



HAL
open science

Simulation of AC Electrical Machines Behaviour Using Discrete Event System Simulator

Laurent Capocchi, Dominique Federici, Humberto Henao, Gerard-André
Capolino

► **To cite this version:**

Laurent Capocchi, Dominique Federici, Humberto Henao, Gerard-André Capolino. Simulation of AC Electrical Machines Behaviour Using Discrete Event System Simulator. 2007 IEEE International Symposium on Industrial Electronics, Jun 2007, Vigo, Spain. pp.945-970. hal-00165433

HAL Id: hal-00165433

<https://hal.science/hal-00165433>

Submitted on 26 Jul 2007

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.

Simulation of AC Electrical Machines Behaviour Using Discrete Event System Simulator

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

H. Henao

University of Picardie
Dpt. of Electrical Eng.
33, rue St Leu
80039 Amiens- FRANCE
Humberto.Henao@ieee.org

G. A. Capolino

University of Picardie
Dpt. of Electrical Eng.
33, rue St Leu
80039 Amiens - FRANCE
Gerard.Capolino@ieee.orgs

Abstract—This paper deals with the adaptation of AC electrical machine models for discrete event system simulator. The formalism chosen is DEVS (Discrete Event system Specification) which has been adapted recently for hybrid system simulation. The software PowerDEVS is close to MATLAB/Simulink(C) but without any toolbox adapted to power systems. As always, the model which is basically based on system of non-linear differential equations has to be integrated to compute the output variables corresponding to any deterministic or stochastic inputs. In this way, the integrator atomic model has been optimized to perform many simulations with realistic CPU time. Results obtained with PowerDEVS and optimized graphical model have been validated by the same simulations using MATLAB/Simulink(C) as found in the classical literature.

I. INTRODUCTION

The simulation programs for electrical power systems have been used since the beginning of the 60s. There are more than hundred of them and this is event difficult to have an exhaustive list and/or a classification. However, some of them are well known and have been described in the literature such as EMTP, ECAP and NETOMAC which have been initiated in the late 60s. Some more recent package such as SPICE (its adaptation to power systems) or MATLAB (its power systems simulator) have been successfully used for many applications. Others such as Dymola, Omsim, EUROSTAG can be classified in software used for power systems simulation [1]. Many of them are adapted to simulation a model defined by sets of ordinary differential equations (ODE) which are related generally to continuous systems under textual or block-diagram representations. During the last twenty years, many efforts have been made in order to give the user modular modelling environments with easy user-based interfaces for transparent and automatic simulation. Nowadays, all the available programs propose graphical interfaces to simplify the modelling process of any physical systems. However, these interfaces are more and more complex and can only be fully exploited by experts of the studied domain. Moreover, the simulation algorithms proposed within the programs are mainly based on numerical integration methods such as Runge-Kutta, Euler, Adams and so on. These methods are specific in the sense that they are all based on a classical discretization resulting in a

discrete time simulation model that could require much more execution time.

So far, electrical machine digital simulation has been under focus for the last forty years and many of the previous software have been tested for this purpose. The result of this extended research is that many of these packages have their own power systems library inside. However, the approach used is always the same since the electrical machine has to be considered as a continuous system which has to be discretized to be digitally simulated in any of these packages. Two different ways are available, the first one being related to direct discretization of non-linear differential equations and the second being indirect by the interpretation of internal model with a circuit-oriented approach. For any simple three-phase AC machine in symmetrical conditions, the problem is very simple to be solved and the two methods are almost equivalent in term of complexity. However, when the windings are not symmetrical and/or if any other fault did occur in the electromechanical interface the circuit-oriented approach has been proved to be more flexible in term of physical interpretation. Since a fault can be considered as a discrete event, it is necessary to define a suitable model corresponding with its nature for any complex system having one (or several) electrical machine inside such as wind turbine or micro-hydro generators. For this purpose, a new formulation of AC electrical machine models is mandatory to export them in the main discrete event simulators developed for different applications.

