Hexagon based Algorithm for Space Vector Modulation on Multilevel Voltage Source Inverters

Santiago de Pablo*, *Member, IEEE*, Alexis B. Rey-Boué**, Luis C. Herrero*, and Fernando Martínez*
(*) Department of Electronics Technology, University of Valladolid, Valladolid, Spain.
(**) Dept. of Electronics, Computer Technology and Projects, Technical University of Cartagena, Cartagena, Murcia, Spain.
E-mail: sanpab@eis.uva.es, alexis.rey@upct.es, lcherrer@tele.uva.es, fer_mart@tele.uva.es

Abstract-A **simple algorithm has been developed to control small and large multilevel inverters using Space Vector Modulation. It looks for a suitable hexagon near the reference voltage rather than the usual three nearest vectors. Then, switching vectors and their duty cycles are computed using straightforward equations, independent of the number of levels and easy to use in digital microprocessors. Several simulations have validated this method, finding that these duty cycles are exactly the same than others, computed using more complex methods.**

I. INTRODUCTION

Multilevel voltage source inverters (VSI) and its technology have experienced a fast growing attention in the last decade [1], [2]. Since they were introduced by Nabae *et al.* in 1981 [3] they have increasingly been used due to their better performance compared to two-level VSI in medium voltage [2], [4] and even low voltage applications [5]. Advantages of this approach include improved power quality, higher voltage capability, lower switching losses, smaller and cheaper filtering and enhanced electromagnetic compatibility [6]. Most used topologies are diode-clamped (neutral-point clamped or multi-point clamped), capacitor-clamped (flying capacitors) and cascaded multi-cell with separate dc sources, all of them sharing mostly the same control methods [1], [2].

The most used control strategies for multilevel VSI are multi-carrier based sinusoidal or nonsinusoidal Pulse Width Modulation (PWM), Space Vector Modulation (SVM), and several offline optimized pulse patterns, for example selective harmonic elimination (SHE) [2]. Actually, time based PWM and SVM techniques may lead to the same results when distorted modulation is used on PWM [7], [8], but SVM may offer more flexibility to implement different optimizations, for example to decide how to keep balanced all dc sources [9] and how to reduce or remove common-mode voltage to avoid premature motor bearing failures [10].

Initially, SVM technique was extended from its two-level version, finding three vectors near a rotating reference vector and then solving three linear equations to obtain their on-times [11]; this method leans on the fact that duty cycles depend only on local information, actually the coordinates of the three vectors modulated during the sampling period, but the weak point is that its computational complexity increases dramatically with the number of levels. Later on, a second approach has solved the problem for the general *n*-level case [12]: firstly it normalizes the reference voltages and then it uses a nonorthogonal coordinate transformation in such a way that the nearest three vectors are found using the integer part of the reference voltages on that basis and duty cycles are, in fact, their fractional parts; although this algorithm gives valid results for any situation, its nonorthogonal transformation would be considered unnatural, thus it would be difficult to work with such unusual variables. A third method considers an *n*-level space vector diagram composed by several smaller diagrams, so it shifts the origin to a virtual two-level inverter near the reference, and it computes duty cycles in situations similar to well known two-level inverters [13], [14], [15]. Eventually, look-up tables are used to store information difficult to seize using regular expressions valid for all situations [15], [16]. These ideas have been applied also on four-leg multilevel VSI [9], [17].

This paper presents an important simplification over all aforementioned references and provides a general solution for multilevel inverters of any size. It is based on natural coordinates, namely *'ab'*-*'bc*-*'ca'* [17], [18], rather than usual α-β components [15] or other transformations [12], that may result harder to implement or more difficult to understand. The salient features of the proposed scheme are as follows.

1) Duty cycles of any *n*-level inverter are computed using only local information, actually a two-level hexagon, rather than the usual three nearest vectors.

2) A joint expression is used to compute duty cycles for all six sectors of the hexagon, rather than six sets of expressions.

3) No trigonometric functions are used, so computations are very efficient in digital microprocessors.

4) No look-up tables are used to compute duty cycles or switching states: regular expressions have been found to solve these problems efficiently for multilevel VSI of any size.

Throughout this paper, new expressions are proposed to compute duty cycles on two-level inverters and then the algorithm is generalized for the general *n*-level case, finding equations that give full flexibility for application specific optimizations. Several simulations have validated this algorithm, finding that these duty cycles are exactly the same than others, computed using more complex methods.

