Performance Evaluation of a Neural Network based General Purpose Space Vector Modulator

A.Muthuramalingam, S.Himavathi

***Abstract*—**Space Vector Modulation (SVM) is an optimum Pulse Width Modulation (PWM) technique for an inverter used in a variable frequency drive applications. It is computationally rigorous and hence limits the inverter switching frequency. Increase in switching frequency can be achieved using Neural Network (NN) based SVM, implemented on application specific chips. This paper proposes a neural network based SVM technique for a Voltage Source Inverter (VSI). The network proposed is independent of switching frequency. Different architectures are investigated keeping the total number of neurons constant. The performance of the inverter is compared for various switching frequencies for different architectures of NN based SVM. From the results obtained, the network with minimum resource and appropriate word length is identified. The bit precision required for this application is identified. The network with 8-bit precision is implemented in the IC XCV 400 and the results are presented. The performance of NN based general purpose SVM with higher bit precision is discussed.

***Keywords*—**NN based SVM, FPGA Implementation, Layer Multiplexing, NN structure and Resource Reduction, Performance Evaluation

I. Introduction

SWITCHING schemes play the most important role in voltage source inverters, which are widely used in several applications, such as motor drives, active filters and uninterruptible power supplies. As a result, number of modulation strategies have been developed and are reported [1]. Space vector modulation technique has been increasingly used in the last decade, because it not only delivers an optimal output but also reduces harmonic content of the output voltage/current [2-5]. However a disadvantage of SVM is that it requires complex online computation that limits the inverter switching frequency. Power semiconductor switching speed has been improved dramatically in recent years. They demand switching frequency as high as 50 kHz. In order to use such high frequency switching power semiconductor devices effectively, the operating frequency of the SVM has to be increased.

Marginal increase in frequency is achieved using high speed Digital Signal Processors (DSP) in combination with a lookup Table [3, 4]. A neural network based SVM can further increase the switching frequency, particularly when implemented on an application specific integrated circuit chip. Parallelism, modularity and dynamic adaptation are three computational characteristics typically associated with ANNs. FPGA-based reconfigurable computing architectures are well suited to implement ANNs [6].

FPGA realization of ANNs with a large number of neurons is still a challenging task because ANN algorithms are “multiplication-rich” and it is relatively expensive to implement multipliers on FPGAs. A body of research exists to show that it is possible to train ANNs with integer weights. The interest in using integer weights stems from the fact that integer multipliers can be implemented more effectively than floating-point ones. There are also special learning algorithms, which use powers-of-two integers as weights [7]. The advantage of powers of two-integer weight learning algorithms is that the required multiplications in an ANN can be reduced to a series of shift operations. A few attempts have been made to implement ANNs in FPGA hardware with floating-point weights. Despite continuing advances in FPGA technology, it is still impractical to implement ANNs on FPGAs with floating-point precision weights [6]. Very recently, the concept of layer multiplexing for implementing ANNs on FPGA is reported [8], where the authors have implemented different ANN architectures with floating point precision weights in the IC XCV400hq-240. The results are very promising to implement a NN with minimum resources.

A multi layer feed forward network based SVM proposed in [3] is shown to perform well. To realize a ANN based SVM with higher switching frequency and desired inverter performance, both design and implementation aspects are studied in this paper. They broadly include the choice of architecture for reduced resource in FPGA, Study of bit precision on inverter performance and Evaluation of inverter performance with NN based SVM for various switching frequency. Theoretically it has been shown that increase in the number of layers improves the performance of the network [9]. Increase in layers also reduces the resource requirement in an FPGA implementation [8]. In this paper, the effect of the increase in the number of layers with the total number of neurons being constant is investigated. The performance of the inverter is compared for the different

Manuscript received in April 2007. The research is supported in part by the grants from the All India Council for Technical Education (AICTE), a statutory body of Government of India. File Number: ***8020/RID/TAPTEC-32/2001-02.***

A.Muthuramalingam, working as an Assistant Professor in the Department of Electrical and Electronics Engg, Pondicherry Engineering College. Puducherry-605014, India. (e-mail: amrlingam@hotmail.com)

S.Himavathi, working as an Assistant Professor in the Department of Electrical and Electronics Engg, Pondicherry Engineering College. Puducherry-605014, India. (e-mail: hima_pondy@yahoo.co.in)

architectures of NN based SVM. The results obtained are discussed. Of the proposed different architectures, the one, which is suitable for FPGA implementation, is identified and implemented in the IC XCV400hq240. The performance of the NN with different bit precisions is investigated.

II. Space Vector Modulation of Voltage Source Inverter

The three phase two level inverter with an active load is shown in Fig. 1. Its switching operation is characterized by eight switch states $\overline{S_i}$ = (SW$_a$, SW$_b$, SW$_c$), i= 0, 1, . . ,7. where SW$_a$ represents the switching status of inverter Leg-A. It is “1”, when switch Q$_1$ is ON & Q$_4$ is OFF and ZERO, when switch Q$_1$ is OFF & Q$_4$ is ON. Similarly SW$_b$ & SW$_c$ is defined for inverter Leg-B and Leg-C. The output voltages of the inverter are controlled by these eight switching states.

