# Harmonics Elimination in PWM Inverters Using Neural Network

*P.P. Mohanlal and S. Dasgupta*

*Control and Guidance Design Group,*
*Vikram Sarabhai Space Centre,*
*Thiruvananthapuram — 695 022*

Abstract:

Programmed PWM schemes require many waveforms to be stored in memory for varying values of output fundamental magnitudes. It is only useful for harmonic reduction when the drive is operating in the steady state and not for operation in the transient state. The applicability of Neural Network for harmonic reduction in PWM inverters is established to handle the continous variation of output fundamental amplitude.A method suggested to incorporate the random phase modulation of the switching angles with the Neural scheme to smear the concentrated energy at the switching frequency.The limitation of the random modulation ,specially at low control ratios and the usefulness of the Neural scheme are brought out.

## INTRODUCTION

The purpose of PWM (pulse width modulation) in inverters is to change the ratio of the fundamental of the ac voltage to the dc voltage . Some of the modulation schemes employed are endpulse modulation ,centrepulse modulation,sinusoidal PWM,stair case PWM etc.The switched waveforms in these modulation schemes contain low order harmonics which are often unacceptable.All these modulation schemes maintain symmetry in the switched waveform due to which the even harmonics will be absent. A typical switched waveform of single phase is shown in Fig. 1 in which symmetry is maintained. By reversing the phase potentials a number of times during each half cycle , the spectra of harmonics can be changed in such a way that some low order harmonics troublesome to the load are cancelled where as some higher order harmonics which are less harmful increase in magnitude. The notches are placed symmetrically about the centre of each half cycle to maintain symmetry.

A waveform with the angles of reversal equal to $\alpha_1, \alpha_2, \alpha_3, \ldots, \alpha_n$ has the following rms values of the n th harmonic voltage

$$U(n) = (0.45 / n)\, u_d \{2(\cos n\alpha_1 - \cos n\alpha_2 + \cos n\alpha_3 - \ldots) - 1\} \quad \ldots\ldots\ldots (1)$$

where U(n) : amplitude of n th harmonic
$u_d$ : Input DC voltage
n : harmonic number

$\gamma$ (control ratio) : (actual fundamental / fundamental of unmodulated wave )

For example, control of the fundamental and cancellation of the 5 th and 7th harmonics will result in the following system of nonlinear equations[4] .

$$2(\cos \alpha_1 - \cos \alpha_2 + \cos \alpha_3) - 1 = \gamma \quad \ldots\ldots\ldots (2)$$

$$2(\cos 5\alpha_1 - \cos 5\alpha_2 + \cos 5\alpha_3) - 1 = 0 \quad \ldots\ldots\ldots (3)$$

$$2(\cos 7\alpha_1 - \cos 7\alpha_2 + \cos 7\alpha_3) - 1 = 0 \quad \ldots\ldots\ldots (4)$$

Each of the reversal corresponds to one degree of freedom , making it possible either to cancel a harmonic or to control the fundamental voltage.In programmed PWM , the switching angles from Equations. (2) - (4) are precomputed numerically for different discrete values of $\gamma$ and stored for use.

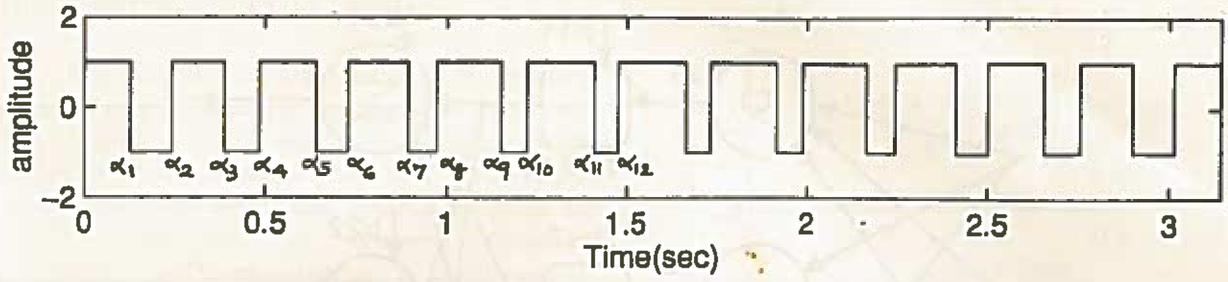


**Fig. 1, PWM switched waveform for half period ($\gamma = 0.37$)**

This requires many waveforms to be stored in addition to other limitations.

