Modeling and Simulation of Hybrid Systems Using a Special Class of Timed Petri Net

ABOLFAZL JALILVAND[1], SOHRAB KHANMOHAMMADI[2], FEREIDOON SHABANI NIA[3]
[1]Department of Electrical Engineering
Islamic Azad University of Abhar
[2]Faculty of Electrical and Computer Engineering
University of Tabriz
[3]Faculty of Electrical and Electronics Engineering
Shiraz University
IRAN

Abstract: Dynamic systems with a mix of continuous and discrete components called hybrid systems frequently arise in engineering applications. By modeling these different components using differential equations and discrete events, it is possible to represent a wide range of phenomena present in physical and technological systems. Since many of these applications are safety critical, it is important to use reliable methods to simulate hybrid systems. This paper deals with modeling and simulation of hybrid systems using an extended form of timed Petri net. The proposed method is used for modeling of an illustrative hybrid system example. The simulation results are obtained by a modified Petri net toolbox.

Key-Words: Hybrid systems, Timed Petri nets, Discrete Event, Modeling, Simulation.

1 Introduction

Hybrid systems have emerged as a technique for modeling and analyzing a class of autonomous control systems containing both continuous-state and discrete event dynamics [1-3]. Many physical systems are hybrid in the sense that they have barriers or limitations. Inside the limitations they are modeled with differential/difference equations. A natural way to model these systems is to use a mixture of differential/difference equations and inequalities. Other systems have switches and relays that can be naturally modeled as hybrid systems. These hybrid models appear in many areas. Typical examples are manufacturing systems, transportation systems, flight control, communication networks, missile guidance, process control, robotics, path planning, traffic control and so on [4-9]. The application of hybrid modeling and control can be an interesting approach to improve plant performance, efficiency, safety and reliability.

In general, a hybrid system can be in one of several modes of operation, whereby in each mode the behavior of the system can be described by a system of differential/difference equations (Fig. 1). The system switches from one mode to another due to the occurrence of events [1].

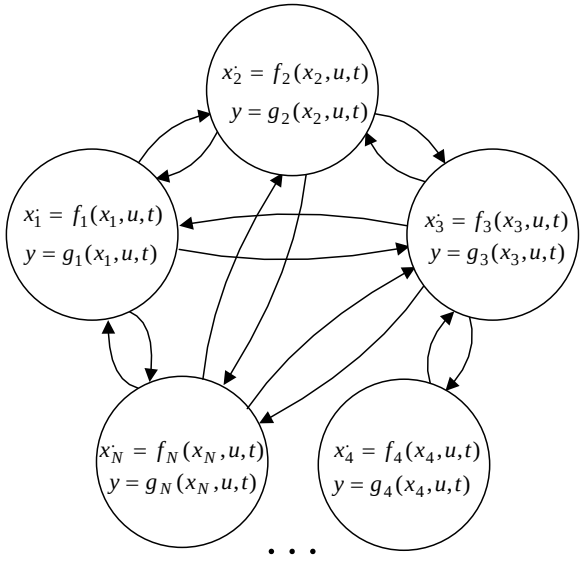

Fig. 1: A schematic representation of a hybrid system with N modes.

In one formulation of hybrid systems the continuous state plant is supervised by a discrete event controller [10-12]. Such a hybrid system consists of three main components: a plant, a discrete event controller and an interface. The continuous process to be controlled, together with any continuous controller, is identified as plant and

is often described in terms of differential/difference equations. It may be desired for a rule based discrete event controller to determine the control and performance objectives of real time plant. The interface makes it possible for these different processes to communicate with each other. As it is illustrated in Fig. 2 it consists of two main parts: the event generator and the actuator. The event generator receives the discrete time measurement signal of hybrid plant and issues appropriate events when the state enters a new region of the state space. The actuator determines the control input which is applied to the hybrid system.

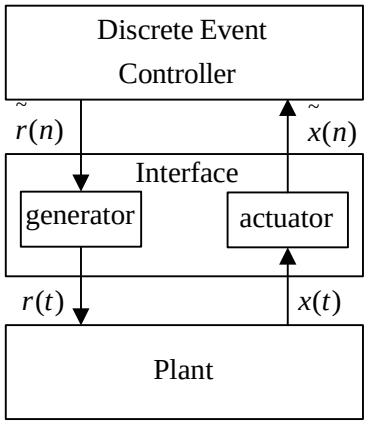

Fig. 2: A hybrid control system.

