Symbolic Analysis Methods for Averaged Modeling of Switching Power Converters

Jian Sun, Member, IEEE, and Horst Grotstollen

Abstract—Symbolic analysis methods for averaged modeling of switching power converters are presented in this paper. A general averaging method suitable for computer-aided modeling is discussed first. Then, a symbolic analysis package that uses this averaging method to automatically generate an analytical averaged model for a switching power converter is described. The package is implemented using the computer algebra system Mathematica and can be used for modeling dc/dc power converters employing different switching techniques, including hard-switching pulse-width modulation (PWM), quasi-resonant soft switching, and soft transition. Several examples are provided to demonstrate the applications of the package. Further applications of symbolic analysis methods in power electronics are also discussed.

Index Terms—Averaging method, CAD, power converters modeling, symbolic analysis.

I. INTRODUCTION

AVERAGED modeling of switching power converters is important for understanding converter dynamic behavior and the design of a proper control loop. The development of averaging methods has been a topic of interest to the power electronic community for over two decades [1]. Recent research has focused on the extension of the classical state-space averaging method for hard-switching pulse-width modulation (PWM) converters to those employing various soft-switching and soft-transition techniques. These methods use similar mathematical principles and actually share a common theoretical basis that, however, has not been fully explored in the literature. As a consequence, many seemingly different averaging schemes exist nowadays, and the choice of a suitable modeling method is not an easy task for practical engineers.

On the other hand, modeling a power converter necessitates to derive many expressions that describe the converter operation. Such operation analysis is becoming more and more complicated due to the use of various resonant subcircuits in the converter for achieving soft switching or soft transition. Modeling a converter is therefore still a time-consuming process and requires skillful handling of complicated mathematical expressions, even when one is already familiar with the modeling method. This is especially the case when many different converter topologies are to be compared so as to identify an optimum one for a specific application. There are, thus, ever-increasing needs for computer-aided modeling tools.

This paper describes a symbolic analysis package for assisting in averaged modeling and analysis of switching power converters. Symbolic analysis of analog circuits is defined in the literature as a formal technique for calculating the behavior or characteristic of a circuit, with some or all circuit elements represented by symbols [2]. Since the late 1980’s, symbolic analysis gained a renewed and growing interest due to the success of modern symbolic analyzers. Compared to numerical simulation, symbolic circuit analysis will provide insight into circuit behavior and can generate an analytic model for circuit optimization and control design. Different symbolic simulators for analog circuits have been developed and are compared in [2].

Symbolic analysis of power electronic systems is an issue of recent research. An example of using MATLAB and its symbolic analysis toolbox for the generation of design curves for new converter topologies was described in [3]. In [4], symbolic analysis is applied to formulate basic circuit equations for PWM converters. As a result, symbolic transfer functions of the converter in each topological state are obtained. Numerical inverse Laplace transformation is then applied to calculate the time-domain responses.

Symbolic generation of state-space models of power converters using MAPLE was reported in [5]. In contrast to the frequency-domain approach used in [4], circuit equations in [5] are formulated directly in the time domain. They are then rearranged in a matrix form such that the classical state-space averaging method [6] can be applied to derive an averaged model. However, the package cannot handle diodes and switches. Therefore, to use it, the user has to provide an input file containing topological information of the converter in each topological state. The application of the package is also limited to hard-switching PWM converters due to the limitation of the implemented classical state-space averaging method.

The symbolic analysis package presented in this paper is based on a general averaging method such that it is applicable to converters employing different switching techniques, including hard-switching PWM, quasi-resonant soft switching, and soft transition. It is implemented using the computer algebra system Mathematica [7]. Diodes and switches can directly appear in the input file as circuit elements without their conducting states being specified. The paper is organized as follows. After an overview of the general averaging method...
in the next two sections, the formulation of circuit equations and their transformation into standard state-space form are discussed in Section IV. The kernel of the package, i.e., the symbolic generation of an averaged model, is described in Section V. Applications of the package are illustrated in Section VI by various examples, including a hard-switching PWM boost converter, a phase-controlled bridge rectifier, a zero-current-switched (ZCS) quasi-resonant buck converter, and a zero-voltage-transition PWM boost converter. Section VII concludes the paper and points out some further applications of symbolic analysis in power electronics.

