Real-Time Digital Hardware Simulation of Power Electronics and Drives

Gustavo G. Parma, *Member, IEEE*, and Venkata Dinavahi, *Member, IEEE*

***Abstract*—This paper presents a digital hardware realization of a real-time simulator for a complete induction machine drive using a field-programmable gate array (FPGA) as the computational engine. The simulator was developed using Very High Speed Integrated Circuit Hardware Description Language (VHDL), making it flexible and portable. A novel device-characteristic based model suitable for FPGA implementation has been proposed for the 2-level 6-pulse IGBT-based voltage-source converter (VSC). The VSC model is computed at a fixed time-step of 12.5 ns allowing a highly detailed and precise accounting of gating signals. The simulator also models a squirrel cage induction machine, a direct field-oriented control system, a space-vector pulse-width modulation scheme (SVPWM) and a measurement system. A multirate simulation of the system shows the slow (machine) as well as the fast (VSC and control) dynamic components. Real time simulation results under steady-state and transient conditions demonstrate modeling accuracy and efficiency.**

***Index Terms*—Field-programmable gate arrays, induction motor drives, pulse-width-modulated power converters, real-time systems.**

I. Introduction

REAL-TIME digital simulators can play a vital role in the design and development of large industrial ac drives [1]–[3]. The two main components of a modern variable-speed ac drive are the power stage and the controller stage. The power stage includes an insulated-gate bipolar transistor (IGBT)-based voltage-source converter (VSC) and the induction machine, while the controller stage includes the digital controller, and a high-frequency pulsewidht-modulated (PWM) gating pattern generator. Before prototyping, the controller stage is usually subjected to several cycles of testing and re-design in an expensive facility containing the power converter, motor-generator sets, sensors, switchgear, and other test equipment. A real-time digital simulator that can model the power stage accurately and efficiently, provides an alternate means for testing the controller in a hardware-in-the-loop (HIL) configuration. This approach has the advantage of substantially reducing cost, human resources, power consumption, and physical space, while providing immunity to damage to the actual equipment due to any malfunction of the controller. However, currently available real-time simulators still suffer from modeling inaccuracies and limited computational bandwidth. The modeling of the power converter is a particular challenge due to the demands its high frequency operation places on the accuracy and precision of accounting the gating signals coming from the digital controller. This has prompted several correction algorithms [4]–[7], both offline and real-time, to be proposed using techniques such as interpolation/extrapolation and variable simulation step-size.

Power-electronic devices are often modeled in time-domain simulation programs mainly using one of the following three behavioral [8] models: 1) ideal model, 2) switching function model, and 3) averaged model. The ideal model represents each switch as a two-valued resistor for its on and off states. A change of switch status, therefore, results in a change of circuit topology which, in turn, requires the modification of the system matrix—a computationally intensive procedure especially at high switching frequencies. For real-time simulation, the preferred approach is to precompute and store all possible matrix permutations (2^n) for a converter with n switches, trading higher memory requirements for computational gains. The switching function model circumvents this precomputation and storage requirement by representing the switches as controlled voltage and current sources. This model is suitable for high-frequency operation, and the switching functions can be usually obtained by directly inspecting the converter circuit; however, the limitation of this model is that it does not accurately represent the rectifier action of the converter when the load is regenerative. The main drawback of the averaged model is that it only shows the low-frequency components of the switching functions ignoring all the high-frequency components. While this is not a concern for controller design, it causes a loss of realism for the real-time simulator.

In this paper, we propose a detailed device-characteristic based model for the VSC suitable for digital hardware realization on a field-programmable gate array (FPGA). The converter model is computed at a fixed time-step, albeit, at an extremely small step-size (of the order of a few nanoseconds). The advantages of this approach include: 1) freedom from reliance on complicated correction algorithms, 2) detailed representation of the device switching characteristics, and 3) ability to interface the model to a large-scale real-time simulator, such as a PC-cluster [9], modeling a larger and more complex host power system in which the power electronic converter is embedded. Potential applications of this approach include modular FPGA-based simulators with a library of models for multipulse and multilevel power electronic converters used in

Manuscript received January 10, 2006; revised July 12, 2006. This work was supported in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), Brazil, and in part by the Natural Science and Engineering Research Council of Canada (NSERC). Paper no. TPWRD-00749–2005.

G. G. Parma is with the Department of Electronics, Federal University of Minas Gerais, Belo Horizonte CEP 3009-210, Brazil (e-mail: parma@cpdee.ufmg.br).

V. Dinavahi is with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB T6G 2V4, Canada (e-mail: dinavahi@ece.ualberta.ca).

Digital Object Identifier 10.1109/TPWRD.2007.893620

0885-8977/$25.00 © 2007 IEEE

READ ONLY

Authorized licensed use limited to: UNIVERSITY OF ALBERTA. Downloaded on April 19,2022 at 07:50:57 UTC from IEEE Xplore. Restrictions apply.

electric drives, and power system applications such as FACTS and HVDC systems.

Since their introduction in the mid-1980s, FPGAs have roughly doubled in capacity every year, with the current state-of-the-art devices containing in excess of 200,000 logic cells, thus enabling large system-on-chip (SoC) implementation. Concurrently, development in the software CAD tools and intellectual property (IP) cores has also kept pace allowing rapid prototyping of complex digital designs. Still designing in VHDL ensures portability and platform independence. In the area of power electronics and drives, FPGAs have hitherto been used mainly for implementing control algorithms [10], [11] and/or PWM gating pattern generators [12], [13], either as stand-alone processors or as companion processors for DSPs. This paper reports the first application of FPGAs for modeling a complete ac drive system. The rest of the paper is organized as follows: Sections II–VI present the modeling details followed by the FPGA realization of the VSC, the induction machine, the control system, the gating signal generator, and the measurement system, respectively. Section VII shows how all of the individual models are brought together to achieve real-time simulation of the complete drive system on a Altera Stratix EP1S80 FPGA. Real-time simulation results are presented in Section VIII, followed by conclusions in Section IX.

II. Voltage-Source Converter Model

The power-electronic drive includes a 2-level, 6-pulse IGBT-based voltage-source converter. This section describes how the VSC model was developed and implemented in the FPGA.

A. IGBT Electrical Characteristics

In a 2-level, 3-phase VSC, six IGBTs are necessary. If a three-level three-phase converter is desired, 12 IGBTs are necessary. Although highly detailed IGBT models, which take into account all the capacitances, inductances and resistances associated with the IGBT, can be found in literature [14], such models are mainly used in simulation programs with integrated circuit emphasis such as SPICE, SABER, etc., for device manufacturing. For real time simulation such models are neither necessary or possible (at least at the present time), due to the computational effort to simulate several connected IGBTs. In this case, only the most important features of the IGBT must be taken into account.

For drive applications, the IGBTs are used only in switched mode, i.e., Region I in Fig. 1. Furthermore, the largest possible value for gate-to-emitter voltage V_{GE} is used to minimize conduction loss in this region. In this case, the forward characteristic of the IGBT can be represented by a linear substitute characteristic (dotted line in Fig. 1), which is given as follows:

$$V_{CE_{sat}}(t) = V_{CE}(T_0) + r_{CE} \cdot i_c(t). \tag{1}$$

In (1), r_{CE} is the device on-state resistance and $V_{CE}(T_0)$ is the collector-emitter threshold voltage. These data can be easily found in the IGBT manufacturer's datasheet.

Another important characteristic of an IGBT for the VSC is the timing chart for switching operation of the device. This chart

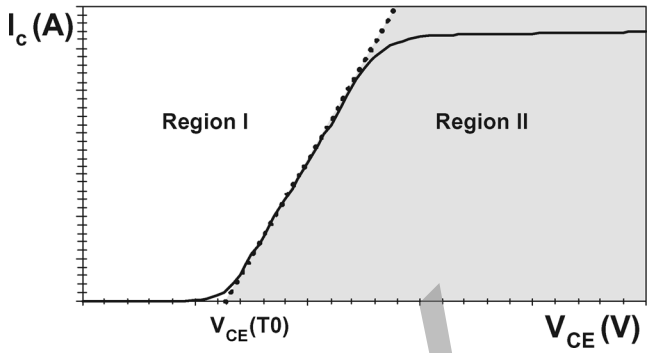


Fig. 1. Characteristic $V_{CE} - I_c$ curve of a generic IGBT for a specific value of V_{GE}.

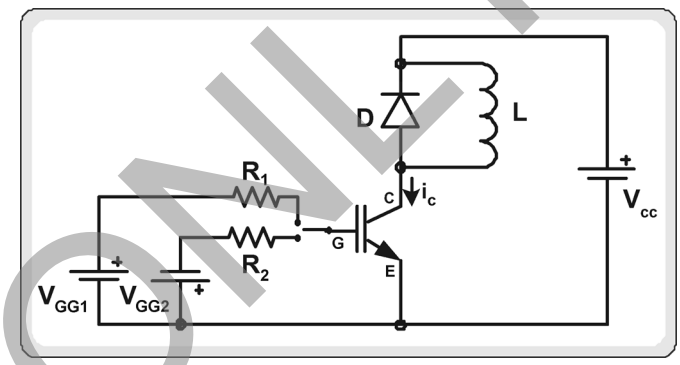


Fig. 2. Basic circuit used to analyze the IGBT timing diagram.

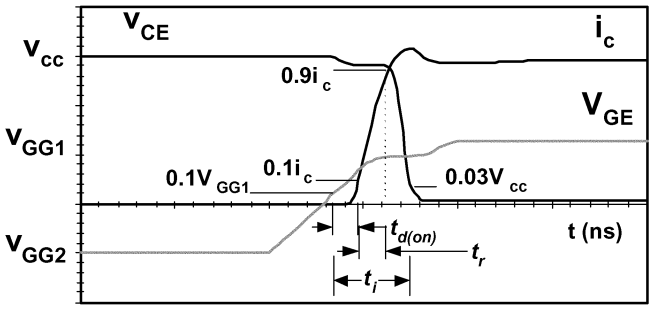


Fig. 3. Timing diagram of an IGBT during turn-on.

is analyzed with the aid of the standard circuit, shown in Fig. 2, used by the manufacturer to determine the IGBT timing characteristics. The values of the circuit elements are also standard values defined by the manufacturer.

