A Diagnostic Technique for Multilevel Inverters Based on a Genetic-Algorithm to Select a Principal Component Neural Network

Surin Khomfoi, Leon M. Tolbert
The University of Tennessee
Electrical and Computer Engineering
414 Ferris Hall, Knoxville, TN 37996-2100, USA

Abstract- **A genetic-algorithm-based selective principal component neural network method for fault diagnosis system in a multilevel inverter is proposed in this paper. Multilayer perceptron (MLP) networks are used to identify the type and location of occurring faults from inverter output voltage measurement. Principal component analysis (PCA) is utilized to reduce the neural network input size. A lower dimensional input space will also usually reduce the time necessary to train a neural network, and the reduced noise may improve the mapping performance. The genetic algorithm is also applied to select the valuable principal components. The neural network design process including principal component analysis and the use of genetic algorithm is clearly described. The comparison among MLP neural network (NN), principal component neural network (PC-NN), and genetic algorithm based selective principal component neural network (PC-GA-NN) are performed. Proposed networks are evaluated with a simulation test set and an experimental test set. The PC-NN has improved overall classification performance from NN by about 5% points, whereas PC-GA-NN has better overall classification performance from NN by about 7.5% points. The overall classification performance of the proposed networks is more than 90%.**

Index Terms **— Fault diagnosis, genetic algorithm, multilevel inverter, neural network, principal component analysis.**

I. INTRODUCTION

Industry has begun to demand higher power ratings, and multilevel inverter drives have become a solution for high power applications in recent years. A multilevel inverter not only achieves high power ratings, but also enables the use of renewable energy sources. Two topologies of multilevel inverters for electric drive application have been discussed in [1]. The cascaded MLID is a general fit for large automotive all-electric drives because of the high VA rating possible and because it uses several dc voltage sources which would be available from batteries or fuel cells [1].

A possible structure of a three-phase cascaded multilevel inverter drive for an electric vehicle is illustrated in Fig. 1. The series of H-bridges makes for modularized layout and packaging; as a result, this will enable the manufacturing process to be done more quickly and cheaply. Also, the reliability analysis reported in [2] indicates that the fault-tolerance of cascaded MLID had the best life cycle cost. However, if a fault (open or short circuit) occurs at a semiconductor power switch in a cell, it will cause an unbalanced output voltage and current, while the traction motor is operating. The unbalanced voltage and current may result in vital damage to the traction motor if the traction motor is run in this state for a long time.

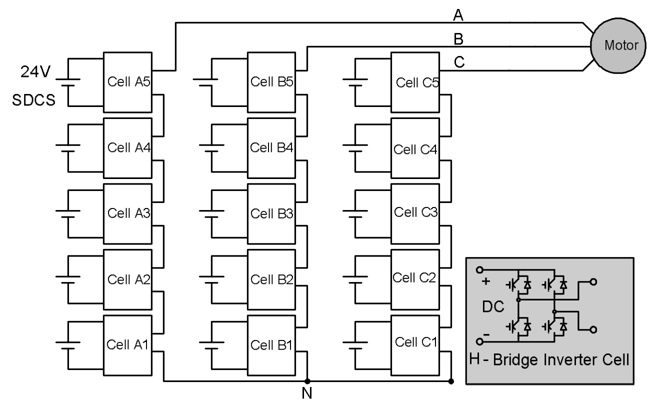


Fig. 1. Three-phase wye-connection structure for electric vehicle motor drive.

Generally, the passive protection devices will disconnect the power sources or gate drive signals from the multilevel inverter system whenever a fault occurs, stopping the operated process. Although a cascaded MLID has the ability to tolerate a fault for some cycles, it would be better if we can detect the fault and its location; then, switching patterns and the modulation index of other active cells of the MLID can be adjusted to maintain the operation under balanced load condition. Of course, the MLID can not be operated at full rated power. The amount of reduction in capacity that can be tolerated depends upon the application; however, in most cases a reduction in capacity is more preferable than a complete shutdown.

A study on fault diagnosis in drives begins with a conventional PWM voltage source inverter (VSI) system [3-5]. Then, artificial intelligent (AI) techniques such as fuzzy-logic (FL) and neural network (NN) have been applied in condition monitoring and diagnosis [6-8]. Furthermore, a new topology with fault-tolerant ability that improves the reliability of multilevel converters is proposed in [9]. A method for operating cascaded multilevel inverters when one or more power H-bridge cells are damaged has been proposed in [2,10]. The method is based on the use of additional magnetic contactors in each power H-bridge cell to bypass the faulty cell.

