Automatic modelling of Power Electronic Converter, Average model construction and Modelica model generation

Loig ALLAIN
LMS
La Cité Internationale, 84 quai Charles de Gaulle,F- 69006 LYON
loig.allain@lmsintl.com

Asma MERDASSI, Laurent GERBAUD
Seddik Bacha
G2ELAB
UMR 5269 INPG/UJF-CNRS
ENSE3, Domaine Universitaire, BP 46 SAINT MARTIN D'HERES F-38402,
Asma.Merdassi@g2elab.grenoble-inp.fr,
Laurent.Gerbaud@g2elab.grenoble-inp.fr,
Seddik.Bacha@g2elab.grenoble-inp.fr

Abstract

Development in semi-conductor production and the control of static converters leads to an increase use of these devices in classical electrical motor control application. But, the simulation of these elements in a complete system involves longer computation time in order to take into account the impact of switching on the global system. Using average models to simulate them brings a drastic decrease of computation time. Such models do not take into account switching. They are a good compromise between complexity, computing time and acceptable accuracy at a system level. Even if the method to build these models is well identified, it still requires a lot of manipulation to get from the very structure description to the simulation code. This paper deals with a method that automates the average modeling of power electronic converter. The associated tool prototype (AMG) that has been implemented is also presented and some simulation results are shown using the Imagine.Lab AMESim[1] and its Modelica Platform.

Keywords: code generation; average model; power electronics, Imagine.Lab AMESim

1 Introduction

Nowadays, numerous electrical actuators and drives are based on the association of power electronics and electrical motors driven by controller. Such association guarantees better time response and better efficiency for the whole actuator. Indeed, the control of electronic commutations allows a fine control of the power supply of motors.

Figure 1presents the most classical structure of an electromechanical actuator.

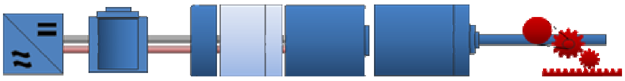


Figure 1: General structure for electro-mechanical actuator

The simulation of Control-Converter-Machine sets is a CPU time consuming task beacause it involves the complete simulation of power static converter and especially of power switch commutations. Indeed, classical PWM frequency may be about 500Hz to 50kHz, and has to be compared to the time response of the whole system that could be about minutes or more. So, it is important to find a formulation for power electronic converter that reduces the CPU time need.

The G2ELAB has a strong experience in this area, and is involved in the automatic analysis and simulation of power electronics converters [5][7]. Besides, a strong knowledge of engineer modeling experience and its specifities lead to the development of software tools to store and manage the engineer knowledge thanks to an object oriented language [9]. In this way, these developments allowed to explore new concepts about average modeling of power electronic structures. In an other direction, LMS-Imagine has a long experience in multiphysical system modeling, the code of his known software Imagine.Lab AMESim has been used world around to simulate some complex, multi-physical devices[4]. Using the facility of Imagine.Lab Modelica Platform, the G2ELAB has been able to develop and test its approach applied to generate Modelica models for power electronic converter.

Obviously, the software tools presented hereafter are not linked to the Imagine.Lab platform. As the produced code is fully Modelica compliant, it may be used in any Modelica tool.

After, an introduction to average modeling in power electronics, the paper explores the way to automate code generation for average model in power electronics [7], [8]. The developed tool (namely AMG) deals with the following kinds of static converters: DC/DC (as in [2], [3], [4], [5]) but also DC/AC and AC/AC. The generated Modelica code is shortly presented and used to create a simulation model within the AMESim.Lab Modelica Platform. The generated code is presented for two static power converters: a three phase voltage inverter (DC/AC conversion) and a boost converter (DC/DC conversion). Every example aims at emphasizing interest point of static converter average models. Then, some numerical results are shown and discussed and the numerical efficiency and reliability are discussed with regards to classic representation. Eventually, advantages and drawbacks of average models are discussed as well as the interest of such an approach.

2 Average Model of power electronics converters

2.1 Electrical Actuator and Power Electronics

An electrical actuator aims at converting electrical energy into mechanical one. One of the eldest technologies used is the electrical motor. Using Direct Current (DC), or Alternating Current (AC) as an electrical power source, it produces a mechanical torque through electromagnetic transformation. Most common electrical motors can be driven through a control of the voltage of their power supply. This is achieved using power electronics switching to produce whether an equivalent voltage value or a given frequency [5]. Designing systems based on electronic power supply requires being able to simulate them. A well known drawback is that the classical simulation of power electronics must take into account the switching frequency and the transient phenomena of electronic switches. This requires a large amount of CPU time and may neither compliant with Real Time Simulation nor with controller design steps. In order to reduce the needed CPU time to simulate such complex associations, average model of power supply can be used.

Moreover, it is obvious for every electromechanical designer that, indirect conversion is a widely used technology to control both the amplitude and the frequency of applied voltage. As a consequence, one has to face the tricky problem of increasing the complexity of a whole system, and so the required simulation time.

2.2 Average Model of power Electronics converter

The purpose of the hereafter presented tool is not to question a well know process leading from a circuit description to an average model [2] [3] [4] [5] [6].

The paper aims at automating fully the process [7], [8] for DC/DC, DC/AC and AC/AC static converters. Please note that, in the paper, only controlled commutations are considered, so discontinuous conduction is not taken into account.

Here, a pure symbolic method (exact or topological model) is chosen to obtain average models.

3 Average model construction and code Generation

3.1 The generated models by AMG

3.1.1 Exact model

For an operating mode of a static converter, N possible configurations during the commutation period T are given.

The global state model for every topology (indexed by i) is:

$$\frac{dx}{dt} = \sum_{i=1}^{N} \left(A_i \cdot x + B_i \cdot e\right) \cdot f_i \quad (1)$$

Where:

- N: the number of topologies (configurations) that happen during the switching period T.
- Ai : the state matrix for the ith configuration.
- Bi : the input matrix.
- e : the external sources.

