A double resonance generator simulation using a hybrid approach

M. Zainea(*), E. Godoy(**), H. Cormerais(*), J. Buisson (*), H. Guéguen(*)

(*) SUPELEC/IETR
Avenue de la Boulaie
Cesson-Sévigné, France
Tel.: +33 / (0) 2 99 84 45 48
Fax: +33 / (0) 2 99 84 45 99
E-Mail: marius.zainea@supelec.fr
URL: http://www.rennes.supelec.fr/ren/rd/ash/

(**) SUPELEC
Plateau du Moulon
3, Rue Joliot-Curie
Gif-sur-Yvette, France
Tel.: +33 / (0) 1 69 85 13 73
Fax: +33 / (0) 1 69 85 13 89

Keywords
Hybrid systems, switching systems, bond graph, Simulink

Abstract
By using a hybrid approach the authors present the modelling of a double resonant DC/DC converter. This kind of generator is used for medical imagery applications, requiring high tensions, typically 40 kV to 200 kV, associated with high powers (about 100 kW). The proposed methodology can constitute an effective method allowing a structured and systematic approach of the dynamic systems modelling.

Introduction
An important and significant class of hybrid systems, also called switching systems, is represented by physical systems with switches. In the electrical domain, switching systems are encountered mainly in power electronics, where the commutating elements are the diodes, the thyristors, IGBTs, etc. Thus, in power electronics applications, systems are operating in commutation, mixing the continuous and the discrete behaviors. The use of hybrid techniques [3] can then constitute an effective method allowing a structured and systematic approach for the modeling, analysis and control.

The objective of this paper is to simulate, using a hybrid approach, the behavior of a double resonance generator (fig. 1). This kind of generator is used for medical imagery applications, requiring high tensions, typically 40 kV to 200 kV, associated with high powers (about 100 kW).

The proposed method in this paper is a general method, based on the bond graph representation [1], to obtain a simulation model with Simulink in the event tracking method framework [9]. The simulation is designed as an enumeration of dynamics associated with each configuration of switches, while a Stateflow block ensures the commutation between configurations. The whole system has to respect the necessary conditions for consistency.

In the Proceedings of 11th European Conference on Power Electronics and Applications, Dresden 2005
The paper is organized as follows: in a first part the mathematical model deduction is discussed, from the bond graph representation to the dynamics equations, jump functions, and switches model; in the second part the realization aspects are treated, a special attention being accorded to the Simulink representation of the dynamics and of the necessary conditions for consistency.

**Mathematical model for the continuous part by an energy based approach**

A general bond graph with switching components can always be represented by the diagram showed in fig. 2a. Four fields model the components behavior, three that belong to the standard bond graph formalism: - source field which produces energy (and will be considered as constant), - R field which dissipates it, - I and C field which can store it, and the Sw field that is added to model switching components [1], [10]. As in standard bond graph, the structure of the system is then modeled by the junction field, which dispatches the power. The convention for the direction of power is shown fig. 2a, i.e. from the sources (Se and Sf) and the switches (Sw) towards the junction structure and from the junction structure towards the storage elements (I – inductive, or C – capacitive) and the dissipative elements (R).

In this paper switches are considered as ideal. That means that they commutate instantaneously and have two discrete states, the first one denoted E where their effort is null and the second one, denoted F, where their flow is null. Then, from an energy point of view, the power on the bond connected to a switch is always null when there is no commutation. In the electrical field, the two logical states correspond to closed (E) and opened (F).
Remark. As bond graph was conceived as a formalism that can model systems from different application domains, the variables used are generic: power variables (effort and flow) and energy variables (displacement and momentum). In the electrical case, these variables correspond to voltage (effort) and current (flow), and to flux (momentum) and charge (displacement). In this paper, when talking about bond graph concepts, the generic variable terminology is used, and when talking about the application, the electrical variables are employed.

