A COMBINED ARTIFICIAL NEURAL NETWORK AND DSP APPROACH TO THE IMPLEMENTATION OF SPACE VECTOR MODULATION TECHNIQUES

A. Bakhshai J. Espinoza G. Joós H. Jin *

Concordia University
Department of Elec. & Comp. Engineering
1455 de Maisonneuve Blvd. West
Montreal. P.Q. - Canada, H3G 1M8
Tel. (514) 848-3116, Fax. (514) 848-2802

Univ. of British Columbia *
Department of Electrical Engineering
2356 Main Hall
Vancouver, B.C. - Canada, V6T 1Z4
Tel. (604) 822-1695, Fax (604) 822-5949

Abstract — **Space Vector Modulation (SVM) in three-phase voltage source and current source converters has become the preferred PWM method for digital implementations. This paper presents an alternative SVM implementation that is based on a neural network structure. The technique reduces hardware and software complexity, and computation time, and increases the accuracy of the positioning of the switching instants. The technique exhibits the following features: (a) possibility of higher switching frequencies, (b) higher bandwidth of the control loops, (c) reduced hardware and software, and (d) reduction of parasitic harmonics in all PWM waveforms. The proposed method is compared to conventional implementations of SVM techniques in terms of hardware/software requirements, switching frequencies, harmonic spectra, and computation times. The method is applied to a 2 kVA unit and experimental results confirm theoretical and simulation results.**

I. INTRODUCTION

The rapid development of the switching frequency capabilities of power semiconductor devices requires faster, more accurate, and simpler modulation techniques [1]. Among the different PWM techniques available, the Space Vector Modulation (SVM) has become the preferred method for digital implementations in three-phase converters [1-3]. Although implementation of the SVM technique in digital systems is in principle simple, the required calculations and corresponding execution times limit the minimum sampling time. Thus, there is a maximum switching frequency attainable, and the maximum bandwidth of the overall control system is limited.

By exploiting the concept of vector classification used in the so called *instar competitive layer* of a counter propagation neural network (CPN competitive layer) [4-5], the hardware and software complexity of the implementation of the SVM can be considerably reduced. This paper proposes such an implementation. It requires less hardware and less computational effort. Therefore, the required sampling time can be reduced and thereby, higher switching frequencies can be achieved compared with conventional SVM implementations.

The proposed method is implemented on a TMS320C30 DSP microprocessor, and key experimental results are compared with those of conventional implementations of SVM techniques to verify theoretical results. Section II describes the SVM principles for voltage and current source converters, and Section III deals with the proposed implementation of the SVM.

II. THE SPACE VECTOR MODULATION

Any set of three-phase quantities that add up to zero in the *abc* frame ($[x]_{abc} = [x_a\ x_b\ x_c]^T$) can be represented in a two dimensional complex space [1]. The resulting vector in complex notation ($\mathbf{X} = [x]_{\alpha\beta}$) is given by:

$$\mathbf{X} = \begin{bmatrix} x_\alpha \\ x_\beta \end{bmatrix} = \begin{bmatrix} 1 & -1/2 & -1/2 \\ 0 & \sqrt{3}/2 & -\sqrt{3}/2 \end{bmatrix} \begin{bmatrix} x_a \\ x_b \\ x_c \end{bmatrix} = \mathbf{T}_{abc-\alpha\beta}[x]_{abc} \quad (1)$$

This transformation can be used with any of the three-phase quantities either voltages or currents associated with three-phase six-switch voltage and current source converters, Fig. 1.

The application of this transformation to the permitted switching states in a voltage source converter results in 6 non-zero phase voltage space vectors ($\mathbf{V}_k$, $k = 1,\ldots 6$) forming an hexagon centered at the origin of the $\alpha\beta$ plane, and the remaining two zero line voltage space vectors ($\mathbf{V}_7$, $\mathbf{V}_8$) are located at the origin of the plane, and they correspond to zero load line voltages. The eight space vectors and the corresponding hexagon are shown in Table I, and Fig. 2 respectively.

