Using Hybrid Automata to Model Power Electronic Circuits

Márcia Verônica C. Miranda[1] and Antonio Marcus Nogueira Lima[2]
[1]Departamento de Ciências Fundamentais e Sociais
[2]Departamento de Engenharia Elétrica
Universidade Federal da Paraíba
Caixa Postal 10.105 - 58109-970 Campina Grande, PB, Brasil
{miranda,amnlima}@dee.ufpb.br Fax: ++55-83-3101015

***Abstract* - The focus of this paper is to show how representing power electronic circuits with hybrid automata. The paper introduces a methodology for obtaining the hybrid automata from the analysis of the operating modes of the power converter circuit including its associated feedback controller. This methodology is a first step toward the use of formal methods in the field of power electronics.**

I. INTRODUCTION

The use of formal specification and verification techniques in the development physical systems has become more and more important in the computer science and industrial fields. The investigation of efficient methodologies for analyzing the behavior of dynamic physical systems is an everlasting challenge for scientists and engineers [1].

If the system specifications are described in terms of a formal language or with a mathematical model, can express the most relevant characteristics of the modelled systems, avoiding redundancies and ambiguities [2]. Problems in the implementation and test phases are reduced since the use of a formal model provides a way for the designers to find out potential problems in the system specification phase. Formal specifications and mathematical analysis present a way to treat and study enormous state space involved in complex systems. Formal techniques have the potential for increasing safety and decreasing the cost of analyzing complex systems.

The global behavior of a given physical system results from the interactions of its components and can be described through the different states visited during its dynamic evolution [3]. The power electronic circuits are networks composed of electronic components and semiconductor devices. The analysis and design of such networks starts by the numerical simulation of the converter under study. If for the purposes of the study the semiconductor devices are treated as ideal switches the network will have variable topology and the power electronic circuit can be classified as a hybrid system. A hybrid system is a dynamical system whose behavior exhibits both discrete and continuous change [4, 5]. The behavior of these systems is described by continuous dynamic with the intervention of discrete events. In this case the discrete events correspond to the closing and opening of the switches and the continuous time dynamics correspond to the differential equations representing each one of the topologies of the network. There are different paradigms to model hybrid systems [4, 5]. Among all the available models, the use of hybrid automata seems to be quite adequate for representing the behavior of power electronic circuits. Hybrid automata are generalized finite-state machines whose states are associated with a set of differential equations for describing the dynamic specification of the system evolution at that particular configuration and whose arcs model an abrupt change in the dynamic behavior of the system that cause the transition from one configuration to another [6, 7].

Formal methods have had difficulty in gaining acceptance in the industrial field because many consider formal methods to be an approach for system specification and analysis that requires a large learning time. Besides, there is the fact that some types of formal methods have not yet been tested with systems of reasonable complexity [8].

The objective of this paper is to introduce a methodology for obtaining the hybrid automata that represents the behavior of power electronic circuits. The motivation behind this objective is to exploit the use of formal tools in the context of power electronics.

The paper is organized as follows. Section II and III describe the hybrid automaton formalism and the proposed methodology for obtaining the hybrid automata for representing power electronic circuits, respectively. To illustrate the methodology, an example for the zero-voltage quasi-resonant boost converter topology is presented. Section IV presents our conclusions.

II. HYBRID AUTOMATA

Hybrid automata are considered one of major formalisms used to specify hybrid systems. Hybrid automata are a generalization of finite automata whose states describe the dynamic behavior of the modelled system. Transitions between states characterize a change of dynamic profile of the designed system. The system components ($C_i's$) are modelled by independent automata, and the global behavior is obtained by the interaction and integration of these independent automata ($H_i's$). State transitions in a component are caused by the events or messages generated by itself or received from the others components or the external environment. The global system is result of the composition of the component automata.

