



HAL
open science

Maximal Permissive Timed Control for a Class of Hybrid Systems

Sonia Batis, Hassane Alla

► **To cite this version:**

Sonia Batis, Hassane Alla. Maximal Permissive Timed Control for a Class of Hybrid Systems. WODES 2012 - 11th International Workshop on Discrete Event Systems, Oct 2012, Guadalajara, Mexico. pp.157-162. hal-00744441

HAL Id: hal-00744441

<https://hal.science/hal-00744441>

Submitted on 23 Oct 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Maximal Permissive Timed Control for a Class of Hybrid Systems

Sonia Batis*. Hassane Alla**

*Gipsa-lab, Grenoble, France, (e-mail : sonia.batis@gipsa-lab.grenoble-inp.fr)

**Gipsa-lab, Grenoble, France, (e-mail : hassane.alle@gipsa-lab.grenoble-inp.fr)

Abstract: A timed control synthesis approach is proposed for a class of hybrid systems modeled with rectangular hybrid automata. The control objective is to constrain the reachable state spaces by the addition of control specifications which can be inconsistent with the system dynamics. Our approach is within an automatic control view and is based on the offline computation of the system new transition guards determined from the reachable state spaces. The approach is illustrated via a production system.

Keywords: control synthesis, rectangular hybrid automaton, forward analysis, reachable state space, maximal permissive controller.

1. INTRODUCTION

In this paper, we study the timed control of a class of systems broadly known as hybrid processes. This type of process can be found in numerous real life automated systems.

This work belongs to the general framework of the Ramadge and Wonham theory (Ramadge et al, 1987) for the control of discrete event systems. These systems are modeled as generators of formal languages; the adjunction of a control structure allows limiting the language generated by the system by accordingly enabling and disabling events. Brandin and Wonham (Brandin et al, 1992) have then extended the theory to the timed discrete event systems by adding discrete timing features to the initial systems.

Maler (Maler et al, 1995) introduced an extension to the Ramadge and Wonham theory, working in the timed automaton framework (suggested by Alur and Dill (Alur et al, 1994)) and defining the notion of timed games. Later, Cassez (Cassez et al, 2005) used this notion to introduce an efficient on-the-fly algorithm for the analysis of timed automata.

Lots of automated systems evolve according to continuous sub processes which are started and stopped by discrete state orders. Therefore, processes have rarely a purely discrete or continuous behavior but a mixture between both of them. These dynamical systems with a double behavioral component are called hybrid systems and can be modeled by many tools among which there is the hybrid automaton (Henzinger, 1996). This model can be considered as a generalization of the timed automaton model (Alur et al, 1994), where the continuous evolution is no longer represented by clocks but by differential equations on the continuous variables of the system.

Henzinger (Henzinger et al, 1999) has studied the control problem of hybrid automata by introducing the hybrid games. The game proceeds in an infinite sequence of rounds and produces an ω -sequence of states. In each round, both players independently choose enabled moves; the pair of chosen moves either results in a discrete state change, or in a passage of time during which the continuous state evolves. In the

special case of a rectangular game, the enabling condition of each move is a rectangular region of continuous state, and when time advances, then the derivative of each continuous variable is governed by a constant differential inclusion. The control problem for hybrid games asks: given a hybrid game B and a formula φ over the discrete states of B , is there a strategy for player 1 so that all possible outcomes of the game satisfy φ ?

Spathopoulos (Spathopoulos, 2000) considered the problem of supervisory control for rectangular hybrid automata and established a supervisory controller that can disable only discrete-event transitions in order to solve the non-blocking forbidden state problem.

All these approaches are within a computer science point of view. Our work is within an automatic control point of view, and as researchers in this domain, we need to synthesize an effectively implementable control which requires realistic assumptions.

In this article, we present an approach for the timed control synthesis of a class of hybrid systems modeled by rectangular hybrid automata. The control is obtained via the automaton new transition guards modified by the addition of desired constraints to the system. It is based on the reachable state spaces computation for any location. Often, in this research field the formal ideas are complex; that is why we start this paper with an intuitive presentation.