0-7803-3544-9/96 $5.00 © 1996 IEEE

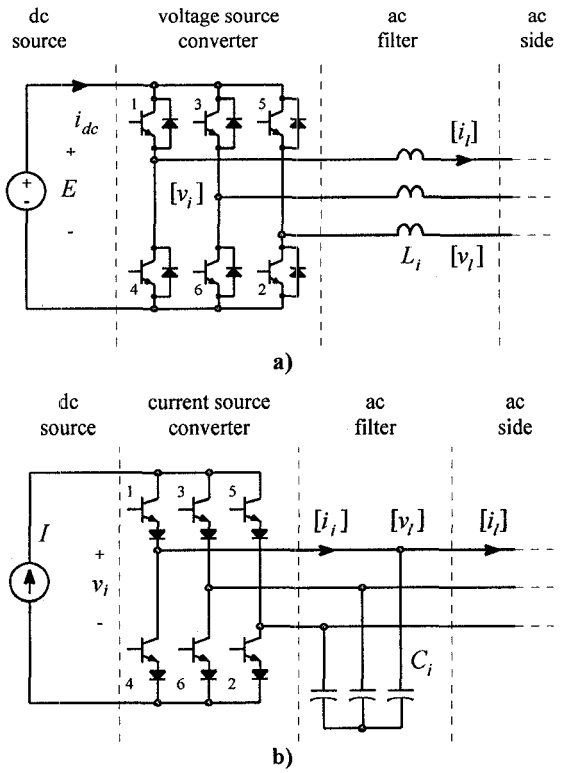

Fig. 1 Generalized six-switch three-phase converters. **a)** Voltage source. **b)** Current Source.

TABLE I
VOLTAGE SOURCE CONVERTER SWITCH COMBINATIONS AND ASSOCIATED SPACE VECTORS

State (k)	On Switches	v_{ia}	v_{ib}	v_{ic}	$\mathbf{V}_k$
1	1, 2, 6	E	0	0	E
2	1, 3, 2	E	E	0	$(1/2 + j\sqrt{3}/2)E$
3	2, 3, 4	0	E	0	$(-1/2 + j\sqrt{3}/2)E$
4	3, 4, 5	0	E	E	$-E$
5	4, 5, 6	0	0	E	$-(1/2 + j\sqrt{3}/2)E$
6	1, 5, 6	E	0	E	$(1/2 - j\sqrt{3}/2)E$
7	1, 3, 5	E	E	E	0
8	2, 4, 6	0	0	0	0

The same mapping applies to the 9 allowed switching states in a current source converter (Table II). This results in 6 non-zero line current space vectors ($\mathbf{I}_k$, k = 1,... 6) which create an hexagon centered at the origin of the αβ plane (Fig. 3). The three remained line current space vectors ($\mathbf{I}_7$, $\mathbf{I}_8$, and $\mathbf{I}_9$) are located at the origin of the αβ plane, and correspond to zero load currents.

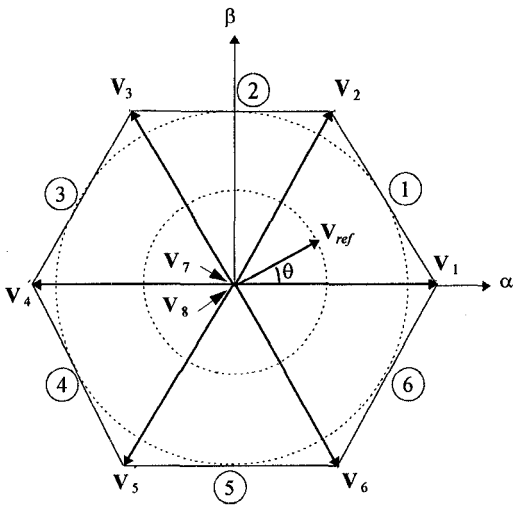

Fig. 2 Sectors and space vectors in the complex plane for VS converters.

TABLE II
CURRENT SOURCE CONVERTER SWITCH COMBINATIONS AND ASSOCIATED SPACE VECTORS

State (k)	On Switches	i_{ia}	i_{ib}	i_{ic}	$\mathbf{I}_k$
1	1, 2	I	0	$-I$	$\sqrt{3}(\sqrt{3}/2 + j1/2)I$
2	2, 3	0	I	$-I$	$j\sqrt{3}I$
3	3, 4	$-I$	I	0	$\sqrt{3}(-\sqrt{3}/2 + j1/2)I$
4	4, 5	$-I$	0	I	$-\sqrt{3}(\sqrt{3}/2 + j1/2)I$
5	5, 6	0	$-I$	I	$-j\sqrt{3}I$
6	6, 1	I	$-I$	0	$\sqrt{3}(\sqrt{3}/2 - j1/2)I$
7	1, 4	0	0	0	0
8	3, 6	0	0	0	0
9	5, 2	0	0	0	0

Depending upon the type of converter, either the instantaneous ac line current reference (in a CS converter), or the instantaneous ac phase voltage reference (in a VS converter), is represented by an equivalent space vector $\mathbf{I}_{ref}$ or $\mathbf{V}_{ref}$ respectively. The SVM strategy, therefore, defines the sequence of the space vectors (switching states) in such a way that their time average reproduces the reference vector within one cycle.