II. TWO-LEVEL SPACE VECTOR MODULATION

The structure of a typical three-phase two-level voltage source inverter (VSI) is shown in Fig. 1(a). The relationship between switching variables (S_a, S_b, S_c) and phase-to-neutral

978-1-4244-6392-3/10/$26.00 ©2010 IEEE

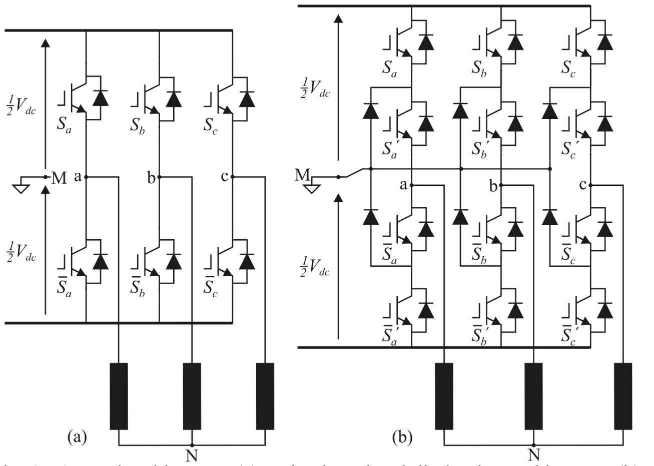

Fig. 1. A two-level inverter (a) and a three-level diode-clamped inverter (b).

voltages (V_{aN}, V_{bN}, V_{cN}) is given by (1), where V_{dc} is the bus voltage and S_k is '1' or '0' when the upper or lower transistor of phase k is on, respectively [11]. Applying the Clarke transformation (2) to these voltages, that satisfy $V_{aN} + V_{bN} + V_{cN} = 0$, it leads to a well known space vector $\boldsymbol{V}$ with the same instantaneous information in a stationary reference frame α-β.

$$\begin{pmatrix} V_{aN} \\ V_{bN} \\ V_{cN} \end{pmatrix} = \tfrac{1}{3}V_{dc}\begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix}\cdot\begin{pmatrix} S_a \\ S_b \\ S_c \end{pmatrix} \tag{1}$$

$$\boldsymbol{V} = \begin{pmatrix} V_\alpha \\ V_\beta \end{pmatrix} = \begin{bmatrix} \frac{2}{3} & \frac{-1}{3} & \frac{-1}{3} \\ 0 & \frac{\sqrt{3}}{3} & \frac{-\sqrt{3}}{3} \end{bmatrix}\cdot\begin{pmatrix} V_{aN} \\ V_{bN} \\ V_{cN} \end{pmatrix} \tag{2}$$

There are eight different vectors for the two-level inverter output, named $\boldsymbol{V_0}$ to $\boldsymbol{V_7}$ as defined in Fig. 2, according to the eight possible switching states. To obtain the required output space vector $\boldsymbol{v_o}$*, that usually is a rotating vector with constant module and pulsation, conduction times of the inverter switches are modulated according to the angle and magnitude of that reference. The null vectors $\boldsymbol{V_{Z1}} = \boldsymbol{V_0}$ and $\boldsymbol{V_{Z2}} = \boldsymbol{V_7}$, and two space vectors adjacent to $\boldsymbol{v_o}$*, $\boldsymbol{V_X}$ and $\boldsymbol{V_Y}$, are chosen and modulated during a short switching time T_S using the volt-second balance method:

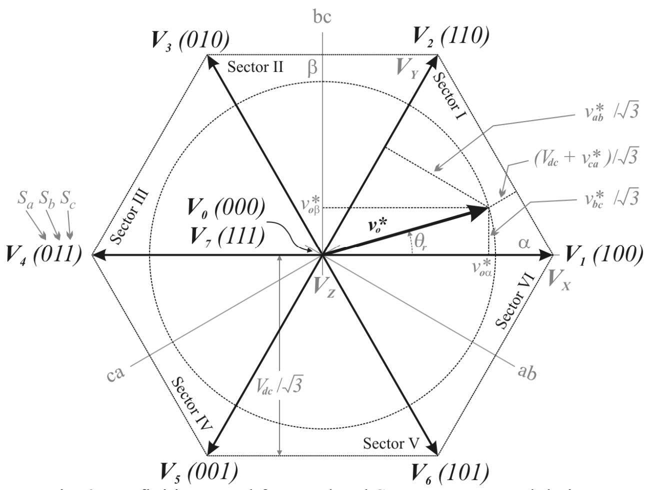

Fig. 2. Definitions used for two-level Space Vector Modulation.

$$\boldsymbol{v_o^*} \cong \frac{1}{T_S}\int_t^{t+T_S}\boldsymbol{v_o}dt = d_{Z1}\cdot\boldsymbol{V_{Z1}} + d_X\cdot\boldsymbol{V_X} + d_Y\cdot\boldsymbol{V_Y} + d_{Z2}\cdot\boldsymbol{V_{Z2}} \tag{3}$$

As the values of d_{Z1} and d_{Z2} have no effect in (3) because of the null magnitude of $\boldsymbol{V_0}$ and $\boldsymbol{V_7}$, solving this equation using rectangular coordinates yields to (4):

$$\begin{bmatrix} V_{X\alpha} & V_{Y\alpha} \\ V_{X\beta} & V_{Y\beta} \end{bmatrix}\cdot\begin{pmatrix} d_X \\ d_Y \end{pmatrix} = \begin{pmatrix} v_{o\alpha}^* \\ v_{o\beta}^* \end{pmatrix} \tag{4}$$

where the matrix arranges the known α-β components of the adjacent inverter vectors $\boldsymbol{V_X}$ and $\boldsymbol{V_Y}$. Equation (4) can be applied, without loss of generality, to a reference phase voltage in the range 0 to $\pi/3$, thus $\boldsymbol{V_1}$ and $\boldsymbol{V_2}$ will be used jointly with $\boldsymbol{V_0}$ and $\boldsymbol{V_7}$ to generate $\boldsymbol{v_o}$*, as shown in Fig. 2. Thus, the amount of time for each voltage vector is computed by solving the inverse problem (5), leading to known equations (6) and (7).

