Fault Diagnosis and Reconfiguration for Multilevel Inverter Drive Using AI-Based Techniques

Surin Khomfoi, Member, IEEE, and Leon M. Tolbert, Senior Member, IEEE

Abstract—A fault diagnostic and reconfiguration method for a cascaded H-bridge multilevel inverter drive (MLID) using artificial-intelligence-based techniques is proposed in this paper. Output phase voltages of the MLID are used as diagnostic signals to detect faults and their locations. It is difficult to diagnose an MLID system using a mathematical model because MLID systems consist of many switching devices and their system complexity has a nonlinear factor. Therefore, a neural network (NN) classification is applied to the fault diagnosis of an MLID system. Multilayer perceptron networks are used to identify the type and location of occurring faults. The principal component analysis is utilized in the feature extraction process to reduce the NN input size. A lower dimensional input space will also usually reduce the time necessary to train an NN, and the reduced noise can improve the mapping performance. The genetic algorithm is also applied to select the valuable principal components. The proposed network is evaluated with simulation test set and experimental test set. The overall classification performance of the proposed network is more than 95%. A reconfiguration technique is also proposed. The proposed fault diagnostic system requires about six cycles to clear an open-circuit or short-circuit fault. The experimental results show that the proposed system performs satisfactorily to detect the fault type, fault location, and reconfiguration.

Index Terms—Fault diagnosis, multilevel inverter, neural network (NN), reconfiguration.

I. INTRODUCTION

INDUSTRY has begun to demand higher power ratings, and multilevel inverter drives (MLIDs) 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. Renewable energy sources such as photovoltaic, wind, and fuel cells can be easily interfaced to a multilevel converter system for a high-power application. The cascaded multilevel inverter is one of the more optimistic solutions for high-power drives or large traction drives [1], [2]. A possible structure of a three-phase cascaded multilevel inverter for a high-power drive 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.

In addition, 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.

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 the fault type and its location can be detected; 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 might not be operated at full rated power after the bypass of a level. The amount of reduction of the rated power that can be tolerated depends upon the MLID application; nevertheless, in most cases a reduction of the rated power is more preferable than a complete shutdown.

Research on fault diagnostic techniques initially focused on conventional pulsed width modulation (PWM) voltage source inverters (VSI). The various fault modes of a VSI system for an induction motor are investigated in [3]. Then, the integration of a fault diagnostic system into VSI drive is described in [4]. This integrated system introduced remedial control strategies soon after failure occurrences; therefore, system reliability and fault-tolerant capability are improved. A noninvasive technique for diagnosing VSI drive failures based on the identification of unique signature patterns corresponding to the
motor supply current Park’s Vector is proposed in [5]–[7]. A comparison of features, cost, and limitations of fault-tolerant three-phase alternating current motor drive topologies is investigated in [8].

It is also advantageous that artificial intelligence (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 and development time could be significantly reduced. AI-based techniques utilize the data sets of the system or expert knowledge [9]. A general review of recent developments in the field of AI-based diagnostic systems in machine drives has been presented in [10]. The possibilities offered by a neural network (NN) for fault diagnosis and system identification are investigated in [11]. A study of a machine fault diagnostic system by using fast Fourier transform (FFT) and NNs is clearly explained in [12]. In addition, a fault diagnostic system for rotary machines based on fuzzy NNs is developed in [13]. A fault diagnostic system in electric drives using machine learning for detecting and locating multiple classes of faults has been implemented in [14].

A classification technique using an NN offers an extra degree of freedom to solve a nonlinear problem; the failure of a single neuron will only partially degrade performance. If an input neuron fails, the network can still make a decision by using the remaining neurons. In contrast, if only a single input, for instance the dc offset of signals, is used as the input data to classify the faults, the diagnosis system may not perform classification when the input data has drifted or the single sensor has failed. Furthermore, an NN also permits parallel configuration and seasonal changes. Additional H-bridges and fault features (short circuit) can be conveniently extended into the system with more training data and a parallel NN configuration.