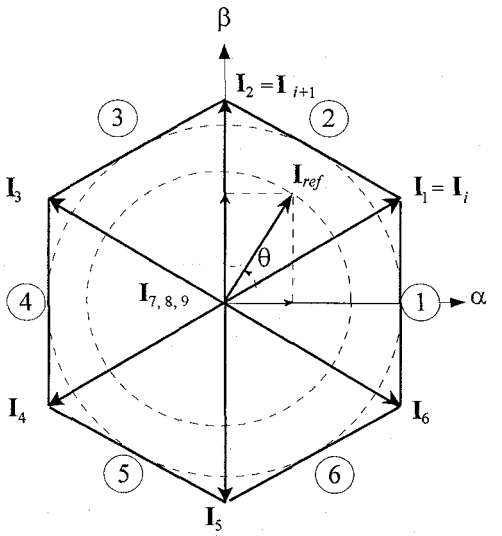

Fig. 3 Sectors and space vectors in the complex plane for CS converters.

The best tracking of the reference vector is obtained when the switching sequence includes only the two vectors adjacent to the reference and a zero vector. Also, the converter operates in its linear mode if the length of the reference vector does not exceed the radius of the inscribed circle of the hexagon. Thus in the linear mode of operation, the reference space vector is generated by the time average of the three switching state vectors located adjacent to it (including one null space vector). This yields:

$$\mathbf{U}_{ref}T = \mathbf{U}_i t_i + \mathbf{U}_{i+1}t_{i+1} + \mathbf{U}_0 t_0 \qquad (2\text{-a})$$

where T is the cycle period, $\mathbf{U}_{ref}$ is the reference vector (current or voltage), and t_i, t_{i+1}, and t_0 are the respective on-duration of the adjacent switching state vectors ($\mathbf{U}_i$, $\mathbf{U}_{i+1}$, and $\mathbf{U}_0$) that are defined as follows:

$$\begin{bmatrix} t_i \\ t_{i+1} \end{bmatrix} = m \begin{bmatrix} \sin(60-\theta) \\ \sin(\theta) \end{bmatrix} T \qquad (2\text{-b})$$

$$t_0 = T - t_i - t_{i+1} \qquad (2\text{-c})$$

where θ is the angle between the reference vector and the closest clockwise state vector (Fig. 2 and Fig. 3), m is the modulation index defined by $\| \mathbf{I}_{ref} \| / I$ for a current source converter, and $(2/\sqrt{3}) \| \mathbf{V}_{ref} \| / E$ for a voltage source converter.

The on-line implementation of the SVM requires that at every sample time the sector where the space vector reference lies be established, the modulation index (m) calculated, and (2-b) and (2-c) computed. Therefore, the trigonometric function *sin* in (2-b) must be on-line computed.

In conventional implementations, the *sin* function is stored in a look-up table and by means of interpolation, the value can be reasonably approximated. However, this approach presents three main disadvantages: a) the use of a look-up table implies the need for additional memory, b) interpolation of non-linear functions leads to poor accuracy in the calculations and thereby, contributes to the deterioration of the harmonic spectrum of the PWM waveforms [7], and c) the use of a look-up table and interpolation as a method of approximation, demands additional computing time that limits the maximum switching frequency of the converter. For instance, on a TMS320C30 DSP microprocessor, this algorithm is performed in around 41 μs, which is approximately 30% of the total control algorithm.

In the next sections, an approach that further simplifies the computing of non-linear functions (e.g. *sin*) based on vector classification is proposed. The concept of vector classification is widely used in the so called Counter Propagation Neural networks (CPN).

III. Proposed SVM Calculation Method

Because, the SVM is a deterministic problem and all classes are known in advance, there is no need to train a neural network. In addition, the signal processing technique employed in this paper is slightly different from the one used in CPN.

The instar hidden layer of the CPN comprises a set of processing elements known as instars. The instar is a single processing element that shares its general structure and processing functions with other processing elements. The training procedure of the traditional CPN competitive layer is described below.

Given a set of input vectors to be classified into a specific number of classes, the number of instars (neurons) in the hidden layer are selected to be equal to the number of the desired classes. The normalized input vectors are then applied to the input layer, and pre-randomly selected weights are adjusted to give the closest vector to a class of clusters. After the training process, when a normalized input vector appears at the input layer of the network, the net input of each instar is the inner product between the input vector and the associated weight vector. Thus, the net input after training represents one class of input vectors. In other words:

$$net = \mathbf{U} \cdot W = \|\mathbf{U}\| \|\mathbf{W}\| \cos(\theta) \qquad (3)$$

Since the input vector and the weight vector are normalized, the instars net input (3) gives the cosine of the angles between the input vector and the weight vectors that represent the classes. The largest instar net input wins the competition and the input vector is then classified in that class. This procedure after training is shown in Fig. 4.

