

Fault Diagnosis and Reconfiguration of Multilevel Inverter Switch Failure-A Performance Perspective

T.G. Manjunath¹, Ashok Kusagur²

¹Department of Electrical and Electronics Engineering, Sai Vidya Institute of Technology, Bangalore, India

²Department of Electrical and Electronics Engineering, UBDT College of Engineering, Davangere, India

Article Info

Article history:

Received Jul 7, 2016

Revised Sep 17, 2016

Accepted Nov 1, 2016

Keyword:

Artificial neural network
Cascaded multilevel inverter
Fault detection and reconfiguration
Genetic algorithm
Modified genetic algorithm
Parameter optimization

ABSTRACT

Multilevel Inverters (MLI) gains importance in Distribution systems, Electrical Drive systems, HVDC systems and many more applications. As Multilevel Inverters comprises of number of power switches the fault diagnosis of MLI becomes tedious. This paper is an attempt to develop and analyze the fault diagnosis method that utilizes Artificial Neural Network to get it trained with the fault situations. A performance analysis of Genetic Algorithm (GA) and the Modified Genetic Algorithm (MGA), which optimizes the Artificial Neural Network (ANN) that trains itself on the fault detection, and reconfiguration of the Cascaded Multilevel Inverters (CMLI) is attempted. The Total Harmonic Distortion (THD) occurring due to switch failures or driver failures occurring in the CMLI is considered for this comparative analysis. Elapsed time of recovery, Mean Square Error (MSE) and the computational budgets of ANN are the performance parameters considered in this comparative analysis. Optimization is involved in the process of updating the weight and the bias values in the ANN network. Matlab based simulation is carried out and the results are obtained and tabulated for the performance evaluation. It was observed that Modified Genetic Algorithm performed better than the Genetic Algorithm while optimizing the ANN training.

Copyright © 2016 Institute of Advanced Engineering and Science.
All rights reserved.

Corresponding Author:

T.G. Manjunath,
Department of Electrical and Electronics Engineering,
Sai Vidya Institute of Technology,
Bangalore, India.
Email: tgmjath@gmail.com

1. INTRODUCTION

Different multilevel inverter (MLI) topologies are explored and are reviewed in [1]. The cascaded multilevel inverter with reduced number of switches are discussed [2]. The CMLI eliminates the bulk transformers available in the multipulse inverters, clamping diode in the Diode Clamped MLI and the capacitors in Flying Capacitor MLI [3]. CMLI gets rid of the capacitor voltage balancing method, which is a major challenge in the Flying Capacitor Multilevel Inverter as; it needs very complex voltage balancing techniques [4]. Modular topology of the CMLI facilitates the topology for the location of fault and the reconfiguration applied in a simpler manner [5]. Whereas the Diode Clamped MLI and the Flying Capacitor MLI has switches continuously connected according to the number of voltage levels it can deliver, which introduces complexity while fault location and reconfiguration. Application of CMLI ranges from STATCOM [6] to Active Power Filter, which demands fault free operation since it involves power quality solutions [7]. The fault diagnosis and the reconfiguration of the MLI by the use of Artificial Neural Network as the fault diagnosis tool was developed which exhibited 95% accuracy in diagnosis of the faults was tabulated [6]. The reconfiguration method proposed in [8] cleared both the open circuit and the closed circuit faults in six periodic cycles. In order to develop the fault diagnosis the development of a mathematical model

is avoided due to the use of AI method that would greatly reduce the engineering and development time. The reliability of the Adjustable Speed Drives (ASD) powered by CMLI is improved by introducing the redundant modules, which would work when any of the modules fails [9]. A novel technique called the “neutral shift” was also introduced which adopted the idea that compensated the failure in one module with the redundant module of the other module. A 99.99% drive reliability was obtained in the implementation [9]. This paper uses the redundant module replacement method for fault diagnosis in the CMLI. The THD obtained from the MLI is the source of fault detection in the method used. ANN gets trained by getting the pair of the THD values for the corresponding fault switch position. The speed of training the ANN is concentrated in this paper and steps are taken to improve it. In order to optimize the training time of the ANN optimization techniques like the Genetic Algorithm and the Modified Genetic Algorithm is implemented on the training of the ANN. The weight values that are used for each node and the bias values are optimized in order to train the ANN faster than the traditional training method. Considering the Mean Square Error (MSE) as the objective function to be minimized, the module number where fault occurs in the CMLI, which has to be predicted, is the output, which would be trained with THD as the input.