It is organized as follows. In Section 2, we present an intuitive example to introduce our main ideas. In Section 3, we define the model assumptions and introduce the reachable state spaces. Section 4 develops our approach of control. This approach will be applied to a production system in Section 5 and a conclusion will be given in Section 6.

2. INTUITIVE PRESENTATION

In order to facilitate the presentation of our approach, we introduce an intuitive presentation based on a simple real example.

We consider a traffic section that can tolerate a maximum number of 200 cars (we suppose that the average distance between two cars is 4 meters, $L=4m$). The cars arrive to the section with an average speed of 36 km/h and leave it with an average speed of 42 km/h.

The example is shown in Fig. 1.

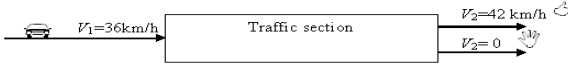


Fig. 1. Example of the traffic section

We suppose that there is a regulation of the traffic that can be done either by a traffic light or by a policeman. Two situations are possible: authorized circulation ($V_2=42$ km/h) and unauthorized circulation ($V_2=0$). The transition from one situation to another can be done at any time, hence the intervention of a clock, denoted by 'h' and expressed in minutes. Initially, the number of cars is equal to 80. Then, the example can be represented by the automaton shown in Fig.2., taking as state variable x_1 the number of cars in the section, and considering the flow as given in the expression:

$$\dot{x}_1 = \frac{V_1 - V_2}{L}$$

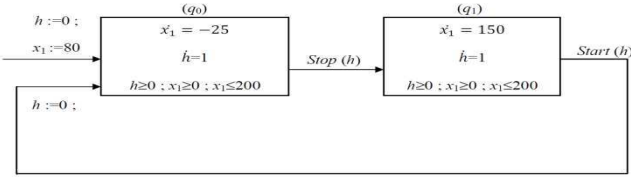


Fig. 2. The traffic section automaton model

Each automaton location is defined by: the variables (x_1 et h), the dynamics of the variables (ex: $\dot{x}_1 = -25$) and the invariants (ex: $x_1 \leq 200$: maximum capacity of the section). The events "Stop" and "Start" are controllable, i.e the occurrence of these events can be forced in their existence intervals.

Now, suppose that we need to add a control in order to limit for example the number of cars in the section. We then consider that a policeman wants that: when the circulation is authorized, the section contains at most the quarter of its capacity ($x_1 \leq 50$), and when the circulation is forbidden, the number of cars is between 30 and 40 ($30 \leq x_1 \leq 40$).

All specifications are added to the automaton as show in Fig.3.

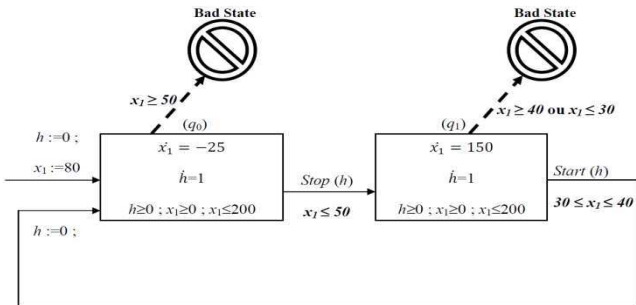


Fig.3. The traffic section constrained automaton

In order to satisfy these different specifications it is necessary to compute new transition guards, this is the asked control problem. For example, the commutation between Locations (q_0) and (q_1) will occur at any time so that the number of cars is less than 50.

However, the specifications are imposed independently from the dynamics of the system then they may not be achievable. A simple indicator can highlight this situation; it is the duration of stay in a location which will be negative. Then, it will be necessary to reduce the commutation guards.

The problem is to find the clock intervals of the commutations between the different locations of the system, and therefore, to reduce the intervals found in the study of the uncontrolled behavior. We want consequently the maximal permissive behavior.

3. PROCESS AND SPECIFICATION MODELING

In this section, we first define the model assumptions. Then, we introduce the control specifications with the notion of reachable state space and the convexity problem that guaranties the convexity of the solution.