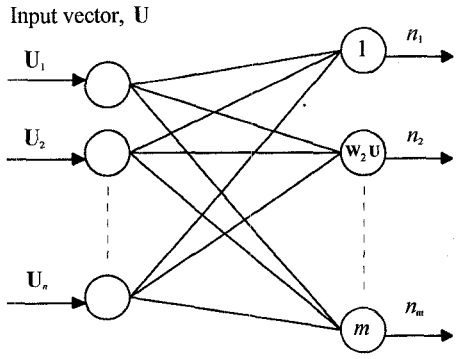

Fig. 4 A layer of instars arranged in competitive layer. Each unit receives the input vector $\mathbf{U} = [\mathbf{U}_1, \mathbf{U}_2, ..., \mathbf{U}_n]^T$ and the i^{th} unit has associated with it the weight vector, $\mathbf{W}_i = [w_{i1}, w_{i2}, ..., w_{in}]^T$. Net-input values are calculated in the net $n_i = \mathbf{U} \cdot \mathbf{W}_i$. The winner of the competition is the unit with the largest net input.

SVM is fundamentally a classification problem. The classes are known in advance, and there is no need to train the network. By its very nature, the SVM technique requires that the sector where the input vector lies be specified. To do this, the borders should be known, thus the competitive layer must have two winners. The on-time interval of each switch thereafter can be obtained by applying another simple signal processing to the winner net inputs. This leads to another modification consisting of simply setting the output of winners equal to their net inputs (in conventional CPN the winner is set to one, and the rest are set to zero). The whole procedure therefore can be explained as follows (Fig. 4).

A. Calculation Method for VS Converters

If $\mathbf{V}_{ref}$ is the space vector associated with the three-phase voltage reference, and $\mathbf{V}_k$ is a normalized VS converter state vector, the inner product of the $\mathbf{V}_{ref}$ and $\mathbf{V}_k$ given by (3) can be written as:

$$n_k = \|\mathbf{V}_{ref}\| \cos(\theta) \tag{4}$$

Since $\|\mathbf{V}_k\| = 1$, and $\mathbf{V}_{ref}$ is constant during one cycle, the largest n_k and the second largest n_k will uniquely specify the two space vectors adjacent to the reference vector, meaning that if the competitive layer has two winners, these two winners are the indexes of the two closest vectors to $\mathbf{V}_{ref}$ (Fig. 5).

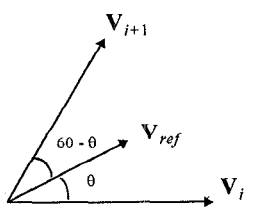

Fig. 5 Two neurons that win the competition are the indexes of the two closest state vectors to $\mathbf{V}_{ref}$.

Equation (3) can be written in a matrix form for all neurons.

$$\begin{bmatrix} n_1 \\ n_2 \\ n_3 \\ n_4 \\ n_5 \\ n_6 \end{bmatrix} = \begin{bmatrix} \mathbf{V}_1 \\ \mathbf{V}_2 \\ \mathbf{V}_3 \\ \mathbf{V}_4 \\ \mathbf{V}_5 \\ \mathbf{V}_6 \end{bmatrix} \mathbf{V}_{ref} \tag{5}$$

Replacing $\mathbf{V}_k$ from TABLE I, and $\mathbf{V}_{ref}$ from (1) will result in:

$$\begin{bmatrix} n_1 \\ n_2 \\ n_3 \\ n_4 \\ n_5 \\ n_6 \end{bmatrix} = \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & -1 \\ -\frac{1}{2} & 1 & -\frac{1}{2} \\ -1 & \frac{1}{2} & \frac{1}{2} \\ -\frac{1}{2} & -\frac{1}{2} & 1 \\ \frac{1}{2} & -1 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} v_{a,ref} \\ v_{b,ref} \\ v_{c,ref} \end{bmatrix} \tag{6}$$

Equation (6) is compared with (4), and thereby the weight matrix and input vectors associated with CPN competitive layer are obtained.

$$\mathbf{W} = \begin{bmatrix} 1 & \frac{1}{2} & -\frac{1}{2} & -1 & -\frac{1}{2} & \frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} & 1 & \frac{1}{2} & -\frac{1}{2} & -1 \\ -\frac{1}{2} & -1 & -\frac{1}{2} & \frac{1}{2} & 1 & \frac{1}{2} \end{bmatrix}^T, \quad \mathbf{U} = \begin{bmatrix} v_{a,ref} \\ v_{b,ref} \\ v_{c,ref} \end{bmatrix} \tag{7}$$

Equation (6) shows that the inner product between the reference $\mathbf{V}_{ref}$ and a space vector is a simple linear combination of the three voltage references. Since the last three rows of the weight matrix are simply the negative of the first three rows, less computational effort is required if efficient programming is used. Moreover because the coefficients in (6) can just take the values 1, -1, 1/2, and -1/2, the calculation could be implemented by hardware.