Modeling and simulation are becoming more important since engineers need to analyze increasingly complex systems composed of components from different domains. A hybrid model, like all models, should be sufficiently complex to capture the rich behavior of the systems. However, it should also be easy enough to be analyzed, and formulated in such a way that simulation is possible. Models of hybrid systems often start from un-timed discrete event systems, adding timing information to events.

The execution of a hybrid system can be defined by a sequence of steps: in each step, the system state evolves continuously according to a dynamical law until a discrete transition occurs. Discrete transitions are instantaneous state changes that separate continuous state evolutions [13]. Due to combination of discrete and continuous behavior, hybrid systems are inherently difficult to analyze formally. Numerical simulation therefore plays an important role in the analysis of such systems. The simulation of hybrid systems alternates between solving differential equations numerically and making discrete transitions. The most used approaches to model hybrid systems are hybrid automata and Petri nets [14].

Petri nets can be viewed as a generalization of finite automata. For Discrete event systems control, Petri nets modeling formalism offers some advantages over finite automaton and it is also useful for hybrid systems control. We can summarize some advantages of the Petri nets as follows [15]:

- They provide an excellent tool for capturing concurrency and conflict within a system.
- They explicitly support parallelism and we are able to express operations like synchronization in a convenient way.
- The graphical and mathematical representation of them causes the expressiveness of Petri nets.
- It have made possible to design supervisors in an efficient and transparent manner.
- They have an inherent quality in representing logic in intuitive and visual way.
- They can be easily combined with other techniques and theories such as programming, fuzzy theory, neural networks, etc.

Regarding the powerful modeling capability of Petri nets , in this paper we use a special version of timed Petri nets as a tool for the mathematical description and simulation of the behaviour of hybrid systems. To simulate the Petri net model of the hybrid system a modified Petri net toolbox is used.

2 Hybrid Dynamical Systems

As it was mentioned previously, a class of hybrid dynamical system, commonly found in the literature, considers the tree layer structure depicted in Fig. 2. Here we deal with this hybrid control system with more details. The continuous system plant is a three-tuple $S = (X, U, f)$, consisting of the state set, input set, state transition function and output function, respectively. The continuous system plant equations are [16]:

$$x^{\cdot}(t) = f(x(t), u(t)) \# \quad (1)$$

Where $x \in X \subset \Re^n$, $u \in U \subset \Re^m$, $f : X \times U \to \Re^n$ is assumed to be lipschitz continuous, for simplicity we assume that the continuous state $x(t)$ can be measured.

The interface equations are:

$$(\Sigma / S) : u(t) = \gamma(r(k)) \# \quad (2)$$

$$(S / \Sigma) : z(k) = \alpha(x(t), u(t)) \# \quad (3)$$

Where $r \in \Re^p$, $u \in U \subset \Re^m$, $x \in X \subset \Re^n$. Function $\alpha : X \times U \to Z$ is a mapping in S / Σ and function $\gamma : R \to U$ is a mapping in Σ / S.

The discrete event controller is a five-tuple (Q, Z, R, δ, ϕ) , consisting of the finite state set,

input set, output set, state transition function and output function, respectively. The discrete event controller is described by the equation:

$$\begin{cases} q(k+1) = \delta(q(k), z(k)) \\ r(k) = \varphi(q(k), z(k)) \end{cases} \# \qquad (4)$$

Where $q(k) \in Q$ is the state after the k th event, $z \in Z$, $r \in R$, $\delta : Q \times Z \to Q$, $\varphi : Q \times Z \to R$, $k \in \mathrm{N}$.

As it is seen from the description of hybrid system, the discrete event controller views the remaining of system as a discrete event system. These events take place when the system passes from one mode to another one. Recognition of current operating mode of the system is based on the information, obtained form the system. Identification of mode depends on the current measured state values of system. In many hybrid systems the events occur when the states or control variables crosses a threshold value. Ofcourse the direction of crossing is usually significant to determine the events. In this paper we use timed Petri net for modeling of hybrid systems. The places denote the possible occurred events in system and each transition represents the dynamic of hybrid system and determines the case when a mode can be switched to its later corresponding mode when a new event occurs.

