Universal Real-Time Model for Active Rectifiers in Versatile Totem-Pole PFC Configurations

Axel Kiffe, Thorben Hoffstadt
dSPACE GmbH
Rathenaustraße 26, 33102 Paderborn, Germany

Tel.: +49 / (0) – 5251 1638-0
Fax: +49 / (0) – 5251 161980
E-Mail: {akiffe, thoffstadt}@dspace.de
URL: http://www.dspace.com

Keywords

Real-time simulation, Modelling, Power factor correction, Field Programmable Gate Array, Battery charger

Abstract

On-board chargers in electric vehicles often include a totem-pole PFC, as this compact active rectifier provides bidirectional energy flow and can be easily adapted to single-phase or three-phase grid operation. To test its control unit in terms of HIL simulations, a real-time capable model of the topology is required. This paper deals with the modeling and realization of a universal real-time model for versatile totem-pole configurations in single-phase or three-phase grid operation.
The proposed model uses the ideal switch representation. An extended approach to detect switch-state changes caused by Dirac impulses is introduced. Additional focus is placed on the implementation of the final model on an FPGA. Simulation results prove the validity of the proposed model.

I. Introduction

Aside from the increasing use of renewable energies, electrifying vehicles is an important task to reach the targets of reducing emissions and preventing significant global warming. Due to the high number of power electronics and electric drives in electric and hybrid-electric vehicles, the interest in hardware-in-the-loop (HIL) simulation for testing is increasing. HIL simulation is a well-known approach for testing control units in the automotive industry: The real plant is replaced by a real-time system that captures the output of the control unit, computes the reaction of the plant, and simulates the input of the control unit. This enables testing in the laboratory under reproducible conditions and reduces the risk of accidents when using the real plant.
HIL simulation of power electronic circuits is an ambitious task due to the typically high eigenvalues compared to mechanical systems, the high switching frequencies, and the fast structural changes of the system based on the semiconductor switches.
Furthermore, the variety of the topologies is quite high, which inspires the wish for a generic and topology-based

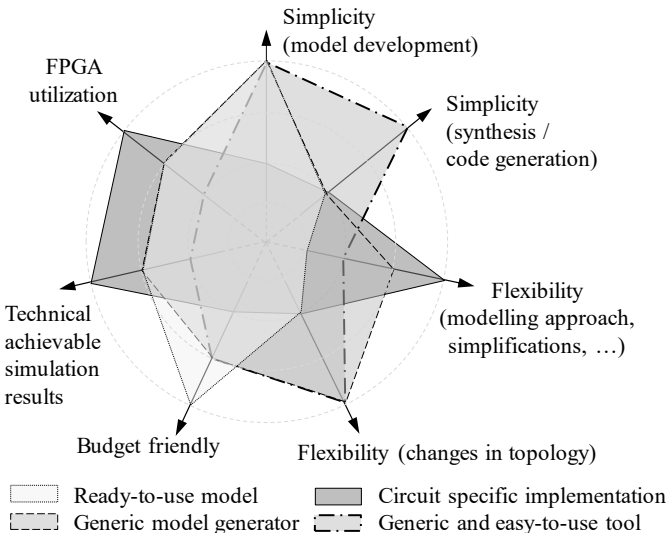


Fig. 1: Relative comparison of approaches for realizing a real-time simulation of power electronic circuits, [1].

tool to develop real-time capable models. However, a trade-off between an easy-to-use generic tool, less synthesis time, resource-efficient computation, and a cutting-edge simulation exists, especially in respect to high switching frequencies. Therefore, the approaches for realizing a FPGA-based HIL-simulation of a power electronic circuit can be distinguished between:

- Generic and easy-to-use tool (topology-based modeling)
- Generic model generator (topology-based modeling)
- Circuit-specific implementation by engineering services
- Ready-to-use model available on the market

The different properties and a relative comparison of the approaches for realizing a FPGA-based HIL-simulation can be found in Fig. 1, [1].
A generic and easy-to-use tool as well as a generic model generator use the topology of the circuit directly as modeling basis. Changes in the topology can be considered with low effort. But the performance of the real-time model will be comparable low. In strict contrast, high performance will be achieved with a circuit-specific implementation. However, the development of such a model is usually difficult and it might be impossible to adapt it to a different topology because simplifications might become untenable. A compromise is given by a ready-to-use model for a common topology. In advance, they should provide mechanisms to satisfy multiple applications, instead of one specific application. An additional advantage of ready-to-use models is the reduced cost caused by a broader user group in different applications.
The focus of this paper is placed on a flexible ready-to-use model. It is referred to [1] for further comparisons and a detailed introduction of all approaches.

II. Application

A typical structure of an on-board charger (OBC) is depicted in Fig. 2a). It consists of an active rectifier (AC/DC) providing power factor correction capability and a DC/DC converter to charge the high voltage (HV), [2]. If the vehicle includes a low voltage (LV) battery as well, the OBC comprises a second DC/DC. For the DC/DC converters, resonant converters are often used. A real-time capable model for an LLC converter was presented in [1].
For the active rectifiers, totem-pole PFC topologies are one of the most popular and promising bidirectional topologies in two-stage bidirectional OBCs for electric vehicles, [2]. A schematic of a totem-pole