This paper is organized as, Section II with the ANN applied on MLI fault detection, and Section III explains the Genetic Algorithm and Modified Genetic Algorithm, Section IV deals about the results and discussion of the implementation with comparative analysis of both GA and MGA.

2. FAULT DIAGNOSIS OF MULTILEVEL INVERTER USING ARTIFICIAL NEURAL NETWORK

2.1. Training Parameter Selection for ANN

The selection of the training parameter is the prime step in any ANN implementation. In this implementation the module in which fault occurs is taken as the output and the THD is considered as the input parameter. The near sinusoidal shape of the multilevel inverter output would be lost if there is a fault occurs in the multilevel inverter. The THD is calculated on the output waveform by applying the Fast Fourier Transform. Due to the fault occurrence the THD value would have been changed. The change in the THD value is influenced by the position where the fault is occurring in the multilevel inverter.

2.2. Faults in MLI

This fault diagnosis system comprises of training and testing steps. During the training step the four main steps that is followed are 1) THD calculation for different switch failure 2) ANN training. The testing phase comprises of 1) ANN classification for fault prediction 2) Reconfiguration

THD calculation can also be called as feature extraction where the Fast Fourier Transform (FFT) applied on the MLI output voltage is used for the THD calculation. The application of FFT on the output voltage waveform of the CMLI provides the amplitude of the fundamental and the harmonic voltage for the different harmonic frequency. The most commonly occurring fault in the CMLI is the open circuit (OC) faults, whereas the short circuit faults are removed by the use of fast acting switch that would be used to remove the leg from the healthy part of the MLI [10], converting it to be an open circuit fault. As the short circuit fault is also converted to open circuit fault the number of the training data to the ANN is reduced. As pulse width variation also would introduce the change in the THD value the ANN can be used for the prediction of PWM change.

2.3. ANN Training

The THD values of the CMLI for the corresponding failure of the switches are tabulated. This pair of observation, which comprises of the position of the switch and the THD values, is used for training the ANN. These THD values are trained as the input and the switch number is given as the output. The fault detection and the reconfiguration is a non-linear problem to be solved, as there is no predetermined time when the fault can occur in the CMLI. ANN acts as the solution for this non-linear problem. Back Propagation Network (BPN) algorithm is used for applying the ANN on this non-linear problem. After the BPN algorithm is applied on the ANN a trained ANN will be ready for fault diagnosis testing. The procedure for this method is included in the implementation section. The Pseudo code for the BPN algorithm for the fault diagnosis on the CMLI is given as follows,

Pseudo Code:

- a. A feed forward Neural Network along with the proper number of input, output and hidden nodes is generated.
- b. Arbitrary Transfer function is chosen for the nodes.
- c. Weight initialisation for each node is provided and a tolerance value, which is very low, is chosen.

- d. The training set of input and output data is taken from the MLI. The input being the THD and the output the leg position in the MLI.
- e. The input is traversed from the input layer to the output layer and the error is found between the actual output and the desired output.
- f. The error is compared with the tolerance value, if the error is not nearer to the tolerance then the following steps has to repeat
- g. The error is propagated to each previous node by calculating the weight correction term and the bias correction term for each node.
- h. The steps 5 to 7 are repeated until the error is becoming less than or equal to the tolerance value considered.
- i. The steps 2 to 8 are repeated for all the input, output pairs.

After this training step a trained network will be ready for testing through which any THD value from the MLI can be provided as the input and the fault position can be found.

3. PARAMETER OPTIMIZATION OF ANN USING GA AND MGA

Parameter optimization is a heuristic problem that would help improve the performance of any decision-making problem like the Artificial Neural Network (ANN). Parameter optimization could be used to decide the weight values of the nodes in the ANN or the bias values in order to improve the learning by reducing the time required for the network to be trained. The advantage of parameter optimization is (1) no experience how to choose the parameter setting of the algorithm is needed, (2) a comparison with other algorithms is needed, (3) a algorithm has to be applied on a complex optimization problem [11].