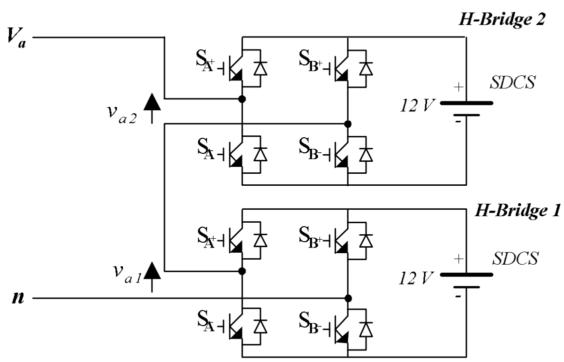
One can see from the concise literature survey that the

1-4244-0714-1/07/$20.00 ©2007 IEEE.

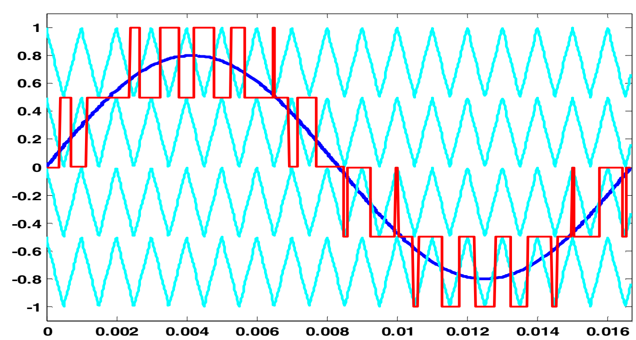
knowledge and information of fault behaviors in the system is important to improve system design, protection, and fault tolerant control. Thus far, limited research has focused on MLID fault diagnosis and reconfiguration. Therefore, a MLID diagnostic system is proposed in this paper that only requires measurement of the MLID's voltage waveforms.

II. Diagnostic Signals

It is possible that AI-based techniques can be applied in condition monitoring and diagnosis. AI-based condition monitoring and diagnosis have several advantages. For instance, AI-based techniques do not require any mathematical models; therefore, the engineering time and development time could be significantly reduced [11]. AI-based techniques utilize the data sets of the system or make full utilization of expert knowledge. In MLID applications, the output phase voltage can convey valuable information to diagnose the faults and their locations. For example, output voltage signals of open circuit faults in each location of two 12 V separate dc source (SDCS) MLID as shown in Fig. 2 with multilevel carrier-based sinusoidal PWM gate drive signals are shown in Fig. 3 and Fig. 4 for H-bridge 1 and H-bridge 2, respectively. Obviously, all output voltage signals are related to the fault locations. Also, the output voltages of a MLID can also be used to diagnose the fault types (open and short circuit) as depicted in Fig. 5.

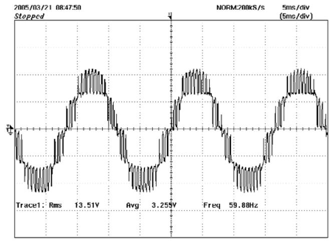


(a)

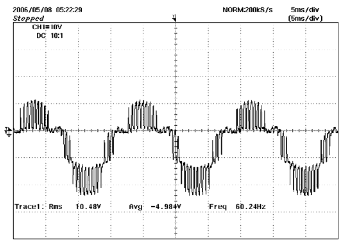


(b)

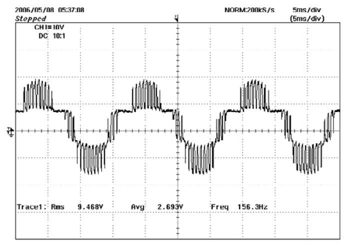
Fig. 2. (a) Single-phase multilevel-inverter system; (b) Multilevel carrier-based sinusoidal PWM showing carrier bands, modulation waveform, and inverter output waveform (m_a = 0.8/1.0).



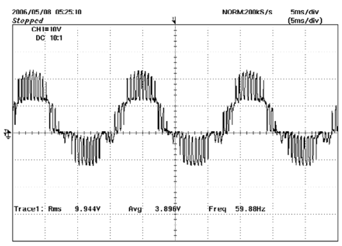
(a)



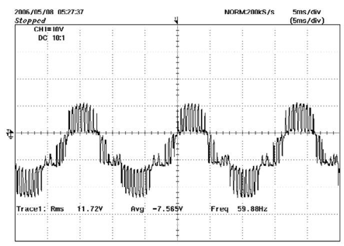
(b)



(c)

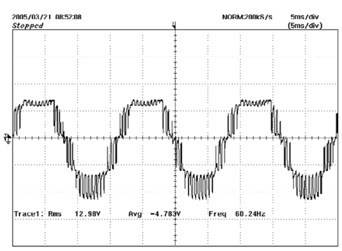


(d)

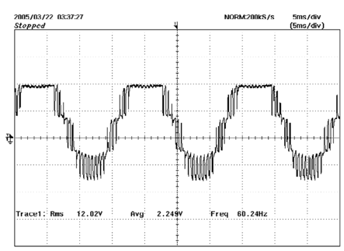


(e)

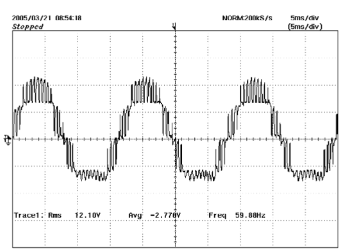
Fig. 3. Experiment of fault features at (a) normal, (b) S_{A+} fault, (c) S_{A-} fault, (d) S_{B+} fault, and (e) S_{B-} fault of H-bridge 1 with modulation index = 0.8 out of 1.0.