3 Petri Nets

Petri nets are a graphical and mathematical modeling tool applicable to many systems. They offer formal graphical description possibilities for modeling of systems consisting of concurrent processes. Petri nets have been used extensively as a tool for modeling, analysis and synthesis for discrete event systems. A Petri Net (PN) is a 5-tuple, $PN = (P, T, F, W, M_0)$ where [17]:

$P = \{p_1 \quad p_2 \quad \ldots \quad p_m\}$ is a finite set of places.

$T = \{t_1 \quad t_2 \quad \ldots \quad t_n\}$ is a finite set of transitions.

$F \subseteq (P \times T) \cup (T \times P)$ is a set of arcs (flow relations)

$W : F \to \{1 \quad 2 \quad 3 \quad \ldots\}$ is a weight function.

$M_0 : P \to \{0 \quad 1 \quad 2 \quad \ldots\}$ is the initial marking .

$P \cap T = \Phi$ and $P \cup T \neq \Phi$.

A Petri net structure $N = (P, T, F, W)$ without any specific initial marking is denoted by N, and a Petri net with the given initial marking is denoted by (N, M_0).

The dynamic behavior of a system is modeled by changing the state or marking in Petri nets according to the following (firing) rules:

1- A transition t is said to be enabled if each input place p of t is marked with at least $w(p,t)$ tokens, where $w(p,t)$ is the weight of the arc from p to t.

2- An enabled transition may or may not fire depending on whether or not the event actually takes place (firing conditions are ok).

3- Firing of an enabled transition t removes $w(p,t)$ tokens from each input place p to t and adds $w(t,p)$ tokens to each output place p of t, where $w(p,t)$ and $w(t,p)$ are the weights of the arcs from p to t or t to p respectively.

In graphical representation of a Petri net, places are represented by circles and transitions are shown by hollow bars. The relationship between places and transitions is represented by directed arcs. Also each Petri net can be denoted by two input and output matrices related to transitions of Petri net. For example The Petri net of Fig. 3 depicts the firing of a transition.

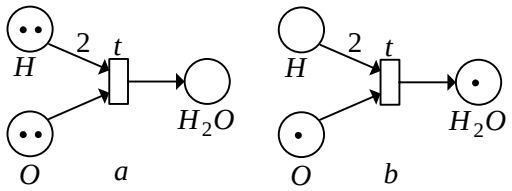

Fig.3: Transition (firing): (a) Marking before firing (b): Marking after firing.

In un-timed Petri net one can prohibit controlled transition from firing but cannot force the firing of a transition at a particular time. In a timed Petri net controlled transitions are forced to fire, by considering the time dependent firing functions. In timed Petri nets, each transition has its specific time which determines the transition's holding time. When a transition is fired during its holding time the network's marking is not changed and as soon as its holding time elapsed the marking of network will be changed based on the mentioned firing rules [18].

To use timed Petri net for modeling hybrid system the discrete events which occurred in system can be demonstrated by places where the occurring of each event is shown by putting a token in corresponding place. Furthermore each continues dynamic of the system is modeled by a transition its firing depicts which dynamic of the system is appeared. In the Petri net we assign to each transition the conditions which determine the firing of transition will be terminated and cause the next event (events) to be occurred. A transition is enabled and it may fire when all its input places (places that are the source to the transition) contain at least a token.

When a transition fires its time starts to increase based on the corresponding dynamic until the specific constraints are occurred. In such a case firing of transition will be terminated and its output transitions will be marked by a token. In deed we can consider such a transition as a timer which its timing starts with its firing and finishes when the corresponding constraints occurred. In other word this timer describes the continuous dynamics of the hybrid system at each operational mode where the hybrid system switches modes based on constraints on the continuous states. In next section we use this class of Petri net for modeling and simulation of the temperature control system as a hybrid system.

4 Temperature Control System

Here a simple but conventional and well known example of hybrid system is used for illustration. The example is a temperature control system [19]. The system consists of a furnace that can be switched on and off. When the furnace is on a continuous input controls the produced heat. The control objective is to control the temperature at a point of the system by applying the heat input at a different point (Fig. 4).

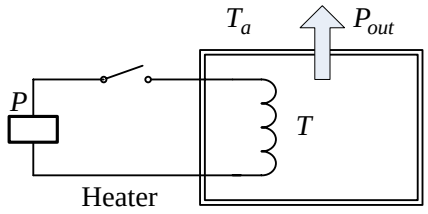

Fig. 4: The furnace with an on/off heater.