The aim of this paper is to present a new modelling and simulation approach of a complex power systems using a DEVS-based (Discrete Event system Specification) graphical environment. The DEVS formalism [2] has been defined thirty years ago to allow the specification of discrete event systems. It provides a way to define complex models in a hierarchical and modular way. This environment allows the implementation of numerical simulation algorithms based on a discretization which is not linked to time anymore but to space on state variables if the system is continuous. Many years ago, DEVS was upgraded in order to permit the modelling of continuous systems and two main formalisms were developed. In GDEVS (Generalized Discrete Event Specification) [3], the trajectories are organized through piecewise polynomial segments. In

the literature, the key contribution of GDEVS is considered as its ability to develop uniform discrete event executable specifications for hybrid dynamic systems. Moreover, discrete event systems including DEVS and GDEVS are simulated at high speed on a host computer because of significant changes in the system. By speaking again of ODE efficient numerical simulation, another DEVS-based method called quantized systems was proposed by Zeigler [4]. In this approach, the time discretization is replaced by the state quantization and a DEVS simulation model is obtained instead of a discrete time one. This idea was reformulated and formalized by Kofman [5] as a simulation method for general ODE in where the quantized state system (QSS) method was defined. In [6], the author shows that from the computational cost point of view, the QSS method can reduce the number of iterations. Many examples like block-oriented DEVS simulation of a RLC circuit have been recently presented [6] but the configuration of the QSS integrator model is not explicit. Another example of the circuit simulation based on bond-graph translation was also presented [7].

In this paper, an extension of the use of the QSS method for the simulation of differential algebraic equations (DAE) which describe any electrical power system. According to [6], the proposed approach is based on block-by-block translation from a block-diagram resulting from the DAE systems. Each step of the modelling process is described and every approach is discussed. In order to achieve the proposed modelling approach, the package PowerDEVS [8] is used and it is adapted to a more complex system than originally designed for. PowerDEVS is a DEVS-based graphical environment for hybrid system modelling and simulation. It proposes several QSS methods which provide a quantization of the state variables to obtain a discrete event approximation of the continuous system.

The paper is organized in four sections. The section 2 presents the background and shows the DEVS formalism with the DEVS-based modelling and simulation software PowerDEVS. This section briefly describes the QSS methods proposed in PowerDEVS. In section 3, the use of PowerDEVS environment for electrical power systems modelling and simulation is proposed. Therefore, the case of a simple electrical circuit with its differential equations is presented. Then, the PowerDEVS simulation is given and the results are analyzed. Finally, the section 4 gives some conclusions and directives for a future direction of research in the usage of discrete event simulation for power systems fault detection.

II. RELATED WORKS

A. DEVS formalism

The DEVS formalism was introduced by Zeigler in the late 70s [9], [10]. It is based on systems theory, allowing a hierarchical and modular way to model the discrete event systems. A system or a model is called modular, if its input and output ports interact with its environment. In DEVS, a model is seen as a “black box” receiving and sending messages on its input and output ports respectively. Two types of models are defined, atomic models and coupled models, representing respectively the behaviour and the internal structure of a model element.

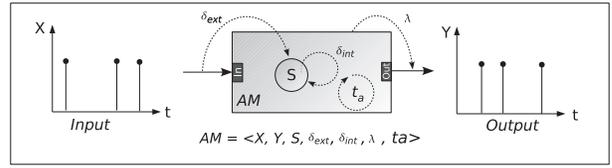


Fig. 1. DEVS atomic model

Figure 1 represents an AM atomic model with its output data Y calculated according to input data X . The atomic model has a state variable S that can be observed during the simulation. When an external event occurs, the model state is modified depending of the following functions:

- δ_{ext} : external transition function
- λ : output function
- δ_{int} : internal transition function
- t_a : 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 sub-elements thanks to the coupling definition between the sub-models.