Fault diagnosis and neutral point voltage control during a fault condition in a three-level diode-clamped multilevel inverter using Park’s Vector has been proposed in [15]. In addition, a unique fault-tolerant design for flying capacitor multilevel inverter has been presented in [16]. A method for operating cascaded multilevel inverters when one or more power H-bridge cells are damaged has been proposed in [2] and [17]. The method is based on the use of additional magnetic contactors in each power H-bridge cell to bypass the faulty cell and use the neutral shift technique to maintain balanced line-to-line voltages. One can see from the concise literature survey that the knowledge and information of fault behaviors in the MLID system is important to improve system design, protection, and fault-tolerant control. Thus far, limited research has focused on MLID fault diagnosis. Therefore, an MLID fault diagnostic system that only requires the measurement of the MLID’s voltage waveforms is proposed in this paper.

II. DIAGNOSTIC SIGNALS

Before continuing on with the discussion, it should be emphasized that the multilevel carrier-based sinusoidal PWM is used for controlling gate drive signals for the cascaded MLID, as shown in Fig. 2. Fig. 2 shows that the output voltages can be controlled by controlling the modulation index $m_a$. To expediently understand, a cascaded MLID structure with two separate dc sources (SDCSs) is used as an example in Sections II and III, but the techniques presented are valid for a multilevel converter with any number of levels, as demonstrated in Section V.

The selection of diagnostic signals is very important because the NN could learn from unrelated data to classify faults, which would result in improper classification. Simulation results (using power simulation (PSIM) from Powersim Inc.) of input motor current waveforms during an open-circuit fault at different locations of the MLID [as shown in Fig. 2(a)] are illustrated in Figs. 3 and 4. As can be seen in Figs. 3 and 4, the input motor currents can classify open-circuit faults at the same power cell by tracking current polarity (see Fig. 4); however, it is difficult to classify the faults at different power cells; the current waveform for a fault of $S_{A+}$ in H-bridge 2 (Fig. 3) is identical to that for a fault of $S_{A+}$ in H-bridge 1 [Fig. 4(a)]. As a result, the detection of fault locations could not be achieved with only using input motor current signals. In addition, the current signal is load dependent: the load variation may lead to misclassification; for instance, under light load operation as reported in [6].

Auspiciously, Fig. 2 indicates that an output phase PWM voltage is related with turn-on and turn-off time of related switches; hence, a faulty switch cannot generate a desired
output voltage: The output voltage for a particular switch is zero if the switch has a short-circuit fault, whereas the output voltage is about $V_{dc}$ of SDCS if the switch has an open-circuit fault. For this reason, the output phase voltage can convey valuable information to diagnose the faults and their locations. The simulation results of output voltages are shown for an MLID with open-circuit faults and short-circuit faults in Fig. 5. One can see that all fault features in both open-circuit and short-circuit cases could be visually distinguished. In addition, experimental results of output voltage signals of open-circuit faults in each location of a two-SDCS MLID, as shown in Fig. 2(a), with multilevel carrier-based sinusoidal PWM gate drive signals are shown in Figs. 6 and 7.

Obviously, the output phase voltage signals are related to the fault locations and fault types (open circuit and short circuit). One can see that all fault features can be visually distinguished in both fault types and fault locations via the output phase voltage signals; however, the computation unit cannot directly visualize as a human does. An NN is one suitable AI-based technique, which can be applied to classify the fault features. In addition, a classification technique using an NN offers an extra degree of freedom to solve a nonlinear problem: the failure of a single neuron will only partially degrade performance. If an input neuron fails, the network can still make a decision by using the remaining neurons. In contrast, if only a single input, for instance the dc offset of signals, is used as the input data to classify the faults, the diagnosis system may not perform classification when the input data has drifted or the single sensor has failed. Furthermore, an NN also permits parallel configuration and seasonal changes. Additional H-bridges and fault features (short circuit) can be conveniently extended into the system with more training data and parallel configuration.

Therefore, we will attempt to diagnose the fault types and fault locations in a cascaded MLID from its output voltage waveform.