We now discuss how the time consuming calculations associated with (2) can be simply avoided.

Assume a specific $\mathbf{V}_{ref}$ is applied to the competitive layer, and n_i and n_{i+1} are the neurons who win the competition. According to (3) we have:

$$\begin{bmatrix} n_i \\ n_{i+1} \end{bmatrix} = \|\mathbf{V}_{ref}\| \begin{bmatrix} \cos(\theta) \\ \cos(60-\theta) \end{bmatrix} \tag{8}$$

but,

$$\begin{bmatrix} \cos(\theta) \\ \cos(60-\theta) \end{bmatrix} = \frac{2}{\sqrt{3}} \begin{bmatrix} 0.5 & 1 \\ 1 & 0.5 \end{bmatrix} \begin{bmatrix} \sin(60-\theta) \\ \sin(\theta) \end{bmatrix} \tag{9}$$

After substituting (9) in (8), and re-writing (8), we get:

$$\begin{bmatrix} n_i \\ n_{i+1} \end{bmatrix} = \frac{2}{\sqrt{3}} \frac{\|\mathbf{V}_{ref}\|}{E} T \begin{bmatrix} 0.5 & 1 \\ 1 & 0.5 \end{bmatrix} \begin{bmatrix} \sin(60-\theta) \\ \sin(\theta) \end{bmatrix} \frac{E}{T} \tag{10}$$

Equation (10) can further be rearranged to get:

$$\frac{2}{3}\frac{T}{E} \begin{bmatrix} -1 & 2 \\ 2 & -1 \end{bmatrix} \begin{bmatrix} n_i \\ n_{i+1} \end{bmatrix} = \frac{2}{\sqrt{3}} \frac{\|\mathbf{V}_{ref}\|}{E} \begin{bmatrix} \sin(60-\theta) \\ \sin(\theta) \end{bmatrix} T \tag{11}$$

The right side of (11) is simply the on-duration of the consecutive adjacent switching state vectors $\mathbf{V}_i$, and $\mathbf{V}_{i+1}$ meaning that:

$$\begin{bmatrix} t_i \\ t_{i+1} \end{bmatrix} = \frac{2}{3}\frac{T}{E} \begin{bmatrix} -1 & 2 \\ 2 & -1 \end{bmatrix} \begin{bmatrix} n_i \\ n_{i+1} \end{bmatrix} \tag{12}$$

The whole procedure can be summarized in the form of a block diagram as shown in Fig. 6. To implement this method the following steps are to be taken :

a) according to (6), n_k for $k = 1,..., 6$ are calculated.
b) the two largest n_i and n_{i+1} and their corresponding indexes (i and i+1) are selected by the *proposed competitive network*.
c) the on-duration (t_i and t_{i+1}) of the two adjacent space vectors selected in b), are obtained by performing a simple signal processing on n_i and n_{i+1}.
d) the SV (switching combination, $\mathbf{V}_i$ and $\mathbf{V}_{i+1}$) are selected by the *decoder* block according to the values i and i+1.

When adjacent vectors and on-times are determined, the procedure for defining the sequence for implementing the chosen combination is identical to that used in conventional SVMs.

B. *Calculation Method for CS Converters*

A similar approach is used to obtain the expressions for the on-time intervals of a current source converter. In this case the inner product of (3) in the matrix form has to be changed to:

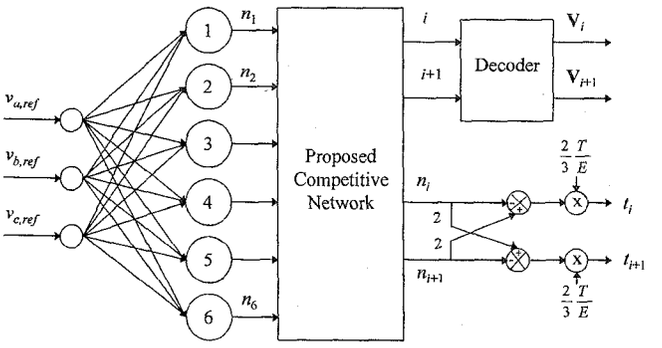

Fig. 6 Proposed space vector implementation for VS converters.