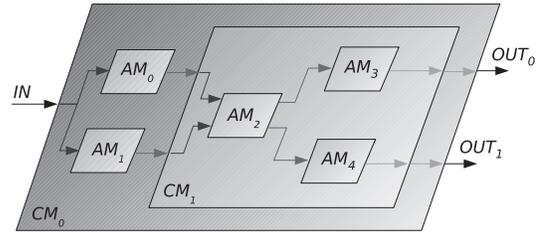


Fig. 2. DEVS coupled model

Figure 2 shows an example of a coupled model hierarchical structure. The coupled model CM_0 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 . This latter is constituted by the atomic models AM_2 , AM_3 and AM_4 . A coupled model is specified by the list of its components (AM_0 , AM_1 , AM_2 , AM_3 , AM_4 and CM_1), the list of its internal couplings ($AM_0 \rightarrow CM_1$ and $AM_1 \rightarrow CM_1$), the list of the external input couplings ($IN \rightarrow AM_0$ and $IN \rightarrow AM_1$), the list of the external output couplings ($CM_1 \rightarrow OUT_0$ and $CM_1 \rightarrow OUT_1$) and the list of the sub-model influence ($CM_1 = \{AM_0, AM_1\}$ or CM_1 and influenced by AM_0 and AM_1).

The DEVS formalism is mainly used for the description of discrete event systems. It constitutes a powerful modelling and simulation tool permitting a system modelling with several levels of description as well as the definition of the model behaviors. It provides a simulator for each model automatically and establishes a distinction between a system modelling and a system simulation. Each atomic model is associated with a simulator which manage the component behavior. Each coupled model is associated with a coordinator which synchronizes temporally its components.

B. PowerDEVS Environment

Some of DEVS simulation tools as AToM3 [11], Diam-Sim [12] and PowerDEVS [8] include graphical interface

and advanced simulation features. These tools were developed before the creation of discrete event methods for numerical integration of ODE, excepting PowerDEVS. This last software is a DEVS-based integrated tool for modelling and simulation of continuous systems. It is composed by four separate programs [8]:

- The model and atomic editor
- The structure generator and the pre-processor

PowerDEVS allows building or managing atomic DEVS models or libraries. These models can be graphically coupled in a hierarchical way to create more complex systems. Atomic and coupled models can compose libraries, facilitating the application of all the defined components. For the power system simulation, the PowerDEVS “Continuous” library is a well adapted tool to represent the complexity of electromagnetic devices. As it is shown in Figure 3, this library composed by most of the components that are used to build the block-schemes corresponding to the ODE of continuous systems.

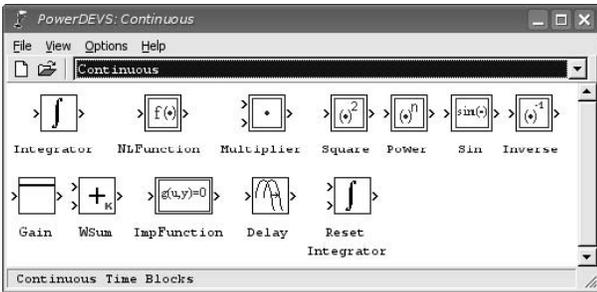


Fig. 3. PowerDEVS “Continuous” Library.

This “Continuous” library proposes an integrator atomic model to perform the ODE integration with optimized state quantization methods [6].

C. State Quantization-Based Methods

The basis of the state quantization-based methods comes from the idea that, it is possible to obtain an ODE numerical simulation by a discrete representation of state variables, instead of a discrete representation in time. This idea introduced initially by Zeigler was reformulated defining in a formal way a quantized state system (QSS) for a first order approximation, to formalize a new numerical integration approach [5]. A QSS is based on a hysteresis quantization method where:

- The quantized variables have piecewise trajectories
- The state variable have also piecewise trajectories
- The state variables have continuous piecewise trajectories

Figure 4 shows a typical quantization function $q(t)$ with uniform quantization intervals, obtained with a hysteresis window ϵ . With this quantization method, the stability properties of the original continuous system are conserved. When the window ϵ is chosen to be equal to the quantum q , there is a reduction in the number of steps performed by the hysteresis algorithm and a consequent reduction of the computational costs. With this method the obtained accuracy depends on the quantum q , which is related to the number of hysteresis cycles.

Then a good accuracy cannot be obtained without increasing significantly the computational cost.

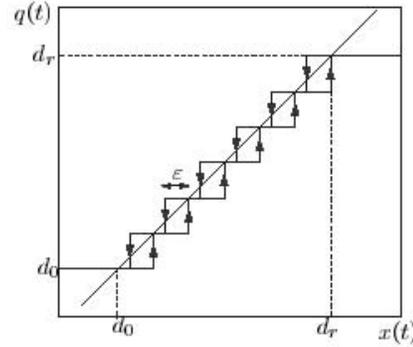


Fig. 4. Quantized function with hysteresis.