III. FAULT DIAGNOSTIC METHODOLOGY

A. Structure of Fault Diagnostic System

The structure for a fault diagnosis system is illustrated in Fig. 8. The system is composed of the following four major states: 1) feature extraction; 2) NN classification; 3) fault diagnosis; and 4) reconfiguration (switching pattern calculation with gate signal output). The feature extraction performs the voltage input signal transformation, with rated signal values as important features, and the output of the transformed signal is transferred to the NN classification. The networks are trained with normal data, all fault feature data, and corresponding output assigned as binary code; thus, the output of this network is nearly 0 and 1 as binary code. The binary code is sent to the fault diagnosis to decode the fault type and its location. Then, the switching pattern is calculated to reconfigure the MLID to bypass and compensate the failed cell.

B. Feature Extraction System

Simulated and experimental output voltages of an MLID are illustrated in Figs. 5–7. 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 an NN for fault diagnosis tools is elucidated in [18]. The fault diagnosis system for an MLID using FFT and NN are proposed in [19]. 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, the principal
component analysis (PCA) can be used to reduce the number of input neurons as illustrated in Fig. 9. PCA is a method used to reduce the dimensionality of an input space without losing a significant amount of information (variability) [20]. The method also makes the transformed vectors orthogonal and uncorrelated. A lower dimensional input space will also usually reduce the time necessary to train an NN, and the reduced noise [by keeping only valuable principal components (PCs)] may improve the mapping performance. The comparison in classification performance between the network proposed in [18] and the PC-NN is discussed in [21]. The results show that the PC-NN has a better overall classification performance by 5% points; consequently, the PC-NN is utilized to perform the fault classification in this paper.

1) PCA: Basically, PCA is a statistical technique used to transform a set of correlated variables to a new lower dimensional set of variables, which are uncorrelated or orthogonal with each other. A distinguished introduction and application of PCA has been provided in [22]. The discussion of PCA presented in this section will be brief, providing only indispensable equations to elucidate the fundamental PCA approach applied to a fault diagnosis system in MLID. The fundamental PCA used in a linear transformation is illustrated as follows:

\[
T = X \cdot P \quad (1)
\]

where \( T \) is the \( m \times k \) score matrix (transformed data); \( m = \) number of observations, \( k = \) dimensionality of the PC space; \( X \) is the \( m \times n \) data matrix; \( m = \) number of observations, \( n = \) dimensionality of original space; and \( P \) is the \( n \times k \) loadings matrix (PC coordinates); \( n = \) dimensionality of original space, \( k = \) number of the PCs kept in the model. The original data matrix \( X \) of \( n \) variables (harmonic orders) and \( m \) observations (different modulation indexes of output voltage of MLID) is transformed to a new set of orthogonal PCs, i.e., \( T \), of equivalent dimension \( m \times k \), as represented in the following expression:

\[
\begin{bmatrix}
    t_{11} & t_{12} & \cdots & t_{1k} \\
    t_{21} & t_{22} & \cdots & t_{2k} \\
    \vdots & \vdots & \ddots & \vdots \\
    t_{m1} & t_{m2} & \cdots & t_{mk}
\end{bmatrix}
= \begin{bmatrix}
    x_{11} & x_{12} & \cdots & x_{1n} \\
    x_{21} & x_{22} & \cdots & x_{2n} \\
    \vdots & \vdots & \ddots & \vdots \\
    x_{m1} & x_{m2} & \cdots & x_{mn}
\end{bmatrix}
\begin{bmatrix}
    p_{11} & p_{12} & \cdots & p_{1k} \\
    p_{21} & p_{22} & \cdots & p_{2k} \\
    \vdots & \vdots & \ddots & \vdots \\
    p_{n1} & p_{n2} & \cdots & p_{nk}
\end{bmatrix}
\]

\[
(m \times k) = (m \times n) \cdot (n \times k) \quad (2)
\]
Fig. 6. Experiment of open-circuit fault of H-bridge 1 with modulation index = 0.8 out of 1.0. (a) Normal. (b) $S_{A+}$ fault. (c) $S_{A-}$ fault. (d) $S_{B+}$ fault. (e) $S_{B-}$ fault.

