Journal of Theoretical and Applied Information Technology
10th December 2013. Vol. 58 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A NEURAL NETWORK-BASED SVPWM CONTROLLER FOR A TWO-LEVEL VOLTAGE-FED INVERTER INDUCTION MOTOR DRIVE

[1]AHMED A. HASSAN, [2]SAMIR DEGHEDIE, [3]MOHAMED EL HABROUK
[1, 2, 3]Department of Electrical Engineering, Faculty of Engineering, Alexandria University
E-mail [1]ahmed_hassan6@hotmail.com, [2]deghedie@gmail.com, [3]eepgmme1@yahoo.co.uk

ABSTRACT

In this paper, a detailed description of a neural network-based implementation of the SVPWM algorithm for two-level voltage source inverters is proposed using a modular approach which facilitates the expansion of the scheme to higher levels SVM, each step in the SVPWM algorithm is achieved using a simple feed-forward artificial neural network that consists of one or more layers. Simulation results are provided by employing the proposed scheme inside a closed loop V/Hz drive system.

Keywords: *Space Vector Modulation (SVM), Artificial Neural Networks (ANNs), Two-Level Inverter*

1. INTRODUCTION

The main objective of inverters is to produce an AC output waveform from a DC power supply where the amplitude, phase, and frequency of the voltage should always be controllable which is a requirement for many applications such as adjustable speed drives (ASDs), uninterruptible power supplies (UPS), active filters, etc [1]

Modulation techniques are responsible for controlling the amount of time and the sequence used to switch the power switches on and off. The modulation techniques most used are the sinusoidal pulse-width modulation (SPWM), space-vector modulation (SVM) and the selective harmonic elimination pulse-width modulation (SHEPWM) [1].

The square-wave or six-step operation mode of the inverter is simple to implement and characterized by low switching losses because there are only six switching per cycle of the fundamental frequency. Unfortunately, the lower order harmonics will cause large distortions of the current wave unless filtered by a bulky low-pass filter [2].

2. FUNDAMENTALS OF SVM

Space-vector pulse-width modulation has recently grown as a very popular PWM method for voltage-fed converter AC drives [3]. SVM is based on the representation of the three phase quantities as vectors in a two-dimensional α-β plane [4].

2.1 Inverter Topology and Space Vectors

The two-Level three-phase (VSI) is shown in figure 1. Valid switch states for the two-level three-phase VSI are shown in table 1, States 1 - 6 produce non-zero ac output voltages while states 0 & 7 produce zero ac line voltages. In this case, the ac line currents freewheel through either the upper or lower components [2].

The selection of the states in order to generate the given waveform is done by the modulating technique that should ensure the use of only the valid states [1]. A “0” denotes that the lower switch is conducting while a “1” denotes that the upper switch is conducting in the respective leg

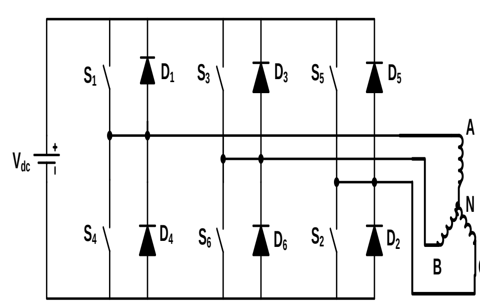

Figure 1: Two Level Inverter Topology

.

Journal of Theoretical and Applied Information Technology
10th December 2013. Vol. 58 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 1: Valid Switch States for Two-Level Inverter

State	"ON" Switches	Space Vector
0	S2, S4, S6	$V_0(000)$
1	S1, S2, S6	$V_1(100)$
2	S1, S2, S3	$V_2(110)$
3	S2, S3, S4	$V_3(010)$
4	S3, S4, S5	$V_4(011)$
5	S4, S5, S6	$V_5(001)$
6	S1, S5, S6	$V_6(101)$
7	S1, S3, S5	$V_7(111)$

2.2 Rotating Space Vector and SVM Theory [2]

If the three-phase sinusoidal and balanced voltages with peak value V_m are given by:

$$v_a = V_m \cos(\theta) \qquad (1)$$

$$v_b = V_m \cos\left(\theta - \frac{2\pi}{3}\right) \qquad (2)$$

$$v_c = V_m \cos\left(\theta + \frac{2\pi}{3}\right) \qquad (3)$$

The space vector in complex notation is given by:

$$\overline{V} = \frac{2}{3}\left[v_a + v_b e^{j2\pi/3} + v_c e^{-j2\pi/3}\right] \qquad (4)$$

Substituting equations (1), (2) and (3) into equation (4) and simplifying, we get:

$$\overline{V} = V_m e^{j\theta} = V_m e^{j\omega t} \qquad (5)$$

This indicates that the vector $\overline{\mathbf{V}}$ rotates in circular orbit with angular velocity $\boldsymbol{\omega}$, where the direction of rotation depends on the phase sequence of the voltages . The vector magnitude is given by:

$$|\overline{V}| = V_m = \sqrt{v_\alpha^2 + v_\beta^2} \qquad (6)$$

At any instant, the vector angle $\boldsymbol{\theta}$ w.r.t the α-axis is given by

$$\theta = \tan^{-1}\frac{v_\beta}{v_\alpha} \qquad (7)$$

If the reference vector $\overline{V}^*$ lies in sector 1 as shown in figure 2, the PWM output is generated by resolving $\overline{V}^*$ using the adjacent vectors $\overline{V}_1$& $\overline{V}_2$ on a part-time basis to satisfy the average output demand.

$$V_a = \frac{2}{\sqrt{3}} V^* \sin\left(\frac{\pi}{3} - \theta\right) \qquad (8)$$

$$V_b = \frac{2}{\sqrt{3}} V^* \sin\theta \qquad (9)$$

Where V_a and $V_b$are the components of $\overline{V}^*$aligned in the directions of $\overline{V}_1$ & $\overline{V}_2$ respectively.