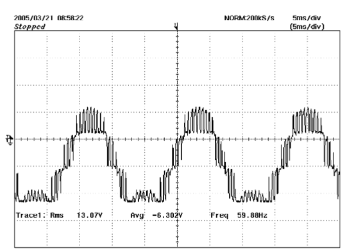
(a)



(b)



(c)



(d)

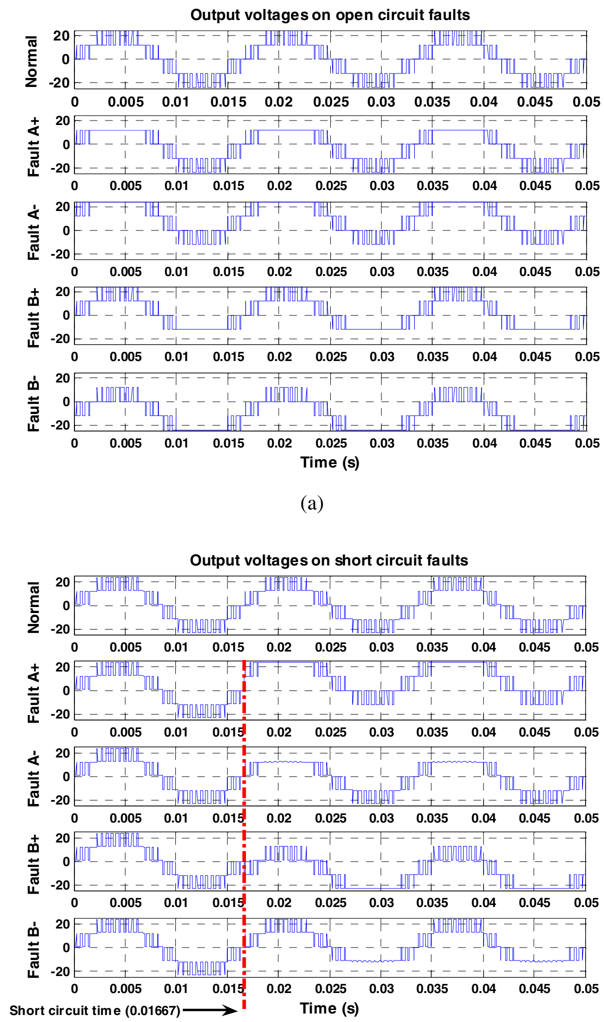
Fig. 4. Experiment of fault features at (a) S_{A+} fault, (b) S_{A-} fault, (c) S_{B+} fault, and (d) S_{B-} fault of H-bridge 2 with modulation index = 0.8 out of 1.0.

As can be seen, the signals are difficult to rate as an important characteristic for classifying a fault hypothesis, and they have high correlation with each other; hence, a signal transformation technique is needed. The comparison of signal transformation suitable to training a neural network for fault diagnosis tools is elucidated in [12]. The fault diagnosis system for a MLID using FFT and neural network are proposed in [14]. The proposed technique has a good classification performance to classify normal and abnormal features. However, many neurons are used to train the network (i.e. one neuron for each harmonic); therefore, principal component analysis (PCA) can be used to reduce the number of input neurons as proposed in [15, 16]. PCA is a method used to reduce the dimensionality of an input space without losing a significant amount of information (variability) [13]. The method also makes the transformed vectors orthogonal and uncorrelated. A lower dimensional input space will also usually reduce the time necessary to train a neural network, and the reduced noise (by keeping only valuable principal components) may improve the mapping performance.

III. Principal Component Selection

The selection of principal component (PC) is very significant because input selected PCs can cause uncertainty results: (1) additionally unneeded input PCs to the neural network can increase the solution variance; (2) absent necessary input PCs can increase bias. Usually, there are three methods to select a valuable PC: observed eigenvalue method, correlated method, and trial and error method. First, the observed eigenvalues method will choose PCs that contain most of the information (variability of original data set). This method is good for function approximation problems; however, it may not be useful for classification problems. Second, the correlated method will select PCs that are well correlated with the response variable. The correlated method is superior in both function approximation and classification problems; nevertheless, the method may not be an optimized solution and may consume a lot of time. Third, the trial and error method will pick the combination of PCs that provides minimum error; for instance, the misclassification error of the neural network. The trial and error method can offer a minimum error of the neural network, but the method requires a lot of time to search the optimum combination of PCs for the model. One possible tool to search the optimized combination of PCs is a genetic algorithm.