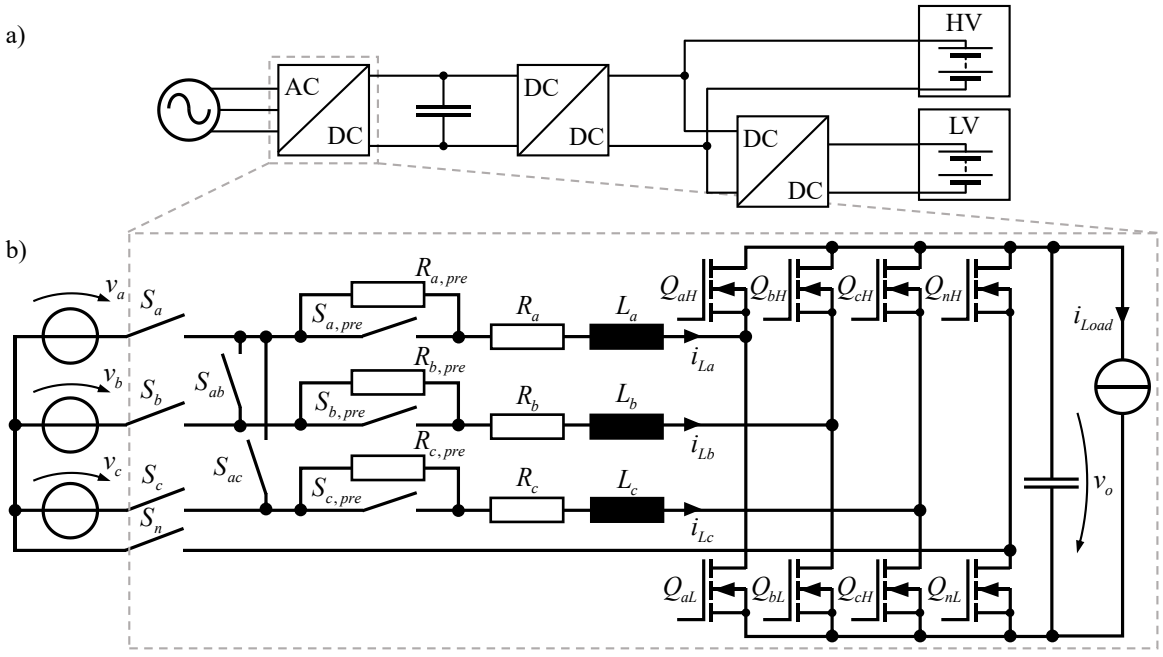


Fig. 2: General structure of an on-board charger (a) and schematic of a totem-pole rectifier (b), which can be used for versatile configurations and operation modes.

rectifier is shown in Fig. 2b). While the semiconductor switches are used for the basic functionality of the converter, the relays are used for pre-charging and configuration of versatile operation modes for single-phase and three-phase grid. A list of configurations can be found in Table I.

Therein, character a stands for active operation, which means the semiconductor is controlled by the electric control unit and its digital outputs, e.g., PWM signals. The character p stands for passive operation of the semiconductor devices, i.e., the diodes of the MOSFETs are considered but the MOSFETs are not controlled by an external gate signal. The symbol $-$ stands for a forced off state of the corresponding switch.

Table I: Configurations of switching devices in Fig. 2 for realizing different topologies.

Topology		Grid relays				Phase relays		Pre-charge relays			Half-bridges				Sources
		S_a	S_b	S_c	S_n	S_{ab}	S_{ac}	$S_{a,pre}$	$S_{b,pre}$	$S_{c,pre}$	Q_{ai}	Q_{bi}	Q_{ci}	Q_{ni}	
Totem-pole	single-phase	a	$-$	$-$	a	$-$	$-$	a	$-$	$-$	a	$-$	$-$	a/p	v_a
	2-phase interleaved	a	a	$-$	a	a	$-$	a	a	$-$	a	a	$-$	a/p	$v_a = v_b$
	3-phase interleaved	a	a	a	a	a	a	a	a	a	a	a	a	a/p	$v_a = v_b = v_c$
Full-bridge	3-phase	a	a	a	$-$	$-$	$-$	a	a	a	a	a	a	$-$	v_a, v_b, v_c

p passive operation, a active operation, $-$ forced to an off state

III. Modeling Concept and Simplification

Because the transient behavior of the currents and voltages are of interest, an oversampling model is investigated in this paper. Oversampling models keep switch-states fixed for one simulation step. Thus, their sample time T must be several times smaller than the switching period T_S. According to [3] and from practical experience, a minimum oversampling factor $\kappa = T_S/T \approx 20$ is required. Under consideration of typical switching frequencies of totem-pole rectifiers in the range of $50\,\text{kHz}$ up to $150\,\text{kHz}$, a simulation step-size T smaller than $333\,\text{ns}$ has to be obtained.

Due to the switching behavior of the n_{sw} semiconductor devices, all $2^{n_{sw}}$ switch-states and their mathematical representations must be considered. In addition to the simulation of the nominal operation, this also enables failure simulation. In respect to the topology shown in Fig. 2, 2^{17} mathematical representations need to be considered. This requires too many resources to directly apply a state-space approach. Thus, simplifications need to be exploited, to overcome this resource issue.

Figure 3 illustrates the realized concept. Usually, the pre-charging relays $S_{i,pre} \forall i \in \{a,b,c\}$ are switching slowly and not frequently. Due to this, there is enough time to update the mathematical representations on the FPGA by a corresponding processor interface depending on the switch-state of the relays. Hence, the three relays can be replaced by variable resistors $R_i \forall i \in \{a,b,c\}$ that change their values between $R_{i,off} = R_{i,pre} + R_i$ and $R_{i,on} = R_i$.

Furthermore, the switches S_{ab} and S_{bc} are used to select between a three-phase operation and an interleaved single-phase operation. In the model, their influence is considered

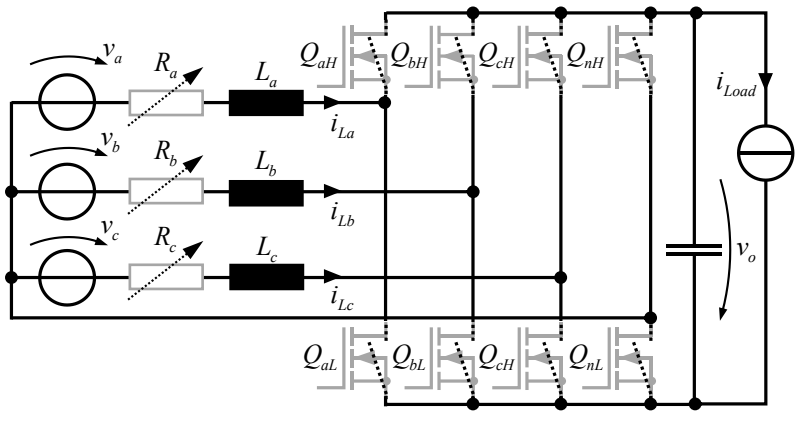


Fig. 3: Topology of the totem-pole PFC after applying the conceptional simplification to the overall circuit in Fig. 2.

by defining the voltage sources appropriately. For example, in three-phase full-bridge mode according to Table I, S_{ab} and S_{bc} are off and v_a, v_b, v_c represent the phase voltages of the grid. In case of a two-phase interleaved operation in totem-pole configuration with S_{ab} and S_{bc} both on, the voltages are chosen identical, $v_a = v_b = v_c$.

