CHAPTER 7

SELECTIVE HARMONIC ELIMINATION USING ARTIFICIAL NEURAL NETWORKS

7.1 INTRODUCTION

In the previous chapter the selective harmonic elimination using computed pulse width modulation (CPWM) technique by solving algebraic equations was discussed and it is found that it gives better solution when compared to carrier based PWM techniques. The CPWM has the ability to eliminate the undesired harmonics, up to the specified order and to control the magnitude of fundamental RMS voltage at the output of voltage source inverter. The execution of CPWM needs solving a set of nonlinear equations, contain trigonometric terms and are transcendental in nature. Consequently multiple solutions are possible. These equations should be solved by numerical methods, first by applying Newton Raphson’s method to obtain a linearized set of equations and the solution of these equations is achieved by Gauss elimination method. The numerical methods have inherent difficulties such as divergence of response, convergence to wrong results, consuming time and requiring large memory size (Chow et al 1993, Kim et al 1996). The convergence problems are highly arising especially when the numbers of equations are increased. Neural networks can be used to solve these problems especially in on/off line applications (Sirisukprasert et al 2002).

The application of artificial neural networks (ANN) is recently growing in power electronics and drives area. In this chapter a feed forward
neural network with back propagation training algorithm is used to generate the computed PWM signals. A feed forward ANN basically implements nonlinear input-output mapping. The computational delay of this mapping becomes negligible if parallel architecture of the network is implemented. For any chosen objective function, the optimal switching pattern depends on the desired modulation index. In the existing practice, the switching patterns are pre-computed for all the required values of this index, and stored in look-up tables of a microprocessor-based modulator. This requires a large memory and computation of the switching angles in real time is, as yet, impossible. To overcome this, an ANN is trained in off-line to obtain the notching angles required to generate an output voltage without using the real time solution of nonlinear harmonic elimination equation. The complete set of solutions to the nonlinear equations is found using the back propagation of the errors between the desired harmonic elimination and the nonlinear equation systems using the switching angle given by the ANN (Bowes 1990).

The ANN offers the following advantages over the conventional techniques. In this technique, it is not necessary to establish specific input-output relationships but they are formulated through a learning process. Complex iterations involved in solving the nonlinear equations using any of the numerical methods are eliminated. Though training takes a long time it is not a disadvantage since the training is carried offline. The conventional lookup table method involves larger memory which is not required in case of neural net. After the weights of the neural net are determined the net can be implemented online. Parallel neuron units enable parallel processing reducing the computation time significantly in real time (Bouhali et al 2005). In this chapter the computed PWM signals for various values of N and modulation index (M) are generated using ANN and it is applied to single phase voltage source inverter to eliminate harmonics selectively for both simulation and hardware models.
7.2 NEURAL APPROACH FOR SELECTIVE HARMONIC ELIMINATION

The Artificial Neural Networks is applied for the execution of the Selective Harmonic Elimination of the single phase voltage controlled inverter given by the Figure 6.12. Using the ANN, it is possible to eliminate the desired harmonics in the output of the inverter in addition to the control of the fundamental RMS voltage. Commonly neural networks are adjusted, or trained, so that a particular input leads to a specific target output. Typically many such input/target pairs are used, in this supervised learning, to train a network. There are many algorithms used for the training of the neural network. They are Hebb net, perceptron, Adaline, Madaline, Back propagation, Kohonen, ART net, etc., among which the “Back propagation algorithm” is the most well known and widely used. It is a multilayered, feed forward network employing sigmoidal activation function. The learning rule is known as back propagation which is a kind of gradient descent technique with backward error propagation. The idea of gradient descent can be put to use in such a derivation and the resulting algorithm is the “Back propagation” algorithm (BP) (Satish Kumar 2004). The BP algorithm is popular for its simplicity of implementation and its ability to quickly generate networks that have the capability to generalize.

The training set for the network must be presented many times in order for the interconnection weights between the neurons to settle into a state for correct classification of input patterns (Satish Kumar 2004). The network learns a mapping from a set of input patterns to a set of output patterns. The ability comes from the nodes in the hidden layers which learn to respond to features found in the input patterns. The back propagation net consists of one input layer, one output layer and one or more hidden layers. The use of sigmoidal function is essential for the system of nonlinear equations since it
enables to obtain more robust solutions (Andrzej Cichoki and Rolf Unbehauen 1992). Sigmoidal function used in back propagation compresses the range of output between 0 and 1. It can accommodate large signals without saturation while allowing the passing of small signals without excessive attenuation.