If we choose an even number (p) of switching angles, there will be (p-1) number of odd harmonics below the switching frequency $f_s = (2p+1)f$, where f is the fundamental frequency. Now we can have p number of equations of the type (2) - (4) with p unknown switching angles ($\alpha_1, \alpha_2, \alpha_3, .... \alpha_p$) for a given value of control ratio. Thus for a given control ratio,p switching angles can be solved from these equations to null all the (p-1) odd harmonics below the switching frequency.This will result in no harmonics energy below switching frequency.

## DESIGN PARAMETERS AND TRAINING DATA FOR NEURAL NETWORK.

In this study, p = 12 , f = 1 rad/sec, $f_s = 25$ rad/sec There are 11 odd harmonics below $f_s$ to be nulled . 12 equations of the type (2) - (4) are formed and solved using gradient method for $\gamma = 0.02, 0.04, 0.06, ....... 0.74$ to get 12 switching angles for each value of $\gamma$. This data set is used to train the Neural Network.Sample data in degrees, for $\gamma = 0.02$ to 0.20 are tabulated in Table.1 .

## TRAINING OF NEURAL NETWORK

It is well known that suitably selected feedforward neural network can approximate any arbitary nonlinear static mapping with desired accuracy.When the training data spans the input space reasonably, the network can interpolate the function quite well for untrained inputs.Multilayer feedforward and Radial basis networks are commonly employed for the above purpose.

Radial basis networks require less training time and interpolate well if the training data spans the input space uniformly. For this application, the radial basis network was found to be more suitable from the point of view of training time,interpolation accuracy and number of neurons.The basic form of the radial basis neural network with r-inputs, m-outputs and q-radial basis neurons is shown in Fig. 2 Orthogonal least squares training algorithm [1] is used for the training of the network.The spread constant is chosen as 0.08. The sum squared error could be reduced to less than 10e-10 with just 29 neurons in the radial basis layer.

Training inputs are the different $\gamma$ values from 0.02 to 0.74 . For each scalar input, there are 12 outputs ($\alpha_1, \alpha_2, ....\alpha_{12}$) which means that 12 neurons in the output layer.The training data generated is used to train the network. To assertain the interpolation accuracy, the test data set is generated as follows. For $\gamma = 0.01, 0.03, ......0.73$ , the corresponding firing angle vectors are again computed by gradient method and used to test the network for interpolation accuracy.

This sample data for $\gamma = 0.01$ to 0.19 are tabulated in Table 2. Fig. 3a , 3b and 3c give the training error and interpolation accuracy respectively for training inputs and test inputs. The interpolation error is less than 5e-5.

