Space Vector Modulator Implemented With a Low Cost FPGA

Mauricio A. Tonelli, María I. Valla* and Pedro E. Battaiotto
LEICI, Electrical Engineering Department, National University of La Plata (UNLP)
CC 91 (1900) - La Plata - ARGENTINA
mtonelli@leici1.ing.unlp.edu.ar - m.i.valla@ieee.org

***Abstract* - In this paper a new circuit implementation of a space vector PWM is presented. The modulator is developed using Field-Programmable Gate Array (FPGA) technology. The modulator takes the reference vector from the PC Bus, and constructs the modulated states to generate driving signals for the power switches. Space vector modulation strategies are reviewed, the main parameters are identified, and designed to fulfil the requirements of different applications. A detailed design of the control circuit to be implemented inside the FPGA is performed. The proposed modulator board is built and tested extensively. Finally, it is applied to the speed control of an induction motor, presenting a satisfactory performance.**

I. Introduction

During the last decades, the great development of power electronics devices as well as VLSI circuits, has given place to more elaborated control strategies for electric drives [1] and AC power conditioning systems [2]. All these applications need PWM DC-AC converters as an interface between the control circuits and the load. On one side, the PWM plays an important role in the minimization of harmonics and switching losses in the converters, so it is important to implement good modulation strategies. On the other side, during the development of the control algorithm of the drive or the power conditioning system, it is important to have a friendly environment to test the different alternatives. So it is possible to distinguished two well differentiated stages: 1) the control of the inverter through a PWM modulator and 2) control strategies of the application. These reasons motivated the use of a PC environment to elaborate the control strategies and to develop an independent PWM modulator. The latter reads the reference voltage vector from the PC Bus and then generates the driving signals to the power IGBTs of the inverter, as it is shown in the block diagram of Fig. 1. The use of Field-Programmable Gate Array (FPGA) technology facilitates the design and implementation of the modulator [3]. In this way we obtain two systems which are practically independent: a universal modulator, suitable for any use of power inverters, and a PC environment for the development of control strategies.

The availability of fast power electronic switches allows

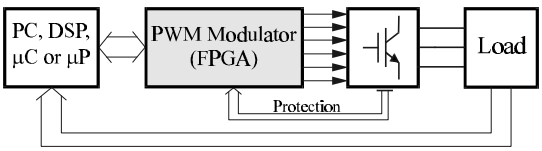


Fig. 1: General block diagram for the universal modulator

This work was support by National University of La Plata (UNLP), Science and Technology Secretarial (SECYT), and National Research Council (CONICET)

* M. I. Valla is also with the National Research Council (CONICET)

the implementation of diverse modulation techniques in the control of power inverters. The most widespread is the suboscillation method [4][5]. Today the space vector modulation provides a natural way to implement the PWM of three-phase inverters [6]-[9]. It performs full utilization of the available DC voltage without injection of zero sequence signals, as is the case of regular sampling technique [10]. In this paper, we report the development of an asynchronous space vector PWM, with constant switching frequency. The modulator takes the reference signals from a PC and delivers the control signals to the IGBTs of the power inverter. The switching times are calculated "off-line" and they are stored in two ROM memories. The control logic and some counters are implemented with a single IC from ALTERA to generate the driving signals to the inverter.

In detail, it is organized as follows. In section II the theory of the space vector and its application to power inverters is briefly reviewed. In section III the proposed modulator is described and the main parameters are determined. In section IV a detailed design of the FPGA circuit is performed, the hardware realization is described and evaluated. Section V illustrates the proposed modulator applied to the speed control of an induction motor. In section VI some conclusions are drawn.

II. Space Vector Modulation

The voltage space vector can be defined in terms of the three phase voltages, $u_{Rn}(t)$, $u_{Sn}(t)$, $u_{Tn}(t)$:

$$\vec{\mathbf{V}} = \frac{2}{3}\left(u_{Rn}(t) + u_{Sn}(t)\, e^{j\frac{2p}{3}} + u_{Tn}(t)\, e^{j\frac{4p}{3}} \right) \qquad (1)$$

The space vector provides a compact and powerful representation of a three-phase system. This representation allows visualizing a three-phase system in the complex plane that includes the three variables (voltage, current, flux, etc.).