Equations derivation

To derive the equations from the fig. 2a representation, the bond graph approach uses the concept of computational causality. A procedure named SCAP [8] is used to establish the causal bond graph. From the causality point of view, ideal switches can be modeled by zero flow sources when they are in state E and by zero effort sources when they are in state F. When switches commutate, the causality of the corresponding sources changes and it is necessary to extend this change of causality to other bonds. Some energy storage elements can lose or recover the integral causality and the state equation is changed.

For any allowed configuration of the switches, i.e. configurations that respect consistency laws such as the Kirchhoff’s laws for electrical systems, causality can be assigned using SCAP or, if necessary, a modified SCAP in order to avoid unity gain causal loop [6]. Fig. 2b represents the block diagram that is deduced from the causal bond graph.

The following key variables are used: - the state vector $x_I$ is composed of the energy variables in integral causality (the momentum $p = \int f dt$ on I elements and displacement $q = \int e dt$ on C elements), and the complementary state vector $Z$ is composed of power variables (the efforts $e$ on C elements and flows $f$ on I elements); - the vector called semi-state vector $x_d$ is composed of the energy variables in derivative causality, ($p$ on I elements and $q$ on C elements), and the complementary state vector $Z_d$ is composed of power variables ($e$ on C elements and $f$ on I elements); - $D_I$ and $D_O$ represent the variables going out of and into the R field; - the vector $U$ is composed of the sources; - $T_i \in \mathbb{R}^{n_w}$ is composed of the variables imposed by the switches in this configuration according to their state; - $T_o \in \mathbb{R}^{n_o}$ is composed of the complementary variables in the switches (the effort for the switches which are in state F and the flow for the switches which are in state E).

![Causal bond graph of the double resonance generator](image)

Fig. 3: The causal bond graph of the double resonance generator.

In the Proceedings of *11th European Conference on Power Electronics and Applications*, Dresden 2005
The causal bond graph of the double resonance generator is shown in fig. 3. There are eight storage elements in integral causality and one in derivative causality, thus the state vector is:

$$\mathbf{x}_i = (p_2 \ p_6 \ p_{10} \ q_{11} \ p_{12} \ q_{17} \ q_{18} \ q_{19})^T, \mathbf{x}_d = p_9,$$

where the indices of the state variables above correspond to the indices given to the bonds of fig. 3.

As there is no unity gain causal loop, each output of the junction structure (\(\mathbf{x}_d\), \(Z_d\), \(D_i\) and \(T_i\)) can be expressed as function of all its inputs (\(\mathbf{x}_i\), \(Z_i\), \(D_i\) and \(T_i\)) (cf. fig. 2b). This linear relation can be written as an implicit differential equation that is called in the following the standard implicit form:

$$
\begin{pmatrix}
1 & -S_{12} \\
0 & 0 \\
0 & 0 \\
0 & -S_{24}^T
\end{pmatrix}
\begin{pmatrix}
\dot{x}_i \\
\dot{x}_d
\end{pmatrix}
=
\begin{pmatrix}
S_{11} & 0 & S_{13} & 0 & S_{14} & 0 & S_{15} \\
-S_{12} & -1 & 0 & 0 & S_{24} & 0 & S_{25} \\
-S_{13} & 0 & S_{33} & -1 & S_{34} & 0 & S_{35} \\
S_{14} & 0 & S_{34} & 0 & S_{44} & -1 & S_{45}
\end{pmatrix}
\begin{pmatrix}
Z_i \\
Z_d \\
D_i \\
D_o \\
T_i \\
T_o \\
U
\end{pmatrix}
$$

where \(I\) and \(0\) denote structural identity and zero matrices, and the other elements are composed of \(0\), and coefficients of the gyrators and transformers which are supposed to be constant in the following. The first row of this relation defines \(\dot{x}_i\), the second one \(\dot{Z}_d\), the third one \(\dot{D}_i\) and the last one \(\dot{T}_i\). Matrices \(S_{11}\), \(S_{33}\) and \(S_{44}\) are skew symmetric. Those properties are due to energy properties (there is no power stored or generated in the junction structure). There is no relation between \(\dot{Z}_d\) and \(\dot{D}_i\) (or \(\dot{D}_o\) and \(\dot{T}_i\)). If it were the case, we could inverse a causal path between an element in derivative causality and a resistor port and give to this element the integral causality. For the same reason, there is no relation between \(\dot{Z}_d\) and \(Z_d\) [2],[8].

