



**HAL**  
open science

# Wound Rotor Induction Generator Inter-Turn Short-Circuits Diagnosis Using a New Digital Neural Network

Samuel Toma, Laurent Capocchi, Gerard-André Capolino

► **To cite this version:**

Samuel Toma, Laurent Capocchi, Gerard-André Capolino. Wound Rotor Induction Generator Inter-Turn Short-Circuits Diagnosis Using a New Digital Neural Network. IEEE Transactions on Industrial Electronics, 2013, 60 (9), pp.4043 - 4052. 10.1109/TIE.2012.2229675 . hal-01083108

**HAL Id: hal-01083108**

**<https://hal.science/hal-01083108>**

Submitted on 4 Jan 2016

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Wound Rotor Induction Generator Inter-Turn Short-Circuits Diagnosis Using a New Digital Neural Network

S. Toma, L. Capocchi, G.A. Capolino, *Fellow, IEEE*

**Abstract**—This paper deals with a new transformation and fusion of digital input patterns used to train and test feed-forward neural network for a wound rotor three-phase induction machine windings short-circuits diagnosis. The single type of short-circuits tested by the proposed approach is based on turn-to-turn fault which is known as the first stage of insulation degradation. Used input/output data have been binary coded in order to reduce the computation complexity. A new procedure, namely addition and mean of the set of same rank, has been implemented to eliminate the redundancy due to the periodic character of input signals. However, this approach has a great impact on the statistical properties on the processed data in terms of richness and of statistical distribution. The proposed neural network has been trained and tested with experimental signals coming from six current sensors implemented around a set-up with a prime mover and a 5.5kW wound rotor three-phase induction generator. Both stator and rotor windings have been modified in order to sort out first and last turns in each phase. The experimental results highlight the superiority of using this new procedure in both training and testing modes.

**Index Terms**—Induction generators, Fault diagnosis, Feed-forward neural network, Digital measurements, Back-propagation, Stator current, Rotor current, Winding short-circuits, Data pre-processing.

## NOMENCLATURE

$Y_j$	Output of neuron $j$ .
$X_i$	Input of neuron $i$ .
$f$	ANN transfer function.
$L$	ANN layer.
$CR$	Compression ratio.
$N_S$	Number of turns of each stator winding.
$N_R$	Number of turns of each rotor winding.

## I. INTRODUCTION

TECHNIQUES for condition monitoring and fault detection of electrical machines have been developed for the last 80 years by starting from human analysis up to modern

S. Toma and L. Capocchi are with the SPE Laboratory (UMR CNRS 6134), University of Corsica, Campus Grimaldi, 20250, Corte, France, e-mail: {toma, capocchi}@univ-corse.fr

G.A. Capolino is with the Laboratory of Innovative Technologies (EA3899), University of Picardie “Jules Verne”, 80000 Amiens, France, e-mail: Gerard.Capolino@ieee.org

decision processes. However, it is only recently that these techniques have been integrated in complex electromechanical systems in which electrical machines are used as prime movers (motors) or prime receivers (generators). In this way, the squirrel-cage three-phase induction machine which is without contest the most interesting machine in term of cost and reliability for the industry has been under focus for the last 20 years. Recently, many review papers have been published to give the state of the art for diagnostics techniques around induction machines mostly related to electrical faults [1], [2]. In these papers, all problems dealing with electrical fault detection in electrical machines have been addressed from modeling up to the decision process through the instrumentation, the signal processing and the applications on real complex systems. As applications have to be under focus for condition monitoring and fault detection, wind turbine generators are key elements in which electrical fault detection is fundamental. In this way, many actual projects are dealing with monitoring techniques of wind farms and some of them have been finalized even with standards. However, the technology is still in evolution and it is very hard to find general and efficient techniques addressing all the reliability objectives [3]. In fact, in modern wind farms, a large majority of generators are based on wound rotor induction machine (WRIM) technology with a low number of poles and a planetary gearbox to adapt the machine rotor shaft speed to the blades speed. A large number of papers dealing with control of wound rotor induction generators (WRIG) have been published but very few on fault detection and localization [4]–[6]. By going back to signal processing techniques allowing to characterize the different electrical faults, they have been qualified of complex and it is always difficult to perform a clear conclusion on any fault associated to time, frequency or time-frequency analysis. In this way, decision techniques mainly based on artificial intelligence (AI) have been used for a while in order to help for the maintenance process on any type of electrical machine [2]. Neural networks have been used in association with diagnostics techniques for almost twenty years [7] and they have been applied to three-phase induction machines rotor faults with success [8]–[10]. In this last case, artificial neural networks (ANNs) have been used to classify different frequencies according to the fault signatures. More recently, ANNs with training algorithms have been implemented with success in the decision process for simple faults in three-phase induction machines [11], [12]. The main problem for the training process is to perform this task in the real world

in order to obtain a good classification and to avoid false alarms or to miss obvious faults. In almost all of the published techniques, ANNs have been used as supervisors of classical signal processing techniques and have been separated of the signal analysis.

In this paper, a classical ANN with back-propagation has been used to be trained directly by signals coming from sensors to eliminate the analysis mode and to simplify the architecture of diagnostics systems. For this purpose, it has been proposed to train a dedicated ANN directly with digital signals coming from the sensors implemented around electrical machines under condition monitoring. It is not so original in term of ANN since digitizing tasks have been imagined long time ago [13]–[15]. It has been even used recently for multiple-fault detection in steam turbines [16]. However, this technique has never been implemented for electrical machines fault detection as well as for power systems condition monitoring.

In recent years, other interesting papers on the usage of ANNs for power systems condition monitoring have been published. The aim was always to add a sort of 'intelligence' to fault detection by superposing a layer for the decision process [17]. Therefore, the Elman network and the back propagation algorithm have been used to build dedicated ANNs [18]–[20]. In these previous cases, the different ANNs are mainly considered as classifiers and the computation process is performed by other techniques. To the best of our knowledge, only one work was fully dedicated to the ANN usage in order to protect windings of large power turbo-generators [21].