$$\begin{bmatrix} n_1 \\ n_2 \\ n_3 \\ n_4 \\ n_5 \\ n_6 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & -1 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \\ 0 & -1 & 1 \\ 1 & -1 & 0 \end{bmatrix} \begin{bmatrix} i_{a,ref} \\ i_{b,ref} \\ i_{c,ref} \end{bmatrix} \tag{13}$$

Equation (13) is compared with (3), and thereby the weight matrix and input vectors associated with CPN competitive layer are obtained.

$$\mathbf{W} = \frac{1}{2} \begin{bmatrix} 1 & 0 & -1 & -1 & 0 & 1 \\ 0 & 1 & 1 & 0 & -1 & -1 \\ -1 & -1 & 0 & 1 & 1 & 0 \end{bmatrix}^T, \quad \mathbf{U} = \begin{bmatrix} i_{a,ref} \\ i_{b,ref} \\ i_{c,ref} \end{bmatrix} \tag{14}$$

Again, if we assume that n_i, and n_{i+1} win the competition for a specific $\mathbf{I}_{ref}$, then we have:

$$\begin{bmatrix} n_i \\ n_{i+1} \end{bmatrix} = \sqrt{3} \|\mathbf{I}_{ref}\| \begin{bmatrix} \cos(\theta) \\ \cos(60-\theta) \end{bmatrix} \tag{15}$$

Combining (9) with (15), will result:

$$\begin{bmatrix} n_i \\ n_{i+1} \end{bmatrix} = \sqrt{3} \|\mathbf{I}_{ref}\| \frac{T}{I} \frac{2}{\sqrt{3}} \begin{bmatrix} 1/2 & 1 \\ 1 & 1/2 \end{bmatrix} \begin{bmatrix} \sin(60-\theta) \\ \sin(\theta) \end{bmatrix} \frac{I}{T} \tag{16}$$

Equation (16) can further be rearranged to yield:

$$\frac{1}{3} \begin{bmatrix} 1 & -2 \\ -2 & 1 \end{bmatrix} \frac{T}{I} \begin{bmatrix} n_i \\ n_{i+1} \end{bmatrix} = \frac{\|\mathbf{I}_{ref}\|}{I} \begin{bmatrix} \sin(60-\theta) \\ \sin(\theta) \end{bmatrix} T \tag{17}$$

The right side of (17) is the on-duration of the consecutive adjacent switching state vectors $\mathbf{I}_i$ and $\mathbf{I}_{i+1}$ meaning that:

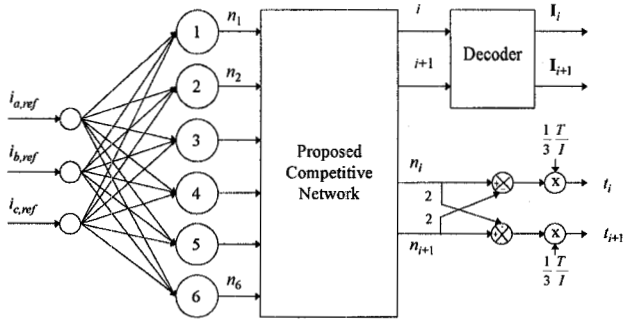

Fig. 7 Proposed space vector implementation for CS converter.

$$\begin{bmatrix} t_i \\ t_{i+1} \end{bmatrix} = \frac{1}{3}\frac{T}{I}\begin{bmatrix} 1 & -2 \\ -2 & 1 \end{bmatrix}\begin{bmatrix} n_i \\ n_{i+1} \end{bmatrix} \tag{18}$$

Equations (13), and (18) show that the inner product between the reference $\mathbf{I}_{ref}$ and a CSC space vector $\mathbf{I}_k$ correspond to the instantaneous difference of two line current references. This operation can be implemented on a DSP microprocessor with a minimum computational effort. The whole procedure can be summarized in the form of a block diagram as shown in Fig. 7, with the following steps:

a) according to (6), n_k for $k = 1,..., 6$ are calculated.
b) the two largest n_i and n_{i+1} and their corresponding indexes (i and i+1) are selected by the *proposed competitive network.*
c) the on-duration (t_i and t_{i+1}) of the two adjacent space vectors selected in b), are obtained by performing a simple signal processing on n_i and n_{i+1}.
d) the SV (switching combination, $\mathbf{I}_i$ and $\mathbf{I}_{i+1}$) are selected by the *decoder* block according to the values i and i+1.