II. A GENERAL AVERAGING METHOD

The general averaging method implemented in the package to be described is based on partitioning the converter circuit into a slow and fast subsystem [8]. The slow subsystem contains the main energy-storage elements of the converter as well as its input and output sections, while switching devices and the associated resonant tank (and the snubber circuit, when needed to be considered) comprise the fast subsystem (see Fig. 1). Suppose that the slow subsystem is described by

\[ \frac{dx_s}{dt} = f(x_s, u_s, \bar{v}_s) \]  

and the fast subsystem by

\[ \frac{dx_f}{dt} = g_k(x_f, v_f) \]  

where \( u_s \) is the input to the slow subsystem. The coupling between two subsystems is represented by

\[ v_f = p(x_s, u_s), \quad \bar{v}_s = q_k(x_f, v_f), \]  

Variables and functions in the above equations are vectors of proper dimension. The index \( i \) of functions \( g_i \) and \( q_k \) in (2) and (3) runs from one to \( N \), indicating that the fast subsystem contains switching devices and assumes \( N \) different topological states in each switching cycle.

With the converter circuit partitioned as such, an averaged model is derived in three steps.

1) Relaxation: The fast subsystem is first analyzed by assuming constant input \( v_f \) from the slow subsystem. The assumption is justified by the slow variation of \( v_f \). In this way, responses of the fast state variables over one switching cycle can be determined:

\[ \bar{\dot{x}}_f = h(v_f, t). \]  

2) Decoupling: The fast subsystem affects the operation of the slow subsystem through its output \( \bar{v}_s \). With (3) and (4), the relaxed solution of \( v_s \) can be written as

\[ \bar{v}_s = q_k(h(v_f, t), v_f) = z(v_f, t). \]  

The slow dynamics of the converter, which is the objective of averaged modeling, may now be analyzed based on (1), with \( v_s \) replaced by the relaxed solution (5). In this way, the fast subsystem, which is usually a dynamic network as it contains energy-storage elements, is replaced by an equivalent time-varying resistive network (see Fig. 2).

3) Averaging: Referring to (5), the time-varying input \( \bar{v}_s \) to the slow subsystem can be expanded using Fourier series as

\[ \bar{v}_s = \overline{z}(v_f) + \sum_{k=1}^{\infty} a_k \cos \left( \frac{2\pi k t}{T_s} \right) + \sum_{k=1}^{\infty} b_k \sin \left( \frac{2\pi k t}{T_s} \right) \]  

where

\[ \overline{z}(v_f) = \frac{1}{T} \int_0^T z(v_f, \tau) \, d\tau \]  

is the dc component of \( z(v_f, t) \) and \( T_s \) is the period of the converter switching cycle. Since the switching frequency \( f_s = 1/T_s \) is usually very high and the slow subsystem is insensitive to fast-varying inputs, the time-varying terms in (6) can be neglected such that only the dc component, \( \overline{z}(v_f) \), remains. A nonlinear, time-invariant-averaged model approximately describing the slow dynamics of the converter is therefore obtained by substituting \( v_s \) in (1), with \( \overline{z}(v_f) \) (see Fig. 3):

\[ \frac{dx_s}{dt} = f(x_s, u_s, \overline{z}(v_f)) = f(x_s, u_s, \overline{z}(p(x_s, u_s))) = f_0(x_s, u_s), \]  

III. STATE-SPACE AVERAGING

The foregoing discussion on averaged modeling based on circuit partition is intended to provide some physical insights into the method. Explicit circuit partition is, in fact, not necessary for practical applications since the same averaged model can be derived based on a state-space description of the converter circuit as a whole, as will be demonstrated below [8].
By substituting (3) into (1) and (2), we obtain a state-space model of the converter without intermediate coupling variables

\[
\frac{dx_s}{dt} = f_i(x_s, x_f, u_s) \tag{9}
\]
\[
\frac{dx_f}{dt} = g_i(x_s, x_f, u_s). \tag{10}
\]

Note that the right-hand sides of both (9) and (10) change when the converter circuit switches from one configuration to another. Evidently, the three steps discussed before for deriving an averaged model is equivalent to the following procedure.

1) Solve (10) for \(\tilde{x}_f = h(x_s, u_s, t)\) by treating the slow variables \(x_s\) as constant (relaxation).

2) Replace \(x_f\) in (9) with \(\tilde{x}_f\) (decoupling).

3) Average the right-hand side of (9) over one switching cycle (averaging).

Both approaches will yield the same averaged model. Thus, the method may also be termed state-space averaging, as the averaging can be carried out in state space. It is applicable to various dc/dc power converters using different switching techniques, including hard-switching PWM, quasi-resonant soft switching, and soft transition. In contrast, the classical state-space averaging method of Middelbrook and Cuk [6] is only applicable to hard-switching PWM converters. It is also quite straightforward to show that the classical state-space averaging is, in fact, a special case of the more general method discussed here.

However, the proposed averaging method is not applicable to all converters. Two basic limitations of the method are the need for an analytical solution of the relaxed fast subsystem and the assumption that this solution is periodic, with a period equal to one switching cycle. To obtain an analytical solution, the fast subsystem may not include any nonlinear element except ideal switches and diodes. The fast subsystem should also be kept as simple as possible to facilitate computer analysis, which implies that most parasitic effects have to be neglected. On the other hand, the relaxed solution \(\tilde{x}_f\) will be periodic only when the fast subsystem doesn’t contain continuous energy-storage elements [9]. Otherwise, the fast subsystem exhibits nontrivial low-frequency dynamics and needs to be replaced by a dynamic network rather than a resistive one.