However, in many recent applications, ANNs have been efficiently used in association with power systems and electrical machines for identification [22], control [23], fault location [24], diagnosis [25], fault detection and classification [26], [27].

The basic idea is to make diagnostics systems more economical and more reliable to be used at a large scale in modern wind farms in which the predictive maintenance is a crucial operation.

On the other hand, short-circuits treated in this paper are almost taken at their early stage with one or two shorted turns in each phase and in each side (stator or rotor). The aim is not to detect or to localize simultaneous faults although one or two turns by phase and by side have been tested. The early stage detection is very important before the thermal propagation in the machine windings. It is well known that any AC electrical machine can still operate with a couple of shorted turns even at rated operation mode. The interest of the early stage detection is clearly based on the need of predictive maintenance avoiding full windings failures. Therefore, this paper will address problems of data collection and scaling for the ANN inputs and it will be used on a set-up able to perform both stator and rotor turn-to-turn short-circuits for a three-phase WRIG similar to the ones used in wind turbine generators. The first part of the paper is dedicated to the description of the proposed feed-forward ANN technique before trying the system on real data coming from a reduced scale set-up. Different problems of input-output data scaling, processing convergence approach and pattern recognition have

been presented to demonstrate the efficiency of the proposed technique. The different tests have been performed on a 5.5kW, 50Hz, 220V/380V, 8 poles WRIG with several sets of one short-circuited turn fault on both stator and rotor sides. For this purpose, a specific set-up has been developed in order to collect digital data from three stator phase currents sensors and three rotor phase current sensors.

## II. BACKGROUND

ANNs are a type of computer program inspired from the human brain. It uses to process information such as learning and generalization procedures and emerged by developing software in actual technology with successful applications in many fields. The majority of these applications is concerned with problems of both pattern recognition and classification. The knowledge of the ANN is gathered by detecting patterns and data relationships through a training process and by no way from programming [28]–[32]. Any ANN is build with layers and the minimum number of layers is two and without any maximum but usually four or five is considered as the maximum. Each layer is build from a number of processing units called neurons. The power of the neural computation comes from the connections between neurons in different layers. There exist many types of connections and their choice depends on the network architecture. Each neuron is composed of a weighted input, a bias, a transfer function and one output. The sum of the weighted input list and the bias creates a value which passes through the activation function and which produces the single output of the neuron. A transfer function introduces a non-linearity in the network. The only purpose to train a neural network is to modify the different weights of each neuron in order to evaluate relationships in between inputs and outputs. Two indicators can evaluate if one specific network is well trained or not. The first is the training error and the second one is the validation error. The neural network is considered as well trained only if both training and validation errors are converging to a small value close to zero. Once the network is trained and tested, a set of new inputs can be applied and it can predict the corresponding output

### A. Network Architecture

There are many different types of ANNs, some of them being more popular than others. So it is important to distinguish between network architectures (topology) and network algorithms (computational methods). In 1940, the first idea of the neural network came out with only two layers (input and output layers) and one weighted layer (output layer) [30]. This single layer of adaptive weights is able to solve simple functions and problems with very important limitations. To allow flexible capabilities to the neural network, more adaptive weighted layers have been added. When no loop connections in between neurons exist in the network, this type of connection can be called a feed-forward network [28]. In most cases, a three-layer architecture (two layers of adaptive weights) feed-forwardly connected can solve many non-linear problems which makes it one of the most widely used architectures.

## B. Network Algorithms

ANNs can learn through two main types of both supervised and unsupervised algorithms [28]. Unsupervised ANNs are used in order to analyze data without having a desired output. This type of learning is used for data clustering, feature extraction, and similarity detection [30], [31]. Usually, supervised learning algorithms are much more used than unsupervised ones. The supervised learning depends on changing every weight in the network in order to meet a desired output for a specified pattern.

To reach this goal, many computational methods have been used to adapt weights after running a significant number of iterations. In this case, the most common method is the back-propagation algorithm. It consists in two phases: (i) the feed-forward phase, which is the propagation of the input vector through the network and (ii) the back-propagation phase which begins to compare the desired output to the real output of the network and then recalibrate the network based on this error. [7], [28], [30].

## C. Feed-Forward Neural Network

The performance of any ANN depends on its architecture, on the training algorithm, on the presentation of data, on the analysis of the output as well as on the timing to stop the training mode. In order to appreciate the effect of one of these configurations, all of them must be fixed and only one can be changed at the same time.

The importance of the good presentation of data to the ANN is the main goal of this paper. The preparation of input data to the ANN and the analysis of its output are labeled as pre-processing and post-processing modes respectively [29], [31].

The proposed ANN is built by using pre-processing blocks and feed-forward architecture (Fig. 1). In this ANN, each layer consists in  $n$  neurons ( $n : 1, \dots, j$ ) and  $x$  inputs ( $x_i : 1, \dots, m$ ). The first input of each layer is a bias input (typical values can be equal to 0.5 or 1). The first layer is the input layer. The  $L^{th}$  layer with nodes is the output layer. If  $m$  inputs and  $n$  neurons are available in the first hidden layer, the output of each of these neurons ( $y$ ) will be calculated by:

$$u_j^{[s]} = f(u_j) \quad (1)$$

Where  $u_j$  is calculated by:

$$u_j = \sum_{i=0}^m w_{ji}^{[s]} x_i \quad (2)$$

where  $j = 1, \dots, n$ .