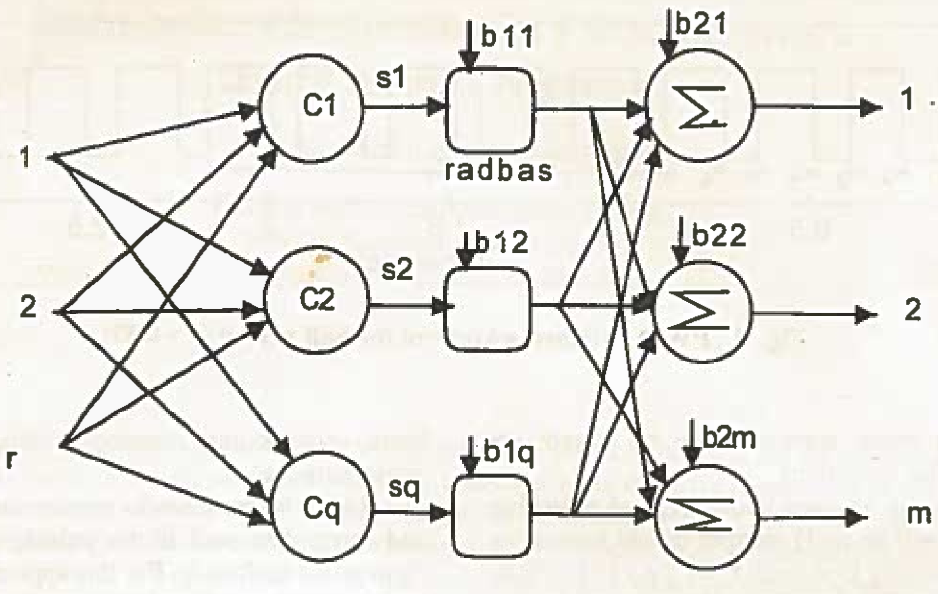


**Fig. 2 , Radial Basis Neural Network.**

Where,

$C_1, C_2, .. C_q$ are radial centres of dimension $R^r$

$s_1, s_2, ... s_q$ are the Euclidean distances between input and radial centres.( i.e. $s_1$ = input - $C_1$ etc.).

radbas$(s_1, b_{11}) = \exp(-(b_{11} * s_1)^2)$

$b_{11}, b_{12}, .... b_{1q}$ = (0.8326)/(spread constant)

$b_{21}, b_{22}, .... b_{2m}$ : bias inputs

## SIMULATION AND ANALYSIS OF SWITCHED PWM WAVEFORM

Three control ratios ($\gamma = 0.01$ , 0.37 , 0.73 ) are chosen to test the Neural Network based PWM switching scheme. It is to be noted that the network was not trained for these inputs.Fig 5a , 5b and 5c give the spectral output of the switched waveform . The harmonics below switching frequency are almost invisible. The linearity of the output amplitude with respect to the control ratios is given in Fig 4.

## SIMULATION OF RANDOM MODULATION OF SWITCHING ANGLES AND SMEARING OF SWITCHING TONE ENERGY

In [2, 3] , random modulation of the carrier of the sine PWM was used to reduce the acoustic noise in PWM drives.The idea is that using random modulation, the energy concentrated at switching frequency can be made to spread over a desired frequency range.This is acheived with the average switching frequency the same as the unmodulated case,unlike other schemes where switching frequency is very high.

In the present Neural Net PWM , to acheive random modulation, uniformly distributed,amplitude limited ( -7 to 7 ) , zero mean random sequence is generated and added to each phase angle output of the Neural Network. The resulting PWM waveform will have average switching frequency same as the unmodulated case.The spectral analysis of the output waveform with random modulation is given in Fig 6 a , 6b and 6c for the three cases for which the unmodulated spectra are in Fig 5 a ,5 b and 5 c respectively.

Here, for $\gamma = 0.73$ , the random modulation is found to be useful. Where as for $\gamma = 0.01$, the random modulation makes the output completely useless.For $\gamma = 0.37$, the effect is moderate. This means that random modulation is useful only at higher control ratios in PWM drives.

## DISCUSSION AND CONCLUSIONS

The Neural Network based PWM switching scheme has good linearity and harmonic rejection.It has the ability of real time and continous output amplitude control. Random phase modulation can be incorporated with the Neural scheme to spread the energy concentrated at switching frequency eventhough random modulation is useful only at high control ratios.

The switching frequency can be increased by increasing the number of phase angle reversal points. For example, with p = 24, $f_s$ nearly doubles ( 49 rad/sec) .Since all the harmonics below switching frequency can be nulled, it is desirable to choose an appropriate value of p such that the resulting switching frequency does not harm the load. This is a better scheme than the random modulation method since the later is not useful at low control ratios. However, the option of random modulation can be retained with Neural Net based PWM .

The Neural Net based PWM is better than programmed PWM due to its capability to handle continous amplitude control. It is also better than the random modulation scheme since the later is not useful at low control ratios. The hardware implementation aspects need to be explored.

## REFERENCES


1. S.Chen, C.F.N. cowan & P.M. Grant, Orthogonal Least Squares learning algorithm, IEEE Trans on Neural Networks,vol 2 , no.2, pp302 -309, March1991.
2. T.G.Habetler & D.M.Diwan; Acoustic Noise Reduction in sinusoidal PWM drives using a randomly modulated carrier, IEEE Trans on Power Electronoics, vol. 6, No. 3 , July 1991.
3. S.Y.R. Hui & S. Sathiakumar , Optimisation of microprocessor based random PWM scheme for power Inverters with low switching frequencies, IEEE 9th Applied Power Electronics Conference, Oriando,FL , pp 214 -218, 1994.
4. Kjeld Thorborg , Power Electronics, Prentice Hall 1988.
5. Howard Demuth & Mark Beale , Neural Network ToolBox manual for use with MATLAB , Mathworks Inc; 1994 .