3.1 The model assumptions

The model chosen here is the Rectangular Hybrid Automaton (RHA) as shown in the intuitive presentation. This model can be considered as a generalization of the Timed Automaton model. In the case of the RHA, the continuous evolution is represented by differential equations on the continuous variables of the system. Our model has specific assumptions; its definition is derived from the definition of Henzinger (1998).

Definition 1: The general model

A Rectangular Hybrid Automaton (RHA) is defined by the tuple $H=(Q, X \cup \{h\}, \Sigma, T, inv, flux, q_0, E_0^{in})$, where:

- $Q=\{q_1, \dots, q_k\}$ is a finite set of locations representing the discrete states of the system;
- $X \cup \{h\}$ is a finite set of real variables and h is a global clock;
- Σ is a set of controllable events;
- $T \subseteq Q \times \Sigma \times Pred(X) \times Pred(X) \times 2^X \times Q$ is a finite set of transitions where $Pred(X)$ being the set of predicates on X .
- $\sigma \rightarrow h$, a clock h is associated with each controllable event;
- $inv: Q \rightarrow Rect(X \cup \{h\})$ is a function which associates with each location $q \in Q$ a rectangular constraint for any variable $x_i \in X$ and for the clock h ;
- $flow: Q \rightarrow Rect(\dot{X} \cup \{\dot{h} = 1\})$ is the function that assign to each location a dynamic for the continuous evolution.
- q_0 is the initial location;
- E_0^{in} is the initial input space of q_0 ; E_0^{in} is a convex region.

□

For example, the system shown in Fig 3 is modeled by a rectangular hybrid automaton, characterized by the continuous variable x_1 and the global clock h . The example fully meets Definition 1.

3.2 Control specification

Our goal is to establish a timed control by imposing some desired constraints, indicated on the transition guards, on the variables of the system. This control must be maximal permissive.

While imposing certain constraints to the location, the system commutes at a certain moment that reduces the state space (Fig.4.). This behavior joins the idea of forbidden states, introduced by Wonham (1987) in the classical supervisory control.

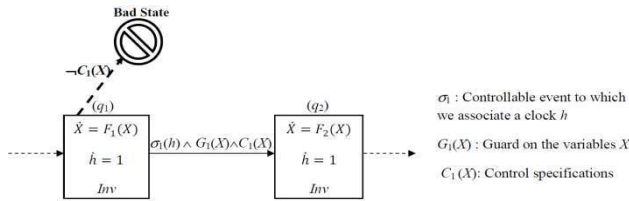


Fig.4. Specification of the control

Definition 2: Constraint: A constraint is a conjunction of polyhedral inequalities over X of the form $c_1x_1 + \dots + c_kx_k \sim c$, where $x_1, \dots, x_k \in X$, $c, c_1, \dots, c_k \in \mathbb{Z}$ and $\sim \in \{<, \leq, =, \geq, >\}$. It is called a polyhedral predicate over X and is denoted $C(X)$.

Definition 3: Control specification: $T \rightarrow (X \rightarrow Pred(X))$, a control specification is a function that assigns to each transition a constraint over X .

For example in Fig.4. above, a constraint $C_1(X)$ has been added in the guard between locations (q_1) and (q_2) .

In the sequel, the automaton to which are added the desired constraints in the transition guards will be called the *structural constrained automaton*.

3.3 Reachable state space

All calculations to be used are starting from formally determining the reachable state spaces by a RHA. This determination is based on the forward analysis of the *structural constrained automaton*.

- **Reachable state:** A state (q, \mathbf{v}) is called *reachable*, if it exists an execution of H that joins this state from an initial state $(q_0, \mathbf{v}_0) \in E_0^{in}$.
- **Forward analysis:** The forward analysis operators are used in order to compute all possible trajectories of the system. This leads to compute the state spaces associated to the stays of the system in each location of the model. One location can be reached with different continuous state spaces at its entrance, especially when the model contains cycles. Consequently, the state space associated to a location is equal to the conjunction of the continuous state spaces of all possible visits to the location.