The necessity to assume a periodic solution of the relaxed fast subsystem can also be understood in the frequency domain. A nonperiodic solution will have components at frequencies lower than the switching frequency, whose effects on the slow subsystem cannot be neglected. Since averaging the fast variable over one switching period effectively retains the dc component, significant error will be introduced into the final averaged model. This assumption precludes the application of the method to load-resonant converters [such as series-resonant converters (SRC’s) and parallel-resonant converters (PRC’s)] in which the fast variables exhibit continuous resonance. An averaging method for load-resonant converters, based on slowly varying amplitude and phase transformation, has been developed in [10] and [11].

Considering the first limitation of the method mentioned above, the following discussions will focus on converters using linear components and ideal switching devices. Such a converter can be treated as a piecewise linear system and can be described by a linear time-invariant model in each topological state as

\[
x' = A_i x + B_i u \tag{11}
\]

where \(x\) is the vector of state variables (inductor currents and capacitor voltages) and \(u\) is the independent source. The index \(i\) runs through one to \(N\) for a converter having \(N\) possible topological states. The task of the package to be described below is to first generate a state-space model for the converter in each topological state and then derive an averaged model using the modeling procedure outlined above. Details are presented in the next two sections.

IV. STATE-SPACE MODEL GENERATION

The state-space model of a converter is generated using three major self-defined functions: NodalAnalysis, StateSpace, and StandardForm, which are defined in a file named NodAI.m, as illustrated by the program flowchart in Fig. 4.

A. Intermediate Circuit Equations

A set of intermediate circuit equations is first generated using the function NodalAnalysis, which is based on a method similar to the modified nodal-analysis technique. The input netlist to be provided by the user for a converter circuit
contains a description of each circuit branch, including the nodes $n_{k1}$ and $n_{k2}$ to which it is connected, the type of the element, and a symbol representing that element. For example, the netlist for the hard-switching PWM boost converter shown in Fig. 5 is (refer to [7] for the definition of and possible operation on lists with Mathematica)

\[
\text{NetList} = \{\text{VoltageSource, \{1, 0\}, } V_i \} \\quad \{\text{Inductor, \{1, 2\}, } L \} \\quad \{\text{Switch, \{2, 0\}, } S \} \\quad \{\text{Switch, \{2, 3\}, } D \} \\quad \{\text{Capacitor, \{4, 0\}, } C \} \\quad \{\text{Resistor, \{3, 4\}, } r \} \\quad \{\text{Resistor, \{3, 0\}, } R \} \}.
\]

For a circuit having $n_0$ nontrivial nodes and $l_0$ branches, NodalAnalysis first generates $n_0$ equations relating the branch currents using Kirchoff’s current law. For each branch, $b_k$, connected between nodes $k_1$ and $k_2$, a constitutive equation that relates the branch current to its node voltages is also formulated using the following rules:

1) voltage source $V_i$:
\[ v_{k_1}(t) - v_{k_2}(t) = V_i \]

2) resistor $R$:
\[ v_{k_1}(t) - v_{k_2}(t) = i_k(t) \cdot R \]

3) inductor $L$:
\[ v_{k_1}(t) - v_{k_2}(t) = L \frac{di_k(t)}{dt} \]

4) capacitor $C$:
\[ C \left[ \frac{dv_{k_1}(t)}{dt} - \frac{dv_{k_2}(t)}{dt} \right] = i_k(t) \]

5) switch $S$ ($S = 1$ for on and zero for off):
\[ (1-S)i_k(t) + S[v_{k_1}(t) - v_{k_2}(t)] = 0. \] (12)

As a result, applying NodalAnalysis will yield totally $n_0 + l_0$ equations in which the $n_0$ node voltages and $l_0$ branch currents are unknown variables. Since switches can be directly handled by the program, the netlist is very compact and is easy to prepare.

B. Equations in State Variables

With the generated circuit equations, state-space description of the converter in each topological state is then obtained by using the function StateSpace. To do this, the program first identifies all inductor currents and capacitor voltages and retains them as state variables. All other branch currents and node voltages are then eliminated from the circuit equations by using the built-in Mathematica function Eliminate. This yields a system of $n_t$ equations in state variables, $n_t$ being the total number of reactive circuit elements (inductors and capacitors).