**Reference Configuration**

In [2] an algebraic criterion is proposed to determine all allowed configurations and for each allowed configuration an equation such as (2) can be derived. Some configurations of the switches maximize the number of storage elements in integral causality. For all of them, \(S_{24} = 0\), which means that there is no causal relation between a switch and an element in derivative causality [2]. One of these configurations, associated with one arbitrary basic set of elements in derivative causality, will be called the reference configuration. Variables corresponding to this mode will be denoted by the subscript \(r\).

For the bond graph of the double resonance generator presented in fig. 3 the causality has been chosen such as to obtain a reference configuration. In the case of the double resonance generator, this reference configuration is unique: (E,E,F,F), where the order of switches in the previous sequence is the ascending order of bond numbers.

Each other configuration \(j\) can be defined by the switches whose state has changed with respect to the reference configuration. In the following, subscript \(j\) is added to all the variables or matrices referring to \(j\) configuration. When analyzing a configuration of switches, the causality of some storage elements may have changed with regard to the reference configuration and some of these storage elements can have lost the integral causality. It is shown in [2] that for all configurations it is possible to find a causal path such as a storage element that have a derivative causality in this reference configuration have also a derivative causality in the new one. Thus, \(\mathbf{x}_{ij}\) can be split into two parts, namely \(\mathbf{x}_{ij}\) and \(\mathbf{x}_{idj}\), which respectively denote the variables that remain in the state vector after the commutation, and the ones that come into the

In the Proceedings of *11th European Conference on Power Electronics and Applications*, Dresden 2005
semi-state vector. In a similar fashion $Z_{ir}$ is divided into $Z_{ij}$ and $Z_{idj}$. So, there exist 2 matrices $P_{ij}$ and $P_{2j}$ such that:

$$
\begin{bmatrix}
  x_{ir} \\
  x_{dr}
\end{bmatrix} =
\begin{bmatrix}
  P_{ij} & 0 \\
  0 & 1
\end{bmatrix}
\begin{bmatrix}
  x_{ij} \\
  x_{idj}
\end{bmatrix}
$$

$$
\begin{bmatrix}
  P_{ij} & P_{2j}
\end{bmatrix}
$$

is a permutation matrix whose dimension equals the number of energy storage elements in integral causality in the reference configuration. $P_{ij}$ specifies which of these elements keep their integral causality and $P_{2j}$ specifies which of them lose their integral causality. It follows that, the dimension as well as the composition of state vectors $x_{ij}$ and $x_{idj}$ changes according to the configuration.

**Remark.** All matrices used in the following of the paper, except the matrices referring to the standard implicit form associated to the reference configuration, are formally deduced by algebraic manipulation with Mathematica [4].

**Dynamic equations and jump functions**

For all allowed configurations, in the case where all circuit components are linear, a state equation can be determined. In the general case, this state equation is expressed as an implicit differential equation:

$$
E \dot{x}_j = \tilde{A}_j x_j + B_j U ,
$$

equation deduced from the standard implicit form, where $x_j = (x_{ij}, x_{idj})^T$. There has been demonstrated [2] that in the case of physical linear switched systems, this implicit form can be transformed via the generic variable change

$$
Q_j x_{ij} = (x_{ij}, x_{idj})^T
$$

in an explicit one:

$$
\begin{align*}
\dot{x}_{1j} &= A_j x_{1j} + B_j U \\
\dot{x}_{2j} &= D_j U
\end{align*}
$$

where $x_{1j}$ represents the explicit state and $x_{2j}$ represents the algebraic constraints. In the case of the double resonance generator, $D_j = 0$ for all allowed $j$ configurations. This result is due to the structural property that there are neither parallel connections between capacitors and voltage sources, neither serial connections between inductances and current sources in the electrical scheme of the double resonance generator.