A quantized state system for a second order approximation (QSS2) [13], [14] has been proposed in order to obtain less number of hysteresis cycles with respect to the QSS method for the same quantum q . The QSS2 method uses first order quantizers, conducting to piecewise linear trajectories in the quantized variables. The development of a third order approximation QSS3 is introduced in [15]. With this new approach, the accuracy can be improved and the quantization choice is not critical as with QSS2. The obtained results show that this last method is an efficient algorithm for accurate numerical integration of discontinuous systems.

III. DEVS SIMULATION OF POWER SYSTEMS

Electrical power systems models are essentially build around resistances (R), inductances (L), capacitors (C) and voltage sources (v) and current sources (i). The equations relating voltages and currents in the passive elements characterized by R, L and C, describe the physical behaviour of electrical devices with respect to the energy converted into heat (R), the energy stored in both inductive (L) and capacitive (C). In this way, the complete model will be translated into a continuous system described by a set of DAE.

A. Induction Machine Stator Side

The proposed circuit is a symmetrical three-phase magnetically linked inductive circuit which is the simple low frequency representation of any AC machine stator windings. In order to obtain a linear circuit-oriented model of this device, the passive elements are used taking into account the resistive and inductive effects (Fig. 5). The capacitive effects are neglected in order to remain within the low frequency analysis range.

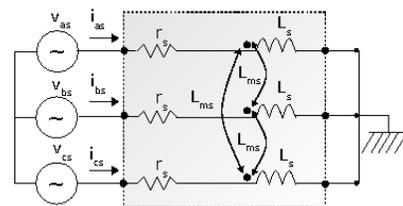


Fig. 5. Symmetrical three-phase circuit

The elements of the equivalent circuit are given by:

- r_s : phase resistance,
- L_s : phase self-inductance,
- L_{ms} : mutual inductances between phases.

The power supply is given by the three-phase symmetrical voltages $v_{as}(t)$, $v_{bs}(t)$ and $v_{cs}(t)$ which are modelled as sinusoidal time functions:

$$\begin{cases} v_{as}(t) = V_m \sin(2\pi ft) \\ v_{bs}(t) = V_m \sin(2\pi ft - \frac{2\pi}{3}) \\ v_{cs}(t) = V_m \sin(2\pi ft - \frac{4\pi}{3}) \end{cases} \quad (1)$$

with $V_m = \frac{U_m \sqrt{2}}{\sqrt{3}}$, U_m being the line-to-line voltage and f the supply frequency.

Applying the electric circuit laws to the circuit of Fig 5, it a set of equations can be obtained:

$$\begin{cases} v_{as} = r_s \cdot i_{as} + L_s \frac{d}{dt} i_{as} - \frac{L_{ms}}{2} \left[\frac{d}{dt} i_{bs} + \frac{d}{dt} i_{cs} \right] \\ v_{bs} = r_s \cdot i_{bs} + L_s \frac{d}{dt} i_{bs} - \frac{L_{ms}}{2} \left[\frac{d}{dt} i_{as} + \frac{d}{dt} i_{cs} \right] \\ v_{cs} = r_s \cdot i_{cs} + L_s \frac{d}{dt} i_{cs} - \frac{L_{ms}}{2} \left[\frac{d}{dt} i_{as} + \frac{d}{dt} i_{bs} \right] \end{cases} \quad (2)$$

The set of formulas 2 shows a DAE system where the continuous variables i_{as} , i_{bs} and i_{cs} represent the state variables (phase currents) to be computed. Moreover, this system has to be translated in the PowerDEVS environment.