$$\begin{pmatrix} d_X \\ d_Y \end{pmatrix} = \begin{pmatrix} d_1 \\ d_2 \end{pmatrix} = \begin{bmatrix} \frac{2}{3}V_{dc} & \frac{1}{3}V_{dc} \\ 0 & \frac{\sqrt{3}}{3}V_{dc} \end{bmatrix}^{-1}\begin{pmatrix} v_{o\alpha}^* \\ v_{o\beta}^* \end{pmatrix} \tag{5}$$

$$d_X = d_1 = \frac{\frac{3}{2}\cdot v_{o\alpha}^* - \frac{\sqrt{3}}{2}\cdot v_{o\beta}^*}{V_{dc}} \tag{6}$$

$$d_Y = d_2 = \frac{\sqrt{3}\cdot v_{o\beta}^*}{V_{dc}} \tag{7}$$

Finally, the duty cycle of null vectors is computed:

$$d_Z = d_{Z1} + d_{Z2} = d_0 + d_7 = 1 - d_X - d_Y \tag{8}$$

Applying this method to all six sectors of Fig. 2, duty cycles can be efficiently computed on all situations using different sets of equations, each one valid on one triangle. These equations are clearly much faster than (9) and (10), proposed originally by [19] and later on applied by [6], [9], [16] and [20], based on polar coordinates that require trigonometric functions:

$$d_X = \frac{\sqrt{3}\cdot V_o}{V_{dc}}\cdot\sin(\tfrac{\pi}{3} - \theta_r) \tag{9}$$

$$d_Y = \frac{\sqrt{3}\cdot V_o}{V_{dc}}\cdot\sin(\theta_r) \tag{10}$$

where V_o is the magnitude of the phase-to-neutral reference voltage $\boldsymbol{v_o}$* ($0 \le V_o \le 0.577\cdot V_{dc}$ in the linear range) and θ_r is the relative angle between $\boldsymbol{v_o}$* and $\boldsymbol{V_X}$ ($0 \le \theta_r \le \pi/3$). The advantage of this last method is that only one set of equations is required for all six sectors.

Finally, the switching sequence during T_S is usually $\boldsymbol{V_{Z1}}$ (25% of d_Z), $\boldsymbol{V_X}$ (50% of d_X), $\boldsymbol{V_Y}$ (50% of d_Y), $\boldsymbol{V_{Z2}}$ (50% of d_Z), $\boldsymbol{V_Y}$ (50% of d_Y), $\boldsymbol{V_X}$ (50% of d_X) and $\boldsymbol{V_{Z1}}$ (25% of d_Z). Moreover, the distribution of d_Z may be changed to reduce the common mode voltage [21].

III. New Algorithm For Two-Level Inverters

Duty cycles computed using (6) to (8), and their equivalent equations for all six sectors, give full control of two-level inverters. In addition, that known method would be the fastest one used to compute SVM duty cycles, but those equations have no symmetry, thus it is difficult to apply them on multilevel inverters. Actually, initial methods proposed to compute SVM duty cycles on *n*-level inverters were based on that scheme, thus they needed to solve $6{\cdot}(n-1)^2$ inverse problems to get suitable equations for all triangles [11]. Obviously, this method increases dramatically the length of the algorithm with the number of levels *n*.

In order to simplify (6) to (8), the Clarke transformation will be used on them, leading to an astonishing simple result [22], [23]:

$$d_X = d_1 = \frac{\frac{3}{2}{\cdot}v_{o\alpha}^* - \frac{\sqrt{3}}{2}{\cdot}v_{o\beta}^*}{V_{dc}} = \frac{v_{an}^* - v_{bn}^*}{V_{dc}} = \frac{v_{ab}^*}{V_{dc}} \tag{11}$$

$$d_Y = d_2 = \frac{\sqrt{3}{\cdot}v_{o\beta}^*}{V_{dc}} = \frac{v_{bn}^* - v_{cn}^*}{V_{dc}} = \frac{v_{bc}^*}{V_{dc}} \tag{12}$$

$$d_Z = d_0 + d_7 = 1 - d_X - d_Y = 1 + \frac{v_{cn}^* - v_{an}^*}{V_{dc}} = 1 + \frac{v_{ca}^*}{V_{dc}} \tag{13}$$

These equations are clearly simpler than (6) to (8), and they can also be applied to all six sectors, yielding to the expressions shown at Table I. Actually, they also match the three heights of the point $\boldsymbol{v_o}^*$ on the triangle $\boldsymbol{V_X}$ - $\boldsymbol{V_Y}$ - $\boldsymbol{V_Z}$, as shown in Fig. 2, leading to a very simple graphical method to obtain them (see [23] for more details).