Fig. 7. Experiment of open-circuit fault of H-bridge 2 with modulation index = 0.8 out of 1.0. (a) $S_{A+}$ fault. (b) $S_{A-}$ fault. (c) $S_{B+}$ fault. (d) $S_{B-}$ fault.
original data set. The subsequent PCs are associated with the variance of the original data set in order; for instance, the second PC indicates the second highest variance of the original data set, and likewise.

Selecting a reduced subset (PCs kept in the model) of PC space results in a reduced dimension structure with respect to the important information available as shown in the following expression:

\[
[t_1 \ t_2 \ \cdots \ t_k] = [x_1 \ x_2 \ \cdots \ x_n] \cdot \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1k} \\ p_{21} & p_{22} & \cdots & p_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ p_{n1} & p_{n2} & \cdots & p_{nk} \end{bmatrix} 
\]

\[= (1 \times k) \cdot (1 \times n) \cdot (n \times k). \quad (3)\]

The objective of PC selection is not only to reduce the dimension structure but also to keep the valuable components. Normally, high variance components could contain related information, whereas small variance components that are not retained are expected to contain unrelated information; for instance, measurement noise. It should be noted that the high variance components may not contain the useful information for a classification problem.

2) PC Selection: The selection of PCs is significant because input selected PCs can cause the following uncertainty results:

1) additional unneeded input PCs to the NN can increase the solution variance and 2) absent necessary input PCs can increase bias. Therefore, a multivariable optimization technique such as genetic algorithm (GA) can be applied to search for the best combination of PCs to train the NN, as shown in Fig. 10.

The discrete GA or binary GA is utilized to select PCs. The idea is to randomly pass the PCs encoded as a binary string into the NN, and then a GA is used to search for the best combination of input PCs, which provides the NN with minimum classification error. The binary string of a gene consists of only one single bit. An example of encoded input PCs is illustrated on the right-hand side of Fig. 10. As can be seen, bit “0” will not be used to train the network, whereas others will be used to train the network. Then, the NNs (multilayer perceptron) are trained by using the Levenberg–Marquardt training paradigm \cite{23} with training set (simulation data), and after that, the test sets (experimental data) are examined. It should be noted that the test sets are not the same as the training set. Finally, the fitness function is evaluated by using the sum of square error (SSE), i.e.,

\[
SSE = \sum_{i=1}^{n} (y - \bar{y}_i)^2 \quad (4)
\]

where \( y \) is the output target binary codes, \( \bar{y}_i \) is the output of the training data, and \( n \) is the number of training data. The minimum SSE should provide the best combination of input PCs. More details in the GA parameter setup by using MATLAB have been explained in \cite{24}.

The fitness function is divided into two parts as follows: 1) SSE of the simulation set and 2) SSE of the experimental test set. In this paper, we weigh the experimental test set higher than the simulation test set because we only use the normal and fault data from the simulation to train the NNs. In addition, the classification performance as presented in \cite{19}–\cite{21} shows that the NNs have higher classification performance in the
TABLE I
CONFUSION TABLE FOR NEURAL NETWORK CLASSIFICATION PERFORMANCE

<table>
<thead>
<tr>
<th>Testing set</th>
<th>Target</th>
<th>Actual Output</th>
<th>% Classification</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>NN</td>
<td>PC-NN</td>
</tr>
<tr>
<td>Simulation</td>
<td>Normal [1 1 1]</td>
<td>1 1 1</td>
<td>1 1 1</td>
</tr>
<tr>
<td>test set</td>
<td></td>
<td>1 1 1</td>
<td>1 1 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 1 1</td>
<td>1 1 1</td>
</tr>
<tr>
<td></td>
<td>Fault A+ [0 0 1]</td>
<td>0 0 1</td>
<td>0 0 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 0 1</td>
<td>0 0 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 0 1</td>
<td>0 0 1</td>
</tr>
<tr>
<td></td>
<td>Fault A- [0 1 0]</td>
<td>0 1 0</td>
<td>0 1 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 1 0</td>
<td>0 1 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 1 0</td>
<td>0 1 0</td>
</tr>
<tr>
<td></td>
<td>Fault B+ [1 0 1]</td>
<td>1 0 1</td>
<td>1 0 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 0 1</td>
<td>1 0 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 0 1</td>
<td>1 0 1</td>
</tr>
<tr>
<td></td>
<td>Fault B- [1 1 0]</td>
<td>1 1 0</td>
<td>1 1 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 1 0</td>
<td>1 1 0</td>
</tr>
</tbody>
</table>

