Generalised direct positioning approach for multilevel space vector modulation: theory and DSP–implementation

M. Tavakoli Bina

Abstract: In three-phase DC/AC multilevel converters, the space vector modulation is the preferred pulse width modulation (PWM) technique. A particular area relative to the space vector modulation (SVM) analysis for multilevel converter is focused and a direct positioning approach for finding the location of the reference vector among available triangles within the SVM diagram is proposed. The developed procedure is then generalised for an n-level converter by proposing a $(n-1)^2 \times 1$ complex matrix that locates the exact place at the end of the reference vector. To verify the proposed method, the whole modulation process was implemented with DSP for three- and four-level converters. To prove the effectiveness of the proposed method in higher-level cases, it was also applied to five- and six-level SVMs. Experimental results validate the direct positioning approach to be an accurate and reliable technique for digital implementation purposes, reducing the total modulation time as well as saving it for complex control techniques needed in industrial applications of SVM.

1 Introduction

Multilevel converters have significant advantages in high-power, medium- and high-voltage applications [1]. Compared with bi-level devices, series interconnection of switches can be avoided, and quality of the output voltage is improved. The complexity of modulation strategy and the number of switches is increased [2]. Fig. 1*a* shows an n-level midpoint diode-clamped (MPDC) converter that includes $(n-1)$ DC capacitors along with $(n-2)$ midpoints [1]. However, it has been shown in [2] that practical and theoretical limits for multilevel converters exist when n is larger than 3 and voltage regulation is not managed by external DC sources. In other words, multilevel MPDC converters with $n > 3$ can potentially be used in applications that involve non-active powers such as active filtering and reactive power compensation (static VAR compensation and STATCOM). Moreover, among various suggested topologies, Fig. 1*b* outlines an n-level cascade converter [3], and Fig. 1*c* describes an n-level flying capacitor converter with n capacitors [4].

Considering a standard space vector modulation (SVM) strategy for an n-level converter, where three-phase abc voltages are transferred to the dq plane, a common approach to compose the reference vector is the linear combination of three nearest switching instants to the reference vector during each modulation cycle. Thus, every sextant of an n-level SVM is divided into $(n-1)^2$ triangles, making a total of $6(n-1)^2$ triangles. Nevertheless, it is imperative for the digital implementation of the SVM to locate the exact position at the end of the reference vector among $(n-1)^2$ available triangles of each sextant. Then, the modulation is followed by the well-known calculation of three duty ratios corresponding to the three switching instants assigned to the classified triangle.

A certain control and regulation strategy produces an instantaneous reference signal for the modulator. The modulator will then determine the switching instants based on this generated reference waveform. Realisation of the whole process must take place within a switching period. Hence, the long processing time delay relative to the modulator limits the maximum bandwidth of the controller (and the sampling time) at a given switching period or the switching frequency. Therefore the decrease in processing time of the modulator results in a better exploitation of the multilevel converter's advantages [1–4]. Thus, it is essential to develop detailed experimental research on various sections of space vector modulation.

Conventional methods utilise the arithmetic and trigonometric evaluation of the functions using look-up tables for positioning purposes, which is implemented on a TMS320C30 DSP microprocessor around 40.7 μs (30% of the total control algorithm) [5]. Further, to reduce the complexity of implementing SVM to a bi-level converter, a counter-propagation neural network is employed. The use of this method uses the two nearest adjacent switching instants to track the reference vector and at the same time avoids the necessity of on-line computation of the trigonometric function 'sin'. Compared with conventional SVM implementation, the requirement of less computation processing time (about 27.3 μs) will lead to the reduction of sampling time and the provision of higher switching frequencies. It is then extended to the classified neural network in three-level cases [6] and simulated for finding the triangle number wherein the end of the reference vector lies. An alternative approach is introduced in [7] for multilevel inverters, whereby algebraic and trigonometric functions

doi:10.1049/iet-epa:20060441

Paper first received 26th October 2006 and in revised form 6th May 2007

The author is with the Faculty of Electrical Engineering, K. N. Toosi University of Technology, Seid Khandan, PO Box 16315-1355, Tehran 16314, Iran

E-mail: tavakoli@kntu.ac.ir

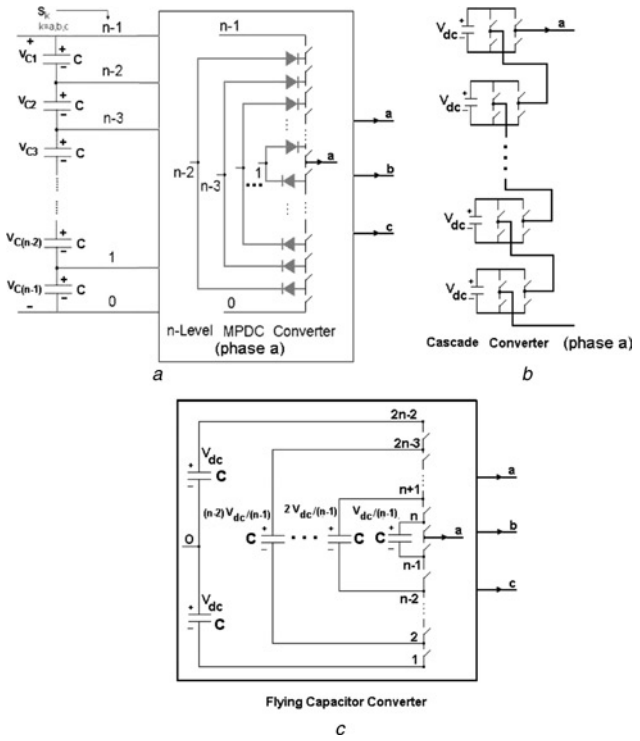

Fig. 1 *General diagrams of multilevel converters*
a Diode clamped
b Flying capacitor
c Cascade family

for the reference vector conditioning are used followed by transition of system coordinates and the reference vector to establish regions and switching times.

The principal objective of this paper is vector classification of SVM which proposes a direct positioning technique. This method can be applied to three-level converters and any higher-level ones. For a general n-level converter, an $(n-1)^2 \times 1$ characteristic vector is introduced in terms of the dq reference components. This enables the process to detect the position of the reference vector, where certain rules are formulated for an n-level converter. Then, the three adjacent switching instants assigned to the detected triangle are selected to track the reference vector, formulating the on-durations of the three switching states for digital implementation. Furthermore, implementation of the proposed method is arranged using a 150 MIPS digital signal processor (TMS3202812F DSP microprocessor), where various experimental results are presented for up to six-level converters in order to emphasise the accuracy and quickness of the proposed technique. Employing conventional control strategies [8, 9], the reference voltages are emulated using a C program for multilevel converters. They are then supplied to the DSP Code Composer for the whole modulation process, including the proposed positioning method and generating modulating signals for the converter's switches.

2 Positioning method

Balanced DC-link voltages of an n-level converter (see Fig. 1) is assumed. Fig. 2*a* shows the first sextant of the SVM diagram of a three-level converter that divided into four triangles, and a four-level converter is introduced by Fig. 2*b* with nine triangles. Every triangle consists of a number of switching instants, some of which are redundant. The employed control strategy selects the most suitable redundant vector for the modulation scheme.