The switches of the grid relays (S_a, S_b, S_c and S_n) are realized by exploiting failure simulation. This means that the switch-logic for the MOSFETs is extended to force a switch-state independently of the active switching by gate-signals or passive switching caused by internal currents or voltages. Based on that, if, for example, a single-phase totem-pole configuration with S_b and S_c in off-state is investigated, the MOSFETs Q_{bH}, Q_{bL}, Q_{cH} and Q_{cL} must be forced to off-state, consistently, to avoid any inductor currents i_{Lb} and i_{Lc}.

Based on these simplifications, a model for the topology in Fig. 3 can be developed. This universal model can simulate all configurations listed in Table I, by forcing the switch-state of the switches indicated with $-$ to off-state with the previously mentioned failure simulation feature.

IV. Basics of Power Electronics Simulation with Ideal Switch Model

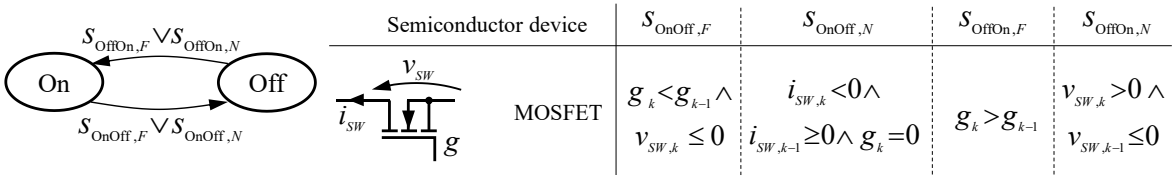
Some switch models, e.g., resistive switch model [4] or inductive-capacitive switch model [5], for representing the semiconductor devices cause high eigenvalues or impair the simulation accuracy. Thus, the ideal switch model is applied here, e.g. [1], [6], [7]. Herein, the approach from [1] is utilized, adjusted, and extended to satisfy the requirements from the application.

By using an ideal switch model, every semiconductor is replaced by a short-circuit in on-state or by an open-circuit in off-state. After the semiconductors are replaced according to their switch-state, only linear devices remain. Based on well-known approaches from the literature, e.g. [8] or [9] with the conversion of [10], the corresponding state-space representations can be determined systematically. This yields a switch-state dependent state-space representation:

$$\dot{\mathbf{x}} = \mathbf{A}_i \mathbf{x} + \mathbf{B}_i \mathbf{u}\,, \quad \mathbf{y} = \mathbf{C}_i \mathbf{x} + \mathbf{D}_i \mathbf{u}\,. \tag{1}$$

While $\mathbf{x}$, $\mathbf{u}$ and $\mathbf{y}$ are the state-, input- and output-vector, $\mathbf{A}_i$, $\mathbf{B}_i$, $\mathbf{C}_i$ and $\mathbf{D}_i$ are the system, input, output, and feedthrough matrix of the i-th switch-state. The selection of the correct switch-state for the next simulation step is done by a switch-state logic. For that, event-based and switching event-oriented conditions were formulated in [1]. These conditions become true only if a switching event currently occurs. Furthermore, the conditions provide information about the type of switching event, by having independent conditions for natural switching events (also called passive or internal switching events) and forced switching events (also called active or external switching events). While forced switching events are triggered by external signals, like a changed gate signal of a relay, natural switching events are triggered on changes of internal currents and voltages of the circuit only, e.g., a diode. An extract of the switch conditions is given in Fig. 4 for a MOSFET, which is required for the investigated totem-pole topologies. Depending on the additional subscript N or F, the condition is related to a natural or forced switching event, respectively.

In addition to the advantages of avoiding high eigenvalues and undesirable inaccuracies in the simulation, the ideal switch model requires a more complex switch-state detection compared with other switch models, e.g., a resistive switch model. The reason lies in the ideal connection or disconnection according to the switch-state, which influences the dependency between the storage devices, i.e., currents of inductors and voltages of capacitors. As an example, the small circuit in Fig. 5a is investigated, where the voltage v as well as the inductor current i_L are assumed to be positive. The diode was replaced by its ideal switch representation. At the beginning, the switch S is closed so that the current i_L flows through it. Then, the



Semiconductor device	$s_{\mathrm{OnOff},F}$	$s_{\mathrm{OnOff},N}$	$s_{\mathrm{OffOn},F}$	$s_{\mathrm{OffOn},N}$
MOSFET	$g_k < g_{k-1} \wedge$ $v_{SW,k} \le 0$	$i_{SW,k} < 0 \wedge$ $i_{SW,k-1} \ge 0 \wedge g_k = 0$	$g_k > g_{k-1}$	$v_{SW,k} > 0 \wedge$ $v_{SW,k-1} \le 0$

Fig. 4: Extract of the event-based and switching event-oriented switching conditions from [1].

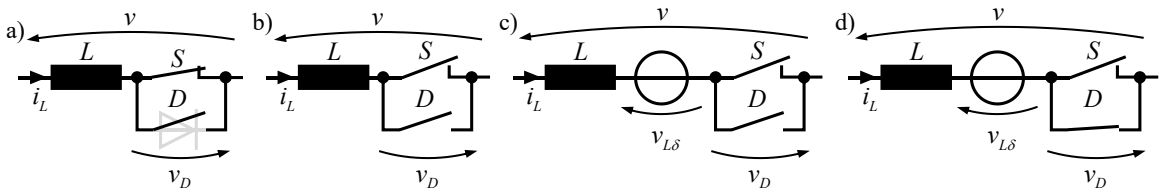


Fig. 5: Illustration of switching transitions caused by a Dirac impulse.

switch S is forced to off-state – see Fig. 5b. The switch for the diode D would change into on-state, if the voltage v_D became positive. However, due to a positive voltage v and inductor current i_L, voltage $v_D = -v$ is negative. In a real circuit, the positive inductor current continuity is ensured by the diode (see Fig. 5d). In the simulation, this is a conditionally caused switching event, which was triggered by trying to force the inductor current from a value i_L^- to another i_L^+. In the example, i_L^+ is zero (Fig. 5c). From mathematical perspective, a step in a state-variable can be explained by the occurrence of a Dirac impulse:

$$\left[x_{sw}^+ - x_{sw}^-\right]\cdot\sigma\left(t-t_{sw}\right) = \int_{-\infty}^{t}\left[x_{sw}^+ - x_{sw}^-\right]\cdot\delta\left(\tau - t_{sw}\right)d\tau\,,\ x_{sw}^- = x\left(t_{sw}^-\right),\ x_{sw}^+ = x\left(t_{sw}^+\right) \tag{2}$$