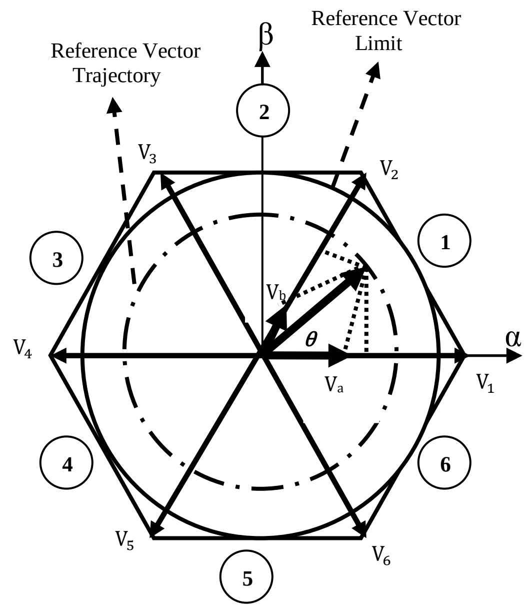

Figure 2: Reference Vector Trajectory and Space Vectors

3. ANN-BASED TWO-LEVEL SVM

Figure 3 shows the general block diagram for generating the gate pulses for two-level SVM. Each block is responsible for achieving certain task which is described in the following sections.

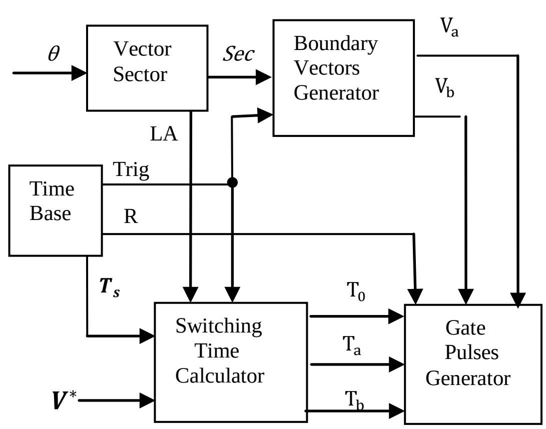

Figure 3: Block Diagram of Two-Level SVM Generator

3.1 Time Base

The purpose of the block diagram shown in figure 4 is to generate different timing signals as follows:

The sampling frequency is integrated to produce a ramp function, when the ramp reaches its peak

Journal of Theoretical and Applied Information Technology
10th December 2013. Vol. 58 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

value, the "Trig" signal is generated and the integrator is reset to indicate the start of a new sampling period. The sampling period "Ts" is generated from the sampling frequency.

Where, $$Ts = \frac{1}{F_s} \quad (10)$$

The ramp "***r***" directly generated from integrating "Fs" is multiplied by the sampling period "Ts" to generate the ramp function "R" with amplitude equal to "Ts" (figure 5).

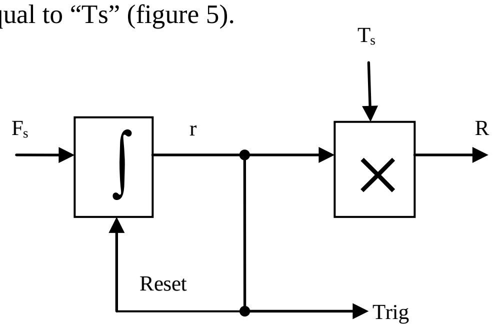

Figure 4: Time Base Generation

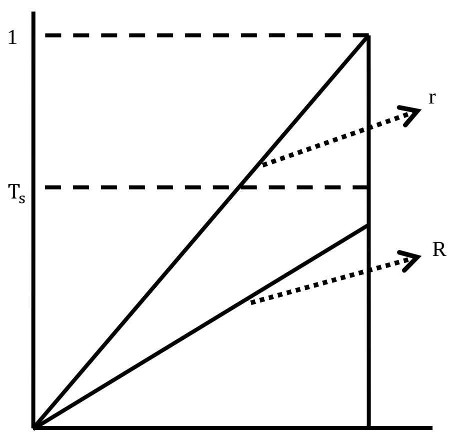

Figure 5: Ramp Generation

3.2 Vector Sector

The purpose of this block is to determine the location of the reference vector at the start of the sampling period. Figure 6 represents a simplified block diagram for implementing this function.

Where,

Θ Reference vector angle, measured from α- axis
Sec Sector number in which the reference vector Lies at the time of sampling
LA Reference vector Local Angle (measured from The beginning of the sector)

The purpose of generating a local angle is to reflect the equations used to calculate the switching times in sector 1 for all the other sectors without the need to develop separate equations for each sector, figure 7 outlines this concept.