2.1 Theoretical description

Let us assume that the three-phase voltages are transferred from *abc* to the *dq* plane, where $\boldsymbol{V}_{\text{ref}} = (\boldsymbol{V}_d, \boldsymbol{V}_q)$ is associated with the three-phase voltage reference. Also, all lines within the first sextant are numbered as shown in Figs. 2*a* and *b*. For a three-level converter, the six line equations are labelled as L_1–L_6, and those of a four-level converter as L_1–L_9. When an arbitrary vector $(\boldsymbol{V}_d, \boldsymbol{V}_q)$ locates above or below a line L_i, then according to the basic algebra, $L_i(\boldsymbol{V}_d, \boldsymbol{V}_q)$ gives either a positive or a negative value. Obviously, $L_i(\boldsymbol{V}_d, \boldsymbol{V}_q) = 0$ is valid for all points located on L_i. Note that for the three-level case of Fig. 2*a* it would also be possible to test a few simple conditional expressions to seek for the reference voltage position (triangle no) as the following C code fragment

$$\text{if } (L_4(\boldsymbol{V}_d, \boldsymbol{V}_q) \leq 0) \text{ triangle\_no} = 4;$$

$$\text{else if } (L_2(\boldsymbol{V}_d, \boldsymbol{V}_q) > 0) \text{ triangle\_no} = 3;$$

$$\text{else if } (L_6(\boldsymbol{V}_d, \boldsymbol{V}_q) \geq 0) \text{ triangle\_no} = 2;$$

$$\text{else triangle\_no} = 1;$$

However, these conditional expressions consist of some lengthy algebraic operations. To distinguish and generate similar C statements for the four-level case of Fig. 2*b* and other higher-level converters is very complicated as the number and complexity of conditional expressions increases dramatically. Vectors located beyond $L_3(\boldsymbol{V}_d, \boldsymbol{V}_q)$ in Fig. 2*a* are also deemed to be inside the sextant by the above C code. Therefore it is necessary to develop a generalised n-level positioning technique that is able to locate the accurate position of a given vector quickly.

Let us begin our proposal by introducing upward and/or downward triangles inside a sextant as triangles headed above and/or below their horizontal sides, respectively. Then, a rotating direction is considered for every triangle, either clockwise for downward triangles or counter-clockwise for upward triangles. Note that rotating direction facilitates generalisation of the proposal to n-level converters (see Section 3). Moreover, because the sign of $L_{i=a,b,c}(\boldsymbol{V}_d, \boldsymbol{V}_q)$ is used for the positioning technique, let us also assign positive values to 'one' and negative values to 'zero'. Extending and evaluating the three sides of an arbitrary triangle creates seven regions with seven different sets of signs and establishes the basis of the proposal. Fig. 3*a* illustrates such an extension for an upward triangle numbered as 4, and Fig. 3*b* for a downward triangle numbered as 0. It is noticeable that a counter-clockwise upward triangle is the mirror image of a clockwise downward triangle with respect to the horizontal line.

Then, points located inside all the eight regions of Figs. 3*a* and *b* were evaluated by their surrounding line equations (using the explained rule) and gathered in Table 1. Six regions numbered 1, 2, 3, 5, 6 and 7 are shared by both figures, introducing exactly the same evaluated signs for the three sides L_a, L_b and L_c. The first column gives the number assigned to these eight possible regions, whereas columns 2–4 show the evaluated signs of triangle's sides for the corresponding regions. Columns 5 and 6 show the signs of L_aL_b and L_aL_c according to the rotating direction of the triangle.

Assume $(\boldsymbol{V}_d, \boldsymbol{V}_q)$ is the vector to be recognised whether or not it is located inside an arbitrary triangle either 4 in Fig. 3*a* or 0 in Fig. 3*b*. Hence, the horizontal side of the

triangle ($L_a(V_d, V_q)$) and the other two sides ($L_b(V_d, V_q)$ and $L_c(V_d, V_q)$) are evaluated to obtain the products of columns 5 and 6 of Table 1. Although both products have negative signs for both the triangles 4 (Fig. 3*a*) and 0 (Fig. 3*b*), it can be seen that for the six common regions at least one product is positive. Note that the horizontal side always stands for L_a, but both clockwise and anti-clockwise directions select sequence of the other two sides L_b and L_c to simplify generalisation of the method for n-level converters. In practice, the positioning technique is based on one simple rule: the vector (V_d, V_q) lies inside a triangle (either upward or downward) when the two products $L_a(V_d, V_q)\, L_b(V_d, V_q)$ and $L_a(V_d, V_q)\, L_c(V_d, V_q)$ are negative.

2.2 Characteristic vector

The above positioning rule for one triangle can now be applied to all available triangles of the first sextant by forming a complex vector, where each row describes one triangle. Elements of this matrix consist of two products similar to those of the columns 5 and 6 in Table 1. When introducing two negative signs, only one row determines the winner. For example, assume an arbitrary vector (V_d, V_q) to be classified inside a three-level SVM diagram like that of Fig. 2*a*. Evaluating three sides of the four triangles results in a complex 4×1 vector as follows

$$\mathrm{CHV}(3) = \begin{bmatrix} L_1L_3 \\ L_2L_6 \\ L_2L_3 \\ L_1L_4 \end{bmatrix} + \mathrm{j} \begin{bmatrix} L_1L_6 \\ L_2L_4 \\ L_2L_5 \\ L_1L_5 \end{bmatrix} \tag{1}$$

where CHV(3) is the three-level characteristic vector for the positioning technique. Applying the same rules, matrices similar to those of (1) can be further extended for higher-level converters. Here, CHV(4) of a four-level converter can be similarly performed by a 9×1 vector

$$\mathrm{CHV}(4) = \begin{bmatrix} L_1L_4 \\ L_2L_9 \\ L_2L_4 \\ L_3L_8 \\ L_3L_4 \\ L_1L_5 \\ L_2L_8 \\ L_2L_5 \\ L_1L_6 \end{bmatrix} + \mathrm{j} \begin{bmatrix} L_1L_9 \\ L_2L_5 \\ L_2L_8 \\ L_3L_5 \\ L_3L_7 \\ L_1L_8 \\ L_2L_6 \\ L_2L_7 \\ L_1L_7 \end{bmatrix} \tag{2}$$

It will be experimentally shown that when the number of levels increases, the reference classification delay of the proposed method will rise very slowly as opposed to other techniques.

2.3 Generalisation of CHV(n) for an n-level case

The theory can be generalised to an n-level case by introducing a typical CHV(n). Let us define an auxiliary matrix $\boldsymbol{C}(n, k)$ as

$$\boldsymbol{C}(n, k) = \begin{bmatrix} L_1L_{n+k-1} \\ L_2L_{3n-k-2} \\ L_2L_{n+k-1} \\ L_3L_{3n-k-3} \\ L_3L_{n+k-1} \\ L_4L_{3n-k-4} \\ L_4L_{n+k-1} \\ \vdots \\ L_{n-k}L_{2n} \\ L_{n-k}L_{n+k-1} \end{bmatrix} + \mathrm{j} \begin{bmatrix} L_1L_{3n-k-2} \\ L_2L_{n+k} \\ L_2L_{3n-k-3} \\ L_3L_{n+k} \\ L_3L_{3n-k-4} \\ L_4L_{n+k} \\ L_4L_{3n-k-5} \\ \vdots \\ L_{n-k}L_{n+k} \\ L_{n-k}L_{2n-1} \end{bmatrix} \tag{3}$$

where $k = 1, 2, 3, \ldots, n-1$, and $\boldsymbol{C}(n, k)$ includes $2n - 2k - 1$ complex elements. Then, the following $(n-1)^2 \times 1$ characteristic vector can be arranged for an n-level SVM

$$\mathrm{CHV}(n) = \begin{bmatrix} \boldsymbol{C}(n, 1) \\ \boldsymbol{C}(n, 2) \\ \vdots \\ \boldsymbol{C}(n, n-1) \end{bmatrix} \tag{4}$$

Note that the size of CHV(n) can be double-checked by the well-known relationship $\sum_{k=1}^{n-1} (2n - 2k - 1) = (n-1)^2$, when one winner row is extracted out of $(n-1)^2$ rows.