Definition 1 *A hybrid automaton is a system* $H = \langle X, V, flow, inv, init, E, jump, \Sigma, syn \rangle$ *that consists of the following components [4, 5, 9].*

- *Variables.* A finite ordered set $X = \{x_1, x_2, \ldots, x_n\}$ of real-numbered variables where n is called the dimension of H. It is denoted by $\dot{X}$ the set $\{\dot{x}_1, \dot{x}_2, \ldots, \dot{x}_n\}$ representing the first time derivatives of X. It is denoted by X' the set $\{x'_1, x'_2, \ldots, x'_n\}$ representing values of X at the end of a discrete change.
- *Control mode.* A finite set V of control modes, also called locations.
- *Flow conditions.* A labelling function $flow$ that assigns a flow condition to each control mode $v \in V$. The flow condition $flow(v)$ is a predicate over $X \cup \dot{X}$.

While the control of the hybrid automaton H is in mode v, the vector X evolves along a differentiable curve such that at all points X and $\dot{X}$ satisfy the flow condition $flow(v)$. The set of flow equations is denoted F.

- *Invariant conditions.* A labelling function inv that assigns an invariant condition to each control $v \in V$. The invariant condition $inv(v)$ is a predicate over X. While the control of hybrid automaton H is in mode v, the vector X must satisfy the invariant condition $inv(v)$.
- *Initial conditions.* A labelling function $init$ that assigns an initial condition to each control mode $v \in V$. The initial condition $init(v)$ is a predicate over X. The control of the hybrid automaton H may start in the control mode v when the initial condition $init(v)$ is $true$. In the graphical representation the initial conditions appear as labels on incoming arrows without source nodes. Initial conditions of the form $false$ are not depicted.
- *Control switches.* A finite multiset E of control switches. Each control switch (v, v') is a directed edge between a source mode $v \in V$ and a target mode $v' \in V$ and represents a transition between the locations.
- *Jump conditions.* A labelling function $jump$ that assigns to each control switch $e \in E$ a predicate. Each jump condition $jump(e)$ is a predicate over $X \cup X'$.
- *Events.* A finite set Σ of events, and a labelling function syn that assigns an event in Σ to each control switch $e \in E$.

The state of the hybrid automaton H is a pair (v, a) consisting of a control mode $v \in V$ and a vector $a = (a_1, \dots, a_n)$ that represents a value $a_i \in I\!R$ for each $x_i \in X$.

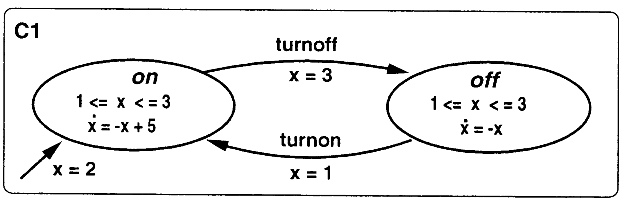

Fig. 1 Simple thermostat automaton

An example of a simple hybrid automaton is shown in Fig. 1 [10]. This automaton is the component C_1 of a heating system. This system also contains the component C_2 as illustrated in the Fig. 2.

The model of the Fig. 1 captures the dynamics of a simple thermostat. The model uses a real variable x, which represents the temperature of the heater. The evolution of x is determined by the initial conditions, the differential equations present in every modes, and by the transitions between modes. The heater is turned off when the temperature reaches 3 degrees, and it is turned on when the temperature falls to 1 degree. In the formal framework defined previously, $X = \{x\}$ and $V = \{on, off\}$. The continuous dynamics for mode *on* is given by $\dot{x} = -x + 5$, which describes an exponential rise of the temperature x. For mode off the continuous dynamics is given by $\dot{x} = -x$, describing an exponential fall in the temperature. For both operating modes, the invariant condition is $1 < x < 3$. The initial condition for mode *on* is $x = 2$. There are 2 transitions in the Fig. 1, namely (on, off) and (off, on). The transition (on, off) fires when the condition given by $x = 3$ is satisfied and similarly the transition (off, on) fires when the condition given by $x = 1$ is satisfied. It is common to use *guards* (G) to represent the conditions for changing the location. The set of discrete events for this example is $\Sigma = \{turnon, turnoff\}$. The function syn associates the event *turnon* to the transition (off, on), and associates the event $turnoff$ to the transition (on, off). The occurrence of these events will synchronize the automata of the system, like for instance with the other component automaton C_2.