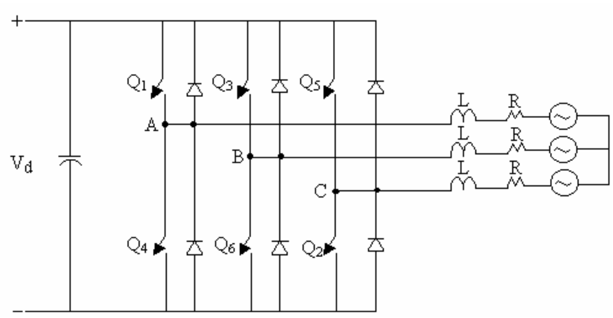


Fig. 1 Three-phase two-level voltage Source Inverter

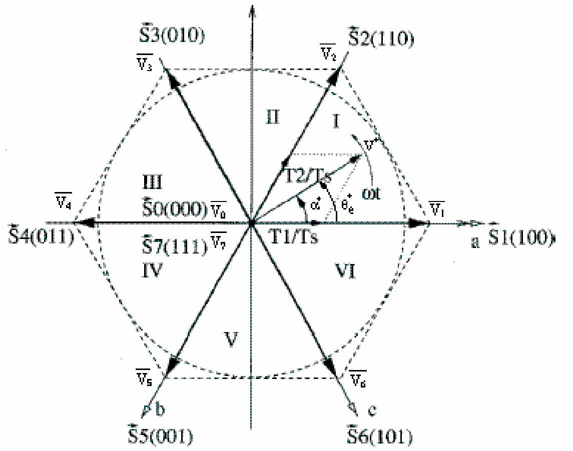


Fig. 2 Voltage vector space

Let the inverter voltage vectors $\overline{V}_0(000)$, ..., $\overline{V}_7(111)$ correspond to the eight switching states. These vectors form the voltage vector space as shown in the Fig.2. The three phase reference voltages can be represented by a space vector $\overline{V}$ with the magnitude V^* and phase angle θ^*_e [10]. In a sampling/switching interval, the output voltage vector $\overline{V}$ is expressed as

$$\overline{V} = \frac{t_0}{T_s}\overline{V_0} + \frac{t_1}{T_s}\overline{V_1} + \cdots + \frac{t_7}{T_s}\overline{V_7}$$

$$t_a = 2.K.V^* \sin\left(\pi/3 - \alpha^*\right)$$

$$t_b = 2.K.V^* \sin\alpha^*$$

$$t_0 = \left(T_s/2\right) - \left(t_a + t_b\right)$$

$$T_{A-ON} = \left(T_S/4\right) + V^* \cdot T_s \cdot g_a\left(\alpha^*\right)$$

$$T_{A\text{-}OFF} = T_S\text{-}T_{A\_ON}$$

$$g_a(\alpha^*) = \begin{cases} \frac{\sqrt{3}}{4\cdot V_d}\left[-\sin(\pi/3-\alpha^*) - \sin\alpha^*\right], S=1,6 \\ \frac{\sqrt{3}}{4\cdot V_d}\left[-\sin(\pi/3-\alpha^*) + \sin\alpha^*\right], S=2 \\ \frac{\sqrt{3}}{4\cdot V_d}\left[\sin(\pi/3-\alpha^*) + \sin\alpha^*\right], \quad S=3,4 \\ \frac{\sqrt{3}}{4\cdot V_d}\left[\sin(\pi/3-\alpha^*) - \sin\alpha^*\right], \quad S=5 \end{cases}$$

where $t_0, t_1, ... t_7$ are the turn on time of the vectors $V_0, V_1, ... V_7$ respectively and T$_S$ is the sampling/switching time period. From the above equation the vector $\overline{V}$ can be decomposed into $V_0, V_1, ... V_7$ in infinite number of ways. However, in order to reduce the number of switching actions and make full use of active turn on time, the vector is commonly split into two nearest adjacent vectors and zero vectors in an arbitrary sector.

The equations of the effective time of the inverter switching states is given as t$_a$, t$_b$ and t$_c$.

where

V*	Magnitude of command or reference voltage vector
t$_a$	Time period of switching vector that lags V*
t$_b$	Time period of switching vector that leads V*
t$_0$	Time period of zero switching vector
T$_S$	Switching time period
α*	Angle of V* in a 60° sector
f$_s$	Switching frequency
V$_d$	DC link voltage

$$K = \left(\sqrt{3}\cdot T_s / 4\cdot V_d\right)$$

The time periods need to be distributed such that symmetrical PWM pulses are produced. To produce such pulses, the instant of switching on for each phase and each sector is calculated. The generalized equation for turn on instant calculation for phase A is given below [10]. Where, g$_a$(α*) is defined as the turn on pulsewidth function.

To maintain the symmetry of switching, the turn off instant $T_{A\text{-}OFF}$ is calculated and given below.

$$T_{A\text{-}OFF} = T_S - T_{A\_ON}$$

For phases B and C, the switching instants are same but phase shifted by 120º.

III. Neural Based SVM

Multilayer feedforward NN is used to implement SVM. The input to the network is the phase angle of the reference voltage vector θ^*_e and the outputs are the turn-on pulsewidth functions $g_a(\alpha^*)$, $g_b(\alpha^*)$, $g_c(\alpha^*)$ for the phases A, B, and C. Different architectures are proposed keeping the total number of neurons unchanged. The networks are trained independent of switching frequency. To study the influence of increase in layers, different architectures are proposed keeping the total number of neurons unchanged. The various proposed architectures are shown in Fig. 3 – Fig. 5 and are referred to as proposed architecture I, proposed architecture II, and proposed architecture III respectively. For the three proposed architectures, the total number of neurons is at 21. Proposed architecture-I has 1-18-3 structure (single input network with 18 neurons in the hidden layer and 3 neurons in the output layer). The proposed architecture-II has 1-9-9-3 structure and architecture-III has 1-6-6-6-3 structure. The different architectures are chosen so as to increase the number of layer. This concept is useful for implementation in FPGA using layer the concept of multiplexing.