B. PowerDEVS Simulation

The proposed circuit can be described by block-diagrams with algebraic loops. In this way, the DAE system must be expressed as a function of state variables and their derivatives:

$$\begin{cases} \frac{d}{dt} i_{as} = \frac{1}{L_s} \left[v_{as} - r_s i_{as} + \frac{L_{ms}}{2} \left[\frac{d}{dt} i_{bs} + \frac{d}{dt} i_{cs} \right] \right] \\ \frac{d}{dt} i_{bs} = \frac{1}{L_s} \left[v_{bs} - r_s i_{bs} + \frac{L_{ms}}{2} \left[\frac{d}{dt} i_{as} + \frac{d}{dt} i_{cs} \right] \right] \\ \frac{d}{dt} i_{cs} = \frac{1}{L_s} \left[v_{cs} - r_s i_{cs} + \frac{L_{ms}}{2} \left[\frac{d}{dt} i_{as} + \frac{d}{dt} i_{bs} \right] \right] \end{cases} \quad (3)$$

For each state variable, a block-diagram has to be described coupling a voltage source with an adder and an integrator (Fig. 6). Then, two types of algebraic loops are presents for the state variable computation. The first one is related to the initial state variable and the second one shows the mutual effect between all the state variables.

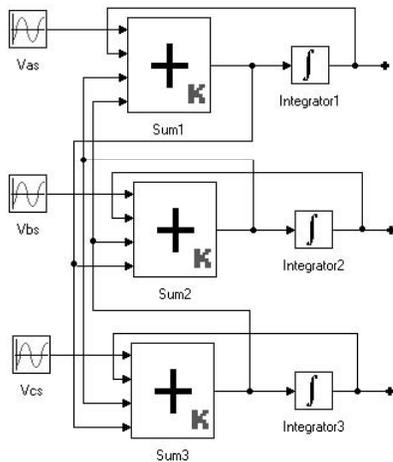


Fig. 6. Stator Block Diagram

The PowerDEVS Continuous library has that element and its implementation requires the determination of weighting coefficients $K_{i,i \in \{0,1,2,3\}}$ of each term of the sum. Since a system of three equations is provide with one equation per phase x ($x \in \{a,b,c\}$), three adders $Sum_{i,i \in \{1,2,3\}}$ with four inputs will be necessary with:

- the input voltage v_{xs} with a weighting coeff. of $K_0 = \frac{1}{L_s}$,
- the stator current i_{xs} with a weighting coeff. of $K_1 = -\frac{r_s}{L_s}$,
- the two derivatives of the two stator currents with the weighting coeff. of $K_2 = K_3 = \frac{L_{ms}}{2 \cdot L_s}$.

It is necessary to use a coupled model called “stator” which includes three adders. As it is shown (Fig. 8), the three-phase stator model has three inputs and three outputs. The three inputs allow the acquisition of the sinusoidal input voltages v_{xs} . The three outputs allow the computation of the three stator currents. The output of each adder provides the stator current derivatives and they are used to compute the other stator currents being injected in the last two inputs of the neighbour adders. However, these derivatives introduce a numeric noise and this can be drastically reduced by implementing one low-pass filter $LP_{i,i \in \{0,1,2\}}$ in each connexion. Each filter is represented by one black-box in each coupled model (Fig. 8).

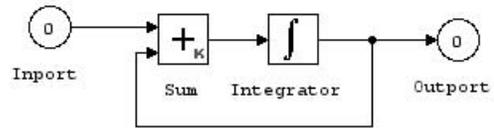


Fig. 7. Low-Pass Coupled Model

A simple block-scheme representation of each low-pass filter is very classical and can be easily provided (Fig. 7). It is composed by one adder and one integrator interconnected within a closed-loop system. The characteristics of the low-pass filters depend on the stator winding time constant. Indeed, if the equivalent time constant is $\tau = \frac{L}{R}$, the reverse of the adder weighting coefficient must be very low in front of τ (10 times less at least).

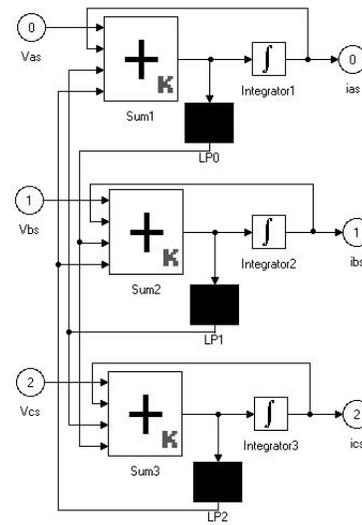


Fig. 8. Stator Coupled Model with Low-pass Filters

The complete system will be simulated considering a set of sinusoidal input voltages V_{xs} . The simulation of the complete three-phase system will be tested using different integration methods with the quantized state systems (QSS, QSS2, QSS3) and the quantum dq within the Integrator atomic models 1, 2 and 3 (Fig. 8).