The forward analysis is realized using the software PHAver (Frehse, 2008). This software provides commands for computing reachable sets of states and simulation relations plus a number of commands for the manipulation and output of data structure. Its language is as user friendly as possible.

The forward analysis of the constrained automaton leads to the state space reduction and can also lead to a state removal. It means that some states may not be reachable. This can be due to the fact that the constraints are stronger than the dynamics. The obtained automaton is called *reachable constrained automaton*.

Remark 1: In order to guarantee the termination of the algorithm, we have to add another assumption to the constrained RHA. This assumption is expressed as follows: For any loop, it exists at least one transition $(q_i, \mathbf{v}_i) \rightarrow (q_{i+1}, \mathbf{v}_{i+1})$ such that we have the assignment over the global state X and the clock h , $(X, h) := (X_0, h_0)$; (X_0, h_0) is a convex region. These transitions will be called initialized transitions. This assumption is often in real life automated systems.

The state spaces obtained from the forward analysis of the constrained automaton shown in Fig.3. are given in Fig.5. below. The constraints are shown in bold italic.

For example, Location (q_1) is characterized by the following inequalities:

$$-150h + x_1 \geq -480; x_1 \leq 200; 25h + x_1 \geq 80; 150h - x_1 \geq 130$$

They express the linear constraints between the variable x_1 and the global clock h . They provide a bijective characterization of the state space and this formalization will allow us to compute the values of the control clock.

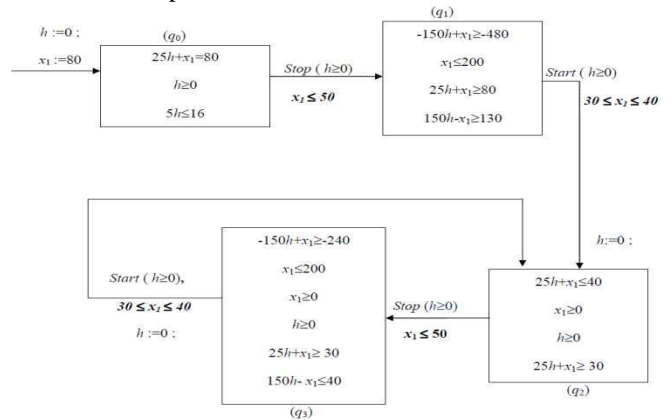
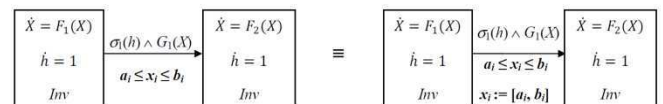


Fig.5. Forward analysis of the constrained automaton

Proposition 1: If the control specification is expressed as $a_i \leq x_i \leq b_i$, then it corresponds at most to an initialization for the variable x_i at the entrance of the successor location.

Proof: It is obvious that the following two schemas are equivalent. Therefore, having a constraint of type $a_i \leq x_i \leq b_i$ corresponds to affecting to x_i the interval $[a_i, b_i]$.



Remark 2: All the reachable state spaces of the automaton are convex. In fact, in each state space, we have only inequalities and intersections between them. It has been proved (J.Ch.Gilbert, 2007) that the image of a convex set by an affine application is convex and that the intersection preserves convexity.

4. COMPUTATION OF THE MAXIMAL PERMISSIVE CONTROL

Finding the maximal permissive controller consists in finding all maximal guards of commutation that respect both the invariants and the constraints.

4.1 New guards

The base of the control computation amounts to solve the general problem for a transition between two locations of the automaton, which model is shown in Fig. 7.

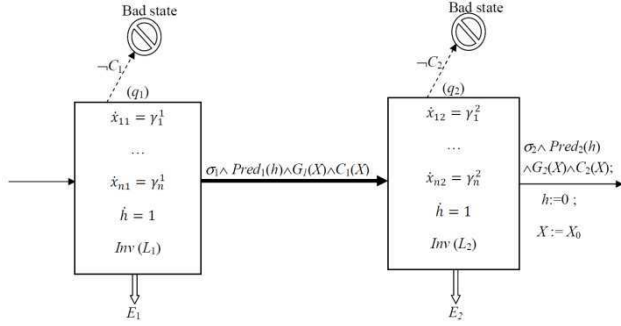


Fig.7. Computation of the maximal permissive control

It is obvious that in each location, the time derivative of any variable x_i has a constant sign. This characterization corresponds to the behavior of most real-life systems.

