = -a), (x_2 < 0)]\ \vee [\ (x_1 = a), (x_2 > 0)]$, to which corresponds the event driven dynamics $x_1' := x_1 = 0,\ x_2' := -x_2$.

The Particle Elastic Collision phenomenon is described by the following hybrid automation.

$$H = (Q, X, Init, f, Inv, E, G, R) \tag{7}$$

with

$$Q = \{q_1\} \quad X = \{x_1 \in [\,-a, a]\,\} \times \{x_2 \in [\,-v, v]\,\}$$
$$Init = q_1 \times \{x \in \mathrm{X} : (-a < x_1 < a), (-v \leq x_2 \leq v)\}$$

$$f(q_1, x_1, x_2) : \begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} x_2 \\ 0 \end{pmatrix}$$

$$Inv(q_1) = \{x \in \mathrm{X} : (x_1 > -a)\} \cup \{x \in \mathrm{X} : [\ (x_1 = -a) \wedge (x_2 \geq 0)]\ \} \cup$$
$$\{x \in \mathrm{X} : (x_1 < a)\} \cup \{x \in \mathrm{X} : [\ (x_1 = a) \wedge (x_2 \leq 0)]\ \}$$

$$E = \{e_1 = (q_1, q_1)\}$$

$$G(e_1) = \{x \in \mathrm{X} : (x_1 = -a,\ x_2 < 0) \vee (x_1 = a,\ x_2 > 0)\}$$

$$R(e_1, x_1, x_2) = \begin{pmatrix} x_1 \\ -x_2 \end{pmatrix}$$

The directed graph corresponding to this hybrid automaton is shown in Fig.4

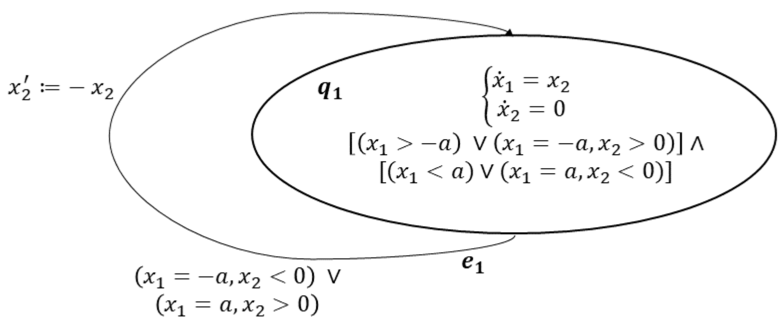

Figure 4: The Automata that solves Homework 2

4 Homework 3

Define a hybrid model for the following phenomenon: Frictionless movement of two particles in a bounded interval subject to elastic collisions between them and at the end points of the interval itself
Provide the automaton H and define its elements Q, X, Init, f, Inv, E, G, R