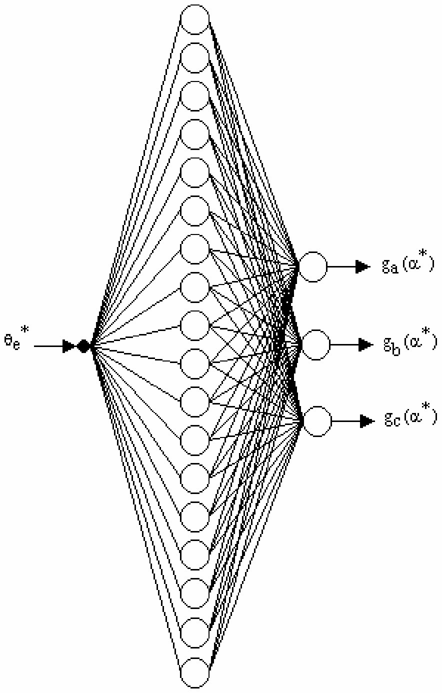


Fig. 3 Proposed architecture I for SVM (1-18-3)

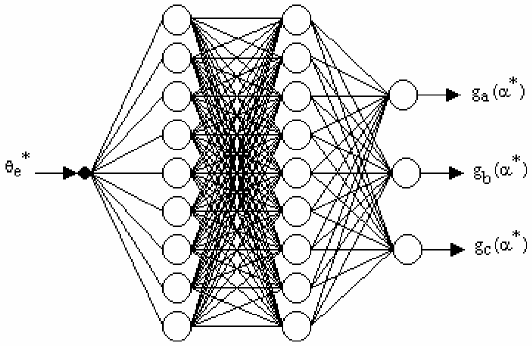


Fig. 4 Proposed architecture II for SVM (1-9-9-3)

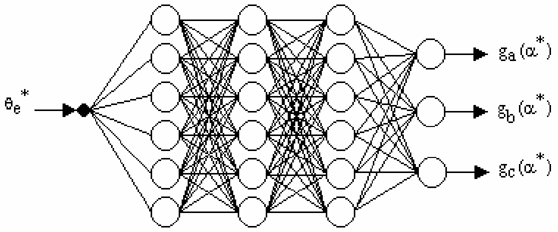


Fig. 5 Proposed architecture III for SVM (1-6-6-6-3)

IV. Performance Of VSI With Different ANN Architectures

The three proposed architectures are trained using back propagation algorithm with 360 input – output target pairs. The intermediate layer use log-sigmoid activation function and the output layer uses linear activation function. The choice of activation function is based on the non-linearity and the output range. To compare the performance of the various architectures, it is decided that all the networks be trained for fixed number of epochs. By trial, it was found that all the networks settled at their least Mean Square Error (MSE) at one-lakh epochs. The mean square error obtained for the proposed networks after one-lakh epochs is shown in Table I. From Table I, it can be observed that the increase in the number of layers reduces the mean square error until an optimum number of layers are reached. Further increase in layers may not improve the performance significantly.

TABLE I
MSE OBTAINED WITH THE PROPOSED ARCHITECTURES

Architecture	Total number of epochs trained	Mean Square Error (MSE) achieved
Architecture I (1-18-3)	1,00,000	5.60e-6
Architecture II (1-9-9-3)	1,00,000	7.30e-7
Architecture III (1-6-6-6-3)	1,00,000	3.48e-6

The performance of the inverter with proposed architectures is evaluated using MATLAB-Tools. The schematic of the ANN based space vector modulated voltage source inverter is shown in the Fig. 6. The inverter and load parameters are given in Table II. The load is an active load with a back emf of 100V/phase. The various performance waveforms are recorded and shown in Fig. 7(a) - 7(h). The line currents shown in Fig. 7 (a) demonstrate the three-phase inverter operation. The line voltage (V_{ab}) and the phase voltage (V_{an}) waveforms shown in Fig. 7(b) and 7(c) respectively, demonstrate the inverter operation for star connected load. The waveforms of Fig. 7(a) – 7(c) are for the proposed architecture II (1-9-9-3).

TABLE II
INVERTER AND LOAD PARAMETERS

DC-link voltage	300 V
Frequency	50 Hz
Load resistance/phase	0.817 Ω
Load inductance/phase	2.38 mH
Back emf/phase	100 V

The THD of the line current is chosen as the inverter performance index. So, a line current waveform of the inverter for various neural structures is shown in the Fig. 7(d). The arrow point in Fig 7(d) is expanded and shown in Fig. 7(e). For all the proposed architectures with typical switching frequencies, THD of line current is given in Table 3 for comparison. The results demonstrate that proposed NN architectures perform equally well. Table 3 highlights that the total harmonic distortion increases with decrease in switching frequency but is independent of the architecture as the mean square error for all the three networks is in the order of 1e-6.