A three-phase power inverter has two switches per phase. One and only one switch in each phase should be closed at every instant. When the space vector definition is applied to the inverter voltage, there result only eight different switching combinations. Each combination defines a space vector, determined by the states of the six semiconductor switches (S_1...S_6);

$$\vec{\mathbf{V}}_{\mathbf{k}} = \begin{cases} \frac{2}{3}\, V_{cc}\, e^{j\frac{\pi}{3}(k-1)} & k = 1,\ldots,6 \\ 0 & k = 0, 7 \end{cases} \qquad (2)$$

The inverter output voltage vector ($\vec{V}_k$) can assume only seven discrete locations in the complex plane, which are

named the switching state vectors. These are shown in Fig. 2, where the notation V_1 = (S_1 S_4 S_6) indicates which switches are closed for each location. The switching state vectors divide the complex plane in six zones, which are limited by two of them, and determine which vectors allow synthesizing other voltages inside the plane. There is a redundancy for the zero voltage vector, (k = 0, 7 in (2)), since it can be generated closing the three upper switches (V_0) or the three lower switches (V_7). We will take advantage of this redundancy to minimize the commutations in the inverter.

Taking the average value of the consecutive application of the switching state vectors, we can generate any voltage vector laying inside the hexagon whose corners are the six active switching state vectors (V_1...V_6). In particular, it is possible to synthesize the voltage vector, which corresponds to a sinusoidal balanced three-phase system $\vec{\mathbf{V}}_s^*$. This vector describes circular trajectories, where the amplitude of the sinusoid fixes the diameter, and the rotating speed correspond to its frequency.

$$\vec{\mathbf{V}}_s^* = V^* e^{j\omega t} \tag{3}$$

The space vector modulation strategy approximate the reference vector $\vec{\mathbf{V}}_s^*$ during a sampling time interval Δt = 1/(2f_s) by a sequence of three switching state vectors V_a, V_b and V_N. Where V_a and V_b are two of the six active vector V_1...V_6, V_N is a zero vector V_0 or V_7, and f_s is the switching frequency.

Therefore, the average voltage that results from the consecutive application of the three switching vectors, over the time interval Δt is equal to the reference vector. Then:

$$\vec{\mathbf{V}}_a\, t_a + \vec{\mathbf{V}}_b\, t_b + \vec{\mathbf{V}}_N\, t_0 = \vec{\mathbf{V}}_a\, t_a + \vec{\mathbf{V}}_b\, t_b = \vec{\mathbf{V}}_s^*\, \text{Ät} \tag{4}$$

with:

$$t_a + t_b + t_0 = \Delta t \tag{5}$$

where t_a, t_b and t_0 are the times interval during which the switching state vectors V_a, V_b and V_N are applied, respectively. The time intervals t_a and t_b are obtained from both components of the vector equation (4); while t_0 is directly calculated with (5),

$$\begin{aligned} t_a &= \sqrt{3}\,\frac{V^*}{V_{cc}}\Delta t\,\text{sen}(60^o - a) \\ t_b &= \sqrt{3}\,\frac{V^*}{V_{cc}}\Delta t\,\text{sen}(a) \\ t_0 &= \Delta t - t_a - t_b \end{aligned} \tag{6}$$

where α is the angle between V_a and $\vec{\mathbf{V}}_s^*$, and V^* is the magnitude of $\vec{\mathbf{V}}_s^*$. This is the generic solution for each sector in the plane; α always varies in the interval [0°, 60°), and a modulation index is defined as $m = \sqrt{3}V^* / V_{DC}$

A convenient sequence of the selected vectors should be chosen in order to minimize the commutations. There are several papers that describe the use of different sequences according to their goals, [8][9]. Hence, the election of the sequence inside each sector, during a sampling interval is chosen as: |<V_a><V_b><V_N>|<V_b><V_a><V_N>|. This