For linear neurons in the input layer, the function used is,

\[ \delta(x) = x \]  

(7.1)

and for sigmoidal neurons in the hidden and output layers,

\[ \delta(x) = \frac{1}{1 + e^{-\lambda x}}, \]  

(7.2)

where \( \lambda = 1 \) typically.

The nature of the problem of learning allows us to demarcate learning algorithms into two categories: supervised and unsupervised. In supervised learning, input-output sample pairs are employed to train the net through a simple form of error correction learning or gradient descent weight adaptation. These procedures are based on global error measures derived from the difference between the desired \( (D_k) \) and actual \( (S_k) \) output of the network to reduce the error \( (D_k - S_k) \) given by the “delta rule”. In unsupervised learning, the system is simply provided with an input \( (X_k) \) and it is allowed to self-organize its weights. The unsupervised learning process is self-organized and is driven by intra-field neuronal competition and cooperation (Laurene Fausett 2004).

### 7.2.1 Feed Forward Network

The basic structure of a three layer feed forward neural network (Tarafdar Haque and Taheri 2005, Bimal K. Bose 2005) is shown in
Figure 7.1. This neural network has one input, some hidden and N output neurons respectively to eliminate N-1 harmonics. The input accepts an input data and distributes it to all neurons in the middle layer. The input layer is usually passive and does not alter the input data. The neurons in the middle layer act as feature detectors. They encode in their weights a representation of the features present in the input patterns. The output layer accepts a stimulus pattern from the middle layer and passes a result to a transfer function block which usually applies a nonlinear function and constructs the output response pattern of the network. The number of hidden layers and the number of neurons in each hidden layer depend upon the network design consideration and there is no general rule for optimum number of hidden layers (Cabrera et al 1997).

Figure 7.1 Basic structure of a multi layer feed forward neural network

The ANN to be used for the generation of the optimal switching angles has a single input neuron fed by the modulation index, one hidden layer and N outputs where each output represents a notching angle. The following sub sections the algorithm to find the updated weights of input-hidden and hidden - output layers.
7.2.2 Steps Involved in Back Propagation Training

According to the delta rule, the instantaneous error is defined by (Satish Kumar 2004),

\[ E_k = D_k - \delta(Y_k) \]  

(7.3)

The mean square error \( \varepsilon \), is computed over the entire training set and is given by,

\[ \varepsilon = \frac{1}{Q} \sum_{k=1}^{Q} \varepsilon_k \]  

(7.4)

1. A pattern \( X_k \) is selected from the training set, and is presented to the network.
2. The activations and signals of the input, hidden and output neurons are computed in that sequence.
3. The error over the output neurons is computed by comparing the generated outputs with the desired outputs.
4. The error calculated in step 3 is used to compute the change in the hidden to output layer weights, and the change in input to hidden layer weights (including all bias weights), such that a global error measure gets reduced.
5. All weights of the network are updated in accordance with the changes computed in step 4.

Hidden to output layer weights:

\[ w_{hj}^{k+1} = w_{hj}^k + \Delta w_{hj}^k \]  

(7.5)

Input to hidden layer weights

\[ w_{ih}^{k+1} = w_{ih}^k + \Delta w_{ih}^k \]  

(7.6)

where \( \Delta w_{hj}^k \) and \( \Delta w_{ih}^k \) are the weight changes computed in step 4.
6. Steps 1 through 5 are repeated until the global error falls below a predefined threshold.

7.2.3 Computation of Neuronal Signals

The back propagation algorithm updates neuronal activations in the network as follows.

1. For the input layer:
\[
\delta(x_i^k) = x_i^k, i = 1, \ldots, n
\]  
where \( x_i^k \) is the \( i \)th component of the input vector \( X^k \) presented to the network.

2. For the hidden layer:
\[
z_h^k = \sum_{i=0}^{n} w_{ih}^k \delta(x_i^k) = \sum_{i=0}^{n} w_{ih}^k x_i^k, h = 1, \ldots, q
\]  
\[
\delta(z_h^k) = \frac{1}{1 + e^{-z_h^k}}, h = 1, \ldots, q
\]  
\[
\delta(z_0^k) = 1, \forall k
\]  
where \( w_{ih}^k \) are the biases of the hidden neurons, and \( \delta(z_0^k) \) is the hidden layer bias neuron signal which is once again independent of iteration index.