We also suppose that each location has only one output transition as mentioned in Remark 1.

The constraints are defined regardless of the process. There is then no guarantee on their achievability. In fact, they can be inconsistent with the dynamic of the location. This amounts to calculate the maximal duration of stay in the location in question, denoted δ_i , for each variable x_i . This value has to be positive, since it is a time value. For a location, it can happen that an input value of the constraint variable is greater than its output value, then $\delta_i < 0$ in case of a positive flow. This will be the key idea of our controller synthesis approach.

As we mentioned in the definition of the RHA, all the events are controllable, this means that we are able to modify the value of the associated clock. The case of uncontrollable events will be treated in future work.

We first compute the guard between Locations (q_1) and (q_2) , which is influenced by the behavior in (q_2) (Fig. 7). The demonstration starts where there is a state reset, i.e., downstream Location (q_2) for our case. In fact, we have cut the problem into parts by resetting the state in each loop, so that we don't have any link with the downstream. Then, the computation of the guard is simple starting where the state is reset. As we mentioned earlier in Remark 2, the transition where the state is reset is called the initialized transition.

In the following, we will denote by U_i the initial guard which is the conjunction $Pred_i(h) \wedge G_i(X) \wedge C_i(X)$. The new guard that we compute will be denoted by V_i .

The problem is decomposed into two parts. We first compute the control for the initialized transition that is completely decoupled from the downstream. Then, we go upstream and compute the control transition by transition until we find again a state initialization.

Proposition 4: Initial control

The guard for the initialized transition is $V_2 = E_2 \wedge U_2$. □

Proof:

Since the initialized transition is completely decoupled from the downstream, the new guard in this transition takes into account only the initial guard and the state space upstream this transition. □

Theorem 1: Upstream control

The guard from location (q_1) is V_1 such that:

- Case $\gamma_i^2 > 0$: $V_1 = E_1 \wedge U_1 \wedge \sup(E_2 \wedge U_2)$ (1)

Where $\sup(E_2 \wedge U_2) = \bigwedge_{i=1}^n x_i \leq \max_{x_i}(E_2 \wedge U_2)$

- Case $\gamma_i^2 < 0$: $V_1 = E_1 \wedge U_1 \wedge \inf(E_2 \wedge U_2)$ (2)

Where $\inf(E_2 \wedge U_2) = \bigwedge_{i=1}^n x_i \geq \min_{x_i}(E_2 \wedge U_2)$ □

Proof:

For a location (q_2) , we have:

$E^{in} = U_1 \wedge E_1$, such that U_1 corresponds to the guard associated to the transition $(L_1) \rightarrow (L_2)$.

$x_i^{in} \in U_1 \wedge E_1$; x_i^{in} is any element of E^{in}

Let's compute the duration of stay in the location (q_2) for the variable x_i :

$x_i = \gamma_i^2 \cdot (h_i - h_i^{in}) + x_i^{in}$ where $(h_i - h_i^{in})$ is the duration of stay in location (q_2) for the variable x_i .

$$\Leftrightarrow (h_i - h_i^{in}) = \frac{x_i - x_i^{in}}{\gamma_i^2}$$

This duration of stay is maximal and is denoted δ_i when x_i verifies the maximal output guard of the location (q_2) :

$$\delta_i = \frac{\max_{x_i}(E_2 \wedge U_2) - x_i^{in}}{\gamma_i^2}$$

This duration of stay has to be positive because it is a time value:

$$\delta_i \geq 0 \Leftrightarrow \text{if } \gamma_i^2 > 0: x_i^{in} \leq \max_{x_i}(E_2 \wedge U_2) \Leftrightarrow \forall x_{i(i=1..n)} \in E^{in}, x_i \leq \max_{x_i}(E_2 \wedge U_2).$$

We denote by $\sup(E) = \bigwedge_{i=1}^n x_i \leq \max_{x_i}(E)$ hence the relation (1) above.