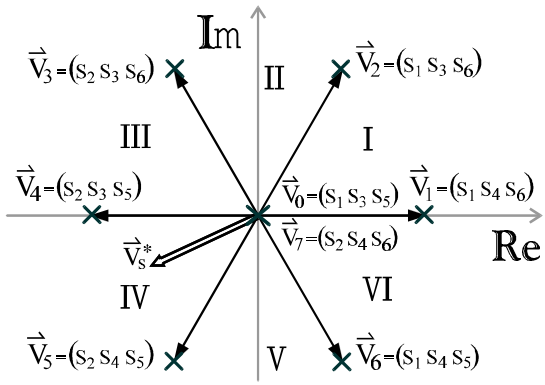


Fig. 2: Locations of the switching state vectors $\mathbf{V_k}$ and a reference vector $\vec{\mathbf{V}}_s^*$.

sequence allows minimizing the commutation of the power switches. Table 1 presents the proposed sequences corresponding to each sector. The transition of any vector to the next is carried out switching only one leg of the inverter.

III. Proposed Modulator

In Fig. 3 a block diagram of the proposed modulator is presented, it consists of three stages:

PC bus Interface: This stage is used for the adequate communication between the modulator and the PC, providing the modulator with the control and data signals. The PC writes 16 bits data on the modulator address in one single writing cycle. The 16 bits, which had information of the sector, magnitude and phase of the reference voltage vector, are latched by the Latch IN to guarantee that the ROMs are adequately addressed. The decoding of the 6 sectors requires 3 bits allow identifying 8 different situations, so there remain two free selections from the bits that decode the actual sector. These are used to control the turn-on and turn-off of the modulator.

Modulator: This is the core of the system. It is mainly implemented inside the FPGA except for two ROMs ICs shown in Fig. 3. The main task of the modulator is to control the time intervals t_a, t_b and t_0 during which different switching state vectors are applied. This task is fulfilled with three counters, a sequential machine and a state vector table (SVT) as it is presented in Fig. 4. The time data stored in ROM I_a (t_a) and ROM I_b (t_b) are loaded into counters Ct_a and Ct_b to implement the time intervals. A binary counter C generates the sub-cycle time Δt, counting a fixed number of clocks cycles. The time t_0 is

Table 1
Proposed Sequences Commutation

Sector	Sequence					
	Δt (odd)			Δt (even)		
I	V_1	V_2	V_0	V_2	V_1	V_7
II	V_3	V_2	V_0	V_2	V_3	V_7
III	V_3	V_4	V_0	V_4	V_3	V_7
IV	V_5	V_4	V_0	V_4	V_5	V_7
V	V_5	V_6	V_0	V_6	V_5	V_7
VI	V_1	V_6	V_0	V_6	V_1	V_7

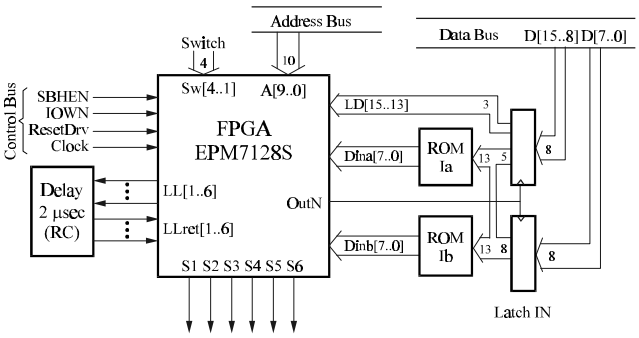


Fig. 3: Block diagram of the proposed modulator.

implemented as the time interval between the stop of Ct_a and Ct_b, and the end of the count of C. Latch II maintains the output data of ROM I_a, and ROM I_b so that they are ready to be loaded into the counters. The sequential machine receives the clock from the PC and the Star/Stop signals from the decoding block. Then it generates the enable signal of Latch II, controls the firing flow of Ct_a and Ct_b, and outputs 3 bits data to address the SVT at the output of the modulator. The SVT stores the states of the six switches of the inverter, for each switching state vector. This table is addressed with 6 bits; the 3 upper bits of the PC Bus, and three generated by the sequential machine.

Power Inverter interface: This stage arranges the signals coming from the SVT, to drive the power switches of the inverter in a proper way. It generates the necessary delays to avoid short circuits in any branch of the power inverter.