% Classification performance in simulation test set

| Normal [1 1 1] | 1 1 1 | 1 1 1 | 1 1 1 |
| Fault A+ [0 0 1]| 0 0 1 | 0 0 1 | 0 0 1 |
| Fault A- [0 1 0]| 0 1 0 | 0 1 0 | 0 1 0 |
| Fault B+ [1 0 1]| 1 0 1 | 1 0 1 | 1 0 1 |
| Fault B- [1 1 0]| 1 1 0 | 1 1 0 | 1 1 0 |

% Classification performance in experimental test set

| Normal [1 1 1] | 1 1 1 | 1 1 1 | 1 1 1 |
| Fault A+ [0 0 1]| 0 0 1 | 0 0 1 | 0 0 1 |
| Fault A- [0 1 0]| 0 1 0 | 0 1 0 | 0 1 0 |
| Fault B+ [1 0 1]| 1 0 1 | 1 0 1 | 1 0 1 |
| Fault B- [1 1 0]| 1 1 0 | 1 1 0 | 1 1 0 |

Total % Classification performance

<table>
<thead>
<tr>
<th></th>
<th>93%</th>
<th>95%</th>
<th>95%</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>85%</td>
<td>95%</td>
<td>100%</td>
</tr>
<tr>
<td></td>
<td>90%</td>
<td>95%</td>
<td>97.5%</td>
</tr>
</tbody>
</table>

C. NN Classification

The multilayer feedforward networks are used in this paper. The NN architecture designs have been proposed in [19] and will not be repeated here. As a comparison among transformation methods, FFT [19], PCA [21], and GA-PCA [24] will be performed, and three different NN architectures are used. The original data from the feature extraction system (FFT) that are used to train and test the networks are exactly the same data set. The first NN architecture [19] has one hidden layer with 40 input nodes, four hidden nodes, and three output nodes. The second NN architecture (PC-NN) [21] has one hidden layer with five input nodes, three hidden nodes, and three output nodes. The third NN architecture (PC-GA-NN) [24] is based on the GA selection, as previously discussed, because the input neurons depend on how many PCs were selected by GA. However, the hidden layer with three hidden nodes and three output nodes is used since the comparison among proposed NNs 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

\[
f = 0.2 \cdot \text{SSE}_{\text{Sim},\text{set}} + 0.8 \cdot \text{SSE}_{\text{Exp},\text{set}}. \tag{5}
\]
the specific application. The selection of number and dimension in the hidden layer is based on NN accuracy in preliminary tests.

The training data set should also cover the operating region, thus the training set is generated from simulation with various operation points (different modulation indexes such as 0.6, 0.7, 0.8, 0.9, and 1). The testing sets have two different sources: first, the test set is generated from simulation with modulation indexes of 0.65, 0.75, 0.85, and 0.95. Second, the test set is measured from experiment at different modulation indexes of 0.7, 0.8, 0.9, and 1.

The performance of the proposed networks is 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 I.
Clearly, the classification performance of PC-GA-NN for experimental data 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 an NN. In addition, the reduced noise could improve the mapping performance, which leads to the improvement of the total classification performance. GA offers the multivariable search of the minimum misclassification error, providing the better NN performance.

It should be mentioned that the feature extraction and NN classification process are performed offline. After the NN is ready to use, the weight and bias matrices are saved. These weight and bias matrices will be used to perform a fault classification that is incorporated by a Simulink model by using the gensim command [23]. The gensim command will automatically generate the NN simulation blocks for use with Simulink, as shown in Fig. 11.