The importance of mixing the pre-processing and the post-processing modes in order to obtain the best performances in the classification of faults is very important. Among  $n$  steps, two fundamental ones are usually implemented for any type of data to prepare the input vector for the dedicated ANN: (i) centering by subtracting the average value, (ii) scaling by dividing by its standard deviation. The problem of using original data without pre-processing is that the dimension of the input vector is very large in front of the small size of target

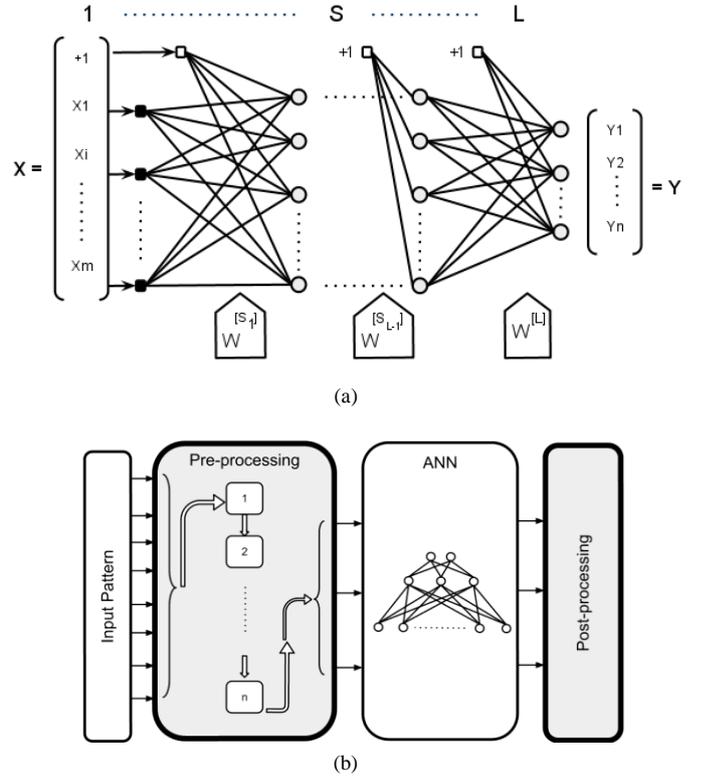


Fig. 1. General configuration of an ANN: a) ANN architecture - b) Pre-processing and post-processing units.

data. The post-processing mode is always an important task which has been often pointed out without many details on its implementation.

In order to use this ANN as a patterns classifier, the post-processing block can be limited to a binary number for each pattern [7]. A feed-forwardly connected ANN with a back-propagation algorithm can be chosen to perform efficient tests as it is the most common and successful design for any ANN. Common activation functions such as sigmoid or hyperbolic tangent will be used by all layers of the ANN. After fixing all radical parameters. By choosing all these parameters and all these configurations, the designed ANN will be transformed in a 'black box' which receives input data and which gives output data making a model to test different pre-processing algorithms. This ANN has only a standard configuration, and this might not be the optimal configuration. The standard configuration is used only for a comparison between different approaches. Another studies could take place after verifying this digital pre-processing approach to find a better ANN auto-configuration and to apply a post-processing approach.

## D. DEVSimPy Software for ANN Modeling

DEVSimPy (Python Simulator for DEVS models) [33], [34] is a user-friendly interface for collaborative modeling and simulation of discrete event systems implemented in Python. Python is a programming language known for its simple syntax and its capacity to allow designers to implement quickly their models. The DEVSimPy project used the Python language and provides a GUI based on PyDEVS [35] application program

interface (API) in order to facilitate both the coupling and the re-usability of PyDEVS models. DEVSImPy uses the wxPython graphic library and is an open source project under GPL V3 license. Its development is supported by the SPE UMR6134 research laboratory team. As is shown in Figure 2, DEVSImPy is used to provide a model and to simulate systems by using a library of basic models in Python. The different libraries (part one in Fig. 2) are composed with models which can be used to describe systems like ANNs or Power System. Once the model has been built (part two in Fig. 2), DEVSImPy allows the simulation in an automated way. The results coming from the simulation can be displayed using embedded tools like graphs or diagrams (part three in Fig. 2).

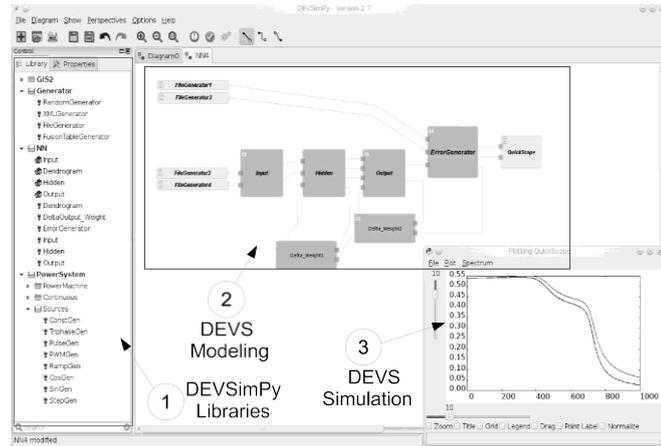


Fig. 2. DEVSImPy cockpit.

The aim of the proposed technique is to implement a new way to configure and to use ANN for wound rotor induction generator fault classification with the DEVS formalism implemented in Python language. DEVS is a formalism allowing a high behavioral level description of a discrete event system with a modular and hierarchical approach. DEVSImPy provides a framework for the DEVS modeling in a graphical way and allows the implementation of re-usable models available in libraries. MATLAB/Simulink <sup>c</sup> has not yet any DEVS formalism in its libraries. With DEVSImPy, the feed-forward ANN algorithm has been divided in sub-parts in order to insulate and to control the learning and testing modes in an efficient way.