3.1. Optimising ANN using GA

As the ANN is defined and constructed in the previous section training time of the ANN has to be improved by the use of GA and MGA. This subsection would deal about GA based parameter optimisation of the ANN. The parameter thus chosen are node weight values and the bias values. As two parameters are taken for optimisation this problem becomes the multi variable heuristic problem. The pseudo code for this implementation is as explained below.

Pseudo Code:

1. With the ANN model created get the initial weight matrix of all the nodes.
2. Unwrap weight and the Bias matrices into a single array each.
3. Consider these arrays as the first chromosomes
4. Randomly generate the rest of the chromosomes for a particular population size, which is selected intuitively.
5. For every chromosome generated in the previous step reform that into the weight and the Bias matrices.
6. Apply these matrices on the ANN and obtain the error and the MSE.
7. Save the minimum MSE if obtained in a array
8. Generate new population of chromosomes using the following steps 9 to 11
9. Select two parents from the previously generated chromosomes.
10. Apply crossover probability on these parents to get a new offspring.
11. Apply mutation probability to obtain the new children in the position of the parent chromosome.
12. Repeat the steps 5 to 7 until the number of iteration is completed.
13. The array of saved MSE is taken and the weight and bias values corresponding to the minimum MSE are selected.
14. The selected weight and bias values are provided as the weight and bias matrices to the ANN and the results are updated.
15. ANN performance can be evaluated using the selected weight and the bias values.
16. The parent selection for the population generation is carried using the Roulette Selection method as defined below,

$$S_i = \frac{Y_i}{Sum(Y)}$$

$$S1=0$$

$$J=1$$

where Y is the set of all fitness values, Y_i is the i th fitness value. Let $S1$ be some random value and $S2$ and J are variables. The Roulette selection algorithm can be defined as follows, while $S2 > S1$ the following steps has to follow

Find $S2 = S2 + S(J)$

Increment J

End

Once the while loop is becoming false the J value is taken and the population at the J th position is selected as the parent.

The arithmetic crossover is used to create the new offspring from the two parents selected using the Roulette selection. Two off springs are generated using the following formulae,

$$\text{Offspring1} = a * \text{parent1} + (1 - a) * \text{parent2}$$

$$\text{Offspring2} = (1 - a) * \text{parent1} + a * \text{parent2}$$

where, a is the random weighting factor. In order to maintain genetic diversity between the off springs the mutation operation is carried out. For the calculation of the mutation the maximum and the minimum weight values are taken from the a values and it is mentioned in the below equations as 'maxweight' and 'minweight'. The mutation operation is defined in the following formulae (1),

$$\phi = 1 - 2 * \text{randnum} \quad (1)$$

where, randnum is the random value.

Using the above value the inverse error function is calculated and utilised in the following formulae,

$$Z = \text{inverseerrorfunction}(\phi) * 2^{0.5} \quad (2)$$

The offspring is calculated by using

$$\text{Offspring} = Z * \sigma + \text{parent}$$

$$\sigma = \frac{(\text{maxweight} - \text{minweight})}{10}$$

σ reduces its value for every iteration $\sigma = \frac{\sigma}{1.05}$.

3.2. Modified Genetic Algorithm

The simple genetic algorithm accounts for only one population in each generation and the genetic operations are done on that generation. It becomes a time consuming process to evaluate the function as the simple GA would consume more iterations. In order to make this convergence faster the algorithm must be modified in such a way that the evaluation of the function must be carried out parallelly instead of individually. The multicore processors available in the industry would help in implementing this simultaneous implementations. The processors can work in master and slave approach in which, master stores a single population and the other processor evaluate the individuals. In the literature [12] it has been proposed that the population must be splitted into many subpopulation to keep diversity in the course of the optimiation process. The process of migration, which helps in sharing a single individual with better fitness in many subpopulation is carried out for which after crossover and mutation would generate the solution space that is not must explored. The amount of diversity in the sub population is proportional to the rate of migration.

3.3. Dynamic Parameter Design in MGA

The random chromosome generation in the beginning of the GA algorithm decides the quality and efficiency of the solution that is obtained. The search quality will be inefficient as the dynamism is choosing the parameter is missing. The crossover and the mutation probability is varied dynamically in the MGA in

order introduce the dynamic characteristics in choosing the parameters. At the beginning of the evolution process the bigger crossover and mutation probability is applied to get the offsprings but as the convergence forwards, the crossover and mutation probability will become of smaller range. The steps involved in the dynamic parameter design of the MGA are as follows,

1. The problem with the parameters is coded in the form of the string. Binary coding method is used to transfer the parameter from the problem space to the coding space. The length of the code is determined by the following formula (3) [12].

$$L = \frac{\ln\left(\frac{x_{\max} - x_{\min}}{\sigma} + 1\right)}{\ln 2} \quad (3)$$

where x_{\max} and x_{\min} are the maximum and minimum values of the independent variables in the objective function and σ is the precision required.