Let *m* be the discrete state representing the state of the heater, which is 0 when the heater is off and *P* when it is on. A model describing the temperature in the furnace is:

$$T^{\cdot} = \frac{1}{C}(u - \lambda(T - T_a))\,\# \qquad (5)$$

Where T and T_a are the inside and outside temperatures of the furnace respectively, $C > 0$ is the heat capacity of the furnace and $\lambda > 0$ is the heat transfer number ($P_{out} = \lambda(T - T_a)$). The discrete input r is to put the furnace either on or off, changing the value of u to 0 and P respectively. The temperature decrease to T_a when the heater is off and increase to $P/\lambda + T_a$ when the heater is on. The open loop plant is true hybrid system since it contains both continuous and discrete state variables.

Based on the above description, we have three partitions (modes) for operation of the system:

$m_1 : T < T_l$:# Low temperature mode
$m_2 : T_l < T < T_h$:# Safe mode
$m_3 : T_h < T$:# High temperature mode

The discrete event system representation of furnace system is shown in Fig. 5. In this Figure the resultant event and the corresponding applicable controller (due to the event) is shown on each arc, separated by "/". Note that the null symbol reflects the fact that nothing actually happens in the system at this transition. When the controller receives a *null* symbol it remains in the same state and reissues the current controller symbol. q_1 means that temperature is becoming cold and q_2 means that temperature is becoming hot.

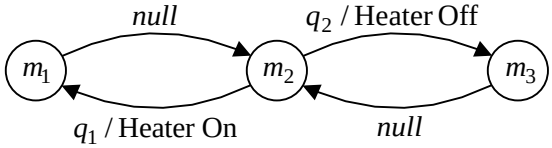

Fig. 5: The discrete event model for the furnace system.

According to the obtained discrete event system model and considering the main occurred events (q_1 and q_2), we can construct the Petri net model of the system. Fig. 6 depicts this Petri model. In this model the places: p_1 and p_2 depict the discrete events: q_1 and q_2 respectively. Furthermore transition t_1 represents the operation mode of the system when the furnace is ON and the statement $T > T_h$ shown beside it depicts the condition when the firing of this transition finishes. Transition t_2 represents the operation mode of the system when the furnace is OFF and the statement $T < T_l$ written beside it depicts the condition when the firing of this transition finishes.

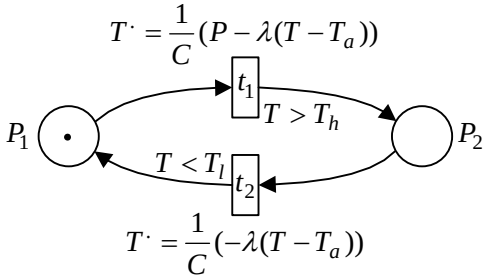

Fig. 6: The Petri net model of the furnace system.

As it can be seen form Fig. 6 the occurrence of each event depends on the current mode of the system (the transition which fires) and the threshold value that defines the event (the corresponding condition related to transition).

5 Simulation Results

For simulating the obtained Petri net model, we use a modified Petri net toolbox [20]. For this purpose we consider the temperature control system where its dynamic equation is $\dot{T} = 0.1(u - 0.4(T - 60))$. For this system it is assumed that: $u = 10$, $T_l = 64^{\circ}F$ and $T_h = 70^{\circ}F$. Furthermore for simulation purpose it is assumed that the initial temperature of system is $60^{\circ}F$ (the outside temperature). The simulation of the system is done for 164 sec whereas the chosen step-time for solving the dynamic equations and simulating the Petri model is 0.05 second.

Fig. 7 shows the simulation results which represent the dynamic behavior of transitions: t_1 and t_2 based on their corresponding dynamics. As it can be seen in Fig. 7 we have assumed that the state of each transition is zero in time intervals that it doesn't fire. Combining the dynamic behavior of all transitions we can obtain the complete behaviour of the hybrid system.