The comparison in classification performance between the network proposed in [14] and the principal component neural network (PC-NN) is discussed in [16]. By using PCA, the size of input neurons can be reduced from 40 nodes to 5 nodes (i.e. 5 harmonics instead of 40 harmonic components) [16]. The results show that the PC-NN has a better overall classification performance by 5 % points; however, the multivariable optimization technique such as genetic algorithm (GA) could improve the classification performance; therefore, a GA is applied to select the valuable principal components to train the neural network as shown in Fig. 6. Since a GA offers multivariable optimized search solution, the best combination of PCs or the minimum misclassification rate could be found, which leads to the improvement of total classification performance of the neural networks. The example of signal transformation using PCA is shown in Fig. 7. These 3-D plots of PC scores are transformed with training data set used in [16]. We can see that the classification between normal and faults could be a linear problem, whereas the classification among faults is a nonlinear problem. That is why the neural network is applied to solve this problem.



(a)

(b)

Fig. 5. Simulation of output voltages signals (a) open circuit faults, (b) short circuit faults showing fault features at S_{A+}, S_{A-}, S_{B+}, and S_{B-} of H-bridge 2 with modulation index = 0.8 out of 1.0.

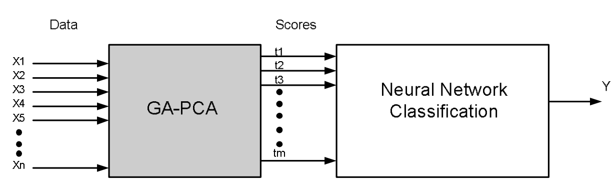


Fig. 6. Genetic algorithm principle component neural network (GA-PC-NN).

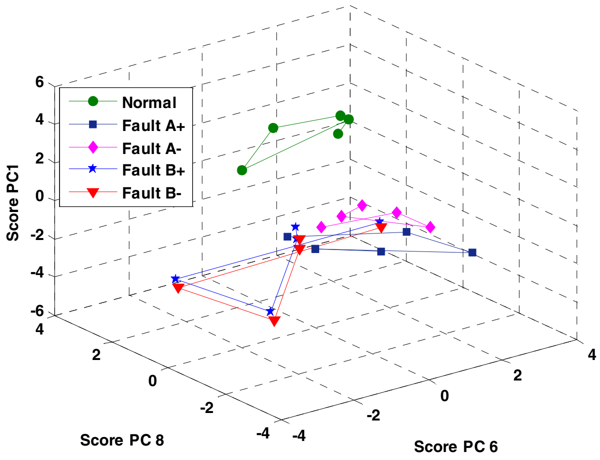


Fig. 7. The 3-D plots of PC scores on PC 6, 8, 1.

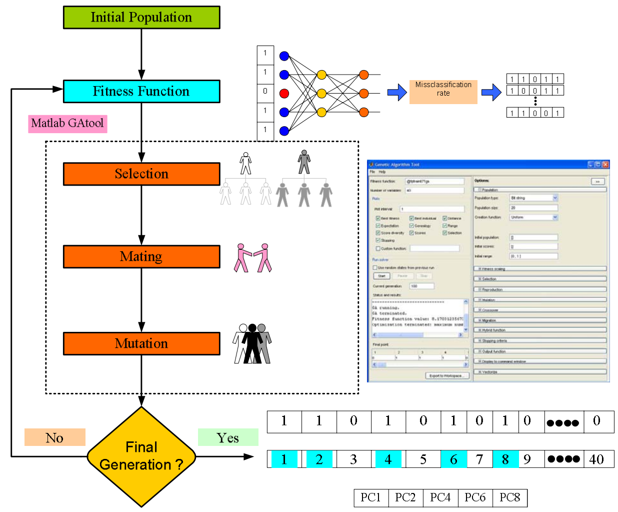


Fig. 8. The flowchart of the GA application for PC selection.