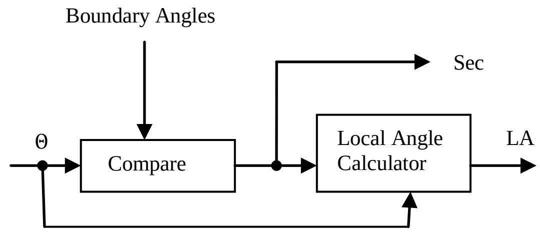

Figure 6 Block Diagram for Determining "Sec" & "LA"

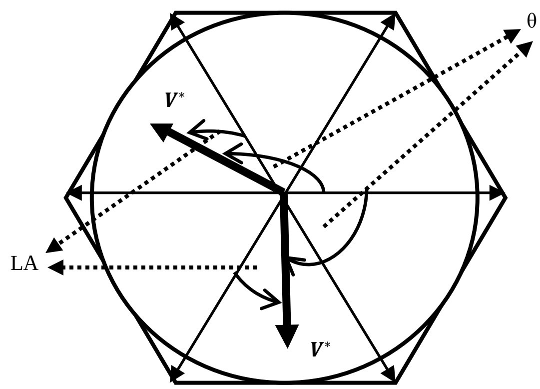

Figure 7 Reference Vector Local Angle (LA)

Table 2 shows the relation between "LA" and "θ" in different sectors

Table 2: Relation Between "θ", "LA" and "Sec"

θ	Sec	LA
$0^o < \theta < 60^o$	1	LA = θ
$60^o < \theta < 120^o$	2	$LA = \theta - 60^o$
$120^o < \theta < 180^o$	3	$LA = \theta - 120^o$
$-180^o < \theta < -120^o$	4	$LA = \theta + 180^o$
$-120^o < \theta < -60^o$	5	$LA = \theta + 120^o$
$-60^o < \theta < 0^o$	6	$LA = \theta + 60^o$

Figure 8 shows the ANN used to generate "Sec" & "LA".

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

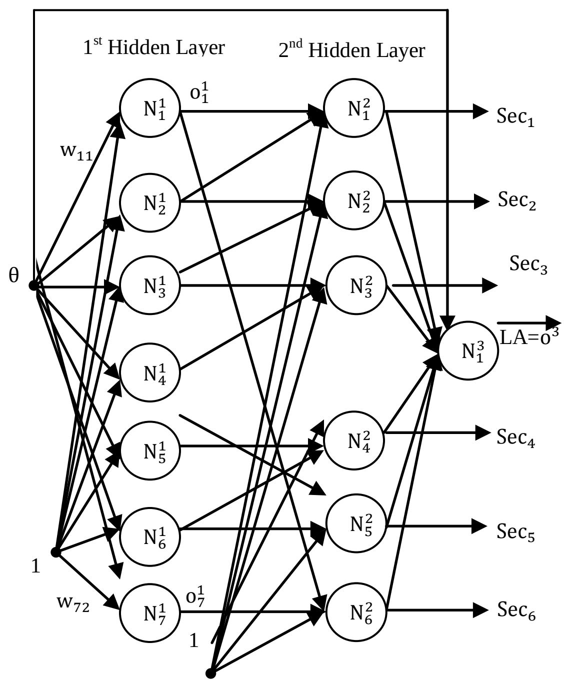

Figure 8: "Vector Sector" ANN used to determine Reference Vector Sector "Sec" *and Local Angle Inside Each Sector* "LA"

3.2.1 1st Hidden layer:

The purpose of this layer is to indicate (at the sampling instant) that the reference vector angle "θ" is greater than which of the angles that represent the beginning of each sector.

Each neuron in the 1st layer will fire when the reference vector angle "θ" is greater than or equal to the starting angle of the respective sector.

Table 3 shows the firing conditions for neurons of the 1st layer, where, "N_x" is the neuron number and "O_x" is the output of the respective neuron, the superscript denotes the layer number (this convention is followed in all neural networks described throughout this paper).

The weight, input and output matrices are defined as follows:

$$W^1 = \begin{bmatrix} 1 & 0 \\ 1 & -1.0472 \\ 1 & -2.0944 \\ 1 & -3.1416 \\ 1 & 3.1416 \\ 1 & 2.0944 \\ 1 & 1.0472 \end{bmatrix} = \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \\ w_{31} & w_{32} \\ w_{41} & w_{42} \\ w_{51} & w_{52} \\ w_{61} & w_{62} \\ w_{71} & w_{72} \end{bmatrix}$$

$$P^1 = \begin{bmatrix} \theta \\ 1 \end{bmatrix}$$

$$O^1 = \begin{bmatrix} \text{hardlim}(\theta + 0) \\ \text{hardlim}(\theta - 1.0472) \\ \text{hardlim}(\theta - 2.0944) \\ \text{hardlim}(\theta - 3.1416) \\ \text{hardlim}(\theta + 3.1416) \\ \text{hardlim}(\theta + 2.0944) \\ \text{hardlim}(\theta + 1.0472) \end{bmatrix} = \begin{bmatrix} o_1^1 \\ o_2^1 \\ o_3^1 \\ o_4^1 \\ o_5^1 \\ o_6^1 \\ o_7^1 \end{bmatrix}$$

Where w_{ij} denotes the weight connecting the i^{th} neuron to the j^{th} input

Table 3: Firing Conditions for Neurons of the 1st Layer for the "Vector Sector" ANN

N_x^1	$O_x^1 = 1$ if
N_1^1	$\theta \geq 0$
N_2^1	$\theta \geq 1.0472$
N_3^1	$\theta \geq 2.0944$
N_4^1	$\theta \geq 3.1416$
N_5^1	$\theta \geq -3.1416$
N_6^1	$\theta \geq -2.0944$
N_7^1	$\theta \geq -1.0472$

3.2.2 2nd Hidden layer:

This layer indicates that the reference vector angle "θ" is less than which of the angles that represent the end of each sector and to use this information along with the information presented from the 1st layer to determine the boundary angles of the sector in which the reference vector lies. The weight, input and output matrices are defined as follows:

$$W^2 = \begin{bmatrix} 1 & -1 & -1 \\ 1 & -1 & -1 \\ 1 & -1 & -1 \\ 1 & -1 & -1 \\ 1 & -1 & -1 \\ 1 & -1 & -1 \end{bmatrix}$$

$$P^2 = \begin{bmatrix} O_1^1 & O_2^1 & O_3^1 & O_5^1 & O_6^1 & O_7^1 \\ O_2^1 & O_3^1 & O_4^1 & O_6^1 & O_7^1 & O_1^1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$O^2 = \begin{bmatrix} \text{hardlim}(O_1^1 - O_2^1 - 1) \\ \text{hardlim}(O_2^1 - O_3^1 - 1) \\ \text{hardlim}(O_3^1 - O_4^1 - 1) \\ \text{hardlim}(O_5^1 - O_6^1 - 1) \\ \text{hardlim}(O_6^1 - O_7^1 - 1) \\ \text{hardlim}(O_7^1 - O_1^1 - 1) \end{bmatrix} = \begin{bmatrix} sec_1 \\ sed_2 \\ sec_3 \\ sec_4 \\ sec_5 \\ sec_6 \end{bmatrix} = \text{Sec}$$

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Which means that neuron 1 will fire and outputs a "1" if $O_1^1 - O_2^1 \geq 1$ which is possible if and only if $O_1^1 = 1$ AND $O_2^1 = 0$

Therefore, neuron 1 will fire and output a "1" if and only if angle "θ" is greater than 0^o but not greater Than 60^o, i.e., the reference vector lies in sector 1

The firing conditions for the other neurons can be obtained in a similar manner

The results are summarized in table .4.

Table 4 Firing Conditions for Neurons Of the 2nd Layer For the "Vector Sector" ANN

N_x^2	$O_x^2 = 1$ if
N_1^2	$O_1^1 = 1$ AND $O_2^1 = 0$
N_2^2	$O_2^1 = 1$ AND $O_3^1 = 0$
N_3^2	$O_3^1 = 1$ AND $O_4^1 = 0$
N_4^2	$O_5^1 = 1$ AND $O_6^1 = 0$
N_5^2	$O_6^1 = 1$ AND $O_7^1 = 0$
N_6^2	$O_7^1 = 1$ AND $O_1^1 = 0$

3.2.3 Output layer

This layer determines the local angle "LA"

The weight, input and output matrices are defined as follows:

$$W^3 = [1 \;\; 0 \;\; -1.0472 \;\; -2.0944 \;\; 3.1416 \;\; 2.0944 \;\; 1.0472]$$

$$P^3 = [\theta \;\; O_1^2 \;\; O_2^2 \;\; O_3^2 \;\; O_4^2 \;\; O_5^2 \;\; O_6^2]^t$$

$$O^3 = [\mathrm{purelin}(\theta - 1.0472 O_2^2 - 2.0944 O_3^2 + 3.1416 O_4^2 + 2.0944 O_5^2 + 1.0472 O_6^2)]$$

only one neuron from the 2nd layer will present a "1" at its output while all other five neurons will present a "0", indicating the presence of the reference vector in one of six sectors, therefore, all the terms in the expression for O^3 will cancel except the first term ($\boldsymbol{\theta}$) and only one of the other six terms, consequently, the expression for O^3 will always reduce to:

$$O^3 = LA = \theta + C$$

Where, ***C*** is a constant whose possible values are chosen to be multiples of 1.0472 rad (60^o) that needs to be subtracted from or added to "θ" to produce **"**LA" (refer to table 2 & figure 7)

All results are summarized in table 5

*Table 5 Results of Output Layer for the "Vector Sector" ANN, Indicating the Expression used to Calculate **LA** in Different Sectors*

O_1^2	O_2^2	O_3^2	O_4^2	O_5^2	O_6^2	Sec	O^3 (LA)
1	0	0	0	0	0	1	θ
0	1	0	0	0	0	2	$\theta - 1.0472$
0	0	1	0	0	0	3	$\theta - 2.0944$
0	0	0	1	0	0	4	$\theta + 3.1416$
0	0	0	0	1	0	5	$\theta + 2.0944$
0	0	0	0	0	1	6	$\theta + 1.0472$

3.3 Switching Time Calculator

This block computes the three time intervals during which the reference vector is sampled to two boundary vectors ($V_1 - V_6$) and one zero-vector (V_0, V_7). The "Trig" signal triggers the block to start computing the switching times according to the following equations:

$$T_a = \sqrt{3}\,\frac{V^*}{V_{dc}} T_s \sin\left(\frac{\pi}{3} - LA\right) \quad (11)$$

$$T_b = \sqrt{3}\,\frac{V^*}{V_{dc}} T_s \sin(LA) \quad (12)$$

$$T_0 = T_s - T_a - T_b \quad (13)$$

Where,

T_a Interval of first boundary vector (V_a)

T_b Interval of second boundary vector (V_b)

To Interval of zero-vector (V_0, V_7)

The non-linear part of equations (11) & (12) which is the "sin" function can be computed using an ANN as shown in figure 9 instead of using look-up tables which requires large memory space.

