A BFS-DEVS Approach for Induction Generator Non Traditional Modelling

L. Capocchi
University of Corsica
UMR CNRS 6134
Quartier Grossetti, BP 52
20250 Corte - FRANCE
capocchi@univ-corse.fr

D. Federici
University of Corsica
UMR CNRS 6134
Quartier Grossetti, BP 52
20250 Corte - FRANCE
federici@univ-corse.fr

P. A. Bisgambiglia
University of Corsica
UMR CNRS 6134
Quartier Grossetti, BP 52
20250 Corte - FRANCE
bisgambi@univ-corse.fr

H. Henao
University of Picardie
Department of Electrical Engineering
33, rue St Leu
80039 Amiens - FRANCE
Humberto.Henao@ieee.org

G. A. Capolino
University of Picardie
Department of Electrical Engineering
33, rue St Leu
80039 Amiens - FRANCE
Gerard.Capolino@ieee.org

Abstract— **The goal of this paper is to propose a behavioral modelling method of the electrical machine functioning, and more particularly of an induction generator machine. Our approach is based on VHDL-AMS (Very high speed integrated circuits Hardware Description Language for Analog and Mixed Signal) language and on the use of the Discrete EVent System specification: DEVS formalism. We show in this paper how, from the traditional circuit-oriented model, a simplified and realistic model of an induction generator machine can be defined. That model results from a transformation of those circuits described in VHDL-AMS language. It will allow to perform electrical default simulation for the induction machine diagnostics.**

I. Introduction

Nowadays, the models used to represent induction machine are based either on differential equations or on electrical circuits. The mathematical model allows a good representation of it but it involves non-linear effects leading to differential equations, which need a considerable resolution time. The schematical model of electrical circuits is handier, and it allows a simplified representation of the phenomena. It also leads to faster simulation. In the case of a modelling based on electrical circuits, software such as EMTDC and EMTP [1] can be used. Other software like SPICE are used but they are dedicated to the field of electronic simulation. However they are not well adapted to the fault simulation in electrical machines such as induction machines. The oriented-circuit model proposed in [2] is a good approach but it is heavy to manipulate when thousands of fault simulations must be performed for the machine diagnostics.

The aim of our works involves first the modelling of electrical machines with a simple method. That method allows an integration into a schematical model of all the parameters that can characterize the most usual faults of induction machines. The ultimate goal of that modelling is the fault automatic simulation within electrical machines with their diagnostics in mind.

A first step has been taken by [2], [3] with the proposal of a modelling method based on circuit-oriented approach which has shown excellent results. The interest of this solution is the calculation swiftness in contrast to the mathematical solution using differential equations which are often non-linear. However, that method is based on circuit-oriented models which are not well adapted for the quick execution of several experiments and with minimal manipulations and time.

We propose to use BFS-DEVS (Behavioral Fault Simulation for DEVS) [4] formalism in order to model, in a hierarchical and modular way, induction machines. Models are object-oriented and easily manageable when they must evolve or be re-used. Moreover, BFS-DEVS models can simulate automatically the faulty electrical machine behaviors. The circuits allowing the modelling of induction machines are composed of analogical elements showing purely physical phenomena.

The VHDL-AMS [5] language permits to model and simulate mixed-technology micro-systems which consist of electrical circuits connected to subsystems described by differential equations. VHDL-AMS allows the description of components and interconnections like conventional simulator input languages such as SPICE, and also allows expression of component model equations in the frequency domain. DEVS (Discrete EVent system Specification) [6] is a theoretical approach, which allows the definition of hierarchical modular models. In [7] the authors show that the use of DEVS formalism is an interesting solution which facilitates the modelling and simulation of continuous and hybrid systems that are described with VHDL-AMS descriptions.

BFS-DEVS formalism is based on DEVS and so permits to simulate the VHDL-AMS descriptions of circuits modelling the induction machines. It allows a simple model of complexe induction machines that can integrate all physical characteristics such as electromagnetic fields. Moreover, BFS-DEVS permits to specify the model's behavior if they are affected by behavioral faults.