### III. PROPOSED DIGITAL PRE-PROCESSING APPROACH

For many applications, it is necessary to transform data into new representations before training any ANN. In many cases, the choice of the pre-processing technique will be one of the most significant factors in achieving performances of the completed system [7], [31]. For the application presented herewith, data coming from sensors around the electrical machine under test will be composed of six current analog signals: three for the rotor side and three for the stator side of the three-phase WRIG. On the other hand, data length of 10s with a sampling period of 0.1ms (10kHz sampling frequency) has been recorded. The current time has been represented in rows and the six variables in columns giving

a single data matrix of 100,000 rows and six columns. The analog currents data oscillate in the interval  $[-23A,+23A]$  and they are almost periodic at least during the steady state. Moreover, the period of the stator currents in steady state is different from the one of the rotor currents. As it has been evaluated during the training phase, the ANN exhibits poor results due to redundant data coming from the periodic character of all signals. As said before, periodicity changes considering the side on which they are measured. Therefore, there is a high frequency (grid frequency) from the stator side and a low frequency (controlled by the power converter) on the rotor side. Moreover, additive frequency components coming from windings faults are superposed to the existing main frequencies. Usually, electrical fault signatures appear as high frequency components in both stator and rotor currents. These fault frequency components put constraints on the number of periods used in order to train the ANN. A sampling period of 0.1ms and 200 points per period for the stator side has been used as a compatible factor for a grid frequency of 50Hz. Of course, for the rotor side the number of periods is not constant due to the slip and it is lower than for the stator side due to the load and the related slip frequency. It has been decided to use 2000 points (ie. 10 periods of the stator current at steady state) for each of the six currents to represent the input of the ANN for one pattern. The number of training patterns used for the learning mode has a significant impact on the performance of any classifier. By using larger training sets, it leads to a better performances and a slower learning rate. On the contrary, by using low number of sets, it leads to lower performances and a faster learning rate [31]. Therefore, it has been decided to use a reasonable number of sets tuned at 10 units for each fault on both stator and rotor sides.

#### A. Linear Transformation

One of the most common steps of the pre-processing mode consists in a simple linear re-scaling of input variables. This re-scaling process of data tunes them in the correct interval, centers them on the activation function range and avoids also the saturation of output signals. As said before, the six currents oscillate in the magnitude interval  $[-23A,+23A]$  making easy the normalization process by dividing every single value by 23. By applying a linear transformation, all of the inputs will be in a similar range. The normalization ensures that the ANN inputs, outputs and weights are in the range of unity at the beginning which helps a lot in the initialization of weights [7].

#### B. Digital Compression

By having a large dimension of input data, this leads to a large number of weights inside the ANN which imposes automatically a long learning period of time. The motivation for reduction of this dimension is that it can help in decreasing the size of the ANN configurable parameters and consequently the number of inputs. An ANN with a low number of inputs can learn faster than the same ANN with more input and it can increase its global performances and its output accuracy which is the most important parameter for classification features.





### C. Learning and Test Patterns

The ANN learning pattern consists in values of three stator currents and three rotor currents when the WRIG is operating in both healthy and faulty modes at steady state.

Let  $x$  be the number of faults considered by the ANN and  $y$  an integer such that  $2^y > x$ . The number of vectors in one learning pattern before pre-processing mode is  $10 \times (1 + x)$  or 10 which correspond with the acquisition time in seconds and 1 for the healthy mode simulation. The dimension of these vectors is equal to the sum of the number of signals and the number of bits to differentiate  $x$  faults whose effects are recorded in the learning pattern signals. Therefore, the dimension of the learning vectors is  $2k \times 6 + y$ . In summary, the pattern of the real learning is a matrix of  $10 \times (1 + x)$  rows and  $12k + y$  columns.

The pre-processing mode described in the previous section is crucial to improve the convergence of the proposed ANN. The goal of the pre-processing mode is to extract the information from data (2k periodic input data) and to present it with the minimum number of inputs to the ANN without losing necessary information. The proposed pre-processing technique can be summarized within 2 steps: (i) coding of real data with four decimal digits in 16-bit binary values and (ii) compression of columns of the new matrix binary package of 2000 points. The size of the learning pattern is now binary  $\frac{10 \times 2000 \times (1 + x)}{2000}$  ie  $10 \times (1 + x)$  rows and  $16 \times 6 + y$  columns with  $x$  and  $y$  similar to the previous values.

It has been chosen to train the ANN with 10 sets of 2000 points among the 100,000 available items. Moreover, in order to avoid the transient period, the acquisition of 20,000 points has been shifted of 20,000 steps.

TABLE I  
SIZE OF LEARNING PATTERNS.

	without pre-processing	with pre-processing
row (size of pattern)	$10(1 + x)$	$10(1 + x)$
column (ANN inputs)	$12k + y$	$96 + y$

Table I shows that the input size (column) with a pre-processing process is much smaller than without it regardless of  $x$ . The number of ANN patterns (row) is the same when the pre-processing process is not performed with the same fault number because rows represent the number of patterns used to train the ANN. The number of ANN inputs is equal to the number of neurons inside the input layer. The re-scaling of input data will reduce the overall number of neurons inside the ANN. Therefore, it can be seen that the ANN learning phase by using data compression will be faster than without modifying data.

The quality of a learning pattern can be measured through its Gaussianity [7]. The distribution of learning pattern normalized occurrences has been shown in cases of the proposed pre-processing technique and without it for comparison purposes (Fig. 7). After the compression, the distribution of the learning pattern is a Gaussian characteristic centered on 0.5. On the contrary, the distribution of the learning pattern without compression does not present the same characteristic. The data distribution affects directly the behavior of the activation