3 Formulation of the direct positioning rules

The proposed positioning technique can now be put together as follows:

- Assign line numbers $L_1, L_2, \ldots, L_{3n-3}$ to the first sextant of SVM diagram of an n-level converter. Number them starting from the lowest horizontal side straight up to the last one, then turn clockwise and similarly continue numbering as illustrated in Fig. 3*c*. Note that the rotational direct arrows help generalisation of the proposed method to an n-level SVM through arranging a unique rule for producing (3) and (4).
- Assign triangle numbers $1, 2, 3, \ldots, (n-1)^2$ to an n-level converter as of three- and four-level cases illustrated by Figs. 2*a* and *b*.
- Evaluate line equations $L_1, L_2, \ldots, L_{3n-3}$ by a given reference voltage vector $\boldsymbol{V}_{\mathrm{ref}} = (V_d, V_q)$ using the following generalised relationships

$$\begin{aligned} L_j &= V_q - \frac{\sqrt{3}}{2(n-1)}(j-1), \qquad j = 1, 2, \ldots, n-1 \\ L_k &= V_q + \sqrt{3}V_d - \frac{\sqrt{3}}{(n-1)}(2n-k-1), \\ &\qquad k = n, n+1, \ldots, 2n-2 \\ L_l &= V_q - \sqrt{3}V_d + \frac{\sqrt{3}}{(n-1)}(l-2n+1), \\ &\qquad l = 2n-1, 2n, \ldots, 3n-3 \end{aligned} \tag{5}$$

- Evaluate CHV(n) using (3)–(5).
- Specify the winner triangle as a row of CHV(n) that shows two negative signs.

3.1 Simplifying CHV(n) for DSP implementation

Here we discuss another step towards the implementation of the proposed method with microprocessors. This is done by simplifying CHV(n) given by (3) and (5), with the objective of lowering the number of arithmetic operations in (4) in

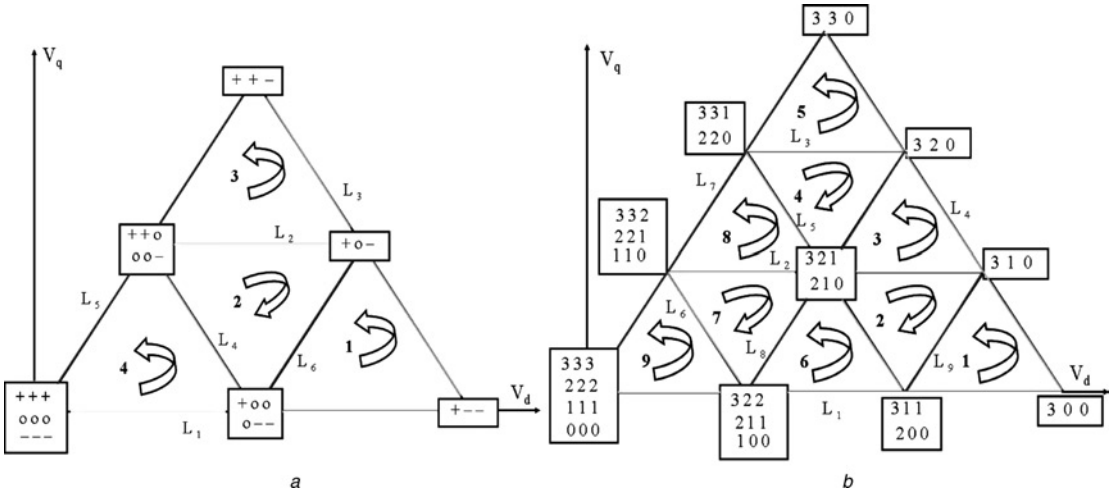

Fig. 2 *Numbering and rotating direction for triangles of the first sextant*

a Three-level converters
b Four-level converters

order to reduce the CPU calculation time for the positioning technique. Initially, two auxiliary variables A and B are defined as follows

$$\begin{aligned} A &= \boldsymbol{V}_q - \sqrt{3}\boldsymbol{V}_d \\ B &= \boldsymbol{V}_q + \sqrt{3}\boldsymbol{V}_d \end{aligned} \tag{6}$$

Then, using (1), CHV (3) for a three-level case can be simplified as

$$\begin{bmatrix} B-\sqrt{3} \\ \left(\boldsymbol{V}_q - \frac{\sqrt{3}}{4}\right)\left(A+\frac{\sqrt{3}}{2}\right) \\ \left(\boldsymbol{V}_q - \frac{\sqrt{3}}{4}\right)\left(B-\sqrt{3}\right) \\ B-\frac{\sqrt{3}}{2} \end{bmatrix} + \mathrm{j}\begin{bmatrix} A+\frac{\sqrt{3}}{2} \\ \left(\boldsymbol{V}_q - \frac{\sqrt{3}}{4}\right)\left(B-\frac{\sqrt{3}}{2}\right) \\ \left(\boldsymbol{V}_q - \frac{\sqrt{3}}{4}\right)A \\ A \end{bmatrix} \tag{7}$$

Here the computation time for evaluation of (7) is much less than that of (1) because repeating terms are computed once, and A is always negative for the first sextant (see Fig. 2). Thus, there are only four different expressions in (7), including $\{\boldsymbol{V}_q - \sqrt{3}/4, A + \sqrt{3}/2, B - \sqrt{3}, B - \sqrt{3}/2\}$. Also, the sign of these expressions are decisive for finding the winner row of the CHV(3). Hence, the quick bitwise operations are employed instead of lengthy multiplications and relational operators. Appendix A lists the C code fragment for direct positioning of a three-level SVM case. Moreover, the simplified CHV(4) for a four-level case was worked out as a 9×1 complex matrix

$$\mathrm{CHV}(4) = \begin{bmatrix} B-\sqrt{3} \\ \left(\boldsymbol{V}_q - \frac{\sqrt{3}}{6}\right)\left(A+\frac{2\sqrt{3}}{3}\right) \\ \left(\boldsymbol{V}_q - \frac{\sqrt{3}}{6}\right)\left(B-\sqrt{3}\right) \\ \left(\boldsymbol{V}_q - \frac{2\sqrt{3}}{6}\right)\left(A+\frac{\sqrt{3}}{3}\right) \\ \left(\boldsymbol{V}_q - \frac{2\sqrt{3}}{6}\right)\left(B-\sqrt{3}\right) \\ B-\frac{2\sqrt{3}}{3} \\ \left(\boldsymbol{V}_q - \frac{\sqrt{3}}{6}\right)\left(A+\frac{\sqrt{3}}{3}\right) \\ \left(\boldsymbol{V}_q - \frac{\sqrt{3}}{6}\right)\left(B-\frac{2\sqrt{3}}{3}\right) \\ B-\frac{\sqrt{3}}{3} \end{bmatrix} + \mathrm{j}\begin{bmatrix} A+\frac{2\sqrt{3}}{3} \\ \left(\boldsymbol{V}_q - \frac{\sqrt{3}}{6}\right)\left(B-\frac{2\sqrt{3}}{3}\right) \\ \left(\boldsymbol{V}_q - \frac{\sqrt{3}}{6}\right)\left(A+\frac{\sqrt{3}}{3}\right) \\ \left(\boldsymbol{V}_q - \frac{2\sqrt{3}}{6}\right)\left(B-\frac{2\sqrt{3}}{3}\right) \\ \left(\boldsymbol{V}_q - \frac{2\sqrt{3}}{6}\right)A \\ A+\frac{\sqrt{3}}{3} \\ \left(\boldsymbol{V}_q - \frac{\sqrt{3}}{6}\right)\left(B-\frac{\sqrt{3}}{3}\right) \\ \left(\boldsymbol{V}_q - \frac{\sqrt{3}}{6}\right)A \\ A \end{bmatrix} \tag{8}$$