Consequently, the theoretical occurrence of a Dirac impulse would lead to infinite voltages or currents for inductances or capacitances, respectively:

$$v_{L\delta}\left(t\right) = L\cdot\left(i_L^+ - i_L^-\right)\cdot\delta\left(t-t_{sw}\right),\ i_{C\delta}\left(t\right) = C\cdot\left(v_C^+ - v_C^-\right)\cdot\delta\left(t-t_{sw}\right). \tag{3}$$

These voltages and currents can be considered by adding voltage sources in series with inductances and current sources in parallel to capacitances. For the example, this is illustrated in Fig. 5c and leads to

$$z_{SWE,D} = v_D = -v_{L\delta} - v = \underbrace{-L\cdot\left(0-i_L\right)}_{z_{SW\delta,D}}\cdot\delta\left(t-t_{sw}\right) \underbrace{-v}_{z_{SW,D}} \tag{4}$$

for the diode voltage. This type of equation can be separated into an impulsive part $z_{SW\delta}\cdot\delta\left(t-t_{sw}\right)$, which includes the Dirac delta function, and a non-impulsive part z_{SW}. Due to the infinite value of the Dirac delta function, the impulsive part $z_{SW\delta}\cdot\delta\left(t-t_{sw}\right)$ overrules the non-impulsive part z_{SW}, [7]. Thus, for the example in Fig. 5, the term $z_{SW,D} = -v$ does not matter, and the impulsive part $z_{SW\delta,D}\cdot\delta\left(t-t_{sw}\right)$ becomes greater than zero. To avoid approximations of the Dirac delta function, it is suitable to separate this equation type into the factor of the Dirac delta function $z_{SW\delta}$ and the non-impulsive part z_{SW}. Based on this, a simple logic can decide if z_{SW} would be overruled by $z_{SW\delta}$ and how this influences the switch-state, [7]. Thus, the initial state-space representations according to (1) with additional sources for considering Dirac impulses in input-vector $\mathbf{u}$ must be prepared for the simulation by a preprocessing. Therein, the

- sources for Dirac impulses were substituted appropriately,
- the output vector $\mathbf{y}$ provides the non-impulsive z_{SW} as well as the impulsive part $z_{SW\delta}\cdot\delta\left(t-t_{sw}\right)$ of all switches,
- the state-space representation is discretized.

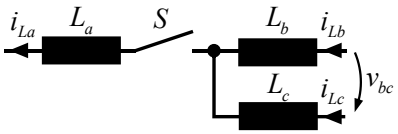
Without further explanations, the following discrete state-space representation results from the preprocessing:

$$\mathbf{x}_{k+1} = \boldsymbol{\Phi}_i\mathbf{x}_k + \mathbf{H}_i\mathbf{u}_{Sk} = \begin{bmatrix}\boldsymbol{\Phi}_i & \mathbf{H}_i\end{bmatrix}\begin{bmatrix}\mathbf{x}_k\\ \mathbf{u}_{Sk}\end{bmatrix},\ \begin{bmatrix}\mathbf{z}_{SWi,k}\\ \dot{\mathbf{z}}_{SWi,k}\\ \mathbf{z}_{SW\delta i,k}\\ \mathbf{y}_{Li,k}\end{bmatrix} = \begin{bmatrix}\mathbf{C}_{\mathbf{d}SWi}\\ \mathbf{C}_{\mathbf{d}dSWi}\\ \mathbf{C}_{\mathbf{d}SW\delta i}\\ \mathbf{C}_{\mathbf{d}Li}\end{bmatrix}\mathbf{x}_k + \begin{bmatrix}\mathbf{D}_{\mathbf{d}SWi}\\ \mathbf{D}_{\mathbf{d}dSWi}\\ \mathbf{D}_{\mathbf{d}SW\delta i}\\ \mathbf{D}_{\mathbf{d}Li}\end{bmatrix}\mathbf{u}_{Sk} = \begin{bmatrix}\mathbf{C}_{\mathbf{d}SWi} & \mathbf{D}_{\mathbf{d}SWi}\\ \mathbf{C}_{\mathbf{d}dSWi} & \mathbf{D}_{\mathbf{d}dSWi}\\ \mathbf{C}_{\mathbf{d}SW\delta i} & \mathbf{D}_{\mathbf{d}SW\delta i}\\ \mathbf{C}_{\mathbf{d}Li} & \mathbf{D}_{\mathbf{d}Li}\end{bmatrix}\begin{bmatrix}\mathbf{x}_k\\ \mathbf{u}_{Sk}\end{bmatrix} \tag{5}$$

In addition to the measurement values $\mathbf{y}_{Li,k}$, the non-impulsive part($\mathbf{z}_{SWi,k}$), and the factor of the impulsive part ($\mathbf{z}_{SW\delta i,k}$) of the switch values, the output vector contains the derivative of the non-impulsive switch values ($\dot{\mathbf{z}}_{SWi,k}$) as well. It is used to detect toggling between switch-states caused by natural switching events. However, the preprocessing and the toggling detection are not essential for understanding the content of this paper. Thus, it is referred to [1] for further details of these topics.

V. Dirac Impulse Detection

In offline simulations, the occurrence of the theoretical Dirac impulses can be detected by variable step-size solvers to detect if the left-hand limit x_0^- (in the example i_L^-) is different from the right-hand limit x_0^+ (in the example i_L^+). In contrast, real-time simulation requires a fixed step-size or at least a fixed calculation time. This avoids the classical usage of variable-step zero-crossing detection mechanisms, so that a different approach is required. A Dirac detection logic was developed in [1], that evaluates if a state variable disappears because of linear dependencies caused by a forced switching event. This seems suitable for the example in Fig. 5. However, for the totem-pole topology shown in Fig. 3, several situations can occur, where the number of state variables is constant, but a Dirac delta impulse must be considered. The necessity can be understood by investigating the example in Fig. 6. Two state variables exist independently from the switch-state of S. However, when a forced switching event occurs and S changes its switch-state, the equations for the linear dependencies change, which represents a step in the inductor current i_{La}. Thus, a Dirac impulse occurs in the simulation, i.e.



