energies

MDPI

Article

FPGA-Based Controller for a Permanent-Magnet Synchronous Motor Drive Based on a Four-Level Active-Clamped DC-AC Converter

Joan Nicolas-Apruzzese *, Emili Lupon, Sergio Busquets-Monge, Alfonso Conesa, Josep Bordonau and Gabriel García-Rojas

Electronic Engineering Department, Universitat Politècnica de Catalunya, 08028 Barcelona, Spain; emili.lupon@upc.edu (E.L.); sergio.busquets@upc.edu (S.B.-M.); alfonso.conesa@upc.edu (A.C.); josep.bordonau@upc.edu (J.B.); gabriel.garcia.rojas@alu-etsetb.upc.edu (G.G.-R.)

* Correspondence: joan.nicolas@upc.edu; Tel.: +34-93-401-7152

Received: 3 August 2018; Accepted: 28 September 2018; Published: 2 October 2018

Abstract: This paper proposes a closed-loop control implementation fully-embedded into an FPGA for a permanent-magnet synchronous motor (PMSM) drive based on a four-level active-clamped converter. The proposed FPGA controller comprises a field-oriented control to drive the PMSM, a DC-link voltage balancing closed-loop control (VBC), and a virtual-vector-based modulator for a four-level active-clamped converter. The VBC and the modulator operate in consonance to preserve the DC-link capacitor voltages balanced. The FPGA design methodology is carefully described and the main aspects to achieve an optimal FPGA implementation using low resources are discussed. Experimental results under different operating conditions are presented to demonstrate the good performance and the feasibility of the proposed controller for motor-drive applications.

Keywords: DC-link voltage balancing; field-oriented control; field-programmable gate array; multilevel active-clamped converter; motor drive

1. Introduction

The use of multilevel power converters for industrial applications has increased significantly in the last years thanks to their advantages compared to conventional two-level converters [1]. Some of these advantages are higher efficiency, higher power density, reduced harmonic distortion, etc. However, multilevel converters present some drawbacks, such as a higher number of switches and an increased control complexity. The higher control complexity is not just because they contain more devices. In some multilevel topologies, the DC-link bus is split into several partial voltages with the inclusion of capacitors. This implies necessary control actions to keep these capacitor voltages balanced.

This is the case, for instance, of the multilevel active-clamped (MAC) topology [2]. Figure 1 depicts the four switching states of a four-level MAC converter leg to illustrate the converter operation. The circled switches are on-state devices and the non-circled ones are off-state devices. The solid-line circled switches conduct the main current (i_o) and the dotted-line circled switches simply clamp the blocking voltage of the off-state devices to the voltage across adjacent levels. Compared to the commonly-used diode-clamped topology, which presents a lower number of switches, the MAC converter advantages are: lower conduction losses, improved switching-losses distribution, blocking voltage of a device always equal to the voltage across adjacent levels, and improved fault-tolerance capacity [3]. Motor drives, and in particular the traction inverter of electric vehicles, is one of the applications where the MAC converter appears to be of interest. Therefore, the authors propose to use a MAC converter to drive a permanent-magnet synchronous motor (PMSM).

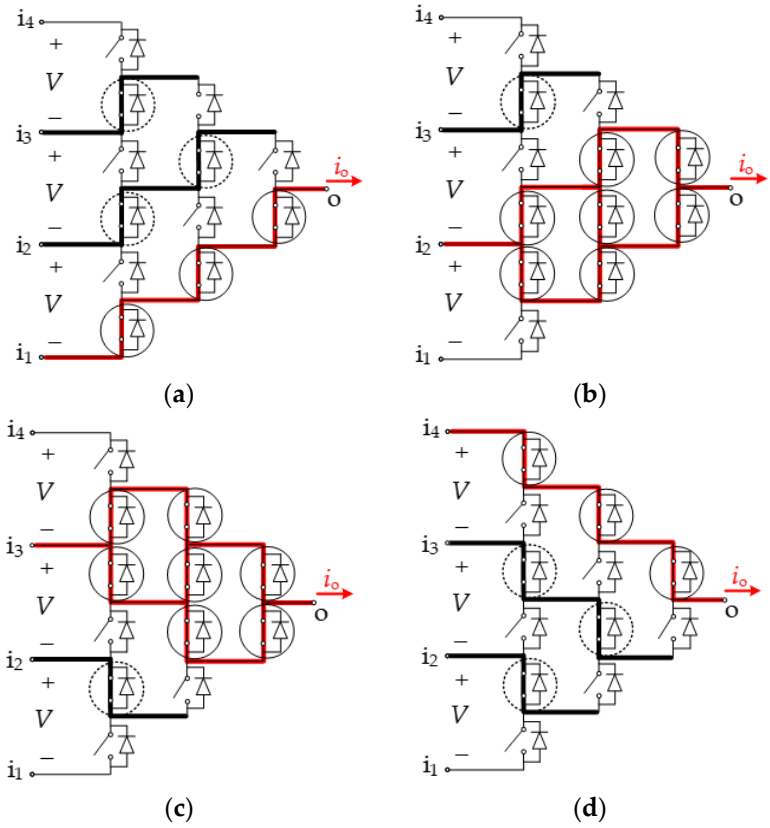


Figure 1. Four-level MAC leg switching states. (**a**) Connection to node i_1. (**b**) Connection to node i_2. (**c**) Connection to node i_3. (**d**) Connection to node i_4.

The MAC topology belongs to the family of neutral-point clamped (NPC) topologies [4]. In this family of topologies, one typical configuration consists of connecting a set of capacitors in series to passively generate the multiple voltage levels, see Figure 2. Other configurations are possible, as for instance connecting DC-voltage sources or batteries instead of capacitors across the adjacent input nodes. When DC-link capacitors are used, as in the present proposal, these topologies intrinsically present the challenge of balancing the capacitor voltages, since the classical modulation schemes lead to a voltage unbalancing. The voltage balancing problem arises from the existence of non-zero currents in the inner DC-link points (nodes i_2 and i_3 in Figure 1). This issue has been widely reported and investigated in the literature [5]. The diverse solutions proposed to solve this problem can be generally classified as hardware and software solutions. Hardware solutions introduce auxiliary circuitry to inject/draw additional current into/from the inner DC-link points to compensate the inherent converter current. Software solutions consist in defining a suitable modulation that is defined so as to maintain the average current of each inner node equal to zero over a specific period of time.

Among the software and hardware solutions, the authors propose to use a software solution since they are cheaper, present better performance, and they are simpler to implement. Among the different software solutions, the authors have selected the virtual-vector-based modulation originally introduced for three levels in [6] and extended to an arbitrary number of levels in [7], in which the average current of each inner DC-link point is maintained equal to zero over a single switching cycle. This modulation strategy enables to minimize the size of the capacitors, which leads to a higher power density. Although the applied modulation scheme is intended to preserve the balance of the DC-link capacitor voltages in every switching cycle, it is necessary to apply an additional control loop to guarantee a tight voltage balancing, since non-idealities lead to DC-link voltage unbalancing. The voltage balancing control (VBC) scheme proposed in [5] is implemented here to perform this action.

It is noteworthy that [8,9] state that the modulation scheme used here cannot work properly with dynamic loads such as motors, because the converter would not be able to keep the capacitor-voltage fluctuations low, leading to a system instability. This modulation is implemented here together with the VBC to drive a PMSM, demonstrating the feasibility of the proposed controller for motor-drive applications. Additionally, a four-level converter is used, in which the capacitor voltage balancing is much more challenging than in a three-level converter since some of the capacitor voltages may collapse [4].