For each switching cell, an equivalent control function is assigned: $h_i(t) \in \{0,1\} \text{or} \{-1,1\}$ that depends on f_i (a configuration validation function of *i*):

$$f_i = \begin{cases} 1 \;\; for \quad t \in [\; t_{i-1}, t_i [\;\; ; i = 1,..,N \\ 0 \;\; elsewhere \end{cases}$$

The global behavior of the static converter is formulated as equation (2):

$$\frac{dx}{dt} = Ax + \sum_{i=1}^{N} h_i (B_i x + b_i) + d \qquad (2)$$

Where:

- N: the number of topologies that happen during the commutation period.
- x : the state vector (dim[n]).
- A :the state matrix (dim[n,n]).
- B_i, b_i : the input matrixes (dim[n,n], dim[n,1]).
- d (dim[n,1]).

A, B_i, b_i are independent of x and the h_i.

3.1.2 Large Signal Average Model

The large signal average model transforms an exact model into an averaged one.
To calculate such an averaged model, the following properties are used for the continuous state:

$$\begin{cases} \left\langle \dfrac{dx}{dt} \right\rangle_0 = \dfrac{d}{dt}\langle x \rangle_0 \\ \langle x \cdot y \rangle_0 = \langle x \rangle_0 \cdot \langle y \rangle_0 \end{cases} \quad (3)$$

By applying equations (3) on equation (2), the equation of the large signal average model is deduced:

$$\begin{aligned} &\frac{d}{d\,t}\langle x \rangle_0 = \langle A \rangle_0 \cdot \langle x \rangle_0 \\ &+ \sum_{i=1}^{N}\left(\langle h_i \rangle_0 \cdot \langle B_i \rangle_0 \cdot \langle x \rangle_0 + \langle b_i \rangle_0 \cdot \langle h_i \rangle_0\right) + \langle d \rangle_0 \end{aligned} \quad (4)$$

In general, with linearization hypothesis, matrixes A, B_i and b_i are time invariant. So (4) can be simplified as (5):

$$\begin{aligned} &\frac{d}{d\,t}\langle x \rangle_0 = A \cdot \langle x \rangle_0 \\ &+ \sum_{i=1}^{N}\left(\langle h_i \rangle_0 \cdot B_i \cdot \langle x \rangle_0 + b_i \cdot \langle h_i \rangle_0\right) + \langle d \rangle_0 \end{aligned} \quad (5)$$

3.1.3 Small Signal Average Model

This model is also called the small signal state space model. It linearises the exact model and does not include any nonlinear elements.
This model is calculated by a linearization around the balance operating point (xe, ue) of the static converter.

This point is calculated for $\dfrac{dx}{dt} = 0$.

There, the modelling approach begins with equation (6).

$$\begin{cases} \dfrac{dx}{dt} = f(x,u) \\ y = g(x,u) \end{cases} \quad (6)$$

where:

- y : the output vector (dim[q])
- u : the input vector (dim[p]).

Equation (6) is transformed into the small signal average model (7):

$$\begin{cases} \dfrac{d\tilde{x}}{dt} = \tilde{A} \cdot \tilde{x} + \tilde{B} \cdot \tilde{u} \\ \tilde{y} = \tilde{C} \cdot \tilde{x} + \tilde{D} \cdot \tilde{u} \end{cases} \quad (7)$$

The matrixes of the small signal state space model are calculated by using the following expressions:

$$\tilde{A} = \left(\frac{\partial f(x,u)}{\partial x}\right)_{x_e,u_e} ; \tilde{B} = \left(\frac{\partial f(x,u)}{\partial u}\right)_{x_e,u_e}$$

$$\tilde{C} = \left(\frac{\partial g(x,u)}{\partial x}\right)_{x_e,u_e} ; \tilde{D} = \left(\frac{\partial}{\partial u} g(x,u)\right)_{x_e,u_e}$$

where:

- $\begin{cases} \tilde{u} = u - u_e \\ \tilde{x} = x - x_e \\ \tilde{y} = y - y_e \end{cases}$
- $\tilde{A}$: the state matrix (dim [n,n]).
- $\tilde{B}$: the input matrix (dim[n,p]).
- $\tilde{C}$: the output matrix (dim[q,n]).
- $\tilde{D}$: the feedforward matrix (dim[q,p]).

The transfer function is also deduced from (7):

$$\frac{y(s)}{u(s)} = \tilde{C} \cdot (s \cdot I - \tilde{A})^{-1} \cdot \tilde{B} + \tilde{D} \quad (8)$$

- I : The matrix identity (dim [n, n]).

3.1.4 Generalized Average Model

This model is applied on static converters which have one or several alternative variable states.
The general formulation of the state derivatives is:

$$\left\langle \frac{dx}{dt} \right\rangle_k = \frac{d\langle x \rangle_k}{dt} + j \cdot k \cdot \omega \cdot \langle x \rangle_k \quad (9)$$

where:

- k: The range of harmonic.

Equation (9) becomes:

- For alternative state variables (first harmonic, i.e.: k=1):

$$\left\langle \frac{dx}{dt} \right\rangle_1 = \frac{d\langle x\rangle_1}{dt} + j\cdot\omega\cdot\langle x\rangle_1 \qquad (10)$$

- For continuous states (k=0):

$$\left\langle \frac{dx}{dt} \right\rangle_0 = \frac{d\langle x\rangle_0}{dt} \qquad (11)$$

The exact model of equation (2) becomes:

$$\begin{aligned}\frac{d}{dt}\langle x\rangle_k &= -j\cdot k\cdot\omega\cdot\langle x\rangle_k + A\cdot\langle x\rangle_k \\ &+\sum_{i=1}^{N}\left(B_i\langle x\cdot h_i\rangle_k + \langle b_i\cdot h_i\rangle_k\right) + \langle d\rangle_k\end{aligned} \qquad (12)$$

The development of its terms is made by using (13):

$$\begin{aligned}&\left\langle \prod_{i=1}^{N} a_i \right\rangle_k = \sum\left(\prod_{i=1}^{N}\langle a_i\rangle_{j_i}\right) \\ &\begin{cases}(j_i)\in\{-1,0,1\} \\ \sum^{N} j_i = k\end{cases}\end{aligned} \qquad (13)$$

The expressions of the alternative state variables are written under their complex shape (14):

$$\begin{cases}\langle a\rangle_1 = \mathrm{Re}(a) + j\,\mathrm{Im}(a) \\ \langle a\rangle_{-1} = \langle a\rangle_1^* = \mathrm{Re}(a) - j\,\mathrm{Im}(a)\end{cases} \qquad (14)$$

In general, the applications of such a model are limited to the fundamental (k=1) and averaged values (k=0).