A hybrid system typically consists of several components which operate concurrently and communicate with each other. Each component is described as a separated hybrid automaton. The synchronization of the global system is obtained by:

1. The use of shared variables, and
2. Synchronization labels on the transitions are used to model message-type coordination.

The composition automata is obtained as following: The locations of the parallel composition of any two automata A_1 and A_2 are pairs of locations, the first from A_1 and the second from A_2. The location (v_1, v_2) has the conjunction of v_1 and v_2's invariants as its invariant, and the conjunction of their flow conditions as its flow condition. A location is initial iff its components are initial in theirs respective automata. The initial predicate is the conjunction of the initial predicates of the components. Transitions from the components are interleaved, unless they share the same synchronization label, in which case they are synchronized and executed simultaneously [11].

Formally, let A_1 and A_2 are hybrid automata, where $X_1 \cap X_2 = \emptyset$. The *composition automaton* of A_1 and A_2 is a system $A = \langle X_1 \cup X_2, V_1 \times V_2, flow, init, inv, E, jump, \Sigma_1 \cup \Sigma_2, syn \rangle$, where: (i) $flow((v_1, v_2)) = flow_1(v_1) \wedge flow_2(v_2)$; (ii) $inv((v_1, v_2)) = inv_1(v_1) \wedge inv_2(v_2)$; (iii) $init((v_1, v_2)) = init_1(v_1) \wedge init_2(v_2)$.

For the transitions, $e = ((v_1, v_1'), (v_2, v_2')) \in E$ iff one of the following conditions hold:

1. $v_1 = v_1'$, $e_2 = (v_2, v_2') \in E_2$ and $syn_2(e_2) \notin \Sigma_1$; $jump(e) = jump_2(e_2)$ and $syn(e) = syn_2(e_2)$.
2. $v_2 = v_2'$, $e_1 = (v_1, v_1') \in E_1$ and $syn_1(e_1) \notin \Sigma_2$; $jump(e) = jump_1(e_1)$ and $syn(e) = syn_1(e_1)$.
3. $e_1 = (v_1, v_1') \in E_1$, $e_2 = (v_2, v_2') \in E_2$, $syn_1(e_1) = syn_2(e_2)$. In this case, $jump(e) = jump_1(e_1) \wedge jump_2(e_2)$ and $syn(e) = syn_1(e_1)$.

To illustrate how to obtain the composition automata, we consider that the thermostat [See Fig. 1] controls the heating machine shown in Fig. 2. The component C_1 and component C_2 communicate by synchronizing the *turnon* and *turnoff* events. When there is a *turnon* or a *turnoff* in the thermostat automaton (C_1), the synchronization with the corresponding transition in the heating machine automaton (C_2) will occur. This synchronization will eventually change the state of the heating machine. The composition process results on a new automaton. However, this new automaton is usually not written explicitly and is dynamically created during the execution of the component automata.

The decomposition of the system in components is in the very nature of the modelling process by hybrid au-

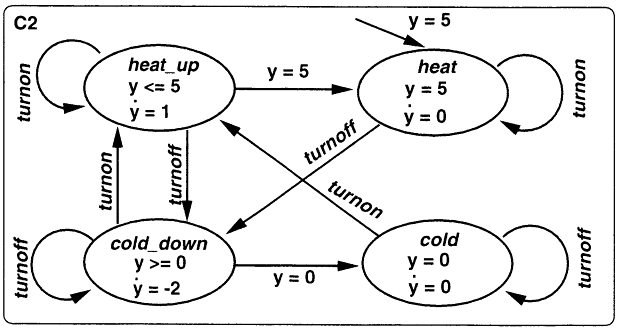

Fig. 2 Heating machine automaton

tomata, and the composition process of these components actually happens in real time. However, during the analysis and specification steps the composition is achieved by the software package employed to study the system.

III. MODELLING METHODOLOGY

For building the hybrid automata models of a complex system, like power electronic circuits, we have used a top-down approach. At first, the designer specify and model the automata for each system components, then the global model is obtained by the composition of these automata, as described before.