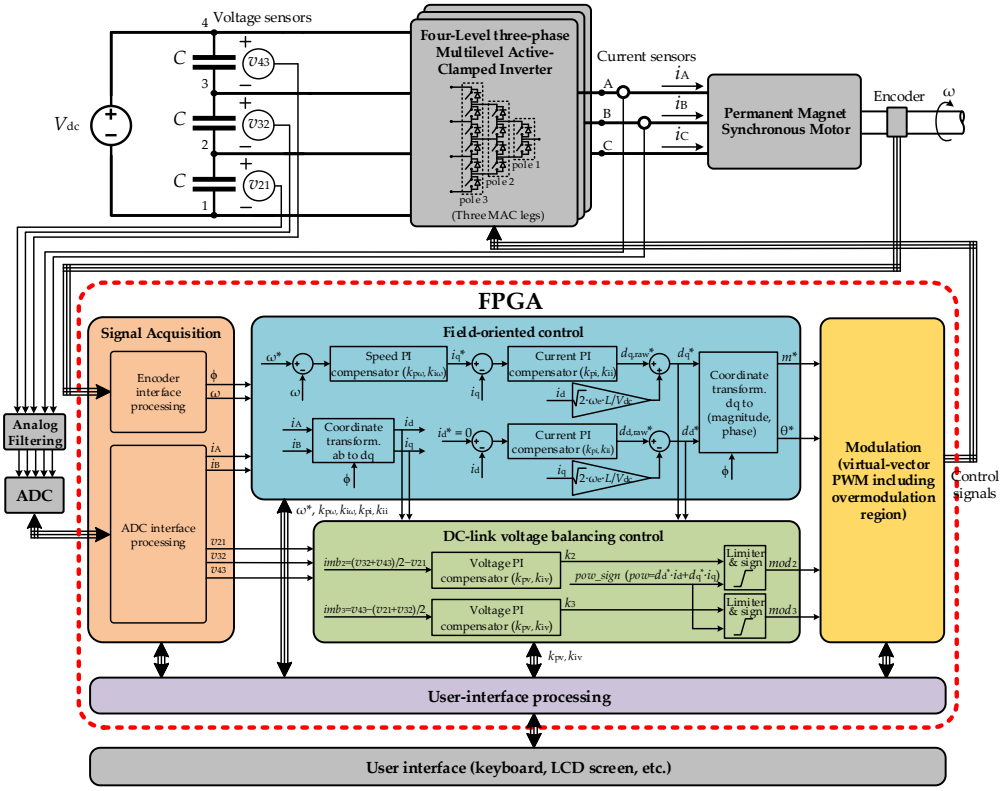


Figure 2. Global system overview.

In order to take full advantage of the traction inverter, a proper and proficient controller has to be developed. Typical power-converter digital controllers are implemented on microprocessors (μP), digital signal processors (DSP) and/or field-programmable gate arrays (FPGA). FPGA architecture permits both the parallel and sequential processing of data at high clock frequencies, which dramatically reduces the needed processing time, compared to μPs and DSPs. In addition, in cases in which the desired controller benefits from the utilization of a general-purpose processor, it can be embedded within the FPGA as many microprocessor cores are available as IPs.

The controller has to generate the 36 signals for the four-level MAC legs (12 devices per leg) at each switching cycle. Then, different automata with some duty ratios as inputs and running at high frequency (i.e., 50 MHz, which allows a time accuracy of 20 ns) have to be implemented. An FPGA implementation of such automata appears as a better solution than using a general-purpose processor with lots of timers. Additionally, for each new switching cycle, the new duty ratios have to be computed from the system input variables (currents, voltages, and rotor angle). For performing these calculations, it is desired to use measured values of the input variables as close as possible to the start of the next switching cycle, in order to maximize the control bandwidth. To this end, an ad hoc processing unit (specific purpose processing unit), implemented in the same FPGA as the above indicated automata, appears as a better solution than using an additional device, as a general-purpose processor or a DSP

platform. Delay between measurement and the application of the resulting duty ratios is reduced, cost is also reduced, and synchronization between different devices is not required.

Due to the general FPGA advantages and to the specific reasons presented above in the last two paragraphs, respectively, the authors propose a full FPGA-based control implementation of a four-level three-phase MAC inverter to drive a three-phase four-pole pairs PMSM. Figure 2 presents the general overview of the electrical circuit and the proposed FPGA control structure. As it can be seen in the Figure 2, as well as the VBC closed-loop control already introduced above, a closed-loop field-oriented control (FOC) is used for driving the PMSM.

A preliminary open-loop FPGA controller implementation with the same virtual-vector modulation was presented in [10]. However, the controller in [10] did not include the closed-loop controls to operate the converter as a motor drive and to preserve the capacitor voltages balanced.

FPGAs have been employed for implementing diverse control schemes of multilevel converters ([11–21]), and also for implementing motor-drive controllers ([17–26]). References [17–21] propose FPGA-based controllers for multilevel converters operating as motor drives, as it is proposed here. However, [17–21] do not explain the FPGA controller structure, do not discuss the design methodology to obtain an efficient implementation, and generally do not deal with the voltage balancing problem.

To the best of the authors' knowledge, for the first time, a complete controller for a four-level converter of the NPC converter family [4], operating as a motor drive and including DC-link voltage balancing control, is fully-embedded into an FPGA. In addition, the controller implementation has been optimized to save FPGA resources and also to take full advantage of the FPGA potential performance capabilities.

The paper is organized as follows: Section 2 presents a summary of the used VBC, field-oriented control (FOC), and modulation scheme, presenting the equations to be implemented in the FPGA. Section 3 details the FPGA structure, and describes relevant aspects to achieve an efficient controller. In Section 4, experimental results are shown to verify the good operation of the controller under different conditions. Finally, Section 5 outlines the conclusion.

2. Closed-Loop Control and Modulation Strategy

Figure 2 presents the overall closed-loop control structure applied to the MAC converter to drive the PMSM. Two autonomous control loops are implemented: FOC and VBC.

2.1. Field-Oriented Control (FOC)

The well-known FOC is used to control the three-phase PMSM. In the blue inset of Figure 2, the FOC structure is depicted. Variables ω and φ correspond to the measured rotor angular speed and rotor electrical angle, respectively. Variables i_d and i_q are the direct and quadrature components of the three-phase currents. Variables d_d* and d_q* are the direct and quadrature components of the normalized reference vector required by the modulator. Command values are designated with an asterisk superscript. The control inputs are: ω*, ω, φ, i_A, and i_B. The control scheme comprises an outer speed loop and an inner current loop. Through the PI compensators, the speed and current loops determine the reference vector polar coordinates m* and θ* to be the input to the modulator. Back-emf feedforward terms could be added in the current loops to improve the controller performance. They have not been implemented here for the sake of simplicity. The equations of the ab-to-dq and dq-to-(magnitude,phase) transformations are:

$$\begin{bmatrix} i_d \\ i_q \end{bmatrix} = \sqrt{2} \cdot \begin{bmatrix} \sin(\phi + 60^\circ) & \sin(\phi) \\ \cos(\phi + 60^\circ) & \cos(\phi) \end{bmatrix} \cdot \begin{bmatrix} i_A \\ i_B \end{bmatrix} \tag{1}$$

$$\begin{aligned} m^* &= \sqrt{\left(d_d^*\right)^2 + \left(d_q^*\right)^2} \\ \theta^* &= \tan^{-1}\left(d_q^*, d_d^*\right) + \phi \end{aligned} \tag{2}$$

Equation (1) assumes an isolated star point, where $i_A + i_B + i_C = 0$.

2.2. DC-Link Voltage-Balancing Closed-Loop Control (VBC)

The control scheme proposed in [5] is the one implemented here. In the green inset of Figure 2, the VBC structure is depicted. From the measured voltages v_{21}, v_{32}, and v_{43}, voltage imbalances associated to the two DC-link inner points (imb_2 and imb_3) are calculated. Then, through PI compensators, the values of variables k_2 and k_3 are determined. The sign of k_2 and k_3 depends on the direction of the converter power flow (*pow_sign*), calculated through the expression shown in Figure 2. Additionally, proper limits are set in k_2 and k_3 to avoid unfeasible dwell times [5]. Table 1 shows the limits and sign modification applied to k_2 and k_3. In this table, variable d_4 corresponds to an auxiliary variable that will be defined later in (4).

Table 1. Computation of variables k'_2 and k'_3.

Case	k'_2	k'_3
$pow\_sign = \text{sign}(k_2)$ $pow\_sign = \text{sign}(k_3)$	$\min\left\{0.5\,;\ \lvert k_2\rvert\,;\ \frac{1-d_4}{2\cdot d_4}\right\}$	$\min\left\{0.5\,;\ \lvert k_3\rvert\,;\ \frac{3\cdot(1-d_4)}{1+6\cdot d_4}\right\}$
$pow\_sign \neq \text{sign}(k_2)$ $pow\_sign \neq \text{sign}(k_3)$	$-\min\left\{0.5\,;\ \lvert k_2\rvert\,;\ \frac{3\cdot(1-d_4)}{1+6\cdot d_4}\right\}$	$-\min\left\{0.5\,;\ \lvert k_3\rvert\,;\ \frac{1-d_4}{2\cdot d_4}\right\}$
$pow\_sign \neq \text{sign}(k_2)$ $pow\_sign = \text{sign}(k_3)$	$-\min\left\{1\,;\ \lvert k_2\rvert\,;\ \frac{1.5\cdot(1-d_4)}{1+3\cdot d_4}\right\}$	$\min\left\{1\,;\ \lvert k_3\rvert\,;\ \frac{1.5\cdot(1-d_4)}{1+3\cdot d_4}\right\}$
$pow\_sign = \text{sign}(k_2)$ $pow\_sign \neq \text{sign}(k_3)$	$\min\left\{1\,;\ \lvert k_2\rvert\,;\ \frac{1-d_4}{2\cdot d_4}\right\}$	$-\min\left\{1\,;\ \lvert k_3\rvert\,;\ \frac{1-d_4}{2\cdot d_4}\right\}$

Finally, the preliminary leg duty-ratios are modified using variables k'_2 and k'_3, so that the balancing of the capacitor voltages can be recovered. This part is explained below at the end of the modulation strategy subsection.