The large signal average model is a specific application of the generalized average model.

3.1.5 The Equivalent Average Generator

The approach used to make such a model is described in Figure 2.

The equivalent average generator eliminates the discontinuous variables. The state variables are separated into continuous (slow) and alternatives (fast) variables.

The state vector is composed of two sub-vectors:

$$x = [x_s; x_f]$$

where:

- x_s : the vector of slow state variables.
- x_f : the vector of fast state variables.

Now, the paper presents the steps of calculation:

- The behavior of the slow state variables given by equation (15):

$$\begin{aligned}\frac{dx_s}{dt} &= A_s^s\cdot x_s + A_s^f\cdot x_f \\ &+\sum_{i=1}^{N}\left(B_{is}^s\cdot x_s + B_{is}^f\cdot x_f\right)\cdot h_i + b_{is}\cdot h_i + d_s\end{aligned} \qquad (15)$$

- The behavior of the slow state variables given by equation (16):

$$\begin{aligned}\frac{dx_f}{dt} &= A_f^s\cdot x_s + A_f^f\cdot x_f \\ &+\sum_{i=1}^{N}\left(B_{if}^s\cdot x_s + B_{if}^f\cdot x_f\right)\cdot h_i + b_{if}\cdot h_i + d_f\end{aligned} \qquad (16)$$

- For the continuous state variables x_s, the large signal average model is applied to (15):

$$\begin{aligned}\frac{d\langle x_s\rangle_0}{dt} &= A_s^s\cdot\langle x_s\rangle_0 \\ &+\sum_{i=1}^{N}\left(B_{is}^s\cdot\langle x_s\cdot h_i\rangle_0 + B_{is}^f\cdot\langle x_f\cdot h_i\rangle_0\right) + b_{is}\cdot\langle h_i\rangle_0 + \langle d_s\rangle_0\end{aligned} \qquad (17)$$

- For the alternative state variables x_f the generalized average model (16) (limited in this example to the first harmonic) is used and x_f is considered to be at its steady state:

$$\begin{aligned}\frac{d\langle x_f\rangle_1}{dt} &= -j\omega\cdot\langle x_f\rangle_1 + A_f^f\cdot\langle x_f\rangle_1 + A_f^s\cdot\langle x_s\rangle_1 \\ &+\sum_{i=1}^{N} B_{if}^f\langle x_f\cdot h_i\rangle_1 + B_{if}^s\cdot\langle x_s\cdot h_i\rangle_1 + b_{if}\cdot\langle h_i\rangle_1 + \langle d_f\rangle_1\end{aligned} \qquad (18)$$

- Equation (18) is solved by assuming that all the continuous state variables are constant.

The development of the terms: $\langle x_f\cdot h_i\rangle_0$, $\langle x_s\cdot h_i\rangle_0$, $\langle x_f\cdot h_i\rangle_1$ and $\langle x_s\cdot h_i\rangle_1$ is made by using (13).

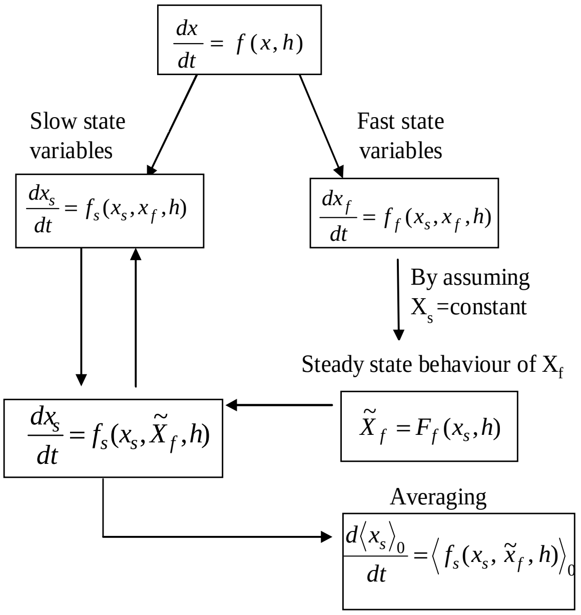

$\frac{dx}{dt}=f(x,h)$
Slow state variables
Fast state variables
$\frac{dx_s}{dt}=f_s(x_s,x_f,h)$
$\frac{dx_f}{dt}=f_f(x_s,x_f,h)$
By assuming X_s=constant
Steady state behaviour of X_f
$\frac{dx_s}{dt}=f_s(x_s,\tilde{X}_f,h)$
$\tilde{X}_f=F_f(x_s,h)$
Averaging
$\frac{d\langle x_s\rangle_0}{dt}=\langle f_s(x_s,\tilde{x}_f,h)\rangle_0$

Figure 2: Required operations to get an equivalent average generator

To obtain the equivalent average generator model, the steady state of the alternative variables x_f (in equation 18) is obtained when $\frac{d\langle x_f\rangle_1}{dt}=0$ and replaced this steady state (stationary behavior) in equation (17).

3.2 Overview of code generation process

A large subset of the required operation as early presented can be automated; Figure 3 presents the general process that has been implemented in AMG generator.