The main parameters of the modulator are:

- The switching frequency (f_s) that defines the sub-cycle $\Delta t = 1 / (2 f_s)$.
- The number of n bits necessary to implement times t_a, t_b and t_0 that determine the quantity of discrete values that can be taken to synthesize the reference vector.
- The clock frequency $f_{CLK} = 2^{(n+1)} f_s$.
- The number of bits, which are used to quantify the magnitude and phase of the reference voltage vector.
- The size of the 2 ROMs, which is determined by the quantization of the reference vector.

In the following subsections we will give design criteria for the selection of the different parameters.

A. Switching State Vectors Quantization Due to Discrete Time Implementation

For the sake of the simplicity, in the following description we assume that we have 5 bits (n = 5) to implement the time intervals, therefore the sub-cycle Δt is equal to 32 (2^n) clock cycles. Then, the smallest fraction of the switching space vector that can be distinguished is $\frac{2}{3} V_{DC} / 32$.

The reference vector is synthesized with the quantified values of the switching state vectors, so it can take only discrete values. For n = 5 bits we have 32x32/2 = 512 discrete values in each sector. Fig. 5 presents all the possible points that can implement a reference vector in sector I. A similar situation could be found in the other sectors.

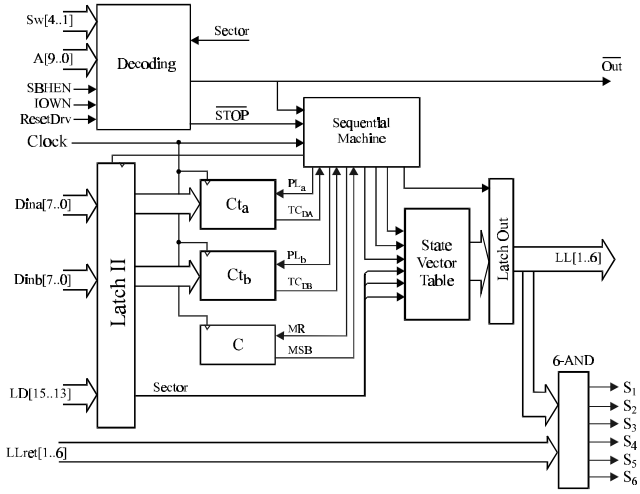


Fig. 4: Block diagram of the circuit inside the FPGA.

When n is increased, we have more discrete values to approach $\vec{V}_R^*$. Each bit that is added to the time quantization, increases four times the voltage resolution inside each sector. At the same time, the clock frequency should be increased, and the access-time of the memories may become significant. So, in the election of n, there exist a trade-off between high voltage resolution and timing of the circuit elements.

We chose n = 8 bits, for the time quantization. The reference vector can take 32768 discrete values per sector. We decided to use the clock, which is provided through the PC Bus. Therefore, the clock frequency is fixed at f_{CLK} = 8.375 MHz, n = 8, and the switching frequency is $f_s = f_{CLK} / 512 = 16.4\,\text{Khz}$, which is reasonable taking into account acoustic problems and the switching time of the power IGBTs.

B. Effects of the Switching Times of the Power Devices

The semiconductors devices used as power switches are not ideal, so they react with a certain delay to the driving signals corresponding to turn-on and turn-off. The delay time depends on the semiconductor, their current and voltage ranges, the device temperature, and the actual current to be switched. In order to avoid a short circuit in each branch of the power inverter, the control circuit should insert a delay time (T_d) in the turn-on signals of the power devices.

A time delay of 2 µsec is recommended in the data sheets of the power inverter that we will be used. So the power inverter cannot generate a pulse width shorter than this

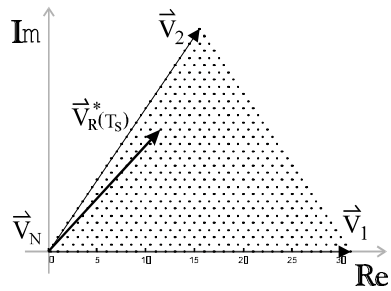


Fig. 5: Synthesis of $\vec{V}_R^*$ with n = 5 bits.

value. This means that when t_a, t_b or t_0 are lower than 2 μsec they cannot be implemented by the switches of the inverter. This value is slightly lower than $21T_{CLK}$, so there are several points near the border of each sector which cannot be synthesized. Then we approach the times t_a, t_b and t_0 to the nearest value between 0 and $21T_{CLK}$. It is worthy to point out that this limitation of the power devices affects the time resolution only in the borders of each sector, everywhere else the resolution is fixed by the number of bits of the counters.