Like in the VS approach, when adjacent vectors and on-times are determined, the procedure for defining the sequence for implementing the chosen combination is identical to that used in conventional SVMs. In this implementation the chosen sequence is $\mathbf{I}_i$, $\mathbf{I}_{i+1}$, and $\mathbf{I}_0$. The resulting switching frequency is one-half the sampling frequency.

IV. Implementation and Experimental Results

The SVM technique using a conventional and the proposed approach were implemented on a 2 kVA current source rectifier laboratory set-up. The prototype uses a TMS320C30 DSP microprocessor. The conventional approach was implemented using look-up tables and then interpolation to perform the non-linear functions required by the modulator (Fig. 8).

The comparative simplicity of the proposed calculation method is shown in Fig. 8. In fact, only basic mathematical operations such as +, -, x, and / are required. This is an important consideration in practical implementations where

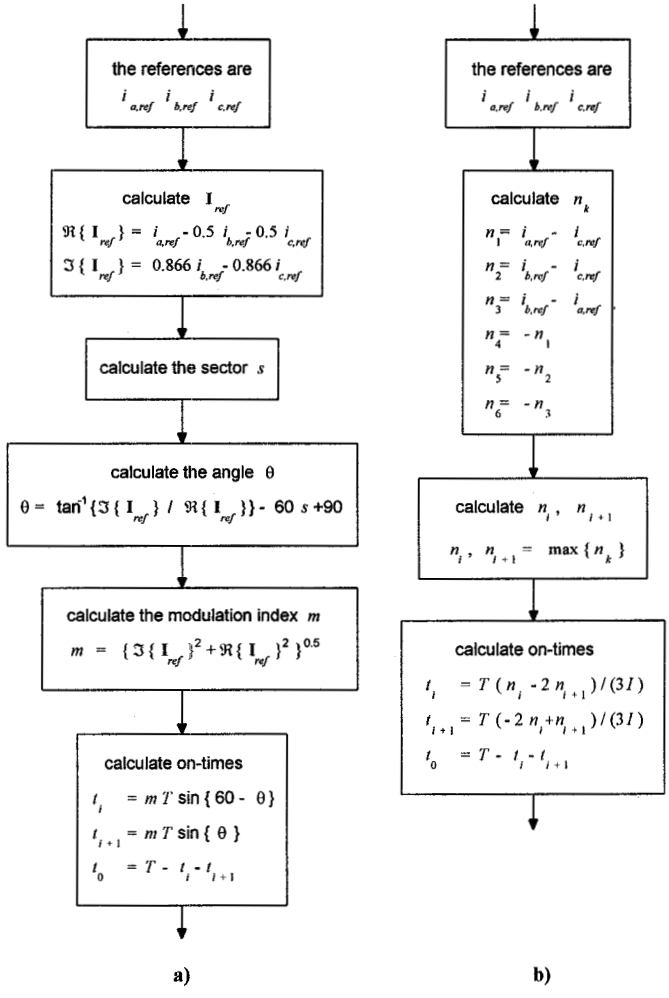

Fig. 8 Conventional and proposed SVM calculation flow diagrams. **a)** Conventional approach. **b)** Proposed approach.

high speed is always required. Experimental results show that the conventional scheme takes 40.7 μs, whereas the proposed one takes only 27.3 μs. Therefore, a reduction of 33% in the execution time is achieved. This allowed the reduction of the total execution time of the control algorithm from 150 μs to 136.6 μs, thereby, allowing the sample frequency to be increased from 6400 Hz (108 pu) to 7200 Hz. (120 pu). Thus, a higher band width for the control loop can be attained.

Furthermore, no approximation is required in order to calculate the on-times of the selected space vectors. Therefore, it can be concluded that accuracy is improved over conventional implementations. The higher accuracy is experimentally proved by plotting the Total Harmonic Distortion as a function of the modulation index for both methods and under the same cycle frequency. Fig. 9 depicts the experimental values. It can be seen that for the whole range of modulation indexes the proposed method exhibits lower distortion.

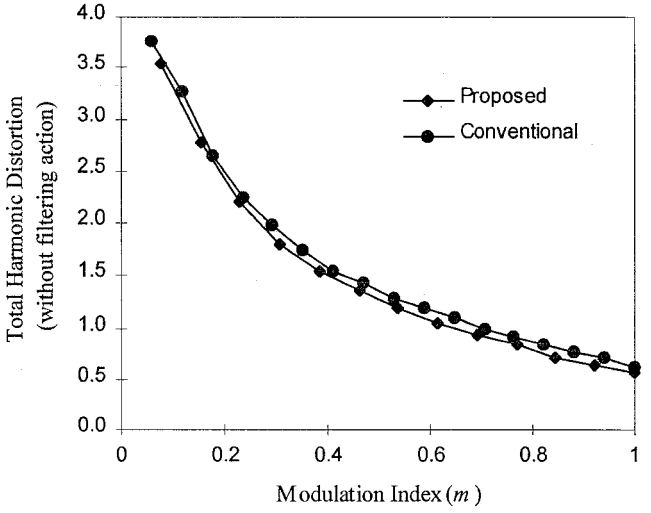

Fig. 9 Total harmonic distortion of the line current or voltage as a function of the modulation index