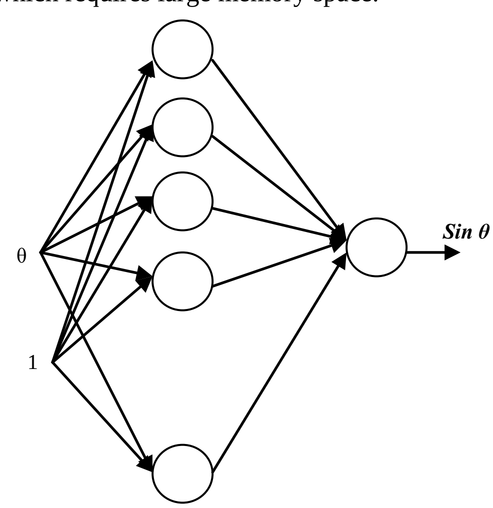

Figure 9: ANN Sine Wave Generator

Journal of Theoretical and Applied Information Technology
10th December 2013. Vol. 58 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The network is trained using MATLAB Toolbox Neural Fitting Tool (nftool)

3.4 Boundary Vectors Generator

The "Trig" signal initiates the operation of the block to evaluate the two non-zero components of the reference vector "V_a" and "V_b" at the start of the sampling period, the input to this block is "Sec" (sector no.) and the outputs are

V_a Component of the reference vector aligned With the start of sector

V_b Component of the reference vector aligned With the end of sector

Figure 10 shows the neural network used to produce the boundary vectors.
The weight, input and output matrices are defined as follows:

$$W = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 1 & -0.5 \\ 0 & 1 & 1 & 1 & 0 & 0 & -0.5 \\ 0 & 0 & 0 & 1 & 1 & 1 & -0.5 \\ 1 & 0 & 0 & 0 & 1 & 1 & -0.5 \\ 1 & 1 & 1 & 0 & 0 & 0 & -0.5 \\ 0 & 0 & 1 & 1 & 1 & 0 & -0.5 \end{bmatrix}$$

$$P = \begin{bmatrix} \text{Sec} \\ 1 \end{bmatrix} = \begin{bmatrix} \text{sec}_1 \\ \text{sed}_2 \\ \text{sec}_3 \\ \text{sec}_4 \\ \text{sec}_5 \\ \text{sec}_6 \\ 1 \end{bmatrix}$$

$$O = \text{hardlim}[WP]$$

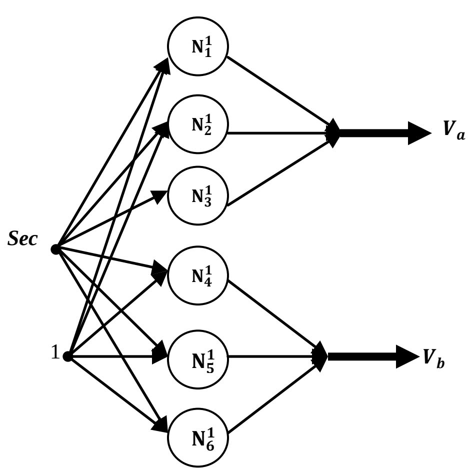

Figure 10: ANN for producing the Boundary Vectors

The elements of the input matrix, denotes the active sector (this is exactly the output matrix of the 2nd hidden layer from the "Vector Sector" ANN)
. The weight matrix elements are chosen such that the space vectors elements are aligned vertically in the first six columns (each column contains the two vectors that are the boundaries of the six sectors), i.e., the boundary vectors of sector 1 (100 & 110) are aligned in column1while the boundary vectors of sector 2 (110 & 010) are aligned in column 2 and so on.(refer to table 1 and figure 2 for boundary vectors in each sector).

The weight of the bias is chosen so that, the elements of the column which corresponds to the active sector will be available at the output. For example, if the reference vector lies in sector 5, then:

$$P = [0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0]^t$$

$$O = \begin{bmatrix} \text{hardlim}(0-0.5) \\ \text{hardlim}(0-0.5) \\ \text{hardlim}(1+0.5) \\ \text{hardlim}(1+0.5) \\ \text{hardlim}(0-0.5) \\ \text{hardlim}(1+0.5) \end{bmatrix} = \begin{bmatrix} \text{hardlim}(-0.5) \\ \text{hardlim}(-0.5) \\ \text{hardlim}(1.5) \\ \text{hardlim}(1.5) \\ \text{hardlim}(-0.5) \\ \text{hardlim}(1.5) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \end{bmatrix}$$

i.e., vectors 001 & 101 will be available at the output, which are the boundary vectors for sector 5.

3.5 Gate Pulses Generator

This block is divided into two sub-blocks

3.5.1 Vector selector

Generates "VS" (Vector Select) signal which indicates the vector that should be present at the output in the correct instant by comparing the ramp signal "R" with different time instants to produce the "symmetric sequence" modulation technique as shown in figure 11, the neural network which implements this function is shown in figure 12