Additionally, integrating (5) at commutation time and reversing the variable change, the jump function can be computed as:

$$
\begin{align*}
x_{1j}^+ &= K_{ij} x_{1j}^- \\
x_{2j}^+ &= D_j U = 0
\end{align*}
$$

where $x_{1j}^+$ and $x_{2j}^+$ represent the value of the explicit state after the commutation towards the $j$ configuration and $x_{1j}^-$ represents the value of the implicit state of the reference configuration before the commutation. From (6) one can deduce that the jump function does not depend on the source configuration of the transition, but only of the target configuration.

**Model for the switches**

Previously in this paper, the model given for the switches was functional, only the discrete state of the switch being significant. To obtain a more complete model, it is mandatory to take into account a more precise model of the switches and of their commutation laws.

In the Proceedings of *11th European Conference on Power Electronics and Applications*, Dresden 2005
For each basic switch of the double resonance generator (diode, IGBT) it is possible to define an elementary automata with the two discrete states E and F and the commutation laws, as in figure 4a) and 4b). In figure 4c), an equivalent switch model for the parallel association of a diode and an IGBT is given. This equivalent model can be deduced from the basic switch models (fig 4a and 4b). Obtaining such equivalent switch models is important as it reduces the complexity of the global model: the maximum number of configuration is $2^n$, where $n$ is the total number of switches. Thus, in the case of the double resonance generator, the number of allowed configurations passes from 32 to 16, when considering the equivalent switch model.

For each basic switch of the double resonance generator (diode, IGBT) it is possible to define an elementary automata with the two discrete states E and F and the commutation laws, as in figure 4a) and 4b). In figure 4c), an equivalent switch model for the parallel association of a diode and an IGBT is given. This equivalent model can be deduced from the basic switch models (fig 4a and 4b). Obtaining such equivalent switch models is important as it reduces the complexity of the global model: the maximum number of configuration is $2^n$, where $n$ is the total number of switches. Thus, in the case of the double resonance generator, the number of allowed configurations passes from 32 to 16, when considering the equivalent switch model.

\[
\begin{align*}
\text{E} & : [v \geq 0]
\begin{cases}
  e = 0 & \lor & u = 0 \\
  e = 1 & \land & u \leq 0
\end{cases} \\
\text{F} & : [u \geq 0]
\begin{cases}
  e = 0 & \lor & i \leq 0 \\
  e = 1 & \land & i \geq 0
\end{cases}
\end{align*}
\]

\[
\begin{align*}
\text{E} & : [i \geq 0]
\begin{cases}
  e = 0 & \lor & u \geq 0 \\
  e = 1 & \land & u \leq 0
\end{cases} \\
\text{F} & : [e = 0]
\begin{cases}
  u = 0 & \lor & i \geq 0 \\
  e = 1 & \land & i \leq 0
\end{cases}
\end{align*}
\]

Fig. 4: Automata like models for the switches used for the double resonance generator, where E stands for effort source (closed) and F stands for flow source (opened). The automaton a) represents the model for the T1 and T2 switches, the automaton b) represents the diodes model, and automaton c) represents the model for the SWT1 and SWT2 unions of switches.