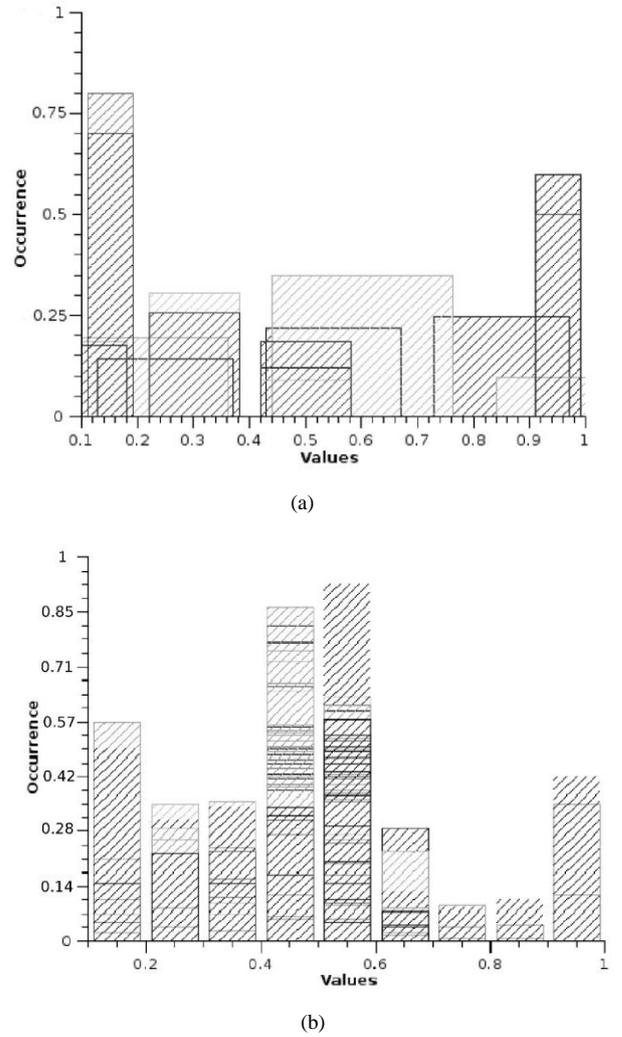


Fig. 7. Statistics for learning patterns: a) without pre-processing. - b) with pre-processing.

function. When the data is centered and uniform (Gaussian distribution) the activation function is applied on the maximum data range ensuring an optimum efficiency of the ANN. The testing pattern has been selected from 100,000 points and its size has been represented by 10 vectors injected during the learning phase. If  $x$  is the number of faults tested, the learning pattern will be a matrix with  $10 \times (1 + x)$  rows and 96 columns in the case of a pre-processing process. It has  $10 \times (1 + x)$  rows and  $2k \times 6$  columns when the pre-processing process is not used. The post-processing phase has not been operated but a right approach is to use a bitstream for the ANN output to differentiate the class of faults in an efficient way.

### D. Neural Network Simulation

In order to show the importance of the proposed compression technique, two ANNs have been used. The first ANN has  $2k \times 6 + y$  inputs for the learning phase without pre-processing process. The second ANN has  $16 \times 6 + y$  inputs for the learning phase with the preprocessing process.

Concerning the configuration of these two ANNs:

- Based on trial and error tests to find the best configuration for the WRIG fault classification [36], both learning and momentum factors (named N and M) have been chosen respectively at 0.1 and 0.9.
- The number of neurons inside the input layer is related to the input data, which gives 2000 neurons for the network without digital compression and 96 neurons when digital compression is applied. The number of neurons in the hidden layer is 56, which is an acceptable number than the common rule  $n/2 + 1$  where n is the number of input data. The number of neurons of the output layer depends on the number of faults to be considered. For 12 faults, 4 binary outputs can be considered as the minimum.

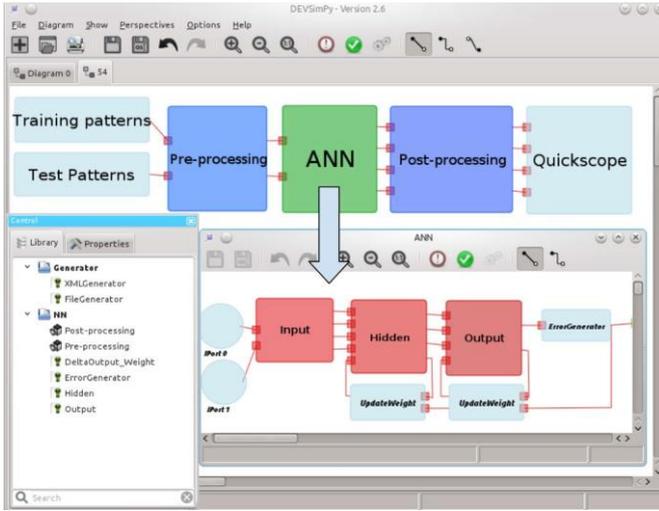
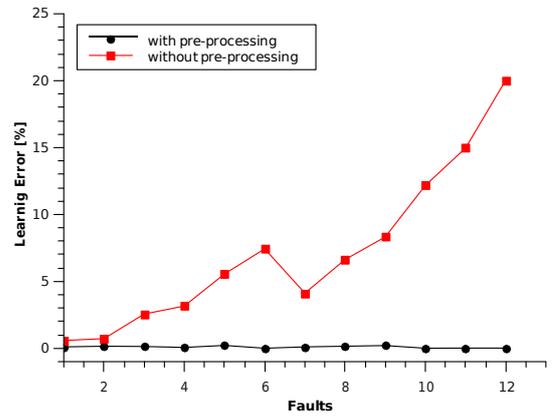


Fig. 8. ANN DEVSIMPy modeling with pre- and post-processing.

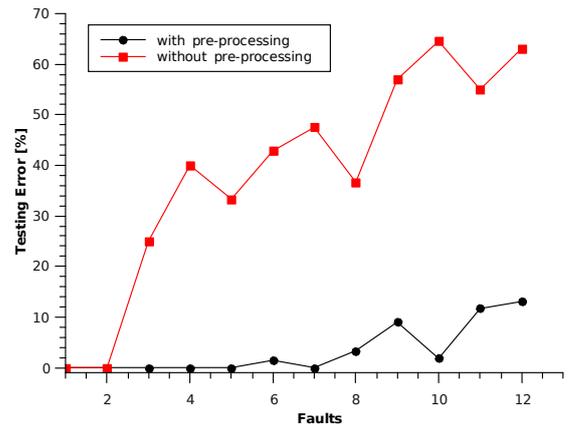
For modeling and simulation purposes, the DEVSIMPy GUI software has been used in order to implement ANN, pre-processing and post-processing algorithms.