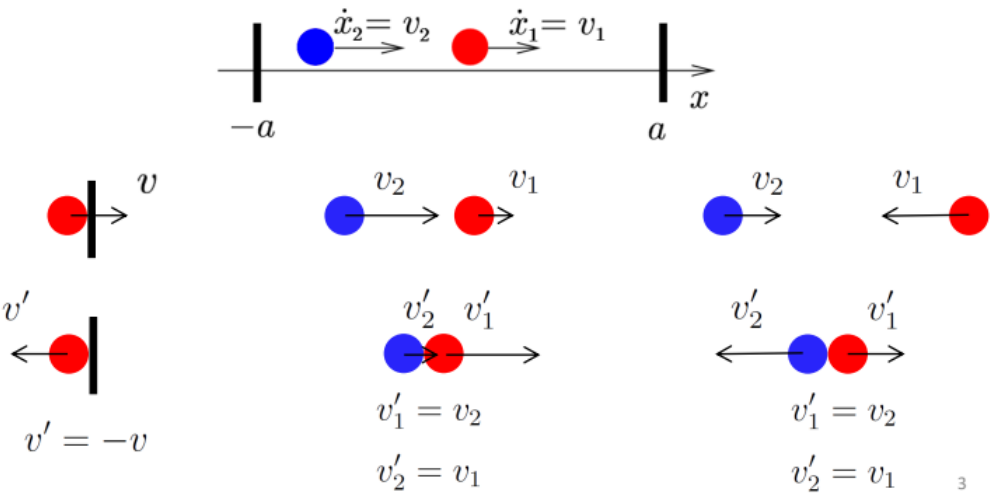

Figure 5: The Problem

4.1 Problem definition

Consider the problem showed in Fig.5, denote with

- $x_1^B(t)$ the position of the **blue** particle $x_1^B \in [\,-a, a]$.
- $x_1^R(t)$ the position of the **red** particle such that $x_1^R \in [\,-a, a]$
- $x_2^B(t)$ the velocity of the **blue** particle such that $x_2^B \in [\,-v_2, v_2]$
- $x_2^R(t)$ the velocity of the **red** particle such that $x_2^R \in [\,-v_1, v_1]$

4.2 Discussion and Formulation

I define three scenarios in which each particle might find itself as time goes on:

1. The particles move between the boundaries without colliding with one another, such that
   $(x_1^B > -a) \vee [\,(x_1^B = -a),\ (x_2^B > 0)]$ and $(x_1^R < a) \vee [\,(x_1^R = a),\ (x_2^R < 0)]$; to which corresponds the discrete state q_1 and the dynamics
   $\dot{x}_1^B = x_2^B,\ \dot{x}_1^R = x_2^R,\ \dot{x}_2^B = 0,\ \dot{x}_2^R = 0;$

2. The particles elastically collide to the boundaries. In particular, the blue particle is able to collide only with the left bound and the red particle is able to collide only with the right bound such that
   $[\ (x_1^B = -a), (x_2^B < 0)]$ and $[\ (x_1^R = a), (x_2^R > 0)]$, to which corresponds the event driven dynamics $x_1^{B\prime} := x_1^B = 0,\ x_1^{R\prime} = x_1^R = 0,\ x_2^{B\prime} := -x_2^B,\ x_2^{R\prime} = -x_2^R$;

3. The particles collide with one another in the two different ways showed in Fig.5 is such that
   a) $v_2 > v_1$, the two particles will go to the same direction
   $(x_1^B - x_1^R > 0) \vee [\ (x_1^B - x_1^R = 0), (x_2^B > x_2^R]$ to which corresponds the discrete state q_1 and the same dynamics in item 1.;
   b) $v_2 < v_1$, the two particles elastically collide and take opposite directions
   $[\ (x_1^B - x_1^R = 0),\ (x_2^B < x_2^R)]$, to which corresponds the event driven dynamics $x_2^{B\prime} := x_2^R,\ x_2^{R\prime} := x_2^B$.

The Particles Elastic Collision phenomenon is described by the following hybrid automation.

$$H = (Q, X, Init, f, Inv, E, G, R) \tag{8}$$

with

$$Q = \{q_1\} \quad X = \{x_1^R \in [\ -a, a]\ \} \times \{x_1^B \in [\ -a, a]\ \} \times \{x_2^R \in [\ -v_1, v_1]\ \} \times \{x_2^B \in [\ -v_2, v_2]\ \}$$

$$Init = q_1 \times \{x \in \mathrm{X} : (-a < x_1^R < a), (-a < x_1^B < a), (-v_1 \le x_2^R \le v_1), (-v_2 \le x_2^B \le v_2)\}$$

$$f(q_1, x_1^B, x_1^R, x_2^B, x_2^R) : \begin{pmatrix} \dot{x}_1^B \\ \dot{x}_1^R \\ \dot{x}_2^B \\ \dot{x}_2^R \end{pmatrix} = \begin{pmatrix} x_2^B \\ x_2^R \\ 0 \\ 0 \end{pmatrix}$$

$$\begin{aligned} Inv(q_1) = \{x \in \mathrm{X} : [\ (x_1^B > -a) \vee (x_1^B = -a, x_2^B > 0)] \wedge \\ [\ (x_1^R < a) \vee (x_1^R = a, x_2^R < 0)] \wedge \\ [\ (x_1^B > x_1^R) \vee (x_1^B = x_1^R, x_2^B > x_2^R)]\ \} \end{aligned}$$

$$E = \{e_1 = (q_1, q_1),\ e_2 = (q_1, q_1)\}$$

$$G(e_1) = \{[\ (x_1^B = -a), (x_2^B < 0)]\ \wedge [\ (x_1^R = a), (x_2^R > 0)]\ \}$$
$$G(e_2) = \{[\ (x_1^B = x_1^R),\ (x_2^B < x_2^R)]\ \}$$

$$R(e_1, x_1^B, x_1^R, x_2^B, x_2^R) = \begin{pmatrix} x_1^B \\ x_1^R \\ -x_2^B \\ -x_2^R \end{pmatrix}$$

$$R(e_2, x_1^B, x_1^R, x_2^B, x_2^R) = \begin{pmatrix} x_1^B \\ x_1^R \\ x_2^R \\ x_2^B \end{pmatrix}$$