Switch-state	Number of state variables	Dependencies
on	2	$i_{La}=0 \quad 0=i_{Lb}+i_{Lc}$
off	2	$i_{La}=i_{Lb}+i_{Lc}$

Fig. 6: Example of a topology where a Dirac impulse occurs, but the number of independent state variables remains constant.

$$v_{La\delta} = L_a \cdot \left[\underbrace{0}_{i_{La}^-} - \underbrace{\left(i_{Lb}+i_{Lc}\right)}_{i_{La}^+} \right] \cdot \delta\left(t-t_{sw}\right). \tag{6}$$

when the switch S changes from off- to on-state. Therefore, a Dirac detection logic requires to consider the equation of the linear dependencies instead of evaluating the number of linear independent state variables, only. To take this into account, an extended Dirac detection logic can be derived. The basis is the Boolean function (7), which becomes true when the evaluated switch-state is included in the set $M_{p,q}$.

$$L_{p,q} = \sum_{m \in M_{p,q}} \prod_{j=1}^{n_{sw}} \tilde{s}_j \quad \text{with} \quad \tilde{s}_j = \begin{cases} \overline{s}_j & \text{for } s_{m,j}=0 \\ s_j & \text{for } s_{m,j}=1 \end{cases} \tag{7}$$

The set $M_{p,q}$ comprises all switch-states, where the q-th dependency equation is used to force the p-th state-variable. An example for illustrating the set definition and the coherence to the different dependency equations can be found in Fig. 7. In the switch-states $(000)_2$, $(010)_2$ and $(100)_2$, the p-th state-variable x_p is forced by equation f_0. Apply (7) to the example with $q=0$ yields:

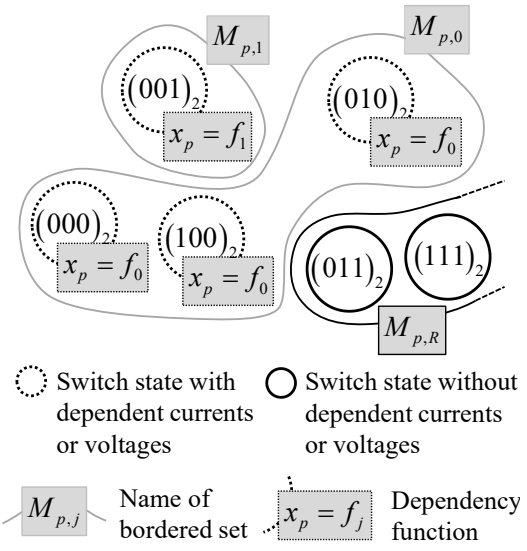


Fig. 7: Example for illustrating the set definition and the coherence to the different dependency equations.

$$L_{p,0} = \sum_{m \in M_{p,0}} \prod_{j=1}^{3} \tilde{s}_j = \left(\overline{s}_1 \wedge \overline{s}_2 \wedge \overline{s}_3\right) \vee \left(\overline{s}_1 \wedge s_2 \wedge \overline{s}_3\right) \vee \left(s_1 \wedge \overline{s}_2 \wedge \overline{s}_3\right), \; M_{p,0} = \left\{ (000)_2, (010)_2, (100)_2 \right\}. \tag{8}$$

To determine the switches, which can cause Dirac impulses when they change their switch-state due to a forced switching event, $L_{p,q}$ is negated and simplified to the minimal disjunctive normal form (MDNF), e.g., by applying the Quine-McCluskey algorithm, [11].

$$\overline{L}_{p,q} \overset{MDNF}{=} \sum_{j=1}^{n_{swg}} h_{p,q,j}\left(s_1,\ldots,s_{n_{sw}}\right) = h_{p,q,1}\left(s_1,\ldots,s_{n_{sw}}\right) \vee \ldots \vee h_{p,q,n_{swg}}\left(s_1,\ldots,s_{n_{sw}}\right) \overset{e.g.}{=} \underbrace{s_3}_{h_{p,0,1}(s_3)} \vee \underbrace{\left(s_1 \wedge s_2\right)}_{h_{p,0,2}(s_1,s_3)} \tag{9}$$

The terms $h_{p,q,j}\left(s_1,\ldots,s_{n_{sw}}\right)$ describe the topologies between the switches, which are crucial for the enforcement of the p-th state-variable by the dependency equation f_q. Based on the knowledge that a switch-state belongs to a certain set and the feedback about the type of switching event from the logic in Fig. 4, a Boolean function can be formulated to detect a Dirac impulse in respect to f_q:

$$\begin{aligned} D_{p,q} &= L_{p,q,k} \cdot \sum_{j=1}^{n_{swg}} h_{p,q,j}\left(s_{1,k-1},\ldots,s_{n_{sw},k-1}\right) \cdot \bar{h}_{p,q,j}\left(s_{1,k},\ldots,s_{n_{sw},k}\right) \cdot g_{p,q,j}\left(s_{1F,k},\ldots,s_{n_{sw}F,k}\right) \\ &\overset{e.g.}{=} L_{p,0,k} \wedge \left[\left(h_{p,0,1}\left(s_{3,k-1}\right) \wedge \bar{h}_{p,0,1}\left(s_{3,k}\right) \wedge \left(s_{3F,k}\right)\right) \vee \left(h_{p,0,2}\left(s_{1,k-1},s_{2,k-1}\right) \wedge \bar{h}_{p,0,1}\left(s_{1,k},s_{2,k}\right) \wedge \left(s_{1F,k} \vee s_{2F,k}\right)\right)\right] \\ &= L_{p,0,k} \wedge \left[\left(s_{3,k-1} \wedge \bar{s}_{3,k} \wedge s_{3F,k}\right) \vee \left(\left(s_{1,k-1} \wedge s_{2,k-1}\right) \wedge \overline{\left(s_{1,k} \wedge s_{2,k}\right)} \wedge \left(s_{1F,k} \vee s_{2F,k}\right)\right)\right] \end{aligned} \tag{10}$$