Experimental results validating the approach are presented in Fig. 10. The gating pattern, PWM line current (i_{ia}), and PWM line current spectrum for a 2 kVA space vector modulated current source rectifier operating in open loop are illustrated.

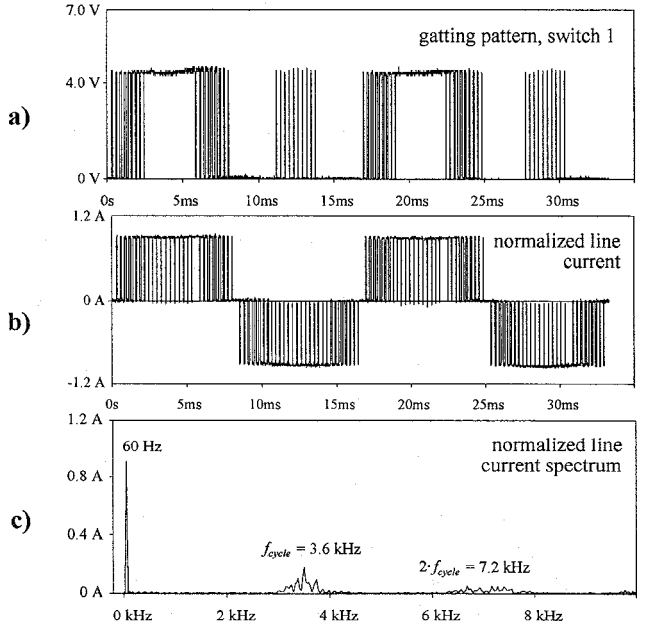

Fig. 10 CSC experimental waveforms using the proposed computation method, (f_{cycle} = 3.6 kHz, m = 0.8). **a)** Gating pattern for switch 1. **b)** Normalized ac line current. **c)** Normalized ac line current spectrum.

V. Conclusions

This paper presents an alternative SVM implementation that is based on a neural network structure and its classification properties. The technique reduces hardware and software complexity, computation time, and increases the accuracy of the positioning of the switching instants. The following characteristics are therefore improved: (a) possibility of higher switching frequencies, (b) higher bandwidth for the control loops, (c) reduced hardware and software, and (d) reduction of parasitic harmonics in PWM waveforms. Selected experimental results show that a reduction of 33% of the execution time compared with the conventional method is achieved.

Acknowledgment

Financial support provided by the Natural Sciences and Engineering Research Council of Canada and the Ministry of Education (Québec) through an FCAR grant is greatly appreciated. Mr. Alireza Bakhshai would like to thank the Ministry of Culture and Higher Education of Iran for the fellowship and Isfahan University of Technology for the leave of absence to pursue the Ph. D. degree at Concordia University.

References

[1] J. Holtz, "Pulsewidth modulation - A survey," *IEEE Trans. Ind. Electron.*, vol. 39, no. 5, pp. 410-420, October 1992.

[2] V. Vlatkovic. and D. Borojevic, "Digital-signal-processor-based control of three-phase space vector modulated converters," *IEEE Trans. Ind. Electron.*, vol. 41, no. 3, pp. 326-332, June 1994.

[3] M. Kazmierkowski, M. Dzieniakowski, and W. Sulkowski, "Novel space vector based current controllers for PWM inverters," *IEEE Trans. Power Electron.*, vol. 6, no. 1, pp. 158-166, June 1991.

[4] J. Freeman and D. Skapura, *Neural networks algorithms, applications, and programming techniques*, Addison-Wesley, 1992.

[5] R. Hecht-Nielsen, *Neurocomputing*, Addison-Wesley, Reading, MA, 1990.

[6] J. Holtz, W. Lotzkat, and A. Khambadkone, "On continuous control of PWM inverters in the overmodulation range including the six-step mode," *IEEE Trans. Power Electron.*, vol. 8, no. 4, pp. 546-553, October 1993.

[7] S. Vukosavic and M. Stojic, "Reduction of parasitic spectral components of digital space vector modulation by real-time numerical methods," *IEEE Trans. Power* Fig. *Electron.*, vol. 10, no. 1, pp. 94-102, January 1995.