The topological states of the circuit are specified by the list TopoStates that is given in such an order as they become activated within a switching cycle. For example, for the hard-switching PWM boost converter shown in Fig. 5, this list is

\[
\text{TopoStates} = \{\{S \to 1, D \to 0\}, \{S \to 0, D \to 1\}\}.
\]

A special problem to be noticed here is the degradation of the circuit model due to the existence of inductor cutsets and/or capacitor loops in certain topological states. In such cases, not all inductor currents or capacitor voltages are justified as state variables, since some are dependent of others. However, to facilitate averaged modeling, it is advantageous to retain all inductor currents and capacitor voltages as state variables such that the state variable vector and dimension of the state-space model in each topological state remains unchanged. But if any inductor cutset or capacitor loop exists, direct elimination of all circuit variables, except inductor currents and capacitor voltages, will result in a system of equations in which some contain no derivative terms of any inductor current or capacitor voltage. The computer will then have difficulties in rearranging these equations into the standard form of a state-space model.

For example, consider the zero-voltage-transition (ZVT) PWM boost converter shown in Fig. 6 [17]. With $S = 0, D = 1, S_1 = 1$, and $D_1 = 0$, the resonant capacitor $C_r$ is directly in parallel with the output capacitor $C_0$. Direct elimination of all circuit variables, except the two inductor currents ($i_q$ and $i_r$) and two capacitor voltages ($v_0$ and $v_r$), will yield equations
\[
L_r \frac{di_r}{dt} = v_r, \quad v_r = v_0
\]
\[
i_0 = i_r + \frac{v_r}{R} + C_r \frac{dv_r}{dt} + C_0 \frac{dv_0}{dt}
\]
\[
V_i = R_0 + v_r + L_0 \frac{dv_0}{dt}
\]
which are insufficient for the computer to solve for the four derivative terms. This problem has been dealt with in the package by instructing the computer to derive new equations from those that do not involve derivatives of any state variables. For the example ZVT PWM boost converter, this would be to add the following new equation to the equation system given before:
\[
\frac{dv_r}{dt} = \frac{dv_0}{dt}.
\]

1The system matrix $A$ is singular in this case, and the model might be better termed as a pseudo-state-space model. For this reason, algorithms for deriving true state-space models of circuits containing inductor cutsets or capacitor loops [12]–[14] cannot be applied here.
C. State-Space Model in Matrix Form

To derive a state-space model in the matrix form (11) for a converter in each topological state, the corresponding equations in state variables are solved for the derivatives of all state variables by using the built-in Mathematica function Solve. As the converter circuit represents a linear time-invariant system in each topological state, each solution is linear in both state variables and inputs, such that it can be further put into the standard matrix form (11). This is accomplished by the function StandardForm in the developed package (see Fig. 4).

For the example ZVT PWM boost converter with \( S = 0 \), \( D = 1 \), \( S_1 = 1 \), and \( D_1 = 0 \), the generated state-space model in the standard matrix form is

\[
\begin{bmatrix}
\frac{\partial i_0}{\partial t} \\
\frac{\partial i_r}{\partial t} \\
\frac{\partial v_r}{\partial t}
\end{bmatrix} =
\begin{bmatrix}
-\frac{1}{L_0} & 0 & -\frac{1}{L_r} & 0 \\
0 & 0 & \frac{1}{L_r} & 0 \\
\frac{1}{C_s} & -\frac{1}{C_s} & \frac{1}{C_s}R & 0 \\
0 & \frac{1}{C_s} & -\frac{1}{C_s}R & 0
\end{bmatrix}
\begin{bmatrix}
i_0 \\
i_r \\
v_r \\
v_0
\end{bmatrix} +
\begin{bmatrix}
\frac{1}{L_0} \\
0 \\
0 \\
0
\end{bmatrix} V_i.
\]

V. AVERAGED MODEL GENERATION

Besides the piecewise linear model (11), the package needs some additional inputs to derive the averaged model. These additional inputs are provided by the lists SlowFast, BC’s, and IV.

SlowFast = \{SlowVarIndex, FastVarIndex\} tells the computer which state variables are slow variables (SlowVarIndex) and which are fast (FastVarIndex). Both SlowVarIndex and FastVarIndex are, again, lists of the sequence numbers of the slow and fast variables as they appear in the state variable vector \( \mathbf{x} \). See the next section for examples.

The list BC’s specifies the boundary conditions under which the circuit changes its topological state (switching). It is given for a switching of the converter under normal-operation condition under which an averaged model is to be generated. Three different types of boundary conditions can be implemented.

1) Absolute Time: The switching takes place at a fixed time measured from the beginning of a switching cycle. In this case, the corresponding element in BC’s is a number, a symbol, or an expression representing the time. For example, the first and last elements of BC’s are usually zero and \( T_s \), respectively, indicating the beginning and end of a switching cycle, \( T_s \) being the switching period of the converter.