To make a general purpose NN based SVM the network is trained independent of switching frequency. Using such networks, the inverter is operated for various frequencies and the results are shown for selected frequencies. The line current of the inverter for frequencies 20 kHz and 2 kHz are shown in Fig. 7 (f) and 7 (g). The results demonstrate that the neural network based SVM, which is trained independent of frequency, operates well for various switching frequencies of the inverter. From the waveform it is seen that the distortion in the current waveform for 2kHz is higher than that for 20kHz. This confirms that the Total Harmonic Distortion (THD) in the line currents increases with the decrease in switching frequency of the inverter. The pulse pattern produced by NN based SVM in one sampling interval are shown in Fig. 7(h). The pulses are centered within the switching interval, which improves the harmonic performance [6]. The parameters of the networks proposed in this paper are independent of the switching frequency. The coding developed is unchanged. Hence when implemented in hardware, it can be used as general purpose NN based SVM modulator for any application.

TABLE III
COMPARISON OF INVERTER LINE CURRENTS THD FOR VARIOUS ARCHITECTURES AND SWITCHING FREQUENCIES

Switching frequency	Architecture	Line Current (i_a) THD
20kHz	Proposed Architecture I (1-18-3)	0.1340%
	Proposed Architecture II (1-9-9-3)	0.1339%
	Proposed Architecture III (1-6-6-6-3)	0.1340%
10kHz	Proposed Architecture I (1-18-3)	0.2673%
	Proposed Architecture II (1-9-9-3)	0.2672%
	Proposed Architecture III (1-6-6-6-3)	0.2673%
2kHz	Proposed Architecture I (1-18-3)	1.348%
	Proposed Architecture II (1-9-9-3)	1.347%
	Proposed Architecture III (1-6-6-6-3)	1.347%

To choose the best network from the three proposed architectures, the architecture II is the winner if the criterion for selection is least MSE. For hardware implementation using FPGA it has been identified that the architecture having minimum number of neurons in the largest layer significantly reduces the resource requirement [9]. The number of neurons in the largest layer for architecture-I is 18, for architecture-II is 9 and for Architecture-III is 6. Hence for FPGA implementation the Architecture III is the best.

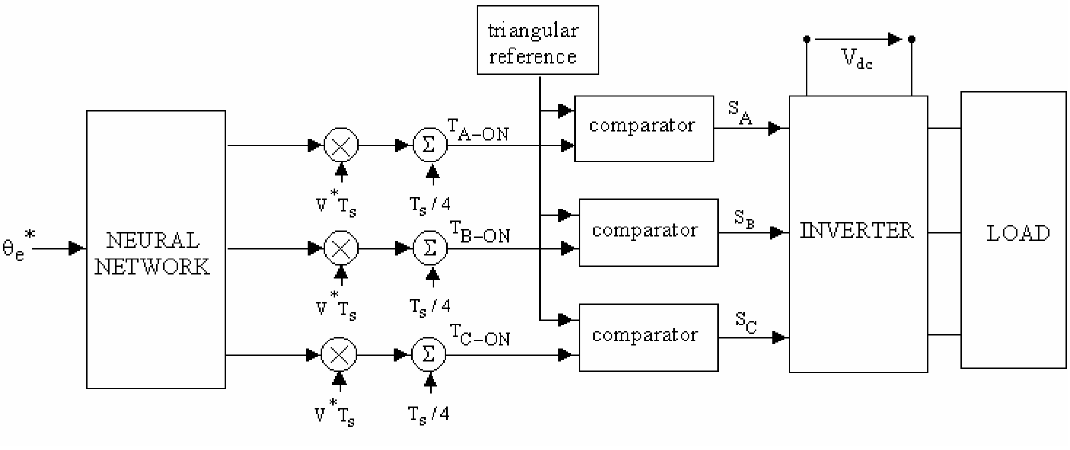


Fig. 6 Schematic of the ANN based SVM inverter

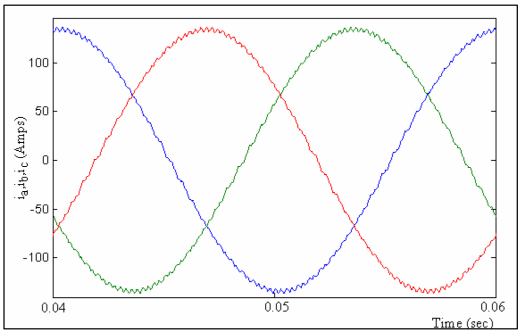


Fig. 7 (a) Three phase currents of a SVM VSI with 1-9-9-3 architecture and T_s=500µs