This model can represent, initially, the most important characteristics of the system operating with idealized semiconductor devices. The obtained model can be further refined by replacing some idealized semiconductor devices by more detailed equivalent ones. In this case, the hybrid automata equivalent to this device is searched in a hybrid automaton library and then replaced into the original automaton. The hybrid automata library (HAL) contains a set of refined automata models representing the detailed behavior of the semiconductor devices. This type of refinement is usually employed when studying power electronic circuits with the help of simulation packages like PSPICE. The refinement process of the hybrid automata model is illustrated in Fig. 3.

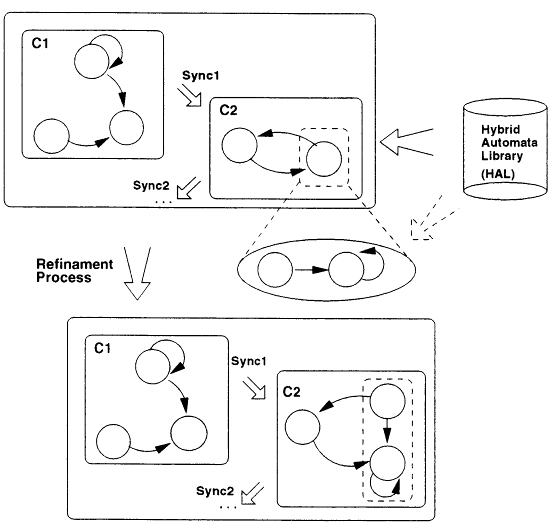

Fig. 3 Diagram of refinement process

The modelling methodology consists of the specification of the components of the circuit by hybrid automata. This requires the definition of all components in $H = \langle X, V, flow, inv, init, E, jump, \Sigma, syn \rangle$. To obtain the model of hybrid automaton that represents a given power electronic circuit, the steps described bellow should be followed:

1. Identify the components of the system ($C_i's$)
2. **For** each identified component (C_i) **do**
   (a) Enumerate all the elements that may change the discrete mode
   (b) Identify all the continuous-time state variables of the component (X's)
   (c) Establish all the operation modes - vertices or locations (V's) - based on identified discrete elements
   (d) Determine the dynamic behavior for each identified operation mode - create the flow equations (F's)
   (e) Establish all the conditions that the system continues in the same operation mode - set of invariants (inv)
   (f) Identify all the transitions between the operations modes that are possible to occur and all events associated to these transitions - edges (E) and events (Σ), respectively.
   (g) Establish all necessary conditions to all transitions occurrences for the each component - guards (G)
3. **EndFor**
4. Define the initial condition for each component ($init$'s)
5. Define the interaction for each component by shared variables or events (syn)
6. Obtain the global model (Mg) by composition.
7. Is it necessary some device refinement ?
   (a) Yes.
      i. Get the refined automaton of this device from the hybrid automata library.
      ii. Go to step 2.
   (b) No.
      i. Global Model obtained.
8. **End of procedure.**

During the modelling process the details of the circuit behavior must be deeply analyzed and then expressed mathematically in the hybrid automaton framework. This forces the designer to deeper his knowledge about converter's operation in a step by step way until the global model is obtained. There is no loss of expressiveness when hybrid automata are used to study power electronic circuits. Besides, it provides a formal model that enables to perform verification and analysis of the converter behavior.

In general, the modelling of power electronic circuit using hybrid automata results in a mathematical model relatively detailed and complex. However, verification and analysis of such model can be successfully done with the aid of (semi) automatic computational tools like CheckMate [12], SHIFT [13] or Simulink. On the other hand, for some analysis a simplified model may be required. The use of reduced complexity models is also treated in the hybrid automata framework by using the approximate automata theory [12,14]. In this theory the global hybrid automaton is converted into discrete and finite models called quotient transition systems or approximating automata. The transitions between states in the approximating automaton are computed using the reachability analysis technique called the flow pipe approximation [12,14].

A. Application: ZVS-QR Boost Converter

The use of the above procedure will be illustrated for the zero-voltage quasi-resonant boost converter shown in Fig. 4.