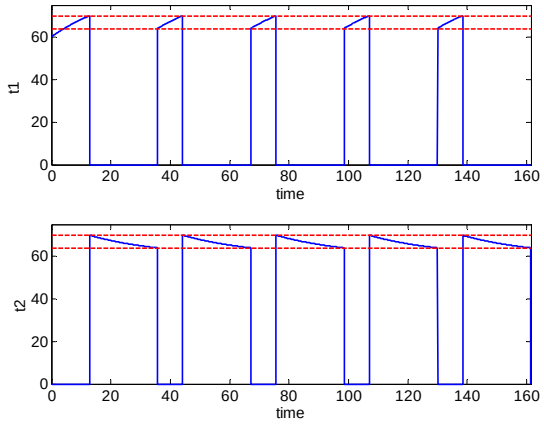

Fig. 7: The simulated dynamic behavior of transitions in Petri model.

Fig. 8 shows the variations of the time of each transition when it fires.

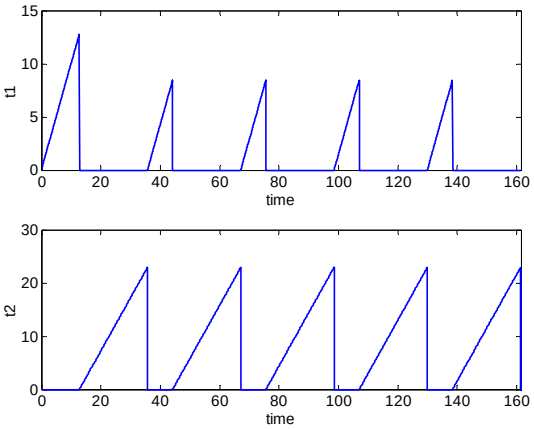

Fig. 8: Variations of the time of transitions.

Again we assume that the time of the transition is zero in the time intervals that it doesn't fire. We can consider each transition as a timer which starts when fires and stops when encounters to the predefined constraints. After stopping, its time reset to zero.

In Fig. 9 state of each transition is shown. In this figure we define the following values for describing the state of the transition.

1 : Firing of the transition finishes.
0 : The transition fires.
-1 : The transition doesn't fire.

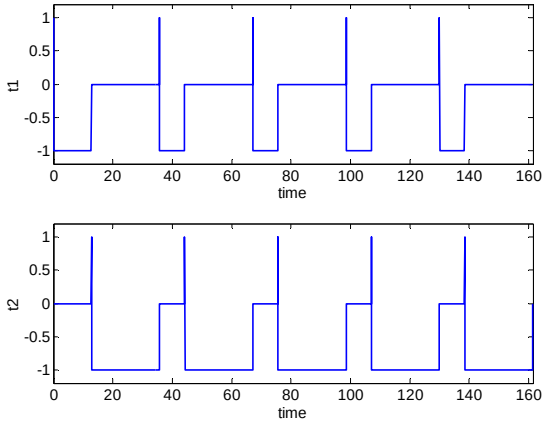

Fig. 9: The states of transitions.

It can be seen in Fig. 9 that the state "1" appears just in an instant when an event takes place in the hybrid system.

Fig. 10 shows the marking of each place. Note that in our Petri net toolbox when the transition fires the token from each input place is removed but putting a token in output place postpones until the firing of the transition finishes.

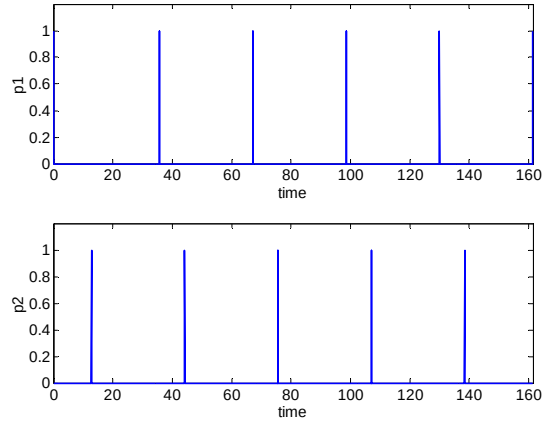

Fig. 10: The marking of each place in simulated Petri model.

Considering both Fig. 10 and Fig. 9 it can be seen that each place has a token just when the firing of its input transition finishes (an event takes place). The

token is removed from the place as its output transition fires (the system enters a new mode).

6 Conclusions

In this paper the modeling and simulation of hybrid systems with a mix of continuous states and discrete dynamics was investigated. For the modeling purpose an extended form of timed Petri net was developed. For simulation purpose a modified Petri net toolbox was used. The proposed method was utilized for modeling and simulation of an illustrative hybrid system example. The Petri net model was simulated by a modified Petri net toolbox. Simulation results represented that the Petri net model can describe both continuous and discrete dynamical behaviour of hybrid systems well.