2. Individual fitness function are calculated by the objective function in the simple GA, but in this MGA it is defined by the following formulae (4)

$$fitness(i) = \frac{2(p_i - 1)}{Q - 1} \quad (4)$$

where, p_i is the order position of the sorted individual i , and Q is the total number of individuals.

3. Let T be the total number of generations, M be the total number of sub population, N the size of each sub population, rate of migration r , probability of selection denoted by s , crossover and mutation probability as c and m respectively. The recommended values of M and N for MGA are $3 \leq M \leq 8$, $N = 10(L + n)$ where n is the number of variables.
4. The selection probability for the i th individual is given as

$$s_i = \frac{fitness(i)}{\sum fitness(i)} \quad (5)$$

where, selection probability S_i is kept constant to make the crossover and mutation probability to vary dynamically. The crossover and the mutation probability would follow the rule of bigger values initially and smaller values later. The crossover and mutation probability are denoted as in (6) and (7),

$$m_j = a_j \cdot \sin\left(\frac{\pi}{2} \cdot \frac{T - t}{T}\right) \quad (6)$$

$$c_j = b_j \cdot \sin\left(\frac{\pi}{2} \cdot \frac{T_j - t}{T_j}\right) \quad (7)$$

where, t is the number of present generation, a_j , b_j is the initial values of m_j and c_j respectively for the j th population, T_j is the scaling factor, which would be larger than or equal to T . The values of m_j and c_j will be initiated with a_j and b_j but it would decrease in the iterations that follows.

5. Initial population is created using the random function as in GA.
6. Individual fitness is evaluated using the population.
7. The individuals with better fitness is supplied to the sub population $i=1 \dots N-1$, and the individual with worst fitness is substituted in the $i+1$ th sub population.
8. Crossover and mutation for the adjusted function is calculated and the genetic operators are applied to develop the new generation.
9. Check whether the current generation number t is less than the total number of generation, if yes stop the iteration, else continue from step 5.

4. FAULT DIAGNOSIS IMPLEMENTATION

The implementation of both the GA and the MGA for optimising the ANN's weight and bias values is carried out on a three phase seven level MLI as shown in Figure 1. It shows the three phase CMLI to be connected to the 5HP motor drive.

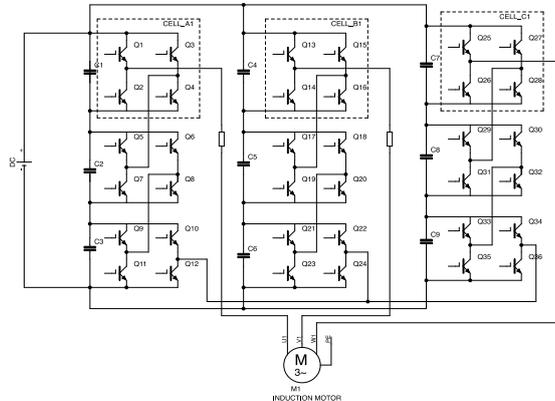


Figure 1. Three Phase CMLI Drive

4.1. Cascaded Multilevel Inverter Drive

For testing purposes a seven-level cascaded multi-level inverter driving induction motor is used. The closed switch faults and open switch faults are applied to the single phase of the three-phase inverter for simplicity. Due to the application of fault the output voltage waveform changes its shape due to the unavailability of some voltage levels. This waveform analysed with total harmonic distortion (THD). Each switch fault creates different waveforms due to the different levels. Each faults has different THD values. Figures portrayed below shows the waveforms due to the fault in each switch of a single cell and its THD values. Using the THD data, identification of switch fault can be done by training the neural network (NN).