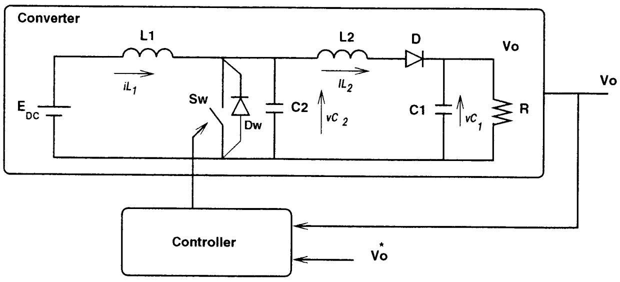

Fig. 4 Circuit of the ZVS-QR boost converter and its feedback controller

In this paper we do not discuss the design of such converter. The reader interested in that subject should look for this in specific paper like [15].

The modelling steps are easily identified as following:

Step 1 The components ($C_i's$) of this system are the power converter and the feedback controller.

A. ZVS-QR Boost Converter

For this converter, we build the respective hybrid automaton according to the following steps:

Step 2(a) The components that may cause discrete mode changes are the switch (Sw), the diode (D) and an anti-parallel diode (Dw). The current flowing through the switch Sw is assumed to be unidirectional.

Step 2(b) The continuous-time variables are i_{L1}, i_{L2}, v_{C1}, v_{C2} and E_{DC}. However, it is assumed that E_{DC} is constant and it will be removed from the vector X. The subscripts in the names of the variables correspond to the labels of the components shown in the Fig. 4.

Step 2(c) In this case there are eight discrete modes as shown below.

Mode	Sw	D	Dw
1	Off	Off	Off
2	Off	On	Off
3	On	On	Off
4	On	Off	Off
5	Off	Off	On
6	Off	On	On
7	On	On	On
8	On	Off	On

The designer must analyze if all the discrete mode are physically possible. In this circuit, the modes 7 and 8 are not possible because when the switch Sw is turned on the anti-parallel diode Dw does not turn on. Then, in this case, the discrete modes are represented by $V = \{Loc1, Loc2, Loc3, Loc4, Loc5, Loc6\}$.

Step 2 (d), (e), (f) and (g) The execution of these steps can be done simultaneously as shown in the following. The switching cycle of this converter starts when Sw, Dw and D are all turned-off ($Loc1$). The initial condition is $i_{L1} = I_1$, $i_{L2} = 0$, $v_{C1} = V_0$, $v_{C2} = 0$, where $I_1 = 0.08A$ and $V_0 = 16V$ and the state vector is defined as $X = [\ i_{L1} \quad i_{L2} \quad v_{C1} \quad v_{C2}\ ]^T$.

The set of invariant conditions and the set of differential equations representing the flow conditions for the all operation modes are:

Loc1: Invariant condition - $i_{L2} = 0, v_{C1} \geq 0, v_{C2} > 0, v_{C2} \leq v_{C1}$

$$\dot{X} = \begin{bmatrix} 0 & 0 & 0 & -\frac{1}{L_1} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{1}{RC_1} & 0 \\ \frac{1}{C_2} & 0 & 0 & 0 \end{bmatrix} X + \begin{bmatrix} \frac{1}{L_1} \\ 0 \\ 0 \\ 0 \end{bmatrix} E_{DC} \quad (1)$$

Loc2: Invariant condition - $i_{L1} > 0, i_{L2} > 0, v_{C1} > 0, v_{C2} > 0, v_{C2} > v_{C1}$

$$\dot{X} = \begin{bmatrix} 0 & 0 & 0 & -\frac{1}{L_1} \\ 0 & 0 & -\frac{1}{L_2} & \frac{1}{L_2} \\ 0 & \frac{1}{C_1} & -\frac{1}{RC_1} & 0 \\ \frac{1}{C_2} & -\frac{1}{C_2} & 0 & 0 \end{bmatrix} X + \begin{bmatrix} \frac{1}{L_1} \\ 0 \\ 0 \\ 0 \end{bmatrix} E_{DC} \quad (2)$$