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

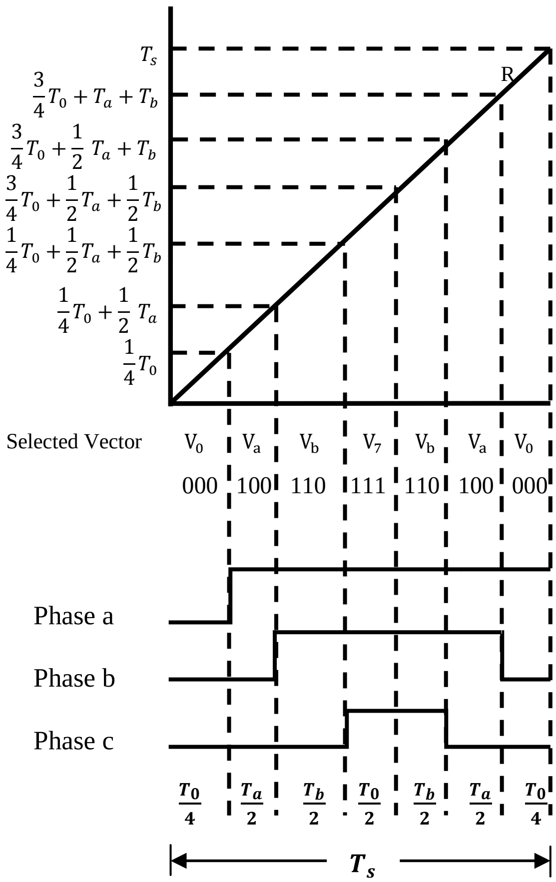

Figure 11: Generation of VS Command during Complete Sampling Period in Sector 1

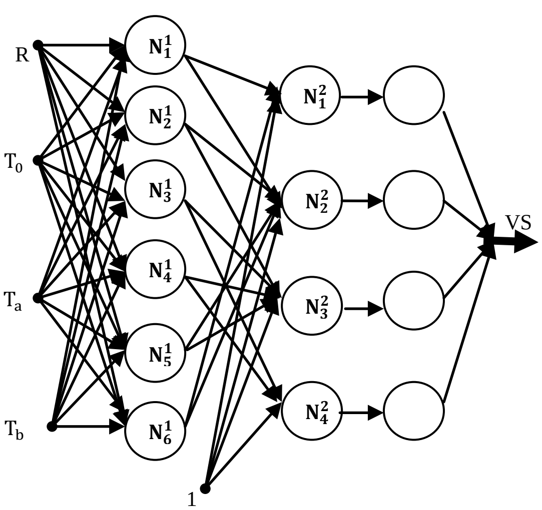

*Figure 12: ANN for Producing the **VS** Command*

3.5.1.1 1st hidden layer:

This layer triggers the change of the output vector Each neuron in the first layer (N_x^1) will fire when "R" increases above the horizontal lines, which represents points in time at which the output vector should be changed.

The weight, input and output matrices for the 1st hidden layer are defined as follows:

$$W^1 = \begin{bmatrix} 1 & -0.25 & 0 & 0 \\ 1 & -0.25 & -0.5 & 0 \\ 1 & -0.25 & -0.5 & -0.5 \\ 1 & -0.75 & -0.5 & -0.5 \\ 1 & -0.75 & -0.5 & -1 \\ 1 & -0.75 & -1 & -1 \end{bmatrix}$$

$$P^1 = \begin{bmatrix} R \\ T_0 \\ T_a \\ T_b \end{bmatrix}$$

$$O^1 = \begin{bmatrix} \text{hardlim}(R - 0.25T_0) \\ \text{hardlim}(R - 0.25T_0 - 0.5T_a) \\ \text{hardlim}(R - 0.25T_0 - 0.5T_a - 0.5T_b) \\ \text{hardlim}(R - 0.75T_0 - 0.5T_a - 0.5T_b) \\ \text{hardlim}(R - 0.75T_0 - 0.5T_a - T_b) \\ \text{hardlim}(R - 0.75T_0 - T_a - T_b) \end{bmatrix}$$

For example, N_1^1, will fire when $R \geq 0.25T_0$, N_2^1, will fire when $R \geq 0.25T_0 + 0.5T_a$ and so on. All results are summarized in table 6.

Table 6: Firing Conditions for Neurons of the 1st Hidden Layer for the "Vector Selector" ANN

N_x^1	$O_x^1 = 1$ if
N_1^1	$R \geq 0.25T_0$
N_2^1	$R \geq 0.25T_0 + 0.5T_a$
N_3^1	$R \geq 0.25T_0 + 0.5T_a + 0.5T_b$
N_4^1	$R \geq 0.75T_0 + 0.5T_a + 0.5T_b$
N_5^1	$R \geq 0.75T_0 + 0.5T_a + T_b$
N_6^1	$R \geq 0.75T_0 + T_a + T_b$

3.5.1.2 2nd hidden layer

This layer produces a code that represents the correct "VS" command.

The weight, input and output matrices for the 2nd hidden layer are defined as follows:

$$W^2 = \begin{bmatrix} -2 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & -1 & 0 & 0 & 1 & -1 & -1 \\ 0 & 1 & -1 & 1 & -1 & 0 & -1 \\ 0 & 0 & 1 & -1 & 0 & 0 & -1 \end{bmatrix}$$

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

$$P^2 = \begin{bmatrix} O_1^1 \\ O_2^1 \\ O_3^1 \\ O_4^1 \\ O_5^1 \\ O_6^1 \\ 1 \end{bmatrix}$$

$$O^2 = \begin{bmatrix} \text{hardlim}(-2O_1^1 + O_6^1 + 1) \\ \text{hardlim}(O_1^1 - O_2^1 + O_5^1 - O_6^1 - 1) \\ \text{hardlim}(O_2^1 - O_3^1 + O_4^1 - O_5^1 - 1) \\ \text{hardlim}(O_3^1 - O_4^1 - 1) \end{bmatrix}$$

N_1^2, will fire (indicating that V_0 should be present at the output) when,

$$-2O_1^1 + O_6^1 + 1 \geq 0$$

This is possible when

- $O_1^1 = 0$ AND $O_6^1 = 0$, i.e., $R < 0.25T_0$ AND $R < 0.75T_0 + T_a + T_b$ (interval 1 in figure 11) OR
- $O_1^1 = 1$ AND $O_6^1 = 1$, i.e., $R > 0.25T_0$ AND $R > 0.75T_0 + T_a + T_b$ (interval 7 in figure 11)

N_1^2, will also fire if $O_1^1 = 0$ AND $O_6^1 = 1$ but this case is not applicable because this means that $R < 0.25T_0$ AND $R > 0.75T_0 + T_a + T_b$ which is not possible.