4.2. Cascaded Multilevel Inverter Drive

For testing purposes a seven-level cascaded multi-level inverter driving induction motor is used. The closed switch faults and open switch faults are applied to the single phase of the three-phase inverter for simplicity. Due to the application of fault the output voltage waveform changes its shape due to the unavailability of some voltage levels. This waveform analysed with total harmonic distortion (THD). Each switch fault creates different waveforms due to the different levels. Each faults has different THD values. Figures portrayed below shows the waveforms due to the fault in each switch of a single cell and its THD values. Using the THD data, identification of switch fault can be done by training the neural network (NN). The three phase CMLI connected to the induction motor drive which has the extra redundant leg that would act when there is a fault occurring in any of the leg in the CMLI. The circuit gets closed through the redundant leg whenever the fault occurs in any of the leg in the CMLI.

4.3. GA and MGA Implementation

The parameter selection algorithm that is explained in the previous sections are developed using the Neural Network toolbox in Matlab and M-files. M-files are used to develop the GA and MGA optimisation on the parameter selection of ANN. The Matlab/Simulink model of the three phase CMLI is developed for a 5HP induction motor drive for an open loop operation. The OC faults are introduced in some of the legs in the MLI and the THD of the output voltage waveform is calculated for different switch failures and tabulated. The THD of the CMLI is given as the input to the ANN and the fault position is given as the output for training. In order to make the ANN train quicker both the GA and MGA is applied to optimize the weight and the bias values of the ANN. The overall block diagram of the implementation is as given in the Figure 3.

The three phase CMLI is connected to induction motor. The output of the MLI is applied with FFT and THD is calculated. The THD will be given to the trained ANN, which finds the fault position in the MLI, and the reconfiguration is applied using the redundant leg.

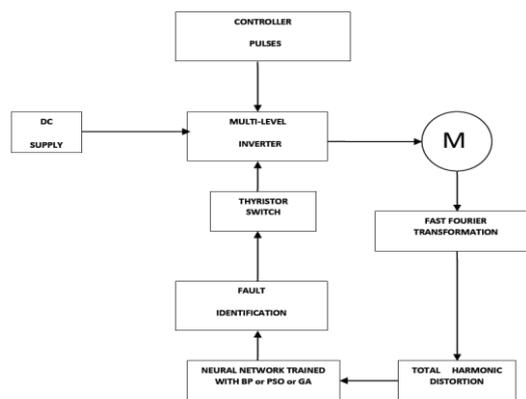


Figure 3. Overall Parameter Selection Implementation

5. RESULTS AND DISCUSSION

The test setup is shown as a block diagram. The MLID is operated for 1 sec. within that 0.01 sec, 0.25 sec, 0.5 sec and 0.75 sec are induced with a switch fault Q1, Q2, Q3 and Q4 (open or closed switch) respectively. With and without fault is shown in the below figures.

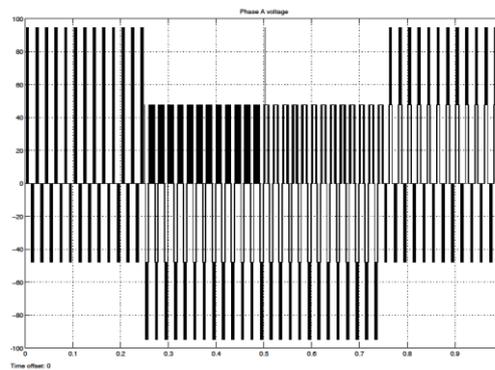


Figure 4. Phase A Voltage Waveform on the fault occurrence at 0.01 sec, 0.25 sec, 0.5 and 0.75 sec on Q1, Q2, Q3 and Q4 in "Voltage" vs "Time"

From the Figure 5 due to fault in switches change in THD values are shown. These values are given as the input to the neural network and output is taken as zero and 1 as binaries. Figure 5 shows the pattern of the THD for the fault applied in cell A1 for the simplicity. Due to the periodically applied fault change in waveform of voltage make the pattern in THD calculation. This pattern is feed to the neural network to train for this pattern. Then if any of this fault occurs it can easily classify the fault phase and cell. Then it can perform the remedial action. Figure 6 depicts the multilevel inverter voltage output for the CMLI considered in this paper.

From Table 1 it can be seen from the table that MGA algorithm gives lesser value of mean square error and it converges faster due to its large search space. Migration calculation makes the search more optimized. The number of epochs that was carried out due to the MGA is less than that of the GA and also the mean square error was also very much reduced as compared to the genetic algorithm. The number of cycles that took to clear the fault was 14 cycles as seen in the Figure 7.