C. Resolution for the Magnitude and the Angle

There are 16 bits available on the PC Data Bus, three of them are used to decode the sector where $\vec{V}_R^*$ is located. Then, the remaining 13 bits are distributed for the resolution of the magnitude and the angle of $\vec{V}_R^*$. A good choice is to take 7 bits for the magnitude (V^*) and 6 bits for the angle (α), whit α between 0º and 60º. The reference vector sample is quantified in magnitude and phase dividing the complex plane with radial lines and concentric circles. On the other hand, the time quantization divides the plane with lines parallel to the voltage vectors that determine the limit of each sector. These points represent the feasible values that can be synthesized by the power inverter. The sample of the reference voltage vector should be approximated to the nearest feasible value, when it has to be implemented by the power inverter.

D. Description of the Sample Implementation

Each sample of the reference vector $\vec{V}_R^*$ is implemented during two successive sub-cycles ($2\Delta t$), as shown in Fig. 6. It is convenient to take the samples of the reference vector at the end of the odd sub-cycles, and to begin their implementation in the next odd sub-cycle, as indicated in Fig. 6. With this criteria there will be a constant delay equal to Δt, and there is not a great demand in the ROMs access times.

IV. Design of the PWM Control Circuit in the FPGA

A. Decoding Circuit

The decoding circuit is the main interface with the EISA PC Bus. It takes signals of the control, address and data buses to be used in the modulator. The following lines are taken from the PC Control Bus: $\overline{IOW}$: it validates the available data in the Data Bus; $\overline{SBHE}$: is used together with A_0 to assure a 16 bits data transfer; ResetDvr: to stop the modulator when there is some problem with the PC; CLK: goes directly to the sequential machine.

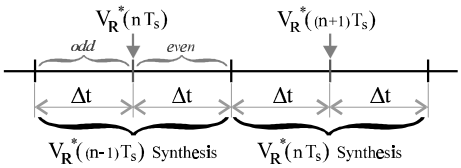


Fig. 6: Sample implementation graph

The modulator board is addressed by the A_0 to A_9 bits of the Address Bus. It occupies only one position in the Address Map of the PC. In order to obtain a flexible location of the board, the bits A_1 to A_4 are compared with a set of four switches. In this way, the modulator can be located in any position between 0300h and 031Eh, according to the complete system requirements. In addition to the address bits, we use the control signals to guarantee a 16 bits transfer, and to be sure that the data in the bus is stable during the data transfer. The decoding circuit generates the $\overline{OUT}$ signal, that enables the Latch IN, and the ROM I_a and I_b.

The D_0 to D_{12} bits of the data Bus, are used to address the two ROMs. D_0 to D_5 bits carry the information of the angle of $\vec{V}_R^*$, D_6 to D_{12} bits have the information of the magnitude of $\vec{V}_R^*$. Finally, D_{13} to D_{15} identifies the sector where $\vec{V}_R^*$ is located. These bits address the state vector table. The sequential machine controls their application to the SVT at the proper time through two latch circuits (Latch IN and Latch II).

The start of the modulator is performed directly when the board is addressed. The $\overline{OUT}$ signal starts the counter C, and it remains counting the sub-cycles Δt until it receives the stop command. The stop command may be generated either by the ResetDrv signal of the Control Bus, or by D_{14} and D_{15} when both are high. The first situation corresponds to some problem in the PC, while the second is generated on purpose when the running program decides to stop the sequential machine. The generated $\overline{STOP}$ signal stops the sequential machine, and resets all the Flip-Flops so that they are ready to start again when they receive the proper command.