The laws of commutation are function of a continuous variable associated to the switch (voltage or current, depending on the discrete state) and, eventually, a logical input. Additionally, a constraint imposed to the continuous variable may be associated to a discrete state of the switch. E.g. in the diode case, if the discrete state is E (closed), then the current can only be positive. As the conditions present in the automata model are function of the power variables, then it is necessary to find the equivalent relation function of the state vector and of the sources vector. Thus, given an allowed configuration of switches $j$, the expression of the variable vector imposed by the switches to the junction structure can be put in the following form [3]:

\[
T_{ij} = A_{ij}x_{1j} + A_{ij}U
\]

**Simulation model**

The simulation of the double resonance generator is obtained using an event tracking approach, which is a typical hybrid systems simulation method. The event tracking method combines the continuous evolution of the state system through the active configuration dynamics and, as the name states it, the event detection. When an event is detected the system changes the active configuration and resets the continuous state according to the jump function. There are two types of events: external and internal. External events occur when external controls are taken, in conjunction with an additional condition or not. For the double resonance generator, external events are the controlled opening and the closing of the IGBT or, if the equivalent switch model is considered, the controlled opening and the controlled closing of the equivalent switch. The problem of detecting this type of events is simple, as the time moment when such an event takes place is totally predictable. Internal events are due to the state evolution, as the state acts on $T_o$. Thus, when the state reaches a certain value a commutation is fired. For the double resonance generator, internal events can cause one or more diodes to commutate, the natural blocking of the IGBTs, etc. The detection of internal events is far more complicated and requires a special mechanism. A strong point in using Simulink in the context of the event tracking method is the built-in zero-crossing detection mechanism that can be used for solving the internal events detection problem.

The realization of the Simulink model is structured as in the following: the first part describes how to build a block for each configuration to simulate the dynamics and estimate the power variables imposed to...
the switches, while the simulation scheme for the 16 allowed configurations of the system is presented afterwards; in the second part the necessary conditions for a consistent simulation are formulated and the Simulink implementation for these conditions is given; finally, in the third part, the global simulation model is given along with some simulation results.

**Simulation of the continuous part**

Equation (5), which is an ODE where $x_{ij}$ is continuous at commutation time, can be used in order to simulate the dynamics for each configuration $j$. From $x_{1j}$ and $x_{2j}$, the state vector in the reference configuration can be reconstructed from (5) by reversing the base transformation and from (3):

$$X_r = \begin{pmatrix} P_1 & P_2 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_{1j} \\ x_{2j} \end{pmatrix} = K_{3j} x_{ij}$$

(8)

Therefore, the dynamics equation can be implemented using equations (5), as shown in fig. 5. Additionally, from (7), $T_{ij}$ can also be computed.

![Fig. 5: The Simulink implementation of the dynamics associated to one switches configuration](image)

For the entire system, the simulation of the continuous part is made by associating all the blocks that simulate the dynamics of all configurations (fig. 6). A decision block makes the selection between configurations dynamics. The **Decoding** block is implemented such as a 1 value is fed to the block corresponding to the active mode and a −1 to all other blocks. Thus, only one dynamics block is active at a time.

![Fig. 6: The simulation of the continuous part is done by associating all the blocks corresponding to each configuration dynamics equation.](image)

The Merge block is used as an output decision block, as it has the capability of selecting the last modified input. As only one dynamic block is active at a certain time instant, only its output changes at the current step. Such a representation makes possible to obtain in an automatic fashion a simulating model, where

In the Proceedings of *11th European Conference on Power Electronics and Applications*, Dresden 2005
the jump functions are hidden in the $K_1$ block and all the computation is function of the current mode junction matrices. Every time the selector changes the active mode, a new initial condition is computed as the image of $x_r$ through the jump function. The changes in the state vector dimension or the order inversion inside the state vector, due to the fact that some storage elements lose their integral causality, are encapsulated inside the dynamics block.

**Necessary conditions formulation and implementation**

For a simulation to be correct, it has to be consistent at all time with the constraints imposed by the models chosen for the switches [3]. That is, for all configurations, the inequalities associated to the discrete states of each switch must be satisfied. Thus, from (7), it can be deduced that at commutation time the following inequality must hold for a configuration $j$ to become active:

$$C_j^T T_{oj} = C_j^T \left(A_{ij} x_i^+ + A_{oj} U \right) = C_j^T \left(A_{ij} K_1 x_r^- + A_{oj} U \right) \geq 0,$$

(9)

where $C_j^T$ is obtained from a diagonal matrix, whose diagonal elements are $\pm 1$ if a constraint is active for the corresponding switch discrete state and zero otherwise, by eliminating the zero rows.