IV. RECONFIGURATION PARADIGM

A. Corrective Action Taken

The basic principle of the reconfiguration method is to bypass the faulty cell (H-bridge); then, other cells in the MLID are used to compensate for the faulty cell. For instance, if cell 2 of MLID in Fig. 2 has an open-circuit fault at $S_{A+}$, accordingly, $S_{A-}$ and $S_{B-}$ need to be turned on (1), whereas $S_{B+}$ needs to be turned off (0) to bypass cell 2. The corrective actions taken for other fault locations are shown in Table II. As can be seen, the corrective action would be the same for cases that have similar voltage waveforms during their faulted mode (for instance, see Fig. 5 for a short-circuit fault in $S_{A+}$ and open-circuit fault in $S_{A-}$). Therefore, even if the fault
Fig. 15. Fault diagnostic system interfaced with PSIM performing power circuit of an MLID.

may be misclassified (an actual short-circuit fault at $S_{A+}$ is misclassified as an open-circuit fault at $S_{A-}$ or vice versa), the corrective action taken would still solve the problem.

**B. Reconfiguration Method**

The reconfiguration diagram for an 11-level MLID with five SDCS is illustrated in Fig. 12. The turn-on intervals of each cell are not equal with the multilevel carrier-based sinusoidal PWM: cell 1 has the longest turn-on interval, then the turn-on interval decreases from cell 2 to cell 5 as a staircase PWM waveform. The desired output voltage of an MLID can be achieved by controlling the modulation index $m_a$. For instance, suppose cell 2 has an open-circuit fault at $S_1$, while the MLID operates at $m_a = 0.8/1.0$ [MLID is operated with four cells (cells 1–4)]. We can see from Fig. 12(b) that $S_3$ and $S_4$ need to be turned on, then the gate signal of cell 2 will be shifted up to control cell 3, then the gate signal of cell 3 will shift to cell 4, and the gate signal of cell 4 will shift to cell 5, respectively.

This reconfiguration also applies to other phases of MLID in order to maintain a balanced output voltage. By using this method, the operation of MLID in a modulation index range of 0.0 to 0.8 (out of 1) can be fully compensated such that the inverter will continue to function as in a normal operation; however, if MLID operates at $m_a > 0.8$ and has a fault, lower order harmonics will occur in the output voltage since the MLID will
operate in the overmodulation region in order to output the full requested voltage as illustrated in Fig. 13. The compensated gain of the MLID operating at $m_a > 0.8$ is shown in Fig. 14. This compensated gain can also be written as a function of $m_a$ by using a polynomial curve fitting. Because the overmodulation region has a nonlinear relationship between the modulation index and the output fundamental voltage, the compensated gain is calculated in particular modulation indexes; then, the polynomial function represents the nonlinear characteristic of this particular application. In addition, this polynomial function can be implemented in a Simulink model. The fitting function can predict the compensated gain with a norm of residuals less than 0.09. We can see that the overmodulation region will occur when the MLID operates at $m_a > 0.825$. To relieve this problem, space vector and third harmonic injection PWM schemes may be used. In addition, a redundant cell can be added into the MLID, but the additional part count should be considered. The reconfiguration effect and limitation of this reconfiguration method have been reported in [25].

V. SIMULATION AND EXPERIMENT VALIDATION

A. Simulation Setup

The following two simulation programs are used in the simulation setup: 1) Matlab–Simulink and 2) PSIM. Matlab–Simulink is used to implement feature extraction (FFT and PCA), NN classification, and reconfiguration. PSIM is used to implement the MLID power circuit. The reason for using PSIM is that the PSIM is a circuit-based simulation and conveniently interfaces with Matlab–Simulink via the toolbox called Simcouple [26]. The simulation validation based on Simulink is illustrated in Fig. 15. It should be noted that the same Simulink model is used in both simulation and experiment.