3. For the output layer:
\[
y_j^k = \sum_{h=0}^{q} w_{hj}^k \delta(z_h^k), j = 1, \ldots, p
\]  
\[
\delta(y_j^k) = \frac{1}{1 + e^{-y_j^k}}, j = 1, \ldots, p
\]  
where \( w_{hj}^k \) are the biases of the output neurons.
7.2.4 Computation of Errors

1. Error at output neurons:
   \[ \xi^k_j = (d^k_j - \delta(y^k_j))\delta'(y^k_j), \quad j = 1, \ldots, p \]  
   \[ \Delta w^k_{hj} = \eta \xi^k_j \delta(z^k_h), \quad h = 0, \ldots, q; \quad j = 1, \ldots, p \]  

2. Error at hidden neurons:
   \[ \xi^k_h = \left( \sum_{j=1}^{p} \xi^k_j w^k_{hj} \right) \delta'(z^k_h), \quad h = 1, \ldots, q \]  
   \[ \Delta w^k_{ih} = \eta \xi^k_h x^k_i, \quad i = 0, \ldots, n; \quad h = 1, \ldots, q \]

where \( \eta \) is the learning rate.

7.2.5 Updation of Weights

For updating the weights, the momentum value (\( \alpha \)) is taken into consideration.

1. updation of input-hidden layer weights:
   \[ w^{k+1}_{ih} = w^k_{ih} + \Delta w^k_{ih} + \alpha w^{k-1}_{ih}, \quad i = 0, \ldots, n; \quad h = 1, \ldots, q \]  

2. updation of hidden-output layer weights:
   \[ w^{k+1}_{hj} = w^k_{hj} + \Delta w^k_{hj} + \alpha w^{k-1}_{hj}, \quad h = 0, \ldots, q; \quad j = 1, \ldots, p \]

7.3 DIRECT SUPERVISED TRAINING FOR SELECTIVE HARMONIC ELIMINATION

In the direct training, the targets, which are required to find the notching angles, corresponding to different modulation indexes are obtained from the Cauchy’s relation. Thus, with the available targets, the net can be trained offline. Once when the net is trained to the desired level such that the error obtained is minimum, the weights are updated. Using these updated
weights, the required notching angles can be found. Back propagation training algorithm is most commonly used in feed forward ANN. When a set of input values are presented to the ANN, step by step calculations are made in the forward direction to drive the output pattern (Bouhali et al 2005). Squared difference between the net output and the desired net output for the set of input patterns is generated and this is minimized by gradient descent method altering the weights one at a time starting from the output layer. This is used as the training data and the net is directly trained.

But in indirect training the targets obtained by solving the nonlinear equations of the inverter output are not required. The equations (6.22), (6.23) and (6.24) are directly incorporated in the algorithm itself such that the right hand side of the equation is satisfied. But since these right hand side values are zeros, it is very difficult to exactly obtain these values. Only approximate values can be obtained which will lead to increased error compared to the direct training. Though the advantage of not requiring the solution of the nonlinear equations makes the indirect training comfortable, but the above mentioned disadvantage makes it complex. Thus direct training is employed for the problem of Selective Harmonic Elimination. The direct supervised training of ANN for selective harmonic elimination is shown in Figure 7.2.

![Figure 7.2 Direct supervised training of ANN for SHE](image)
7.3.1 Network Training for Selective Harmonic Elimination

For the elimination of first two odd harmonics, that is, third and fifth harmonics, the equation (6.21) is expanded for N=3 and the three simultaneous equations (6.23), (6.24) and (6.25) are obtained to find three notching angles are required. The prototype model architecture of the direct supervised training network employing back propagation algorithm is given in Figure 7.3. Table 7.1 represents the number of neurons or units in each layer of the network which is represented in Figure 7.3.

![Figure 7.3 Training network architecture for N=3](image)

<table>
<thead>
<tr>
<th>Layer</th>
<th>Number of neurons</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input layer</td>
<td>1 + bias</td>
</tr>
<tr>
<td>Hidden layer</td>
<td>2 + bias</td>
</tr>
<tr>
<td>Output layer</td>
<td>3</td>
</tr>
</tbody>
</table>

Table 7.1 Number of neurons in each layer of the network for N=3

A bias unit is added in input and hidden layers, which is not represented in the Figure 7.3. The Bias neuron in the input layer receives no external input but generates a ‘+1’ signal that feeds all bias connections of the
neurons of the hidden layer. Similarly, the bias neuron in the hidden layer generates a ‘+1’ signal that feeds all bias connections of the neurons of the output layer (Sathish Kumar 2004)

1. The training is carried out offline using the back propagation algorithm. Input is the modulation index for output voltage control and the output is the alpha angles for selective harmonic elimination (SHE) corresponding to the input training pattern.

2. Since the training is carried offline the training time is not a consideration, whereas accuracy is an important factor.