2.3. Modulation Strategy

The modulation scheme originally introduced for three levels in [6], extended to an arbitrary number of levels in [7], and extended to the overmodulation region in [27], is the one used to operate the converter. This modulation, originally defined applying the virtual-vector concept, guarantees the dc-link capacitor voltage balance in every switching cycle, provided that the phase currents remain constant over the switching cycle and that their addition is equal to zero. The modulation assumes that the switching frequency ($f_s = 1/t_s$, where t_s is the switching period) is much larger than the fundamental frequency f. This PWM allows modulation index values $m \in [0, hbc \cdot 1.1027]$, where $m = v_{ab,1,pk}/V_{dc}$, $v_{ab,1,pk}$ is the peak value of the fundamental component of the line-to-line voltage, and hbc is the overmodulation hexagonal-boundary-compression index [27]. Therefore, the PWM covers both the undermodulation (UM) and overmodulation (OM) operating modes. The OM region is further divided into two subregions (OMI and OMII), which present different reference vector trajectories [27].

A comprehensive explanation of the used modulation scheme is presented in [7,27]. A simplified description showing the final equations that have to be implemented within the FPGA is presented below.

The modulation is implemented taking advantage of the hexagonal symmetry of the space vector diagram (SVD), optimizing the FPGA resources. Therefore, the command value of the reference vector angle ($\theta^* \in [0°, 360°[$), which has been calculated previously in the FOC control algorithm, is transformed into a sextant ($sextant \in \{0, 1, 2, 3, 4, 5\}$) and an angle within a sextant ($\theta_{sext}^* \in [0°, 60°[$).

The command values of modulation index (m^*) and reference vector angle (θ_{sext}^*) are modified in case the reference vector is located in the overmodulation region to obtain corrected values of

modulation index (m_c) and reference vector angle (θ_c). Tables 2 and 3 summarize these calculations. The index *hbc* is fixed to 0.98. Then, the range limits are:

$$\begin{aligned} hbc \cdot m^*_{\text{maxI}} &= 0.98 \cdot 3\ \ln(3)/\pi = 1.0281 \\ hbc \cdot m^*_{\text{maxII}} &= 0.98 \cdot 2\sqrt{3}/\pi = 1.0806 \end{aligned} \tag{3}$$

Table 2. Limiting reference vector angle θ_{lim} for overmodulation region.

Region	Application Range	θ_{lim}
UM	$0 < m^* \leq 0.98$	-
OMI	$0.98 < m^* \leq 1.0281$	$30^\circ \cdot \frac{1.0281 - m^*}{1.0281 - 0.98}$
OMII	$1.0281 < m^* \leq 1.0806$	$30^\circ \cdot \frac{m^* - 1.0281}{1.0806 - 1.0281}$

Table 3. Corrected values of modulation index (m_c) and reference vector angle (θ_c).

Region	Application Range	m_c	θ_c
UM	$0^\circ \leq \theta_{\text{sext}}{}^* < 60^\circ$	m^*	$\theta_{\text{sxt}}{}^*$
OMI	$0^\circ \leq \theta_{\text{sext}}{}^* < \theta_{\text{lim}}$	$0.98/\sin(\theta_{\text{lim}} + 60^\circ)$	$\theta_{\text{sext}}{}^*$
	$\theta_{\text{lim}} \leq \theta_{\text{sext}}{}^* \leq (60^\circ - \theta_{\text{lim}})$	$0.98/\sin(\theta_{\text{sext}}{}^* + 60^\circ)$	$\theta_{\text{sext}}{}^*$
	$(60^\circ - \theta_{\text{lim}}) < \theta_{\text{sext}}{}^* < 60^\circ$	$0.98/\sin(\theta_{\text{lim}} + 60^\circ)$	$\theta_{\text{sext}}{}^*$
OMII	$0^\circ \leq \theta_{\text{sext}}{}^* < \theta_{\text{lim}}$	$0.98/\sin(60^\circ) = 1.1316$	0°
	$\theta_{\text{lim}} \leq \theta_{\text{sext}}{}^* \leq (60^\circ - \theta_{\text{lim}})$	$0.98/\sin(\theta_{\text{sext}}{}^* + 60^\circ)$	$\theta_{\text{sext}}{}^*$
	$(60^\circ - \theta_{\text{lim}}) < \theta_{\text{sext}}{}^* < 60^\circ$	$0.98/\sin(60^\circ) = 1.1316$	60°

From m_c and θ_c, the auxiliary variables d_1, d_4 and d_5 are calculated as follows:

$$\begin{aligned} d_1 &= m_c \cdot \cos\left(\theta_c + 30^\circ\right) \\ d_4 &= m_c \cdot \cos\left(\theta_c - 30^\circ\right) \\ d_5 &= d_4 - d_1 \end{aligned} \tag{4}$$

The preliminary leg duty ratios d_{x1} and d_{x4} (indicating the duty ratio of connection of phase x to levels 1 and 4, respectively) are determined according to Table 4 from the value of the sextant of each phase *sextant_x*, and from the auxiliary variables d_1, d_4 and d_5.

Table 4. Leg duty ratios of Levels 1 and 4 depending on the sextant.

sextant_x	0	1	2	3	4	5
$\boldsymbol{d_{x1}}$	0	d_5	d_4	d_4	d_1	0
$\boldsymbol{d_{x1}}$	d_4	d_1	0	0	d_5	d_4

The leg duty ratios of the inner levels 2 and 3 are then calculated as follows:

$$d_{x2} = d_{x3} = (1 - d_{x1} - d_{x4})/2 \tag{5}$$

In order to finally implement the VBC, preliminary leg duty ratios are modified according to the following equations:

$$\begin{aligned} d'_{x1} &= d_{x1} \cdot (1 - k'_2 - k'_3) \cdot k_{\text{mod}} \\ d'_{x2} &= 0.5 + k'_2 \cdot k_{\text{mod}} \cdot (d_{x1} - d_{x4}) - 0.5 \cdot d_4 \cdot k_{\text{mod}} \\ d'_{x4} &= d_{x4} \cdot (1 + k'_2 + k'_3) \cdot k_{\text{mod}} \\ d'_{x3} &= 1 - d'_{x1} - d'_{x2} - d'_{x4} \end{aligned} \tag{6}$$

where $k_{\mathrm{mod}} = 3/(3 + k'_2 - k'_3)$.

3. FPGA Design and Control Implementation

Modulation and control structures presented in the previous Section have been implemented on an Altera Cyclone IV EP4CE22F17C6N FPGA device driven by a 50-MHz system clock (Altera, Intel, San Jose, CA, USA). The FPGA application has been described in VHDL. Figure 3 presents a block diagram of the FPGA application, together with its peripherals. In this figure, six different subsystems are separated in a set of six color boxes, following the same color selection as in Figure 2. Input signals are located on the left side of the boxes, while output signals are located on the right side. This criterion does not apply to the signals exchanged between the FPGA and peripherals, in which an arrow indicates the direction. For simplicity, only the main variables and constants are shown.