B. Experimental Setup

The experimental setup is represented in Fig. 16. A three-phase wye-connected cascaded multilevel inverter using 100-V 70-A metal–oxide–semiconductor field-effect transistors as the switching devices is used to produce the output voltage signals. The MLID supplies an induction motor (1/3 hp) coupled with a dc generator (1/3 hp) as a load of the induction motor. The Opal RT-Lab system [27] is utilized to generate gate drive signals and interfaces with the gate drive board. The switching angles are calculated by using Simulink based on multilevel carrier-based sinusoidal PWM. A separate individual power supply acting as SDCS is supplied to each cell of the MLID, consisting of five cells per phase as shown in Fig. 12(b). Open- and short-circuit fault occurrences are created by physically controlling the switches in the fault-creating circuit. A Yokogawa DL 1540c is used to measure output voltage signals as ASCII files. The measured signals are set to $N = 10032$; sampling frequency is 200 kHz. Voltage spectrum is calculated and transferred to the Opal-RT target machine.

C. Results

1) Open-Circuit Case: The simulation of open-circuit fault occurrences is created by using a faulty power cell as shown in Fig. 17: the auxiliary switches ($F_1$ and $F_2$) are normally close type; then, the faulty cell will be simulated by disconnecting switch $S_1$ at time $T$ (by controlling $F_1$ and $F_2$) commanded by a unit step from Simulink. This faulty power cell is in cell 2 of phase A (see Fig. 1), and the MLID is operating at 0.8/1.0 modulation index before the fault occurs. In the experiment, an open-circuit fault occurrence is created by physically turning on the switch in the fault-creating circuit. The simulation and experimental results of an open-circuit fault in cell 2 switch $S_1$ are represented in Fig. 18. We can see that the simulation and experimental results agree with each other. The fault diagnostic system requires about six cycles ($\sim$100 ms at 60 Hz) to clear the open-circuit fault. Obviously, the open-circuit fault causes an unbalanced output voltage $V_{an}$ of the MLID during the fault interval, and the average current in phase A, i.e., $I_a$, has a negative polarity during the fault interval.

2) Short-Circuit Case: The simulation of short-circuit fault occurrences is created by using a faulty power cell as shown in Fig. 19. The auxiliary switches are normally open type;
Fig. 18. Results of the open-circuit fault at cell 2 $S_1$ of the MLID that is operated at $m_a = 0.8/1.0$. (a) Simulation of current waveforms. (b) Experiment showing the line current $I_a$ in the faulty phase.

then, the faulty cell will be simulated by closing switch $F_1$ at time $T$ commanded by a unit step from Simulink. This faulty power cell is in power cell 3 on phase A (see Fig. 1), and the MLID is operating at 0.8/1.0 modulation index before the fault occurs.

The simulation results of a short-circuit fault at cell 3 switch $S_1$ are represented in Fig. 20. The fault diagnostic system also requires about six cycles to clear the short-circuit fault. Obviously, the output voltage $V_{an}$ of the MLID is unbalanced during the fault interval (lost negative voltage at phase A), and the average current in phase A, i.e., $J_a$, has a positive polarity during the fault interval. The peak of the fault current increases to about 1.5 times compared with the normal operation. It should be noted that, practically, the fuse protecting the SDCS may blow (disconnect the SDCS from an MLID) before the diagnostic system performs fault clearing so that the output phase

Fig. 19. Short-circuit faulty power cell at $S_1$. 
Fig. 20. Simulation results of the short-circuit fault at cell 3 $S_1$ of the MLID that is operated at $m_a = 0.8/1.0$.

Fig. 21. Results of the short-circuit fault at cell 3 $S_1$ under loss of SDCS condition at the faulty cell of the MLID that is operated at $m_a = 0.8/1.0$.
(a) Simulation. (b) Experiment showing the line current $I_a$ in the faulty phase.
voltage will be zero. This behavior of output phase voltage signals should also be taken into account for training the NN.

The proposed diagnostic system can also detect a short fault under loss of SDCS at the faulty cell condition, as shown in Fig. 21. The clearing time for this particular case is about nine cycles. In addition, the NN can detect which cell has a fault and whether the switch was connected to the positive bus (S1 or S2) or the negative bus (S3 or S4). However, the NN could not determine which specific switch (S1 or S2) or (S3 or S4) had failed. Nevertheless, the proposed corrective action taken would still solve this problem.