Figs. 3 and 4 show a typical timing chart of an IGBT during turn-on and turn-off, respectively. These figures are also used to show the switching time definitions. As can be observed in Fig. 3, when the IGBT is turned-on, there is a initial time delay $t_{d(on)}$ between the on-signal at the gate and the growing value of i_c (and decreasing of V_{CE}). After this time delay i_c grows exponentially until it reaches its steady value after an overshoot, defining the rise time (t_r) depicted in Fig. 3. Fig. 4 shows the time delay between the turn-off gate signal and the instant at which i_c starts to decrease, defining $t_{d(off)}$. After this delay, the current decreases until it reaches 10% of its initial value, defining the fall time t_f. These figures also define the current time t_i, which is the elapsed time between the sensed gate signal

READ ONLY

Authorized licensed use limited to: UNIVERSITY OF ALBERTA. Downloaded on April 19,2022 at 07:50:57 UTC from IEEE Xplore. Restrictions apply.

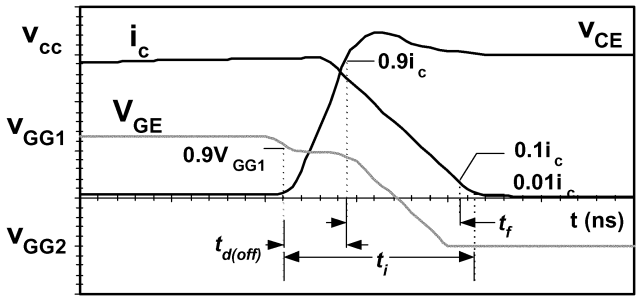


Fig. 4. Timing diagram of an IGBT during turn-off.

TABLE I
IMPLEMENTED DATA FOR CM100DU-24F IGBT UNIT FROM POWEREX

$t_{d(on)}$	$100ns$
t_r	$50ns$
$t_{d(off)}$	$400ns$
t_f	$300ns$
$V_{CE_{sat}}$	$1.8V$
$V_{d_{sat}}$	$1.25V$

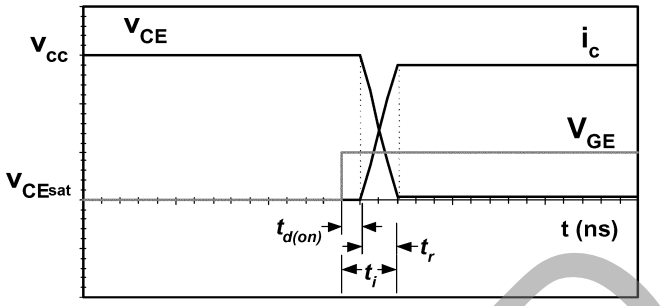


Fig. 5. Idealized timing characteristic of an IGBT during turn-on.

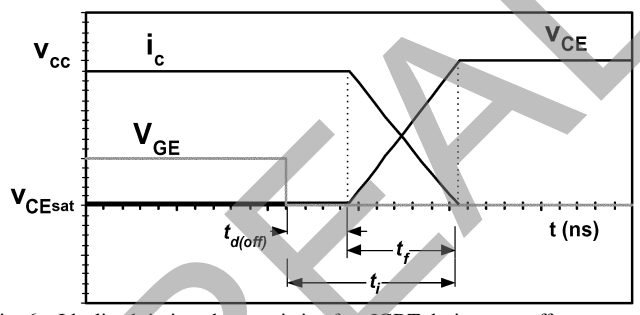


Fig. 6. Idealized timing characteristic of an IGBT during turn-off.

and the moment when the current reaches 99% of its steady-state value.

The characteristics of the implemented IGBT model are based on the CM100DU-24F IGBT from Powerex (Table I) with the idealized timing charts shown in Figs. 5 and 6. As can be observed in these figures, the major characteristics of time-delay, rise time and fall time are taken into account.

The VSC model has been developed based first on the model for one single leg, as shown in Fig. 7. For this leg, all three possible combinations for the two IGBTs are considered:

1) $A1$ switch closed and $A2$ switch open;
2) $A2$ switch closed and $A1$ switch open;
3) Both switches open (during the dead-time):
   - both switches are open just after $A1$ was closed;
   - both switches are open just after $A2$ was closed.

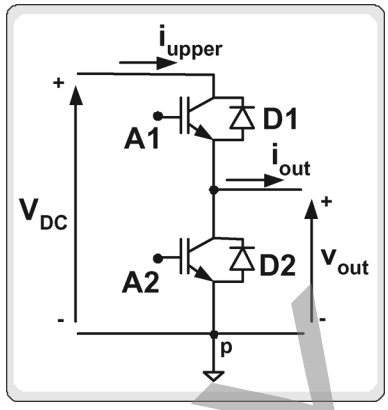


Fig. 7. One leg of the three-phase VSC.

For each combination, in every time-step of the real-time simulation, the value of $i_{\rm out}$ is used to determine whether the current is flowing through the IGBT or the anti-parallel diode. If $i_{\rm out} = 0$ and both gate signals are zero, the IGBTs are replaced by their off-state resistances and the steady-state output voltage becomes $V_{\rm out} = V_{dc}/2$.

For example, consider that at time $t = t_1$, the gate signals of $A1$ and $A2$ are changed to 1 and 0, respectively. If $i_{\rm out}$ is positive, for $t < t_1$ it would flow through the diode $D2$ and the steady-state output voltage would be $V_{\rm out} = -V_{d_{sat}}$, the voltage drop across the diode. After $t = t_1 + t_{d(on)}$, $A1$ will start to conduct and $i_{\rm out}$ will flow through the IGBT. The output voltage will then rise linearly until it reaches its steady-state of $V_{\rm out} = V_{\rm dc} - V_{CE_{sat}}$. On the other hand, if $i_{\rm out}$ is negative and assuming that for $t < t_1$ the gate signals of $A1$ and $A2$ were equal to 0 (dead-time), the current would flow through the diode $D1$, and at $t = t_1$ it will continue flowing through the same diode, resulting in the steady-state output voltage of $V_{\rm out} = V_{\rm dc} - V_{d_{sat}}$. The third and last possibility is when $i_{\rm out}$ is equal to zero. In this case, the steady-state value for the output voltage will be similar to the situation when the current was positive. The remaining switching combinations listed above can be analyzed in a similar fashion.

B. FPGA Implementation

The functional block diagram of the hardware realization of a single leg of the VSC is shown in Fig. 8. The implementation was done entirely in VHDL.

Instead of receiving both gate signals, the model receives only the gate signal for the switch $A1$. This signal goes through a block that adds the dead-time and then generates the two gate signals for $A1$ and $A2$. This approach was chosen to simulate the presence of the IGBT gate drivers, which usually add the dead-time after the PWM generation. With the gate signals, $V_{\rm dc}$, and $i_{\rm out}$ as inputs, the model then implements a state machine with four states, one for each possible combination of the switches. In each state, the output current value of the leg is verified in order to determine the path of current flow. The state machine interacts with the counters responsible for modeling the time delays, rise time and fall time, generating control signals that drive these counters. The counter outputs are used to determine the output voltage with respect to the negative terminal of dc

READ ONLY

Authorized licensed use limited to: UNIVERSITY OF ALBERTA. Downloaded on April 19,2022 at 07:50:57 UTC from IEEE Xplore. Restrictions apply.

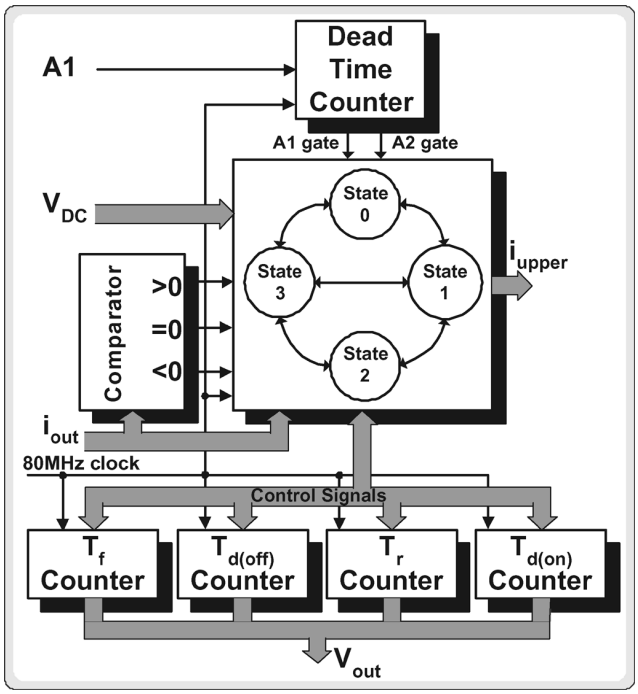


Fig. 8. Functional block diagram of the hardware realization for one VSC leg.

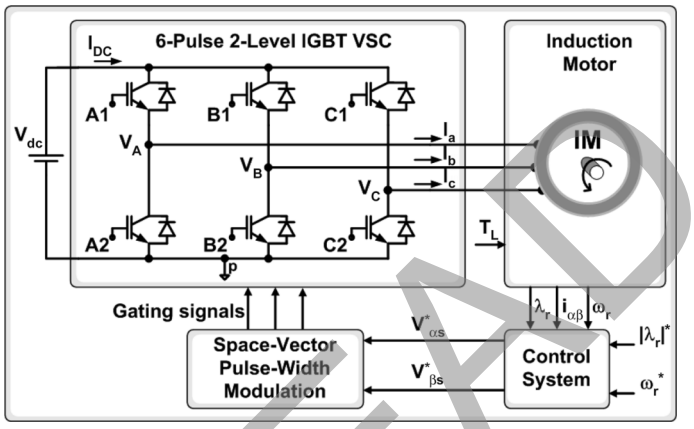


Fig. 9. System components for real-time simulation.

bus. The model also outputs the current i_{upper} through $A1$ or $D1$, which will be used to determine the output current of the dc link.