Depending on the state value before the commutation, at commutation time, the vector $T_{oj}$ may also have an impulsive component. The amplitude of the Dirac pulses at commutation time can be expressed as a function of the state vector and sources vector as:

$$T^j = G_{ij} x_r^- + G_{oj} U$$

(10)

and must respect the same inequality as $T_{oj}$:

$$C_j^T T^j \geq 0$$

(11)

Therefore, the implementation of (9) and (11) in a Simulink block diagram is represented in fig. 7. The **guard output** is a Boolean, whose logical value is **true** if the configuration $j$ can become active and **false** otherwise. Such blocks are necessary for all configurations in order to determine which configuration of switches can become active.

The global Simulink model and simulation results

The global Simulink model has four main blocks as it is shown in fig 8. Two blocks (**Dynamics** and **Test of the guard condition**) were already discussed in the previous subsections. The **Control** block manages

In the Proceedings of *11th European Conference on Power Electronics and Applications*, Dresden 2005
the external events generation and its architecture is specific to the desired control law. The Automaton block manages the active configuration choice and this function is implemented as a Statechart in a Stateflow block. The Statechart design is based on the switches model and on the necessary conditions for a consistent simulation. The disadvantage of this approach for the representation of the discrete evolution with Stateflow is the rapid illegibility with the number of switches, which is the main reason the chart is not showed here. However, the global simulation architecture is made such as the Automaton block and the Test of the guard condition are executed only when an event (external or internal) occurs.

Fig. 8: The global Simulink model of the double resonance converter.

The memory blocks are used to cut the algebraic loops that are formed, as \(x_r\) is fed to the input of the dynamic blocks and their output is a part of the Merge block decision process. A second algebraic loop is formed as \(T_o\) affects the \(loc\) value and \(loc\) decides on \(T_o\). The introduction of these memory blocks does not affect the global result because \(loc\) and \(x_r\) are synchronized and they can be restored to the real evolution in a post simulation manipulation.

Fig. 9: The results of simulation exemplified by the time evolution of \(I_{Lr}\), \(I_{lp}\) and \(U_t\).

In fig. 9, the time evolution of some physical variables (\(I_{Lr}\), \(I_{lp}\) and \(U_t\)) from the simulation is plotted. The control parameter involves the notion of time delay separating the instant defined by the zero crossing of
In the Proceedings of 11th European Conference on Power Electronics and Applications, Dresden 2005

the current $I_{Lr}$ in the resonant serial inductor at the turn-off instant of one IGBT - and the next turn-on instant of the complementary IGBT. The new controlled commutation instant is taken at the zero crossing of the current $I_{Lr}$ in the resonant serial inductor at the turn-off instant of the second IGBT [7]. These curves correspond to real evolution of the system described in [5].

Conclusions

The use of the hybrid techniques can constitute an effective method allowing a structured and systematic approach of the dynamic systems modeling, particularly in the case of switching systems mixing continuous and discrete behaviors. An important application domain concerns power electronics where this method allows making rigorous calculations without considering approximate arguments of static or quasi-static kind. The proposed methodology has been applied with good results on a double resonant DC/DC converter developed, in industrial collaboration, for medical imagery applications. To resume, the approach presented in this paper offers a systematic method for modeling switching linear physical systems in a Matlab/Simulink environment. The system bond graph is used as a starting point in order to determine each possible mode dynamics and the matrices used to recompose the reference configuration state vector. Afterwards, a scheme for Simulink implementation was proposed. Future work will concern in developing a more transparent and systematic method to describe the system discrete evolution and, in a second phase, the analysis and the study of the control laws of this system.

References