The clearing time of the short-circuit fault under loss of SDCS at faulty cell condition is longer than the open-circuit and short-circuit faults by about three cycles. This result suggests that using only output voltage signals in the loss of SDCS case may not adequately provide a unique feature to detect the faults. Therefore, the current signals may be required to additionally train the NN because Fig. 4 shows that the current polarity of the faulty cell can be used to classify the faults as occurring either at the positive or the negative dc bus.

The clearing time of the proposed system can be shorter than this if the proposed system is implemented as a single chip using a field-programmable gate array or digital signal processor. The Opal-RT system needs a few cycles to load the output voltage signals from the target machine to the console PC machine via Ethernet. In addition, the window of FFT function requires at least a cycle to perform signal transformation. The clearing time could be within two cycles if the algorithm is implemented with an embedded single chip, which we will report in the near future. However, if the cascaded MLID can tolerate a few cycles of faults, the proposed system can detect the fault and can correctly reconfigure the MLID; therefore, the results are satisfactory.

VI. CONCLUSION

A fault diagnostic and reconfiguration method for a cascaded H-bridge MLID using AI-based techniques has been proposed. Output phase voltages of the MLID can be used as diagnostic signals to detect faults and their locations. The proposed fault diagnostic paradigm has been validated in both simulation and experiment. The results show that the proposed fault diagnostic technique performs quite satisfactory. The fault diagnostic system requires about six cycles (~100 ms at 60 Hz) to clear the open-circuit and short-circuit faults and about nine cycles to clear the short circuit under loss of SDCS at faulty cell condition. The experimental results show that the proposed diagnostic and reconfiguration system performs satisfactorily to detect the fault type, fault location, and reconfiguration.

REFERENCES

Surin Khomfoi (S’03–M’07) received the B.Eng. and M.Eng. degrees in electrical engineering from King Mongkut’s Institute of Technology Ladkrabang, Bangkok, Thailand, in 1996 and 2000, respectively, and the Ph.D. degree in electrical and computer engineering from the University of Tennessee, Knoxville, in 2007.

From 1996 to 1997, he was with the Engineering Division, Telephone Organization of Thailand (TOT). Since December 1997, he has been a Lecturer with the Department of Electrical Engineering, Faculty of Engineering, King Mongkut’s Institute of Technology Ladkrabang. His research interests include multilevel power converters, ac drives, diagnostic system, fault diagnosis, and in particular artificial intelligence-based techniques applied to power electronics and drive applications.

Dr. Khomfoi is a member of the Eta Kappa Nu Honor Society. He was a recipient of academic scholarship awards, including the full academic scholarship for his B.Eng. degree from TOT and the full academic scholarships for his M.Eng. and Ph.D. degrees from the Energy Policy and Planning Office, Thailand.


He joined the Engineering Division, Lockheed Martin Energy Systems in 1991 and worked on several electrical distribution projects at the three U.S. Department of Energy plants in Oak Ridge, TN. In 1997, he became a Research Engineer with the Power Electronics and Electric Machinery Research Center, Oak Ridge National Laboratory. In 1999, he was appointed as an Assistant Professor with the Department of Electrical and Computer Engineering, University of Tennessee, Knoxville, where he is currently an Associate Professor. He is an adjunct participant at the Oak Ridge National Laboratory and conducts joint research at the National Transportation Research Center. His research interest is focused on the areas of electric power conversion for distributed energy sources, motor drives, multilevel converters, hybrid electric vehicles, and application of SiC power electronics.

Dr. Tolbert is a Registered Professional Engineer in the State of Tennessee. He has been the Chairman of the Education Activities Committee of the IEEE Power Electronics Society since 2003. He was an Associate Editor for the IEEE POWER ELECTRONICS LETTERS from 2003 to 2006. He was the recipient of a National Science Foundation CAREER Award and the 2001 IEEE Industry Applications Society Outstanding Young Member Award.