IV. Principal Component Selection by Genetic Algorithm

The principal components (PC) selected in [16] are the 1, 2, 4, 6, and 8 PCs. The PC selection method used in [16] is the correlated method. (The PCs are correlated with response variables). In this paper, the genetic algorithm (GA) is used to perform the PC selection as illustrated in Fig. 8. The result of PC selection by GA will be compared with the result from the correlated method as proposed in [16]. We can see from the flow chart in Fig. 8 that the discrete GA (DGA) or binary GA can be applied to selecting PC. The idea is to randomly pass the PCs into the neural network, and then a GA is utilized to search for the best combination of input PCs. The steps of the GA process can be explained as follows:

Encoded input PCs: the PCs to be optimized are represented by chromosomes where each PC is encoded as a binary string known as a gene. Therefore, a chromosome consists of multiple genes as PCs to be selected. A population, consisting of a provided number of chromosomes is initially generated by haphazardly assigning "1" and "0" to all genes except for one chromosome which assigns to use all PCs. The binary string of the chromosomes has the same size as PCs to select from, whereby the presence of a PC is coded as "1", whereas the nonappearance of a PC is coded as "0". Accordingly, the binary string of a gene consists of only one single bit. The example of encoded input PCs is illustrated in Fig. 8 on the right hand side. We can see that the bit "0" will not be used to train the network, whereas others will be used to train the network.

Fitness function: The best chromosomes have the highest probability to survive as evaluated by the fitness function. An important point in applying GA is the design of the fitness function. A fitness function determines what a GA should optimize. In this research, the goal is to find the combination of selective PCs for fault classification which provides the minimum classification error. In this case, the classification is based on neural networks for modeling the relationship between input variables (PCs scores) and the responses (fault classes). Therefore, the evaluation of the fitness function begins with the encoding of the chromosomes into neural networks. Then, the networks are trained with a training set; and after that, the test set is examined. It should be noted that the test set in this research consists of simulation test set and experiment test set. Finally, the fitness function is evaluated by using (1) and (2).

The fitness function is divided into two parts: sum of square error (SSE) of simulation set and SSE of experiment test set. In this research, we weigh the experiment test set higher than the simulation test set because we only use the normal and fault data from the simulation to train the neural networks. Also, the classification performance as presented in [14-16] shows that the neural networks have higher classification performance in the simulation test set than experimental test set. The networks in [14,16] have misclassification of 1 out of 5 fault classes in the simulation test set; therefore, in this particular case, weighted factors of 0.2 and 0.8 are used for the simulation test set and the experimental test set, respectively.

$$f = 0.2\,SSE_{Sim,set} + 0.8\,SSE_{Exp,set}\,, \tag{1}$$

$$SSE = \sum_{i=1}^{n} \left(y - \bar{y}_i \right)^2\,, \tag{2}$$

where y is the output target binary codes,
$\bar{y}_i$ is output of training data,
n is the number of training data.

GA parameter selection: As previously mentioned, the *gatool* is used to set the GA options as shown in Fig. 8 on the right hand side. The GA parameters can be conveniently selected by *gatool* [17]. It should be noted that different GA parameters could give different results. For this particular example, the number of variables is 40, the population size is 20, and the fitness scaling is by rank.

After evolving the fitness function of the population as shown in Fig. 8, the individuals are selected by using a roulette wheel; this can be directly set in *gatool* as “Roulette” in “selection” toolbar. Thereby, the chromosomes are allocated space on a roulette wheel proportional to their fitness value, and thus the individuals with a higher fitness score are more likely selected. The next step is the mating process; a single point crossover technique is utilized. A crossover process will create offspring chromosomes which randomly select a crossover point within the chromosome. Then, two parent chromosomes are interchanged at this point to generate two new offspring. After that, the chromosomes are mutated with a probability of 0.005 per gene by erratically changing genes from 0 to 1 and vice versa. The mutation prevents the GA from converging too quickly in a small area of search space [18]. Again, it should be noted that different GA parameters may give different results. Therefore, the GA parameter selection might need some experience in a particular application.

Stopping Criteria: The evaluation and reproduction steps are repeated until a certain number of generations, a defined fitness, or a convergence criterion of the population are reached. In this research, the maximum number of generations is 100. Ideally, all chromosomes of the last generation have the same genes representing the optimal solution.

By using the same original data set represented in [14, 16], the best result from *gatool* after several attempts is shown in Table I. The final point shows that the 8 PCs are selected by GA consisting of **1, 2, 3, 5, 7, 8, 13,** and **14** with a minimum SSE 0.205. The PCs selected by correlated method [16] are **1, 2, 4, 6,** and **8**. This GA result is interesting because we know that both 13 and 14 PCs contain small variance of the information as discussed in [16]; however, we can see from the plot of the score on PC 14 in Fig. 9 that PC 14 can be used to categorize between Fault A+ and A-; also PC 14 can be used to classify between Fault A+, Fault A- and Fault B+ and Fault B-. Meanwhile, the loading plot on PC 13 that the sampling from 16 to 20 has mostly positive eigenvalue, whereas the sampling from 21 to 25 has mostly negative eigenvalue. This shows that PC 13 can be used to classify between Fault B+ and Fault B-.