B. Sequential Machine

The sequential machine is designed keeping in mind that each sample of the reference vector is implemented during two consecutive sub-cycles. The sequence to be implemented is $<V_a><V_b><V_0>$ during the odd sub-cycle, and $<V_b><V_a><V_0>$ during the even one. This sequence allows minimizing the switching losses in the inverter. During an odd sub-cycle, the most significant bit of the counter C (MSB) is low, while it is high during even sub-cycles. A high to low transition in MSB determines the beginning of an odd sub-cycle, then counter Ct_a should start counting t_a. When it finishes, it must give a signal to start counter Ct_b. After the end of the interval t_b, t_o is implemented waiting for a low to high transition in MSB. Then an even sub-cycle starts and the sequence should be reversed as shown in Fig. 7, where a schematic representation of the sequential machine is presented. PL is a signal that is low while the counter Ct is counting. When it goes high, it starts the successive counter.

Since the operation of the sequential machine is based on counter C, we should control the turn-on and turn-off of this counter. The first falling edge of the signal $\overline{OUT}$ starts counter C. Thus, when the modulator board is accessed for the first time, the $\overline{OUT}$ signal is generated starting the operation of the sequential machine. The $\overline{STOP}$ signal should be put to low level, to turn-off C. When the $\overline{STOP}$

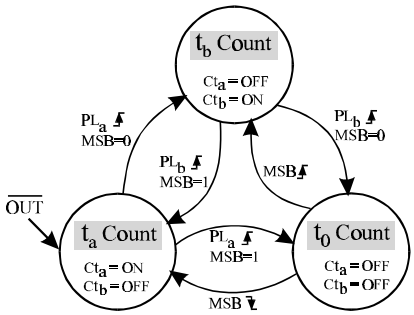


Fig. 7: State diagram of the sequential machine.

signal is low, the modulator generates six zeros to guarantee that all the power switches remain open.

C. Output Stage and Inverter's Interface

The SVT, at the modulator output, stores all switches states that allow implementing the optimum sequence presented in Table 1. In order to determine the proper state we have to know the sector where $\vec{V}_R^*$ is located, identify the even and odd sub-cycles, and recognize if t_a or t_b is in the present count. These requirements may be fulfilled using the lines D_{13} to D_{15}, MSB, PL_a and PL_b to address the SVT.

The modulator output signals S_1-S_6 are the driving signals of the power switches of the inverter, except for their power level. Therefore, it is very important for them to be stable at every instant, and to fulfil the requirements regarding the switching times of the power devices. The first condition is met introducing a latch (Latch OUT) that inhibits any change of the states LL_1-LL_6 when the address lines of the SVT are changing. The second condition is taken into account implementing a delay in the turn-on signals, to avoid the short circuit of any leg of the inverter. This delay is implemented with RC networks external to the FPGA, and AND gates which are inside the FPGA. The implemented delay is equal to 2 µsec, in accordance with the requirements of the IGBT power inverter.

D. Hardware Implementation

Cost considerations led to selecting the FPGA EPM7128S from ALTERA, for the implementation of the proposed modulator. We used the CAD tools MAX+plus II provided by ALTERA. These tools consist of a schematic input, an AHDL input (VHLD based program), a functional simulator, and a timing simulator. Fig. 8 illustrates the simulation results of the FPGA design, obtained from the timing simulator software. The first line ILD[15...13] identifies the sector, the second line corresponds to t_a and the third to t_b. The next two lines are the control lines of the counters Ct_b, Ct_a. The following one is MSB which has the information of Δt. The last two lines represent the output of the SVT in which each number corresponds to a vector in hexadecimal code, and the LatchOut enable that inhibits the output of the SVT when its addressing is changing. Our design of the control circuit inside the FPGA requires 106 logic cells of the 128 available in the IC. To communicate with the surrounding circuit it uses 56 I/O pins of the 64 available in the IC. All this result in a usage percentage of 80%, which is near the upper limit recommended by ALTERA.

V. Speed Variation of an Induction Motor

The proposed modulator is now tested with a power inverter driving a 0.5 HP induction motor. There is no closed loop speed control. In the PC environment, we generate different reference vector that rotates at different speed maintaining the V/f ratio. In the following figures, we present the experimental results obtained with this setup.

Fig. 9 shows the motor voltages and currents for a fundamental frequency of 60 Hz and a modulator index m = 0.95. In Fig. 9.a the voltages of two legs of the inverter are shown. The switching period has been measured, being equal to 61.28 µseg which correspond to a switching frequency of 16.3 Khz. Fig. 9.b shows the currents in each phase of the motor. It is evident that they are good sinusoids with a phase shift of 120° between them.