The directed graph corresponding to this hybrid automaton is shown in Fig.6

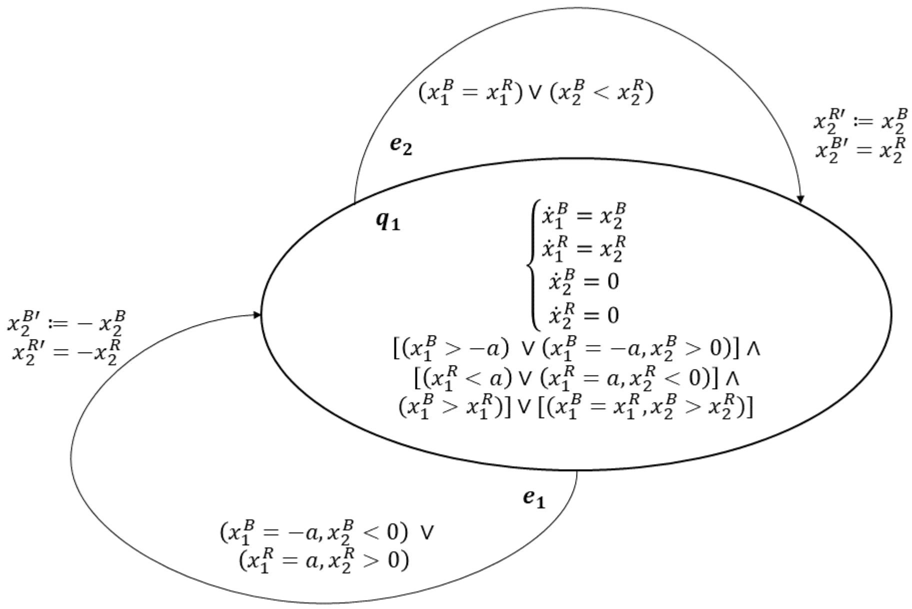

Figure 6

5 Homework 4

Define a hybrid model for the following phenomenon:

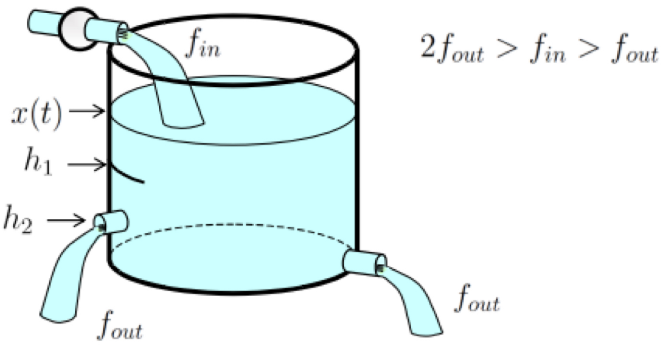

Figure 7: The Problem

Provide the automaton H and define its elements Q, X, Init, f, Inv, E, G, R

5.1 Problem definition

Fig. 7 describes a tank containing water that leaks at a constant rate from two pipes at different heights. Water is added to the system at a constant rate through another pipe equipped with a valve. Call *first pipe* the output pipe at the bottom of the tank and *second pipe* the other one.

Denote with

- $x(t)$ the level of the water tank,
- f_{in} constant input flow,
- f_{out} constant output flow from each pipe,
- h_1 threshold value for the water level,
- h_2 height of the second output pipe.

The valve opens and closes according to the following control logic:

$$\begin{cases} input\ valve\ open\ if\ x(t) < h_1 \\ input\ valve\ close\ if\ x(t) \geq h_1 \end{cases} \tag{9}$$

5.2 Discussion and Formulation

I will consider three discrete states that represent three possible behaviours of the water tank:

- q_1 the valve is closed then $x(t) \geq h_1$,
- q_2 the valve is opened and the water flows from the two output pipes then $h_2 < x(t) < h_1$,
- q_3 the valve is opened and the water flows from the first output pipe then $x(t) \leq h_2$.