V. Neural Network Classification

The multilayer feedforward networks or MLP are used in this research. The neural network architecture designs have been proposed in [14] and will not be repeated here. Since the comparison among transformation methods: only FFT [14], PCA [16], and GA-PCA will be performed, three different neural network (NN) architectures are used. The original data from the feature extraction system (FFT) used to train and test the networks are exactly the same data set. The first NN architecture has one hidden layer with 40 input nodes, 4 hidden nodes, and 3 output nodes as discussed in [14]. The second NN architecture has one hidden layer with 5 input nodes, 3 hidden nodes, and 3 output nodes. The PCA is applied in this network to reduce the number of input neurons as discussed in [16]. The combination of ***1, 2, 4, 6, and 8 principal components*** will be used to perform the neural network classifications.

TABLE I
The Output Final Solution From GA Using *gatool*.

Description	**Outputs from *gatool (40 PC variables)***
Final point	*1 2 3 4 5 6 7 8 9 10 11 12 13 14* **1 1 1 0 1 0 1 1 0 0 0 0 1 1** ***PC 15-40 are all 0***
Final SSE	**0.205**

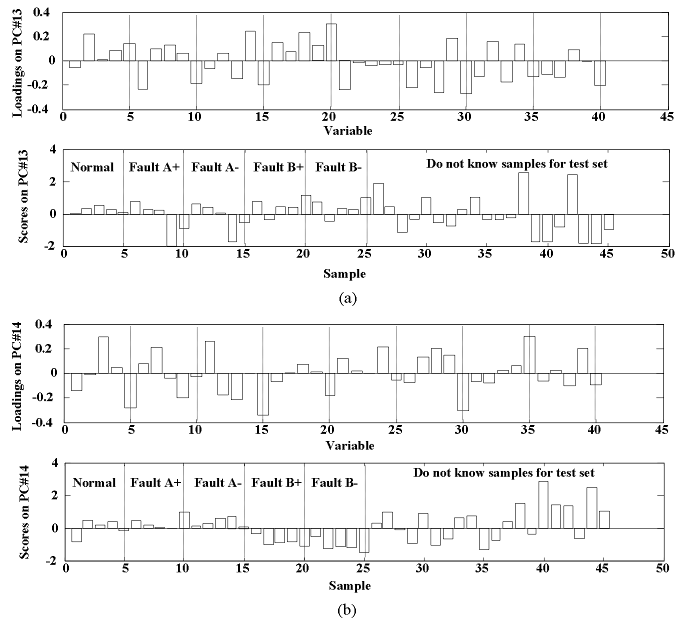


Fig. 9. The plot of principal component loading and score of (a) PC 13 and (b) PC 14.

The third NN architecture is based on GA selection as discussed in section IV because the input neurons depend on how many PCs selected by GA. However, the one hidden layer with 3 hidden nodes and 3 output nodes are used since the comparison among proposed NN will be performed so that the NNs should have the same complexity and degree of freedom. The first network requires more neurons because the network has more input neurons. The sigmoid activation function is used: *tansig* for hidden nodes and *logsig* for an output node. A *logsig* activation function is used for an output node because the target output is between 0 and 1. It should be noted that the number of nodes for the input and output layers depends on the specific application. The selection of number and dimension in the hidden layer is based on neural network accuracy in preliminary tests.

The performances of the proposed networks are tested in two categories. First, the networks are tested with the simulation test sets as previously mentioned. Second, the networks are evaluated with the experimental test set. The tested results along with the testing data sets are illustrated in Table II.

Clearly, in the simulation test set, all proposed networks have a good classification performance (about 95%); therefore, the classification performance of the networks is quite satisfactory. The misclassification samples are the same operation point and class which are 0.65 modulation index and fault B-. This result suggests that both networks have confusion between Fault A- and Fault B- at low modulation index. The second category of experimental testing results is also illustrated in Table II. Obviously, the classification performance of PC-GA-NN is better than NN by 15% points and PC-NN by 5%. The NN has 85 % classification performance, and PC-NN has 95% classification, whereas PC-GA-NN has 100% classification performance. As expected, PCA conveys lower dimensional input space, reducing the time necessary to train a neural network. Also, the reduced noise could improve the mapping performance which leads to the improvement of total classification performance. GA offers the multivariable search of the minimum misclassification error providing the better neural network performance.

TABLE II

CONFUSION TABLE FOR NEURAL NETWORK CLASSIFICATION PERFORMANCE .