To implement the ANN model with the proposed pre-processing method, the DEVSIMPy ANN library has been used. This library is composed by six models that will help to build a full ANN: input, output, hidden layers, error calculators and input file (Fig. 8). The input file is a model that extracts data from a file and gives it to them network in form of patterns.

Figure 9(a) and 9(b) shows both learning and testing error progresses depending on the number of considered faults. When the number of considered faults increases, the percentage of the learning error decreases when the pre-processing is used. The same phenomenon can be observed on the testing error. It can be seen that the testing error converges always and it is around to 4% when the pre-processing process is implemented. Table II shows the test results with the ANN already trained to classify all the 12 faults. The detection error represents the percentage of undetected faults on 10 tested patterns. For example, for the fault F2 has been detected on all the 10 patterns. On the contrary, the fault F7 has shown only three detection on all the test pattern. The difference is due to the fact that the fault F7 is almost similar to some other faults.



(a)



(b)

Fig. 9. Comparison of learning and testing errors: a) Learning error. - b) Testing error.

TABLE II  
RESULTS ON CLASSIFICATION OF THE 12 FAULTS.

Fault type	Desired output	Real output	Detection error [%]
Healthy	0000	0.08, 0.05, 0.03, 0.04	0.0
F1	0001	0.07, 0.09, 0.04, 0.90	0.0
F2	0010	0.07, 0.05, 0.98, 0.04	0.0
F3	0011	0.05, 0.15, 0.91, 0.93	0.0
F4	0100	0.09, 0.93, 0.10, 0.08	20.0
F5	0101	0.06, 0.80, 0.26, 0.96	40.0
F6	0110	0.10, 0.92, 0.88, 0.07	0.0
F7	0111	0.38, 0.70, 0.86, 0.77	70.0
F8	1000	0.99, 0.05, 0.04, 0.01	0.0
F9	1001	0.90, 0.02, 0.18, 0.75	0.0
F10	1010	0.92, 0.08, 0.94, 0.04	0.0
F11	1011	0.95, 0.08, 0.85, 0.91	0.0
F12	1100	0.90, 0.98, 0.09, 0.19	0.0

Therefore, the proposed pre-processing technique allows the ANN construction for the classification of 12 different faults with an average testing error close to 6%. If two errors are similar, it is very difficult for the ANN to differentiate them. This is the reason why the testing error with pre-processing process is not close to zero when the number of faults is larger than one.

After validating the digital pre-processing mode [36], it is very interesting to see the impact of changing the number of

digits used to code the analog data extracted from the test bench. All the results presented above have been based on 16-bit analog-to-digital (AD) converter and use an accuracy level of the four digits after the decimal point. An interesting approach is to test the accuracy required for the ANN to distinguish between all faults. To implement this, analog data will be converted to digital with multiple accuracies: (i) three digits after the decimal point has been used with a 12-bit AD converter, (ii) two digits after the decimal point, with a 9-bit AD converter, (iii) one digit after the decimal point, with 6-bit AD converter. As the AD converter influences the number of input data, Table III shows different number of input that corresponds with different AD accuracies. If a NB-bit accuracy is used with the NC currents of the WRIG (NC being in between 1 and 6), the number of ANN input data is  $NB * NC$ . For a 16-bit AD converter with six currents, it gives  $16 * 6 = 96$  ANN inputs.

With the given ANN structure, the accuracy depending on NB has been changed keeping all other ANN parameters unchanged (one hidden layer, number of hidden neurons, transfer function, number of outputs) in order to evaluate the training error, the testing error and the number of iterations to reach a given minimum error (Table III).

TABLE III  
INFLUENCE OF THE AD CONVERTER ACCURACY.

NB-bit	ANN inputs	Training error [%]	Testing error [%]	Iterations
6	36	0.12	5.3	10,000
9	54	0.15	10.0	10,000
12	72	0.25	13.9	10,000
16	96	0.25	13.9	12,000

As shown (Table III) the ANN performances are improved when the number of bits used by the AD converter in the pre-processing mode is reduced. This means that the ANN does not need the average of LSBs. The ANN can find the fault signature inside only on the average of MSBs, and any other additional data are considered by the neural network as noise. The 6-bit AD converter is the minimal configuration that can be used to encode the different current with a minimal accuracy of only one digit after the decimal point. The average of LSBs does not offer to the ANN enough information because it is always close to 0.5 whatever the signal is and that is due to the constant change of these bits.

In conclusion, the pre-processing process has given a Gaussian character to data and has enhanced the information extraction of input signals which is very important in both system identification and ANN theory.

## V. CONCLUSION

In the field of electrical machines and drives, ANNs have been regularly used as classifiers or to substitute complex control algorithms. In this paper, an original ANN with digital inputs has been used to detect and to localize turn-to-turn windings short-circuits in WRIG in order to be applied in wind farms. The ANN architecture is very simple and the proposed method does not require a large computational burden. The ANN method is based on the back-propagation algorithm

and it has been built with an efficient training mode and verified in the testing mode with many real signals coming from a dedicated test-bench. The ANN performances have been evaluated by varying some key parameters such as data acquisition, bits number and data pre-processing. At this stage, it has not yet been implemented in a real wind farm but its efficiency and its simplicity can lead to much simple industrial development with basic hardware such as FPGA boards to decrease the cost of modern condition monitoring systems adapted to wind farms.