When the valve is closed the water is going out from the two output pipes. After a while the water reaches h_1, the valve opens, the inflow is going to the tank and the water flows out from the two output pipes. When the water reaches h_2, with the valve still opened, the water flows out only from the first pipe.

The Water Tank phenomenon is described by the following dynamics:

$$\begin{cases} \dot{x} = -2f_{out},\ x(t) \geq h_1 \\ \dot{x} = f_{in} - 2f_{out},\ h_2 < x(t) < h_1 \\ \dot{x} = f_{in} - f_{out},\ x(t) \leq h_2 \end{cases}$$

with initial condition for the valve to be *opened.*

It is straightforward, now, to define a hybrid automaton to describe this process

$$H = (Q, X, Init, f, Inv, E, G, R) \tag{10}$$

The automaton will have:

$$Q = \{q_1,\ q_2,\ q_3\} \quad X = \mathbb{R}^+ \quad Init = q_1 \times \{x \in \mathrm{X} : x \geq h_1\}$$

$$\begin{aligned} f(q_1, x) &: \dot{x} = -2f_{out} \\ f(q_1, x) &: \dot{x} = f_{in} - 2f_{out} \\ f(q_1, x) &: \dot{x} = f_{in} - f_{out} \end{aligned}$$

$$\begin{aligned} Inv(q_1) &= \{x \in \mathrm{X} : x \geq h_1\} \\ Inv(q_2) &= \{x \in \mathrm{X} : h_2 < x < h_1\} \\ Inv(q_3) &= \{x \in \mathrm{X} : x < h_2\} \end{aligned}$$

$$E = \{e_1 = (q_1, q_2),\ e_2 = (q_2, q_3),\ e_3 = (q_3, q_2)\}$$

$$G(e_1) = \{x \in \mathrm{X} : x < h_1\} \tag{11}$$

$$G(e_2) = \{x \in \mathrm{X} : x = h_2\} \tag{12}$$

$$G(e_3) = \{x \in \mathrm{X} : x = h_2\} \tag{13}$$

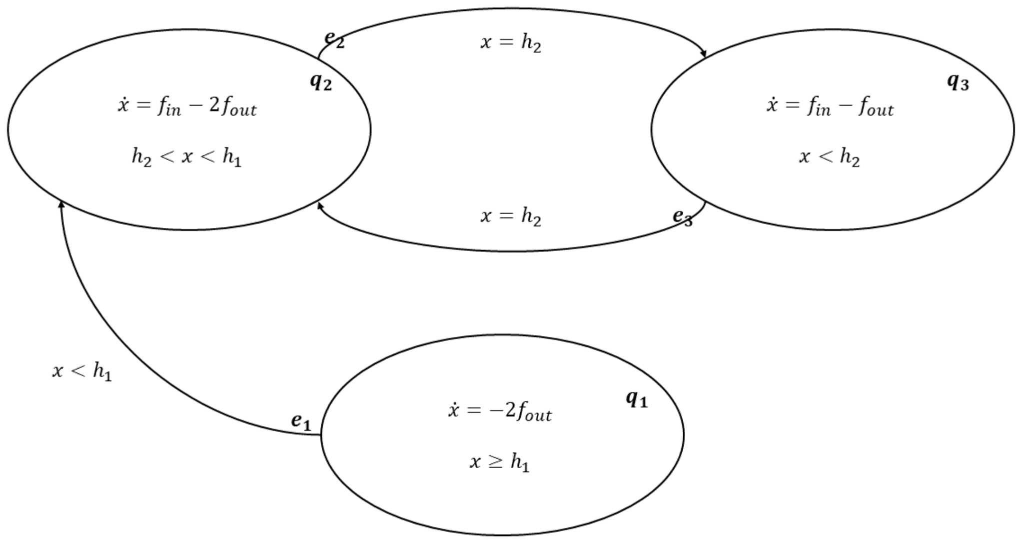

Figure 8: The Automata that solves Homework 4

$$R(e_1, x) = x$$
$$R(e_2, x) = x$$
$$R(e_3, x) = x$$

since the continuous state does not change.

The directed graph corresponding to this hybrid automaton is shown in Fig.8.