Fig. 10 correspond to a test with f = 10 Hz and m = 0.2. It is evident that the quality of the waveforms is preserved. We convert the three-phase current into their quadrature components with the help of the IC AD2S100AP. Then, the locus of the stator currents is obtained and plotted in Fig. 11.

VI. Conclusions

In this paper the design and implementation of a universal pulse width modulator for power inverters was presented. The modulator executes a Space Vector PWM with a single low cost FPGA IC. This means a simple hardware realization free of adjustments. The developed board, takes the reference space vector from the PC Bus,

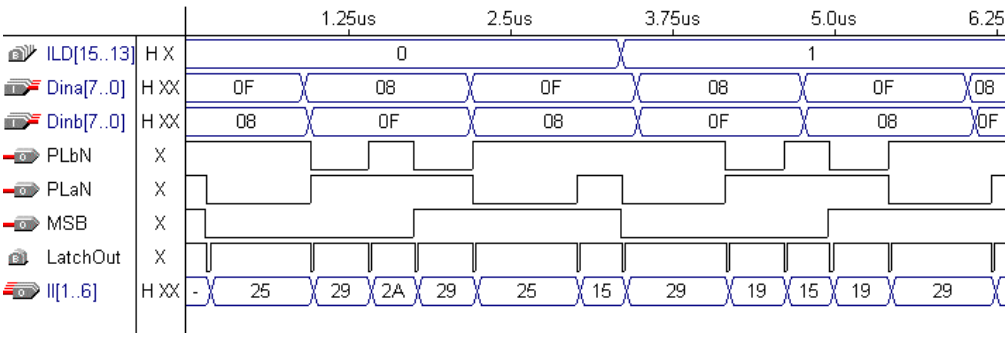
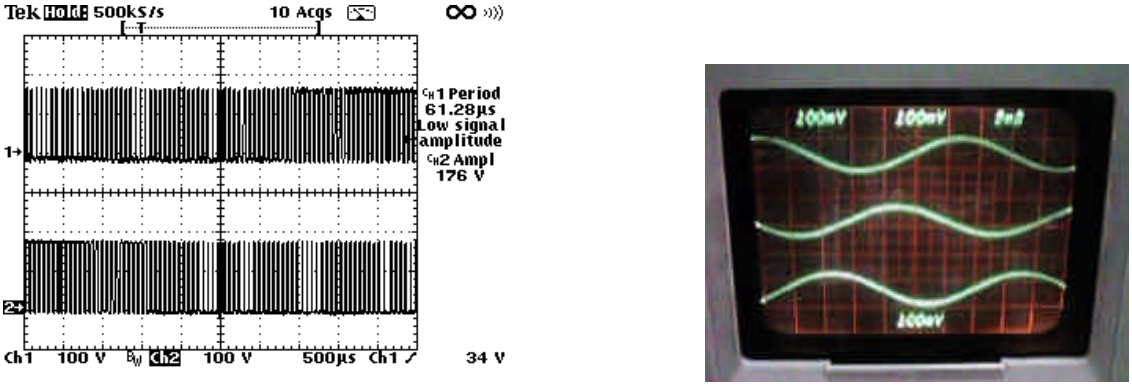
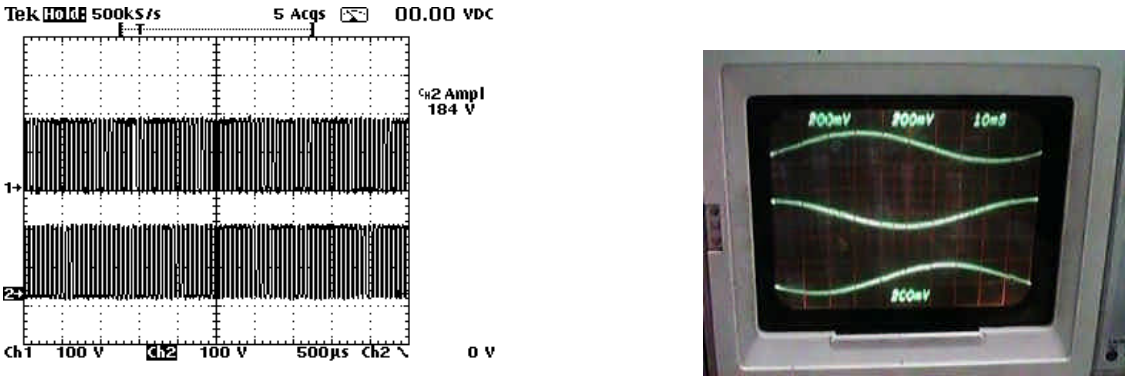