Here, $g_{p,q,j}\left(s_1,\ldots,s_{n_{sw}}\right)$ represents the OR-operated switch-states s_i which occur in $h_{p,q,j}\left(s_1,\ldots,s_{n_{sw}}\right)$ and s_{iF} represents the feedback from the event- and switch-type oriented switching condition for forced switching events, see Fig. 4. To comprehend (10), $L_{p,q,k}$ evaluates if for the switch-state in the current simulation step (t_k) the p-th state-variable is forced by the dependency function f_q. Furthermore, the sum evaluates for each essential switch topology $h_{p,q,j}\left(s_1,\ldots,s_{n_{sw}}\right)$ if it was true in the last simulation step (t_{k-1}) so that $\bar{L}_{p,q}$ was true, $h_{p,q,j}\left(s_1,\ldots,s_{n_{sw}}\right)$ is false for the current simulation step (t_k) and if a forced switching event occurred for switches, which are involved in $h_{p,q,j}\left(s_1,\ldots,s_{n_{sw}}\right)$. Using (10), the comprehensive Dirac impulse detection for the p-th state-variable is given by

$$D_p = \sum_{\forall q} D_{p,q} \,. \tag{11}$$

In conclusion, if D_p becomes true, the Dirac impulse source for the p-th state-variable needs to be considered. Thus, the impulsive conditions of the switch-values, which correspond to the involved switches (see $g_{p,q,j} \forall i,j$), must be evaluated and can overrule the results of the non-impulsive conditions.

VI. Failure simulation

One intention of failure simulation is to test the reaction of the circuit and the control unit, when one or more semiconductors do not work properly or when the connections for the gate-signal are interrupted. This requires that all $2^{n_{sw}}$ switch-states for the discrete state-space representation in (5) are considered, although not all of them might be relevant for the nominal operation of the topology.

Here, the failure simulation is also used to set constraints for the switches to simulate different totem-pole configurations with one and the same model, see Table I.

An interruption of a gate signal can be easily reproduced, by setting the corresponding model input for the gate-signal to continuously low or high. To force a switch into a certain switch-state, e.g., to consider a faulty semiconductor, requires the extension of the event-based and switching event-oriented switching conditions from [1] (extraction shown in Fig. 4). For the failure simulation feature, the results of the corresponding switching conditions $S_{OnOff,F}$, $S_{OffOn,F}$, $S_{OnOff,N}$ and $S_{OffOn,N}$ are discarded when failure simulation is activated. They are replaced by $S_{OnOff,F} = c_k < c_{k-1}$, $S_{OffOn,F} = c_k > c_{k-1}$ and $S_{OffOn,N} = S_{OnOff,N} = 0$ where c is the control signal for the failure simulation. This signal defines if the switch should be forced in off-state ($c = 0$) or in on-state ($c = 1$).

VII. Integration on FPGA

Beside the general modeling approach introduced in the previous sections, its implementation on an FPGA is important as well. Amongst others, FPGAs are well known for their high flexibility and the capability of massive parallel processing. However, resources like logic cells or DSPs to execute these operations are limited. This requires the consideration of pipelining and multiplexing for the implementation of huge and complex models like the one proposed within this paper for the totem-pole PFC.

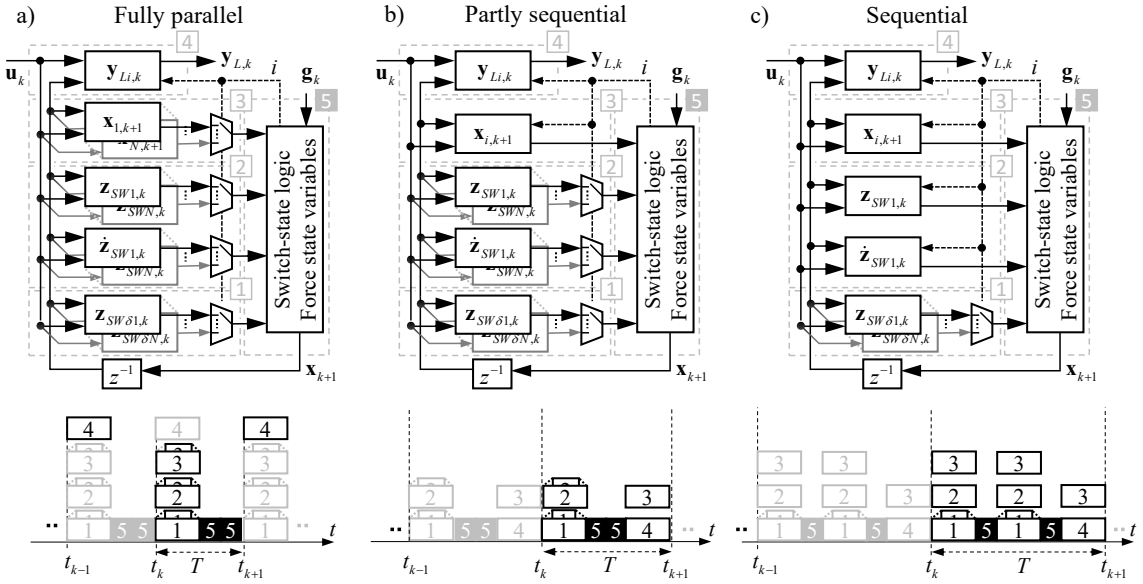


Fig. 8: Different model structures and their timing sequences for the subcomponents considering two iterations of the switch-state logic.

In [1] a systematic approach for the implementation of a matrix times vector multiplication was introduced. It allows adjustment of the trade-off between utilized hardware resources, routing effort, and latency required for the multiplication. In addition to this, Fig. 8 depicts three different model structures and their calculation sequences that can be applied to implement models according to the proposed modeling approach.