It is known that, by construction, the inflow is less than the sum of the outflows ($2f_{out} > f_{in} > f_{out}$). This implies that the volume of the water tank will vary at a high frequency for different input values. The switch between the discrete events corresponding to the guard conditions (12) and (13) will be instantaneous, creating this way an hysteretic behaviour (Zeno behaviour).

In this case the solution to eliminate the Zeno behaviour is simple. It will be considered a certain tolerance value $\delta > 0$ in the control logic (12) and (13) such that the switching moment between the dynamics is delayed of the factor δ. There will still be an oscillatory behaviour, but it will not be a Zeno one.

The control logic is modified as follows:

$$Inv(q_1) = \{(x \geq h_1)\}$$
$$Inv(q_2) = \{(h_2 - delta < x < h_1)\}$$
$$Inv(q_3) = \{(x < h_2 + \delta)\}$$

$$G(e_1) = \{x < h_1\}$$
$$G(e_2) = \{x = h_2 - \delta\}$$
$$G(e_3) = \{x = h_2 - \delta\}$$

And in Fig. 9 shows the modified automation that solves the Homework.

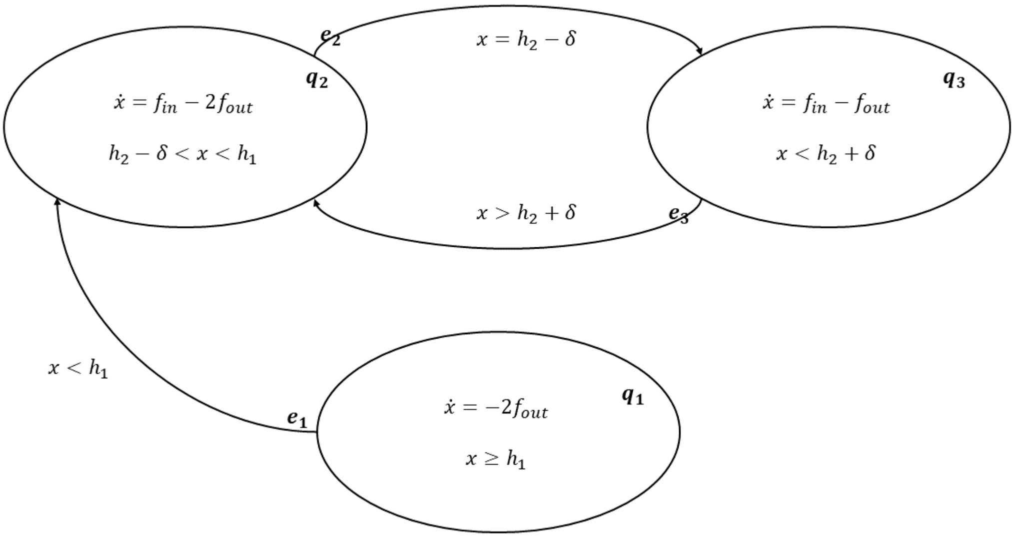

Figure 9: The Automata that solves Homework 4 that prevents Zeno Behaviour

6 Homework 5

1. *Determine if the given hybrid system is deterministic.*
2. *Determine if the given hybrid system is non blocking.*
3. *Compute the sets Out and Reach.*

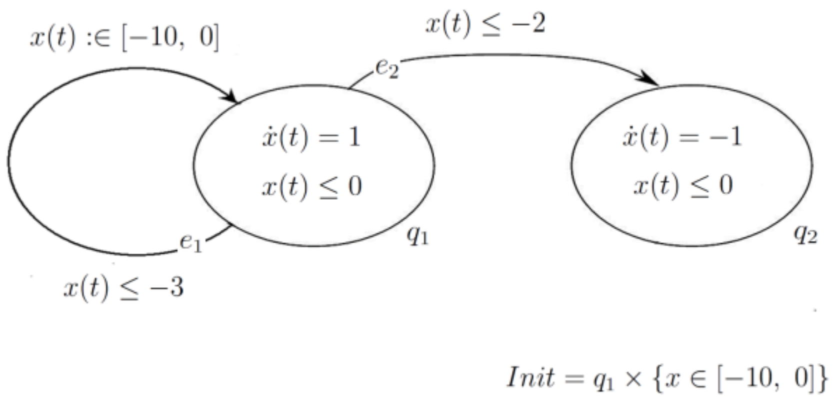

Figure 10: The Problem

6.1 Discussion

The directed graph shown in Fig.10 is described by the following hybrid automation.