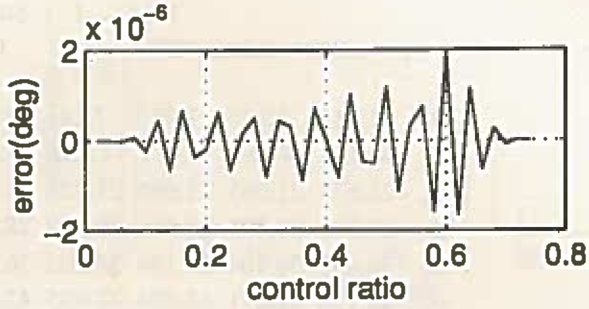


Fig. 3a ,Training errors ($\alpha_{12}$)

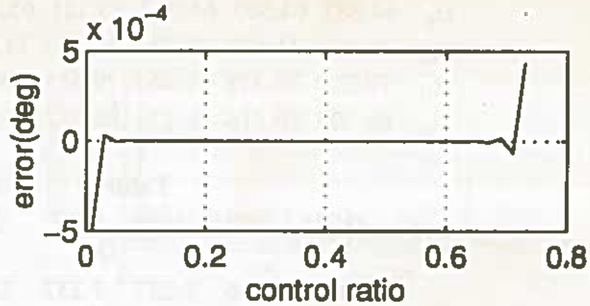


Fig. 3b, Interpolation errors ($\alpha_{12}$)

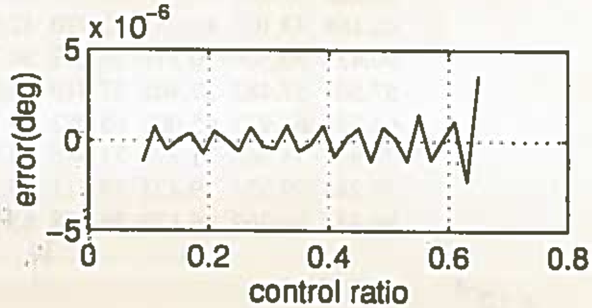


Fig. 3c, Interpolation errors ( $\alpha_{12}$ )