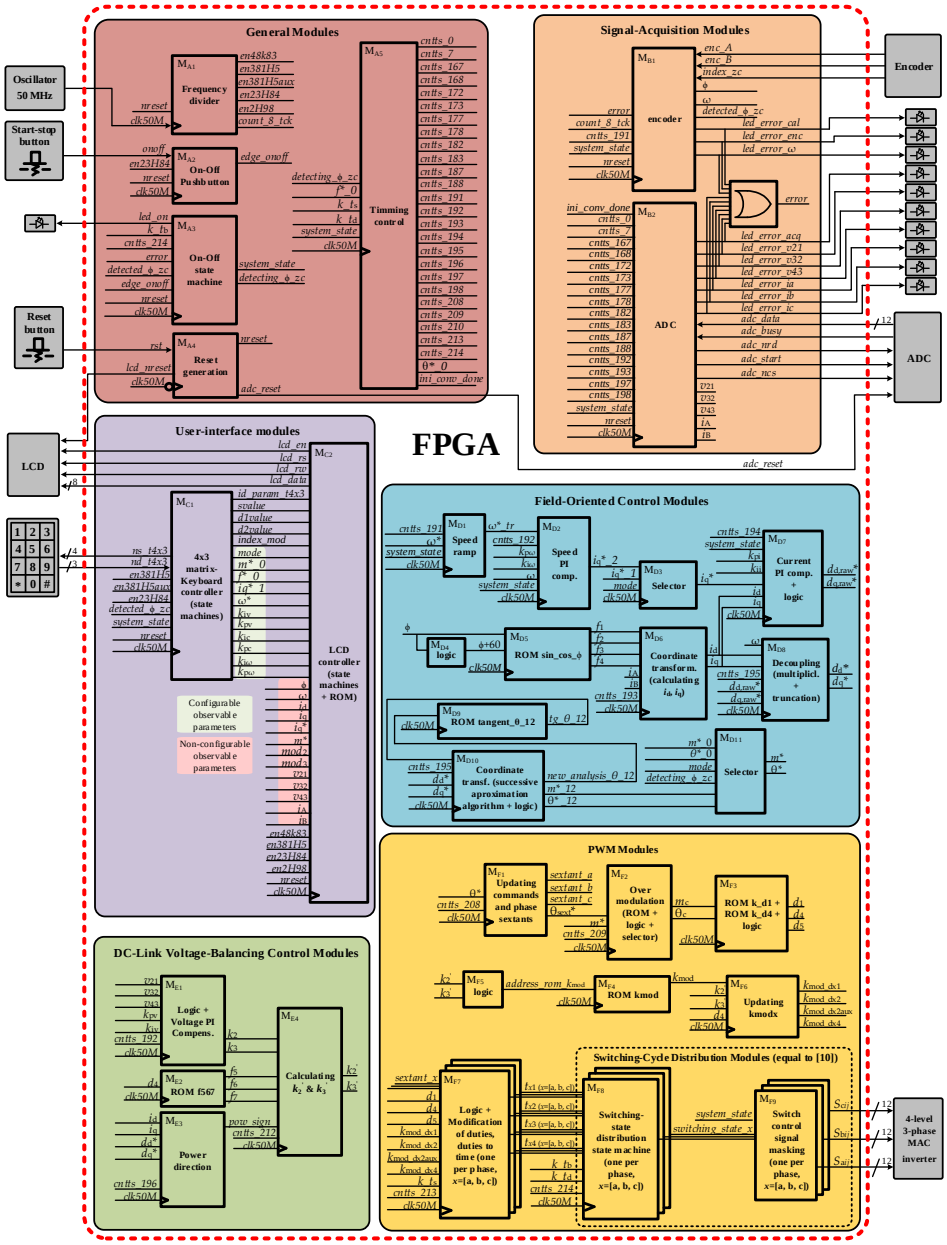


Figure 3. FPGA-controller design overview.

The FPGA has been mounted on a printed circuit board together with the sensors, the filtering circuitry, the ADC chip, and the user-interface (see Figure 3). The user-interface comprises a 4 × 3 matrix keyboard to introduce the values of some control variables, a three-line sixteen-character LCD screen, two pushbuttons, and eleven LEDs to easily visualize some errors and the ON-state.

Three operating modes have been defined in order to enable the use of the inverter under different system configurations, and also to bring the possibility of evaluating and tuning the different control structures individually:

- Voltage-balancing controller mode (mode 0): in this mode, FOC is disabled, but the dc-link voltage control is enabled. With this mode, it is necessary to introduce the command values of m^* and f^* to make it operate at desired conditions.
- Torque controller mode (mode 1): this mode enables the inner current control loop of the FOC, but the outer speed control loop is disabled. DC-link voltage control is also enabled. With this mode, it is necessary to introduce the value of i_q^*, which is proportional to the torque, to make it operate at desired conditions.
- Speed controller mode (mode 2): this mode enables the whole closed-loop control. With this mode, it is necessary to externally introduce the command value of rotational speed ω^* to make it operate at desired conditions.

Since the control scheme requires five analog inputs (sensed i_A, i_B, v_{21}, v_{32}, and v_{43}), and the position of the rotor, an analog-to-digital converter chip and an encoder are necessary.

The AD7658 from Analog Devices has been selected as ADC chip, as it allows converting up to six analog signals synchronously. This chip is configured in parallel interface to minimize the time needed for data transmission.

Encoder RP1410 (IFM, El Prat de Llobregat, Spain) is used for obtaining the rotor position. This encoder generates three digital signals: *enc_A*, *enc_B*, and *index_zc*. Signals *enc_A* and *enc_B* present each 1024 pulses for a rotor revolution (with a phase delay of 90°). To determine the rotor angle, any single rising or falling edge of *enc_A* and *enc_B* can be counted, giving a resolution of 4096 edges per revolution, so 1024 edges per electrical cycle (the motor is a four-pole PMSM). Then, the mechanical rotor angle φ_m presents a resolution of 0.08789° (360°/4096 edges), and the electrical angle φ resolution is 0.35156°. Signal *index_zc* presents a small pulse each time a rotor revolution is accomplished (zero-crossing detection).

3.1. FPGA Basic Design Aspects

A similar design methodology to the one used in [10] has been also considered here. Variables are represented as integers. For each variable, both units and width (number of bits) have been meticulously selected to obtain the desired resolution and range avoiding underflows/overflows and to reduce FPGA used resources (truncations are often applied). Most of the variables are coded with 12 to 16 bits. Better resolution is not required as sampled variables (i.e., voltages, currents, and rotor position) are acquired just with a 12-bit resolution. Unsigned variables are usually coded in natural binary, while signed variables are coded in two's complement or sign plus magnitude when required. Divisions by constants (i.e., when changing units or calculating speed) are replaced by products followed by truncations to take advantage of the multipliers integrated in the FPGA. Other divisions are avoided as much as possible. When required, as well as for transcendental functions, they are implemented by ROMs built with the RAM blocks integrated in the FPGA. Symmetries, scaling, and offset addition are applied whenever is possible to minimize the number of bits required to achieve a certain resolution. More details are explained below in the FPGA-module-description subsection.

A 50-MHz system clock clk50M (period t_{ck} = 20 ns) is used to manage the FPGA. Relevant time variables, as delay time $t_d = 2\ t_{ck}$, blanking time $t_b = 40\ t_{ck}$, and switching period $t_s = 10{,}000\ t_{ck}$ are defined as multiples of t_{ck}, to optimize FPGA resources. Note that t_s = 200 µs implies f_s = 5 kHz. Note

in Figure 3 the time constants k_t_b, k_t_d, and k_t_s, that are defined as $k\_t_b = t_b/t_{ck} = 2$, $k\_t_d = t_d/t_{ck} = 40$, and $k\_t_s = t_s/t_{ck} = 10{,}000$.

As in [10], most of the FPGA processing is synchronized with the switching cycle. A divider-by-10,000 counter *cnt_ts* is used to this purpose. Figure 4 illustrates the timing overview of a *cnt_ts* cycle and its synchronization with a switching period. A *cnt_ts* cycle starts performing all the samplings and calculations required to stablish the behavior of the next switching cycle, which starts at *cnt_ts* = 234, as soon as possible after these calculations are completed at *cnt_ts* = 214. The delay between these two events, half a blanking time, is required to properly transit from a switching cycle to the next one (d_{x4} can change from a null/non-null value to a new null/non-null value, as explained in [10]).

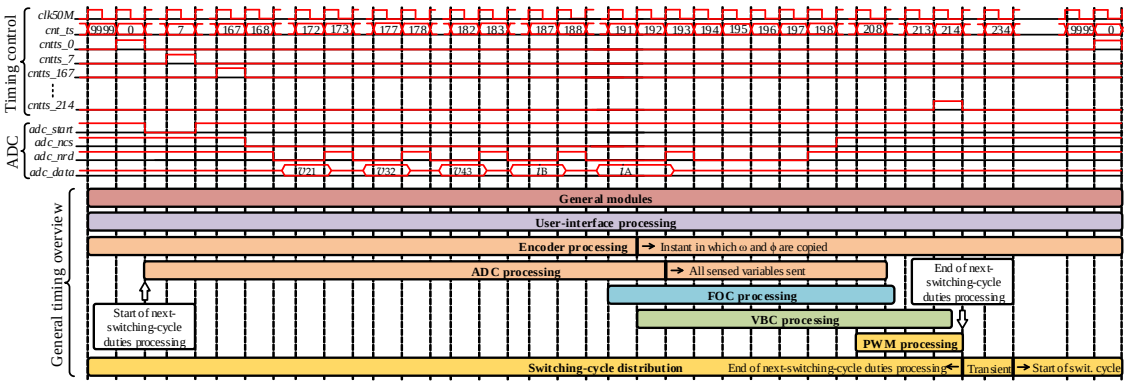


Figure 4. FPGA timing overview.

A versatile implementation has been conceived, allowing the user to modify important operational parameters through the user interface, such as the operating mode (mode), the PI compensators constants $k_{p\omega}$, $k_{i\omega}$, k_{pi}, k_{ii}, k_{pv} and k_{iv}, or the command values ω^*, i_q^*, m^* and f^*.