$$H = (Q, X, Init, f, Inv, E, G, R) \tag{14}$$

with

$$Q = \{q_1, q_2\} \quad X = \{x \in [\ -10, 0]\ \} \quad Init = q_1 \times \{x \in \mathrm{X}\}$$

$$f(q_1, x) : \dot{x} = 1$$
$$f(q_2, x) : \dot{x} = -1$$

$$Inv(q_1) = \{x \in \mathrm{X} : x \le 0\}$$
$$Inv(q_2) = \{x \in \mathrm{X} : x \le 0\}$$

$$E = \{e_1 = (q_1, q_1)\}$$
$$E = \{e_2 = (q_1, q_2)\}$$

$$G(e_1) = \{x \in \mathrm{X} : x \le -3\}$$
$$G(e_2) = \{x \in \mathrm{X} : x \le -2\}$$

$$R(e_1, x) = x :\in [\ -10, 0]$$
$$R(e_2, x) = x$$

1. Since an hybrid system is called deterministic if, for each initial state $(q, x) \in Init$ corresponds a unique execution ([1]) where for execution one means the definition given in Introduction (1). Then, for the hybrid system H (14) in Fig.10 there are two possible executions $(e_1,\ e_2)$ corresponding to the guard conditions (6.1) and (6.1) that belong to the initial state set $Init$ $\forall x\ : -10 \leq x \leq -3$. The hybrid system H (14) is *non deterministic* because it accepts multiple executions for a single initial state.

2. An hybrid system is called non-blocking if there exists an infinite execution starting at each initial state $(q, x) \in Init$. Then for the hybrid system H (14) in Fig.10 it is easy to see that the state q_2 is only reachable from the event e_2 and there are no events that can release state x once state q_2 is reached. This implies that the executions are finite.

3. A state $(q, x) \in Q \times X$ of an hybrid system H is reachable if there exists a finite execution ending in the said state (q, x): $Init \subseteq Reach$ where $Reach$ is the set of all reachable states (q, x) ([1]). Follows that for the hybrid system H (14) in Fig.10 has $Reach = Init$.
   Instead, a Out is the set of states for which continuous evolution from the state $(q, x) \in Q \times X$ forces the system to exit the domain instantaneously.
   $Out = \{(q, x) \in Q \times X | \forall \epsilon > 0, \exists t \in [\ 0, \epsilon]\ \ such\ that\ (q, x(t)) \notin Inv(q)\}$: $\bigcup_{q \in Q} \{q\} \times \overline{Inv(q)} \subseteq Out$ ([1]). Follows that for the hybrid system H (14) in Fig.10 has $Out = (\{q_1\} \times \{x \in X | x > 0\}) \cup (\{q_1\} \times \{x \in X | (-3 < x < 0)\}) \cup \{q_2\} \times \{x \in X | x > 0\}$.

Knowing these last definitions it is possible to say that an hybrid system H is deterministic ([1]) if and only if $\forall (q, x) \in Reach$. This is true if

1. $x \in G(e)$ for some $e = (q, q') \in E$, then $(q, x) \in Out$;

2. $e = (q, q') \in E$ and $e' = (q, q'') \in E$, then $x \notin G(e) \cap G(e')$;

3. $e = (q, q') \in E$ and $x \in G(e)$, then $R(e, x)$ contains a single element.

From this definition, by item 2., it is possible to say that the hybrid system considered is in fact **non-deterministic** because $x \in G(e_1) \cap G(e_2)$.

Moreover, a hybrid system H is non-blocking if $\forall q \in Q,\ f(q, \cdot)$ is globally Lipschitz continous and if $\forall (q, x) \in Out \cap Reach,\ \exists (q, q') \in E : x \in G(q, q')$ ([1]). Follows that the hybrid system considered in in fact **blocking** because for $\{(q_1, x) : (-3 < x \leq -2)\} \in Out \cap Reach,\ \nexists (q_2, q_1) \in E : x \in G(q_2, q_1)$.

7 Homework 6

Regularize the given hybrid system (which has a Zeno behavior) by adding a timer.