The steps involved in training the network are:

1. Firstly, the training pattern is decided. Here, various values of modulation indices from 0.4 to 1 in steps of 0.05 are taken as the training pattern. Thus totally, there are thirteen values in the training pattern.

2. For the various modulation indices taken as training pattern, the notching angles are obtained as targets, by solving the nonlinear equations.

3. Then the initial weights for both input and hidden layers including the bias are assumed.

4. With the available training pattern, targets and the initial weights, the network is trained by the back propagation algorithm explained in section 7.3.1.

5. Once when the error is less than the tolerance value, training is stopped.

6. Thus using the updated weights, the notching angles are obtained for any value of modulation index.
Figure 7.4 shows the flow chart of the training process. It is found that the results obtained are accurate which is discussed in the subsequent sections. It took more than twenty thousands epochs or iterations to train a complete training pattern. It is also obvious that, the time taken for finding the notching angles from the trained net is lesser than that of the numerical methods since training is offline.

![Flowchart of the training process](image-url)
7.4 SIMULATION AND HARDWARE RESULTS

The refined weights for the various values of N are obtained in training process. From these refined weights final notching angles are obtained from a simple matlab program for different values of N and M respectively. These notching angles are given to the simulink model of the single phase voltage source inverter of Figure 6.12 with same specifications for selective harmonic elimination. The simulation results are validated using the same hardware of Figure 6.14. In this section the selective harmonic elimination of the single phase voltage is carried out with the Computed PWM gate pulses for M=0.9 and the variation of fundamental RMS voltage for 0.1<M<1 for N=5.

7.4.1 Elimination of the First Two Odd Harmonics

For eliminating first two lower order harmonics three values of notch angles need to be calculated; one for fundamental component and the other two for the first two lower order harmonics 3rd and 5th. After training the net for about forty thousand epochs, the refined or updated weights for the net are obtained. Figure 7.5 represents the graph between error(e) and number of epochs.

![Figure 7.5 Graph between error(e) and number of epochs](image)
The updated weights are:

1. input-hidden layer weights:
   \[ W_{ih} = \begin{bmatrix} -1.54746300827386 & -14.08166606402045 \\ 1.78607231077035 & 11.36517358395824 \end{bmatrix} \]

2. hidden-output layer weights:
   \[ W_{hj} = \begin{bmatrix} -1.60936690773359 & -0.88867113806576 \\ -0.11185770000560 & -1.01114268367744 \\ 0.22520407676621 & -0.98307068323344 \\ -1.64551339484126 & -4.67857664312716 \\ -4.06162901990384 \end{bmatrix} \]

Then the final notching angles are obtained from these weights. Using the artificial neural networks discussed in the previous section, the final notch angles are found to be \( \alpha_1 = 18.5407^\circ \), \( \alpha_2 = 53.1499^\circ \) and \( \alpha_3 = 59.6258^\circ \), where \( \alpha_1 \), \( \alpha_2 \), \( \alpha_3 \) are the notch angles. The switching waveforms and output voltage waveforms and the harmonic spectrum of the output voltage of the simulation model are shown in Figures 7.6 and 7.7 respectively. The switching waveforms and output voltage waveforms and the harmonic spectrum of the output voltage of the hardware model are shown in Figure 7.8 and Figure 7.9 respectively. Figure 7.10 shows the comparative results of the individual harmonic distortions.
Figure 7.6 Switching waveforms and output voltage waveforms for N=3

Figure 7.7 Harmonic spectrum of the output voltage (for N=3)

Figure 7.8 Switching waveforms and output voltage waveforms of the hardware model for N=3
From the harmonic spectrum it can be concluded that the $3^{rd}$ and $5^{th}$ harmonics are eliminated and their magnitudes are very close to zero.

### 7.4.2 Elimination of the First Three Odd Harmonics

To eliminate the first three odd lower order harmonics, the following refined or updated weights for the net are obtained.
The updated weights are:

1. **input-hidden layer weights:**
   
   \[ \text{Wih} = \begin{bmatrix} 0.75750935968257 & -8.64663051688767 \\ -1.36390775742541 & 6.00405545433029 \end{bmatrix} \]

2. **hidden-output layer weights:**
   
   \[ \text{Whj} = \begin{bmatrix} -2.65474343431168 & -1.00366499596395 \\ -1.24910320184740 & 0.19765058710538 \\ 0.87237542110480 & -0.32536746211568 \\ 0.83282737475620 & -0.60609276128295 \\ -1.30677847977681 & -2.60202991130945 \\ -2.32523448661473 & -0.09375785858243 \end{bmatrix} \]