2) Relative Time: The next switching is delayed by a certain time from the previous one. The corresponding element in BC’s is a list of the single element representing the delay time. For example, for hard-switching PWM converters in continuous conduction mode and with duty-cycle control, BC’s can be written either as

\[
\text{BCs} = \{0, d \times T_s, (1 - d) \times T_s\}
\]

where zero and \( d \times T_s \) are absolute times and \( ((1 - d) \times T_s) \) represents a relative time or

\[
\text{BCs} = \{0, d \times T_s, T_s\}
\]

where all times are absolute, i.e., measured from the beginning of the switching cycle.

3) Implicit Conditions: This is the case when switching is dependent of state variables. For example, the conduction of a diode depends on the polarity of the voltage across it. In this case, the corresponding element in BC’s is a list of two expressions, \{exp1, exp2\}, and the switching instant is to be solved from the expression \( \text{exp1} = \text{exp2} \). Both expressions may be functions of slow variables, and at least one is dependent of the fast variables. The solution is a relative time, i.e., it represents the delay from the previous switching.

Besides SlowFast and BC’s, it is also necessary to specify the initial values of the fast state variables at the beginning of the switching cycle (by list IV). Each initial value can be given as either a number, a symbol, or a function dependent of slow variables and converter inputs. With these additional inputs and the state-space models in all topological states, an averaged model is derived using three functions: relaxation, decoupling, and averaging, which directly resemble the three modeling steps outlined in Section III. Some details of the implementation are discussed below.

A. Relaxation

According to the list SlowFast, the whole state variable vector \( \mathbf{x} \) is first partitioned into a slow-variable vector SlowVars and a fast-variable vector FastVars as follows, shown at the bottom of the page. In each topological state, a

SlowVars = Table[x[[SlowFast[[i, 1]]]], {i, 1, Length[SlowFast[[1]]]]},
FastVars = Table[x[[SlowFast[[2, i]]]], {i, 1, Length[SlowFast[[2]]]]}.

---

Fig. 6. Circuit diagram of the ZVT PWM boost converter.
system of ordinary differential equations describing the fast subsystem of the converter is then generated by removing from (11) the equations for slow variables (corresponding to SlowVarIndex). Since the time variable \( t \) in SlowVars has been replaced by \( t_0 \) (see the definition of SlowVars at the bottom of the previous page), slow variables are automatically treated as constants in these equations. Together with the given initial values expressed by IV, the response of the fast subsystem in the present topological state is calculated by solving these equations analytically using the built-in Mathematica function DSolve.

The termination of the present topological state and the transition to the next configuration are determined by using BC’s. In case of implicit conditions, an algebraic equation will be formulated and is solved using the built-in function Solve. Evaluating the solution of fast state variables at this timepoint yields the initial values for the next switching interval. The program proceeds then to the next topological state.

Since the fast subsystem responses often involve trigonometric and, possibly, other nonlinear functions, as is the case with quasi-resonant converters (QRC’s) and soft-transition converters, the algebraic equation defining an implicit switching instant may have multiple solutions among which only one is physically meaningful. It may also happen that the computer cannot find an analytical solution at all if the equation is not properly formulated. To tackle these problems, novel methods have been used in the package.

Solving the differential equations for the fast subsystem is not a very difficult task for Mathematica due to the linear nature of the equations. However, the solution may become very lengthy which, when not properly simplified, will cause difficulties for the computer in determining the switching instant and, later on, also in averaging the slow subsystem. Special efforts are therefore made in implementing the function to keep the solutions as compact as possible.

B. Decoupling and Averaging

The function Decoupling uses the fast-variable solutions and the matrices \( A_2 \) and \( B_1 \) to generate a decoupled model for the slow subsystem in each topological state, as outlined in Section III. An averaged model is finally derived by integrating the decoupled model over one switching cycle. This is accomplished by using the function Averaging. The implementation of both functions is quite straightforward by using several built-in Mathematica functions. Details are thus omitted here.

VI. APPLICATION EXAMPLES

Due to the broad application range of the general averaging method implemented, the package can be used to model dc/dc power converters using different switching techniques, such as hard-switching PWM, quasi-resonant soft switching, and soft transition. Some examples are given below to demonstrate the applications of the package. Simulation results will also be provided to verify the modeling results.

A. Hard-Switching PWM Boost Converter

The converter circuit has been given in Fig. 5. Since neither auxiliary resonant network has been used in the converter nor parasitic inductance and capacitance associated with the switch and the diode are considered, the fast subsystem consists merely of the switch and diode and is a (time-varying) static network, as indicated by the gray box in Fig. 5. Notice that a resistor, \( r \), has been put in series with the capacitor \( C_0 \) to represent its power losses.