The parameters that are used for the Genetic Algorithm implementation is

Minimum error threshold= 0.04.

Population size= 100.

Maximum no. of epochs=1700.

Fitness function= Mean squared error.
 No. of layers=2.
 No. of neurons in input layer= 4.
 Number of epochs trained= 78.
 Mean squared error: **Mse** = 0.008744818256473

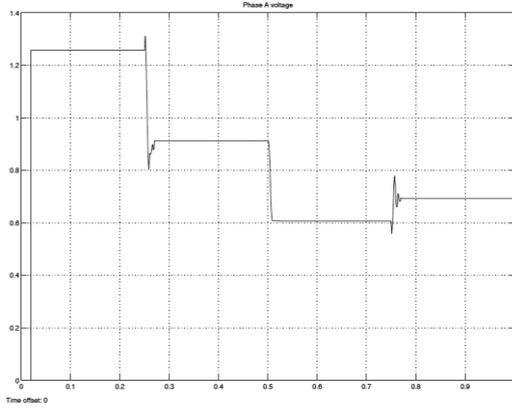


Figure 5. THD Values for the Switch Fault S1, S2, S3 and S4 in “THD” vs “Time”

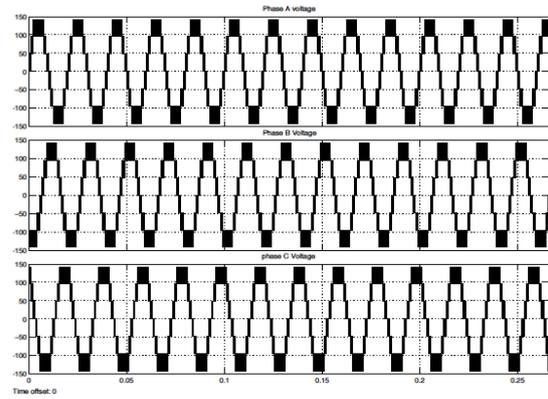


Figure 6. Without any Fault in the MLID System in “Voltage” vs “Time”

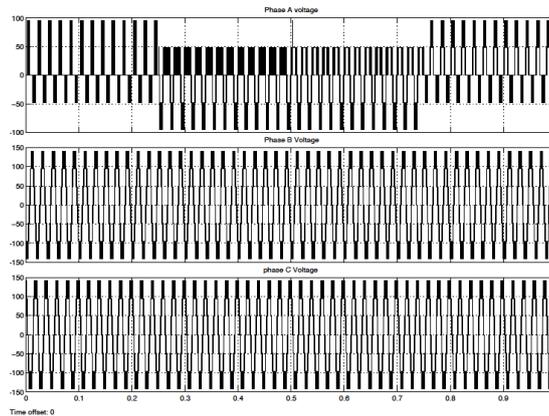


Figure 7. Fault at the Phase A at Cell A1; Q1, Q2, Q3 and Q4 Fault as in the Figure 4. “Voltage” vs “Time”

Table 1. Mean Square Error and Number of Epochs Analysis

SL.No.	Parameters	Genetic Algorithm	Modified Genetic Algorithm
1	Mean Square Error	0.008744818256473	77.0977×10^{-4}
2	Number of Epochs	78	64

The Table 2 tabulate the MSE obtained for every iteration until the 20th iteration.
 The parameters that are engaged in the Modified Genetic Algorithm
 Minimum error threshold= 0.0009.
 Population size= 100.
 Maximum no. of epochs=1700.
 Fitness function= Mean squared error.
 No. of layers=2.
 No. of neurons in input layer= 4.
 Number of epochs trained= 809.
 Mean squared error: **Mse** = 7.0977×10^{-4} .