As mentioned earlier, the model implements the voltage drop across the IGBT. Due to the small value of r_{CE} (typically around 1 $\mathrm{m}\Omega$ according to the datasheet), it is possible to consider a constant voltage drop independent of the current i_c. In this case, (1) is replaced by $V_{CE_{sat}} = V_{CE}(T_0)$ when the current is flowing through the IGBT. The voltage drop $V_{d_{sat}}$ across the anti-parallel diode is also considered a constant value when the current is flowing through it. All of the necessary data retrieved from the component's data sheet are shown in Table I.

After the model of one leg of the converter was built, it was duplicated to build the other two legs to implement the three-phase VSC shown in Fig. 9. The overall VSC model requires as inputs, the three phase currents, the dc voltage and the three gate signals. The outputs of the model are the three voltages V_a, V_b and V_c with respect to the negative dc bus and the current I_{dc}. All the variables are considered as being signed fixed point numbers except the gate signals which are single bits. Table II summarizes the number format for the input and output variables. In this table, 19.5 means that 19 b are used to represent the integer part of the number and 5 b are used to represent the fractional part. The internal variables of the model usually have more fractional bits than the input/output variables in order to preserve the resolution of the simulation calculations. The developed model considers that in one simulation time-step there is no change in the current signal. This is a valid assumption since the time-step chosen for the VSC model was 12.5 ns.

TABLE II
NUMBER FORMAT FOR THE VSC MODEL VARIABLES

Variable	Format	input/output
V_{DC}	19.5	input
i_a	19.5	input
i_b	19.5	input
i_c	19.5	input
A1, B1, C1	1	input
v_a	19.5	output
v_b	19.5	output
v_c	19.5	output
I_{DC}	19.5	output

III. INDUCTION MACHINE MODEL

A. Induction Machine Electrical Characteristics

The model is based on the fifth-order stator stationary reference frame squirrel cage induction machine model [15]. This model can be described using two subsets of equations. The first is a state-space equation for the electrical side, using the stator and rotor fluxes as state variables. The second is for the mechanical side, using the rotor electrical speed as state variable. The output variables are the stator and rotor currents for the first subset and the rotor speed for the second subset. The model can be completely described by (2)–(6)

$$\begin{bmatrix} \dot{\lambda}_{\alpha s}(t) \\ \dot{\lambda}_{\beta s}(t) \\ \dot{\lambda}_{\alpha r}(t) \\ \dot{\lambda}_{\beta r}(t) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_{\alpha s}(t) \\ V_{\beta s}(t) \end{bmatrix} + \begin{bmatrix} -c_3 r_s & 0 & c_1 r_s & 0 \\ 0 & -c_3 r_s & 0 & c_1 r_s \\ c_1 r_r & 0 & -c_2 r_r & -\omega_r \\ 0 & c_1 r_r & \omega_r & -c_2 r_r \end{bmatrix} \times \begin{bmatrix} \lambda_{\alpha s}(t) \\ \lambda_{\beta s}(t) \\ \lambda_{\alpha r}(t) \\ \lambda_{\beta r}(t) \end{bmatrix} \tag{2}$$

$$\begin{bmatrix} i_{\alpha s}(t) \\ i_{\beta s}(t) \\ i_{\alpha r}(t) \\ i_{\beta r}(t) \end{bmatrix} = \begin{bmatrix} c_3 & 0 & -c_1 & 0 \\ 0 & c_3 & 0 & -c_1 \\ -c_1 & 0 & c_2 & 0 \\ 0 & -c_1 & 0 & c_2 \end{bmatrix} \begin{bmatrix} \lambda_{\alpha s}(t) \\ \lambda_{\beta s}(t) \\ \lambda_{\alpha r}(t) \\ \lambda_{\beta r}(t) \end{bmatrix} \tag{3}$$

$$\dot{\omega_r}(t) = \frac{P}{2J}\left(T_e(t) - T_L(t)\right) \tag{4}$$

$$T_e(t) = \frac{3}{2}\frac{P}{2}\left(i_{\beta s}(t) i_{\alpha r}(t) - i_{\alpha s}(t) i_{\beta r}(t)\right) \tag{5}$$

READ ONLY

Authorized licensed use limited to: UNIVERSITY OF ALBERTA. Downloaded on April 19,2022 at 07:50:57 UTC from IEEE Xplore. Restrictions apply.

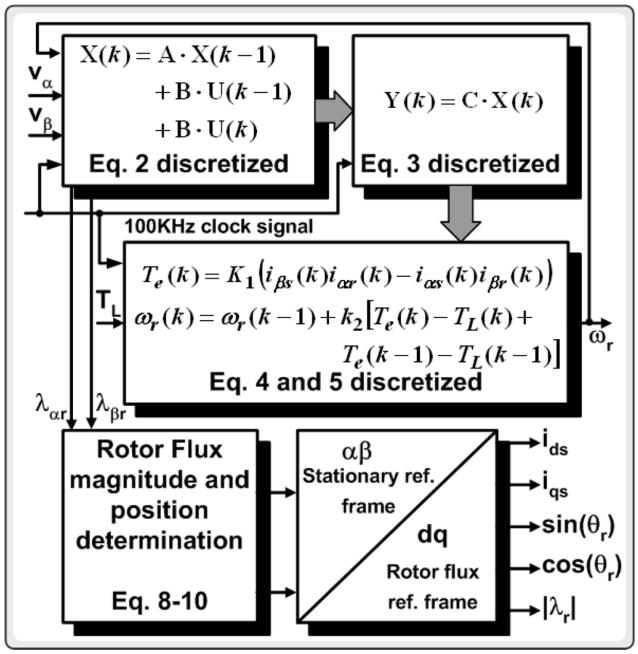


Fig. 10. Functional block of the implemented induction machine model.

where

$$c_1 = \frac{L_m}{L_s L_r - L_m^2};\ \ c_2 = \frac{L_s}{L_s L_r - L_m^2};\ \ c_3 = \frac{L_r}{L_s L_r - L_m^2}. \tag{6}$$

L_m is the mutual inductance; L_s and L_r are the stator and rotor self inductances; R_s and R_r are the stator and rotor resistances; ω_r is the electrical rotor speed; P is the number of poles and J is the total rotor inertia. T_L is the load torque; $\lambda_{\alpha s}$, $\lambda_{\beta s}$, $\lambda_{\alpha r}$ and $\lambda_{\beta r}$ are the stator and rotor flux components; $i_{\alpha s}$, $i_{\beta s}$, $i_{\alpha r}$ and $i_{\beta r}$ are the stator and rotor current components; $V_{\alpha s}$ and $V_{\beta s}$ are the stator voltage components in the stationary stator reference frame.

B. FPGA Implementation

For hardware realization, the machine model must be discretized. As can be observed from (2)–(6) the model is nonlinear, not only due to the parameter nonlinearity, but also due to the multiplication of state variables. Owing to the mechanical time constant being larger than the electrical time constants, the system can be analyzed as two decoupled systems. Therefore, during discretization the rotor speed can be assumed to be constant during one time-step of the electrical system solution, which is followed by the mechanical system solution. This approach avoids the nonlinearity due to variable multiplication.

Trapezoidal rule with a time-step $\Delta t = 10\ \mu$s was used to discretize the model based on the following transformation:

$$s = \frac{2}{\Delta t}\frac{z-1}{z+1}. \tag{7}$$

The functional block diagram of the discretized machine model is shown in Fig. 10. The discretized versions of (2)–(6) were implemented using basic arithmetic functions such as adders and multipliers, and registers were used to add the necessary time-delays. The block in Fig. 10 responsible for determining the flux magnitude, given by (8), was implemented using a binary square root function. The flux position determination, given by (9) and (10), was implemented with two dividers and two multiplexers to implement the two position options in these equations

$$|\lambda_r| = \sqrt{\lambda_{\alpha r}\lambda_{\alpha r} + \lambda_{\beta r}\lambda_{\beta r}} \tag{8}$$

$$\cos(\theta_r) = \begin{cases} \frac{\lambda_{\alpha r}}{|\lambda_r|} & \text{when} \quad \lambda_r \neq 0 \\ 1 & \text{when} \quad \lambda_r = 0 \end{cases} \tag{9}$$

$$\sin(\theta_r) = \begin{cases} \frac{\lambda_{\beta r}}{|\lambda_r|} & \text{when} \quad \lambda_r \neq 0 \\ 0 & \text{when} \quad \lambda_r = 0 \end{cases}. \tag{10}$$

The model also converts $i_{\alpha s}$ and $i_{\beta s}$ to the dq rotor flux reference frame values, which are required by the controller presented in the next section.

The parameters of the 4-pole, 50 HP, 460 V squirrel cage induction motor used for the real-time simulation are shown in Table III.

The system shown in Fig. 10 was implemented in MATLAB/SIMULINK, using the DSP Toolbox from Altera, and the generated code was integrated with the rest of the system. Table IV shows the number format for the input and output variables.

TABLE III
INDUCTION MACHINE PARAMETERS

R_s	0.087Ω
R_r	0.228Ω
L_m	$34.7mH$
L_s	$35.5mH$
L_r	$35.5mH$
J	$1.662Kg \cdot m^2$

TABLE IV
NUMBER FORMAT FOR THE INDUCTION MACHINE MODEL VARIABLES

Variable	Format	input/output
v_α	19.5	input
v_β	19.5	input
T_L	19.13	input
$i_{\alpha s}$	19.13	output
$i_{\beta s}$	19.13	output
i_{ds}	19.13	output
i_{qs}	19.13	output
$\sin(\theta_r)$	2.11	output
$\cos(\theta_r)$	2.11	output
$\lvert\lambda_r\rvert$	19.13	output
ω_r	19.13	output

IV. CONTROL SYSTEM

A. Direct Field-Oriented Control

Since its introduction by Blaschke [16], it is well known that field-oriented control allows high performance speed and torque response of induction machines. When the control is oriented by the rotor flux, the induction machine behaves like a dc machine with separate excitation. Considering a dq machine representation, where the d axis is aligned with the rotor flux space vector rotating at ω_e rad/s and the q axis is 90° apart, the following equations [17] can be obtained:

$$\frac{\lambda_r}{\tau_r} + \frac{d\lambda_r}{dt} = \frac{L_m}{\tau_r} i_{ds} \tag{11}$$

READ ONLY

Authorized licensed use limited to: UNIVERSITY OF ALBERTA. Downloaded on April 19,2022 at 07:50:57 UTC from IEEE Xplore. Restrictions apply.