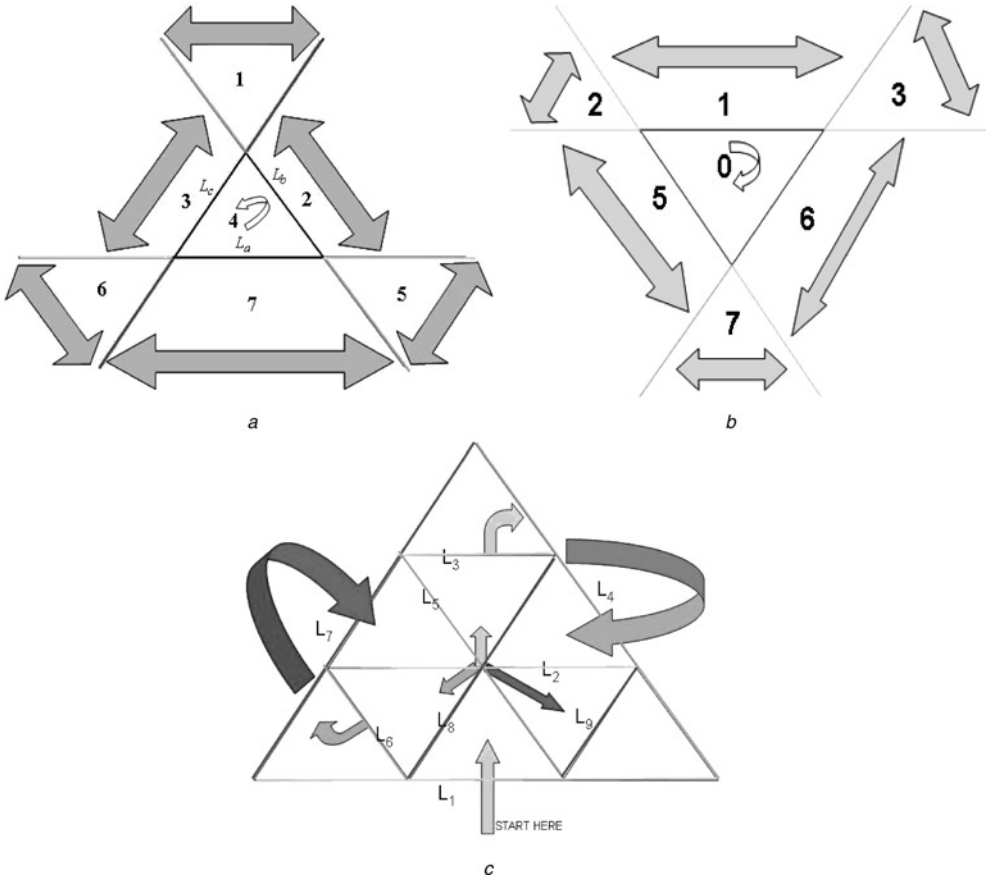

Fig. 3 *Illustration of the seven regions inside and outside*

a Upward triangle
b Downward triangle
c Numbering $3(n-1)$ lines of the fist sextant of an n-level converter for $n = 4$

The four-level case includes seven different terms

$$\left(\left\{\begin{array}{c} V_q - \sqrt{3}/6,\ V_q - \sqrt{3}/3,\ A + \sqrt{3}/3,\ A + 2\sqrt{3}/3, \\ B - \sqrt{3}/3,\ B - 2\sqrt{3}/3,\ B - \sqrt{3} \end{array}\right\}\right),$$

where their signs can be used in detecting the position of the reference vector using bitwise operations. Thus, only three more terms have to be calculated compared with the three-level case. Also, only 10 and 13 different terms have to be evaluated for five- and six-level cases, where this figure for a general n-level case equals $3n - 5$. Therefore it is expected that the proposed technique offers a desirable classification speed for higher-level converters because the involved arithmetic operation linearly increases slowly relative to the number of levels. This is also confirmed by the executed practical work. Appendix A also provides the C code for the four-level case to be compared with three-level vector classification.

Table 1: Evaluation of all possible regions when the three sides of a triangle are extended

Region no.	Side 1 L_a	Side 2 L_b	Side 3 L_c	L_aL_b	L_aL_c
1	1	1	1	1	1
2	1	1	0	1	0
3	1	0	1	0	1
4 (Δ)	1	0	0	0	0
0 (∇)	0	1	1	0	0
5	0	1	0	0	1
6	0	0	1	1	0
7	0	0	0	1	1

4 Assessment of the proposed method

Let us consider a multilevel DC/AC converter. One common approach is to supply a predetermined AC voltage by this converter in order for the DC-link capacitor voltages to be well balanced [8, 9]. Here it is assumed that the reference vector is modulated by the converter as shown in the applied references to the modulation (Fig. 4).

In brief, the reference waveforms are transferred to the dq plane. The resultant dq voltages are also normalised by the effective line-to-line voltage using abc/dqN (outgoing signals are V_d and V_q) to push the values within the region [0, 1]. For an arbitrary instant, this vector will lie

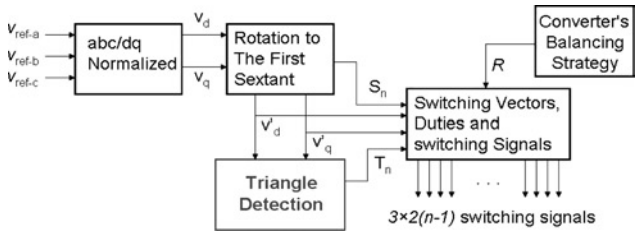

Fig. 4 *Whole space vector modulating (SVM) diagram of an n-level converter, showing the positioning technique (Triangle Detection) in greyscale*

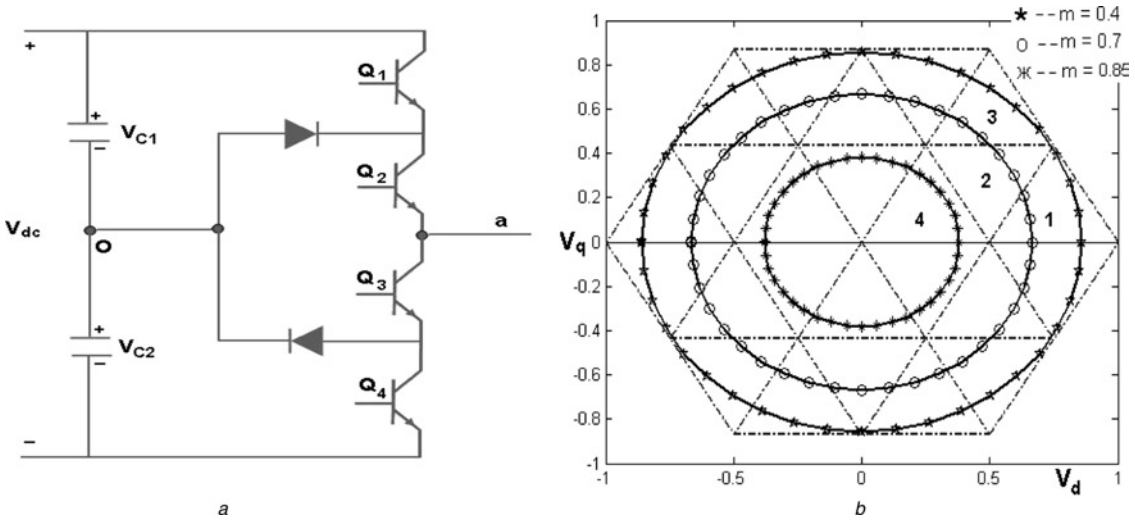

Fig. 5 *Conventional space vector diagram of a three-level inverter*

a Three-level diode clamped inverter
b Three different sinusoidal voltage references ($m = 0.4$, 0.7, 0.85) that practically arrange the dq-sample positions in all 24 triangles