The graph obtained in terms of the number of iteration versus the MSE is as shown in the Figure 8. It is observed that

Table 2. Mean Square Error for the first 20 Iteration

SL NO	Mean Square Error		
	GA with ANN	MGA with ANN	ANN
1	0.008744818256473	7.7318e-004	1.2378e-008
2	0.016569798565873	7.2050e-004	1.6401e-007
3	0.036848922124476	6.2239e-004	1.1747e-010
4	0.039006833678980	8.9865e-004	8.8295e-008
5	0.028214233993611	8.7416e-004	1.3922e-009
6	0.025954975655545	6.9973e-004	2.6955e-010
7	0.024663467793999	8.9208e-004	6.7710e-010
8	0.034720373725770	8.6376e-004	2.1662e-009
9	0.023586805197258	8.7523e-004	3.4198e-008
10	0.032668187608547	5.0078e-004	6.7636e-010
11	0.039077233098494	8.8526e-004	6.9386e-013
12	0.016923061624203	7.0103e-004	5.9816e-008
13	0.038761728049251	7.4750e-004	3.8834e-010
14	0.038403166474452	7.2447e-004	3.4912e-010
15	0.035171295319183	4.5894e-004	9.5231e-009
16	0.039787677166057	5.8214e-004	1.9748e-009
17	0.035801775255144	4.1583e-004	4.3747e-010
18	0.038249568911908	7.4492e-004	4.9736e-012
19	0.034758617723503	7.5912e-004	1.3351e-009
20	0.030514183397513	8.1756e-004	3.2446e-014

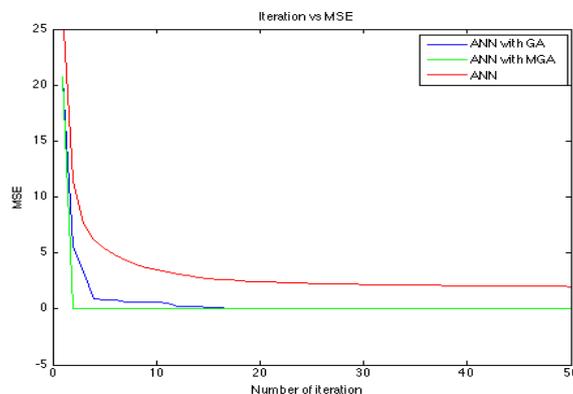


Figure 8. “Iteration” vs “MSE” Graph for ANN, ANN with GA and ANN with MGA

The Matlab/Simulink based simulation of the multilevel inverter and the M-file based ANN, ANN with GA optimisation and ANN with MGA optimisation of the ANN weights and bias values has been implemented. The optimisation of these three methods has been depicted in the figure .8, which shows that the ANN method has taken around 18th iteration for settling to the lowest MSE, ANN with GA has taken 4 iteration to reach the lowest MSE it could attain, ANN with MGA would reach its lowest MSE in just 2 iterations. And also we could observe that the MSE is at its lowest while we use the ANN with MGA method, then in the second position comes the ANN with GA and then the ANN. If a parallel processing is applied which would help in making the calculations in GA, MGA and ANN possible on the hardware then it would be suggested to apply the MGA with ANN for the implementation of the fault detection on the CMLI.

6. CONCLUSION

The fault diagnosis system is applied to the meta-heuristic algorithms to get optimal solutions, which makes the diagnosis system to identify faster and give the control signal. The GA, and MGA methods are compared to produce optimal solution if the fault occurs. The test system is tested with the Matlab/Simulink and the algorithms are coded as per the algorithm using M-files. The objective function used for this test system is the mean square error (MSE) on classification of fault and the results shows that the MGA algorithm gives fast and accurate results compared to the other methods. The results during the testing portion had given 100% accuracy for all the fault condition thus has performed well than the [12] where at some fault conditions it has performed 80% accuracy.

ACKNOWLEDGEMENTS

Authors gratefully acknowledge the support of Mr. V Ganeshan and Dhamodharan S from ViswaJothi Technologies Private Limited, Inc. for their valuable suggestions and software support.