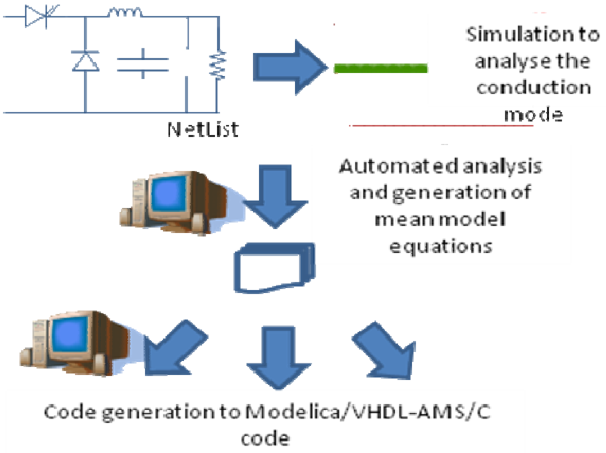

NetList
Simulation to analyse the conduction mode
Automated analysis and generation of mean model equations
Code generation to Modelica/VHDL-AMS/C code

Figure 3: Average model generation

Each model is entirely made in an automatic way by the generator AMG, from an initial description of a static converter structure:

- a netlist file: here the AMG user describes its structure in P-Spice by using a dedicated simplified component library); any tool giving a P-Spice netlist is also available.
- the switching mode, i.e. the cyclic sequence of switched configurations in an operating period,
- the switch control, mainly the control links of switches.

This description is given by the AMG user and induces a great "*a priori*" on the behaviour of the described static converter.

The average models are made from the state equations of every configuration of the converter operating mode.

3.3 Global model of static converters and causality

In order to automatically build the Average or exact model for static power converter, the kind of the connected electrical sources (or loads) must be defined. One could understand that a kind of causality is "de facto" imposed to the converter, but it should be noted that the resulting equations could be dealt in the same way that any other Modelica model. The point is that the limits of the converter have to be carefully chosen according to the kind of average model to generate:

- An exact model shall gather only the electronics switches and need a "classical" control law. This law shall be strictly identical to the one used for non average converter
- An average model aims at building the average value of state variable. As a consequence, one must cautious to take into account the component introducing state variable in the final description. Moreover the required input is also different as it should also be averaged. This is a fully explained with the boost converter example, below.

The model of the static converter is linked to the models of the other components of the system to simulate.

The causality on the pluging points is required **only to build the model**. This causality is cancelled by connecting the generated models with other models. This is useful to create the models of static converter that fed an electrical machine, without knowing the machine model that is generally not available in a circuit shape (see Figure 4). Eventually, the true causality is "set" by the Modelica compiler while it reorganizes mathematical relations to produce an executable code.

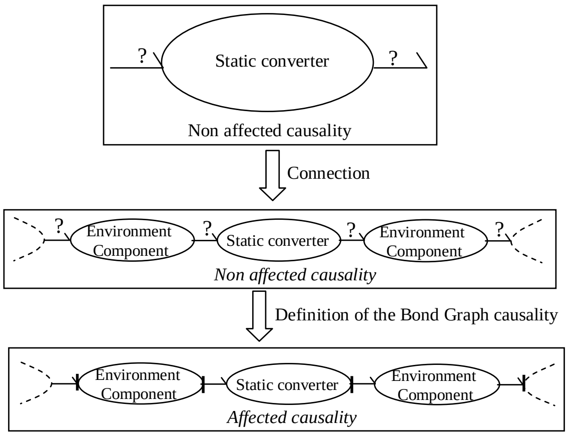


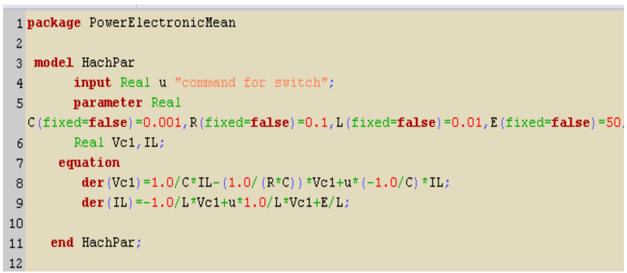
Figure 4 From a non oriented representation to an oriented representation: Bond Graph representation

One has to be cautious, while assessing the causality for electrical connectors, to respect the definition given in Bond Graph representation for the electrical field. In other words, the environment of a static converter must be translated into equivalent sources: a voltage source or a current source is imposed at every connection point.

3.4 Modelica Code and simulation

The process as explained above leads to the following code as it appears in Imagine.Lab Modelica Code Editor (Figure 5). This code has been generated for a Boost converter and includes both the source and load of static converter.

As a reminder, eventually this code shall be used either by Imagine.Lab platform or any other Modelica tool to create a simulation code. The Imagine.Lab platform allows to create a dedicated C fully compliant with the rest of the Imagine.Lab multi-physical libraries.



```
package PowerElectronicMean

 model HachPar
       input Real u "command for switch";
       parameter Real
C(fixed=false)=0.001,R(fixed=false)=0.1,L(fixed=false)=0.01,E(fixed=false)=50
       Real Vc1,IL;
     equation
        der(Vc1)=1.0/C*IL-(1.0/(R*C))*Vc1+u*(-1.0/C)*IL;
        der(IL)=-1.0/L*Vc1+u*1.0/L*Vc1+E/L;

   end HachPar;

```

Figure 5: Boost Chopper Mean Model

4 Implementation

Every model is entirely made in an automatic way by the AMG software tool, from an initial description of a state converter structure (here a netlist file generated by PSpice), the commutation mode (cyclic sequence of switched configuration), the switch control sequence and the nature of state variables (slow or fast). Figure 6 introduced an overview of this process.

Firstly, the switch control and the operating mode are defined and the nature of state variables are deduced from the analyze of a simulation of the studied static converter, i.e. by using a power electronics simulation software.

Secondly, a topological description is carried out from a Netlist file extracted by PSpice.

This information is introduced into the software tool AMG and finally several models are generated according to the nature of the power conversion and the state variables.

The obtained result is a file which contains the desired models in an ASCII (text) shape.