Then the final notching angles are obtained from these weights, using the artificial neural networks discussed in the previous section. The final notch angles are found to be \( \alpha_1 = 15.4116^\circ \), \( \alpha_2 = 40.8202^\circ \), \( \alpha_3 = 47.8394^\circ \) and \( \alpha_4 = 88.5629^\circ \), where \( \alpha_1, \alpha_2, \alpha_3 \) and \( \alpha_4 \) are the notch angles. The switching waveforms and output voltage waveforms and the harmonic spectrum of the output voltage of the simulation model are shown in Figures 7.11 and 7.12 respectively. The switching waveforms and output voltage waveforms and the harmonic spectrum of the output voltage of the hardware model are shown in Figures 7.13 and 7.14 respectively. Figure 7.15 shows the comparative results of the individual harmonic distortions.
Figure 7.11 Switching waveforms and output voltage waveforms for N=4

Figure 7.12 Harmonic spectrum of the output voltage (for N=4)

Figure 7.13 Switching waveforms and output voltage waveforms of the hardware model for N=4
From the harmonic spectrum it is found that the lower order harmonics i.e. 3rd, 5th and 7th are eliminated and their magnitudes are very close to zero.

### 7.4.3 Elimination of the First Four Odd Harmonics

In this case the first four odd harmonics i.e., 3rd, 5th, 7th and 9th are eliminated. The updated weights to find the notching angles are as follows.
1. input-hidden layer weights:
   \[ W_{ih} = \begin{bmatrix} 2.97267017310424 & -21.13989259067333 \\ -3.88865783705090 & 18.00933611773307 \end{bmatrix} \]

2. hidden-output layer weights:
   \[ W_{hj} = \begin{bmatrix} -2.65670174620098 & -1.47709598852908 \\ -1.36762721298317 & -0.42408435530661 \\ -0.50320239633687 & 0.34118918451468 \\ 0.03786216565584 & 0.37802220772611 \\ -0.08074322184961 & 0.30205762525535 \\ -1.53594300289831 & -2.73366784418547 \\ -2.31255468818479 & -5.23900999226246 \\ -4.97892976528136 \end{bmatrix} \]

Then the final notching angles are obtained from these weights, using the artificial neural networks discussed in the previous section. The final notch angles are found to be \( \alpha_1 = 13.1453^\circ \), \( \alpha_2 = 33.3056^\circ \), \( \alpha_3 = 40.2771^\circ \), \( \alpha_4 = 68.3135^\circ \) and \( \alpha_5 = 71.0824^\circ \). The switching waveforms are generated from these notching angles and applied to the inverter. The switching waveforms and output voltage waveforms and the harmonic spectrum of the output voltage of the simulation model are shown in Figures 7.16 and 7.17 respectively. The switching waveforms and output voltage waveforms and the harmonic spectrum of the output voltage of the hardware model are shown in Figures 7.18 and 7.19 respectively. Figure 7.20 shows the comparative results of the individual harmonic distortions. Based on the harmonic spectrum, it is found that the lower order harmonics i.e. \( 3^{rd} \), \( 5^{th} \), \( 7^{th} \), and \( 9^{th} \) are eliminated.
Figure 7.16 Switching waveforms and output voltage waveforms for N=5

Figure 7.17 Harmonic spectrum of the output voltage (for N=5)

Figure 7.18 Switching waveforms and output voltage waveforms of the hardware model for N=5
7.4.4 Elimination of the First Five Odd Harmonics

To eliminate the first five odd harmonics totally six notching angles are needed. To obtain the notching angles the following updated weights are found.

1. input-hidden layer weights:

\[ W_{ih} = [1.22495549427263 -12.32697541366884 -1.48659766195548 9.38784689049154] \]
2. hidden-output layer weights:

\[ \text{Whj} = [-2.9749042245129 \ -1.64638653890371 \\
-1.72738584758761 \ -0.63328612377440 \\
-0.92865954451643 \ 0.17339924804372 \\
0.65151131116425 \ -0.02314055732072 \\
0.69712606171701 \ -0.19371028151658 \\
0.59941375592692 \ -0.40894610515970 \\
-1.00729474645858 \ -1.68569103782279 \\
-1.40116900759532 \ -2.66149234772621 \\
-2.44064380264878 \ 0.01093666959411] \]

Then the final notching angles are obtained from these weights, using the artificial neural networks discussed in the previous section. The final notch angles are found to be \( \alpha_1 = 11.4638^\circ \), \( \alpha_2 = 28.0046^\circ \), \( \alpha_3 = 34.8448^\circ \), \( \alpha_4 = 56.6305^\circ \), \( \alpha_5 = 59.9120^\circ \) and \( \alpha_6 = 89.1309^\circ \). The switching waveforms are generated from these notching angles and applied to the inverter. The switching waveforms and output voltage waveforms and the harmonic spectrum of the output voltage of the simulation model are shown in Figures 7.21 and 7.22 respectively. The switching waveforms and output voltage waveforms and the harmonic spectrum of the output voltage of the hardware model are shown in Figures 7.23 and 7.24 respectively. Figure 7.25 shows the comparative results of the individual harmonic distortions.