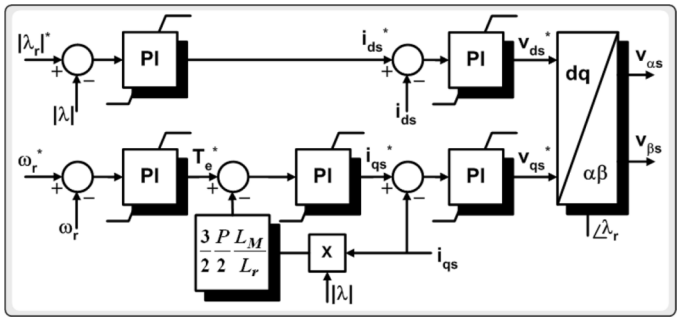


Fig. 11. Direct field-oriented controller based on rotor flux.

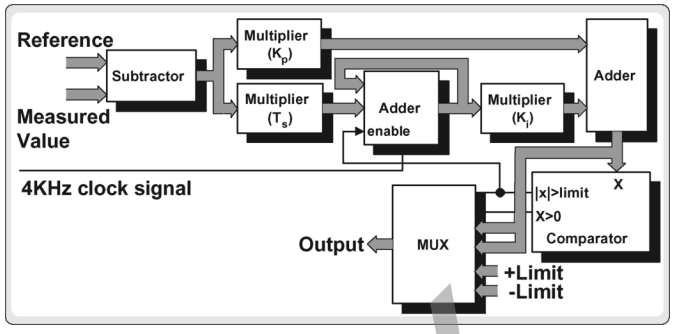


Fig. 12. Implementation of a PI controller.

$$(\omega_e - \omega_r) = \frac{L_m}{\tau_r} i_{ds} \tag{12}$$

$$T_e = \frac{3}{2}\frac{P}{2}\frac{L_m}{L_r}\lambda_r i_{qs}. \tag{13}$$

As can be observed in (11)–(13), the d axis current controls the flux while the q axis current, considering the flux constant, controls the torque and the machine speed. In these equations, τ_r is the rotor time constant defined as $\tau_r = (L_r/R_r)$.

When the induction machine is fed by a voltage source converter instead of a current source converter, the following equations can also be obtained:

$$V_{ds} = \left(R_s + \frac{L_m^2}{\tau_r L_r}\right) i_{ds} + \left(1 - \frac{L_m^2}{L_s L_r}\right) L_s \frac{di_{ds}}{dt} + V_{dx} \tag{14}$$

$$V_{dx} = -\omega_e \left(1 - \frac{L_m^2}{L_s L_r}\right) L_s i_{qs} - \frac{1}{\tau_r}\frac{L_m}{L_r}\lambda_r \tag{15}$$

$$V_{qs} = \left(R_s + \frac{L_m^2}{\tau_r L_r}\right) i_{qs} + \left(1 - \frac{L_m^2}{L_s L_r}\right) L_s \frac{di_{qs}}{dt} + V_{qx} \tag{16}$$

$$V_{qx} = \omega_e \left(1 - \frac{L_m^2}{L_s L_r}\right) L_s i_{ds} + \omega_r \frac{L_m}{L_r}\lambda_r. \tag{17}$$

Equations (14)–(17) reveal that there is an axis coupling represented by the terms V_{dx} and V_{qx}, which is contrary to the principle of vector (decoupled) control of the induction motor. However, it is addressed by the high gain current controller. Fig. 11 shows the controller block diagram. The controller requires the measurement of rotor speed (ω_r), stator currents (i_{ds} and i_{qs}), the modulus of the rotor flux $(|\lambda_r|)$ and rotor flux position $(\angle\lambda_r)$. $|\lambda_r|^*$ and ω_r^* are the references for the rotor flux and electrical speed, respectively. The information about the rotor flux (magnitude and angle) is usually obtained with the aid of a flux observer [18], [19].

B. FPGA Implementation

The hardware realization is a straightforward implementation of the block diagram shown in Fig. 11. All PI controllers have saturated outputs and are implemented in the same fashion, as shown in Fig. 12.

The control frequency was chosen to be 4 kHz in order to maintain compatibility to average commercial controllers, although a higher control frequency is also possible. Each PI controller uses one two-input subtractor, two adders with two inputs and three multipliers with one input (the other input is a constant value). The comparator and multiplex blocks are used to saturate the controller. The adder that is used as an integrator has an enable input, which is disabled every time the controller's output reaches its limit (saturation), avoiding a continuing integration of the error when the controller is saturated. Due to the inherent parallelism of the FPGA, all the controllers execute their functions at the same time, which would allow a much higher operational frequency than the chosen 4 kHz.

TABLE V
NUMBER FORMAT FOR THE CONTROLLER VARIABLES

Variable	Format	input/output
i_{ds}	19.13	input
i_{qs}	19.13	input
$\sin(\theta_r)$	2.11	input
$\cos(\theta_r)$	2.11	input
$\lvert\lambda_r\rvert$	19.13	input
$\lvert\lambda_r\rvert^*$	19.13	input
ω_r	19.13	input
ω_r^*	19.13	input
v_α	19.5	output
v_β	19.5	output

The gains of the PI controllers were chosen using the pole-placement method followed by fine tuning in offline simulation. The model was implemented in MATLAB/SIMULINK with *a posteriori* conversion to the VHDL code. It is important to notice that the control system was directly implemented in the hardware, without the use of any co-processor such as the embedded softcore NIOS processor, available for the Stratix FPGA.

As mentioned in the previous subsection, the controller requires the information about the rotor flux, which is usually obtained from an observer. As this was a simulated system, the rotor flux information was readily available in the induction machine model and the observer was not modeled. Table V shows the number format for the input and output variables of the implemented controller.

V. GATING SIGNAL GENERATION

A. Space Vector Pulse-Width Modulation

The aim of symmetrical space vector pulse-width modulation (SVPWM) is to generate a controllable three-phase voltage for the ac machine by using only two adjacent available voltage space vectors. This is achieved in a way that the mean value of the voltage output within the PWM period is equal to the reference value used to generate the PWM pattern, in order to

READ ONLY

Authorized licensed use limited to: UNIVERSITY OF ALBERTA. Downloaded on April 19,2022 at 07:50:57 UTC from IEEE Xplore. Restrictions apply.

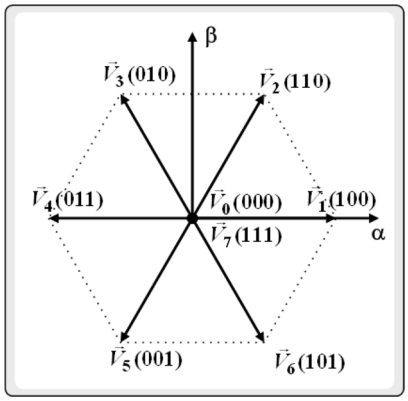


Fig. 13. Synthesized voltage space vectors.

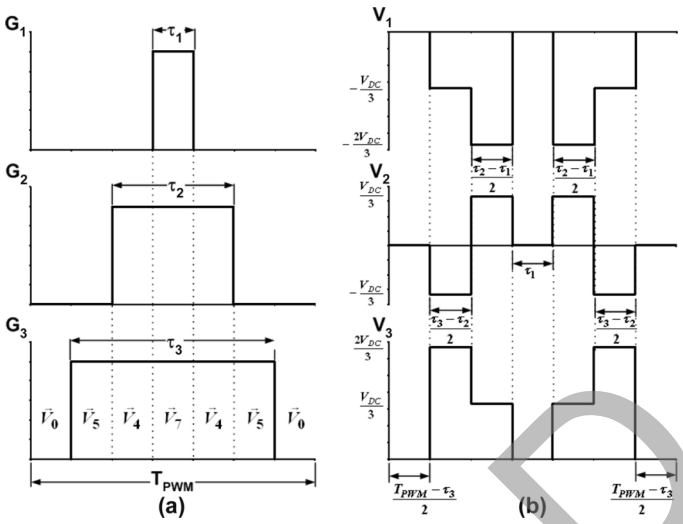


Fig. 14. (a) Generic gating signal pattern generated by the SVPWM. (b) Phase voltage generated by the VSC, with respect to the load neutral point.

minimize the harmonic distortion of the rotating magnetomotive force in the ac machine.

Fig. 13 shows the eight possible voltage space vectors the VSC can synthesize, as a function of the state of switches $A1$, $B1$, and $C1$ (numbers inside the parentheses).