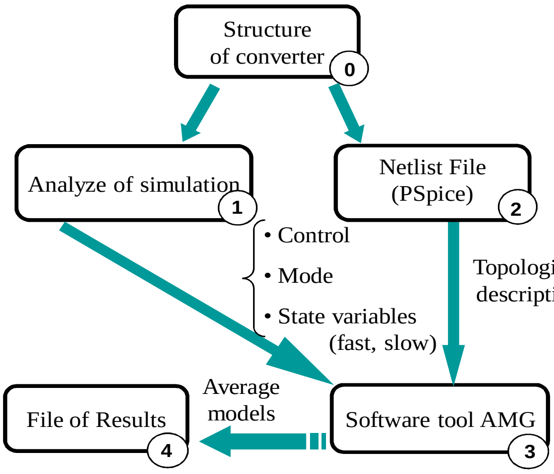


Figure 6: Steps of modeling

4.1 Exact model building

The first step is the analysis of the circuit NetList to carry out the incidence matrix of the whole circuit. Then, this matrix is reduced for each state, according to the conduction state of switches (state ON switches are considered as "short-circuits" or state OFF switches are considered as "open circuits"). For each configuration, the state matrixes are calculated from their simplified node equations. So, the global state space model is obtained by a weighted sum of the state matrixes of each mode state.

4.2 Implementation

In AMG, Java and Maple are combined to implement the previous approach. The building of the state matrixes for every configuration is developed in Java. The extraction of the averaged models from the symbolic approach is made in Maple as shown in Figure 7.

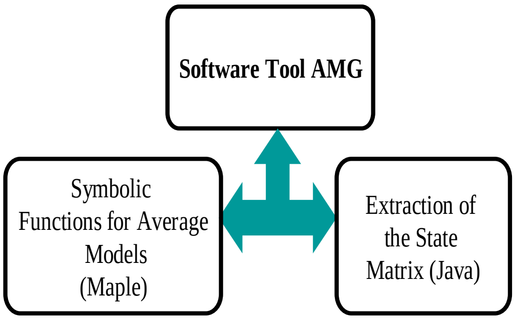

Software Tool AMG
Symbolic Functions for Average Models (Maple)
Extraction of the State Matrix (Java)

Figure 7: Software tool AMG

At the present time, AMG deals only with continuous conduction operating modes.

5 First application: Three phase inverter

5.1 Description of the three-phase voltage inverter

Let's have a look at the static converter known as three phase inverter and the assumption needed to build the average model. Figure 8 presents a three phase voltage inverter with a basic inductive load. Obviously, such an inductive load may even be an electrical motor.

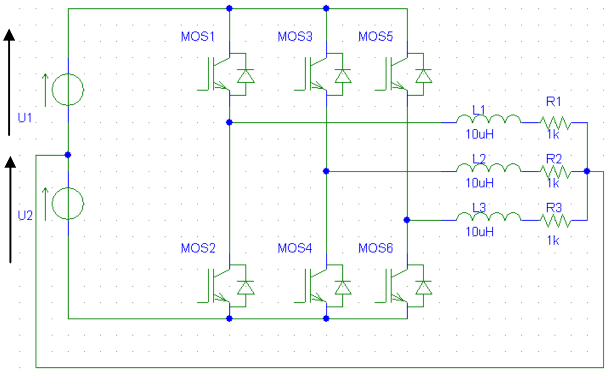


Figure 8: Three phase voltage inverter feeding a three phase electrical load

In this case, each commutation cell is an inverter legs. For, one leg when a switch is controlled to be on (i.e. passing) the other one is off. As a consequence, the inverter needs only three control signals corresponding to the upper switch of the arm (MOS1, MOS3 and MOS5). In the sequel, these signals will be known as h1, h2 and h3 respectively connected to MOS1, MOS3 and MOS5.

Each control signal is a full wave control one (not a pulse wave modulation, i.e. PWM). It is a square wave with a duty cycle of 50%. Moreover, to produce equilibrated three phase system, the three control signals are shifted of $2\pi/3$ rad. (see Figure 9).

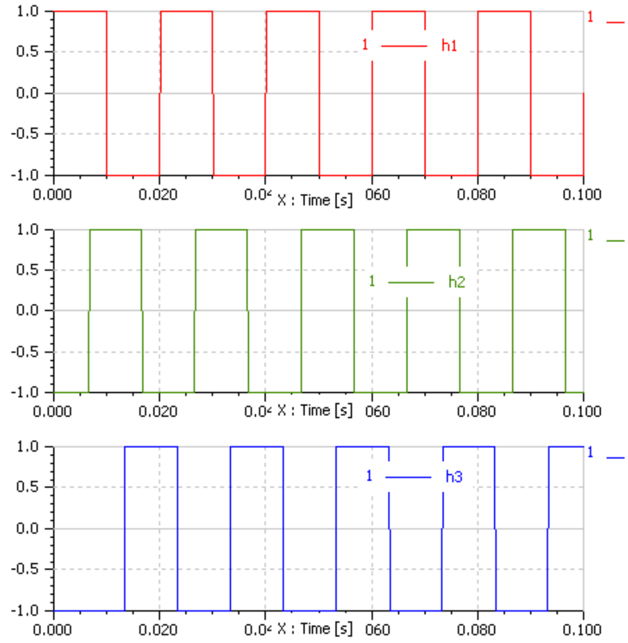


Figure 9: Full wave control for a three phase voltage inverter

5.2 Causality of the static converter

As explained above, AMG needs basic assumptions on the electrical sources to connect. For a voltage inverter, one cans obvious state that the power source is a voltage while the electrical load (inductive one) is a current source.

This intellectual building allows to extract the power static converter from its electrical neighborhood, as a consequence, it can be seen as presented on Figure 10.