This paper is organized in the following way: in the first part, we present the related works in the field of circuit-oriented modelling for electrical machines; we describe the Discrete Event systems Specification formalism (DEVS); we also present the Behavioral Fault Simulation for DEVS (BFS-DEVS) with its implementation concerning digital circuits testing domain. The second part describes our approach to model the induction machine from both the circuit-oriented model and the use of BFS-DEVS formalism. In the third part, we implement the method described by the example of a complete induction generator machine with its electrical model. Finally, we proposed some conclusions as well as some work prospects.

II. Related Works

A. *Circuit-oriented modelling for electrical machine*

The circuit-oriented model of an induction generator can be expressed with circuit elements such as resistances, inductances, capacitances and voltage or current sources [2], [8]. With this approach, the use of complex electrical circuits allows the time domain investigation without having to implement any transformation technique. Then, physical characteristics of the electrical machine are taken into account. Moreover, the changes in electrical connections or circuit parameters can be easily implemented.

Starting from the concept of magnetically coupled circuits, an induction machine can be represented taking into account the distributed nature of the windings and stator and rotor equivalent circuits can be represented on a conductor-by-conductor basis. Resistive and inductive effects are taken into account and the capacitive effect is neglected to remain in the low frequency range.

Assuming a sinusoidal distribution of windings for both the three-phase stator and rotor circuits, the circuit modeling of this machine is described in figure 1.

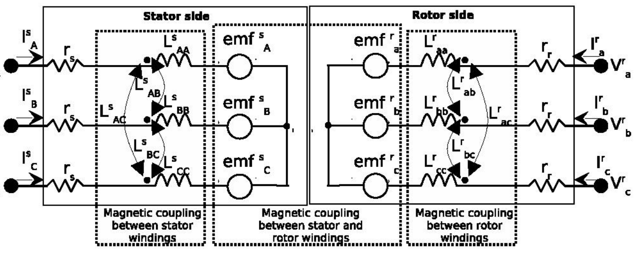

Stator side
Rotor side
Magnetic coupling between stator windings
Magnetic coupling between stator and rotor windings
Magnetic coupling between rotor windings

Fig. 1. Circuit-oriented modelling of an induction generator

In both stator and rotor sides, the conductor resistances (r_s and r_r), the self inductances (L^s_{AA}, L^s_{BB}, L^s_{CC} and L^r_{aa}, L^r_{bb}, L^r_{cc}) and the mutual inductances (L^s_{AB}=L^s_{BA}, L^s_{BC}=L^s_{CB}, L^s_{AC}=L^s_{CA} and L^r_{ab}=L^r_{ba}, L^r_{bc}=L^r_{cb}, L^r_{ac}=L^r_{ca}) are concentrated inside a single block to represent the constant parameters.

The magnetic coupling between stator and rotor coils are subject to the non-linear variation due to the stator-rotor relative position. To model this phenomenon, controlled voltage sources (emf^s_A, emf^s_B, emf^s_C, for the stator side and emf^s_a, emf^s_b, emf^s_c, for the rotor side) can be used. Moreover, (emf^s_A, emf^s_B, emf^s_C) depends on rotor currents (I^r_a, I^r_b, I^r_c) and (emf^r_a, emf^r_b, emf^r_c) depends on stator currents (I^s_A, I^s_B, I^s_C).

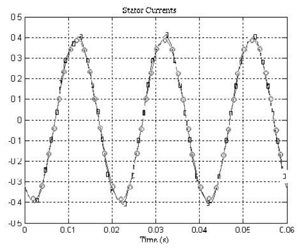
A specific experimental set-up has been designed in order to perform measurements on a lab-scaled three-phase 0.09kW, 50Hz, 220V/380V, 4-poles induction generator. The corresponding parameters are provided in table I.

Rated power	0.09kW
Rated line voltage	380V
Rated frequency	50Hz
Poles	4
Rated torque	0.6Nm
Inertia	0.05kg.m2
Stator resistance r_S	79.13Ω
Stator inductance L_S	2.83H
Stator magnetizing inductance L_{ms}	1.1H
Rotor resistance r_r	3.68Ω
Rotor inductance L_r	0.23H
Rotor magnetizing inductance L_{mr}	0.11H
Mutual inductance L_{sr}	0.68H

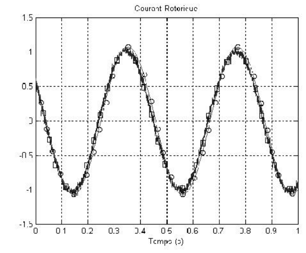
TABLE I
PARAMETERS VALUES FOR EXPERIMENTAL SIMULATION

In figures 2 (a) and (b), the experimental (square) and simulated (circle) stator and rotor currents are presented. On each figure, the currents are shown with a zoom to check the waveform magnitude and period with accuracy.