Fig. 14 shows a generic gating signal pattern generated by the symmetrical SVPWM. Referring to Figs. 14 and 9, if the gating signals G_1, G_2, and G_3 are applied to switches $A1$, $B1$, and $C1$, respectively, then the voltage space vectors used to synthesize the voltage output are those denoted in Fig. 14(a), and the phase voltage V_a, V_b, and V_c will be equal to V_1, V_2, and V_3, respectively. Fig. 14(b) clearly shows that only two adjacent active voltage space vectors are used in one PWM period and that the application time of the active vectors depends only of the switch time difference. If $\bar{V}_1$, $\bar{V}_2$, and $\bar{V}_3$ represent the mean value of the output voltages to be generated with $\bar{V}_1 > \bar{V}_2 > \bar{V}_3$, the following equation can be obtained from Fig. 14:

$$\tau_i = \frac{T_{PWM}}{V_{\rm dc}} \left(\bar{V}_i + \frac{V_{\rm dc}}{2} + \frac{\bar{V}_2}{2} \right) \tag{18}$$

where $i = 1, 2, 3$ represents the legs A, B, C after the required ordering ($\bar{V}_1 > \bar{V}_2 > \bar{V}_3$); $V_{\rm dc}$ is the dc link voltage; T_{PWM} is

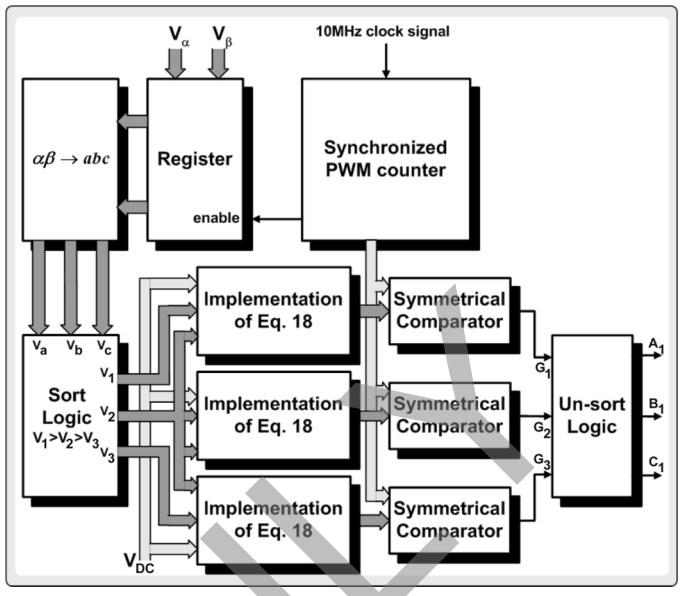


Fig. 15. Functional block diagram of the SVPWM hardware realization.

the PWM period; τ_i is the time during which the upper switch of the leg i is ON.

The SVPWM algorithm can be described as follows.

1) At the beginning of the PWM period, the voltages to be generated are sampled.
2) The voltages are sorted according to $\bar{V}_1 > \bar{V}_2 > \bar{V}_3$.
3) τ_1, τ_2 and τ_3 are evaluated from (18).
4) The turn-off time of each switch is calculated in order to produce a symmetrical SVPWM.
5) The gate signals of Fig. 14(a) are applied to each upper switch of the VSC ($A1$, $B1$, or $C1$). The gate signals for the lower switches are the complements of those for the upper switches. Dead-time insertion is part of the VSC model discussed in Section II.

B. FPGA Implementation

Fig. 15 shows the functional diagram of SVPWM implementation with a PWM frequency of 8 kHz. The synchronized PWM counter is reset in each new PWM cycle when it generates an enable signal for the register. Once the $\alpha\beta$ voltage values are sampled, they are transformed to the three phase system and sorted according to $\bar{V}_1 > \bar{V}_2 > \bar{V}_3$. Then the model implements (18) to obtain the τ_1, τ_2, and τ_3 values, which are within the PWM period, by symmetrical comparators. The output of the symmetrical comparators are the gate signals G_1, G_2, and G_3, which are properly related to the outputs $A1$, $B1$, and $C1$ by the unsorted logic block.

The model also generates a synchronization signal (one single clock pulse on each new PWM period) not shown in Fig. 15 which will be used by the measurement block explained in the next section. Table VI summarizes the number format for the input and output variables of the implemented SVPWM.

VI. Measurement System

The voltage input to the machine as a consequence of the PWM VSC control is a waveform with high frequency com-

READ ONLY

Authorized licensed use limited to: UNIVERSITY OF ALBERTA. Downloaded on April 19,2022 at 07:50:57 UTC from IEEE Xplore. Restrictions apply.

TABLE VI
NUMBER FORMAT FOR THE SVPWM VARIABLES

Variable	Format	input/output
v_α	19.5	input
v_β	19.5	input
V_{DC}	19.5	input
$A1$, $B1$, $C1$	1	output
$synchronization$	1	output

ponents. The fundamental component of this supply voltage is often needed by the control system, for example, to implement a flux observer. However, control systems usually lack the high input frequency bandwidth to measure the high frequency components of the input voltage. This voltage measurement is usually done by filtering the input voltage by analog or digital means with the drawback of adding a phase delay proportional to the fundamental frequency of the voltage. When the control system is a variable frequency control, such as vector control, this phase delay may be difficult to predict and to compensate. In order to avoid this varying time delay, a digital integration of the PWM voltage in one PWM period was implemented. The measurement system implements (19) and (20) given as follows:

$$v_{\alpha(N)} = \frac{1}{T_{PWM}} \int_{t}^{t+T_{PMW}} v_\alpha^{PWM}(t)dt \tag{19}$$

$$v_{\beta(N)} = \frac{1}{T_{PWM}} \int_{t}^{t+T_{PMW}} v_\beta^{PWM}(t)dt \tag{20}$$

where $v_{\alpha(N)}$ and $v_{\beta(N)}$ are the measured value of the output's fundamental voltage component in the Nth PWM period; $v_\alpha^{PWM}(t)$ and $v_\beta^{PWM}(t)$ are the PWM output voltages of the VSC.

The integration described in (19) and (20) must be synchronized with the PWM period in order to precisely implement the measurement system. Therefore, it is necessary that the PWM system generator outputs a synchronizing signal to the measurement system. This signal is used to output the integrated value and to reset the integrators in every new PWM period. The measurement system is also responsible for determining the mean value of $I_{\rm dc}$ within the PWM period.

VII. REAL-TIME DIGITAL SIMULATION OF PWM VSC-FED INDUCTION MACHINE DRIVE

The simulator was build on a Stratix EP1S80 FPGA Development Board (Fig. 16), supplied by the Canadian Microelectronics Corporation (CMC). The FPGA (Stratix EP1S80 from Altera) used in this board has as main features: 79,040 logic elements (LEs); 7,427,520 total RAM bits; 176 DSP blocks with 9 bit elements; 12 PLLs; and 679 maximum user I/O pins.

The development board includes an 80-MHz crystal oscillator used as a basic clock generator, two digital-to-analog converters (DACs), and two analog-to-digital converters (ADCs), as shown in Fig. 17. The development board also includes two 256 K $\times$ 36 SRAM memory banks, which are not being used by the simulator. To create a human interface to the simulator, the three push buttons are used to reset the system, change the speed set-point

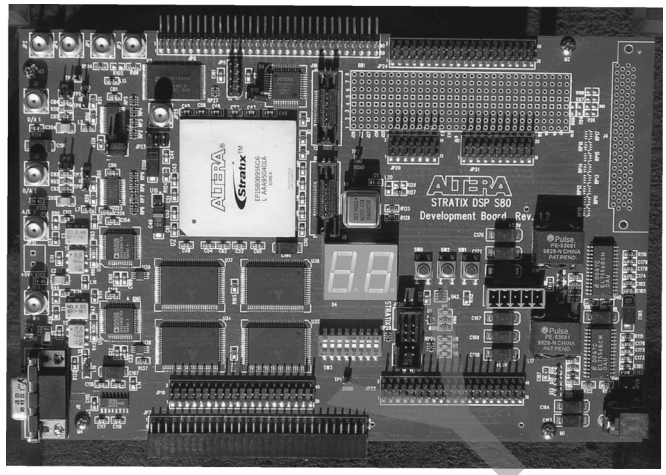


Fig. 16. Stratix FPGA development board used to build the real-time simulator.

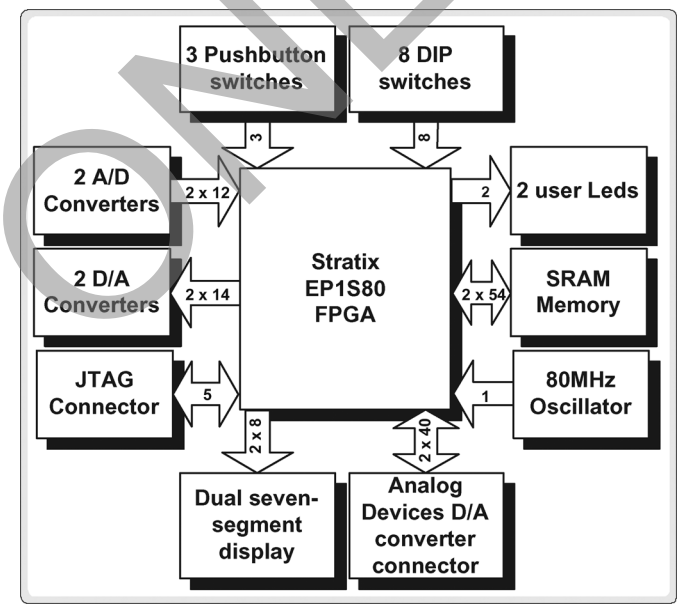


Fig. 17. Block diagram on the Stratix FPGA Development Board used to build the real-time simulator. The number inside the arrow indicates the bus width of the component.

and to reverse the speed set-point. The DIP switches were used to load the machine and to control the increment/decrement of speed set-point. In order to view all the major signals of the real-time simulation, an auxiliary board by Analog Devices was also used. This board implements a 40-channel DAC based on the AD5380 converter, and interfaces directly to the FPGA board using one of the two 40-pin connectors.

Once all the models presented in the previous sections were developed using VHDL, a final file was written to properly connect all of them to achieve real-time simulation of the complete system shown in Fig. 9.

The overall system requires three main inputs:

1) dc-link voltage $(V_{\rm dc})$;
2) reference for the rotor speed (ω_r^*);
3) reference for the magnitude of the rotor flux $(|\lambda_r|^*)$.

There is also one additional input (T_L) used to apply load onto the induction machine shaft, to simulate load transients.

READ ONLY

Authorized licensed use limited to: UNIVERSITY OF ALBERTA. Downloaded on April 19,2022 at 07:50:57 UTC from IEEE Xplore. Restrictions apply.