3.2. FPGA Module Description

3.2.1. General Modules

Module M_{A1} generates periodic enable signals of lower frequencies that are necessary to manage the periodicity of several processes. Enable signals are active high, with a pulse lasting a single t_{ck}, which is repeated at the corresponding frequency. For example, enable signal *en48k83* (48.83 kHz) is used for handling the writing process of LCD, and enable signal *en381H5* (381.5 Hz) is used for the state machine of keyboard inspection and for the state machine of the LCD screen operation. Last output signal *count_8_tck* is a 3-bit divider, which is used, together with the remaining enable signals, in other processes.

Module M_{A2} detects an active edge in signal *onoff* provided by the "ON-OFF" pushbutton, and generates an active high signal *edge_onoff*, that lasts a single t_{ck}. Pushbutton inspection is done at low frequency (23.84 Hz) for filtering possible pushbutton bounces.

Module M_{A3} implements the system ON-OFF finite state machine, which is presented in Figure 5. Binary values in each state indicate the *system_state* output of M_{A3}. From right to left, bit 0 represents the transition-and-holding bit between OFF and ON states, and bits 1, 2, and 3 represent masking bits for switches in poles 1, 2 and 3, with reference to Figure 2. In the transition from OFF state to ON state, first pole 1 is enabled, then pole 2, and lastly pole 3. This sequence is reversed in the transition from ON to OFF. When the FPGA is powered on, the motor is driven to rotate at a very low speed in mode 0 ($m^* = 0.04$ and $f^* = 1.22$ Hz) to allow an automatic initial zero-crossing detection of the rotor position. Signal *detecting_φ_zc* is activated when this automatic detection process is going on. When a filtered first pulse of encoder signal *index_zc* is detected, signal *detected_φ_zc* is activated permanently

(this part of the process is carried out by module M_{B1}), the motor is stopped, and signal *detecting_φ_zc* is deactivated.

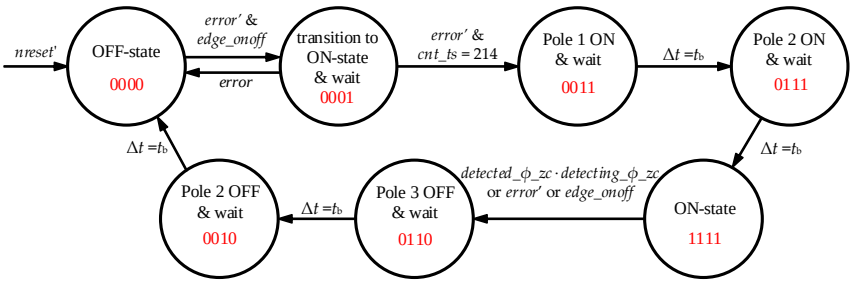


Figure 5. ON-OFF finite state machine.

Module M_{A4} generates the global reset signal *nreset* from the signal *rst* provided by the "reset" pushbutton. Signal *nreset* is active low, lasting 120 μs to guarantee minimum duration required by LCD (*lcd_nreset* and *nreset* are the same signal). Signal *adc_reset*, which lasts 10 t_{ck} and is active high, is used to initialize the ADC chip.

Module M_{A5} is the main synchronization module. It implements the divider-by-10,000 counter *cnt_ts* and generates a set of enable signals (*cntts_X*) to manage the timing of most actions performed by other modules, as indicated in Figure 4. This module also updates the value of line angle in mode 0 (*θ*_0*), from the frequency command value in mode 0 (*f*_0*).

3.2.2. Signal-Acquisition Modules

Module M_{B1} processes the encoder input signals *enc_A*, *enc_B*, and *index_zc* and generates the angular position φ_m of the motor, its angular speed ω, and the electrical angle φ. Input signals from the encoder are initially filtered at a sampling frequency of 6.25 MHz (50 MHz/8), discarding any value that has not remained constant for a minimum of eight consecutive samples. The angular position φ_m is updated when a new edge in any of the filtered signals derived from *enc_A* and *enc_B* is detected. A pulse of *index_zc* signal should take place every 4096 edges (1 revolution). Non-consistency produces a LED error indication. The angular speed ω is updated every 2.5 ms, and is calculated dividing the angle rotated in the last 10 ms by this time. The existence of two simultaneous edges in filtered signals derived from *enc_A* and *enc_B*, as well as an excessive speed, also result in LED error indications. Obviously, this module works asynchronously to the main synchronization module. To synchronize data, φ and ω are copied at *cnt_ts* = 191 to have convenient values for FOC and VBC processing (see Figure 4).

Module M_{B2} is in charge of handling the ADC chip according to Figure 4. Signals *adc_start*, *adc_ncs* and *adc_nrd* are generated in the FPGA to control the ADC chip. FPGA receives the acquired data through the 12-bit bus *adc_data*. Current i_c is determined from the measured ones ($i_c = -i_a - i_b$). Values out of acceptable range result in LED error indications.

3.2.3. User-Interface Modules

Module M_{C1} comprises two finite state machines that handle the 4 × 3 matrix keyboard operation. A first finite state machine handles the detection and identification of the keyboard buttons when any of them is pressed. A second finite state machine identifies the sequence of the different buttons pressed in order to determine the actions to be executed: acquisition of the identifier of the parameter to be shown in the LCD screen, modification of the value of a configurable parameter, etc.

Module M_{C2} implements three concurrent finite state machines that manage the configuration and visualization of the LCD screen. The LCD screen shows the name of the variable selected by the user to be visualized, with its current value, its identifier number and its allowed range (just for the

configurable parameters). The module also includes a ROM containing the visualization format for the 24 variables that can be visualized in the LCD screen (see Figure 3).

The user-interface modules consume substantial resources of the FPGA (21.5% of the used logic elements and 27% of the used multipliers). Its explanation is simplified here because they are considered to be of less technical importance, compared to other parts of the design.

3.2.4. FOC Modules

Module M_{D1} simply generates a limiting speed ramp whenever the speed command changes. Module M_{D2} consists of the speed-error PI compensator with limited proportional, integral and total outputs. Its output value is the quadrature current command value in mode 2 (i_q*_2), which is updated when *cnt_ts* = 192. In mode 1, the quadrature current command value (i_q*_1) is set externally. Module M_{D3} is a selector of i_q* depending on the chosen operating mode.

Modules $M_{D4,}$ M_{D5}, and M_{D6} are in charge of implementing the coordinate transformation. Module M_{D4} implements simple logic to calculate angle $\varphi + 60°$. Module M_{D5} implements a dual-port ROM that allows obtaining the following four trigonometrical functions in a single t_{ck}, which are necessary to determine the values of i_d and i_q:

$$\begin{aligned} f_1 &= \sqrt{2}\ \sin(\phi) \\ f_2 &= \sqrt{2}\ \cos(\phi) \\ f_3 &= \sqrt{2}\ \sin(\phi + 60^\circ) \\ f_4 &= \sqrt{2}\ \cos(\phi + 60^\circ) \end{aligned} \tag{7}$$

Module M_{D6} implements Equation (1) to calculate the values of i_d and i_q. Module M_{D7} contains the i_d and i_q error PI compensators with limited proportional, integral and total outputs. Output variables of PI compensators are direct and quadrature raw duties $d_{d,raw}$* and $d_{q,raw}$*, which are updated when *cnt_ts* = 194.

Module M_{D8} implements the calculation of the decoupling factor, which is equal to $\sqrt{2} \cdot \omega_e \cdot L / V_{dc}$ (where L and V_{dc} are usually constant values), its products by currents i_d and i_q, the addition of the first product to $d_{q,raw}$* and the subtraction of the second product from $d_{d,raw}$*, to obtain duties d_d* and d_q*, respectively (see Figure 2). In general, products are done followed by a truncation (change of units), thus allowing optimizing FPGA resources.

Modules M_{D9} and M_{D10} implement the dq-to-(magnitude,phase) transformation indicated in Equation (2) to calculate the modulation index and reference vector angle in modes 1 and 2 (m*_12 and θ*_12). Both variables are obtained through a successive approximation algorithm. A ROM is used to calculate the tangent of the provisional reference vector angle used in the algorithm. Octagonal symmetry is taken into account to reduce the size of the ROM.

Module M_{D11} is just a selector for the modulation index and the reference vector angle, depending on the operating mode.

3.2.5. VBC Modules

Module M_{E1} calculates the voltage imbalances at dc-link points 2 and 3 (imb_2 and imb_3) and includes the PI compensators for the error of these variables, also calculated in the module.