References:

[1] R. K. Boel, B. De Schutter, G. Nijsse, J. M. Schumacher and J. H. van Schupoen, Approaches to modeling, analysis and control of hybrid systems, *Journal A*, Vol. 40, No. 4, Dec. 1999, pp. 16-27.

[2] W. P. M. H. Heemels, B. De Schutter and A. Bemporard, On the equivalence of classes of hybrid dynamical systems, *Proc. Of the 40th IEEE Conference on Decision and Control*, Orlando, Florida, USA, Dec. 2001.

[3] J. A. Stiver and P. J. Antsaklis, An invariant based approach to design of hybrid control systems, *Int. Journal of Robust and Nonlinear Control*, Feb. 2000.

[4] C. J. Tomlin, Hybrid control of air traffic management systems, *PhD Thesis*, Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, 1998.

[5] E. Frazzoli, Robust hybrid control for autonomous vehicle motion planning, *PhD Thesis*, Department of Aeronautics and Astronautic, MIT, 2001.

[6] R. Fierro, Frank L. Lewis and Andy Lowe, Hybrid control for a class of under actuated mechanical systems, *IEEE Trans. On Systems*, Man and Cybernetics, Vol. 29, No. 6, Nov. 1999, pp. 646-654.

[7] M. Egerstedt and X. Hu, A hybrid control approach to action coordination of mobile robots, *Automatica*, (38), 2002, pp. 125-130.

[8] A. Tornis and M. P. Spathoulos, Supervisory target control for hybrid systems, *Int. Journal of Control*, Vol.76, No. 11, 2003, pp. 1142-1158.

[9] Chutinan, Hybrid system verification using discrete model approximation, *PhD Thesis*, Carnegie Mellon University, USA, 1999.

[10] X. D. Koutsoukos, P. J. Antsaklis, J. A. Stiver and M. D. Lemmon, Supervisory control of hybrid systems, *Proc. Of IEEE*, Vol. 88, No. 7, July 2000, pp. 1026-1049.

[11] X. D. Koutsoukos and P. J. Antsaklis, Design of hybrid system regulators, *Proc. Of 38th IEEE Conference on Decision and Control*, Phoenix, Dec. 1999, pp. 3990-3995.

[12] X. D. Koutsoukos and P. J. Antsaklis, Hybrid control of a robotic manufacturing system, *Proc. Of 7th IEEE Mediterranean Conference on Control and automation*, Haifa, Israel, June 1999, pp. 144-159.

[13] L. Chaimowicz, M. F. M. Campos, and V. Kumar, Hybrid Systems Modeling of Cooperative Robots. *Proceedings of the 2003 IEEE International Conference on Robotics and Automation*, Taipei, Taiwan, September 2003, pp. 4086-4091.

[14] A. Gambier and E. Badreddin, Application of hybrid modeling and control techniques to desalination plants, *Elsevier, Desalination*, 152 (2002), pp. 175–184.

[15] X.D. Koutsoukos and P.J. Antsaklis, Hybrid control systems using timed Petri nets: Supervisory control design based on invariant properties, *Hybrid Systems V, Lecture Notes in Computer Science*. Springer, 1999.

[16] A. Jalilvand and S. Khanmohammadi, Integrating of Event Detection and Mode Recognition in Hybrid Systems by Fuzzy Petri Net, *Proc. of IEEE Conference on Robotics, Automation and Mechatronics (RAM2004)*, December 1-3, 2004, Singapore, pp. 265-270.

[17] Tado Murata, "Petri nets: properties, analysis and application", *Proc. of IEEE*, Vol. 77, No. 4, April 1989, pp. 541-580.

[18] A. Jalilvand and S. Khanmohammadi, Using Petri Nets and Branch and Bound Algorithm for Modeling and Scheduling of a Flexible Manufacturing System, *WSEAS Transaction on Systems*, Issue 7, Vol.3, September 2004, pp. 2580-2585.

[19] S. Pettersson, Analysis and design of hybrid systems, *PhD Thesis*, Chalmers University of Technology, Sweden, 1999.

[20] G. Ribichini, and P. Messina, *Un toolbox per la progettazione la simulazione el' Analisi di Rti di Petri Temporizzate*, Universita Degli Studi Pisa, Facolta di Ingegneria, 7 Marzo 2002.