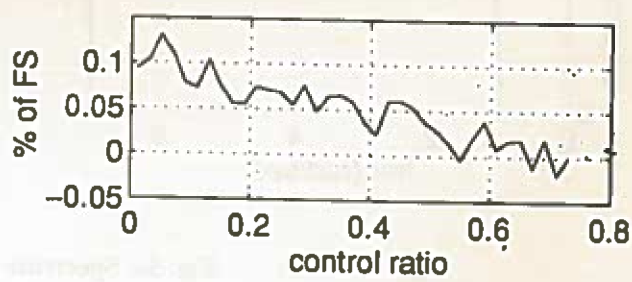


Fig. 4, Linearity error with NNet PWM

**Table . 1 ( SampleTraining Data )**

| $\gamma$ | 0.02 | 0.04 | 0.06 | 0.08 | 0.10 | 0.12 | 0.14 | 0.16 | 0.18 | 0.20 |
|---|---|---|---|---|---|---|---|---|---|---|
| $\alpha_1$ | 7.211 | 7.222 | 7.232 | 7.242 | 7.252 | 7.261 | 7.269 | 7.278 | 7.285 | 7.292 |
| $\alpha_2$ | 14.376 | 14.352 | 14.327 | 14.302 | 14.275 | 14.248 | 14.220 | 14.192 | 14.162 | 14.132 |
| $\alpha_3$ | 21.633 | 21.665 | 21.695 | 21.725 | 21.754 | 21.781 | 21.807 | 21.831 | 21.855 | 21.877 |
| $\alpha_4$ | 28.755 | 28.708 | 28.661 | 28.612 | 28.561 | 28.509 | 28.456 | 28.401 | 28.345 | 28.287 |
| $\alpha_5$ | 36.053 | 36.104 | 36.154 | 36.202 | 36.249 | 36.293 | 36.336 | 36.378 | 36.417 | 36.455 |
| $\alpha_6$ | 43.136 | 43.071 | 43.004 | 42.935 | 42.865 | 42.793 | 42.719 | 42.643 | 42.566 | 42.487 |
| $\alpha_7$ | 50.469 | 50.537 | 50.604 | 50.669 | 50.732 | 50.793 | 50.853 | 50.910 | 50.966 | 51.020 |
| $\alpha_8$ | 57.521 | 57.442 | 57.360 | 57.278 | 57.193 | 57.107 | 57.020 | 56.930 | 56.839 | 56.746 |
| $\alpha_9$ | 64.882 | 64.963 | 65.043 | 65.121 | 65.198 | 65.274 | 65.348 | 65.421 | 65.493 | 65.562 |
| $\alpha_{10}$ | 71.912 | 71.823 | 71.733 | 71.643 | 71.551 | 71.458 | 71.364 | 71.268 | 71.172 | 71.074 |
| $\alpha_{11}$ | 79.289 | 79.378 | 79.467 | 79.555 | 79.642 | 79.729 | 79.815 | 79.901 | 79.986 | 80.070 |
| $\alpha_{12}$ | 86.308 | 86.216 | 86.124 | 86.032 | 85.939 | 85.846 | 85.753 | 85.659 | 85.565 | 85.471 |

**Table . 2 ( Sample Test Data)**

| $\gamma$ | 0.01 | 0.03 | 0.05 | 0.07 | 0.09 | 0.11 | 0.13 | 0.15 | 0.17 | 0.19 |
|---|---|---|---|---|---|---|---|---|---|---|
| | 7.205 | 7.216 | 7.227 | 7.237 | 7.247 | 7.256 | 7.265 | 7.273 | 7.281 | 7.289 |
| | 14.388 | 14.364 | 14.340 | 14.315 | 14.289 | 14.262 | 14.234 | 14.206 | 14.177 | 14.147 |
| | 21.616 | 21.649 | 21.680 | 21.710 | 21.739 | 21.767 | 21.794 | 21.819 | 21.843 | 21.866 |
| | 28.777 | 28.732 | 28.685 | 28.636 | 28.586 | 28.535 | 28.483 | 28.429 | 28.373 | 28.316 |
| | 36.026 | 36.076 | 36.129 | 36.178 | 36.226 | 36.271 | 36.315 | 36.357 | 36.398 | 36.436 |
| | 43.168 | 43.103 | 43.037 | 42.970 | 42.900 | 42.829 | 42.756 | 42.681 | 42.605 | 42.526 |
| | 50.435 | 50.504 | 50.571 | 50.637 | 50.700 | 50.763 | 50.823 | 50.882 | 50.938 | 50.993 |
| | 57.561 | 57.482 | 57.401 | 57.319 | 57.236 | 57.150 | 57.064 | 56.975 | 56.887 | 56.793 |
| | 64.841 | 64.922 | 65.003 | 65.082 | 65.160 | 65.236 | 65.311 | 65.385 | 65.457 | 65.528 |
| | 71.956 | 71.868 | 71.778 | 71.688 | 71.597 | 71.504 | 71.411 | 71.316 | 71.220 | 71.123 |
| | 79.244 | 79.334 | 79.423 | 79.511 | 79.599 | 79.686 | 79.772 | 79.858 | 79.943 | 80.028 |
| | 86.354 | 86.262 | 86.170 | 86.078 | 85.985 | 85.893 | 85.799 | 85.706 | 85.612 | 85.518 |