The netlist has been given before in Section IV. Other input lists necessary for deriving the averaged model are

\[
\text{TopoStates} = \{ \{ S \rightarrow 1, D \rightarrow 0 \}, \{ S \rightarrow 0, D \rightarrow 1 \} \}
\]

\[
\text{SlowFast} = \{ \{ 1, 2 \}, \{ \} \}
\]

\[
\text{BCs} = \{ d \ast Ts, Ts \}; \quad \text{IV} = \{ \}.
\]

Using these inputs, the following averaged model is generated by using the package without any further intervention from the user:

\[
\frac{d \omega}{dt} = \frac{r R (d - 1) \omega}{L (R + r)} + \frac{R (d - 1) \omega}{L (R + r)} + \frac{V_i}{L}
\]

\[
\frac{d \chi_i}{dt} = \frac{R (1 - d) \omega}{C (R + r)} - \frac{\nu_0}{C (R + r)}.
\]

As can be expected, the averaged model is the same as one would obtain through direct application of the classical state-space averaging method.

B. Phase-Controlled Bridge Rectifier

Fig. 7(a) shows a phase-controlled bridge rectifier with ac-side commutating reactance \( L_c \). The dc-side inductance is usually large for smoothing the load current \( i_0 \). The commutating inductance is, however, relatively small, and the current through it changes the polarity once every half-line period due to the commutation between the two pairs of thyristors, \( S_1 - S_4 \) and \( S_2 - S_3 \). Thus, the fast subsystem comprises, in this case, \( L_c \) and the four thyristors, as shown in Fig. 7(a).

We now use the package to derive an averaged model that describes the averaged behavior of the circuit at the dc side.
Referring to Fig. 7(a), the netlist of the circuit is

\[
\text{NetList} = \{\{\text{VoltageSource, \{1, 2\}, } v_i\}, \\{\text{Inductor, \{1, 3\}, } L_c\}, \\{\text{Switch, \{3, 4\}, } S_1\}, \\{\text{Switch, \{3, 0\}, } S_3\}, \\{\text{Switch, \{2, 4\}, } S_2\}, \\{\text{Switch, \{2, 0\}, } S_4\}, \\{\text{Inductor, \{4, 5\}, } L\}, \\{\text{Resistor, \{5, 0\}, } R\}\}.
\]

The state variable vector corresponding to this netlist is \(x = [i_c, i_0]^T\). Since \(i_c\) is a fast variable and \(i_0\) is a slow variable, we have \(\text{SlowFast} = \{(\{2\}, \{1\})\} \). Suppose that the ac-side input voltage is \(v_i(t) = V_s \sin \omega t\) and the firing angle is \(\alpha\) (see Fig. 8). Due to the symmetry of the converter operation, the averaged model can be derived by considering only the interval \(\omega t \in [\alpha, \pi + \alpha]\). The circuit undergoes two topological states in this interval: all thyristors conduct during the commutation interval \(\omega t \in [\alpha + \mu]\) and only \(S_1\) and \(S_4\) conduct when commutation is completed, which is characterized by \(i_c = i_0\). Notice also that \(i_c = -i_0\) before the commutation begins, i.e., at \(\omega t = \alpha\) (see the bottom of the page). With these inputs and by using the package, the required averaged model is derived to be

\[
L_0 \frac{di_0}{dt} = \frac{L_0}{L_0 + L_c} \left(\frac{2V_s}{\pi} \cos \alpha - \frac{2\omega L_c i_0}{\pi}\right) - \frac{L_0}{L_0 + L_c} \cdot R i_0 + \frac{L_c}{L_0 + L_c} \cdot \mu i_0
\]

\[L_0 \frac{di_0}{dt} = \left(\frac{2V_s}{\pi} \cos \alpha - \frac{2\omega L_c i_0}{V_s}\right) - R i_0 \tag{13}\]

where

\[\mu = \cos^{-1} \left(\frac{2\omega L_c i_0}{V_s} - \cos \alpha\right) - \alpha\]

is the length of commutation interval. In case that \(L_c \ll L_0\), such that

\[\frac{L_0}{L_0 + L_c} \approx 1 \quad \text{and} \quad \frac{L_c}{L_0 + L_c} \approx 0\]

(13) can be reduced to

\[L_0 \frac{di_0}{dt} = \left(\frac{2V_s}{\pi} \cos \alpha - \frac{2\omega L_c i_0}{V_s}\right) - R i_0 \tag{14}\]

which is identical to the classic model of the phase-controlled bridge rectifier [15]. An equivalent circuit model for (14) is given in Fig. 7(b) [16].