C. Simulation Results and Performance Analysis

The numerical parameters used to simulate the set of equations (3) are summarized in Table I. These values allow to compute the weighting coefficients $K_{i,i \in \{0,1,2,3\}}$ of each adder which are part of the induction machine stator model. The weighting coefficients $K'_{i,i \in \{0,1\}}$ of the adders included in the low-pass filters are computed considering the fact that $\tau = \frac{L}{R} \simeq 0.03s$. As it has been already said previously, the reverse coefficients must be much lower than to 0.03s. In this way, the final choice has been $K'_0 = -K'_1 = 10000$. Moreover, each integrator within the filters uses the QSS2 method with the quantum $dq = 0.001$ and without any initial condition.

Line-to-line voltage (U_m)	380V
Supply frequency (f)	50Hz
Stator resistance (r_s)	79.13 Ω
Stator inductance (L_s)	2.83H
Stator magnetic inductance (L_{ms})	2.2H
$K_0 = \frac{1}{L_s}$	0.353357
$K_1 = \frac{r_s}{L_s}$	-27.96110
$K_2 = \frac{L_{ms}}{2L_s}$	0.388690
$K_3 = K_2$	0.388690
$K'_0 = -K'_1$	10000

TABLE I
VALUES OF ALL PARAMETERS

As it is shown in the complete block-scheme (Fig. 9), the output stator currents are observed by connecting *QuickScope* _{$i,i \in \{1,2,3\}$} atomic models from the “Sinks” library to the three outputs $i_{as}(t)$, $i_{bs}(t)$ and $i_{cs}(t)$ of the stator coupled model. The complete system was simulated with four different values of the quantum dq which influences at the level of the stator integrators. The effect of the quantization methods QSS, QSS2 and QSS3 has been also examined.

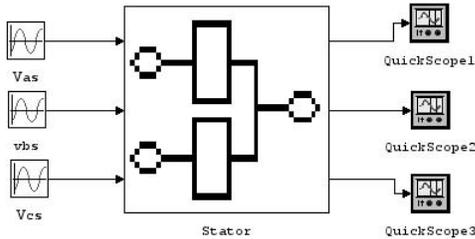


Fig. 9. Experimental framework

For each simulation of the system during $t_f = 0.4s$, the simulation time (in second) and the number of events obtained under an Intel-Pentium M processor 1.70GHz with 1Go of RAM are reported (Table II).

dq	QSS		QSS2		QSS3	
	Time	Event	Time	Event	Time	Event
0.1	1.485	163	1.255	86	0.857	71
0.01	2.632	2000	1.198	322	0.922	199
0.001	10.969	20301	1.289	1040	0.925	440
0.001	96.652	203381	2.549	3287	0.938	1010

TABLE II
SIMULATION RESULTS

These data are available using the functions “Time Counter” and “Event Counter” atomic models from the PowerDEVS “Sinks” library. The previous results show that the number of events and the simulation time are in reverse proportion to the quantum dq for any method used. It is due to the fact that the smaller this quantum is, the larger the number of events will be. Therefore, the simulation time will be larger with small quantum.

The accuracy of the results has been demonstrated using a simple steady state simulation (Fig. 10). The variable plotted is time evolution of the stator current $i_{as}(t)$ in the first phase considering quantum dq for the QSS, QSS2 and QSS3 methods. The proposed simulation does not show the transient which clearly disappears after regime which disappears after 10 periods (with a period $T = 0.02s$) to remain in steady state. It is more than clear that the quantum has a large influence of the result accuracy. A zoom is performed during the steady state (Fig. 10.b) in between the time 2.5s and 2.9s for quantum values ranging from 0.1 to 0.0001 by steps of 10^{-1} . The expected solution is computed by integration of the set of ODE with classical Euler method with a time step of 0.001ms. When the quantization step is 0.1, it can be observed that the step-by-step solution is exact compared with the solutions given for smaller quantum. If a linear interpolation is used, the numerical error between this first solution and the other with smaller quantum is very important and it is not acceptable. On the other hand, when the quantization step reaches the value 0.001, the error using a linear interpolation is smaller than in the previous case and the numerical error becomes acceptable. When the quantum decreases, there is a natural increase on the computation time. The QSS3 method allows an interesting simulation time depending in the quantum choice. The QSS2 method is not as fast as the QSS3 method (rate of $\frac{1}{2}$ with QSS3 for $dq=0.0001$) but is less restrictive.