## REFERENCES

- [1] G. A. Capolino, "A comprehensive analysis of the current status in low voltage induction motor diagnosis," in *Proc. of the ICEM'2000*, August 2000, pp. 595–602, Espoo, Finland.
- [2] A. Bellini, F. Filippetti, C. Tassoni, and G. A. Capolino, "Advances in diagnostic techniques for induction machines," in *IEEE Trans. on Industrial Electronics*, vol. 55, no. 12, Dec. 2008, pp. 4109–4126.
- [3] A. Yazidi, G. A. Capolino, C. Rossi, M. Artioli, and F. Filippetti, "Wireless system for monitoring wind turbine generators," in *Proc. of the SPEEDAM*, June 2004, pp. 815–820, Capri, Italy.
- [4] H. Henao, C. Martis, and G. A. Capolino, "Analytical approach of the frequency response for the wound rotor induction machine for diagnosis purpose," in *Proc. of IEEE-SDEMPED05*, Sept 2005, pp. 21–26, Vienna, Austria.
- [5] A. Yazidi, H. Henao, G. A. Capolino, F. Betin, and L. Capocchi, "Experimental inter-turn short-circuit fault characterization of wound rotor induction machines," in *Proc. of IEEE-ISIE10*, July 2010, pp. 2615–2820, Bari, Italy.
- [6] A. Stefani, A. Yazidi, C. Rossi, F. Filippetti, D. Casadei, and G. A. Capolino, "Doubly-fed induction machines diagnosis based on signature analysis of rotor modulating signals," in *IEEE Trans. on Industrial Applications*, vol. 44, no. 6, Nov./Dec 2008, pp. 1711–1721.
- [7] C. M. Bishop, *Neural Networks for Pattern Recognition*, 1st ed. Oxford University Press, USA, Nov. 1995.
- [8] F. Filippetti, G. Franceschini, and C. Tassoni, "Neural networks aided on-line diagnostics of induction motor rotor faults," in *Conf. Rec.1993 IEEE IAS Annual Meeting*, vol. 1, Dec. 1993, pp. 316–323.
- [9] —, "Neural networks approach to electric machine on-line diagnostics," in *Proc. of the Fifth EPE Conf*, vol. 4, 1993, pp. 213–218.
- [10] —, "Neural networks aided on-line diagnostics of induction motor rotor faults," in *IEEE Trans. on Industrial Applications*, vol. 31, no. 4, Jul./Aug. 1995, pp. 892–899.
- [11] C. Demian, G. Cirrincione, and G. A. Capolino, "A neural approach for the fault diagnostics in induction machines," in *Proc. of IEEE-IECON02*, vol. 4, Nov. 2002, pp. 3372–3376, Sevilla, Spain.
- [12] R. Tallam, T. Habetler, and R. Harley, "Self-commissioning training algorithms for neural networks with applications to electric machine fault diagnostics," in *IEEE Trans. on Power Electronics*, vol. 17, no. 6, Nov. 2002, pp. 1089–1095.
- [13] M. Morisue and H. Koinuma, "Neural network for digital applications," in *Proc. of IEEE Intern. Symp. on Circuits and Systems*, vol. 3, May 1989, pp. 2189–2192.
- [14] T. Szabo, L. Antoni, G. Horvath, and B. Feher, "A full-parallel digital implementation for pre-trained nns," in *Proc. of the IEEE-INNS-ENNS Intern. Joint Conf. on Neural Networks*, vol. 2, July 2000, pp. 49–54.
- [15] M. Badgero, "Digitizing artificial neural networks," in *Proc. of the IEEE Intern. Conf. on Neural Networks*, vol. 6, July 1994, pp. 3986–3689.
- [16] C. Karlsson, J. Arriagada, and M. Genrup, "Detection and interactive isolation of faults in steam turbines to support maintenance decisions," *Simulation Modelling Practice and Theory*, vol. 16, no. 10, pp. 1689–1703, 2008.
- [17] G. K. Singh and S. A. Al Kazzaz, "Development of an intelligent diagnostic system for induction machine health monitoring," *Systems Journal, IEEE*, vol. 2, no. 2, pp. 273–288, 2008.
- [18] A. K. Mahamad and T. Hiyama, "Improving elman network using genetic algorithm for bearing failure diagnosis of induction motor," in *Proc. SDEMPED 2009. IEEE International Symposium on*, 2009, pp. 1–6.