Loc3: Invariant condition - $i_{L1} > 0, i_{L2} > 0, v_{C1} > 0, v_{C2} = 0$

$$\dot{X} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{1}{L_2} & 0 \\ 0 & \frac{1}{C_1} & -\frac{1}{RC_1} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} X + \begin{bmatrix} \frac{1}{L_1} \\ 0 \\ 0 \\ 0 \end{bmatrix} E_{DC} \quad (3)$$

Loc4: Invariant condition - $i_{L1} > 0, i_{L2} = 0, v_{C1} > 0, v_{C2} = 0$

$$\dot{X} = \begin{bmatrix} 0 & 0 & 0 & -\frac{1}{L_1} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{1}{RC_1} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} X + \begin{bmatrix} \frac{1}{L_1} \\ 0 \\ 0 \\ 0 \end{bmatrix} E_{DC} \quad (4)$$

Loc5: Invariant condition - $i_{L2} = 0, v_{C1} > 0, v_{C2} = 0$

$$\dot{X} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{RC_1} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} X + \begin{bmatrix} \frac{1}{L_1} \\ 0 \\ 0 \\ 0 \end{bmatrix} E_{DC} \quad (5)$$

Loc6: Invariant condition - $i_{L2} > 0, v_{C1} \geq 0, v_{C2} = 0$

$$\dot{X} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{1}{L2} & \frac{1}{L2} \\ 0 & \frac{1}{C1} & -\frac{1}{RC_1} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} X + \begin{bmatrix} \frac{1}{L_1} \\ 0 \\ 0 \\ 0 \end{bmatrix} E_{DC} \quad (6)$$

Figure 5 shows the hybrid automaton that represents the converter is illustrated in Fig. 4.

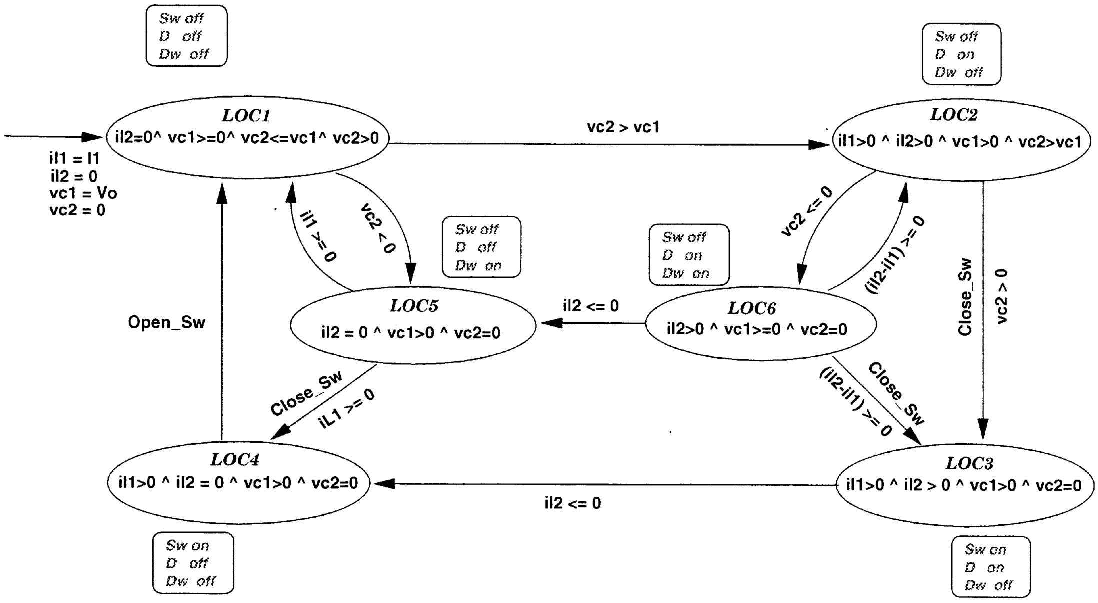

Fig. 5 Hybrid automaton for the ZVS-QR boost converter

B. The Controller

In this application, the control action consists in opening or closing the power switch Sw. The events generated by the converter are synchronized with the controller. There are two operating modes: open and close. The controller automaton generates two events *close_Sw* and *open_Sw* that are sent to converter automaton. The control law employed is a standard proportional integral controller designed to provide the regulation of the output voltage. The output of the controller (u) is compared to a high frequency triangular carrier signal (r) to provide the pulse width modulation of the converter. The automaton representing the controller is shown in Fig. 6.