The simulated results obtained from this circuit-oriented model are very close to those obtained from experimental results. The approach proposed in [8] shows that the circuit-oriented model allows a good representation of an induction generator functioning.



(a) Stator Current



(b) Rotor Current

Fig. 2. Experimental (square) and simulated (circle) stator and rotor currents at 1700 rpm

In the following part, we propose to exploit the oriented-circuit representation and to simplify the that model using DEVS formalism.

B. DEVS formalism

Based on systems theory, DEVS formalism was introduced by Professor B.P. Zeigler in the late 70s [9], [10]. It allows a hierarchical and modular way to model the discrete event systems. A system (or model) is called modular if it possesses the input and output ports permitting interaction with its outside environment. In DEVS, a model is seen as a "black box"S which receives and broadcasts messages on its input and output ports. DEVS defines two kinds of models: atomic models and coupled models, representing respectively the behavior and the internal structure of a part of a model.

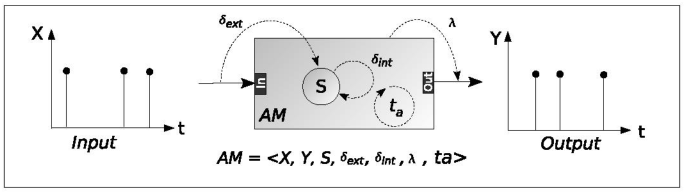


Fig. 3. DEVS atomic model

Figure 3 represents an AM atomic model with its output data Y calculated according to input data X. The AM atomic model has a state variable S that can be reached during the simulation. The functions δ_{ext}, λ, δ_{int} and t_a respectively allow the model's change of state when an external event occurs on one of those outputs (external transition function), the disposal of the output Y (output function), the model's change of state after having given an output (internal transition function) and finally the determination of the duration of the model's state (time advance function).

The coupled models are defined by a set of sub-models (atomic and/or coupled) and express the internal structure of the system's sub-parts thanks to the coupling definition between the sub-models.

Figure 4 shows an example of the hierarchical structure of coupled model CM_0 which has an input port IN and two output ports OUT_0 and OUT_1. It contains the atomic sub-models AM_0, AM_1 and also the coupled model CM_1. The latter can encapsulate other models such as atomic models AM_2, AM_3 and AM_4. A coupled model is specified through the list of its components (AM_0, AM_1,AM_2, AM_3, AM_4 and CM_1 in figure 4), the list of its internal couplings ($AM_0 \rightarrow CM_1$ and $AM_1 \rightarrow CM_1$ in figure 4), the list of the external input couplings ($IN \rightarrow AM_0$ and $IN \rightarrow AM_1$ in figure 4), the list of the external output couplings ($CM_1 \rightarrow OUT_0$ and $CM_1 \rightarrow OUT_1$ in figure 4) and the list of the sub-model's influence ($CM_1 = \{AM_0, AM_1\}$ or CM_1 and influenced by AM_0 and AM_1).

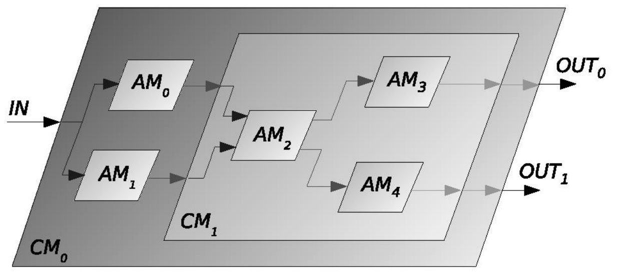


Fig. 4. DEVS coupled model

DEVS formalism is mainly used for the description of discrete event systems. It constitutes a powerful modeling and simulation tool permitting a system modelling on several levels of description as well as the definition of the models' behaviors.