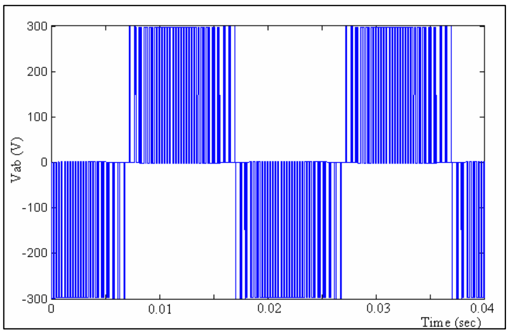


Fig. 7 (b) Line voltage (Vab) of SVM VSI with 1-9-9-3 network and T_s=500µs

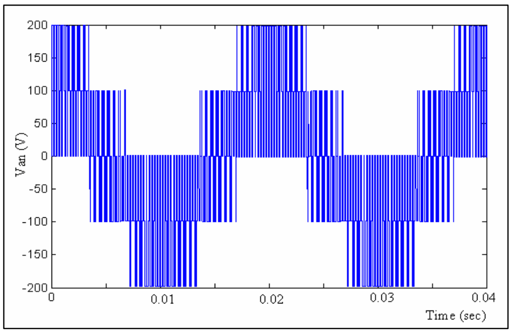


Fig 7 (c) Phase voltage (Van) of SVM VSI with 1-9-9-3 network and T_s=500µs

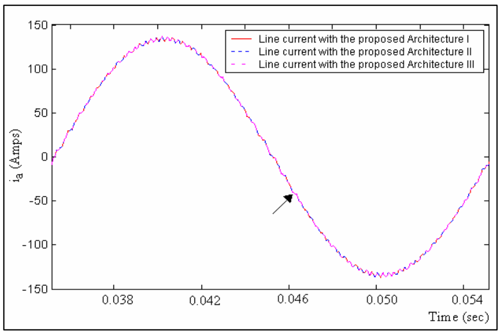


Fig 7 (d) Line current of phase A with proposed architectures for T_s=500µs

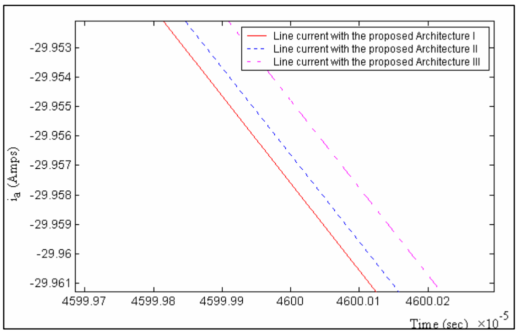


Fig 7 (e) Expanded view of line currents at the instant shown by the arrow in Fig 7(d)

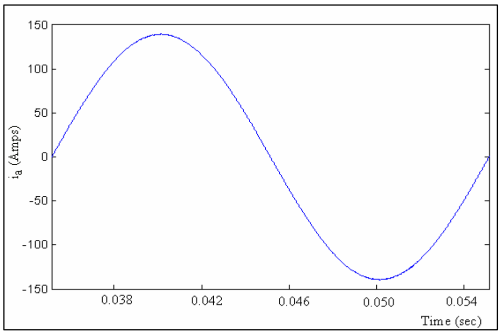


Fig. 7 (f) Phase A line current of SVM VSI with 1-9-9-3 network and T_s=50µs

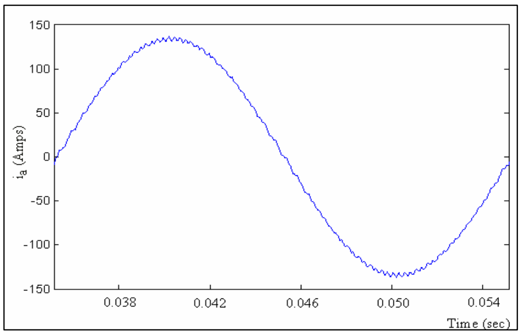


Fig. 7 (g) Phase A line current of SVM VSI with 1-9-9-3 network and T_s=500µs

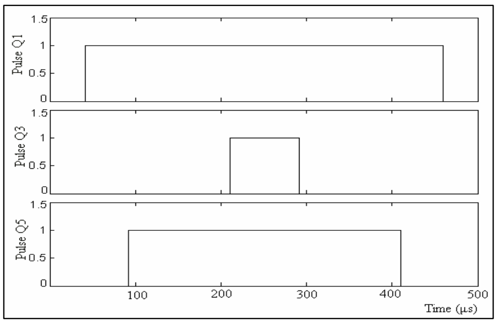


Fig 7 (h) SVM Gate Pulses for upper devices in a sampling interval of T_s=500µs

V. Implementation Of Architecture iii In FPGA

Reconfigurable Field Programmable Gate Arrays (FPGAs) provide an effective programmable resource for implementing hardware based ANNs. They are low cost readily available and reconfigurable. In order to implement the NN with minimum hardware, the concept of layer multiplexing [9] is used. With layer multiplexing it is sufficient to implement only largest layer (layer with largest number of neurons) in the network. The total number of inputs of each neuron in the implemented layer is equal to the number of neurons in the largest layer. The implemented single layer is multiplexed to execute the functions of all layers of the network. In the present case, the network to be implemented has a structure of 1-6-6-6-3. Hence the largest layer in this case is the hidden layer with 6 neurons. Thus it is sufficient to implement only 6 neurons with 6 inputs each.

To realize a network with a total of 21 neurons, only 6 neurons need to be implemented and that leads to considerable resource reduction in FPGA. The saving in the resources with layer multiplexing, to realize the network is given in Table IV in which the resources required to implement the network with and without layer multiplexing is determined using the approximate formulae given in [9]. The neurons in a layer compute in parallel but the layers compute sequentially. Thus the parallelism of the NNs is not affected.

TABLE IV
COMPARISON OF RESOURCE REQUIREMENT FOR ANN (1-6-6-6-3) WITH AND WITHOUT LAYER MULTIPLEXING

Resource requirement in terms of slices		
Without Layer Multiplexing	With Layer Lultiplexing	Saving (%)
6132	**3627**	**40.85%**

The general architecture of the network 1-6-6-6-3 with layer multiplexing is shown in the Fig. 8. The final values of weights and biases obtained after training are stored in an array in the control unit. The control unit controls the operation of the implemented layer by placing appropriate inputs, weights and biases to each of the six neurons at appropriate instances, so as to realize the network.