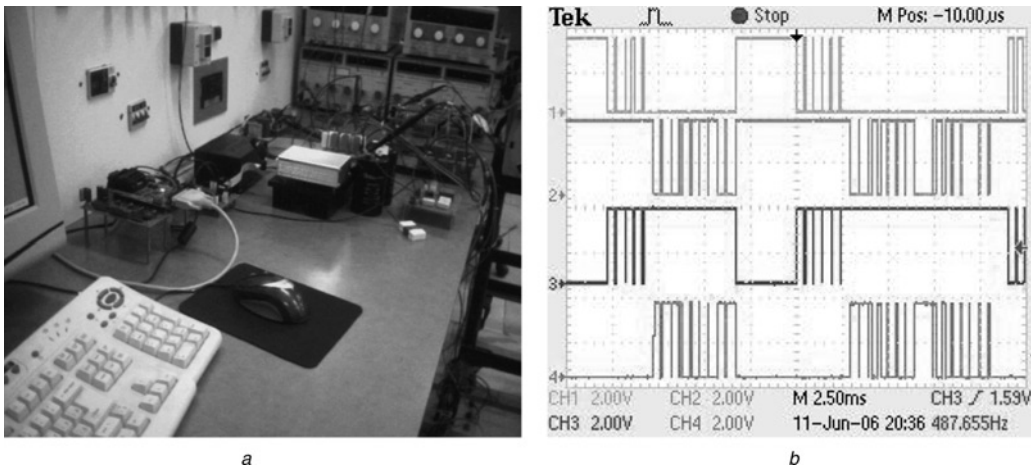

Fig. 6 *DSP-implementation of switching pulses for a three-level inverter*

a Practical test bench including the DSP microcontroller system
b typical switching signals for one arm of a three-level inverter

in one of the six possible sextants (S_n), which is rotated back to the first sextant using RFS (outgoing signals are V'_d and V'_q). This is then applied to the proposed positioning technique, labelled by Triangle Detection in Fig. 4, to examine the exact location of (V'_d, V'_q) among $(n-1)^2$ triangles employing TD (out going signal is T_n) using (3)–(5). Then, the sextant number S_n and triangle number T_n as well as the rotated reference vector are used to select switching states surrounding the triangle in question. Finally, the reference vector is synthesised using the chosen

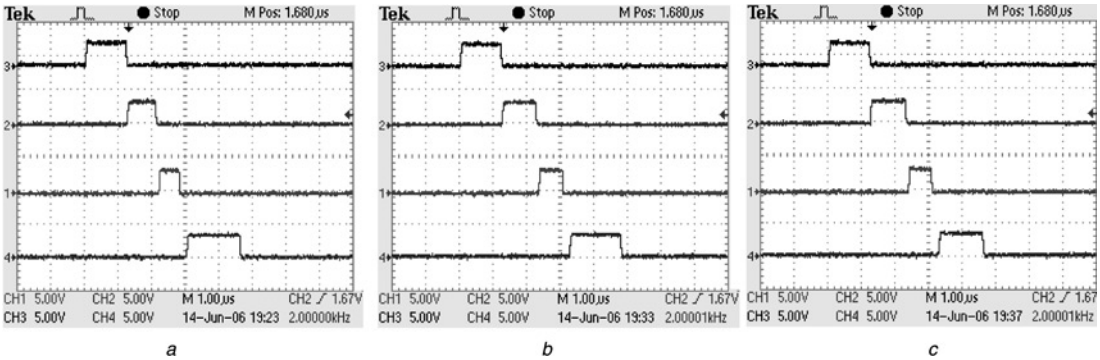

Fig. 7 *Typical measured times for processing the abc/dqN, RFS, TD and SVDS by the DSP for three different modulation indexes*

a $m = 0.85$
b $m = 0.7$
c $m = 0.4$

switching states. Considering a switching period, duty ratios of the selected switching states are calculated using a fast algorithm and $2(n-1)$ switching signals are generated for each phase of an n-level converter using SVDS ($3 \times 2(n-1)$ signals in total). The balancing strategy of the DC-link capacitors provides the signal R to enable the modulator in order to make a decision on choosing a switching instant among possible redundant states.

4.1 Experimental results

The modulation diagram of Fig. 4 has been fully programmed with Code Composer Studio for the fixed-type 150MIPS TMS320F2812 DSP microprocessor, highlighting the foregoing positioning methodology as the principal objective. Experimental results include switching modulation for both three- and four-level converters, extending it to five- and six-level cases for an overall assessment.

4.1.1 Three-level modulation:sinusoidal implementation: Fig. 5*a* shows the phase a of an MPDC three-level inverter having four switches per phase. Assume that the three-phase reference voltages applied to the modulator (see Fig. 4) are like $\{220\,m\cos(\omega t), 220\,m\cos(\omega t - 2\pi/3), 220\,m\cos(\omega t + 2\pi/3)\}$, where $\omega = 100\pi$ is in radian frequency, and m is the variable modulation index that can be changed by the program. Fig. 5*b* illustrates the locus of the reference vector within the dq-plane for the three chosen modulation indices $m \in \{0.85,\ 0.7,\ 0.4\}$; each includes 40 samples at a 2 kHz switching frequency. Thus, all 24 triangles of the six sextants are involved in the assessment process.

To implement the three-level SVM technique, all the four modulating blocks (see Fig. 4) were programmed, and then uploaded by the DSP microcontroller system. Fig. 6*a* illustrates the practical test bench, consisting of voltage and current monitoring modules applied to the DSP to control and modulate the converter. Fig. 6*b* depicts a typical