This direct relationship between line-to-line voltages and the space vectors was previously found by Zhou and Wang [7], as reported at [22] and [23], but later on they propose to use (9) and (10) for duty cycles computation. Anyway, proposed equations of Table I are not necessarily faster on two-level inverters than those obtained when using (6) to (8). Actually, in most situations the voltage reference $\boldsymbol{v_o}^*$ is provided using the stationary reference frame α-β, thus additional computations are required in this framework to preprocess the reference.

TABLE I
Proposed Expressions to Control Two-Level Inverters

Sector	Duty cycles and vector sequences	Sector	Duty cycles and vector sequences
I	$d_X = v_{ab}^*/V_{dc}$ $\boldsymbol{V_X}$ = 100	IV	$d_X = -v_{bc}^*/V_{dc}$ $\boldsymbol{V_X}$ = 001
0–π/3	$d_Y = v_{bc}^*/V_{dc}$ $\boldsymbol{V_Y}$ = 110	π–4π/3	$d_Y = -v_{ab}^*/V_{dc}$ $\boldsymbol{V_Y}$ = 011
	$d_Z = 1 + v_{ca}^*/V_{dc}$ $\boldsymbol{V_Z}$ = 000 & 111		$d_Z = 1 - v_{ca}^*/V_{dc}$ $\boldsymbol{V_Z}$ = 000 & 111
II	$d_X = -v_{ab}^*/V_{dc}$ $\boldsymbol{V_X}$ = 010	V	$d_X = v_{ca}^*/V_{dc}$ $\boldsymbol{V_X}$ = 001
π/3–2π/3	$d_Y = -v_{ca}^*/V_{dc}$ $\boldsymbol{V_Y}$ = 110	4π/3–5π/3	$d_Y = v_{ab}^*/V_{dc}$ $\boldsymbol{V_Y}$ = 101
	$d_Z = 1 - v_{bc}^*/V_{dc}$ $\boldsymbol{V_Z}$ = 000 & 111		$d_Z = 1 + v_{bc}^*/V_{dc}$ $\boldsymbol{V_Z}$ = 000 & 111
III	$d_X = v_{bc}^*/V_{dc}$ $\boldsymbol{V_X}$ = 010	VI	$d_X = -v_{ca}^*/V_{dc}$ $\boldsymbol{V_X}$ = 100
2π/3–π	$d_Y = v_{ca}^*/V_{dc}$ $\boldsymbol{V_Y}$ = 011	5π/3–2π	$d_Y = -v_{bc}^*/V_{dc}$ $\boldsymbol{V_Y}$ = 101
	$d_Z = 1 + v_{ab}^*/V_{dc}$ $\boldsymbol{V_Z}$ = 000 & 111		$d_Z = 1 - v_{ab}^*/V_{dc}$ $\boldsymbol{V_Z}$ = 000 & 111

Regardless of which method is faster, the simplicity and high level of symmetry of these new equations allows achieving further developments. In fact, a joint expression can be found for all six sectors, as shown following. First of all, equation (14) normalizes the line-to-line reference voltages to simplify further computations and (15) gets three polarities used to identify the sector where the reference vector is located. Then, duty cycles are evaluated by means of (16), where polarities of (15) are arranged in a matrix in such a way that they select the proper reference voltages, the same ones included at Table I. Finally, required duty cycles are sorted with (17) in such a way that commutations are minimized.

$$\begin{aligned} u_{ab}^* &= v_{ab}^* \,/\, V_{dc} \\ u_{bc}^* &= v_{bc}^* \,/\, V_{dc} \\ u_{ca}^* &= v_{ca}^* \,/\, V_{dc} \end{aligned} \tag{14}$$

$$\begin{aligned} &if\ (u_{ab}^* \geq 0.0)\quad p_{ab} = 1 \quad else \quad p_{ab} = -1 \\ &if\ (u_{bc}^* \geq 0.0)\quad p_{bc} = 1 \quad else \quad p_{bc} = -1 \\ &if\ (u_{ca}^* \geq 0.0)\quad p_{ca} = 1 \quad else \quad p_{ca} = -1 \end{aligned} \tag{15}$$

$$\begin{pmatrix} d_Z \\ d_A \\ d_B \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} - \frac{1}{2} \begin{bmatrix} p_{ab} & p_{bc} & p_{ca} \\ p_{ca} & p_{ab} & p_{bc} \\ p_{bc} & p_{ca} & p_{ab} \end{bmatrix} \cdot \begin{pmatrix} u_{ab}^* \\ u_{bc}^* \\ u_{ca}^* \end{pmatrix} \tag{16}$$

$$\begin{aligned} &if\,(p_{ab} + p_{bc} + p_{ca} > 0) \\ &\qquad d_X = d_A \\ &\qquad d_Y = d_B \\ &else \\ &\qquad d_X = d_B \\ &\qquad d_Y = d_A \end{aligned} \tag{17}$$

These equations are clearly different from all previous methods, although computed duty cycles match exactly the same results, because the solution to this problem is unique.