C. Buck ZCS QRC

The third example is a buck ZCS QRC [18] depicted in Fig. 9. The converter differs from the conventional PWM converter by the introduction of a resonant network consisting of a resonant inductor \((L_r)\) in series with the switch \((S)\) and a resonant capacitor \((C_r)\) in parallel with the freewheeling diode \((D)\), which are used to shape the switch current. With the nodes numbered in the order as shown in Fig. 9, the netlist of the converter can be written as follows:

\[
\text{NetList} = \{\{\text{VoltageSource, \{1, 0\}, } v_i\}, \\{\text{Switch, \{1, 2\}, } S\}, \\{\text{Inductor, \{2, 3\}, } L_r\}, \\{\text{Switch, \{3, 0\}, } D\}, \\{\text{Capacitor, \{3, 0\}, } C_r\}, \\{\text{Inductor, \{3, 4\}, } L_0\}, \\{\text{Capacitor, \{4, 0\}, } C_0\}, \\{\text{Resistor, \{4, 0\}, } R\}\}.
\]

A switching cycle of the converter starts when \(S\) is turned on. The resonant current \(i_r\) then ramps up linearly until it reaches \(i_0\), at which time \(D\) is turned off. The \(C_r\) voltage increases now due to the resonance between \(L_r\) and \(C_r\). During this resonance interval, \(i_r\) first continues to increase and then falls down when \(v_r\) reaches the input voltage \(V_i\). When \(i_r\) becomes zero, \(i_0\) discharges \(C_r\) and \(v_r\) will reach zero after a certain time. Zero-current switching, thus, can be achieved if \(S\) is turned off during this time interval.
Apparently, the fast subsystem of this converter consists of $L_r$, $C_r$, $S$, and $D$ (see Fig. 9). Based on the above analysis and by noticing that the state variable vector is $\mathbf{x} = [\mathbf{v}_r, \mathbf{i}_r, \mathbf{u}_r, \mathbf{v}_0]^T$, the following input lists can be defined:

- **TopoStates** = \{\{S \rightarrow 1, D \rightarrow 1\}, \{S \rightarrow 1, D \rightarrow 0\}, \{S \rightarrow 0, D \rightarrow 0\}, \{S \rightarrow 0, D \rightarrow 1\}\}
- **SlowFast** = \{\{2, 4\}, \{1, 3\}\}
- **BCs** = \{0, \{FastVars[[1]], SlowVars[[1]]\}\}
  - \{FastVars[[1]], 0\}
  - \{FastVars[[2]], 0\}, \{Ts\}
- **IV** = \{0, 0\}.

With these, an averaged model is derived using the package

$$\frac{di_0}{dt} = \frac{q_0}{L_0} + \frac{f_sV_i}{L_0}[\frac{\alpha - \sin \alpha}{\omega_n} + \frac{C_rV_i}{2\omega_0}(1 - \cos \alpha)^2]$$

$$\frac{dv_0}{dt} = \frac{i_0}{C_0} - \frac{v_0}{C_0R}$$

where

$$\alpha = \pi + \frac{i_0Z_n}{V_i}, \quad Z_n = \sqrt{\frac{L_r}{C_r}}, \quad \text{and} \quad \omega_n = \frac{1}{\sqrt{L_rC_r}}.$$ 

Large-signal transient responses of the converter can be simulated by use of this averaged model. As an illustration, the responses of the filter current $i_0$ and the output voltage $v_0$, as subjected to a step change in the switching frequency, have been simulated and are illustrated in Fig. 10 by dashed lines. The responses obtained from exact simulation of the original piecewise linear converter model are also shown (by the solid lines) for comparison. The parameters used for the simulation are $L_0 = 50 \text{ mH}, C_0 = 50 \text{ mF}, L_r = 1.6 \text{ mH}, C_r = 64 \text{ nF}, R = 4 \text{ }\Omega$, and $V_i = 10 \text{ V}$; the switching frequency is changed from 20 to 100 kHz for the simulation.

### D. ZVT PWM Boost Converter

Circuit diagram of the ZVT PWM boost converter, together with the numbering of the nodes, are given in Fig. 6. The shunt resonant network consisting of the resonant inductor ($L_r$), the resonant capacitor ($C_r$), the auxiliary switch ($S_1$), and the diode ($D_2$) is used to create a time interval prior to turning on the main switch ($S$) in which the voltage across $S$ is zero. A detailed steady-state analysis of the converter can be found in [17] in Section VI-A. The fast subsystem consists of the two switches ($S$, $S_1$) and two diodes ($D_1$, $D_1$) as well as the resonant inductor and capacitor (see the gray box in Fig. 6). The remaining part constitutes the slow subsystem. Based on the converter analysis presented in [17] and by using the