Testing set	Target	Actual Output			% Classification		
		NN	PC-NN	PC-GA-NN	NN	PC-NN	PC-GA-NN
Simulation test set	Normal [1 1 1]	1 1 1 1 1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1 1 1 1 1	100%	100%	100%
	Fault A+ [0 0 1]	0 0 1 0 0 1 0 0 1 0 0 1	0 0 1 0 0 1 0 0 1 0 0 1	0 0 1 0 0 1 0 0 1 0 0 1	100%	100%	100%
	Fault A- [0 1 0]	0 1 0 0 1 0 0 1 0 0 1 0	0 1 0 0 1 0 0 1 0 0 1 0	0 1 0 0 1 0 0 1 0 0 1 0	100%	100%	100%
	Fault B+ [1 0 1]	1 0 1 1 0 1 1 0 1 1 0 1	1 0 1 1 0 1 1 0 1 1 0 1	1 0 1 1 0 1 1 0 1 1 0 1	100%	100%	100%
	Fault B- [1 1 0]	1 1 0 1 1 0 1 1 0 0 1 0	1 1 0 1 1 0 1 1 0 0 1 0	1 1 0 1 1 0 1 1 0 0 1 0	75%	75%	75%
% Classification performance in simulation test set					***95%***	***95%***	***95%***
Experiment test set	Normal [1 1 1]	1 1 1 1 1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1 1 1 1 1	100%	100%	100%
	Fault A+ [0 0 1]	0 1 1 0 1 1 0 1 1 0 1 0	0 0 1 0 0 1 0 0 1 0 0 1	0 0 1 0 0 1 0 0 1 0 0 1	75%	100%	100%
	Fault A- [0 1 0]	0 1 0 0 1 0 0 1 0 0 0 1	0 1 0 0 1 0 0 1 0 0 1 0	0 1 0 0 1 0 0 1 0 0 1 0	75%	100%	100%
	Fault B+ [1 0 1]	1 0 1 1 0 1 1 0 1 1 0 1	1 0 1 1 0 1 1 0 1 1 0 1	1 0 1 1 0 1 1 0 1 1 0 1	100%	100%	100%
	Fault B- [1 1 0]	1 1 0 1 1 0 1 1 0 0 1 0	1 1 0 1 1 0 1 1 0 0 1 0	1 1 0 1 1 0 1 1 0 1 1 0	75%	75%	100%
% Classification performance in experimental test set					***85%***	***95%***	***100%***
Total % Classification performance					**90%**	**95%**	**97.5%**

Obviously, PC-GA-NN has a better overall classification performance of about 2.5% and 7.5 % points compared with PC-NN and NN, respectively. The results show that the application of GA and PCA can improve the classification performance of the neural networks. Consequently, we know that the higher classification performance of the proposed neural networks could give higher reliability of a fault diagnostic system in an MLID.

VI. Experiment Validation

The PC-GA-NN is implemented for a fault diagnostic system by using the reconfiguration technique proposed in [19]. The detail of the experiment setup with a three-phase wye-connected cascaded multilevel inverter is also presented in [19]. Fig. 10 shows that the proposed system utilizes about 6 cycles to clear the fault. It should be noted that the window of FFT function requires at least one cycle to perform signal transformation. The clearing time can be shorter than this if the proposed system is implemented as a single chip using an FPGA or DSP. Since the cascaded MLID can tolerate a few cycles of faults, and the proposed system can detect the fault and can correctly reconfigure the MLID, the results are satisfactory.

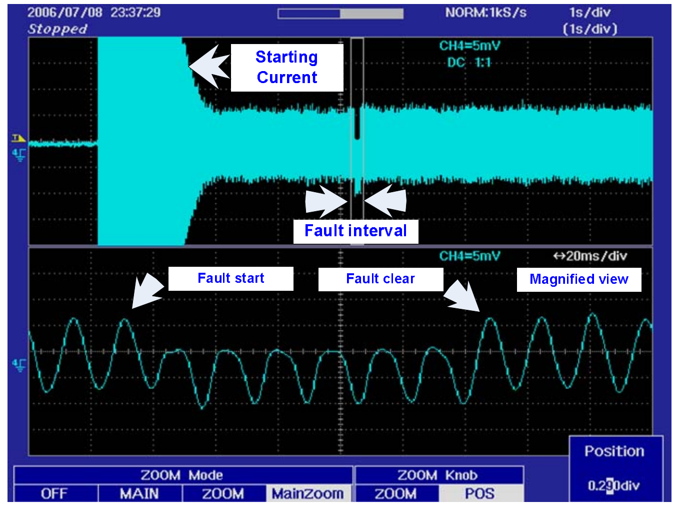


Fig. 10. Experiment results of the open circuit fault at S1, cell 2 of the 5 SDCS-MLID at ma = 0.8/1.0

VII. Conclusion

The methodology of a genetic-algorithm-based selective principal component neural network applied to fault diagnostic system in a cascaded multilevel inverter has been presented. The GA-PC-NN performs very well with both simulation and experimental testing data set. The total classification performance is very good by about 97.5% points. Obviously, the results show that the PC-GA-NN has a better overall classification performance than PC-NN by about 2.5% points. PCA conveys lower dimensional input space and reduces the time necessary to train a neural network. Also, the reduced noise may improve the mapping performance; in addition, GA offers multivariable optimized search so that the best combination of PCs or the minimum misclassification rating could be found, which leads to the improvement of total classification performance of the neural networks.