For $\gamma_i^2 < 0$, the proof is symmetric. In this case,

$$\delta_i = \frac{\min_{x_i}(E_2 \wedge U_2) - x_i^{in}}{\gamma_i^2} \text{ and } \delta_i \geq 0 \Leftrightarrow x_i^{in} \geq \min_{x_i}(E_2 \wedge U_2). \quad \square$$

For example, let us extract from the traffic section reachable constrained automaton the initial two locations (Fig. 5). In this case, we have:

- $V_1 = E_1 \wedge U_1 = \{30 \leq x_l \leq 40 \wedge h \geq 0 \wedge -150h + x_l \geq -480 \wedge 25h + x_l \geq 80 \wedge 150h - x_l \geq 130\}$

- $V_0 = E_0 \wedge U_0 \wedge \text{sup}(E_1 \wedge U_1) = \{x_1 \leq 40 \wedge h \geq 0 \wedge 25h + x_1 = 80 \wedge 5h \leq 16\}$

Remark 3: The boundary $x_1 \leq 50$ has been reduced to $x_1 \leq 40$. In fact, since the flow increases in Location (q_1), the input value of the variable x_1 has to be less than the output one. \square

4.2 Maximal permissive controller

It is interesting at this stage to show the system controller where commutation clock intervals appear. These intervals contain all possible trajectories of the reachable constrained automaton, and correspond to the maximal permissive controller. Since all the RHA events are controllable, we can replace the initial clock values by the computed ones. In order to compute these intervals, we use the linear programming as follows:

Proposition 5: Clock commutation intervals

The clock commutation interval in the transition downstream Location (q_i) is $h_i = [h_{i\min}, h_{i\max}]$, such that:

- * $h_{i\min} = \text{Minimum } h$
 - * $h_{i\max} = \text{Maximum } h$
- with the constraints: V_i

V_i being the new guard in the transition downstream Location (q_i), it is chosen depending on whether or not the transition is initialized (Proposition 4 and Theorem 1). \square

Proof :

The interval h_i is never empty because Location (q_i) is reachable in the *reachable constrained automaton*. \square

Our objective being to construct the system timed controller that respects in a maximal permissive way the desired constraints, we have to iterate what was established starting with the initialized transition. The global algorithm has been developed in an internal note (Batis et al, 2012). However, we have not added it in this paper due to a lack of space.

Remark 4: The structure of the offline controller automaton is exactly the same as the corresponding RHA. To obtain the controller, we empty each location of the RHA leaving only the clock h . \square

The application of the approach for the traffic section example gives the controller shown in Fig.8.

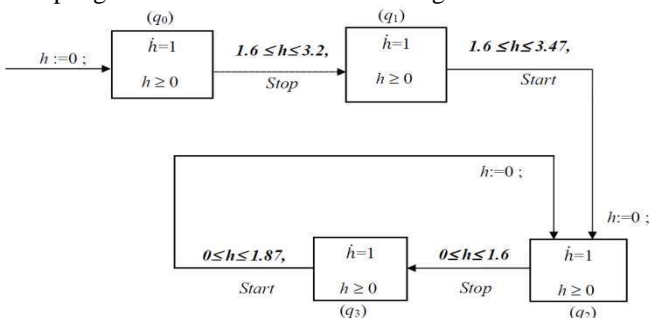


Fig.8. Maximal permissive timed controller of the traffic section example

Remark 5: The solution given above corresponds to the optimal solution, i.e., all the possible trajectories of the *reachable constrained automaton*. It is obvious that in a real-time control, the choice of a commutation value for the first transition after the state reset will have influence in the downstream control. This amounts to determine a particular trajectory among all possible ones.

5. EXAMPLE OF A PRODUCTION SYSTEM

In order to apply our approach, we have chosen a more complex system which is a production one composed of a shared production center that supplies two buffers for consumption with a maximal capacity of 100 for each buffer. The system is shown in Fig.9.