Possible input combinations and the corresponding outputs for N_1^2 are summarized in table 7.

Table 7: Input Combinations of N_1^2 for the "Vector Selector" ANN

O_1^1	O_6^1	O_1^2	Interval
0	0	1	1
0	1	1	N/A
1	0	0	N/A
1	1	1	7

Similarly, N_2^2 will fire (indicating that V_a should be present at the output) when,

$$O_1^1 - O_2^1 + O_5^1 - O_6^1 - 1 \geq 0$$

N_3^2, will fire (indicating that V_b should be present at the output) when,

$$O_2^1 - O_3^1 + O_4^1 - O_5^1 - 1 \geq 0$$

N_4^2, will fire (indicating that V_7 should be present at the output) when,

$$O_3^1 - O_4^1 - 1 \geq 0$$

3.5.1.3 Output layer:

This layer implements a bit-wise NOT function on the output of the 2nd hidden layer to produce the "VS" vector

Where, $VS = [VS_1 \quad VS_2 \quad VS_3 \quad VS_4]^t$

The output of the "Vector Selector" ANN for Two-Level SVM is summarized in table 8

Table 8: Output of the "Vector Selector" ANN

VS				Interval	Selected Vector
O_1^3	O_2^3	O_3^3	O_4^3		
0	1	1	1	1, 7	V_0
1	0	1	1	2, 6	V_a
1	1	0	1	3, 5	V_b
1	1	1	0	4	V_7

3.5.2 Vector decoder

Ensures that the appropriate vector is present at the output in the correct instant so that the inverter switches are switched ON or OFF according to the selected modulation scheme.

The output vector V_{out} can assume one of four values (V_a, V_b, V_0 or V_7) at different instants in the sampling period T_s

The neural network used to implement this function is shown in figure 13.

3.5.2.1 1st Hidden layer:

The purpose of this layer is to select one of the 4 vectors (V_0, V_a, V_b and V_7) and pass it to the next layer. The selected vector depends on the value of the "VS" vector.

This layer contains 12 neurons divided into 4 groups of 3 neurons each (3 neurons for each one of the 4 input vectors), each neuron in this layer represents a ANN switch port, where the 3 neurons of each group is enabled/disabled by a 0/1from the respective bit in the "VS" vector, as shown in table 8

3.5.2.2 2nd Hidden layer

This layer contains 3 neurons, the input to each neuron is similar elements in the 4 vectors so that each neuron passes one element of the selected vector to the output layer. The output of this layer represents the states of the 3 upper inverter switches.

The weight, input and output matrices for this layer are defined as follows:

Journal of Theoretical and Applied Information Technology
10th December 2013. Vol. 58 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

$$P^2 = \begin{bmatrix} O_1^1 \\ O_2^1 \\ O_3^1 \\ O_4^1 \\ O_5^1 \\ O_6^1 \\ O_7^1 \\ O_8^1 \\ O_9^1 \\ O_{10}^1 \\ O_{11}^1 \\ O_{12}^1 \end{bmatrix} = \begin{bmatrix} V_{01} \\ V_{02} \\ V_{03} \\ V_{a1} \\ V_{a2} \\ V_{a3} \\ V_{b1} \\ V_{b2} \\ V_{b3} \\ V_{71} \\ V_{72} \\ V_{73} \end{bmatrix}$$

$$W^2 = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

$$O^2 = \begin{bmatrix} \mathrm{purelin}(V_{01} + V_{a1} + V_{b1} + V_{71}) \\ \mathrm{purelin}(V_{02} + V_{a2} + V_{b2} + V_{72}) \\ \mathrm{purelin}(V_{03} + V_{a3} + V_{b3} + V_{73}) \end{bmatrix}$$

3.5.2.3 Output layer:

This layer implements the NOT function on the output from the 2nd hidden layer to produce the states of the lower 3 inverter switches.

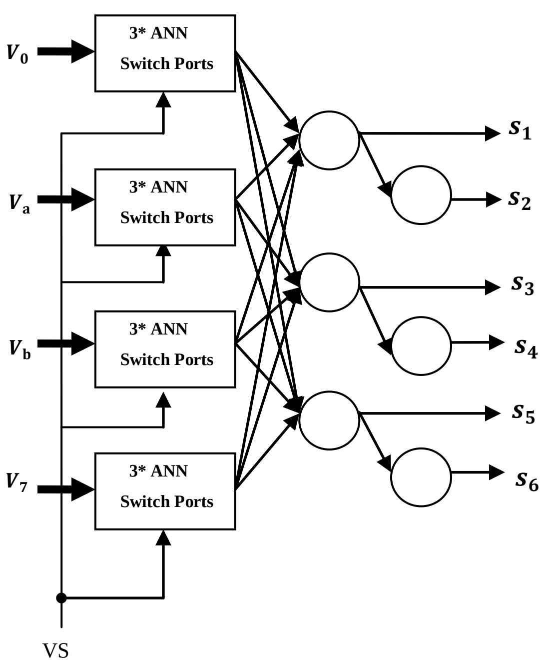

Figure 13: "Vector Decoder" ANN

4. SIMULATION RESULTS

The two-level ANN SVM generator is employed inside a scalar control drive system to validate its operation. Figure 14 shows the block diagram of the closed loop V/Hz scalar control used in the simulation.