The package, an averaged model is derived:

$$\frac{di_0}{dt} = \frac{V_i - ri_0 - (1 - d)v_0}{L_0} - \frac{L_r}{L_0T_s} \frac{Z_n - \frac{v_0^2}{Z_n^2}}{2\omega_0Z_n^2}$$

$$\frac{dv_0}{dt} = \frac{(1 - d)i_0}{C_r} - \frac{i_0}{C_r R} (1 + k_r d)$$

$$+ \frac{C_r}{C_r T_s} \frac{Z_n - \frac{v_0^2}{Z_n^2}}{2\omega_0} (k_r - 1) + i_0Z_n(1 + k_r)$$

$$+ \frac{k_r i_0}{R} \left( Z_n^2 + \frac{v_0^2}{Z_n^2} \right) + \frac{v_0^2Z_n^2}{v_0} \left( 1 + \frac{k_r}{2} \right)$$

where

$$C_r = C_0 + C_r, k_r = C_r/C_0, Z_n = \sqrt{L_r/C_r}.$$ 

Details are omitted to limit the length of the paper.

To validate the modeling result, transient responses of the converter as simulated by using the averaged model are illustrated in Fig. 11 (by the dashed lines) and are compared with simulation results based on the piecewise linear switched model. The parameters used for both simulations are $L_0 = 1 \text{ mH}, C_0 = 10 \text{ mF}, L_r = 12 \text{ mH}, C_r = 0.33 \text{ nF}, R = 150 \text{ }\Omega, V_i = 150 \text{ V},$ and $f_s = 300 \text{ kHz}$; the duty ratio $d$ is changed from 0.52 to 0.6. As can be seen, the two responses agree very well.

Frequency responses of the small-signal control-to-output transfer function, as obtained by linearizing the nonlinear averaged model, are given in Fig. 12, where they are compared with those of a counterpart hard-switching PWM converter (see Section VI-A). The parameters are the same as used
VII. CONCLUSIONS

A symbolic analysis package for averaged modeling of switching power converters has been presented in this paper. The package enables the user to generate an averaged model for a converter using simple inputs and with minimum knowledge about the operation of the converter. It is applicable to various power converters using different switching techniques, including hard-switching PWM, quasi-resonant soft switching, and soft transition.

The package would be especially useful for practical engineers who need to evaluate and compare different topologies. Instructors teaching power electronics would also find it useful in helping students to learn the modeling and analysis methods as well as understand the dynamics of various converters.

Symbolic analysis opens a new area of computer applications in power electronics. Based on the package described in this paper, it would be possible to develop an integrated environment for automatic averaged modeling, transient response analysis, and control design of switching power converters. Such an environment would greatly simplify the daily work of many designers. Other possible applications of symbolic analysis in power electronics include hybrid analysis using both numerical and symbolic calculations, analytical model generation for automatic power circuit design, and repetitive evaluation of formulas for circuit optimization. With the ever-increasing speed of computers and as more and more sophisticated computer algebra systems continue to become available, it can be expected that symbolic analysis applications in power electronics will experience a rapid development in the coming years.

REFERENCES


Jian Sun (M’95) was born in Jiangsu, P.R.C., in 1964. He received the B.Eng. degree from Nanjing Institute of Aeronautics (now Nanjing University of Aeronautics and Astronautics), Nanjing, P.R.C., in 1984, the M.Eng. degree from Beijing University of Aeronautics and Astronautics, Beijing, P.R.C., in 1989, and the Dr.-Ing. degree from the University of Paderborn, Paderborn, Germany, in 1995, all in electrical engineering. From 1984 to 1987, he was an Electrical Engineer with Xian Aircraft Company, Xian, P.R.C. From 1989 to 1991, he was a Ph.D. student at Beijing University of Aeronautics and Astronautics. In 1991, he joined the Institute for Power Electronics and Electrical Drives, University of Paderborn, where he worked as a Research Associate until September 1996. Since October 1996, he has been a Post-Doctoral Fellow with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta. His research interests include power electronic system modeling and simulation, high-performance pulse width modulation, active power filtering, and high-power factor ac/dc conversion.

Horst Grotstollen received the Ing.-grad., Dipl.-Ing., and Dr.-Ing. degrees in electrical engineering from Staatliche Ingenieurschule Duisburg, Duisburg, Germany, in 1960, Rheinisch-Westfälische Technische Hochschule Aachen, Aachen, Germany, in 1965, and Technische Universität Berlin, Berlin, Germany, in 1972, respectively.

He joined AEG in 1965, where he developed electrical servo drives at the Frankfurt Research Center from 1965 to 1970 and investigated drive problems at the Department of Industrial Equipment from 1970 to 1973. He has been the Chair for Electrical Drives of the University of Erlangen-Nuernberg since 1973 as a Chief Engineer, where he taught courses on electrical machines and power electronics while conducting research on servo drives with permanent magnet synchronous motors. Since 1981, he has been Professor at the University of Paderborn, Paderborn, Germany. His research interests are digital control of ac drives and switch-mode power supplies.