Module M_{E2} consists of a ROM delivering the following auxiliary functions, which are necessary to determine the values of k'_2 and k'_3, as shown in Table 1:

$$\begin{aligned} f_5 &= \min\{1\ ;\ (1 - d_4)\ /\ (2 \cdot d_4)\} \\ f_6 &= \min\{0.5\ ;\ 3 \cdot (1 - d_4)\ /\ (1 + 6 \cdot d_4)\} \\ f_7 &= \min\{1\ ;\ 1.5 \cdot (1 - d_4)\ /\ (1 + 3 \cdot d_4)\} \end{aligned} \tag{8}$$

Module M_{E3} is in charge of calculating the sign of power flow, according to the expression shown in the green inset of Figure 2. Output value of *pow_sign* is updated at *cnt_ts* = 196.

Module M_{E4} applies simple logic to obtain values of variables k'_2 and k'_3, according to Table 1.

3.2.6. Modulation Modules

Module M_{F1} adapts the value of the reference vector angle to be an angle within the first sextant ([0°, 60°[). The sextant values corresponding to the three phases are also determined, and given through the variables *sextant_x*. Module M_{F2} modifies θ and m^*, according to Tables 2 and 3. A ROM is used to calculate the function $m = 0.98/\sin(\theta + 60°)$. Module M_{F3} calculates the values of d_1, d_4 and d_5, according to Equation (4). Two ROMs are used to calculate d_1 and d_4.

Module M_{F4} includes a ROM for calculating the value of k_{mod}, according to Equation (6). The ROM address, which is defined as the result of expression $3 + k'_2 - k'_3$, is previously calculated through module M_{F5}. Module M_{F6} calculates the following products, which are useful to calculate modified duty ratios d'_{x1}, d'_{x2}, d'_{x3}, and d'_{x4}:

$$\begin{aligned} k_{mod\_dx1} &= (1 - k'_2 - k'_3) \cdot k_{mod} \\ k_{mod\_dx2} &= k'_2 \cdot k_{mod} \\ k_{mod\_dx2aux} &= 0.5 \cdot d_4 \cdot k_{mod} \\ k_{mod\_dx4} &= (1 + k'_2 + k'_3) \cdot k_{mod} \end{aligned} \quad (9)$$

Module M_{F7} (one per phase) calculates d'_{x1}, d'_{x2}, d'_{x3}, and d'_{x4} according to Equation (6) and Table 4, and transforms them into time variables t_{x1}, t_{x2}, t_{x3} and t_{x4} as multiples of t_{ck}. Modules M_{F8} and M_{F9} (one per phase) are the same used in [10].

3.3. Consumed FPGA Resources

The resources used to synthesize the whole design using the Quartus II software are shown in Table 5. The complete FPGA implementation has been done making a substantial effort to save FPGA resources. As it can be seen, enough FPGA resources are still available. They can be used, for example, to include additional features in the design, to increase the switching frequency, and/or to improve the variables resolution. Furthermore, it is relevant to recall that, for practical applications, the user-interface processing is usually not required, which would increase even more the remaining FPGA resources.

Table 5. Consumed FPGA resources.

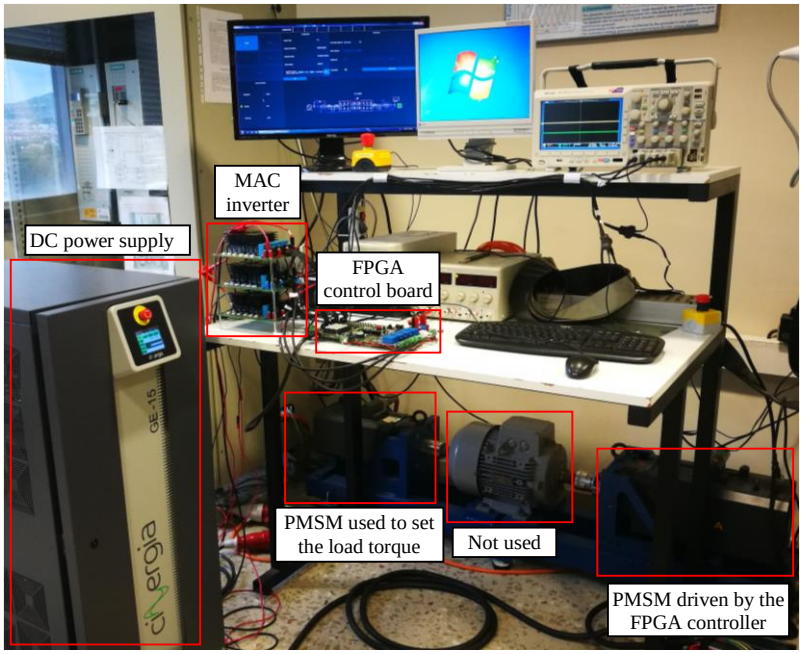
Resource	Amount Used/Total Available
Logic elements	7272/22,320 (33%)
Combinational functions	7156/22,320 (32%)
Dedicated logic registers	1183/22,320 (5%)
Pins	89/154 (58%)
Memory bits	285,056/608,256 (47%)
Embedded Multiplier 9-bit elements	74/132 (56%)
PLLs	0/4 (0%)

3.4. Control Processing Time

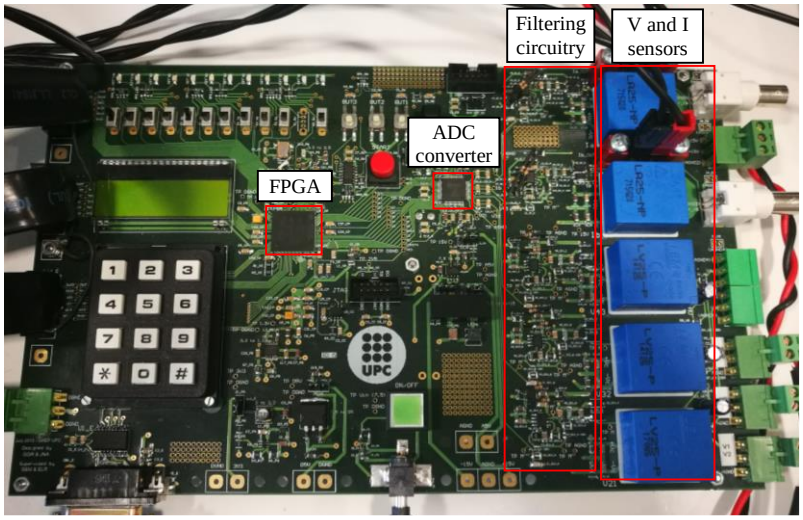
In order to minimize the processing time for obtaining the next-switching-cycle duties and therefore maximize the closed-loop control bandwidth, an ad hoc processing unit has been fully embedded within the FPGA, instead of using a general-purpose processor. The resulting processing time, taken from the command to sample and convert the analog signals delivered by the sensors (*cnt_ts* = 0), until the instant when all the duties needed to generate the following switching cycle become available (*cnt_ts* = 214), is 4.28 µs (see Figure 4). Thus, the switching frequency could be increased until approximately 1/5 µs = 200 kHz. To maximize the control bandwidth, the processing occurs as close as possible to the start of a new switching cycle.

4. Experimental Tests

The proper operation of the FPGA controller has been tested experimentally within the system shown in Figure 6a. Main parts of the system are labelled in the figure. The shaft of the PMSM driven by the FPGA controller is coupled to an induction machine, which is not used, and also to another PMSM, which is used to set the load torque. A further detailed overview of the experimental test bed can be observed in a supplementary video attached with this paper. Figure 6b depicts FPGA control board, presenting its main parts/subcircuits.



(**a**)



(**b**)

Figure 6. Experimental testbed. (**a**) Overview of the system; (**b**) FPGA control board.

The four-level three-phase MAC inverter employs 200 V STP20NF20 MOSFET devices (ST microelectronics, Amsterdam, Netherlands). The reference of the PMSM driven by the FPGA controller is 1FT6105-8SB71-2AA0 (Siemens, Berlin, Germany). This motor is a surface-magnet type PMSM, with a nominal speed of 1500 rpm and a nominal torque of 59 Nm, thus, a nominal power of 9.27 kW.

The proper system operation can be observed in Figures 7–9. In Figure 7, the good performance of the VBC under a start-up transition is depicted. Initially, in OFF state, capacitor voltages are unbalanced, but after 200 ms of operation, they become balanced. It is also remarkable the behavior of the phase current i_A. After the start-up, while the motor is accelerating, the current magnitude keeps constant at a certain level, set by the speed ramp. As soon as the motor reaches the command speed of 500 rpm, the current magnitude decreases to a lower level.