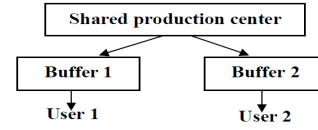


Fig.9. Production system

The state variables are the quantities in each buffer x_1 and x_2 . We assume that, at start, the buffers are half full ($x_1 = x_2 = 50$). We add to those variables a global clock, denoted by h , which determines the delivery dates in a defined interval (controllable events).

The model has only one loop. The balance flow varies with time. The transition ($q_i \rightarrow q_j$) occurs following the controllable event σ_j . The corresponding automaton (for a normal daily behavior), without taking into account the expressions in bold, is shown in Fig.10.

The state changes are non-deterministic. It leads to a degree of freedom for control. For example, the controllable event σ_1 corresponds to an increase of consumption in buffer 1, which can occur for a value of $h \in [6, 8]$.

The states (q_0), (q_1) and (q_2) are reserved to the production in buffer 1. The states (q_3), (q_4) and (q_5) are reserved to the production in buffer 2.

The example meets all hypotheses.

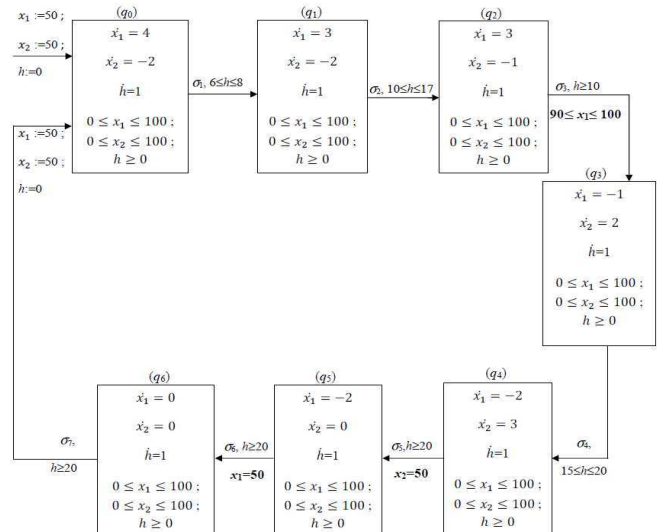


Fig.10. The production system automaton model

We need to impose the following constraints to the system (added in bold in the automaton model shown in Fig.10. above):

- Transition $(q_2) \rightarrow (q_3)$: We need to stop the production in buffer 1 when the level in $x_1 \in [90;100]$
- Transition $(q_4) \rightarrow (q_5)$: The quantity in the second buffer $x_2=50$
- Transition $(q_5) \rightarrow (q_6)$: The quantity in the first buffer $x_1=50$

The other transitions don't have constraints. In order to determine the offline timed controller of this system, we apply our approach developed in Section 4. The forward analysis of our automaton gives 7 state spaces as shown in Fig.11.

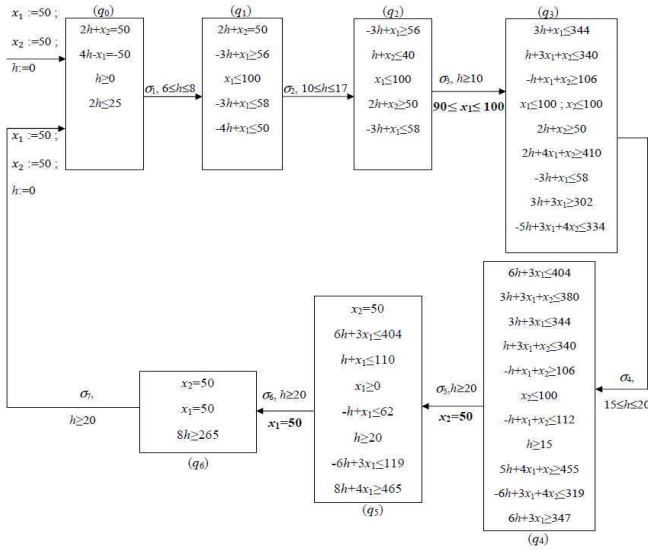


Fig.11. Forward analysis of the constrained automaton

In our automaton, there is only one state reset, which is in the transition σ_7 . Then, it is considered as our starting point for the initial control. After applying our algorithm, we obtain the maximal permissive timed controller shown in Fig.12.