Figure 7.21 Switching waveforms and output voltage waveforms for N=6
Figure 7.22 Harmonic spectrum of the output voltage (for N=6)

Figure 7.23 Switching waveforms and output voltage waveforms of the hardware model for N=6

Figure 7.24 Harmonic spectrum of the output voltage of the hardware model (N=6)
Based on the harmonic spectrum, it is found that the lower order harmonics i.e. 3\textsuperscript{rd}, 5\textsuperscript{th}, 7\textsuperscript{th}, 9\textsuperscript{th} and 11\textsuperscript{th} are eliminated.

### 7.4.5 Elimination of the First Six Odd Harmonics

The first six odd harmonics can be eliminated by finding seven notching angles. To obtain the notching angles the following updated weights are found.

1. input-hidden layer weights:
   
   \[ W_{ih} = \begin{bmatrix} 3.01699359299495 & -27.27354336632342 \\ -3.30406669345851 & 23.87897883847024 \end{bmatrix} \]

2. hidden-output layer weights:
   
   \[ W_{hj} = \begin{bmatrix} -2.98775212654163 & -1.88162769899010 \\ -1.76583773717956 & -0.99726948502866 \\ -1.07921627408571 & -0.27156625553306 \\ -0.42454221754194 & 0.34337453426347 \\ 0.05921006015437 & 0.37618025962423 \\ 0.04168557401864 & 0.38655988051121 \end{bmatrix} \]
Then the final notching angles are obtained from these weights, using the artificial neural networks discussed in the previous section. The final notch angles are found to be $\alpha_1=10.1643^\circ$, $\alpha_2=24.2911^\circ$, $\alpha_3=30.7818^\circ$, $\alpha_4=48.9271^\circ$, $\alpha_5=52.4155^\circ$, $\alpha_6=74.9740^\circ$ and $\alpha_7=76.6068^\circ$. The switching waveforms are generated from these notching angles and applied to the inverter. The switching waveforms and output voltage waveforms and the harmonic spectrum of the output voltage of the simulation model are shown in Figures 7.26 and 7.27 respectively. The switching waveforms and output voltage waveforms and the harmonic spectrum of the output voltage of the hardware model are shown in Figures 7.28 and 7.29 respectively. Figure 7.30 shows the comparative results of the individual harmonic distortions. Based on the harmonic spectrum, it is found that the lower order harmonics i.e. 3$^{rd}$, 5$^{th}$, 7$^{th}$, 9$^{th}$, 11$^{th}$ and 13$^{th}$ are eliminated.

![Switching waveforms and output voltage waveforms for N=7](image)

Figure 7.26 Switching waveforms and output voltage waveforms for N=7
Figure 7.27 Harmonic spectrum of the output voltage (for N=7)

Figure 7.28 Switching waveforms and output voltage waveforms of the hardware model for N=7

Figure 7.29 Harmonic spectrum of the output voltage of the hardware model (N=7)
7.4.6 Elimination of the First Seven Odd Harmonics

In this case the 3\textsuperscript{rd}, 5\textsuperscript{th}, 7\textsuperscript{th}, 9\textsuperscript{th}, 11\textsuperscript{th}, 13\textsuperscript{th} and 15\textsuperscript{th} harmonics are eliminated from eight notching angles from the following updated weights.