One of DEVS formalism's important properties is that it automatically provides a simulator for each model. DEVS establishes a distinction between a system modeling and a system simulation so as any model can be simulated without the need for a specific simulator to be implemented. Each atomic model is associated with a simulator in charge of managing the component's behavior and each coupled model is associated with a coordinator in charge of the time synchronization of underlying components.

C. BFS-DEVS formalism

BFS-DEVS formalism *(Behavioral Fault Simulation for DEVS)* has the same modelling rules as DEVS but it also provides the possibility to define the faulty behavior of the models. It provides the possibility to construct the library of re-usable BFS-DEVS models that are specific to a field of study. After defining the system's fault model, the model faulty behavior can be specified through a new transition function called "faulty". As it is shown in figure 5, the user has to define a fault model, the control structure, the data structure as well as the behavioral rules of the system in order to construct the BFS-DEVS library.

A behavioral fault or state fault is represented by [11] as a change of the transition function and/or the output function of DEVS model. So a fault has an influence on the state of the model and can lead to a behavior that can be qualified as faulty. Thus we can define a behavioral fault model representing the whole of fault types leading to the unexpected behavior of a model. Faults strongly depend on the nature of the physical system we want to model and they must be the most representative of the real behavioral defaults. Fault model strongly depends on the control structure and the application data. Once the fault model is chosen it will be integrated into the behaviors of each BFS-DEVS model thanks to the faulty transition function δ_{fault}.

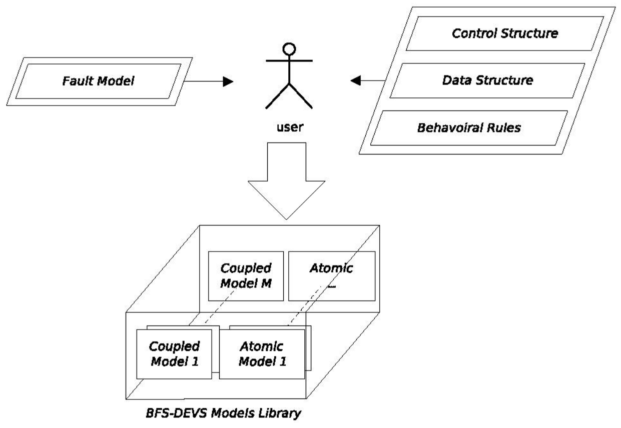


Fig. 5. BFS-DEVS models library

The determination of the control and data structure of an application needs experience. It ensues from the discrete event modelling and it will be integrated into the number, the behavior and the coupling of the BFS-DEVS models that compose the library. So, the construction of a library specific to a field is not an easy matter but it is the only delicate stage of BFS-DEVS modeling. Indeed once that task is performed, the formalism automatically ensures the simulation of the library models.

BFS-DEVS formalism constitutes an ideal interface to perform the behavioral fault simulation of a system belonging to an application field. The BFS-DEVS simulation kernel integrates algorithms which permit simulations of behavioral faults in a concurrent way within the models. This simulation is based on a fault list propagation technique through the BFS-DEVS components network. That technique allows the gathering of faulty simulations in order to obtain a better management of the simulation and it also makes the result's observability at the end of the simulation easier.

In [12], BFS-DEVS formalism has been used to make a behavioral fault simulation on high-level behavioral VHDL descriptions. The chosen approach consists in the transformation of the VHDL behavioral description corresponding to a gate-level circuit into a BFS-DEVS components network [13]. The latter is made thanks to the definition of the BFS-DEVS library composed of models describing all the basic statements of a VHDL language subset. As the representation is textual (circuit VHDL description), both the control and data structures are deduced from VHDL language. The chosen fault model is based on the behavioral rules of the VHDL subset statements. It takes into account the stuck-at of signals and VHDL variables ,the stuck-at of the conditional statement branches as well as the statement jump. The authors validate their approach on ITC'99 benchmarks [14] and they show how the concurrent and comparative algorithms accelerate the fault detection as soon as the beginning of the first simulation steps.

III. BFS-DEVS FOR INDUCTION MACHINE MODELLING

The goal of our modelling approach is to give a simplified model of induction machines from BFS-DEVS formalism in order to simulate the most usual defaults on those machines. The main idea, shown in figure 6, consists in the transformation of the circuit-oriented models proposed in [13] into VHDL-AMS descriptions and then in the translation of those descriptions into BFS-DEVS components networks.