Table II compares the structures in terms of required number of matrix multiplications and expected model step-size. For the fully parallel structure in Fig. 8a), the state space representation from (5) must be calculated for all $2^{n_{sw}}$ switch-states in parallel. All required quantities for the switch-state logic are available at the same time so that they just need to be evaluated in an iterative manner. Unless some switch-states can be described with the same state space representation as other switch-states, this requires that $2^{n_{sw}}$ matrix times vector multiplications, e.g., to update $\mathbf{x}_{k+1}$, have to be implemented and executed in parallel. For the outputs $\mathbf{y}_{Li,k}$, model pipelining is applied by calculating the output equation in parallel to the next simulation step – see calculation sequence under the block diagram of Fig. 8a). As only one matrix multiplication is required in this case, the resources can be saved without affecting the step-size. Compared to the other model structures, the fully parallel structure requires a lot of hardware resources. However, it will result in the smallest model step-size T, i.e., the highest PWM resolution or oversampling factor κ can be achieved with this structure. It was applied for the model of the LLC converter with a step-size of $T = 144\,\text{ns}$ in [1].

As mentioned above, it is possible to adjust the required hardware resources for the matrix and vector multiplications to a certain extent with the multiplexing approach from [1]. However, if not enough resources are available or if the latency for the multiplications becomes too high due to a high multiplexing grade, the partly sequential model structure from Fig. 8b) can be used. Here, the switch

Table II: Comparison of the computational effort of the model structures in Fig. 8.

model structure	required number of matrix times vector multiplications for					minimum model step-size T/T_{FPGA}
	$\mathbf{x}_{k+1}$	$\mathbf{y}_{Li,k}$	$\mathbf{z}_{SW,k}$	$\dot{\mathbf{z}}_{SW,k}$	$\mathbf{z}_{SW\delta,k}$	
fully parallel	$\le 2^{n_{sw}}$	1	$\le 2^{n_{sw}}$	$\le 2^{n_{sw}}$	$\le 2^{n_{sw}}$	$\max\left(l_{x_{k+1}}, l_{z_{SW}}, l_{\dot{z}_{SW}}, l_{z_{SW\delta}}\right) + n_{Iter} \cdot l_{Logic}$
partly sequential	1	1	$\le 2^{n_{sw}}$	$\le 2^{n_{sw}}$	$\le 2^{n_{sw}}$	$\max\left(l_{z_{SW}}, l_{\dot{z}_{SW}}, l_{z_{SW\delta}}\right) + n_{Iter} \cdot l_{Logic} + l_{x_{k+1}}$
sequential	1	1	1	1	$\le 2^{n_{sw}}$	$\left(\max\left(l_{x_{k+1}}, l_{z_{SW}}, l_{\dot{z}_{SW}}, l_{z_{SW\delta}}\right) + l_{Logic}\right) \cdot n_{Iter} + l_{x_{k+1}}$

n_{sw} number of switches, l_j latency for operation j, n_{Iter} number of iterations for switch-state logic

quantities $\mathbf{z}_{SWi}$, $\dot{\mathbf{z}}_{SWi}$ and $\mathbf{z}_{SW\delta i}$ are also calculated in parallel so that they can be directly evaluated by the switch-state logic. However, the calculation of $\mathbf{x}_{k+1}$ is postponed compared to the parallel structure. This increases the step-size by the latency $l_{x_{k+1}}$ for this operation but reduces the number of matrix multiplications from $2^{n_{SW}}$ to just one for updating $\mathbf{x}_{k+1}$.

A further significant reduction of required hardware resources can be achieved by applying the sequential model structure as shown in Fig. 8c). In contrast to the afore mentioned structures, the switch quantities $\mathbf{z}_{SWi}$ and $\dot{\mathbf{z}}_{SWi}$ are calculated for the actual valid switch-state i, only. Usually, the impulsive switch quantities in $\mathbf{z}_{SW\delta i}$ are equal to zero for the most switch-states. Furthermore, often state-variables like an inductor current represent these impulsive parts (see Fig. 6) so that $\mathbf{C}_{\mathbf{d}SW\delta i}$ and $\mathbf{D}_{\mathbf{d}SW\delta i}$ from (5) are typically sparse matrices. Thus, the number of required multiplications is typically much smaller than for the other quantities. Due to this, all $\mathbf{z}_{SW\delta i}$ are calculated in parallel so that even with the sequential model structure fast model step-sizes are achieved.

Based on $\mathbf{z}_{SWi}$ and $\dot{\mathbf{z}}_{SWi}$ for the actual valid switch-state i the switch-state logic determines a new temporary switch-state. For this new state the corresponding impulsive quantities $\mathbf{z}_{SW\delta i}$ are selected and evaluated, followed by an update of the switch quantities for the resulting temporary switch-state within the next iteration step. Compared to the parallel model structure, this sequential approach yields an increase of the step-size by a factor almost equal to the number of iterations n_{Iter}. However, significantly less matrix and vector multiplications need to be implemented. This enables the implementation of models for huge and complex topologies like the totem pole PFC from Fig. 2 and 3.

VIII. Simulation results

For the topology in Fig. 3 with $n_{sw}=8$ switches, a model was realized based on the approaches presented in the previous sections. For all $2^{n_{sw}}=256$ switch-states, a state-space representation according to (5) was determined with four state-variables $\mathbf{x}=\begin{bmatrix} i_{La} & i_{Lb} & i_{Lc} & v_o \end{bmatrix}^T$, inputs $\mathbf{u}_S=\begin{bmatrix} v_a & v_b & v_c & i_{Load} \end{bmatrix}^T$ and outputs $\mathbf{y}_L=\begin{bmatrix} i_{La} & i_{Lb} & i_{Lc} & v_o \end{bmatrix}^T$. To achieve an oversampling factor κ of at least 20 for switching frequencies up to 150 kHz and to reduce the required hardware resources under consideration of the mentioned dimensions, the sequential model structure is applied.

The model is implemented on a dSPACE HIL simulator with a high-end DS6602 FPGA board that includes a Xilinx Kintex Ultrascale+ FPGA. The clock period for the DS6602 is $T_{FPGA}=8\,\text{ns}$. The model uses $n_{Iter}=2$ iterations and is calculated within 40 sample clocks. This results in a simulation step-size of $T=320\,\text{ns}$. It fulfills the specifications for the oversampling factor κ for switching frequencies up to 150 kHz.

Simulation results are shown in Fig. 9 for a single-phase grid operation. For comparison, a simulation of the topology in Fig. 2 with Simscape Electrical™ Specialized Power Systems is used as reference.