TABLE VII
FPGA RESOURCES UTILIZED BY SYSTEM COMPONENTS

Component	LEs	DSPs	PLLs
VSC	2377(3.01%)	0	0
Induction Machine	19285(24.40%)	136(77.27%)	0
Controller	7335(9.28%)	0	0
Space Vector PWM	3159(4.00%)	24(13.64%)	0
Measurement System	5539(7.01%)	16(9.09%)	0
Connection System	223(0.28%)	0	1(8%)
AD Interface	154(0.19%)	0	0
Total	38072(48.17%)	176(100%)	1(8%)

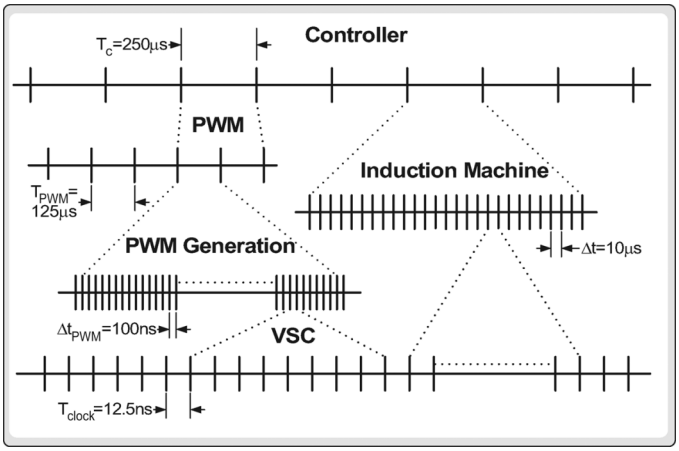


Fig. 18. Real-time simulator timing resolution.

Once the VHDL code for the simulator was compiled, the bitstream was downloaded to the development board through the JTAG interface. Table VII summarizes the FPGA resources used by individual components of the simulator. In this table, the connection system refers to the final file necessary to connect all developed components and D/A interface refers to the VHDL component necessary to interface the real-time simulator to the auxiliary Analog Devices board. Table VII shows that the overall design occupies only about half of the logic elements of the FPGA; the induction machine model being computationally intensive takes up most of the DSP blocks.

Fig. 18 shows the multirate nature of the real-time simulation. As can be observed, the components of the system use different frequencies. The most important component, the VSC, requires the highest frequency ($T_{clock} = 12.5$ ns) to model its dynamics, while it is directly connected to the induction machine model ($\Delta t = 10\ \mu$s) and the measurement system ($\Delta t = 100$ ns), which runs at a slower frequencies.

All the required frequencies were derived from the common clock frequency of 80 MHz. The subfrequencies were obtained directly inside the FPGA using a PLL or after a further frequency division to achieve the desired frequency.

A. Impact of Bit Length on Simulation Accuracy

An important issue about the FPGA-based real-time simulation is the bit-length of the implemented variables, and the number of fractional bits reserved for each variable. This issue was resolved as a compromise between the required numerical accuracy and the available FPGA resources. The bit-length of input/output variables, such as voltages measured in a practical application, was limited to 24 b, in order to simulate a 24-b ADC. The fractional part of such variables was represented using 5 b, which results in a reasonable resolution (0.03125) commonly found in real world applications where ADCs are introduced. The bit-length of the internal variables of the simulator was limited to a maximum value of 32 b with a longer fractional part in order to reduce the discretization error of the input variables, and to minimize the round-off error of output variables.

For example, for the implementation of the field-oriented controller (Section IV-B), it was necessary to implement a $sin(\cdot)$ function for axis conversion. This function was implemented using a look-up-table, whose output has a 2.11 representation (two b for the integer part and 11 b for the fractional part), while the reference voltage was coded using a 19.5 representation, with only 5 b for the fractional part. With a 2.11 representation, the possible numerical range is $+1.99951171875$ to -2.00000000000, which easily accommodates the range of the $sin(\cdot)$ function ($-1 \leq sin(\cdot) \leq +1$) with an accurate resolution (0.00048828125), while minimizing the round-off error of the output variable (reference voltage). The rest of the control variables, such as currents and speed signals listed in Table V, use a 32-bit resolution (19.13) to achieve required accuracy.

VIII. RESULTS AND DISCUSSION

This section presents the real-time experimental results obtained from the simulator. The results are shown with the aid of a four-channel oscilloscope connected to the DACs. The results show a detail of the IGBT switching characteristic, the VSC steady-state waveforms, and the induction machine transients.

A. IGBT Switching Characteristics

Fig. 19 shows the output voltage $V_{\rm out}$ of one leg of the three-phase VSC (Fig. 7) with respect to the negative side of the dc bus as a function of the gate signal $A1$, when $I_{\rm out}$ is positive. Considering $t_{\rm off}$ the instant at which $A1$ is turned-off and $t_{\rm on}$ the instant at which $A1$ is turned-on it can be observed at $t = t_{\rm off} + t_{d(off)}$ the output voltage linearly decreases until it reaches $V_{\rm out} = V_{d_{sat}}$ at $t = t_{\rm off} + t_{d(off)} + t_f$. The voltage remains $V_{\rm out} = V_{d_{sat}}$ until $t = t_{\rm on} + t_{d(on)}$, when $V_{\rm out}$ linearly increases until it reaches $V_{\rm out} = V_{\rm dc} - V_{CE_{sat}}$ at $t = t_{\rm on} + t_{d(on)} + t_r$. In this case, the IGBT $A2$ does not even turn on, since its on-time is the minimum value allowed, equal to $t_{d(on)} + t_r$, which is less than the dead-time added by the circuit that drives the gate signal.

Fig. 20 is similar to Fig. 19 but when the current $I_{\rm out}$ is zero and $A2$ is turned-on after the dead-time. In this case, as can be observed, at $t = t_{\rm off} + t_{d(off)}$, $V_{\rm out}$ linearly decreases until it reaches $V_{\rm out} = (V_{\rm dc}/2)$ at $t = t_{\rm off} + t_{d(off)} + t_f$ and remains at this value during the entire dead-time, when both switches are off ($t_{dead-time} = t_{d(off)} + t_f$). When $A2$ is turned-on, after the dead-time, $V_{\rm out}$ linearly falls again, accordingly to the rise-time of switch $A2$, until it reaches $V_{\rm out} = V_{CE_{sat}}$ at $t = t_{\rm off} + t_{d(off)} + t_f + t_{d(on)} + t_r$.

READ ONLY

Authorized licensed use limited to: UNIVERSITY OF ALBERTA. Downloaded on April 19,2022 at 07:50:57 UTC from IEEE Xplore. Restrictions apply.

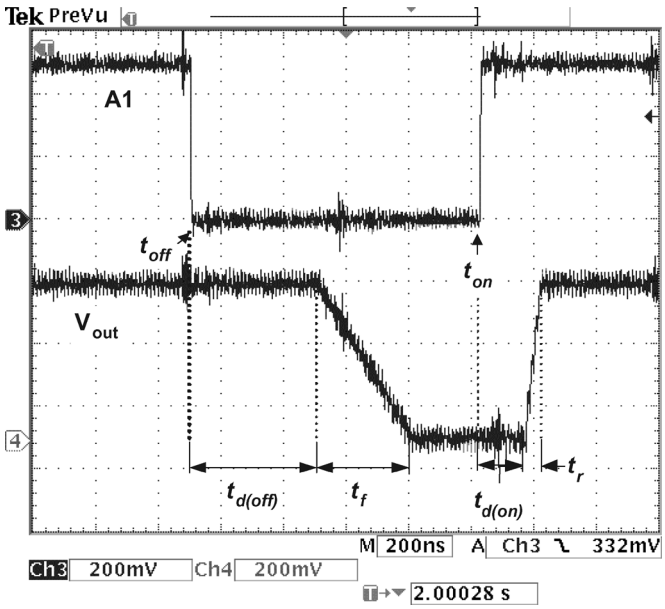


Fig. 19. Output voltage ($V_{\rm out}$) of one leg of the three-phase VSC with respect to the negative dc bus (Fig. 7) as a function of the gate signal ($A1$) when $I_{\rm out}$ is positive.

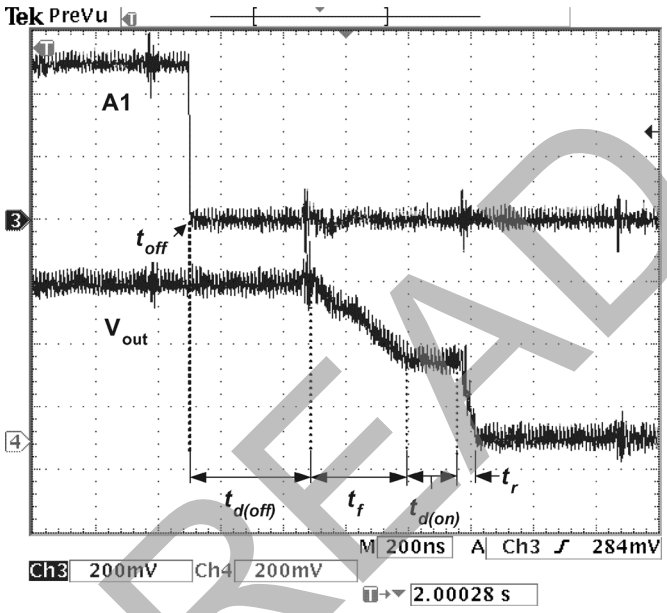


Fig. 20. Output voltage ($V_{\rm out}$) of one leg of the three-phase VSC with respect to the negative dc bus (Fig. 7) as a function of the gate signal ($A1$) when $I_{\rm out}$ is zero.