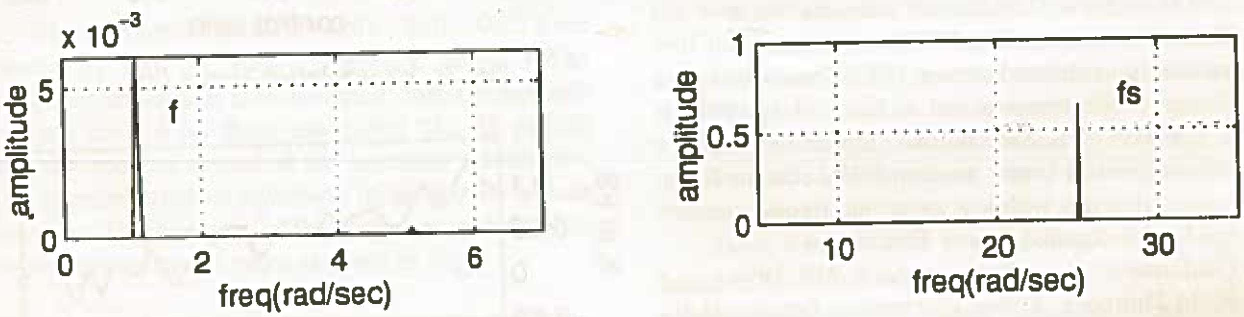


**Fig. 5a, Spectrum of o/p ( $\gamma = 0.01$ )**

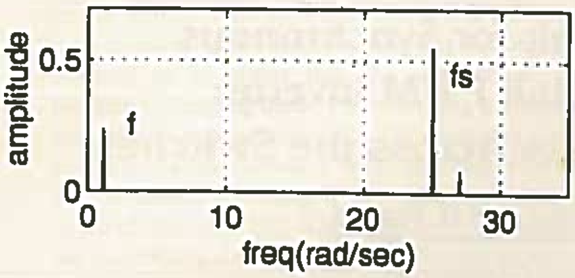


**Fig. 5b, Spectrum of o/p ( $\gamma = 0.37$ )**

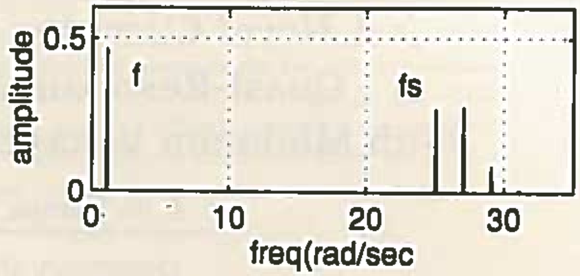


**Fig. 5c, Spectrum of o/p ( $\gamma = 0.73$)**

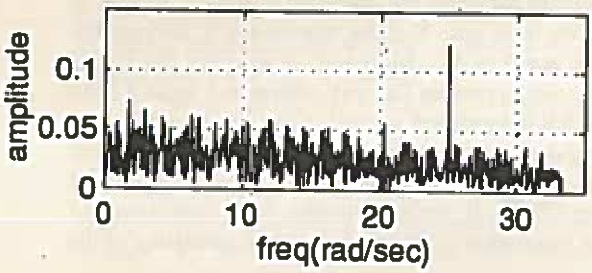


**Fig 6 a , Spectrum ( $\gamma = 0.01$ ) rand mod**

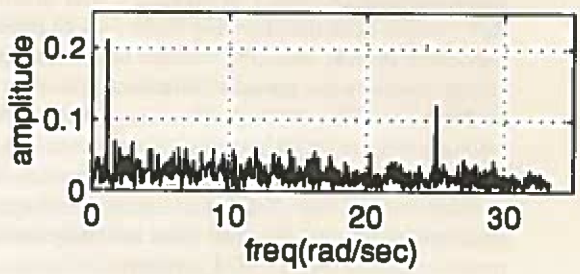


**Fig 6 b , Spectrum ($\gamma = 0.37$ ) rand mod**

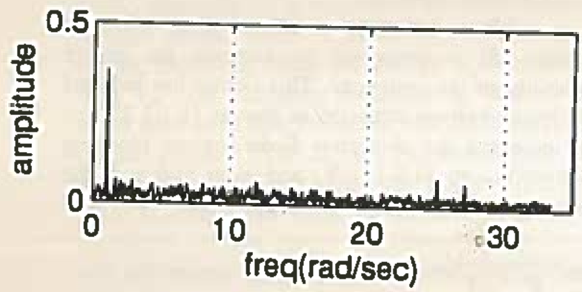


**Fig 6 c , Spectrum ($\gamma = 0.73$ ) rand mod**