1. input-hidden layer weights:
   \[
   W_{ih} = \begin{bmatrix}
   2.22977004029490 & -15.70352528676553 \\
   -3.01252596179895 & 12.42957061085575
   \end{bmatrix}
   \]

2. hidden-output layer weights:
   \[
   W_{hj} = \begin{bmatrix}
   -3.01910404353325 & -1.99484593954602 \\
   -1.80454236192511 & -1.13850156852038 \\
   -1.14088946584883 & -0.50650188697408 \\
   -0.58003719253263 & 0.04657792444291 \\
   0.26740637380996 & 0.01669884622631 \\
   0.28733238109629 & -0.00312073687214 \\
   0.28918167373416 & -0.07818360922939 \\
   0.23326795189576 & -0.15875879964441 \\
   -1.08764768209626 & -1.55143630106666 \\
   -1.34879765264715 & -2.07379081249119 \\
   -1.89498509910154 & -3.10281141171952 \\
   -2.99253960411752 & 0.03333225229689
   \end{bmatrix}
   \]
Then the final notching angles are obtained from these weights, using the artificial neural networks discussed in the previous section. The final notch angles are found to be \( \alpha_1=9.1352^\circ \), \( \alpha_2=21.3579^\circ \), \( \alpha_3=27.5838^\circ \), \( \alpha_4=42.8915^\circ \), \( \alpha_5=46.6285^\circ \), \( \alpha_6=65.0274^\circ \), \( \alpha_7=66.9751^\circ \) and \( \alpha_8=89.3839^\circ \). The switching waveforms are generated from these notching angles and applied to the inverter. The switching waveforms and output voltage waveforms and the harmonic spectrum of the output voltage of the simulation model are shown in Figures 7.31 and 7.32 respectively. The switching waveforms and output voltage waveforms and the harmonic spectrum of the output voltage of the hardware model are shown in Figures 7.33 and 7.34 respectively. Figure 7.35 shows the comparative results of the individual harmonic distortions.

Figure 7.31 Switching waveforms and output voltage waveforms for N=8

Figure 7.32 Harmonic spectrum of the output voltage (for N=8)
Figure 7.33  Switching waveforms and output voltage waveforms of the hardware model for N=8

Figure 7.34  Harmonic spectrum of the output voltage of the hardware model (N=8)

Figure 7.35 Comparative results of % harmonic distortion (N=8)
From the harmonic spectrum, it is found that the lower order harmonics i.e. 3\(^{rd}\), 5\(^{th}\), 7\(^{th}\), 9\(^{th}\), 11\(^{th}\), 13\(^{th}\) and 15\(^{th}\) are eliminated.

### 7.4.7 Elimination of the First Eight Odd Harmonics

In this case totally nine notching angles are found to eliminate 3\(^{rd}\), 5\(^{th}\), 7\(^{th}\), 9\(^{th}\), 11\(^{th}\), 13\(^{th}\), 15\(^{th}\) and 17\(^{th}\) order harmonics from the following updated weights.

1. **input-hidden layer weights:**
   
   \[ W_{ih} = \begin{bmatrix} 3.15108026819347 & -30.41734926527383 \\ -3.19088457586424 & 27.23449316022440 \end{bmatrix} \]

2. **hidden-output layer weights:**
   
   \[ W_{hj} = \begin{bmatrix} -3.20588466347427 & -2.15954324186276 \\ -2.01371913070697 & -1.33362089912018 \\ -1.38362580443414 & -0.75316033184140 \\ -0.88039309570138 & -0.1840842199762 \\ -0.34657631538230 & 0.31038440561600 \\ 0.05810117374514 & 0.33540410756490 \\ 0.05617703978108 & 0.3521998204666 \\ 0.02763670617688 & 0.34111893711712 \\ -0.10453178874276 & 0.22305848794599 \\ -0.70115663140867 & -1.23700863514976 \\ -0.86624833605258 & -1.58936037889529 \\ -1.21965234381772 & -2.32317301882868 \\ -1.96961455442363 & -3.96107865070664 \\ -3.67485888596124 \end{bmatrix} \]

Then the final notching angles are obtained from these weights, using the artificial neural networks discussed in the previous section. The
final notch angles are found to be $\alpha_1=8.2876^\circ$, $\alpha_2=19.1218^\circ$, $\alpha_3=24.9881^\circ$, $\alpha_4=38.3698^\circ$, $\alpha_5=42.1025^\circ$, $\alpha_6=57.9940^\circ$, $\alpha_7=60.0652^\circ$, $\alpha_8=78.6153^\circ$ and $\alpha_9=79.7154^\circ$. The switching waveforms are generated from these notching angles and applied to the inverter. The switching waveforms and output voltage waveforms and the harmonic spectrum of the output voltage of the simulation model are shown in Figures 7.36 and 7.37 respectively. The switching waveforms and output voltage waveforms and the harmonic spectrum of the output voltage of the hardware model are shown in Figures 7.38 and 7.39 respectively. Figure 7.40 shows the comparative results of the individual harmonic distortions.