IV. Proposed Algorithm for N-Level Inverters

In this section, the proposed method for two-level inverters is extended to the general case of *n*-level inverters, leading to straightforward and easy to use equations. Given a reference voltage vector specified through its phase-to-neutral voltages $\boldsymbol{v_o}^* = (v_{aN}^*, v_{bN}^*, v_{cN}^*)$ for an *n*-level VSI, see Fig. 1(b), the purpose of this algorithm is to decide which vectors $\boldsymbol{V_{Z1}}$, $\boldsymbol{V_X}$, $\boldsymbol{V_Y}$ and $\boldsymbol{V_{Z2}}$ will be modulated during a small switching period T_S, and compute their duty cycles d_X, d_Y and d_Z. Then, they will be applied as described at the end of Section II.

The first step of this algorithm, partially common with other techniques [9], [12], [15], [17] is a normalization process (18), where line-to-line reference voltages are computed and then reduced by V_c, which is the full dc bus voltage (V_{dc}) divided by ($n - 1$). As long as the sum of the three phase voltages is always zero, using their instantaneous values is equivalent to use their α-β space components.

$$\begin{aligned} u_{ab}^* &= (v_{aN}^* - v_{bN}^*) / V_c = v_{ab}^* / V_c \\ u_{bc}^* &= (v_{bN}^* - v_{cN}^*) / V_c = v_{bc}^* / V_c \\ u_{ca}^* &= (v_{cN}^* - v_{aN}^*) / V_c = v_{ca}^* / V_c \end{aligned} \tag{18}$$

The second step usually consists of looking for the nearest three vectors surrounding the reference vector [9], [12], selected to minimize the harmonic components of the output line voltage [8], [12]. This method will find them later, using (23) to (26), but now it will choose a suitable hexagon that encloses the reference vector (see Fig. 3). At any given time, up to three different candidates can be the center of such hexagon, thus this step is clearly open for different policies and/or optimizations.

Following Holmes *et al.* [8], [14], the nearest even redundant space vector will be selected using (19) to (21) to reduce overall commutations (vectors with even redundancy are highlighted in Fig. 3). The location of the hexagon center (c_{ab}, c_{bc}, c_{ca}) is firstly estimated by (19), where normalized voltages are rounded to the nearest integer[1]:

$$\begin{aligned} c_{ab} &= \lfloor u_{ab}^* + 0.5 \rfloor \\ c_{bc} &= \lfloor u_{bc}^* + 0.5 \rfloor \\ c_{ca} &= \lfloor u_{ca}^* + 0.5 \rfloor \end{aligned} \tag{19}$$

Afterwards, phase i with the maximum value of $|u_i^*|$ is selected (i = {*'ab'*, *'bc'*, *'ca'*}), and indexes j and k are assigned to the other two phases. In addition, two variables named *even* and *odd* are set to *1* or *0* depending on the parity of n. Then, coordinate c_i is changed using (20) in order to set the hexagon center on an even redundant space vector, and c_j and c_k are fixed applying (21) to satisfy $c_i + c_j + c_k = 0$, assuring at the same time that the nearest even redundant space vector is selected, as shown at Fig. 3.

$$c_i = \lfloor (u_i^* + even) / 2.0 \rfloor \cdot 2 + odd \tag{20}$$

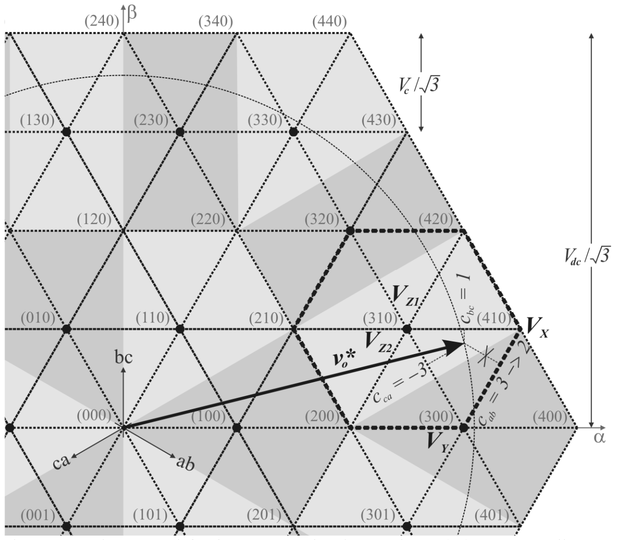

Fig. 3. Regions used for hexagon selection, using *ab-bc-ca* coordinates.

[1] Expression $\lfloor x \rfloor$ gets the lower bounded integer of x.

$$\begin{aligned} & if\ (|u_j - c_j| > |u_k - c_k|) \\ & \quad c_j = - c_i - c_k \\ & else \\ & \quad c_k = - c_i - c_j \end{aligned} \tag{21}$$

In the example of Fig. 3, the hexagon center is estimated using (19) at invalid coordinates *(3, 1, –3)*, and then it is modified by (20) and (21) to *(2, 1, –3)*, changing c_{ab} because the height $|u_{ab}^* - c_{ab}|$ was greater than $|u_{bc}^* - c_{bc}|$. This example demonstrates how simple and powerful can be an algorithm when using *ab-bc-ca* natural coordinates rather than usual α-β space components.