IV. CONCLUSION AND FUTURE DIRECTIONS

In this paper, it has been shown that the PowerDEVS environment allows the modelling and the simulation of complex electrical power systems. The proposed implementation for the induction machine stator model has permitted to test the QSS quantization methods and to appreciate how efficient they can be in term of computational cost and efficiency. The QSS2 method gives the better results compared to its former version QSS. Indeed, the usage of QSS2 within the integration procedure used in each phase of the stator model permits to converge on the expected solution. The QSS3 method is faster than QSS2 but the choice of a valid quantum is very critical. It has been shown that the insertion of a low-pass filter

connected to the output is necessary since it corresponds with the derivative of each stator current in the three phases.

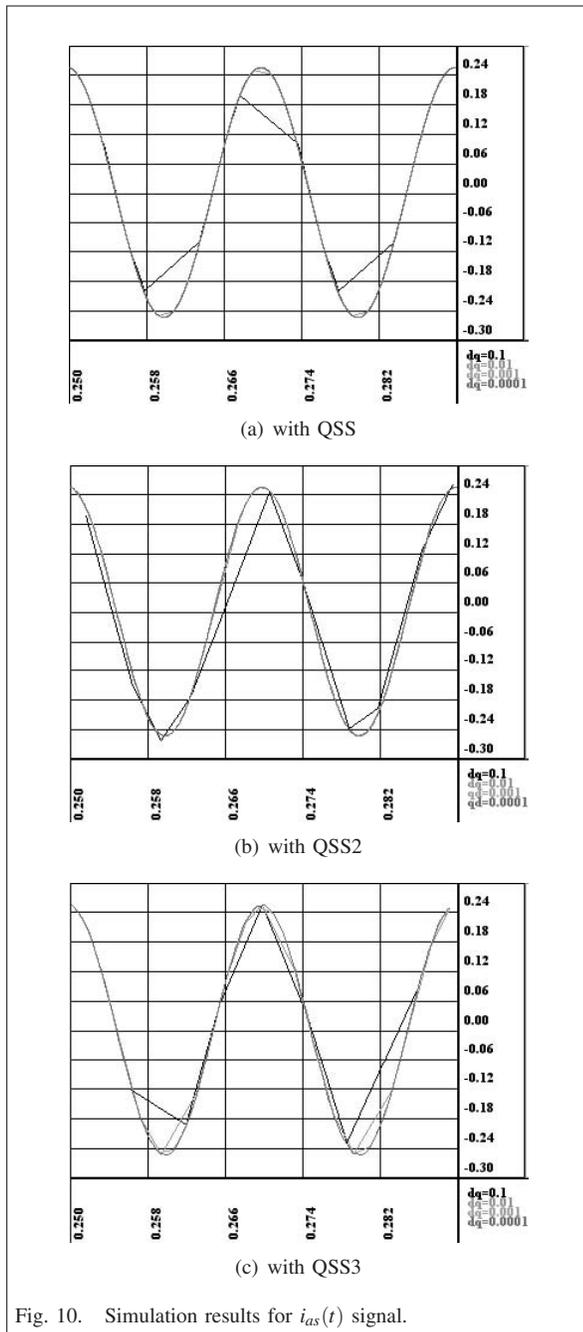


Fig. 10. Simulation results for $i_{as}(t)$ signal.