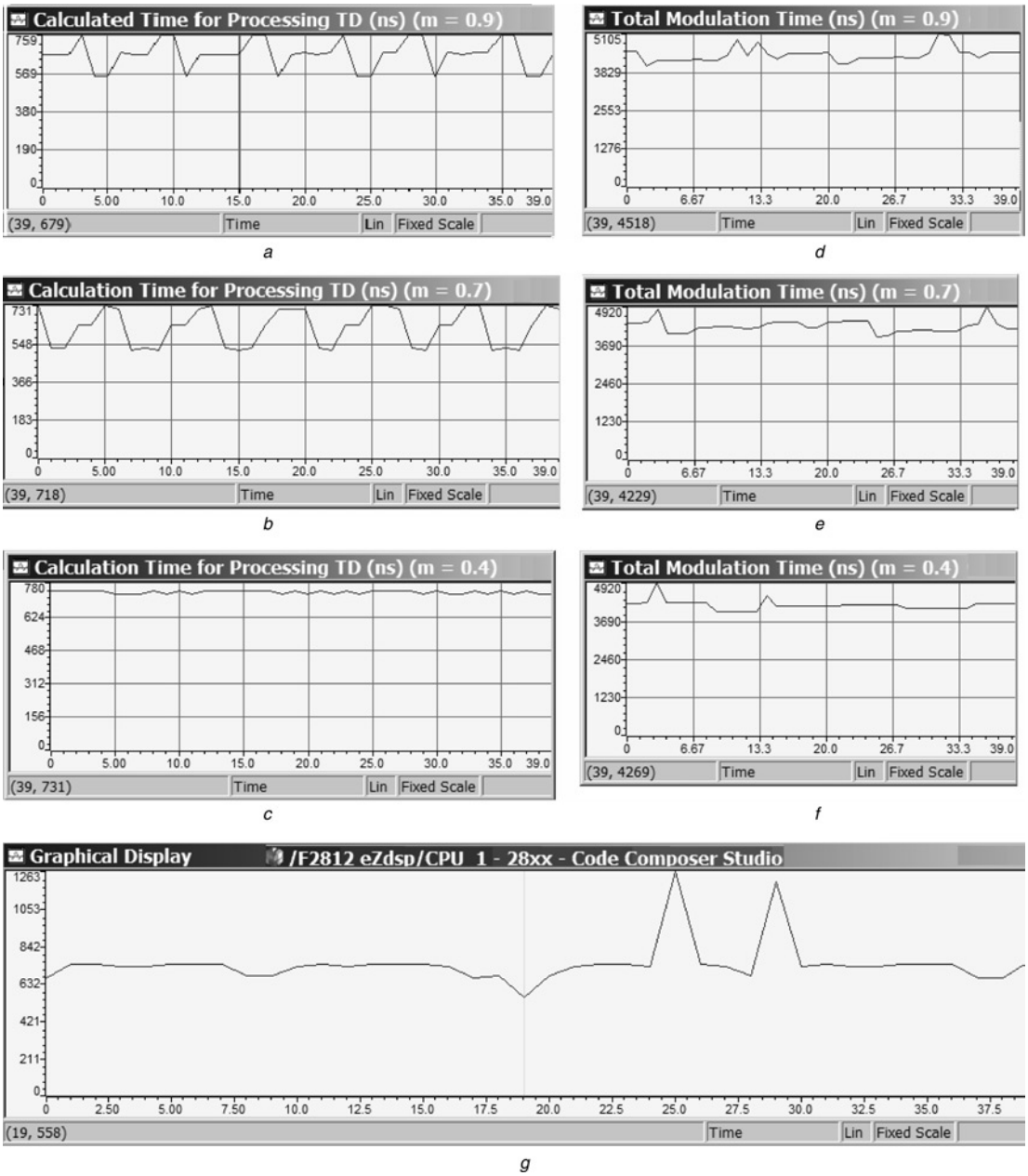

Fig. 8 *Calculation times of the TD and the total modulation diagram of Fig. 4 under sinusoidal voltage references*

a–c Calculated time by DSP for processing TD in ns
d–f total modulation times in ns
g possible typical peak time of the TD due to the interrupt service by the CPU

Table 2: Experimental results for the implemented positioning technique and SVM by DSP

Modulation index	Process time for the TD, ns	Peak time of the TD, ns	Three-level total modulation time, μs
0.4	740–758	1263	4.920
0.7	560–731	1263	4.920
0.85	560–758	1263	5.105

experimental switching signal generated by DSP when $m = 0.85$ for the four switches Q_1–Q_4. Furthermore, to assess the proposed positioning technique, Fig. 7 shows typical measured times for the four modules in Fig. 4 (abc/dqN, RFS, TD and SVDS) for the three engaged modulation indices. These measurements have been arranged by toggling four general purpose I/Os, where the four pulses (from the top) show the needed time to perform abc/dqN, RFS, TD and SVDS, respectively.

Experimental results indicate that the four tasks are being done sequentially (Fig. 4). Although the TD (the third pulse) managed by the proposed positioning method is very fast (<0.8 μs), the whole modulation process also takes less than 5 μs for various three-level experiments. It should also be noted that because the triangle detection (TD) is always done in the first sextant, the measured time for the TD excludes almost the fixed time taken to perform RFS. However, this can be compensated by adding it to the measured time for the TD.

Additionally, a continuous up-counting timer of the DSP is used to calculate the required time to perform the TD for a whole power system period (20 ms). Since the sampling frequency of the implemented application is 2.0 kHz, 40 samples per period were processed and stored; they were then displayed by the DSP graph to obtain an extensive assessment on the speed of the direct positioning technique. This has been repeated for the three modulation indices, where triangles 1–4 are detected in a sequential manner. The resultant graphs are displayed in Fig. 8. When $m = 0.85$ or $m = 0.7$ (Figs. 8*a* and *b*), three different detection times (560, 675 and 758 ns) are observed, which correspond to three triangles labelled as 1, 2 and 3 in Fig. 5*b*, respectively. When $m = 0.4$, only triangle number 4 is involved in the reference locus, which is the longest detected triangle (typically 750 ns) as it is shown by Fig. 8*c*. Table 2 lists the calculated times needed for the TD as well as the three-level total modulation times for different experimental cases.

The third column in Table 2 gives the peak calculation times for the TD, which can be typically observed in Fig. 8*g*. These times are longer than those of Figs. 8*a*–*c*, because the multilevel modulation is managed by DSP using two timers to control the whole process: one for arranging duty ratios and switching signals and another for allowing the DSP to perform the control strategy and the SVM within a switching period. Whenever the TD is processed by the DSP, if the CPU receives an interrupt request by the first timer, then the time taken by the interrupt service is added up to the time of the TD. Measured times show that the proposed technique is very fast even when the CPU is serving another interrupt request.

4.1.2 Three-level modulation: non-sinusoidal implementation: Let us now consider a three-phase non-sinusoidal voltage reference (phase a: $83.6\cos(\omega t) + 45\cos(3\omega t) + 75\cos(5\omega t)$), where its dq locus is shown by Fig. 9*a*. The chosen reference passes through 22 triangles. Fig. 9*b* depicts typical measured times to process the four

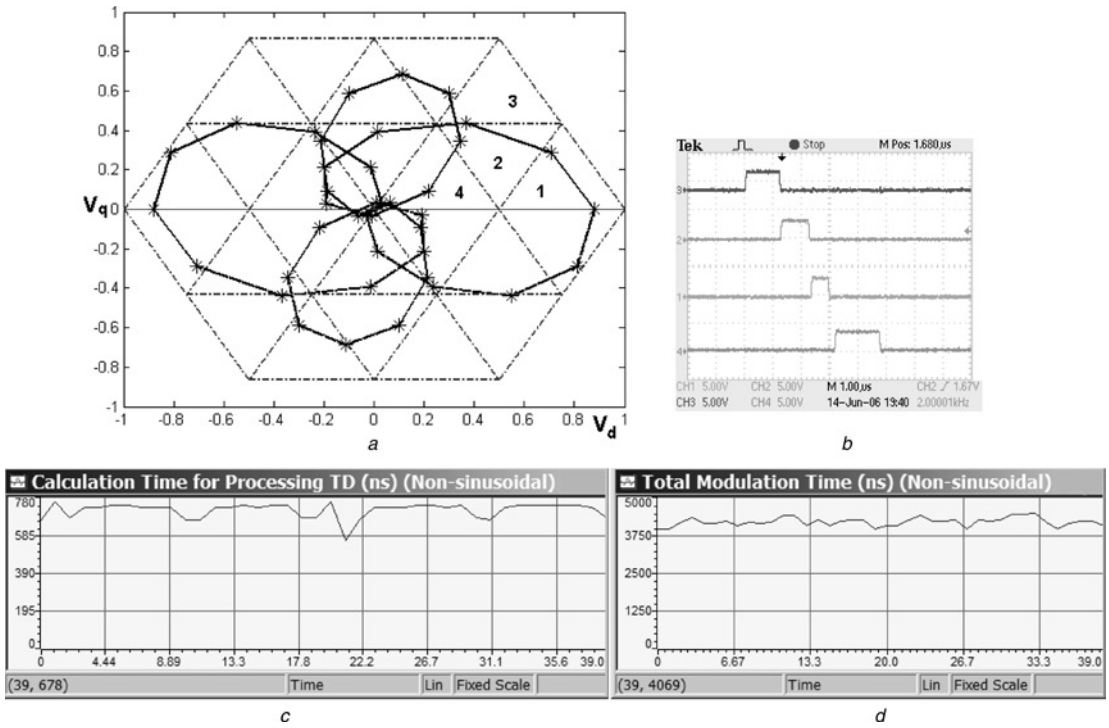

Fig. 9 *Measuring and calculated times of the SVM diagram for a non-sinusoidal voltage reference*

a Three-level dq-locus of the non-sinusoidal reference passing through 22 triangles
b Measured times of the abc/dqN, RFS, TD and SVDS by toggling technique
(c)–(d) Calculated time for 40 samples using an up-counting timer for a non-sinusoidal three-level reference for the TD and the whole modulation process, respectively

modules in Fig. 4 using the DSP toggling technique. Experimental results in Fig. 9*b* show that the TD (the third pulse) is a very fast algorithm, taking up a small fraction of the whole modulation process (typically less than 1 μs).