- [19] A. Medoued, A. Lebaroud, A. Boukadoum, T. Boukra, and G. Clerc, "Back propagation neural network for classification of induction machine faults," in *Proc. Diagnostics for Electric Machines, Power Electronics & Drives (SDEMPED), 2011 IEEE International Symposium on*, 2011, pp. 525–528.
- [20] B. M. Ebrahimi, J. Faiz, and M. J. Roshtkhari, "Static-, dynamic-, and mixed-eccentricity fault diagnoses in permanent-magnet synchronous motors," *Industrial Electronics, IEEE Transactions on*, vol. 56, no. 11, pp. 4727–4739, 2009.
- [21] H. A. Darwish, A. M. Taalab, and T. A. Kawady, "Development and implementation of an ANN-based fault diagnosis scheme for generator winding protection," *Power Delivery, IEEE Transactions on*, vol. 16, no. 2, pp. 208–214, 2001.
- [22] A. Bechouche, H. Sediki, D. Abdeslam, and S. Haddad, "Identification of induction motor at standstill using artificial neural network," in *Proc. IECON2010, 7-10 Nov, Glendale, AZ (USA)*, 2010, pp. 2908–2913.
- [23] M. Pucci and M. Cirrione, "Neural MPPT control of wind generators with induction machines without speed sensors," *Industrial Electronics, IEEE Transactions on*, vol. 58, no. 1, pp. 37–47, Jan 2011.
- [24] I. Sadinezhad and V. Agelidis, "An adaptive precise one-end power transmission line fault locating algorithm based on multilayer complex adaptive artificial neural networks," in *Proc. ICIT'2009, 10-13 Feb, Gippsland (Australia)*, 2009, pp. 1–6.
- [25] P. Fritzen, G. Cardoso, J. Zauk, A. de Moraes, U. Bezerra, and J. Beck, "Alarm processing and fault diagnosis in power systems using artificial neural networks and genetic algorithms," in *Proc. ICIT2010, 14-17 March, Vina del Mare (Chile)*, 2010, pp. 891–896.
- [26] M. Kaminski, C. Kowalski, and T. Orłowska-Kowalska, "General regression neural networks as rotor fault detectors of the induction motor," in *Proc. ICIT2010, 14-17 March, Vina del Mare (Chile)*, 2010, pp. 1239–1244.
- [27] V. Ghate and S. Dudul, "Cascade neural-network-based fault classifier for three-phase induction motor," *Industrial Electronics, IEEE Transactions on*, vol. 58, no. 5, pp. 1555–1563, May 2011.
- [28] S. Agatonovic-Kustrin and R. Beresford, "Basic concepts of artificial neural network (ANN) modeling and its application in pharmaceutical research," *Journal of Pharmaceutical and Biomedical Analysis*, vol. 22, no. 5, pp. 717–727, Jun. 2000, PMID: 10815714.
- [29] T. Kavzoglu, "Increasing the accuracy of neural network classification using refined training data," *Environmental Modelling & Software*, vol. 24, pp. 850–858, July 2009, ACM ID: 1518636.
- [30] P. J. Drew and J. R. T. Monson, "Artificial neural networks," *Surgery*, vol. 127, no. 1, pp. 3–11, Jan. 2000.
- [31] J. F. Mas and J. J. Flores, "The application of artificial neural networks to the analysis of remotely sensed data," *International Journal of Remote Sensing*, vol. 29, no. 3, pp. 617–663, 2008.
- [32] P. Melin and O. Castillo, *Hybrid Intelligent Systems for Pattern Recognition Using Soft Computing: An Evolutionary Approach for Neural Networks and Fuzzy Systems*, 1st ed. Springer, April 2005.
- [33] L. Capocchi, J. Santucci, B. Poggi, and C. Nicolai, "DEVSimPy: A collaborative python software for modeling and simulation of DEVS systems," in *Proc. on 20th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*, June 2011, pp. 170–175.
- [34] L. Capocchi. (2012) DEVSimPy software. [Online]. Available: <http://code.google.com/p/devsimpy/>
- [35] J. Bolduc and H. Vangheluwe, "PythonDEVS : A modeling and simulation package for classical hierarchical DEVS," in *Technical report, MSDL, University of McGill*, June 2001.
- [36] L. Capocchi, S. Toma, G. A. Capolino, F. Fnaiech, and A. Yazidi, "Wound-rotor induction generator short-circuit fault classification using a new neural network based on digital data," in *Proc. on 2011 IEEE International Symposium on Diagnostics for Electric Machines, Power Electronics Drives (SDEMPED)*, Sept. 2011, pp. 638–644.



Samuel Toma was born in Alexandria, Egypt in 1987. In 2010 he received the BSc in communication and network engineering from the French University in Egypt (UFE), Cairo, Egypt. In the same year he received a MSc in computer science from University of Corsica, Corte, France. Currently he is preparing a PhD. in the University of Corsica. His main research interests are based on the Discrete Event Systems (DEVS) and their applications in the electrical domain using artificial neural networks.



Laurent Capocchi was born in Bastia, Corsica, France. He received a MSc in electrical engineering from "Ecole Supérieure d'Ingénieurs de Nice Sophia Antipolis", ESINSA, Nice, France in 2001 and the PhD in computer science from University of Corsica "Pasquale Paoli", Corte, France in 2005. He is SCS (Society for Modeling and Simulation International) member since 2001 and he is also faculty of the SPE ("Sciences pour l'environnement") Laboratory at the University of Corsica. His main research concerns the modeling and simulation of complex systems like

digital and electrical equipments by using distributed or sequential discrete event approaches. His research goal is to propose a new approach for the modeling and the concurrent simulation of behavioral faults for complex systems using a discrete event approach. He is founder member of the DEVSimPy open source project (<http://code.google.com/p/devsimpy/>) and he contributes actively to its development.



Gérard-André Capolino (A'77, M'82, SM'89, F'02) was born in Marseille, France. He received the BSc in electrical engineering from Ecole Centrale de Marseille, Marseille, France in 1974, the MSc from Ecole Supérieure d'Electricité, Paris, France in 1975, the PhD from University Aix-Marseille I, Marseille, France in 1978 and the D.Sc. from Institut Polytechnique de Grenoble, Grenoble, France in 1987. He held tenure positions in University of Yaoundé I, Yaounde, Cameroon, University of Burgundy, Dijon, France and Mediterranean Institute of Technology in

Marseille, France. In 1994, he joined the University of Picardie "Jules Verne" in Amiens, France as a Full Professor and he is now Director of the European Master in Advanced Power Electrical Engineering (MAPEE) recognized by the European Commission in 2004. He has been Visiting Professor in many universities including the Polytechnic University of Barcelona (UPC), Barcelona, Spain, the Catholic University of Leuven (KUL), Leuven, Belgium, Texas A&M, College Station, USA, the University of Oviedo, Gijon, Spain, the University of Bologna, Bologna, Italy, the University of Tunis, Tunis, Tunisia, the Ecole de Technologie Supérieure, Montreal, Canada, Politecnico di Torino, Italy, the University of Texas at Dallas and the University of Corsica "Pasquale Paoli", Corte, France. His research interests have been focused on modeling and control of induction machines for at least 15 years. For the last 25 years, he has been involved in condition monitoring and fault detection of AC electrical machines for which he has developed many innovative techniques. He has published more than 450 papers in scientific journals and conference proceedings since 1975. He has been the advisor of 25 PhD students. Dr. Capolino is the 2012-2013 President of the IEEE Industrial Electronics Society. He is also the chair for the Steering Committee of the International Conference on Electrical Machines (ICEM). Dr. Capolino has been the recipient of the 2008 IEEE-IES Dr.-Ing. Eugene Mittelmann Achievement Award, the 2010 ICEM Arthur Ellison Achievement Award and the 2011 IEEE-PELS Diagnostics Achievement Award.