REFERENCES

- [1] Madichetty, Sreedhar, "Modular Multilevel Converters PartI: A Review on Topologies Modulation Modeling and Control Schemes", *International Journal of Power Electronics and Drive Systems*, Vol-4, 36-50, IJAES 2014.
- [2] Manjunatha, Y. R., "Multilevel DC Link Inverter with Reduced Number of Switches and Batteries", *International Journal of Power Electronics and Drive Systems*, Vol 4, 299-307, IJAES 2014.
- [3] Maryam. S et al., "Analysis and Control of DC Capacitors-Voltage- Drift Phenomenon of a Passive Front-End Five-Level Converter", *IEEE Trans. Industrial Electronics*, Vol. 54, No. 6, Dec. 2007, pp.3255-3206.
- [4] J.S. Lai and F.Z. Peng, "Multilevel Converter—A New Breed of Power Converters", *IEEE Trans. Industry Applications*, Vol. 32, No. 3, May/June 1996, pp. 509-517.
- [5] A. Lesnicar and R. Maruardt, "An Innovative Modular Multilevel Converter Topology Suitable for a Wide Power Range", *IEEE Power Tech Conference*, Vol. 3, Bologna, Italy, 23-26 June 2003.
- [6] X. Xu, Y. Zou, K. Ding, and F. Liu, "Cascade multilevel inverter with phase-shift SPWM and its application in STATCOM", in *Proc. IEEE IECON*, 2004, vol. 2, pp. 1139–1143.
- [7] C. Newton and M. Sumner, "Neutral Point Control for Multilevel Inverters: Theory, Design and Operational Limitations", *IEEE Industry Applications Society Annual Meeting*, New Orleans, Louisiana, October 1997, pp. 1336-1343.
- [8] Surin Khomfoi and Leon M. Tolbert, "Fault Diagnosis and Reconfiguration for Multilevel Inverter Drive Using AI-Based Techniques", *IEEE Transactions On Industrial Electronics*, Vol. 54, No. 6, December 2007 .
- [9] D. Eaton, J. Rama, and P. W. Hammond, "Neutral shift", *IEEE Ind. Appl. Mag.*, vol. 9, no. 6, pp. 40–49, Nov./Dec. 2003.
- [10] Karimi S, Gaillard A, Poured P, et al., "FPGA-Based Real-Time Power Converter Failure Diagnosis for Wind Energy Conversion Systems", *IEEE Tran. Ind. Electron.*, vol.55, no. 12, pp. 4299-4308, Dec. 2008.
- [11] T. Bartz-Beielstein, C. Lasarczyk, and M. Preuss, "Sequential parameter optimization", in *Proc. IEEE CEC*, San Francisco, CA, USA, Sep. 2005, pp. 773–780.
- [12] Rongjun Li and Xianying Chang, "A Modified Genetic Algorithm with Multiple Subpopulations and Dynamic Parameters Applied in CVaR model", *International Conference on Computational Intelligence for Modelling Control and Automation, and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06) IEEE* 2006.
- [13] Rammohan Rao Errabelli et al, "Fault-Tolerant Voltage source Inverter for Permanent Magnet Drives", *IEEE Trans. On Power Electronics*, Vol.27, 500-508, IEEE 2012.
- [14] Jiang Wei et al, "Fault detection and Remedy of Multilevel Inverter based on BP Neural Network", *Power and Energy Engineering Conference (APPEEC)*, IEEE 2012 Asia-Pacific.

BIOGRAPHIES OF AUTHORS



T.G.Manjunath, email ID: tgmnaht@gmail.com, working as an Associate professor & HOD at Sai Vidya Institute of Technology Bangalore, Karnataka. He received his B.E Degree from R.V. College of Engineering, Bangalore (Bangalore University) and Master Degree from U.V.C.E .Bangalore (Bangalore University) in the year 1999 and 2004 respectively. He worked in National Aerospace Laboratories (NAL) as project Associate & having around 15 years of teaching experience in reputed Intitutions.He is currently pursuing his Doctoral research in Visveswaraya Technological University, Belgavi. His research includes in the field of Multi level Inverters, DC-DC Converters and ANN.



Dr. Ashok Kusagur, email ID: ashok.kusagur@gmail.com, working as an Associate professor & HOD at U.B.D.T. College of Engineering, Davanagere, Karnataka. He received his B.E Degree from Bapuji Institute of Engineering & Technology, Davanagere (Kuvempu University) and Master Degree from PDA College of Engineering Gulbarga (Visveswaraya Technological University, Belgavi) in the year 1996 and 2001 respectively. He received his Ph.D. in the area “Design and implementation of Neuro –Fuzzy based speed control of Induction motor Drive by Space Vector Pulse width Modulation for Voltage source Inverters” from J.N.T.U Hyderabad in the year 2012. Under his supervision currently 8 research scholars are working. He has published 32 papers in various National /International journals and conferences in India & Abroad. He is having around 18 years of teaching experience in reputed Institutions. His areas of interest are Fuzzy logic control of motor Drives & Artificial Neural Network.