The third step computes the three duty cycles using the information of the hexagon centre, but with no detailed knowledge of the nearest three vectors, which are found later. Actually, this method is equivalent to a local two-level SVM, as proposed by [13], [14] and [15], but using a different approach because of the hexagon. First of all it computes the relative coordinates of the reference voltage at the hexagon using (22).

$$\begin{aligned} w_{ab}^* &= u_{ab}^* - c_{ab} \\ w_{bc}^* &= u_{bc}^* - c_{bc} \\ w_{ca}^* &= u_{ca}^* - c_{ca} \end{aligned} \tag{22}$$

The following steps of computing polarities and sorting duty cycles (d_X, d_Y, d_Z) is immediate: it can be accomplished by means of previously described (15) to (17), just replacing absolute reference voltages u_i^* by their relative counterparts w_i^* (i = {*'ab'*, *'bc'*, *'ca'*}). Herein it is shown the advantage of using just one expression for all six sectors. These equations are clearly different from all previous methods, and show how natural coordinates do simplify all the process.

Afterwards, a forth step of this algorithm must choose a suitable vector pair $\boldsymbol{V_{Z1}}$ = (S_{Z1a}, S_{Z1b}, S_{Z1c}) and $\boldsymbol{V_{Z2}}$ = (S_{Z2a}, S_{Z2b}, S_{Z2c}), both located at the hexagon center. The use of small look-up tables has been proposed and successfully applied at [15], but using the notation of [8], [9], [12] (see Fig. 3) the equations for the general n-level case are very simple:

$$\begin{aligned} S_{Z1a} &= max(0, c_{ab}, -c_{ca}) + \rho \\ S_{Z1b} &= max(0, c_{bc}, -c_{ab}) + \rho \\ S_{Z1c} &= max(0, c_{ca}, -c_{bc}) + \rho \end{aligned} \tag{23}$$

$$\begin{aligned} S_{Z2a} &= S_{Z1a} + 1 \\ S_{Z2b} &= S_{Z1b} + 1 \\ S_{Z2c} &= S_{Z1c} + 1 \end{aligned} \tag{24}$$

where ρ is a free integer due to redundant states. This parameter can be used to select a suitable set of vectors in order to balance dc voltages [9] and/or reduce or remove common-mode voltage [10], provided that $0 \le \{S_{Z1a}, S_{Z1b}, S_{Z1c}\} < \{S_{Z2a}, S_{Z2b}, S_{Z2c}\} \le n - 1$ is always satisfied. Anyway, these issues depend on the inverter topology, so they are beyond the scope of this paper.

The last step of this method decides the location of two neighbors $\boldsymbol{V_X}$ = (S_{Xa}, S_{Xb}, S_{Xc}) and $\boldsymbol{V_Y}$ = (S_{Ya}, S_{Yb}, S_{Yc}) used jointly with $\boldsymbol{V_{Z1}}$ and $\boldsymbol{V_{Z2}}$ during the switching period T_S. Equations (25) and (26) choose them in such a way that one and only one switch changes at any time (be aware that this policy may be changed when removing common-mode voltage, see [10]) on the sequence $\boldsymbol{V_{Z1}}$-$\boldsymbol{V_X}$-$\boldsymbol{V_Y}$-$\boldsymbol{V_{Z2}}$-$\boldsymbol{V_Y}$-$\boldsymbol{V_X}$-$\boldsymbol{V_{Z1}}$.

$$\begin{aligned}
&S_{Xa} = S_{Z1a};\ \ if\ [\ (p_{ab} > 0)\ and\ (p_{ca} < 0)\ ]\ \ S_{Xa} \mathrel{+}= 1\\
&S_{Xb} = S_{Z1b};\ \ if\ [\ (p_{bc} > 0)\ and\ (p_{ab} < 0)\ ]\ \ S_{Xb} \mathrel{+}= 1\\
&S_{Xc} = S_{Z1c};\ \ if\ [\ (p_{ca} > 0)\ and\ (p_{bc} < 0)\ ]\ \ S_{Xc} \mathrel{+}= 1
\end{aligned} \tag{25}$$

$$\begin{aligned}
&S_{Ya} = S_{Z1a};\ \ if\ [\ (p_{ab} > 0)\ \ or\ \ (p_{ca} < 0)\ ]\ \ S_{Ya} \mathrel{+}= 1\\
&S_{Yb} = S_{Z1b};\ \ if\ [\ (p_{bc} > 0)\ \ or\ \ (p_{ab} < 0)\ ]\ \ S_{Yb} \mathrel{+}= 1\\
&S_{Yc} = S_{Z1c};\ \ if\ [\ (p_{ca} > 0)\ \ or\ \ (p_{bc} < 0)\ ]\ \ S_{Yc} \mathrel{+}= 1
\end{aligned} \tag{26}$$

The last step would map the obtained space vectors to the particular switching states of the inverter. This mapping depends on the inverter topology, thus it has not been implemented in this work.

V. Algorithm Validation

At each sampling period, the three nearest vectors and their duty cycles are unique, so it does not matter which method is used to find and compute them. The election of one set of vectors among several compatible ones (our free parameter ρ), and also their distribution on time (which vectors are used as $\boldsymbol{V_{Z1}}$ and $\boldsymbol{V_{Z2}}$, and also the value of d_{Z1} compared to d_{Z2}), can be changed and results on dc voltage source balance and common mode voltage are different accordingly.