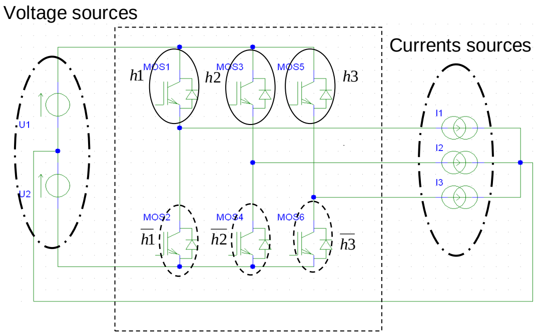
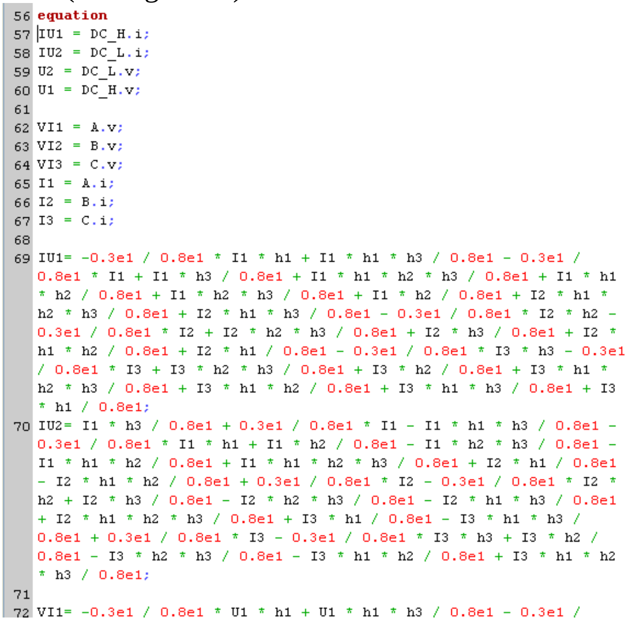


Figure 10: Causality for the three phase voltage inverter

5.3 Generation of the exact model

With the above assumptions on both electrical sources and load and on control laws, AMG is now able to generate the average model of the converter. Hereafter is a snapshot of the generated Modelica code (see Figure 11).



```
equation
IU1 = DC_H.i;
IU2 = DC_L.i;
U2 = DC_L.v;
U1 = DC_H.v;

VI1 = A.v;
VI2 = B.v;
VI3 = C.v;
I1 = A.i;
I2 = B.i;
I3 = C.i;

IU1= -0.3e1 / 0.8e1 * I1 * h1 + I1 * h1 * h3 / 0.8e1 - 0.3e1 /
0.8e1 * I1 + I1 * h3 / 0.8e1 + I1 * h1 * h2 * h3 / 0.8e1 + I1 * h1
* h2 / 0.8e1 + I1 * h2 * h3 / 0.8e1 + I1 * h2 / 0.8e1 + I2 * h1 *
h2 * h3 / 0.8e1 + I2 * h1 * h3 / 0.8e1 - 0.3e1 / 0.8e1 * I2 * h2 -
0.3e1 / 0.8e1 * I2 + I2 * h2 * h3 / 0.8e1 + I2 * h3 / 0.8e1 + I2 *
h1 * h2 / 0.8e1 + I2 * h1 / 0.8e1 - 0.3e1 / 0.8e1 * I3 * h3 - 0.3e1
/ 0.8e1 * I3 + I3 * h2 * h3 / 0.8e1 + I3 * h2 / 0.8e1 + I3 * h1 *
h2 * h3 / 0.8e1 + I3 * h1 * h2 / 0.8e1 + I3 * h1 * h3 / 0.8e1 + I3
* h1 / 0.8e1;
IU2= I1 * h3 / 0.8e1 + 0.3e1 / 0.8e1 * I1 - I1 * h1 * h3 / 0.8e1 -
0.3e1 / 0.8e1 * I1 * h1 + I1 * h2 / 0.8e1 - I1 * h2 * h3 / 0.8e1 -
I1 * h1 * h2 / 0.8e1 + I1 * h1 * h2 * h3 / 0.8e1 + I2 * h1 / 0.8e1
- I2 * h1 * h2 / 0.8e1 + 0.3e1 / 0.8e1 * I2 - 0.3e1 / 0.8e1 * I2 *
h2 + I2 * h3 / 0.8e1 - I2 * h2 * h3 / 0.8e1 - I2 * h1 * h3 / 0.8e1
+ I2 * h1 * h2 * h3 / 0.8e1 + I3 * h1 / 0.8e1 - I3 * h1 * h3 /
0.8e1 + 0.3e1 / 0.8e1 * I3 - 0.3e1 / 0.8e1 * I3 * h3 + I3 * h2 /
0.8e1 - I3 * h2 * h3 / 0.8e1 - I3 * h1 * h2 / 0.8e1 + I3 * h1 * h2
* h3 / 0.8e1;

VI1= -0.3e1 / 0.8e1 * U1 * h1 + U1 * h1 * h3 / 0.8e1 - 0.3e1 /
```

Figure 11: Exact model for a three phase voltage inverter generated by AMG

Analyzing the code, one can see the expression of DC bus current as a function of:

- Control signal: h1, h2, h3
- Current in phase I1, I2 and I3

In the same way, the voltage drop applied in the phase is a function of:

- The DC bus voltage (U1, U2)
- The control signal h1, h2, h3

The interest of the AMG tool becomes obvious by having a look at the equation generated for this very basic power converter structure. One can easily imagine the awful they may have for a multi-level converter.

5.4 Building of a complete application

Let us observe the numerical results obtained with a three phase inductive load. This can be achieved either by adding specific relation like presented in the Figure 12, or by connecting this component to electrical components from the Modelica Standard Library. (Figure 13)

```
/* Added relation for Three phase load*/

der(IL1)= (V1-(R4*I1))/L1;
der(IL2)= (V5-(R4*I2))/L2;
der(IL3)= (V3-(R4*I3))/L3;
```

Figure 12: Equations for a three phase RL load

Figure 13 presents the integration of the above model within a “complete circuit”.