The angular velocity, is compared with the reference speed, ω_M^* The speed error signal, $\Delta\omega_M$, is applied to a slip controller, usually of the PI type, which generates the reference slip speed, ω_{SL}^*. The slip speed must be limited for stability and over-current prevention. The reference synchronous speed ω_{Syn}^* is obtained by adding ω_{SL}^* to ω_M .The latter signal used to generate the reference values, $V^*$and ω^* [5].

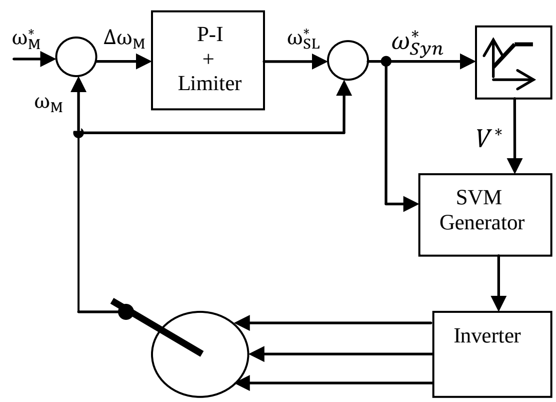

Figure 14: Closed Loop V/Hz Scalar Control

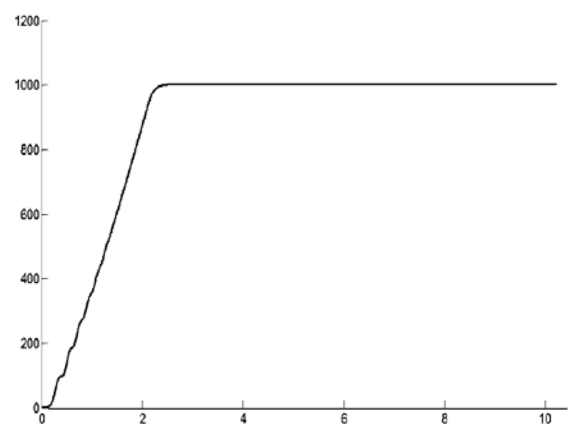

Figure 15: Rotor Speed ω_M

Journal of Theoretical and Applied Information Technology
10th December 2013. Vol. 58 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

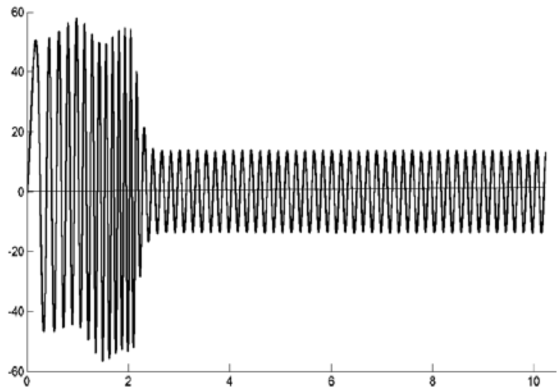

Figure 16: Phase a Stator Current i_a

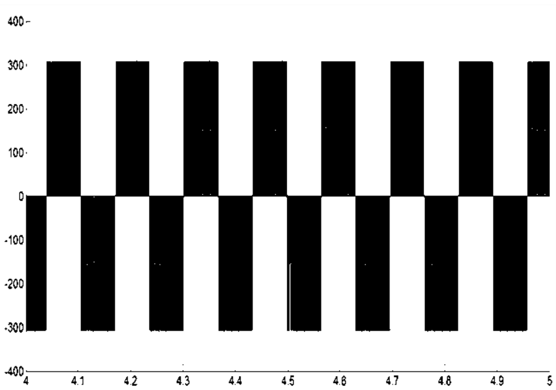

Figure 17: Line Voltage v_{ab}

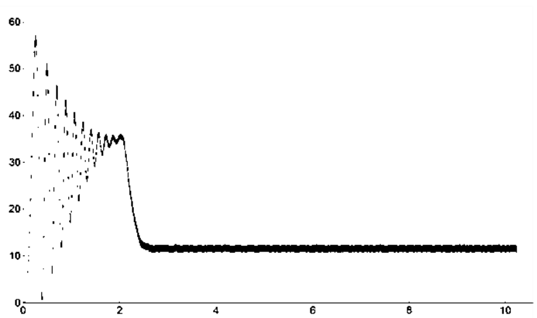

Figure 18: Electromagnetic Torque T_M

5. CONCLUSION

Space vector pulse width modulation (SVPWM) can be implemented using a set of interconnected artificial neural networks, where each network achieves a certain step in the SVPWM algorithm. The proposed SVPWM scheme was employed inside a closed loop v/Hz scalar control system and simulated using MATLAB Simulink to verify its operation.

REFERENCES:

[1] Muhammad H. Rashid, "Power Electronics Hand Book", Academic Press, Academic Press Series in Engineering, First Edition, 2001, ISBN-0125816502

[2] Bimal K. Bose, "Modern Power electronic and AC Drives", Prentice Hall, First Edition, 2001, ISBN-0130167436

[3] Subrata K. Mondal, João O. P. Pinto and Bimal K. Bose, "A Neural-Network-Based Space-Vector PWM Controller for a Three-Level Voltage-Fed Inverter Induction Motor Drive", IEEE transactions on Industry Applications, VOL. 38, No. 3, May/June 2002

[4] Prasad, V. H., Master of Science In Electrical Engineering, "Analysis and Comparison of Space Vector Modulation schemes for Three-Leg and Four-Leg Voltage Source inverters", Virginia Polytechnic Institute and State University, 1997

[5] Andrzej M. Trzynadlowski, "Control of Induction Motors", Academic Press, First Edition, 2000, ISBN-0127015108