This process is used to reduce the delay introduced by the derivation and it is known as a stabilization of the final solution. Moreover, this paper presents the adaptation of the low-pass filter time constant to reduce the simulation time. This approach is usual to that kind of systems but it has never been tested in the PowerDEVS environment. In a short-term prospect, it is needed to extend the stator model to the complete three-phase squirrel-cage or wound-rotor induction machine in order to perform effective fault simulation as a discrete event phenomenon. The obtained results with PowerDEVS show some advantages compared with other simulation packages which use discrete time integration methods as MATLAB/SIMULINK(C). Firstly, the

accuracy with the DEVS approach is not greatly affected by the quantum value and then, the simulation time can be minimized using high order interpolation methods. Moreover, the simple DEVS approach shows a performance superior to the complex implicit, high order and variable step methods. This first approach of the three-phase induction machine stator model has allowed to perform a simple and efficient model of three-phase AC electrical machines. Indeed, the choice of the QSS quantized method within this complete model is based on the present paper results. The final objective will require the definition of effective fault models from the experiments achieved on a laboratory test-bench or on a real industrial system. Once the AC machine models will be established and simulated in PowerDEVS, the complete model will be transformed into BFS-DEVS (Behavioral Fault Simulation for DEVS) [16] in order to build a fault simulator based on the concurrent and comparative simulation approach.

REFERENCES

- [1] M. Boussak and G. Capolino, "Comparison of some integration procedures related to universal machine equations," October 1988. paper 88R-028, 15 pages, Leuven (Belgium).
- [2] B. P. Zeigler, H. Praehofer, and T. G. Kim, *Theory of Modeling and Simulation, Second Edition*. Academic Press, 2000.
- [3] N. Giambiasi and J.-C. Carmona, "Generalized discrete event abstraction of continuous systems : GDEVs formalism," *Simulation Modelling Practice and Theory, Elsevier*, vol. 14, pp. 47–70, january 2006.
- [4] B. P. Zeigler and J. Lee, "Theory of quantized systems: Formal basis for DEVS/HLA distributed simulation environment," in *Proceedings of SPIE*, pp. 49–58, 1998.
- [5] E. Kofman and S. Junco, "Quantized-state systems: a DEVS approach for continuous system simulation," *Trans. Soc. Comput. Simul. Int.*, vol. 18, no. 3, pp. 123–132, 2001.
- [6] E. Kofman, "Quantization-based simulation of differential algebraic equation systems," *Trans. Soc. Comput. Simul. Int.*, vol. 79, no. 7, pp. 363–376, 2003.
- [7] M. C. D'Abreu and G. A. Wainer, "Hybrid dynamic systems: models for continuous and hybrid system simulation," in *WSC '03: Proceedings of the 35th conference on Winter simulation*, pp. 641–649, Winter Simulation Conference, 2003.
- [8] E. Kofman, M. Lapadula, and E. Pagliero, "PowerDEVS: A DEVS-based environment for hybrid system modeling and simulation," tech. rep., Rosario National University, 2003. <http://fceia.unr.edu.ar/~kofman/pubs.html>.
- [9] B. P. Zeigler, *Theory of Modeling and Simulation*. Academic Press, 1976.
- [10] B. P. Zeigler, "An introduction to set theory," tech. rep., ACIMS Laboratory, University of Arizona, 2003. <http://www.acims.arizona.edu/EDUCATION/>.
- [11] J. de Lara and H. Vangheluwe, "AToM3: A tool for multi-formalism and meta-modelling," in *FASE '02: Proceedings of the 5th International Conference on Fundamental Approaches to Software Engineering*, (London, UK), pp. 174–188, Springer-Verlag, 2002.
- [12] A. Naamane and N. Giambiasi, "Diamsim : a discrete event modeling and simulation environment," in *International Conference Automatics and Informatics*, (Sofia), pp. 159–168, juin 2001.
- [13] E. Kofman, "A second order approximation for DEVS simulation of continuous systems," *Trans. Soc. Comput. Simul. Int.*, vol. 78, no. 2, pp. 76–89, 2002.
- [14] B. P. Zeigler, "DEVS theory of quantized systems," tech. rep., ACIMS Laboratory, University of Arizona, 2004. www.acims.arizona.edu/PUBLICATIONS/CDRLs/UnivArizonaCDRL1.doc.
- [15] E. Kofman, "A third order discrete event simulation method for continuous system simulation. part i: Theory," tech. rep., 2005. <http://fceia.unr.edu.ar/~kofman/pubs.html>.
- [16] L. Capocchi, F. Bernardi, D. Federici, and P. Bisgambiglia, "BFS-DEVS: A general DEVS-based formalism for behavioral fault simulation," *Simulation Modelling Practice and Theory, Elsevier*, vol. 14, pp. 945–970, October 2006.