According to [15] VHDL-AMS " is an IEEE standard and extension of a high-level hardware description digital language VHDL. VHDL-AMS is widely used in electronic design flow for modeling various mixed-signal (analog and digital) circuits and systems. It can also be used to model mixed-technology systems if their description is limited to differential algebraic equations". The analog circuits representing the induction machines make models based on differential equations intervene (mutual inductance for instance). The VHDL-AMS language permits a homogeneous textual representation of those analog circuits.

In [7], the authors show how DEVS formalism makes the simulation of the described models easier with a simplified version of the sAMS-VHDL language. This simulation is made after a transformation of models written in VHDL-AMS into equivalent DEVS models. The nature of DEVS formalism allows the integration of those models in component form.

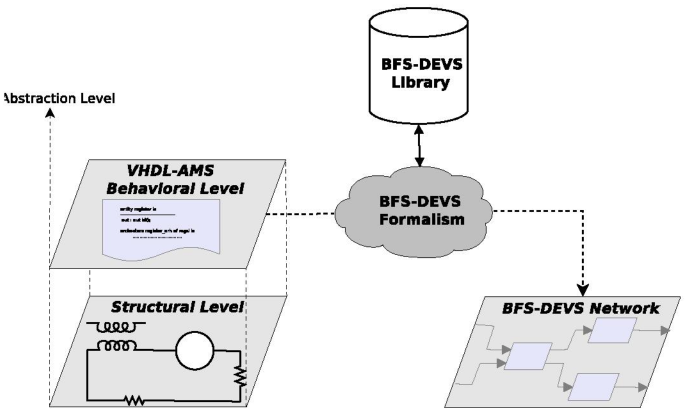


Fig. 6. BFS-DEVS for VHDL circuit modelling

BFS-DEVS formalism permits the transformation of VHDL-AMS textual model into BFS-DEVS model because it benefits from DEVS properties. Moreover, BFS-DEVS models belonging to the network integrate the system's faulty behaviors. The advantages of the construction of those networks are:

- a modular and hierarchical representation of electrical machines behavior,
- an easy way to define behavioral faults,
- an automatic simulation of behavioral faults.

Concerning induction machines, circuit-oriented modelling methods separate the rotor and stator models [16]. The stator model results from the representation of each coil turn. All the electrical parameters such as resistances and mutual inductances are obtained by applying the magnetic laws of the circuits. The model of the rotor cage is based on the same principle by identifying each electrical element according to each mesh. It is the connection of different meshes which gives the global representation of the rotor cage.

The two types of circuit representing the stator and rotor cage of the induction machine are transcribed into VHDL-AMS language by translating each equation of each electrical element. The voltage relationships of active elements such as mutual inductances or the resulting electromotive forces are implemented thanks to "Simultaneous" statements because they allow the description of algebraic differential equations [5]. Then their resolution calls on Euler or Runge-Kutta methods[7]. The passive elements such as resistances are represented by using the VHDL language basic sequential statements.

It is from VHDL-AMS language textual descriptions that we propose a construction of basic BFS-DEVS models. Each statement of the VHDL-AMS code in the circuit architecture is converted into an atomic or coupled model. The rules used in [13] for the transformation of behavioral VHDL description can be implemented. However, an atomic model has to be added in the case of the "Simultaneous" statement whose behavior is the digital integration of differential equations.

Now we are going to show how to construct the BFS-DEVS library specific to the field of induction machine in the case of a circuit which models a stator winding.

IV. INDUCTION GENERATOR MODELLING EXAMPLE

The three-phase electrical circuit modelling induction generator is shown in figure 7 coming from [8]. The representation of that circuit, simplified in VHDL-AMS language, goes through the behavioral definition of each passive and active element composing it.