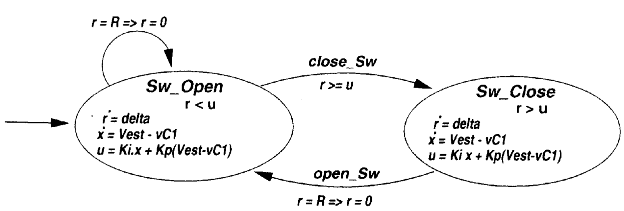

Fig. 6 Hybrid automaton for the converter controller

The set of invariant conditions and the flow equations for the all operation modes are:

Sw_open: Invariant condition: $r < u$

$$\begin{aligned} \dot{r} &= delta \\ \dot{x} &= Vest - v_{C1} \\ u &= K_i x + K_p(Vest - v_{C1}) \end{aligned} \tag{7}$$

Sw_close: Invariant condition: $r > u$

$$\begin{aligned} \dot{r} &= delta \\ \dot{x} &= Vest - v_{C1} \\ u &= K_i x + K_p(Vest - v_{C1}) \end{aligned} \tag{8}$$

The variables $Vest$, K_i and K_p represent the desired output voltage, integral gain and proportional gain, respectively.

The hybrid automata models described above are ready for analysis and simulation. As said before, there are some tools developed specially to study hybrid automata, like SHIFT, CheckMate or Simulink. The program code for implementing the controller automaton by using the SHIFT package shown bellow.

```
type Controller {
    state continuous number r,
                            x,
                            u;
    //**** Variables from Circuit automaton
    input continuous number vC1;

    state number Ki, Kp, R, delta, Vest;
    //**** Discrete states
    discrete
        Sw_open {
                r' = delta;
                x' = Vest - vC1;
                u = Ki * x + Kp * (Vest-vC1);
        } invariant (r < u),
        Sw_close {
                r' = delta;
                x' = Vest - vC1;
                u = Ki * x + Kp * (Vest-vC1);
        } invariant (r > u);
    //**** Synchronization events
    export close_Sw,
           open_Sw;
    //**** Transitions between discrete events
    transition
        Sw_open -> Sw_close {close_Sw}
            when (r >= u),
        Sw_open -> Sw_open {}
            when (r = R)
            do {
                r := 0;
            },
        Sw_close -> Sw_open {open_Sw}
            when (r = R)
            do {
                r := 0;
            };
```

```
    setup
        do {
            Vest := global_Vest;
            R := global_Ref;
            Ki := global_Ki;
            Kp := global_Kp;
            delta := global_delta;
        };
}
```

IV. CONCLUSION

It has been shown how to use the hybrid automata framework to model switching power electronic circuits with feedback control. In the proposed approach the converter and the controller are represented by different hybrid automata and its composition is made by message synchronization. The hybrid automaton is obtained by using a top-down approach that is expressed in terms of a step by step systematic methodology. When constructing the hybrid automaton of a given power switching circuit, the designer must express carefully all the details of the converter behavior in mathematical terms, i.e., invariant, guard and jumping conditions as well as the relevant state variables. This exercise forces the designer to deeper his knowledge about the operation of the converter.

The use of formal tools, like hybrid automata, has not been exploited by the power electronic specialists. Indeed, the proposed methodology is a first step toward the use of formal methods in the field of power electronics. This should allow one to perform formal verification and express the dynamic properties of the power conversion system using temporal logic that eventually simplify the description of rather subtle qualitative temporal relations or interactions between concurrent components that are complicated to verify manually. It is our expectation that filling this gap we are providing a rich field for future investigations.