![Switching waveforms and output voltage waveforms for N=9](image1)

**Figure 7.36** Switching waveforms and output voltage waveforms for N=9

![Harmonic spectrum of the output voltage (for N=9)](image2)

**Figure 7.37** Harmonic spectrum of the output voltage (for N=9)
Figure 7.38  Switching waveforms and output voltage waveforms of the hardware model for N=9

Figure 7.39  Harmonic spectrum of the output voltage of the hardware model (N=9)

Figure 7.40  Comparative results of % harmonic distortion (N=9)
From the harmonic spectrum, it is found that the lower order harmonics i.e., 3\textsuperscript{rd}, 5\textsuperscript{th}, 7\textsuperscript{th}, 9\textsuperscript{th}, 11\textsuperscript{th}, 13\textsuperscript{th}, 15\textsuperscript{th} and 17\textsuperscript{th} are eliminated.

7.4.8 Variation in the fundamental RMS voltage for different modulation indexes

The proposed simulated and hardware model can be tested with various modulation index values varied from 0.1 to 1 for 3<N<9. Whenever the modulation index varies for a particular value of N correspondingly the fundamental RMS is also varied. The variation of the fundamental RMS voltage for N=5 with modulation index varies from 0.1 to 1 is shown in Figures 7.41(a-j) and 7.42(a-j) for both simulated and hardware models. The Table 7.2 shows the magnitude of the fundamental RMS voltage of the simulated results (both algebraic approach and ANN approach) and hardware. Figure 7.43 represents the comparison of simulation and hardware results for fundamental RMS voltage. The simulated results are very close to the hardware results.

Table 7.2 Magnitude of the fundamental RMS voltage for N=5 with 0.1<M<1.0

<table>
<thead>
<tr>
<th>Sl. No.</th>
<th>Modulation Index</th>
<th>RMS value of the fundamental component in volts</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Simulation (Algebraic Approach)</td>
<td>Simulation (ANN Approach)</td>
<td>Hardware</td>
</tr>
<tr>
<td>1.</td>
<td>1.0</td>
<td>227.7</td>
<td>228.0</td>
<td>232.7</td>
</tr>
<tr>
<td>2.</td>
<td>0.9</td>
<td>205.2</td>
<td>206.6</td>
<td>210.8</td>
</tr>
<tr>
<td>3.</td>
<td>0.8</td>
<td>182.7</td>
<td>183.0</td>
<td>188.3</td>
</tr>
<tr>
<td>4.</td>
<td>0.7</td>
<td>159.2</td>
<td>159.2</td>
<td>165.3</td>
</tr>
<tr>
<td>5.</td>
<td>0.6</td>
<td>136.9</td>
<td>137.0</td>
<td>140.2</td>
</tr>
<tr>
<td>6.</td>
<td>0.5</td>
<td>113.6</td>
<td>114.0</td>
<td>115.5</td>
</tr>
<tr>
<td>7.</td>
<td>0.4</td>
<td>94.7</td>
<td>95.4</td>
<td>94.9</td>
</tr>
<tr>
<td>8.</td>
<td>0.3</td>
<td>69.7</td>
<td>68.9</td>
<td>72.6</td>
</tr>
<tr>
<td>9.</td>
<td>0.2</td>
<td>45.1</td>
<td>45.7</td>
<td>47.5</td>
</tr>
<tr>
<td>10.</td>
<td>0.1</td>
<td>22.3</td>
<td>21.7</td>
<td>28.0</td>
</tr>
</tbody>
</table>
Figure 7.41 (Continued)
Figure 7.41  Harmonic spectrum with the variation in fundamental RMS voltage for N=5 with 0.1<M<1 (Simulation)
Figure 7.42 (Continued)
Figure 7.42  Harmonic spectrum with the variation in fundamental RMS voltage for N=5 with 0.1<M<1(Hardware Model)
The implementation of the selective harmonic elimination using neural network has many advantages over the conventional techniques. In which complex iterations involved in Newton Raphson’s method are eliminated. The training takes a long time; it is not a disadvantage since the training is carried offline. Using the Back propagation algorithm the net can be trained to give sufficiently accurate alpha angles. The conventional lookup table method involves larger memory which is not required in case of neural net. After the weights of the neural net are determined, the net can be implemented online. Parallel neuron units enable parallel processing reducing the computation time significantly in real time. Figure 7.44 represents the comparison of the computation time of the program developed for algebraic approach and ANN approach for $3 < N < 9$. From this it found that the computation time is very much reduced in ANN approach.
Thus for the elimination of the harmonics in the inverter circuits, the back propagation algorithm is preferred to the other algorithms of training the neural networks. The notching angles thus obtained after training the network using the direct training method are satisfactory which could be seen clearly in the results produced.