Proposed algorithm has been evaluated in detail by simulation, and it has been found that vector selection and duty cycles match point by point to others computed using other known techniques [12], [15]. The election of the central vector ($\boldsymbol{V_{Z1}}$ and $\boldsymbol{V_{Z2}}$) obviously differs from time to time.

Fig. 4 illustrates a result of a simulation of this method applied on a two-level inverter in open-loop mode and just in the limit of the linear modulation range ($m \equiv \sqrt{2} \cdot V_{ac} \,/\, V_{dc} = 1.0$). Fig. 5 illustrates the same situation for a nine-level inverter. In both cases the main dc bus was V_{dc} = 566 V, the 50 Hz line-to-line output voltage was V_{ac} = 400 V (rms) and the switching frequency f_S was 6 kHz. PWM signals were generated using a triangular carrier with a resolution of 250 nanoseconds neglecting dead-band effects. All output signals were sampled for displaying at 200 KSPS.

VI. Conclusions

A new method to compute SVM duty cycles has been presented. It is a fast and straightforward algorithm, valid for three-phase multilevel voltage source inverters with any number of levels. A first key idea has been to use instantaneous phase information, actually line voltages, rather than α-β coordinates, because this election has simplified all equations. A second key point has been looking for a suitable hexagon near the reference voltage rather than the usual three nearest vectors: it simplifies local computations and eases the election of the central space vector. Free parameters are left for application specific optimizations, like common-mode voltage rejection and/or balancing of dc sources. Several simulations on different n-level inverters have validated this method, which implementation on digital processors is very easy and direct.

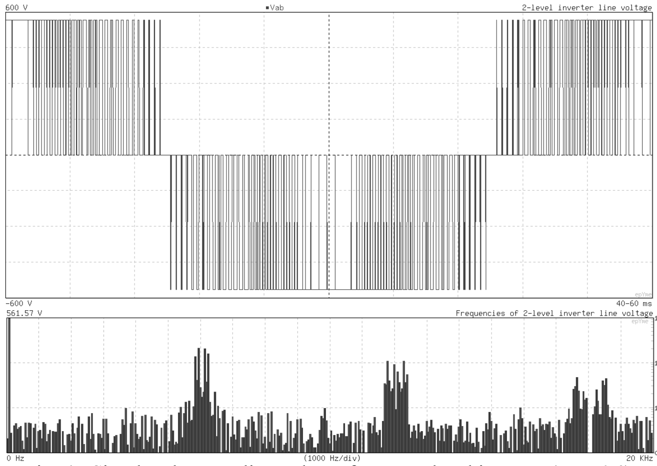

Fig. 4. Simulated output line voltage for a two-level inverter ($m = 1.0$).

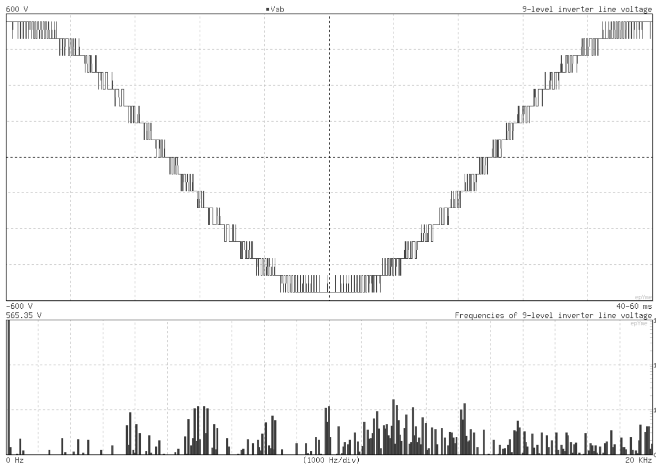

Fig. 5. Simulated output line voltage for a nine-level inverter ($m = 1.0$).

Acknowledgment

Authors would like to acknowledge the financial support for these developments of the *Ministerio de Ciencia e Innovación*, Spain, using FEDER funds, under grant CICYT ENE2007-67417/ALT.

References

[1] J. Rodríguez, J. S. Lai, and F. Z. Peng, "Multilevel inverters: a survey of topologies, controls, and applications," *IEEE Trans. Ind. Electron.*, vol. 49, no. 4, pp. 724–738, Aug. 2002.

[2] J. Rodríguez, S. Bernet, B. Wu, J. Pontt, and S. Kouro, "Multilevel voltage source converter topologies for industrial medium-voltage drives," *IEEE Trans. Ind. Electron.*, vol. 54, no. 6, pp. 2930–2945, Dec. 2007.

[3] A. Nabae, I. Takahashi, and H. Akagi, "A new neural point clamped PWM inverter," *IEEE Trans. Ind. Appl.*, vol. IA-17, no. 5, pp. 518–523, Sep./Oct. 1981.