Fig. 8: Timing diagram obtained with the timing simulator.



a) b)

Fig. 9: Reference vector: f_0 =60Hz and m = 0,95. a) Voltage of two inverter legs. b) Stator currents.



a) b)

Fig. 10: Reference vector: f_0 =10Hz and m = 0,2. a) Voltage of two inverter legs. b) Stator currents.

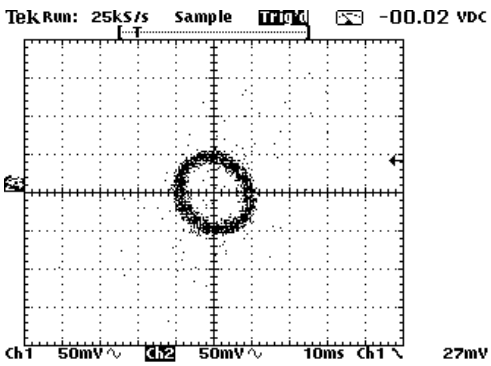


Fig. 11: Locus of the stator current.

and generates the driving signals for the switches of the power inverter. In this way, the proposed modulator is easily applied in the control of AC drives and AC power conditioning systems.

Although the modulator was designed to interface with a PC environment, the present design can be readily changed to interface with any industrial Bus or micro-controller system. This is effortlessly achieved reprogramming the decoding stage inside the FPGA. Once it is adapted with the adequate interface to industrial applications, it can be upgraded to a Mask Programmed Gate Array (MPGA) which allows reducing production costs, when relatively large quantities are produced.

VII. References

[1] B. K. Bose, *Power Electronics and Variable Frequency Drives*, IEEE PRESS, New York 1997.

[2] B. K. Bose, *Modern Power Electronics: Evolution, Technology, and Applications*, IEEE PRESS, New York 1992.

[3] Y.Y. Tzou and H-J Hsu, "FPGA Realization of Space-Vector PWM Control IC for Three-Phase PWM Inverters", *IEEE Trans. on Power Electronics*, Vol. 12, N° 6, Nov. 1997, pp. 953-963.

[4] Bowes S. R., "New Sinusoidal pulsewidth-modulated inverter", *Proc. IEE*, Vol 122, Nº 11, pp. 1279-1285, Nov. 1975.

[5] Bowes Sidney R., "Advanced Regular-Sampled PWM Control Techniques for Drives and Static Power Converters", *IEEE Trans. on Industrial Electronics*, Vol. 42, Nº 4, August 1995, pp. 367-373

[6] Holtz J., "Pulse Width Modulation for Electronic Power Conversion", *IEEE Proceedings*, Vol. 82, N° 8, Aug. 1994, pp. 1194-1214.

[7] Del Valle J., Rodriguez P. y Contreras T., "Técnicas de modulación vectorial en inversores trifásicos", *Brazilian Power Electronic Conference (COBEP'91)*, Dec. 1991, pp 231-238.

[8] Holtz J. and Bube E., "Field-Oriented Asynchronous Pulse-Width Modulation for High-Performance ac Machine Drives Operating at Low Switching Frequency", *IEEE Trans. on Industry Applications*, Vol. 27, Nº 3, May-June 1991, pp. 574-580.

[9] Kazmierkowski M. and Sulkowski W., "A Novel Vector Control Scheme for Transistor PWM Inverter-Fed Induction Motor Drive", *IEEE Trans. on Industrial Electronics*, Vol. 38, N° 1, Feb. 1991, pp. 41-47.

[10] Bowes S. R. and Yen-Shin Lai, "The Relationship Between Space-Vector Modulation and Regular-Sampled PWM", *IEEE Trans. on Industrial Electronics*, Vol. 44, Nº 5, October 1997.