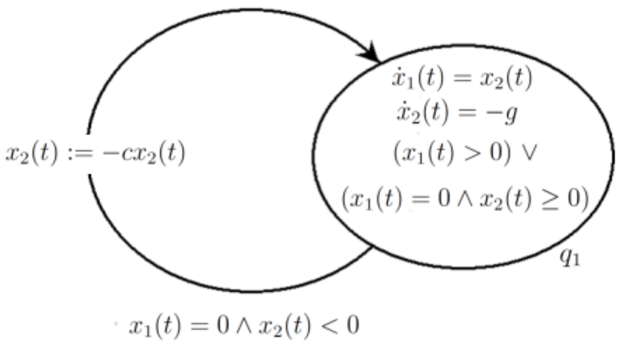

Figure 11: The Problem

7.1 Problem definition

It is possible to notice that the problem considered in Fig.11 is equivalent to the case of the Bouncing Ball Automation (BBA) discussed in the text book [1] and in the article [2].

The problem considers a simple model of an elastic ball bouncing on the ground where

- x_1 denotes the vertical position,
- x_2 denotes the vertical velocity.

The model is represented as a simple hybrid system with a single discrete state and a continuous state.

$$H = (Q, X, Init, f, Inv, E, G, R) \tag{15}$$

that describes the problem is defined by

$$Q = \{q_1\} \quad X = \mathbb{R}^2 \quad Init = q_1 \times \{x \in \mathrm{X} : x_1 \geq 0\}$$

$$f(q_1, x_1, x_2) : \begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} x_2 \\ -g \end{pmatrix} \quad with \quad g > 0$$

$$Inv(q_1) = \{x \in \mathrm{X} : (x_1 > 0)\} \cup \{x \in \mathrm{X} : [\ (x_1 = 0) \wedge (x_2 \geq 0]\ \}$$

$$E = \{e_1 = (q_1, q_1)\}$$

$$G(e_1) = \{x \in \mathrm{X} : (x_1 = 0,\ x_2 < 0)\}$$

$$R(e_1, x_1, x_2) = \begin{pmatrix} x_1 \\ -cx_2 \end{pmatrix} \quad with \quad c \in [\,0, 1]$$

By [2] it is possible to prove that the Bouncing Ball Automation is Zeno by computing two successive bounces. However, intuitively, it is possible to say that for small values of x_1 the ball will bounce very fast to meet the guard condition; this leads to a Zeno behaviour.

In general to avoid the Zeno Behaviour it is good practice to temporal regularize the hybrid system by adding a timer.

7.2 Discussion and Formulation

It is assumed that the bounces are not instantaneous, this means that each bounce of the ball takes time $\varepsilon > 0$.

By adding a new discrete state (a shaded state) q_2 which models the delay through the differential equation

$$\dot{x}_3 = 1 \tag{16}$$

The Eq.16 is the timer that counts the time ε required for the ball to bounce. To do this it is needed to suppose that $x_1(t)$ is lower than zero because of the "bouncing effect" on the ball and depends on the value of the velocity x_2 after each reset.

It is known by [2] that regularization of the hybrid model H requires constructing a family of deterministic, non-blocking and non-Zeno automata H_ε parameterized by a $\varepsilon > 0$ and a continuous map $\phi : \mathrm{Q}_\varepsilon \times \mathrm{X}_\varepsilon \to \mathrm{Q} \times \mathrm{X}$ that relates the state of each H_ε to the state of H.

Referring to the new model of the Bouncing Ball Automation, as BBA_ε, its state is related to the state of the original BBA by the continuous map

$$\phi(q_1, (x_1, x_2, x_3)) = \phi(q_2, (x_1, x_2, x_3)) = (q_1, (x_1, x_2))$$

The new hybrid model

$$H_\varepsilon = (Q_\varepsilon, X_\varepsilon, Init_\varepsilon, f_\varepsilon, Inv_\varepsilon, E_\varepsilon, G_\varepsilon, R_\varepsilon) \tag{17}$$

that describes the new problem is defined by

$$Q_\varepsilon = \{q_1, q_2\} \quad X_\varepsilon = \mathbb{R}^3 \quad Init_\varepsilon = q_1 \times \{x \in \mathrm{X} : x_1 \geq 0\}$$

$$f_\varepsilon(q_1, x_1, x_2, x_3) : \begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{pmatrix} = \begin{pmatrix} x_2 \\ -g \\ 0 \end{pmatrix}$$