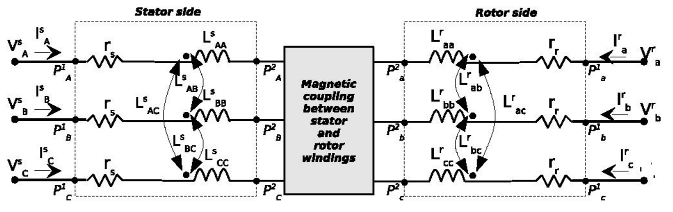


Fig. 7. Simplified circuit modeling of an induction generator

```
—————————————Testbench—————————————
LIBRARY IEEE; USE IEEE.electrical_systems.all; work.all;

ENTITY testbench IS
 PORT(TERMINAL $P^1_A$, $P^1_B$, $P^1_C$: ELECTRICAL;
      TERMINAL $P^1_a$, $P^1_b$, $P^1_c$: ELECTRICAL);
END testbench;

ARCHITECTURE arch OF testbench IS
BEGIN

 ENTITY stator(arch_stator)
  GENERIC map ($r_s \Rightarrow 79.13$, $L^S_{AA} \Rightarrow 2.83$, $L^S_{AB} \Rightarrow 1.1$,
           $L^S_{BB} \Rightarrow 2.83$, $L^S_{AC} \Rightarrow 1.1$, $L^S_{CC} \Rightarrow 2.83$,
           $L^S_{BC} \Rightarrow 1.1$)
  PORT map ($v_1 \Rightarrow P^1_A$, $v_2 \Rightarrow P^1_B$, $v_3 \Rightarrow P^1_C$, $v_4 \Rightarrow P^2_A$,
        $v_5 \Rightarrow P^2_B$, $v_6 \Rightarrow P^2_C$);

 ENTITY rotor(arch_rotor)
  GENERIC map ($r_r \Rightarrow 3.68$, $L^r_{aa} \Rightarrow 0.23$, $L^r_{ab} \Rightarrow 0.68$,
           $L^r_{bb} \Rightarrow 0.23$, $L^r_{ac} \Rightarrow 0.68$ $L^r_{cc} \Rightarrow 0.23$,
           $L^r_{bc} \Rightarrow 0.68$)
  PORT map ($v_1 \Rightarrow P^1_a$, $v_2 \Rightarrow P^1_b$, $v_3 \Rightarrow P^1_c$, $v_4 \Rightarrow P^2_a$,
        $v_5 \Rightarrow P^2_b$, $v_6 \Rightarrow P^2_c$);

 ENTITY magnetic_coupling(arch_magnetic_coupling)
  PORT map ($v_1 \Rightarrow P^2_A$, $v_2 \Rightarrow P^2_B$, $v_3 \Rightarrow P^2_C$, $v_4 \Rightarrow P^2_a$,
        $v_5 \Rightarrow P^2_b$, $v_6 \Rightarrow P^2_c$);
END arch;
```

Fig. 8. Example of VHDL-AMS testbench for stator model

The VHDL-AMS testbench allowing the simulation of the circuit is given in figure 8. It is composed of a global entity made of the following electrical ports $P^1_A = V^s_A$, $P^1_B = V^s_B$, $P^1_C = V^s_C$, $P^1_a = V^r_a$, $P^1_b = V^r_b$ and $P^1_c = V^r_c$. The "arch" architecture is composed of three main entities of the highest level: an entity corresponding to the stator model, an entity corresponding to the rotor model and an entity corresponding to the magnetic coupling between the stator and the rotor. Each entity allows the definition of the element's constants (resistance, inductance, ...) and the connection between the ports P. The "stator" entity has the "arch_stator" architecture, the line called GENERIC permits the definition and assignment of the resistance r_s, the inductances L_s and the mutual inductances L_{ms} thanks to the grid I.

The line called PORT permits the connection of the ports $v_{1\leq i\leq 6}$ to the ports of the element P^1_A,P^1_B,P^1_C,P^2_A,P^2_B, and P^2_C. Generally each element of the circuit, either passive or active, is represented by its entity which defines its characteristics (resistance, inductance, ...) and its interface (ports). Of course the modelling can be more explicit by considering each basic element (passive or active) as the full entity. We propose a model described at a high level of abstraction in order to present a handy and simple representation.