REFERENCES

[1] Pieter J. Mosterman and Ravi Kapadia Gautam Biswas. Using bond graphs for diagnosis of dynamic physical systems. *Proceeding of DX-95, the Sixth International Workshop on Principles of Diagnosis*, pages 81–85, oct 1995. Goslar, Germany.

[2] Adilson Luiz Bonifácio, Arnaldo Vieira Moura, and João Batista Camargo Jr. Análise, verificação e síntese de segmentos de via de uma malha metroviária. *2nd Workshop on Formal Methods (WMF'99)*, pages 549–572, oct 1999. Florianópolis, SC, Brazil.

[3] Pieter J. Mosterman and Gautam Biswas. A formal hybrid modeling scheme for handling discontinuities in physical system models. *Proceeding of AAAI*, pages 985–990, aug 1996. Portland, Oregon.

[4] R. Alur, C. Courcoubetis, T.A. Henzinger, and P.-H. Ho. Hybrid automata: an algorithmic approach to the specification and verification of hybrid systems. In R.L. Grossman, A. Nerode, A.P. Ravn, and H. Rischel, editors, *Hybrid Systems I*, Lecture Notes in Computer Science 736, pages 209–229. Springer-Verlag, 1993.

[5] T.A. Henzinger. The theory of hybrid automata. In *Proceedings of the 11th Annual Symposium on Logic in Computer Science*, pages 278–292. IEEE Computer Society Press, 1996.

[6] O. Maler, Z. Manna, and A. Pnueli. From timed to hybrid systems. In J. W. de Bakker, K. Huizing, W. P. de Roever, and G. Rozenberg, editors, *Real Time: Theory in Practice*, Lecture Notes in Computer Science 600, pages 447–484. Springer-Verlag, 1992.

[7] X. Nicollin, J. Sifakis, and S. Yovine. From atp to timed graphs and hybrid systems. In J. W. de Bakker, K. Huizing, W. P. de Roever, and G. Rozenber g, editors, *Real Time: Theory in Practice*, Lecture Notes in Computer Science 600, pages 549–572. Springer-Verlag, 1992.

[8] Marc Zimmerman, Mario Rodriguez, and Benjamin Ingram. Making formal methods practical. *IEEE*, pages 1.B.2.1–2.8, 2000.

[9] R. Alur, C. Courcoubetis, N. Halbwachs, T.A. Henzinger, P.-H. Ho, X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine. The algorithmic analysis of hybrid systems. *Theoretical Computer Science*, 138:3–34, 1995.

[10] T.A. Henzinger, P.-H. Ho, and H. Wong-Toi. HyTech: a model checker for hybrid systems. In O. Grumberg, editor, *CAV 97: Computer-aided Verification*, Lecture Notes in Computer Science 1254, pages 460–463. Springer-Verlag, 1997.

[11] T.A. Henzinger, P.-H. Ho, and H. Wong-Toi. A user guide to HyTech. In E. Brinksma, W.R. Cleaveland, K.G. Larsen, T. Margaria, and B. Steffen, editors, *TACAS 95: Tools and Algorithms for the Construction and Analysis of Systems*, Lecture Notes in Computer Science 1019, pages 41–71. Springer-Verlag, 1995.

[12] A. Chutinan and B.H. Krogh. Computing polyhedral approximations to flow pipes for dynamic systems. *In The 37th IEEE Conference on Decision and Control: Session on Synthesis and Verification of Hybrid Control Laws (TM-01)*, 1998.

[13] A. Deshpande, A. Gollu, and L. Semenzato. The shift programming language and run-time system for dynamic networks of hybrid automata. *Technical Report - Department of Electrical Engineering and Computer Sciences*, 1995. University of California - Berkeley - USA.

[14] A. Chutinan and B.H. Krogh. Verification of polyhedral-invariant hybrid automata using polygonal flow pipe approximations. *In Hybrid systems: Computation and Control, Second International Workshop*, 1999.

[15] Fred C. Lee. High-frequency quasiresonant converter topologies. *Proceedings of the IEEE*, 76(4):377–390, oct 1988. USA.