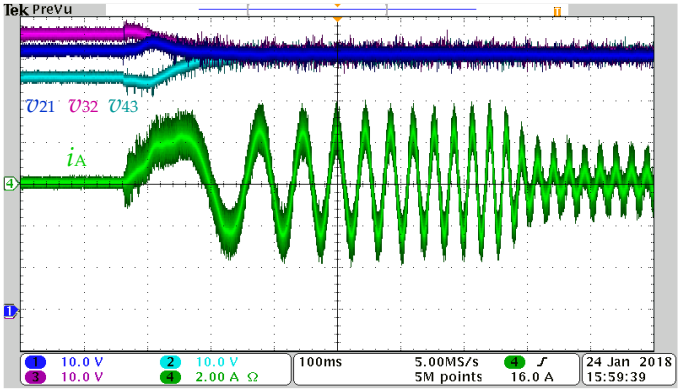


Figure 7. Experimental results of DC-link voltages and phase current i_A under a start-up transition. Conditions: V_{dc} = 180 V, C = 155 µF, ω^* = 500 rpm, $k_{p\omega}$ = 0.1 A/rpm, $k_{i\omega}$ = 0.1 A/(rpm·s), k_{pi} = 0.01 A^{-1}, k_{ii} = 1 $(A{\cdot}s)^{-1}$, k_{pv} = 0.02 V^{-1}, k_{iv} = 0 $(V{\cdot}s)^{-1}$, t_s = 100 µs, load torque = 0 Nm.

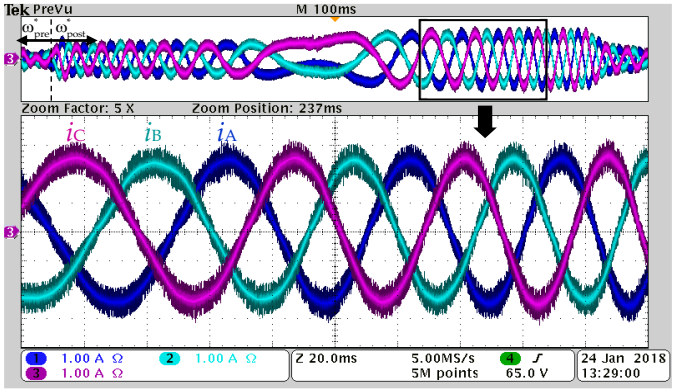


Figure 8. Experimental results of phase currents i_A, i_B, and i_C, under a change of the rotation direction. Conditions: ω^*_{pre} = 350 rpm, ω^*_{post} = −350 rpm (remaining conditions are the same as in Figure 7).

Figure 8 shows the phase currents under a change of rotation direction, from 350 rpm to −350 rpm. In the upper part of the figure, the whole transition is depicted. As it can be seen, the current sequence changes once the motor starts rotating in opposite direction. As in Figure 7, once the motor reaches the command speed value, the current magnitude decreases to a lower level.

Figure 9 presents the three phase currents and the phase voltage v_{A1} operating in UM and OM regions. Figure 9a shows the waveforms for the UM region, with m^* = 0.76. Figure 9b shows the waveforms for region OMI (0.98 < m^* < 1.028), operating with m^* = 1.01. Figure 9c shows the waveforms for region OMII (1.028 < m^* < 1.081), operating with m^* = 1.03. In UM, current waveforms are sinusoidal with almost no distortion. However, in OMI and OMII, the current waveforms present a noteworthy distortion, with a higher distortion under OMII. This is the expected behavior, since the overmodulation intrinsically introduces low-order harmonics in the phase currents [27]. It is also interesting to note that in waveforms of the phase voltage v_{A1}, the duty ratio of connection to inner levels 2 and 3 is lower when operating in OM region.

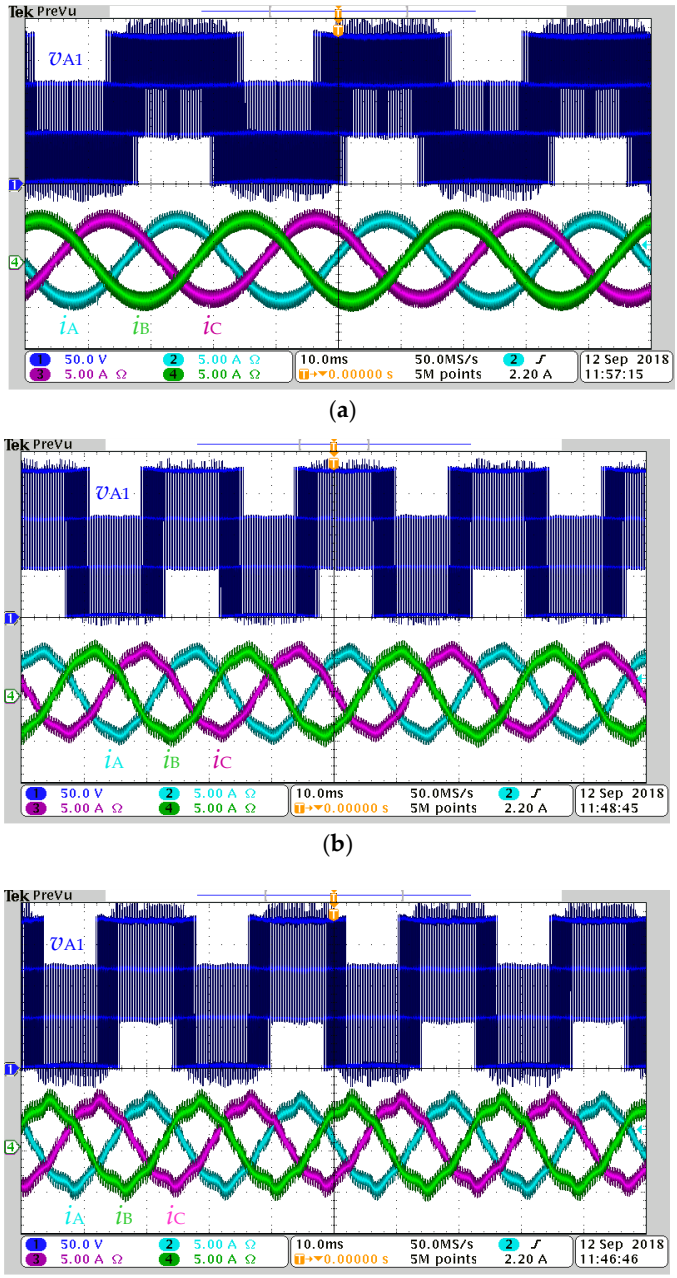


Figure 9. Experimental results of phase currents i_A, i_B, and i_C, and phase voltage v_{1A}, operating in UM and OM regions. (**a**) Operation in UM: ω^* = 450 rpm, load torque = 10 Nm, m^* = 0.76; (**b**) Operation in OMI: ω^* = 610 rpm, load torque = 10 Nm, m^* = 1.01; (**c**) Operation in OMII: ω^* = 620 rpm, load torque = 10 Nm, m^* = 1.03 (remaining conditions are the same as in Figure 7 for (**a**), (**b**) and (**c**)).

As stated previously, a complementary video is included with this study. The video shows the whole system operating under conditions of Figures 7 and 8, and also under other different conditions.

5. Conclusions

A low-cost closed-loop controller fully embedded into an FPGA has been successfully implemented for a PMSM motor drive based on a four-level three-phase MAC inverter, taking full advantage of the drive potential performance capabilities. An efficient and robust implementation into a mid-range FPGA of the closed-loop VBC and FOC, together with a modulation scheme including the overmodulation region, has been achieved, consuming less than 50% of its total resources and

obtaining a very low processing time. The remaining FPGA resources can be employed to increase the switching frequency, or to further improve the controller performance, including, for example, fault-tolerant controls [3], or an intelligent distribution of switching losses to better distribute the total semiconductor losses [2]. The proper operation of the whole system demonstrates the feasibility of using virtual-vector-based PWMs for neutral-point-clamped converters in motor drive applications, which had been questioned in the previous literature. In the future, the authors envision a motor drive design approach where an inexpensive switch with good performance is selected, and MAC leg structures are used to match the motor voltage rating by simply adjusting the number of levels.

Supplementary Materials: The following are available online at http://www.mdpi.com/1996-1073/11/10/2639/s1. A supplementary video included with this study shows the whole system operating under different conditions.

Author Contributions: Conceptualization, J.N.-A., E.L., S.B.-M. and J.B.; Funding acquisition, S.B.-M.; Investigation, J.N.-A., E.L. and A.C.; Methodology, J.N.-A., E.L. and S.B.-M.; Project administration, S.B.-M.; Software, E.L., A.C. and G.G.-R.; Supervision, S.B.-M., A.C. and J.B.; Validation, A.C. and G.G.-R.; Writing—original draft, J.N.-A.