Each element of the circuit in figure 8 has an interface and a behavior that can be described thanks to an entity and a specific architecture. If we take the example of the stator, its VHDL-AMS description is given in figure 9. In the case of the description of an active element composed of differential equations we use the "Simultaneous" statement [5].

```
——————————————stator——————————————
LIBRARY IEEE; USE IEEE.electrical_systems.all; work.all;

ENTITY stator IS
 GENERIC ($r_s$, $L^S_{AA}$, $L^S_{AB}$, $L^S_{BB}$, $L^S_{AC}$, $L^S_{CC}$, $L^S_{BC}$: real);
 PORT(TERMINAL $v_1$,$v_2$,$v_3$,$v_4$,$v_5$,$v_6$: ELECTRICAL);
END stator;

ARCHITECTURE arch_stator OF stator IS
 QUANTITY $U^s_A$ ACROSS $I^s_A$ THROUGH $v_1$ TO $v_4$;
 QUANTITY $U^s_B$ ACROSS $I^s_B$ THROUGH $v_2$ TO $v_5$;
 QUANTITY $U^s_C$ ACROSS $I^s_C$ THROUGH $v_3$ TO $v_6$;
BEGIN
 $U^s_A$ == f($r_s$,$I^s_A$,$L^s_{AA}$,$L^s_{AB}$,$L^s_{AC}$)
 $U^s_B$ == f($r_s$,$I^s_B$,$L^s_{BB}$,$L^s_{BA}$,$L^s_{BC}$)
 $U^s_C$ == f($r_s$,$I^s_C$,$L^s_{CC}$,$L^s_{CB}$,$L^s_{CA}$)
END arch_stator;
```

Fig. 9. Example of VHDL-AMS testbench for stator model

The BFS-DEVS network of a generator machine includes the behavior of each circuit element presented in figure 7. The control and data structures of the modeled system are those of the VHDL-AMS code. The fault model can be based on the data values (resistance, inductance, mutual inductance, electromagnetic field, ...) in order to consider the stuck-at faults. The components library for the modelling of a generator machine can be constructed by considering a coupled model for each high-level element (stator, rotor and the stator/rotor coupling model). Each of those coupled models contains an interconnection of atomic models corresponding to the control graph of the statements in the element's VHDL-AMS architecture. The fault model is contained in the faulty transition function in each of those atomic models. So, the network can be directly simulated thanks to DEVS formalism properties.

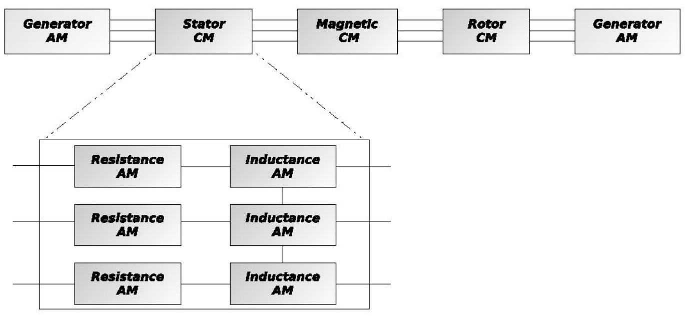


Fig. 10. BFS-DEVS network for generator induction circuit model

Figure 10 represents the BFS-DEVS network of a three-phase generator induction circuit. Each entity of the highest level such as the stator, the rotor and the magnetic coupling between the stator and the rotor is represented by a coupled model. Coupled models contain an interconnection of atomic models corresponding to the VHDL-AMS statements which describe the entity's behavior. Figure 10 shows that the stator coupled model is composed of three atomic models "Inductance " which are interconnected. This interconnection allows the simulation of the mutual inductance effects within the stator. Moreover, the atomic models "Generator" are present in order to simulate the system's behavior.

V. Conclusion and Future Works

We have proposed a method for high-level behavioral modelling of the circuit-oriented induction generator. That is why we used the VHDL-AMS language to obtain a homogeneous description of electrical machines such as induction generators that can integrate representations based on differential equations. In order to take advantage from the re-use notion, we have transformed these textual descriptions into BFS-DEVS components network. DEVS formalism permitting a hierarchical and modular way to specify the behavior of systems, we show more particularly how, from circuit-oriented models, a realistic and simplified model of the induction generator stator can be defined.