[4] D. Krug, S. Bernet, S. Saeed Fazel, K. Jalili, and M. Malinowski, "Comparison of 2.3 kV medium-voltage multilevel converters for industrial medium-voltage drives," *IEEE Trans. Ind. Electron.*, vol. 54, no. 6, pp. 2979–2992, Dec. 2007.

[5] R. Teichmann and S. Bernet, "A comparison of three-level converters versus two-level converters for low-voltage drives, traction and utility applications," *IEEE Trans. Ind. Appl.*, vol. 41, no. 3, pp. 855–865, May/Jun. 2005.

[6] K. A. Corzine and J. R. Baker, "Multilevel voltage-source duty-cycle modulation: analysis and implementation," *IEEE Trans. Ind. Electron.*, vol. 49, no. 5, pp 1009–1016, Oct. 2002.

[7] K. Zhou and D. Wang, “Relationship between space-vector modulation and three-phase carrier-based PWM: a comprehensive analysis,” *IEEE Trans. Ind. Electron.*, vol. 49, no. 1, pp. 186–196, Feb. 2002.
[8] B. P. McGrath, D. G. Holmes, and T. A. Lipo, “Optimized space vector switching sequences for multilevel inverters,” *IEEE Trans. Power Electron.,* vol. 18, no. 6, pp. 1293–1301, Nov. 2003.
[9] J. Pou, R. Pindado, and D. Boroyevich, “Voltage-balance limits in four-level diode-clamped converters with passive front ends,” *IEEE Trans. Ind. Electron.,* vol. 52, no. 1, pp. 190–196, Feb. 2005.
[10] H. Zhang, A. von Jouanne, S. Dai, A. K. Wallace, and F. Wang, “Multilevel Inverter Modulation Schemes to Eliminate Common-Mode Voltages,” *IEEE Trans. Ind. Appl.*, vol. 36, no. 6, pp. 1645–1653, Nov./Dec. 2000.
[11] K. A. Corzine, *Operation and design of multilevel inverters*, University of Missouri - Rolla, Tech. Report developed for the Office of Naval Research, Dec. 2003, revised Jun. 2005.
[12] N. Celanovic and D. Boroyevich, “A fast space-vector modulation algorithm for multilevel three-phase converters,” *IEEE Trans. Ind. Appl.,* vol. 37, no. 2, pp. 637–641, Mar./Apr. 2001.
[13] J. H. Seo, C. H. Choi, and D. S. Hyun, “A new simplified space-vector PWM method for three-level inverters,” *IEEE Trans. Power Electron.*, vol. 16, no. 4, pp. 545–550, Jul. 2001.
[14] P. C. Loh and D. G. Holmes, “Flux Modulation for Multilevel Inverters,” *IEEE Trans. Ind. Appl.*, vol. 38, no. 5, pp. 1389–1399, Sep./Oct. 2002.
[15] A. K. Gupta and A. M. Khambadkone, “A space vector PWM scheme for multilevel inverters based on two-level space vector PWM,” *IEEE Trans. Ind. Electron.,* vol. 53, no. 5, pp. 1631–1639, Oct. 2006.
[16] S. K. Mondal, J. O. P. Pinto, and B. K. Bose, “A neural-network-based space-vector pwm controller for a three-level voltage-fed inverter induction motor drive,” *IEEE Trans. Power Electron.*, vol. 38, no. 3, pp. 660–669, May/Jun. 2002.
[17] L. G. Franquelo *et al.*, “Three-dimensional space-vector modulation algorithm for four-leg multilevel converters using *abc* coordinates,” *IEEE Trans. Ind. Electron.*, vol. 53, no. 2, pp. 458–466, Apr. 2006.
[18] R. Teodorescu, F. Blaabjerg, J. K. Pedersen, E. Cengelci, and P. N. Enjeti, “Multilevel Inverter by Cascading Industrial VSI,” *IEEE Trans. Ind. Electron.*, vol. 49, no. 4, pp 832–838, Aug. 2002.
[19] H. W. Van der Broeck, H. C. Skudelny, and G. V. Stanke, “Analysis and realization of a pulsewidth modulator based on voltage space vectors,” *IEEE Trans. Ind. Appl.*, vol. 24, no. 1, pp. 142–150, Jan./Feb. 1988.
[20] J. Youm and B. Kwon, “An effective software implementation of the space-vector modulation,” *IEEE Trans. Ind. Electron.*, vol. 46, no. 4, pp. 866–868, Aug. 1999.
[21] G. L. Peters and G. A. Covic, “Eliminating low frequency common-mode voltage using space vector techniques,” *Electronics Letters*, vol. 39, no. 18, Sep. 2003.
[22] S. de Pablo, A. B. Rey, L. C. Herrero, and J. M. Ruiz, “A simpler and faster method for SVM implementation,” in *Proc. 12th Eur. Conf. Power Electronics and Applications (EPE’2007)*, Aalborg, Denmark, Sep. 2007.
[23] S. de Pablo, A. B. Rey, L. C. Herrero, and F. Martínez, “A graphical method for SVM duty cycles computation. Application on two-level and multi-level inverters,” in *Proc. 13th Eur. Conf. Power Electronics and Applications (EPE’2009)*, Barcelona, Spain, Sep. 2009.