Funding: This research was funded by Ministerio de Ciencia, Innovación y Universidades under grant DPI2017-89153-P.

Acknowledgments: The authors would like to thank Miquel Teixidor and his colleagues from CINERGIA, for their useful training on the usage of the GE15 grid emulator, which has been used in this study as a DC power supply.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Leon, J.I.; Vazquez, S.; Franquelo, L.G. Multilevel Converters: Control and Modulation Techniques for Their Operation and Industrial Applications. *Proc. IEEE* **2017**, *105*, 2066–2081. [CrossRef]
2. Busquets-Monge, S.; Nicolas-Apruzzese, J. A Multilevel Active-Clamped Converter Topology—Operating Principle. *IEEE Trans. Ind. Electron.* **2011**, *58*, 3868–3878. [CrossRef]
3. Nicolas-Apruzzese, J.; Busquets-Monge, S.; Bordonau, J.; Alepuz, S.; Calle-Prado, A. Analysis of the Fault-Tolerance Capacity of the Multilevel Active-Clamped Converter. *IEEE Trans. Ind. Electron.* **2013**, *60*, 4773–4783. [CrossRef]
4. Busquets-Monge, S. Neutral-Point-Clamped DC-AC Power Converters. In *Wiley Encyclopedia of Electrical and Electronics Engineering*; Webster, J.G., Ed.; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2018; pp. 1–20. ISBN 9780471346081.
5. Busquets-Monge, S.; Maheshwari, R.; Nicolas-Apruzzese, J.; Lupon, E.; Munk-Nielsen, S.; Bordonau, J. Enhanced DC-Link Capacitor Voltage Balancing Control of DC-AC Multilevel Multileg Converters. *IEEE Trans. Ind. Electron.* **2015**, *62*, 2663–2672. [CrossRef]
6. Busquets-Monge, S.; Bordonau, J.; Boroyevich, D.; Somavilla, S. The nearest three virtual space vector PWM—A modulation for the comprehensive neutral-point balancing in the three-level NPC inverter. *IEEE Power Electron. Lett.* **2004**, *2*, 11–15. [CrossRef]
7. Busquets-Monge, S.; Alepuz, S.; Rocabert, J.; Bordonau, J. Pulsewidth Modulations for the Comprehensive Capacitor Voltage Balance of n-Level Three-Leg Diode-Clamped Converters. *IEEE Trans. Power Electr.* **2009**, *24*, 1364–1375. [CrossRef]
8. Choudhury, A.; Pillay, P.; Williamson, S.S. A Hybrid PWM-Based DC-Link Voltage Balancing Algorithm for a Three-Level NPC DC/AC Traction Inverter Drive. *IEEE J. Emerg. Sel. Top. Power Electron.* **2015**, *3*, 805–816. [CrossRef]
9. Choudhury, A.; Pillay, P.; Williamson, S.S. DC-Bus Voltage Balancing Algorithm for Three-Level Neutral-Point-Clamped (NPC) Traction Inverter Drive with Modified Virtual Space Vector. *IEEE Trans. Ind. Appl.* **2016**, *52*, 3958–3967. [CrossRef]
10. Lupon, E.; Busquets-Monge, S.; Nicolas-Apruzzese, J. FPGA Implementation of a PWM for a Three-Phase DC-AC Multilevel Active-Clamped Converter. *IEEE Trans. Ind. Inform.* **2014**, *10*, 1296–1306. [CrossRef]
11. Cossutta, P.; Aguirre, M.P.; Engelhardt, M.A.; Valla, M.I. Control System to Balance Internal Currents of a Multilevel Current-Source Inverter. *IEEE Trans. Ind. Electron.* **2018**, *65*, 2280–2288. [CrossRef]

12. Zhang, Z.; Wang, F.; Wang, J.; Rodríguez, J.; Kennel, R. Nonlinear Direct Control for Three-Level NPC Back-to-Back Converter PMSG Wind Turbine Systems: Experimental Assessment With FPGA. *IEEE Trans. Ind. Inform.* **2017**, *13*, 1172–1183. [CrossRef]
13. Zhang, Z.; Hackl, C.; Kennel, R. Computationally Efficient DMPC for Three-Level NPC Back-to-Back Converters in Wind Turbine Systems With PMSG. *IEEE Trans. Power Electron.* **2017**, *32*, 8018–8034. [CrossRef]
14. Bennani-Ben Abdelghani, A.; Ben Abdelghani, H.; Richardeau, F.; Blaquière, J.M.; Mosser, F.; Slama-Belkhodja, I. Versatile Three-Level FC-NPC Converter with High Fault-Tolerance Capabilities: Switch Fault Detection and Isolation and Safe Postfault Operation. *IEEE Trans. Ind. Electron.* **2017**, *64*, 6453–6464. [CrossRef]
15. Moranchel, M.; Huerta, F.; Sanz, I.; Bueno, E.; Rodríguez, F. A Comparison of Modulation Techniques for Modular Multilevel Converters. *Energies* **2016**, *9*, 1091. [CrossRef]
16. Rodríguez-Rodríguez, J.; Venegas-Rebollar, V.; Moreno-Goytia, E. Single DC-Sourced 9-level DC/AC Topology as Transformerless Power Interface for Renewable Sources. *Energies* **2015**, *8*, 1273–1290. [CrossRef]
17. Zhang, Z.; Hackl, C.; Kennel, R. FPGA Based Direct Model Predictive Current Control of PMSM Drives with 3L-NPC Power Converters. In Proceedings of the International Exhibition and Conference for Power Electronics, Intelligent Motion, Renewable Energy and Energy Management, Nuremberg, Germany, 10–12 May 2016; pp. 1–8.
18. Umesh, B.S.; Sivakumar, K. Multilevel Inverter Scheme for Performance Improvement of Pole-Phase-Modulated Multiphase Induction Motor Drive. *IEEE Trans. Ind. Electron.* **2016**, *63*, 2036–2043. [CrossRef]
19. Kumar, P.S.; Satyanarayana, M. Comparative analysis of modulation strategies applied to seven-level diode clamped multi-level inverter fed induction motor drive. In Proceedings of the 2015 Conference on Power, Control, Communication and Computational Technologies for Sustainable Growth (PCCCTSG), Kurnool, India, 11–12 December 2015; pp. 231–237.
20. Janik, D.; Kosan, T.; Blahnik, V.; Kamenicky, P.; Peroutka, Z.; Danek, M. Complete solution of 4-level flying capacitor converter for medium-voltage drives with active voltage balancing control with phase-disposition PWM. In Proceedings of the 2014 16th European Conference on Power Electronics and Applications, Lappeenranta, Finland, 26–28 August 2014; pp. 1–8.
21. Hirani, M.; Gupta, S.; Deshpande, D.M. Comparison of performance of induction motor fed by sine pulse width modulated inverter and multi level inverter using XILINX. In Proceedings of the 2014 IEEE International Conference on Advanced Communications, Control and Computing Technologies, Ramanathapuram, India, 8–10 May 2014; pp. 264–269.
22. Chun, T.W.; Tran, Q.V.; Lee, H.H.; Kim, H.G.; Nho, E.C. A simple capacitor votage balancing scheme for the cascaded five-level inverter fed AC machine drive. In Proceedings of the 6th IET International Conference on Power Electronics, Machines and Drives (PEMD 2012), Bristol, UK, 27–29 March 2012; pp. 1–5.
23. Ponce, R.R.; Talavera, D.V. FPGA Implementation of Decimal Division for Motor Control Drives. *IEEE Lat. Am. Trans.* **2016**, *14*, 3980–3985. [CrossRef]
24. Jabeen, S.; Srinivasan, S.K.; Shuja, S.; Dubasi, M.A.L. A Formal Verification Methodology for FPGA-Based Stepper Motor Control. *IEEE Embed. Syst. Lett.* **2015**, *7*, 85–88. [CrossRef]
25. Gdaim, S.; Mtibaa, A.; Mimouni, M.F. Design and Experimental Implementation of DTC of an Induction Machine Based on Fuzzy Logic Control on FPGA. *IEEE Trans. Fuzzy Syst.* **2015**, *23*, 644–655. [CrossRef]
26. Quang, N.K.; Hieu, N.T.; Ha, Q.P. FPGA-Based Sensorless PMSM Speed Control Using Reduced-Order Extended Kalman Filters. *IEEE Trans. Ind. Electron.* **2014**, *61*, 6574–6582. [CrossRef]
27. Busquets-Monge, S.; Maheshwari, R.; Munk-Nielsen, S. Overmodulation of n-Level Three-Leg DC-AC Diode-Clamped Converters with Comprehensive Capacitor Voltage Balance. *IEEE Trans. Ind. Electron.* **2013**, *60*, 1872–1883. [CrossRef]