$$f_\varepsilon(q_1, x_1, x_2, x_3) : \begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

$$Inv_\varepsilon(q_1) = \{x \in \mathrm{X}_\varepsilon : (x_1 > 0)\} \cup \{x \in \mathrm{X} : [\ (x_1 = 0) \wedge (x_2 \geq 0]\ \}$$
$$Inv_\varepsilon(q_2) = \{x \in \mathrm{X}_\varepsilon : x_3 \leq \varepsilon\}$$

$$E_\varepsilon = \{e_1 = (q_1, q_2),\ e_2 = (q_2, q_1)\}$$

$$G_\varepsilon(e_1) = \{x \in \mathrm{X}_\varepsilon : (x_3 \geq \varepsilon)\}$$
$$G_\varepsilon(e_2) = \{x \in \mathrm{X}_\varepsilon : (x_1 = 0,\ x_2 < 0)\}$$

$$R_\varepsilon(e_1, x_1, x_2, x_3) = \begin{pmatrix} x_1 \\ x_2 \\ 0 \end{pmatrix}$$
$$R_\varepsilon(e_2, x_1, x_2, x_3) = \begin{pmatrix} x_1 \\ -cx_2 \\ x_3 \end{pmatrix}$$

Fig.12 represents the BBA_ε.

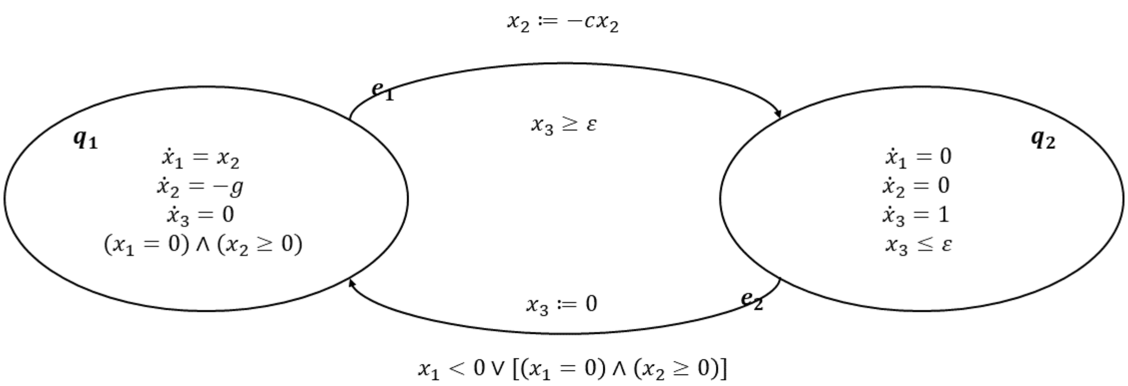

Figure 12: The Automata that solves Homework 6

8 Conclusions

To sum up, to solve Homework 1 (2) I have used Finite State Automata (1), Game Theory applied to optimal control and Theory of Sets [3] knowledge to prove that the sets $W \backslash UPre(W)$ and $W \cap CPre$ are equivalent and so that the two algorithms in Fig.1 are identical. To deal with Homework 2 (3), Homework 3 (4) and Homework 4 (5) I have exploited mechanics knowledge from previous studies and I have applied the definition of Hybrid System H as discussed in Introduction (2). To address Homework 5 (6) I have applied the definitions of deterministic and non-blocking hybrid systems as well as the definitions of reachable *Reach* and outside *Out* sets to define the reachable and outside states to the directed graph in Fig.10. In Homework 6 (7) I have recognised that the direct graph in Fig.11 represented the Bouncing Ball Automation Problem(BBA). I have studied in detail the formulation and solution of [1] and [2]. After I have recognised the Zeno Behaviour of BBA and I have regularized it through the addition of a timer, just as the authors of the references cited have done.

References

[1] John Lygeros Shankar Sastry and Claire Tomlin. *Hybrid Systems: Foundations, advanced topics and applications*. 2020.

[2] Karl Henrik Johansson Magnus Egerstedt John Lygeros Shankar Sastry. On the regularization of zeno hybrid automata. *Systems Control Letters*, 38:141–150, 1999.

[3] Youmath insiemi.

[4] Leslie Lamport. *LaTeX: a Document Preparation System*. Addison Wesley, Massachusetts, 2 edition, 1994.