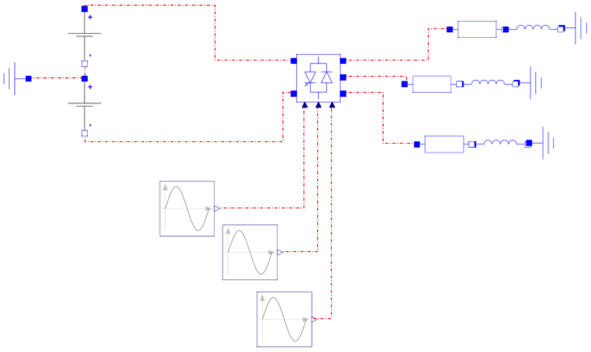


Figure 13: Three phase voltage inverter with electrical load

5.5 Numerical Results

Let us run a simulation with the following parameters:

R = 2 ohm; L=0.01H;E=320V;freq=50Hz

The simulated currents are represented on Figure 14.

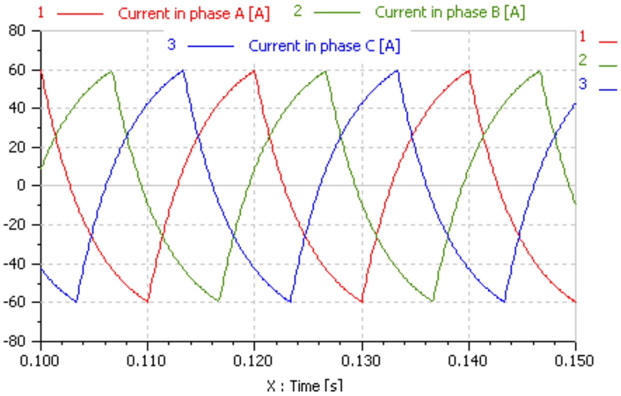


Figure 14: Simulation results: currents in the inductors

The influence of inductors may be observed on the following results.

One has to note that such numerical result is observable as there is no parasitic time constant due to inner resistance of electronic. In fact, in classical models like Ron/Roff ones, additional time constants are observed when switches are connected to inductors (or capacitors).

Then using exact model (as well as average one) brings the guaranty that no additional time constant is added to the whole electrical circuit.

6 Second application

6.1 Description of the DC-DC converter

The studied converter in this part is a Boost converter. The electrical conversion is DC/DC. Boost converter.

The static DC/DC boost converter allows increasing the voltage level. This simple example demonstrates both the capacity to analyze DC/DC converter and to generate both exact and average power converter models.

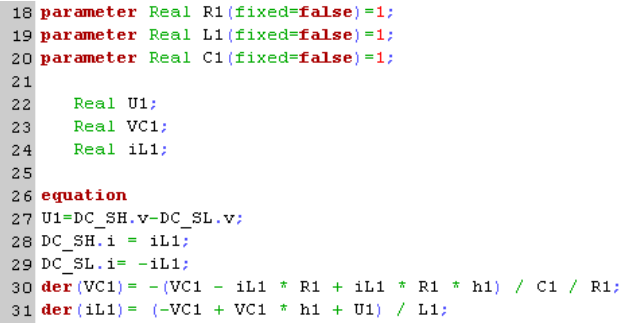
An exact model of a power converter allows to take into account the real control on electronic switches. Contrary average model aims at averaging the state variable of the circuit.

It should be noted that using an average model, the control law shall also be averaged:

- In case of DC/DC conversion, the input value is supposed to be the duty cycle
- In DC/AC conversion, average control is the sinus law for output voltage (or the shape is)

6.2 Average model of the static converter

Two kinds of model have been generated by AMG. Figure 15 presents the exact model and Figure 16 the average model (a large signal model).

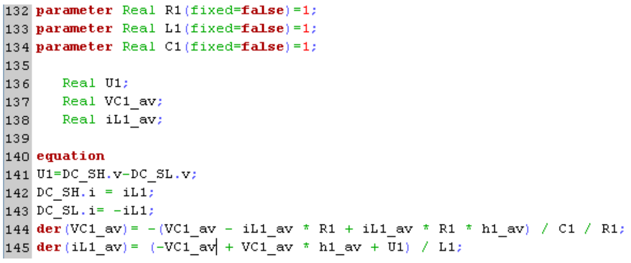


```
parameter Real R1(fixed=false)=1;
parameter Real L1(fixed=false)=1;
parameter Real C1(fixed=false)=1;

   Real U1;
   Real VC1;
   Real iL1;

equation
U1=DC_SH.v-DC_SL.v;
DC_SH.i = iL1;
DC_SL.i= -iL1;
der(VC1)= -(VC1 - iL1 * R1 + iL1 * R1 * h1) / C1 / R1;
der(iL1)= (-VC1 + VC1 * h1 + U1) / L1;
```

Figure 15: Modelica exact model of the boost converter



```
parameter Real R1(fixed=false)=1;
parameter Real L1(fixed=false)=1;
parameter Real C1(fixed=false)=1;

   Real U1;
   Real VC1_av;
   Real iL1_av;

equation
U1=DC_SH.v-DC_SL.v;
DC_SH.i = iL1;
DC_SL.i= -iL1;
der(VC1_av)= -(VC1_av - iL1_av * R1 + iL1_av * R1 * h1_av) / C1 / R1;
der(iL1_av)= (-VC1_av + VC1_av * h1_av + U1) / L1;
```

Figure 16: Modelica large signal model of the boost converter

In the case of the boost converter, the average model and the exact model are similar. As it has explained above, the difference is the model of the control signal. For the exact model, the control signal is a square wave. For the average model, the control signal is the average value of the square wave control signal, i.e. the duty cycle on every switching period (α) (see Figure 17).