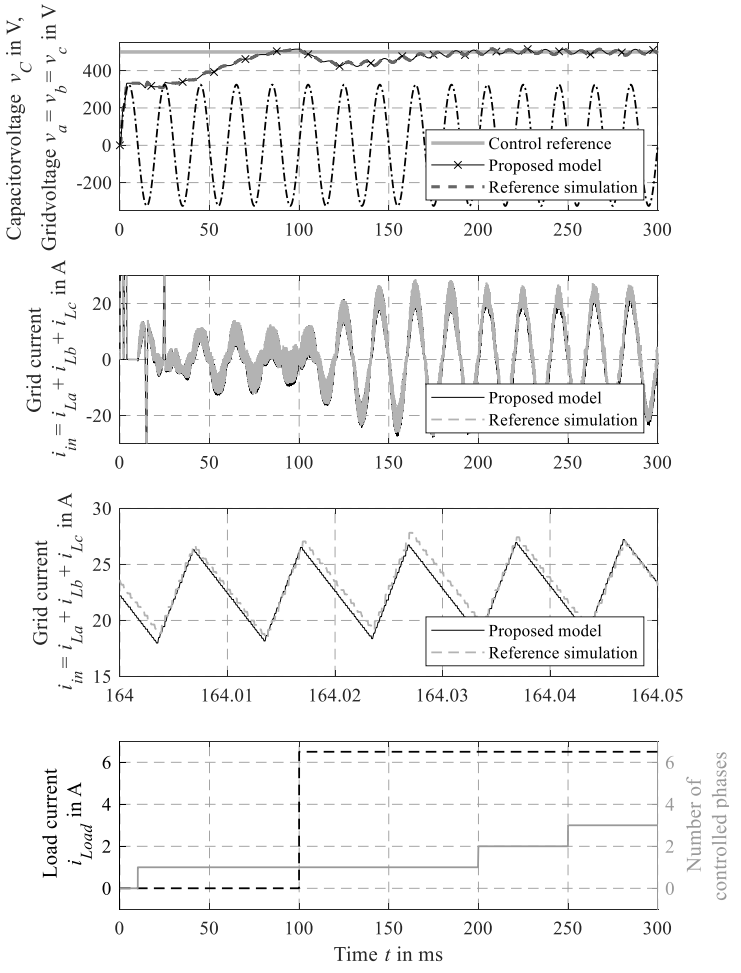


Fig. 9: Simulation results of the totem-pole PFC rectifier model.

The simulation starts with passive operation. At $t = 10\,\text{ms}$, the current control for i_{La} of the first phase as well as the voltage controller are activated. After the voltage reaches the reference value, a load step occurs at $t = 100\,\text{ms}$. Later, the current controllers of the second (i_{Lb}) and third (i_{Lc}) interleaved phases are activated sequentially at $t = 200\,\text{ms}$ and $t = 250\,\text{ms}$. This shows the appropriate consideration of the grid relays S_a, S_b, and S_c with the simplified topology in Fig. 3 combined with failure simulation. In general, the simulation results of the reference and the model show good agreement.

IX. Conclusion

Within this paper, the development of a ready-to-use model for an FPGA-based HIL simulation of a totem-pole topology was presented. The model is able to simulate several totem-pole configurations, like two-phase or three-phase interleaved operation, without loading different specific models to the FPGA.

The considered totem-pole circuit as well as the possible operations modes were introduced in section II. Afterwards, the concept and simplifications required to realize the FPGA-based HIL simulation for such a complex topology were described in section III. As a result, a simplified topology with equivalent behavior but only eight switches was obtained.

Section IV recapped basics of the applied modeling approaches based on ideal switches, while in section V an extended and systematic approach to detect switch-state changes caused by Dirac impulses was presented. An approach to force certain switches into a specific switch-state was introduced in section VI. It can be used for failure simulation, but here it was also applied to define constraints for the switches depending on the investigated totem-pole configuration.

For the implementation on an FPGA, the available resources must be considered as well. To adjust the trade-off between required resources and obtained step-size, three different model structures were compared in section VII. Finally, the proposed real-time model was validated by a simulation in section VIII. The results were in good agreement with the reference simulation.

References

[1] A. Kiffe, T. Hoffstadt, F. Puschmann: Modeling approach for real-time capable oversampling models applied to a fast switching LLC resonant converter, 2021 23rd European Conference on Power Electronics and Applications (EPE'21 ECCE Europe), 2021, pp. P.1-P.11.

[2] J. Yuan, L. Dorn-Gomba, A. D. Callegaro, J. Reimers, A. Emadi: A Review of Bidirectional On-Board Chargers for Electric Vehicles, IEEE Access, vol. 9, pp. 51501-51518, 2021.

[3] H. F. Blanchette, T. Ould-Bachir and J. P. David, "A State-Space Modeling Approach for the FPGA-Based Real-Time Simulation of High Switching Frequency Power Converters," in IEEE Transactions on Industrial Electronics, vol. 59, no. 12, pp. 4555-4567, Dec. 2012.

[4] R. Champagne, L.-A. Dessaint, H. Fortin-Blanchette, G. Sybille: Analysis and validation of a real-time AC drive simulator, in IEEE Transactions on Power Electronics, vol. 19, no. 2, pp. 336-345, March 2004.

[5] P. Pejovic and D. Maksimovic, "A method for fast time-domain simulation of networks with switches," in IEEE Transactions on Power Electronics, vol. 9, no. 4, pp. 449-456, July 1994.

[6] K. De Cuyper: “Automated modeling and implementation of power converters on a real-time FPGA-based emulator.”, PhD Thesis, 2015.

[7] Allmeling, J. H., Hammer, W. P.: PLECS-piece-wise linear electrical circuit simulation for Simulink, Proceedings of the IEEE 1999 International Conference on Power Electronics and Drive Systems. PEDS'99, pp. 355-360 vol.1, 1999.

[8] E. S. Kuh, R. A. Rohrer: The state-variable approach to network analysis, Proceedings of the IEEE, vol. 53, no. 7, pp. 672-686, July 1965.

[9] Unbehauen, R.: Elektrische Netzwerke. Springer Verlag, 1981.

[10] S. Natarajan, "A systematic method for obtaining state equations using MNA," in IEE Proceedings G - Circuits, Devices and Systems, vol. 138, no. 3, pp. 341-346, June 1991.

[11] W. V. Quine (1955) A Way to Simplify Truth Functions, The American Mathematical Monthly, 62:9, 627-631, DOI: 10.1080/00029890.1955.11988710.