Further, a continuous up-counting timer worked out the time needed to process the TD as well as the whole modulation process. Again, the sampling period was 500 μs, providing the graph of Fig. 9*c* to process all samples that varies within [560, 758] ns. This is because the *dq*-locus of the reference passes through nearly all triangles as shown by Fig. 9*a*. Moreover, the required time for managing the entire modulation is illustrated by Fig. 9*d*, which varies within [4000, 4650]ns. The calculated times could also be longer, if the CPU gets involved in serving another interrupt request about 500 ns. However, this has little effect on the proposed scheme, as was discussed in the previous section.

4.1.3 Four-level modulation: Here the proposed method is applied to the four-level converters. Considering (7) and the listed procedure in Appendix A, the modulation diagram of Fig. 4 was adapted and implemented for diode-clamped four-level converters. Meanwhile, the SVDS needs to be adapted suitably for 18 switches (six switches per phase), and the TD for nine triangles. The non-sinusoidal reference locus shown by Fig. 9*a* was programmed to be synthesised with a four-level converter. Experimental results of the developed four-level program are shown in Fig. 10. Figs. 10*a* and *b* present typical measured times concerned with the four modulation blocks of Fig. 4, whereas Fig. 10*c* illustrates the calculated times to process the TD by DSP for all 40 three-phase samples that vary within [1204, 1400] ns. Also, Fig. 10*d* gives the total modulation time with small variations below 5.4 μs. Although the number of triangles is multiplied by 2.25 compared with the three-level case, the processing times for the TD and the total modulation time are not proportionally increased.

4.1.4 Five- and six-level implementation: Furthermore, five- and six-level cases were developed using the proposed algorithm to obtain a better perspective on the processing time of the TD by the suggested method. Also, the non-sinusoidal reference locus of Fig. 9*a* was modulated for the SVM diagrams of Figs. 11*a* and *b*, where the locus of the reference passes through innermost triangles (taking the longest processing time for the TD) and outermost triangles (taking the shortest processing time for the TD). Thus, the calculated times include both minimum and maximum recorded times. Fig. 11*c* illustrates the calculated time by the DSP to process the five-level TD, where 16 triangles are available in each sextant. It can be seen that the processing times for positioning the reference voltages vary within [1400, 2200] ns. Similarly, Fig. 11*d* shows calculation times for a six-level TD among 25 triangles, which vary within [1470, 2500] ns. Note that these ranges of time variations occur because of the sequential detection approach managed by the typical TD procedures in Appendix A.

4.2 Overall assessment of proposed method

To make an overall assessment, processing times obtained by the experimental results are listed in Table 3. Experimental results show that although the proposed method is very fast, increasing the number of levels (or

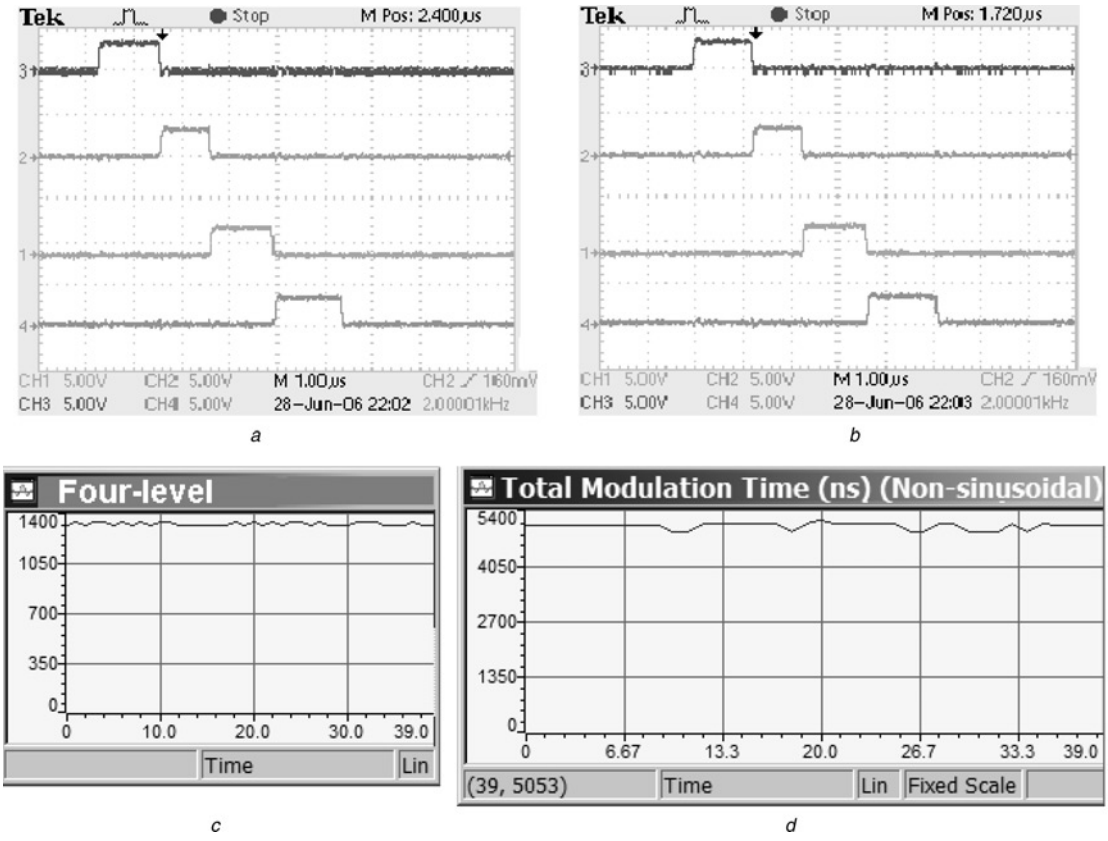

Fig. 10 *Typical timing of the four elements of the four-level SVM for a non-sinusoidal reference*

a and *b* Measured times
c Calculated time of the TD
d Calculated time of the entire modulation

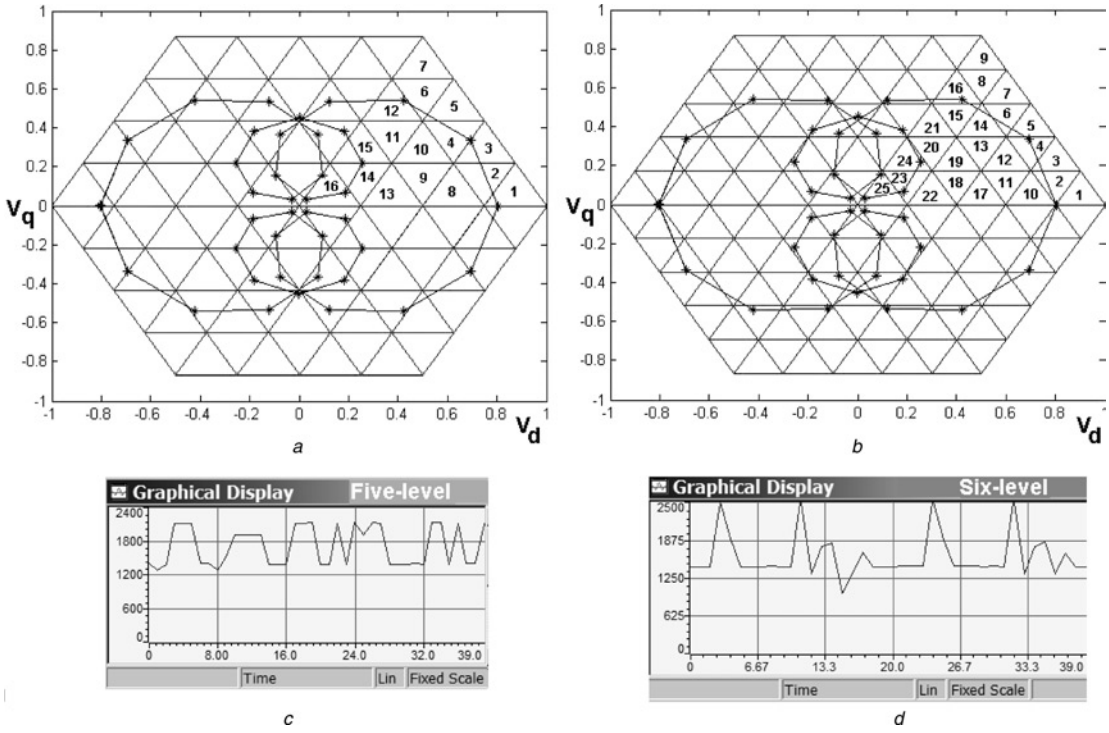

Fig. 11 *Calculated times of the TD for both five-level and six-level converters under non-sinusoidal references*

a and *b* Non-sinusoidal reference locus on the five-level and six-level SVM diagrams
c and *d* Typical processing time for the TD for five-level and six-level SVM, respectively