The future works are mainly based on the possibility of obtaining a concurrent and automatic modelling and simulation environment. This environment will allow a simulation of the most usual defaults that can intervene on the induction machines. Indeed, the BFS-DEVS simulation kernel is based on concurrent and comparative algorithms which use the fault list propagation technique. After having entirely modeled the induction machine stator and rotor and having chosen a realistic fault model we plan to perform the concurrent fault simulation on the whole model.

References

[1] F. Soares and P. C. Branco, "Simulation of a 6/4 switched reluctance motor based on matlab/simulink environment," in *IEEE Transactions on Aerospace and Electronic Systems*, vol. 37, pp. 989–1009, 2001.

[2] H. Henao, C. Martis, and G. Capolino, "An equivalent internal circuit of the induction machine for advanced spectral analysis," in *Proceedings of IAS Annual Meeting (IAS'02)*, vol. 2, pp. 739–745, october 2002. Pittsburgh (USA).

[3] H.Hénao, G.A.Capolino, and M.Poloujadoff, "A circuit-oriented model of induction machine for diagnostics," in *Proceedings of International Symposium on Diagnostics for Electrical Machines, Power Electronics & Drives (SDEMPED'97)*, pp. 185–190, 1997. Carry-le-Rouet (France).

[4] L. Capocchi, *Simulation de Fautes Comportementales pour des Systèmes à Evénements Discrets: Application aux Circuits Digitaux*. PhD thesis, november 2005. University of Corsica.

[5] E. Moser and N. Mittwollen, "VHDL-AMS: The missing link in system design - experiments with unified modelling in automative angineering," in *Proceedings of the Conference on Design, Automation and Test in Europe (DATE'98)*, pp. 59–65, 1998. Le Palais des Congrès de Paris, France.

[6] B. P. Zeigler and S. Vahie, "DEVS formalism and methodology - unity of conception diversity of application," in *Proceedings of 1993 Winter Simulation Conference* (S. Editions, ed.), pp. 573–579, 1993.

[7] S. Mehta and G. Wainer, "Devs for mixed-signal modeling based on VHDL," in *Proceedings of 2005 DEVS Integrative M&S Symposium, Spring Simulation Conference*, 2005. San Diego, CA. U.S.A.

[8] A. Yazidi, H. Henao, G. Capolino, D. Casadei, and F. Filippetti, "Double-fed three-phase induction machine abc model for simulation and control purposes," in *Proceedings of IEEE Industrial Electronics Conference (IECON'05)*, vol. 4, pp. 2560–2565, November 2005.

[9] B. P. Zeigler, *Theory of Modeling and Simulation*. Academic Press, 1976.

[10] B. P. Zeigler, "An introduction to set theory," tech. rep., 2003. ACIMS Laboratory, University of Arizona.

[11] E. Kofman, N. Giambiasi, and S. Junco, "FDEVS: A general DEVS-based formalism for fault modeling and simulation," in *Proceedings of European Simulation Symposium*, pp. 77–82, 2000. Hamburg, Germany.

[12] L. Capocchi, D. Federici, F. Bernardi, and P. Bisgambiglia, "Behavioral fault simulation for VHDL descriptions using the DEVS formalism," in *IEEE Pacific Rim Dependable Computing International Conference*, 2004.

[13] L. Capocchi, F. Bernardi, D. Federici, and P. Bisgambiglia, "Transformation of VHDL descriptions into DEVS models for fault modeling and simulation," in *Proceedings of IEEE Systems, Man and Cybernetics Conference (SMC'03)*, pp. 1205–1211, 2003. Washington D.C., USA.

[14] F. Corno, M. S. Reorda, and G. Squillero, "RT-level itc'99 benchmarks and first ATPG results," in *Proceedings of IEEE Design and Test of Computers*, pp. 44–53, 2000.

[15] P. Nikitin and C.-J. R. Shi, *VHDL-AMS Based Modeling and Simulation of Mixed-Technology Microsystems: A Tutorial.* Integration, the VLSI Journal, 2006.

[16] H.Hénao, G. Capolino, M. Melero, and M. Cabanas, "A new model of the induction motor rotor cage for diagnostics," in *Proceedings of International Symposium on Diagnostics for Electrical Machines, Power Electronics & Drives (SDEMPED'99)*, pp. 383–388, november 1999. Gijon (Spain).