References

[1] L. M. Tolbert, F. Z. Peng, T.G. Habetler, "Multilevel Converters for Large Electric Drives," *IEEE Trans. Industry Applications*, vol. 35, no. 1, Jan/Feb. 1999, pp. 36-44.

[2] D. Eaton, J. Rama, and P. W. Hammond, "Neutral Shift," *IEEE Industry Applications Magazine*, Nov./Dec. 2003, pp. 40-49.

[3] D. Kastha, B. K. Bose, "Investigation of Fault Modes of Voltage-fed Inverter System for Induction Motor Drive," *IEEE Trans. Industry Applications*, vol. 30, no. 4, Jul. 1994, pp. 1028-1038.

[4] D. Kastha, B. K. Bose, "On-Line Search Based Pulsating Torque Compensation of a Fault Mode Single-Phase Variable Frequency Induction Motor Drive," *IEEE Trans. Industry Applications*, vol. 31, no. 4, Jul./Aug. 1995, pp. 802-811.

[5] A. M. S. Mendes, A. J. Marques Cardoso, E. S. Saraiva, "Voltage Source Inverter Fault Diagnosis in Variable Speed AC Drives by Park's Vector Approach," in *Proceedings of the 1998 IEE 7th International Conference on Power Electronics and Variable Speed Drives,* pp. 538-543.

[6] S. Hayashi, T. Asakura, S. Zhang, "Study of Machine Fault Diagnosis Using Neural Networks," in *Proceedings of the 2002 Neural Networks, IJCNN '02*, Vol. 1, pp. 956 – 961.

[7] S. Zhang, T. Asakura, X. Xu, B. Xu, "Fault Diagnosis System for Rotary Machines Based on Fuzzy Neural Networks," in *Proceedings of the 2003 IEEE/ASME Advanced Intelligent Mechatronics,* pp. 199-204.

[8] A. Bernieri, M. D'Apuzzo, L. Sansone, M. Savastano, "A Neural Network Approach for Identification and Fault Diagnosis on Dynamic Systems," *IEEE Trans. Instrumentation and Measurement*, vol. 43, no. 6, Dec. 1994, pp. 867-873.

[9] A. Chen, L. Hu, L. Chen, Y. Deng, X. He, "A Multilevel Converter Topology With Fault-Tolerant Ability," *IEEE Trans. on Power Electronics*, vol. 20, no. 2, March. 2005, pp. 405-415.

[10] J. Rodriguez, P. W. Hammond, J. Pontt, R. Musalem, P. Lezana, M. J. Escobar, "Operation of a Medium-Voltage Drive Under Faulty Conditions," *IEEE Trans. on Industrial Electronics*, vol. 52, no. 4, August 2005, pp. 1080-1085.

[11] P. Vas, *Artificial-Intelligence-Based Electrical Machines and Drives,* Oxford University Press, Inc.,New York, 1999.

[12] J. A. Momoh, W. E. Oliver Jr, J. L. Dolc, "Comparison of Feature Extractors on DC Power System Faults for Improving ANN Fault Diagnosis Accuracy," in *Proceedings of the 1995 IEEE Intelligent Systems for the 21st Century,* vol. 4, pp. 3615-3623.

[13] I. T. Joliffe, *Principal Component Analysis,* Springer; 2nd edition, 2002.

[14] S. Khomfoi, L. M. Tolbert, "Fault Diagnosis System for a Multilevel Inverters Using a Neural Network," *IEEE Industrial Electronics Conference,* November 6-10, 2005, pp. 1455-1460.

[15] J. Ding, A. Gribok, J. W. Hines, B. Rasmussen "Redundant Sensor Calibration Monitoring Using ICA and PCA," *Real Time Systems Special Issue on Applications of Intelligent Real-Time Systems for Nuclear Engineering,* 2003.

[16] S. Khomfoi, L. M. Tolbert, "Fault Diagnosis System for a Multilevel Inverters Using a Principal Component Neural Network," 37th IEEE Power Electronic Specialists Conf., June 18-22, 2006, pp. 3121-3127.

[17] Genetic Algorithm and Direct Search Toolbox User's Guide, Matworks,Inc ; Version 2, 2004.

[18] U. Depczynski, V.J. Frost, K. Molt,"Genetic Algorithms Applied to the Selection of Factors in Principal Component Regression," Analytica Chimica Acta, Elsevier Science, Vol. 420, No. 2, Sep. 14, 2000, pp. 217-227

[19] S. Khomfoi, L. M. Tolbert, "A Reconfiguration Technique for Multilevel Inverters Incorporating a Diagnostic System Based on Neural Network," *10th IEEE Workshop on Computers in Power Electronics,* July 16-19, 2006, pp. 317-323.