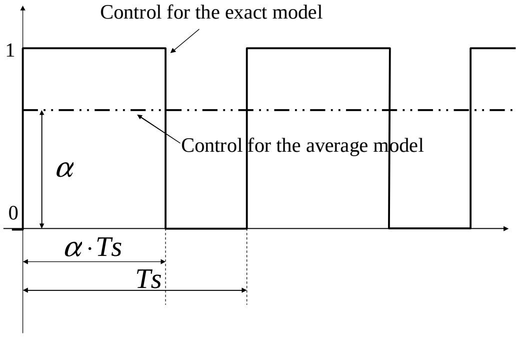


Figure 17: Control signal according to the model

6.3 Average model of the static converter

Numerical results for both of them are compared using AMESim.Lab. Figure 18 and 20 show these comparisons

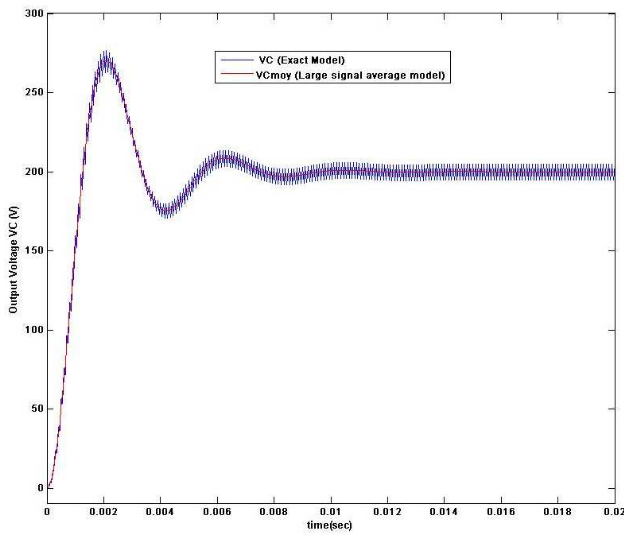


Figure 18: Capacitor Voltage [V] general overview

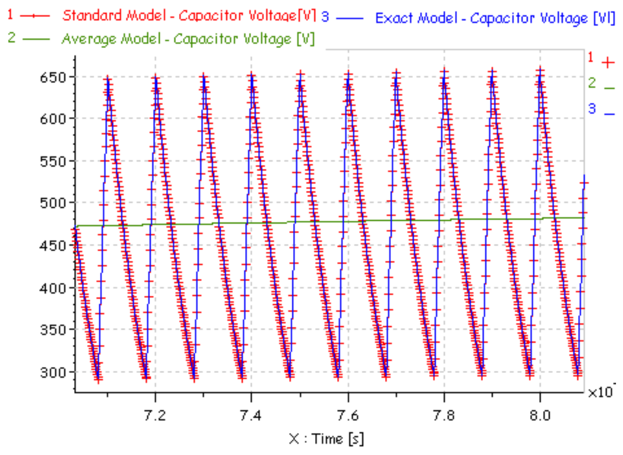


Figure 19: Capacitor Voltage [V] comparisons average/exact/standards models

The following array emphasizes the CPU time needed to perform the complete simulation for each kind of representation (see tables 1 and 2)

Parameter	**Value**
Time simulation	5 s
PWM frequency	10kHz
R	10 Ohm
L	0.001 H
C	1e-5 F
E	100 V
Duty cycle	0.8

Table 1: Parameters Value

Model	**CPU Time**
Standard representation	6.64s
Exact model	5.5s
Average model	0.016s

Table 2: CPU time

7 Advantages and drawbacks

Average models of static converter do not represent the commutations, so the simulation is faster. So, average models fit very well to system simulation or real time simulation.

The exact model represents switches by a short-circuit at on state and an open circuit at off state. Such a modeling that does not introduce parasitic time constants like the classical modeling with a binary resistance. For this last modeling, every switch is represented by a resistor which resistance is high at off state and low at on state.

However, for both average models and exact models, an a priori has to be carried out on the behavior of the studied static converter. This means that a change of the control mode may induce a new generation of a new dedicated model.

The use of Modelica as non causal modeling language allows achieving the model generation of power electronics converters by not being constrained by the causality. This aspect is very important and allows to deals with small power electronics structures as well as with large power electronics structures.

The use of Modelica also allows describing and generating the model of the static converter independently of the Modelica simulation environment.

In order to help the engineer to select the appropriate representation, we will also provide some guideline to select the model to be used.

8 Conclusions

In this paper, a methodology to get a fully functional Modelica code for a average model of power electronics converter has been presented. Its implementation within software has been briefly presented. Numerical tests for this code have also been introduced using the Imagine.Lab Modelica Platform. The efficiency of the proposed method has been discussed as well as some criteria to select the right appropriated sub-model according to the needs of engineer. A Modelica library of static converter controls (e.g. PWM), will be developed in future works as well as a guidelines for the modeling in discontinuous mode.

References

[1] http://www.lmsintl.com/imagine-amesim-1-d-multi-domain-system-simulation

[2] R.D. Middlebrook, S Cuk, "A general unified approach to modelling switching converter stages", IEEE Power Specialists Conf., pp 18-34, 1976.

[3] P. R. K. Chetty, "Current injected equivalent circuit approach to modelling and analysis of current programmed switching DC-to-DC converters (discontinuous inductor conduction mode)", IEEE Trans. on Industry Applications, Vol. IA-18, N°3, pp 295-299, 1982.

[4] S.R. Sanders, J. M. Noworolski, X. Z. Liu, G. Varghese, "Generalized averaging method for power conversion circuits", IEEE Power Electron. Specialists Conf. (PESC), Records 1990, pp 333-340.

[5] S. Bacha, M. Brunello, A. Hassan, "A general large signal model for DC-DC symmetric switching converters" Electric Machines and Power Systems, Vol 22, N° 4, July 1994 ; pp 493-510.

[6] Richard M.bass and Jian Sun, "Using symbolic computation for power electronics".Computers in Power Electronics, 1998. 6th Workshop, 19-22 July 1998, pp I IV.

[7] F.Verdiere, S.Bacha, L.Gerbaud, "Automatic modelling of static converter averaged models". *EPE* 2003- Toulouse .pp.1-9.

[8] Piero G.Maranesi, Marco Riva. "Automatic modelling of PWM DC-DC converters". IEEE Power Electron. Letters, Vol.1,N°4, December 2003.

[9] L. Allain, L. Gerbaud, C. Van Der Schaeghe, "Capitalisation and treatment of models for the optimisation of electric drives", in Optimisation and Inverse Problems in Electromagnetism, edited by M. Rudnicki and S. Wiak, Kluwer Academic Publishers, 2003, ISBN 1-4020-1506-2, pp 205-212 (8 pages)