Table 3: Summary of experimental results for processing the TD using the foregoing methodology

Number of levels	Number of triangles per sextant	Processing time for the TD ns	Average processing time for the TD ns
3	4	560–758	659
4	9	1204–1400	1302
5	16	1220–2200	1710
6	25	1470–2500	1985

triangles in the SVM) adds very small average delays to the processing time. Additionally, at every instant, the higher the modulation index, the lower the processing time for the TD. Typically two sequential conditional procedures are shown in Appendix A, which can be rearranged appropriately conforming to our required applications.

5 Conclusion

A direct approach is proposed for positioning the reference in the *dq* plane. For an *n*-level converter, each sextant is divided into $(n-1)^2$ triangles. Every triangle has one inside region as well as six other outside regions. Determination of some rules will lead to the exact location of a point either inside or outside a triangle. This rule is applied to all triangles, which guides us to a complex characteristic vector. Using the characteristic vector allows to distinguish which of the triangles have negative real and imaginary parts as a way of positioning technique. Once a particular reference is placed within a triangle, three adjacent switching instants are selected and corresponding duty ratios are obtained. Different C programs, optimised for DSP implementation, are provided to confirm the validity of the proposed reference vector classification. The proposed positioning technique was uploaded by the developed C programs, and various experimental tests were performed for three-, four-, five-, and six-level converters. The results confirm that the positioning method is precise and quick, providing an effective way for higher-level SVM modulations. Additionally, the proposed C functions have the flexibility of being adapted for a required application in order to achieve less delayed response time.

6 References

1 Lai, J.S., and Peng, F.Z.: 'Multilevel inverters: a survey of topologies, controls, and applications', *IEEE Trans. Ind. Electron.*, 2002, **49**, pp. 724–738
2 Marchesoni, M., and Tenca, P.: 'Theoretical and practical limits in multilevel MPC inverters with passive front ends'. 9th European Power Electronics and Applications (EPE'01), Graz, Austria, CD Record, September 2001, pp. p.1–p.12
3 Soto, D., and Green, T.C.: 'A comparison of high power converter topologies for the implementation of FACTS controllers', *IEEE Trans. Ind. Electron.*, 2002, **49**, (5), pp. 1072–1080
4 Lai, J.S., and Peng, F.Z.: 'Multilevel converters–a new breed of power converters', *IEEE Trans. Ind. Appl.*, 1996, **32**, (3), pp. 509–517
5 Bakhshai, A., Espinoza, J., Joos, G., and Jin, H.: 'A combined artificial neutral network and DSP approach to the space vector modulation techniques', IEEE-IAS Annual Meeting, October 1996, pp. 934–940
6 Saeedifard, M., and Bakhshai, A.: 'Vector classification and voltage control in PWM three-level inverters'. IEEE Power Electronics Specialists Conf. (PESC'04), CD record, June 2004, pp. 4411–4417
7 Alonso, O., Marroyo, L., and Sanchis, P.: 'A generalized methodology to calculate switching times and regions in SVPWM modulation of multilevel converters'. 9th European Power Electronics and Applications (EPE'01), Graz, Austria, CD Record, September 2001, pp. 1–8
8 Pou, J., Pindado, R., Boroyevich, D., Rodriguez, P., and Vicente, J.: 'Voltage balancing strategies for diode-clamped multilevel converters'. IEEE Power Electronics Specialists Conf. (PESC'04), June 2004, vol. 1, pp. 3988–3993

9 Pou, J., Boroyevich, D., and Pindado, R.: 'New feedforward space vector PWM method to obtain balanced ac output voltages in a three-level NPC converters', *IEEE Trans. Ind. Electron.*, 2002, **49**, (5), pp. 1026–1034

7 Appendix A:Three- and four-level C code for the TD

Assume a three-level SVM diagram, where the reference vector is rotated back to sextant one by multiples of 60°. The following C code fragment provides the corresponding triangle number (triangle_no) in which the rotated reference vector resides

$$A = V_q - ((ep * V_d) >> 10);$$
$$B = V_q + ((ep * V_d) >> 10);$$
$$J_1 = (V_q - gp)\ \&\ 0x8000;$$
$$J_2 = (A + fp)\ \&\ 0x8000;$$
$$J_3 = (B - ep)\ \&\ 0x8000;$$
$$J_4 = (B - fp)\ \&\ 0x8000;$$
$$\text{if } (J_2\ \&\ J_3) \text{ triangle\_no} = 1;$$
$$\text{else if } ((J_1 \wedge J_2)\ \&\ (J_1 \wedge J_4)) \text{ triangle\_no} = 2;$$
$$\text{else if } (\sim J_1\ \&\ J_3) \text{ triangle\_no} = 3;$$
$$\text{else triangle\_no} = 4;$$

Where ep, fp and gp are the integer values of $\sqrt{3}$, $\sqrt{3}/2$, and $\sqrt{3}/4$ that are shifted left by ten bits. Moreover, for the four-level case, the end of the rotated reference vector can be positioned by the following C code fragment giving the corresponding triangle number

$$B = V_q + ((ep * V_d) >> 10);$$
$$A = V_q - ((ep * V_d) >> 10);$$
$$J1 = (B - ep)\ \&\ 0x8000;$$
$$J2 = (A + dp)\ \&\ 0x8000;$$
$$J3 = (V_q - mp)\ \&\ 0x8000;$$
$$J4 = (B - dp)\ \&\ 0x8000;$$
$$J5 = (A + lp)\ \&\ 0x8000;$$
$$J6 = (V_q - lp)\ \&\ 0x8000;$$
$$J7 = (B - lp)\ \&\ 0x8000;$$
$$\text{if } (J_1\ \&\ J_2) \text{ triangle\_no} = 1;$$
$$\text{else if } ((J_3 \wedge J_2)\ \&\ (J_3 \wedge J_4)) \text{ triangle\_no} = 2;$$
$$\text{else if } ((J_3 \wedge J_1)\ \&\ (J_3 \wedge J_5)) \text{ triangle\_no} = 3;$$
$$\text{else if } ((J_6 \wedge J_4)\ \&\ (J_6 \wedge J_5)) \text{ triangle\_no} = 4;$$
$$\text{else if } (\sim J_6\ \&\ J_1) \text{ triangle\_no} = 5;$$
$$\text{else if } (J_4\ \&\ J_5) \text{ triangle\_no} = 6;$$
$$\text{else if } ((J_3 \wedge J_7)\ \&\ (J_3 \wedge J_5)) \text{ triangle\_no} = 7;$$
$$\text{else if } (\sim J_3\ \&\ J_4) \text{ triangle\_no} = 8;$$
$$\text{else if } (J_1) \text{ triangle\_no} = 9;$$

where dp, lp and mp are the integer values of $2\sqrt{3}/3$, $\sqrt{3}/3$, and $\sqrt{3}/6$ that are shifted left by ten bits.