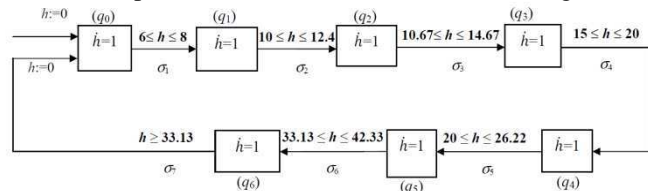


Fig.12. Maximal permissive timed controller of the production system

6. CONCLUSION

In this article, we have introduced an approach for the control of a sub class of hybrid systems modeled by rectangular hybrid automata and characterized by a single global clock. This sub class is interesting since it can represent a great class of real life automated systems. The control is based on the computation of the clock commutation intervals respecting the specification constraints. They are deduced from the reachable spaces of the process and the constrained behavior. For the offline control, we have focused our presentation on

two transitions in sequence with a state reset in the last one since they contain the main original ideas. Then we have presented the way for the generalization of our approach to the whole automaton.

Our future works consists in a first time in implementing a complete synthesis algorithm for the whole model (Batis et al, 2012); and in a second time, we want to extend our approach to the case of a RHA containing uncontrollable events.

REFERENCES

- Alur, R., Dill, D. (1994). A theory of timed automata, *Theoretical Computer Science*, 126(2), 183-235.
- Asarin, E., Maler, O., Pnueli, A., Sifakis, J. (1998). Controller Synthesis for Timed Automata, In Proc. IFAC Symp. *On System Structure & Control*, 469-474. Elsevier Science.
- Batis, S., Alla, H. (2012). Timed control for a class of hybrid systems, *Internal Note*.
- Bradin, B.A., Wonham, W.M. (1994). The supervisory control of timed discrete-event systems, *IEEE Transactions on Automatic Control*, 39, 3357-3362.
- Cassez, F., David, A., Fleury, E., Larsen, K.G, Lime, D. (2005). Efficient On-the-Fly Algorithms for the Analysis of Timed Games, *CONCUR2005*, 66-80
- Frehse, G. (2008). PHAVer: Algorithmic Verification of Hybrid Systems past Hytech, *International journal on software tools for technology transfer (STTT)*, volume 10 number 3.
- Gilbert, J.Ch. (2007), Ensembles convexes, *Eléments d'optimisation différentiable : Théorie et Algorithmes*, p.13-45, to be published.
- Henzinger, T.A. (1996). The theory of hybrid automata, *Hybrid Systems II*, LNCS, vol.999, 278-292.
- Henzinger, T.A., Kopke, P.W. (1997). Discrete-time control for rectangular hybrid automata, In Proceedings of the 24th International Colloquium on Automata, *Languages and Programming*, Lecture Notes in Computer Science 1256, Springer-Verlag, 582-593.
- Henzinger, T.A., Horowitz, B., Majumdar, R. (1999). Rectangular hybrid games, In Proceedings of the Tenth International Conference on *Concurrency Theory*, Lecture Notes in Computer Science 1664, 320-335, Springer-Verlag.
- Maler, O., Pnueli, A., Sifakis, J. (1995). On the synthesis of discrete controllers for timed systems, In Proc. 12th Symp. On *Theoretical Aspects of Computer Science (STACS'95)*, vomume 900, 229-242. Springer.
- Ramadge, P., Wonham, W.M. (1987) Supervisory control of a class of discrete event systems, *SIAM, J.Control and Optimisation*, vol.25, No. 1, 206-230.
- Spathopoulos, M.P (2000). Supervisory Control for Rectangular Hybrid Automata, In Proceedings of the 39th IEEE Conference on *Decision and Control*, Sydney, Australia, 35-41.