VI Selection Of Bit Precision

Selecting bit precision is one of the important choices when implementing ANNs on FPGAs. Bit precision is used to trade-off the capabilities of the realized ANNs against the implementation cost. A higher bit precision means fewer quantization errors in the final implementations, while a lower precision leads to simpler designs, greater speed and reductions in area requirements and power consumption. One way of resolving the trade-off is to determine the “minimum precision” required for a given problem. Traditionally, the minimum precision is found through “trial and error” by simulating the solution in software before implementation [10]. The MSE of the network with different input –output (I/O) bit precisions are tabulated in Table VI. The MSE is the sum of squared error between the actual targets of the network and the outputs obtained practically due to bit truncation for the given inputs divided by the total number of patterns. From Table V, it is seen that as the bit precision is increased, the mean square error is reduced.

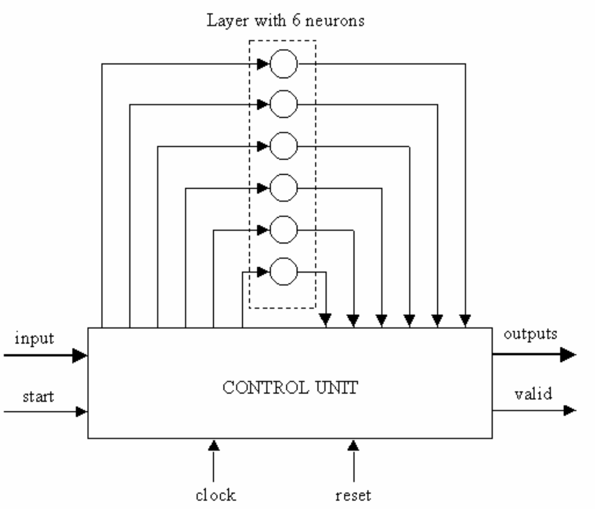


Fig. 8 General architecture of the network

TABLE V
MEAN SQUARE ERROR **(MSE)** ACHIEVED WITH VARIOUS INPUT-OUTPUT BIT PRECISIONS

Input Output BIT PRECISION	MSE
8-bit	78.23 e-2
10-bit	4.89 e-2
12-bit	2.50 e-3
16-bit	3.33 e-5

In order to study the effect of the bit truncation and consequently the MSE, on the inverter performance, the THD of the inverter line currents with the ANN implemented with different bit precisions is determined and is tabulated in Table VI. As the network is independent of switching frequency, the performance of the inverter is studied with the switching frequencies of 2kHz and 20kHz. It can be seen from Table VI that the % THD of line currents with 8-bit precision is not only too large but also it is not same in all the phases. With 10 and 12 bit precisions, even though the % THD is small, it is different in the three phases. This non uniformity is due to the existence of considerable amount of dc quantities. It is observed that, with lower bit precisions, there is considerable amount of dc component in the line currents as tabulated in Table VII.

TABLE VI
THD OF THE LINE CURRENTS WITH DIFFERENT BIT PRECISIONS

Bit Precision	% THD for f_s=2kHz			% THD for f_s=20kHz		
	i_a	i_b	i_c	i_a	i_b	i_c
8	9.634	6.946	7.797	8.607	5.880	5.821
10	1.549	1.867	2.093	0.804	0.855	1.553
12	1.363	1.383	1.406	0.313	0.199	0.441
16	1.347	1.348	1.348	0.134	0.131	0.131
32	1.347	1.347	1.348	0.130	0.130	0.130

TABLE VII
DC COMPONENT IN THE LINE CURRENTS WITH DIFFERENT BIT PRECISIONS

Bit Precision	DC Component for f_s = 2kHz			DC component for f_s = 20kHz		
	i_a	i_b	i_c	i_a	i_b	i_c
8	7.874	3.389	4.485	7.648	3.534	4.114
10	0.358	1.034	0.393	0.663	0.818	1.481
12	0.212	0.142	0.354	0.269	0.142	0.411
16	0.0365	0.005	0.031	0.029	0.014	0.015
32	6.41e-4	5.85e-3	6.49e-3	4.62e-4	6.51e-5	5.27e-4

16-bit precision reduces the dc quantities in the line currents and maintains equal THD in all lines. The same conclusion can be drawn form the line current waveforms shown in Fig. 9. The waveforms show three line-currents of the inverter, which is operated with NN, based SVM implemented with 8-bit and 16-bit precision. It is seen from the Fig. 9 that, with 8-bit precision, the three phase line currents are unbalanced which is due to the presence of dc quantities whereas 16-bit precision shows better results. Thus, it is identified that, the NN based SVM requires minimum of 16-bit precision for acceptable inverter performance for the practical application.

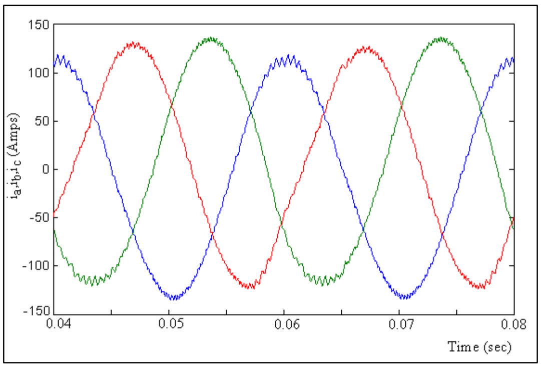


Fig. 9 (a) Line currents of the inverter with the ANN having 8-bit precision for f_s=2kHz