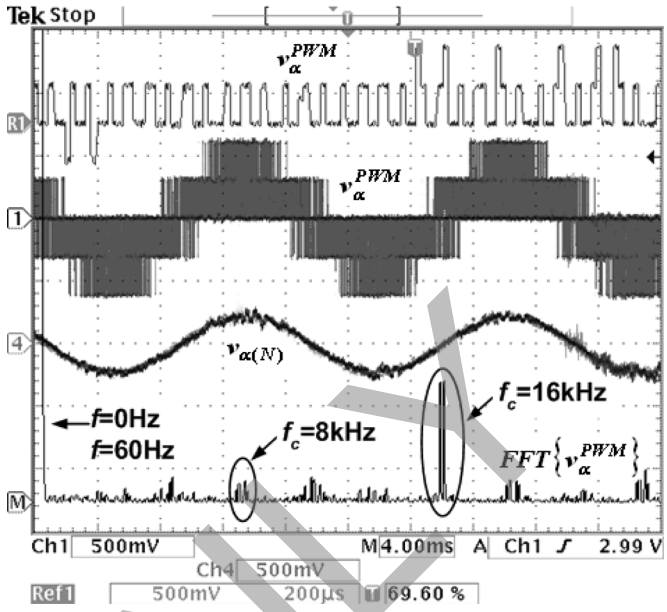


Fig. 21. Steady state of the 60-Hz voltage v_α applied to the machine. (v_α^{PWM}, $v_{\alpha(N)}$): 555 V/div.

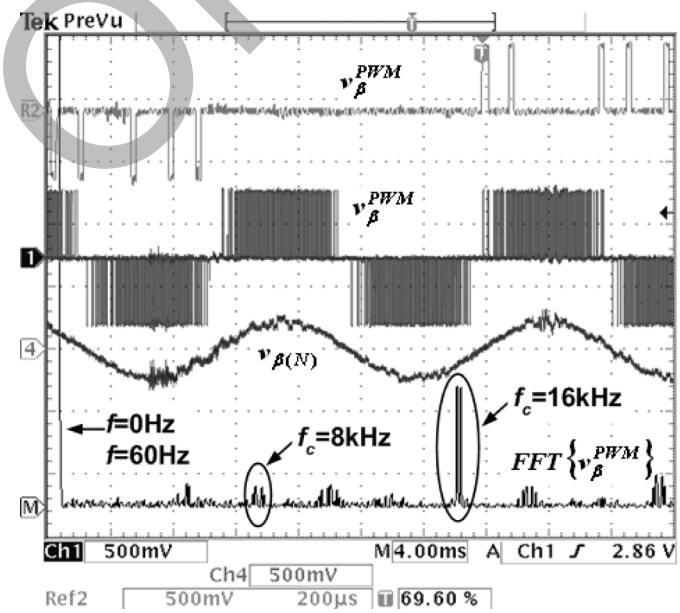


Fig. 22. Steady-state of the 60-Hz voltage v_β applied to the machine. (v_β^{PWM}, $v_{\beta(N)}$): 480 V/div.

B. Steady-State Results

Figs. 21 and 22 show the VSC output after the induction machine reaches its steady-state with $\omega_r^* = 377$ rad/s. Fig. 21 shows the VSC output voltage v_α^{PWM}, a detailed portion of it, its mean value $v_{\alpha(N)}$ within a PWM period determined by the measurement system and its fast Fourier transform (FFT). The voltage v_α^{PWM} is equal to the phase voltage v_a, of the balanced three-phase *abc* system that supplies the machine. In this figure is possible to observe the typical 2 level voltages present in the converter phase voltage. Besides the fundamental frequency component at $f = 60$ Hz, the FFT calculated by the oscilloscope shows a fictitious dc component ($f = 0$ Hz) which was added by the DAC since its output is a monopolar value between 0 V and 5 V. Other significant components are the sidebands centered at $f_c = 8$ kHz and $f_c = 16$ kHz, since the PWM frequency is equal to 8 kHz. This FFT analysis is coherent with the analytical results presented in [20].

Fig. 22 is equivalent to the previous figure, but shows v_β^{PWM} instead of v_α^{PWM}. Considering a balanced load, the voltage v_β^{PWM} is proportional to the phase-to-phase voltage v_{bc}; it is possible to observe the one level voltage present in the phase-to-phase voltage. The FFT results agree very well with analytical analysis.

READ ONLY

Authorized licensed use limited to: UNIVERSITY OF ALBERTA. Downloaded on April 19,2022 at 07:50:57 UTC from IEEE Xplore. Restrictions apply.

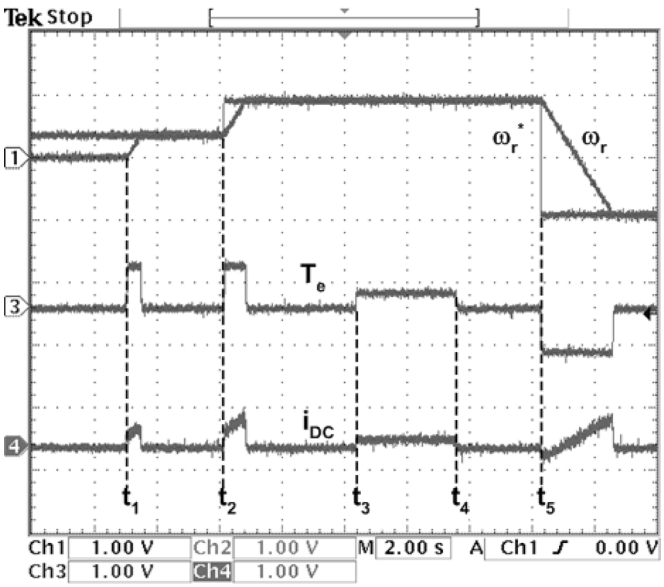


Fig. 23. Machine speed (ω_r), electromagnetic torque (T_e) and VSCs $I_{\rm dc}$ current for: machine startup at t_1, speed set-point variation at t_2, application and removal of load at t_3 and t_4, respectively, and speed reversal at t_5. (ω_r, ω_r^*): 377 (rad/s)/div., T_e: 400 N.m./div., $I_{\rm dc}$: 150 A/div.

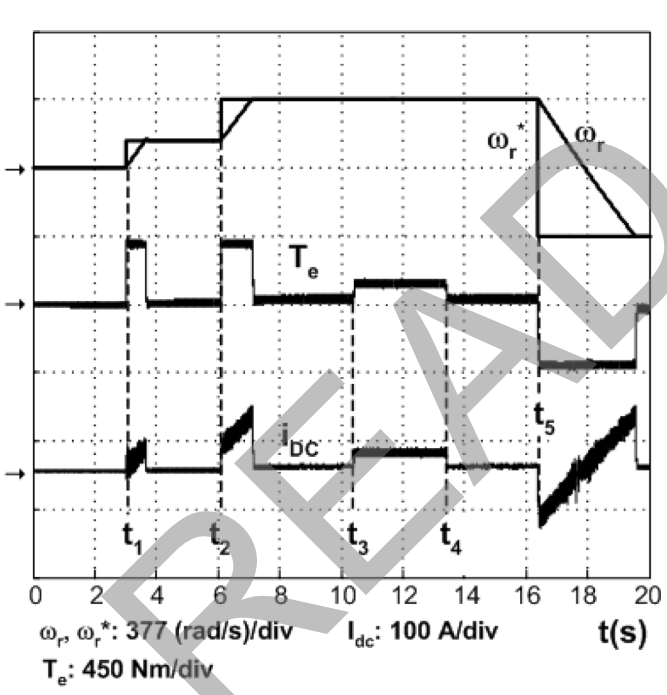


Fig. 24. Offline simulation: machine speed (ω_r), electromagnetic torque (T_e) and VSCs $I_{\rm dc}$ current for: machine startup at t_1, speed set-point variation at t_2, application and removal of load at t_3 and t_4, respectively, and speed reversal at t_5.

C. Transient Results

Fig. 23 shows the machine speed ω_r, the speed reference ω_r^*, the electromagnetic torque T_e, and the mean value of $I_{\rm dc}$ within one PWM period when the machine is subjected to the following transients: (1) machine startup with a speed set-point of 150 rad/s at t_1, (2) speed set-point variation to 377 rad/s at t_2, (3) application of 100 N.m. load at t_3 and removal of the load at t_4, and (4) speed reversal to -377 rad/s at t_5. As can be

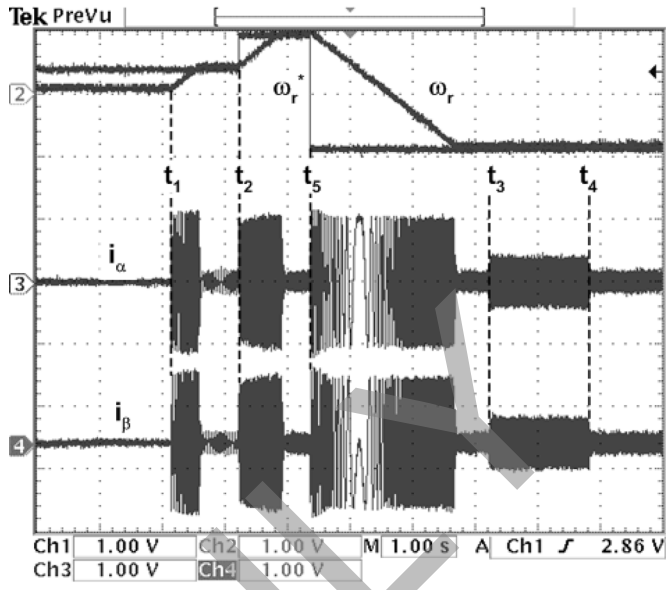


Fig. 25. Machine speed (ω_r) and currents (i_α and i_β) for: machine startup at t_1, speed set-point variation at t_2, application and removal of load at t_3 and t_4, respectively, and speed reversal at t_5. (ω_r, ω_r^*): 377 (rad/s)/div., (i_α, i_β): 150 A/div.