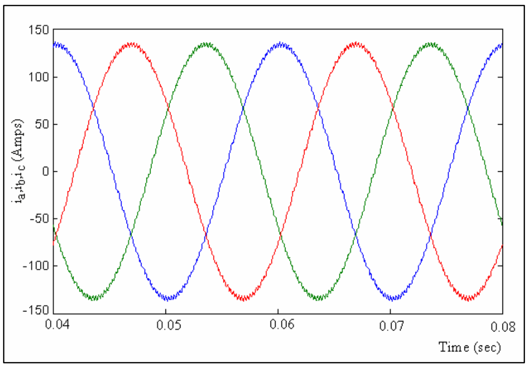
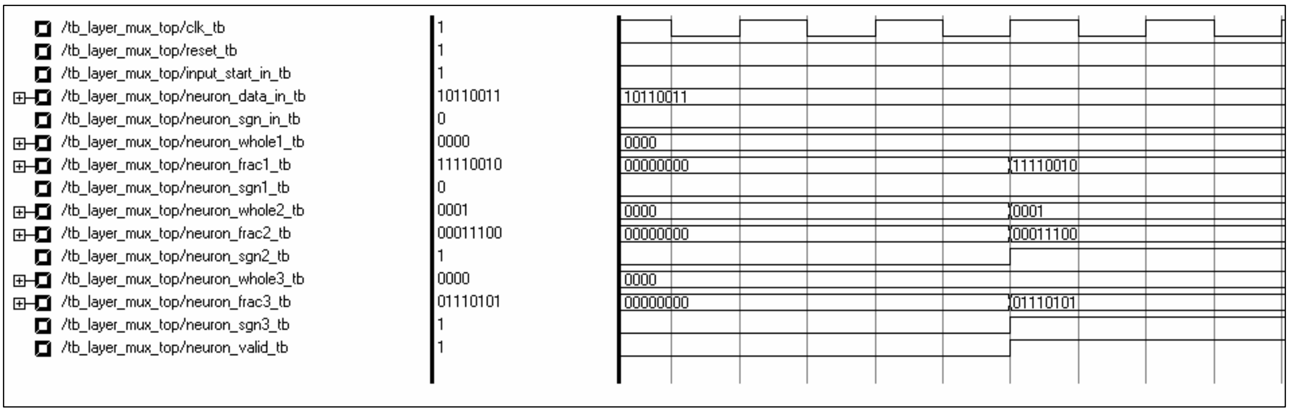


Fig. 9 (b) Line currents of the inverter with the ANN having 16-bit precision for f_s=2kHz

The NN with 8-bit precision has been implemented using Xilinx 6.1i, simulated with ModelSim XE II 5.7c, and downloaded and tested in the IC 'XCV400hq240' using Test equipment-model: MXUK-SMD-001. The outputs of the network are tested with the outputs simulated using the same precision. They are found to be the same and are given in Table VIII. Thus the network is found to perform well. The synthesis report and device utilization summary for the network 1-6-6-6-3 implemented in the IC XCV400hq240 with 8-bit precision are given in the appendix. The output of the network is shown in Fig. 10.

TABLE VIII
SAMPLE INPUT AND OUTPUT OF THE ANN

Input	Value of Outputs with and without Bit Truncation		
	Simulated		Practical FPGA Outputs
	Without Truncation	With 8-Bit Precision	With 8-Bit Precision
0. 701172 0 10110011	0.978147 0 0000 11111010	0.941406 0 0000 11110001	0.941406 0 0000 11110001
	-0.978147 1 0000 11111010	-1.109375 1 0001 00011100	-1.109375 1 0001 00011100
	-0.360113 1 0000 01011100	-0.457031 1 0000 01110101	-0.457031 1 0000 01110101



```
/tb_layer_mux_top/clk_tb              1
/tb_layer_mux_top/reset_tb            1
/tb_layer_mux_top/input_start_in_tb   1
/tb_layer_mux_top/neuron_data_in_tb   10110011
/tb_layer_mux_top/neuron_sgn_in_tb    0
/tb_layer_mux_top/neuron_whole1_tb    0000
/tb_layer_mux_top/neuron_frac1_tb     11110010
/tb_layer_mux_top/neuron_sgn1_tb      0
/tb_layer_mux_top/neuron_whole2_tb    0001
/tb_layer_mux_top/neuron_frac2_tb     00011100
/tb_layer_mux_top/neuron_sgn2_tb      1
/tb_layer_mux_top/neuron_whole3_tb    0000
/tb_layer_mux_top/neuron_frac3_tb     01110101
/tb_layer_mux_top/neuron_sgn3_tb      1
/tb_layer_mux_top/neuron_valid_tb     1
```

Fig. 10 Results of the ANN based SVM implemented in FPGA with 8-bit input output Precision

VI. Conclusion

Various implementation and performance aspects of NN based SVM are investigated. Towards these objective, different architectures of NN based SVM are proposed, trained and evaluated. The extensive results obtained are detailed. From the investigation, the 1-6-6-6-3 network is the best architecture with minimum resource for FPGA implementation and good inverter performance. The above architecture reduces resources by 40% using a concept of layer multiplexing. The identified architecture with 8-bit precision is implemented and tested on Xilinx XCV400hq-240 and the inverter performance is studied. The effect of bit precision on the inverter performance is evaluated and reported. The general purpose NN based space vector modulator is demonstrated for various switching frequency. The performance of the inverter is presented. The NN based SVM with 1-6-6-6-3 architecture and 16-bit precision is concluded to be optimum in terms of implementation and inverter performance.