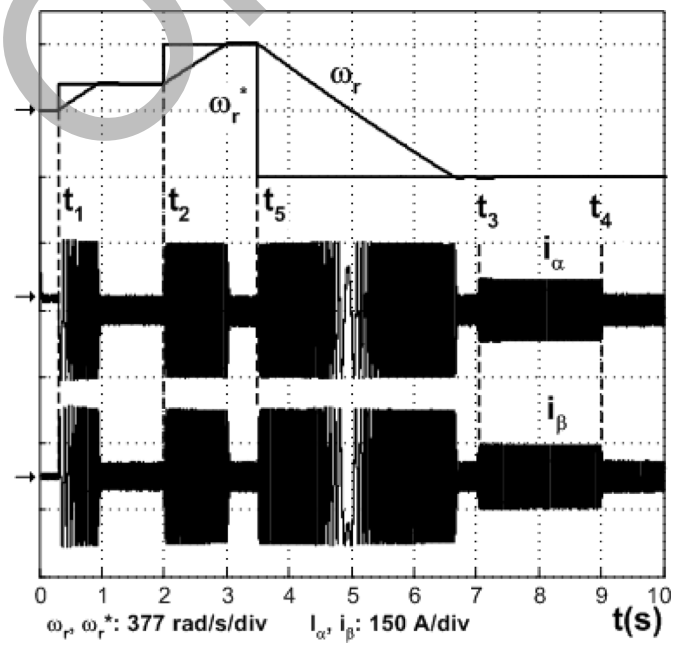


Fig. 26. Offline simulation: machine speed (ω_r) and currents (i_α and i_β) for: machine startup at t_1, speed set-point variation at t_2, application and removal of load at t_3 and t_4, respectively, and speed reversal at t_5.

observed from the machine torque and speed, the controller has a fast performance during the transients, and no perturbation can be seen in the speed during the application or removal of the load. As the machine was modeled ignoring the mechanical losses, and the machine resistances are quite small, the value of $I_{\rm dc}$ is almost zero once the machine reaches steady-state. A significant value of $I_{\rm dc}$ different from zero can be observed during the machine transients and when the load is applied to the shaft. During the transients, the converter must supply or absorb the power consumed or generated by the machine as a function of the controller action. It is possible to observe that this current $I_{\rm dc}$

READ ONLY

Authorized licensed use limited to: UNIVERSITY OF ALBERTA. Downloaded on April 19,2022 at 07:50:57 UTC from IEEE Xplore. Restrictions apply.

can be positive or negative, depending on the transient, which shows the inverter or rectifier operation of the VSC.

Fig. 25 shows ω_r, ω_r^*, and the currents i_α and i_β for the same transients listed above. Once again the fast behavior of the controller can be seen during the transients. It is also possible to observe the current frequency variation during the speed reversal. These results have been validated by offline simulation using MATLAB/SIMULINK as shown in Figs. 24 and 26, and they clearly prove that the machine model and the control system are working properly.

IX. Conclusion

With the emergence of FPGAs with higher density, lower cost, and higher processing rates, it is imperative that this technology be exploited for the real-time simulation of power electronics and drives. With this goal in mind, this paper presented the digital hardware realization of a complete ac drive system on a FPGA. The main challenge in achieving this realization was the VSC for which a novel device-characteristic based model was proposed. The model is extremely detailed allowing a precise accounting of switching signals coming from the digital controller without the need for any correction algorithms. The other components of the implemented real-time simulator include a detailed induction machine model, a direct field-oriented control system, a space-vector PWM scheme, and a measurement system. The entire model was written in VHDL to preserve the portability and flexiblity of the design. Real-time digital simulation results are presented under steady-state and transient conditions to demonstrate the successful implementation of the drive system. The FPGA resource utilization table provides an idea of the level of model complexity that can be attained with current FPGA technology. We envision that in future real-time simulators, the concurrency and high-speed parallel operation of FPGAs can be taken advantage of by hardwiring computationally intensive portions of the system model. Such an implementation can be interfaced with a larger and more complex power system model running on a large-scale real-time simulator. The applications of such implementation include the modeling of multilevel multipulse converters not only for drives but also for power system applications such as FACTS and HVDC systems.

READ ONLY

References

[1] A. G. Jack, D. J. Atkinson, and H. J. Slater, "Real-time emulation for power equipment development. I. Real-time simulation," *Proc. Inst. Elect. Eng.*, vol. 145, no. 2, pp. 92–97, Mar. 1998.

[2] H. Le-Huy, G. Sybille, P. Giroux, J. Soumagne, and F. Guay, "Digital real-time simulation of a four-quadrant dc drive for static transfer switch testing," in *Proc. IEEE Power Eng. Soc. Summer Meeting*, 1998, pp. 761–765.

[3] R. Champagne, L.-A. Dessaint, H. Fortin-Blanchette, and G. Sybille, "Analysis and validation of a real-time AC drive simulator," *IEEE Trans. Power Electron.*, vol. 19, no. 2, pp. 336–345, Mar. 2004.

[4] P. Kuffel, K. Kent, and G. Irwin, "The implementation and effectiveness of linear interpolation in digital simulation," in *Proc. IPST Int. Conf. Power Systems Transients*, Lisbon, Portugal, Sep. 1995, pp. 499–504.

[5] V.-Q. Do, D. McCallum, P. Giroux, and B. D. Kelper, "A backward-forward interpolation technique for a precise modeling of power electronics in HYPERSIM," in *Proc. Int. Conf. Power Systems Transients*, Rio de Janeiro, Brazil, Jun. 2001, pp. 337–342.

[6] V. R. Dinavahi, M. R. Iravani, and R. Bonert, "Real-time digital simulation of power electronic apparatus interfaced with digital controllers," *IEEE Trans. Power Del.*, vol. 16, no. 4, pp. 775–781, Oct. 2001.

[7] M. O. Faruque, V. Dinavahi, and W. Xu, "Algorithms for the accounting of multiple switching events in digital simulation of power electronic systems," *IEEE Trans. Power Del.*, vol. 20, no. 2, pt. 1, pp. 1157–1167, Apr. 2005.

[8] H. Jin, "Behavior-mode simulation of power electronic circuits," *IEEE Trans. Power Electron.*, vol. 12, no. 3, pp. 443–452, May 1997.

[9] L. Pak, M. O. Faruque, X. Nie, and V. Dinavahi, "A versatile cluster-based real-time digital simulator for power engineering research," *IEEE Trans. Power Syst.*, vol. 21, no. 2, pp. 1–11, May 2006.

[10] S. Jung, M. Chang, J. Jyang, L. Yeh, and Y. Tzou, "Design and implementation of an FPGA-based control IC for ac-voltage regulation," *IEEE Trans. Power Electron.*, vol. 14, no. 3, pp. 522–532, May 1999.

[11] A. de Castro, P. Zumel, O. Garcia, T. Riesgo, and J. Uceda, "Concurrent and simple digital controller of an ac/dc converter with power factor correction based on an FPGA," *IEEE Trans. Power Electron.*, vol. 18, no. 1, pp. 334–343, Jan. 2003.

[12] Y. Tzou and H. Hsu, "FPGA realization of space-vector PWM control IC for three-phase pwm inverters," *IEEE Trans. Power Electron.*, vol. 12, no. 6, pp. 953–963, Oct. 1997.

[13] V. Dinavahi, M. R. Iravani, and R. Bonert, "Design of a real-time digital simulator for a D-STATCOM system," *IEEE Trans. Ind. Electron.*, vol. 51, no. 5, pp. 1001–1008, Oct. 2004.

[14] K. Sheng, B. W. Williams, and S. J. Finney, "A review of IGBT models," *IEEE Trans. Power Electron.*, vol. 15, no. 6, pp. 1250–1266, Nov. 2000.

[15] P. C. Krause, O. Wasynczuk, and S. D. Sudhoff, *Analysis of Electric Machinery*. New York: IEEE Press, Jan. 1995.

[16] F. Blaschke, "The principles of field orientation as applied to the new transvector closed-loop system for rotating field machines," *Siemens Rev.*, vol. 34, pp. 217–220, 1972.

[17] D. W. Novotny and T. A. Lipo, *Vector Control and Dynamics of AC Drives*. Oxford, U.K.: Oxford Univ. Press, 2000.

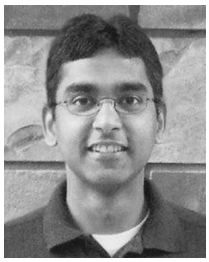
[18] G. C. Verghese and S. R. Sanders, "Observers for flux estimation in induction machines," *IEEE Trans. Ind. Electron.*, vol. 35, no. 1, pp. 85–94, Feb. 1988.

[19] Y. Hori and T. Umeno, "Implementation of robust flux observer based field orientation (fofo) controller for induction machines," in *Proc. Conf. Rec. IEEE Industry Applications Soc. Annu. Meeting*, Oct. 1989, vol. 1, pp. 523–528.

[20] J. T. Boys and P. G. Handley, "Harmonic analysis of space vector modulated PWM waveforms," *Proc. Inst. Elect. Eng. B*, vol. 137, no. 4, pp. 197–204, Jul. 1990.

Gustavo G. Parma (S'98-M'06) received the B.Eng. and Ph.D. degrees in electrical engineering from Universidade Federal de Minas Gerais, Belo Horizonte, Brazil, in 1996 and 2000, respectively.

Currently, he is a Professor at the University of Minas Gerais, Belo Horizonte, Brazil. He was a Postdoctoral Fellow at the University of Alberta, Edmonton, AB, Canada during 2005–2006. His research interests include machine control, slid mode control, digital electronics, power electronics, and real-time digital simulation and control.



Venkata Dinavahi (M'00) received the B.Eng. degree in electrical engineering from VRCE Nagpur University, Nagpur, India, in 1993, the M.Tech. degree from the Indian Institute of Technology, Kanpur, in 1996, and the Ph.D. degree in electrical and computer engineering from the University of Toronto, Toronto, ON, Canada, in 2000.

Currently, he is an Associate Professor at the University of Alberta, Edmonton, AB, Canada. His research interests include electromagnetic transients, power electronics, and real-time digital simulation and control.

Authorized licensed use limited to: UNIVERSITY OF ALBERTA. Downloaded on April 19,2022 at 07:50:57 UTC from IEEE Xplore. Restrictions apply.