Appendix

1. HDL Synthesis Report for the Network 1-6-6-6-3 with 8-Bit Precision

```
RTL Top Level Output File Name : network_top.ngr
Top Level Output File Name        : network_top
# IOs: 52
# RAM    :1
            # 1024x8-bit single-port block RAM: 1
# ROMs: 5
            # 1024x8-bit ROM: 5
# Registers: 427
            # 1-bit register: 247
            # 10-bit register: 12
            # 16-bit register: 42
            # 2-bit register: 6
            # 24-bit register: 18
            # 28-bit register: 42
            # 4-bit register: 10
            # 7-bit register: 6
            # 8-bit register: 44
# Counters: 1
            # 7-bit up counter: 1
# Multiplexers: 77
            # 2-to-1 multiplexer: 77
# Adders/Subtractors: 25
            # 28-bit adder: 6
            # 28-bit subtractor: 12
            # 4-bit adder: 1
            # 7-bit adder: 6
# Multipliers: 6
            # 16x8-bit multiplier: 6
# Comparators: 6
            # 28-bit comparator greatequal: 6
```

2. Device Utilization Summary for the Network 1-6-6-6-3 with 8-bit Precision Selected Device: v400hq240-5

Number of Slices	*: 3627 out of*	*4800*	*75%*
Number of Slice Flip Flops	*: 3013 out of*	*9600*	*31%*
Number of 4 input LUTs	*: 6663 out of*	*9600*	*69%*
Number of bonded IOBs	*: 51 out of*	*170*	*30%*
Number of BRAMs	*: 2 out of*	*10*	*20%*
Number of GCLKs	*: 1 out of*	*4*	*25%*

Acknowledgment

The research is supported by the grants from the All India Council for Technical Education (AICTE), a statutory body of Government of India. File Number: ***8020/RID/TAPTEC-32/2001-02.***

References

[1] J. Holtz, " Pulsewidth Modulation – A survey", *IEEE Trans. Ind. Electr.,* Vol. 39, pp. 410-419, Dec 1992.

[2] H.W.Van Der Broek, H.C.Skundelny and G.V.Stanke, "Analysis and realization of a pulsewidth modulator based on voltage space vectors", *IEEE Trans. Ind. Appl.,* Vol. 24, pp. 140-150, Jan/Feb 1988.

[3] J.O.P.Pinto, B.K.Bose, L.E.Borges da Silva and M.P.Kazmierkowski, " A Neural Network based space vector pwm controller for voltage fed inverter induction motor drive", *IEEE Trans. Ind. Appl*., Vol. 36, No. 6, Nov/Dec 2000.

[4] A. Bakhshai, J. Espinoza, G. Joos, and H. Jin, "A combined artificial neural network and DSP approach to the implementation of space vector modulation techniques," in *Conf. Rec. IEEE-IAS Annu. Meeting*, 1996, pp. 934–940.

[5] K.Zhou and D.Wang, " Relationship between space vector modulation and three phase carrier base PWM- A comprehensive analysis", *IEEE Trans. Ind. Electr.,* Vol. 49, No. 1, Feb. 2002.

[6] Jihan Zhu and Peter Sutton, "FPGA implementations of neural networks – a survey of a decade of progress", 2003.

[7] M.Marchesi, G.Orlandi, F.Piazza, and A.Uncini, " Fast Neural Networks Without Multipliers", *IEEE Trans. Neural Networks*, Vol. 4, No.1, Jan.1993, pp.53-61.

[8] D.Anitha "FPGA Implementation of Estimators for Sensorless Control of DTC Drives", M.Tech(EDC) Thesis, EEE Dept., PEC., Pondicherry University, June 2005.

[9] S.Tamura and M.Tateishi, " Capabilities of a Four-Layered Feedforward Neural Network: Four Layers Versus Three", *IEEE Trans. Neural Networks*, Vol. 8, No.2, March 1997.

[10] B.K.Bose, *Modern Power Electronics and ac drives*, Pearson Education (Singapore) Pvt. Ltd., India, 2003.

A.Muthuramalingam received his BE degree in Electrical and Electronics Engg with Distinction and Gold Medal from Annamalai University, Tamil Nadu, India, in 1982, M.Tech degree in Industrial Electronics with Distinction from KREC under Mangalore University, Karnataka, India, in 1985 and Ph.D degree in Power Electronics and Drives from Indian Institute of Technology, Madras, India, in 2002.

He is working as Assistant Professor in Pondicherry Engineering College where he has been harnessing the developmental activities of the Electrical and Electronics Engineering department. He has taken an assignment in which students learn technological advancements in depth through design-oriented courses using CAD tools. His current research interest is on application of soft switching techniques for medium power conversion systems and AI techniques to PE systems and Solid State Drives.

S.Himavathi completed her BE degree in Electrical and Electronics Engg from College of Engineering, Guindy, Chennai, India, in the year 1984. She obtained her M.E. degree in Instrumentation Technology in the year 1987 from Madras Institute of Technology, Chennai, India. She completed her Ph.D degree in the area of Fuzzy modeling in the year 2003 from Anna University, Chennai, India.

She is an Assistant Professor in the Department of Electrical and Electronics Engineering, Pondicherry Engineering College. She has around 30 publications to her credit. She is a reviewer of the AMSE journal of Modeling, IEEE Industrial Electronics Society and Asian Neural Networks Society. Her research interests are Fuzzy systems, Neural Networks, Hybrid systems and their applications.