

Expert System for Power Quality Disturbance Classifier

Author

Reaz, Mamun Bin Ibne, Choong, Florence, Sulaiman, Mohd Shahiman, Mohd-Yasin, Faisal, Kamada, Masaru

Published

2007

Journal Title

IEEE Transactions on Power Delivery

DOI

[10.1109/TPWRD.2007.899774](https://doi.org/10.1109/TPWRD.2007.899774)

Rights statement

© 2007 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

Downloaded from

<http://hdl.handle.net/10072/46704>

Griffith Research Online

<https://research-repository.griffith.edu.au>

Expert System for Power Quality Disturbance Classifier

Mamun Bin Ibne Reaz, Florence Choong, Mohd Shahiman Sulaiman, Faisal Mohd-Yasin, *Member, IEEE*, and Masaru Kamada

Abstract—Identification and classification of voltage and current disturbances in power systems are important tasks in the monitoring and protection of power system. Most power quality disturbances are non-stationary and transitory and the detection and classification have proved to be very demanding. The concept of discrete wavelet transform for feature extraction of power disturbance signal combined with artificial neural network and fuzzy logic incorporated as a powerful tool for detecting and classifying power quality problems. This paper employs a different type of univariate randomly optimized neural network combined with discrete wavelet transform and fuzzy logic to have a better power quality disturbance classification accuracy. The disturbances of interest include sag, swell, transient, fluctuation, and interruption. The system is modeled using VHDL Hardware Description Language (VHDL), a hardware description language, followed by extensive testing and simulation to verify the functionality of the system that allows efficient hardware implementation of the same. This proposed method classifies, and achieves 98.19% classification accuracy for the application of this system on software-generated signals and utility sampled disturbance events.

Index Terms—Artificial neural network, classification, feature extraction, fuzzy logic, power quality, VHDL hardware description language (VHDL), wavelet transform.

I. INTRODUCTION

THE scarceness of power quality (PQ) experts in the electric power industry poses a problem in handling the huge amount of data gathered by the distributed PQ monitoring systems. Besides that, knowledge about PQ is dispersed and fragmented [1]. The disturbance waveforms contain serious imprecision of data and directly provides very little information on PQ, making conventional programs fail to identify any PQ problems. Existing recognition methods need much improvement in terms of their versatility, reliability and accuracy in order to process the disturbance waveforms. The measured data obtained are not self-explanatory and requires the intervention of human expertise to classify the type of PQ disturbance. Complex PQ disturbances are non-stationary and transient by nature and points of sharp variation such as singularities in transient

signals usually carry the most important information about the disturbances and it is vital to efficiently extract and interpret this information from the disturbance waveforms.

Discrete wavelet transform (DWT) have been proven to be very efficient in signal analysis [2]. Apart from filtering noise, it accurately detects sharp changes and discontinuities in PQ disturbance signals, and extracts their characteristic features for subsequent disturbance identification. Real-time identification and classification of different PQ events become one of the most important tasks for making a system decision to solve a PQ problem. Artificial neural network (ANN) can be used to solve power system protection problems, particularly those where traditional approaches have difficulty in achieving the desired speed, accuracy and selectivity. Fuzzy logic (FL) systems allow describing fuzzy rules, which fit the description of real-world processes to a greater extent. Thus, the use of automated systems employing intelligent methods such as DWT, ANN, and FL will be able to overcome the limitations mentioned above.

Over the last ten years a number of different approaches to the PQ disturbance analysis using intelligent methods were suggested. In one of the research [3], DWT is employed to detect PQ disturbances. This approach unifies time and frequency information and provides an integral signal-processing paradigm, where its embedded wavelet basis serves as a window function to monitor the signal variations efficiently. The prototype is realized on an FPGA. Another work [4], proposed a classification scheme, using an ANN trained to recognize patterns of PQ disturbances which are incorporated in a PC-based system that analyses data files from the digital fault recorders located in substations. Another work [5] utilizes a rule-based method and a wavelet packet-based hidden Markov model (HMM) for the classification of PQ disturbances. The rule-based method classifies time-characterized-feature disturbances and the wavelet packet-based HMM is utilized for the frequency-characterized-feature power disturbances. Lastly, a classifier to carry out waveform recognition in the wavelet domain using multiple ANNs is proposed by Surya *et al.* [6]. The classifier is able to provide a degree of belief for the identified waveform that gives an indication about the correctness of the decision made.

The benefits of integrating all three technologies: DWT, ANN, and FL are obvious for better accuracy. Huang *et al.* [7] proposed a classifier combining the three individual technologies: DWT, frequency sensitive competitive learning and learning vector quantization (FSL-LVQ) and FL. The FSL-LVQ can choose an appropriate criterion from a number of selections according to the specific problem. Apart from this, it allows common training for general classification tasks and subsequent special training for a particular application. The system is then integrated on an FPGA.

Manuscript received December 22, 2005; revised September 29, 2006. This work was supported by Intensified Research in Priority Areas (IRPA), a Ministry of Science and Technology Malaysia sponsored program for the advancement of R&D activities, under Project 03-99-01-0074-EA071. Paper no. TPWRD-00744-2005.

M. B. I. Reaz is with the Department of Electrical and Computer Engineering, International Islamic University Malaysia, Kuala Lumpur 53100, Malaysia (e-mail: mamun.reaz@iiu.edu.my).

F. Choong, M. S. Sulaiman, and F. Mohd-Yasin are with the Faculty of Engineering, Multimedia University, Cyberjaya 63100, Selangor, Malaysia.

M. Kamada is with the Department of Computer and Information Sciences, Ibaraki University, Hitachi Ibaraki 316-8511, Japan.

Digital Object Identifier 10.1109/TPWRD.2007.899774

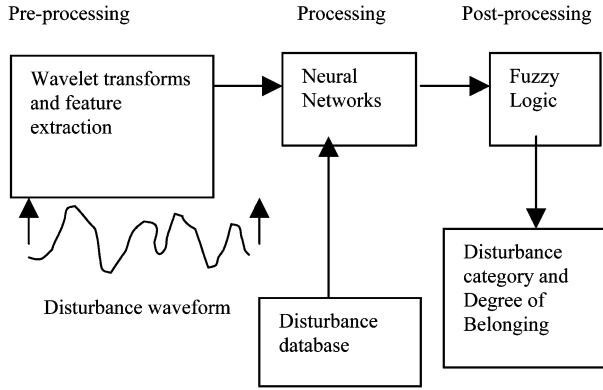


Fig. 1. Block diagram of the automatic disturbance recognition system.

In this paper, the authors present a similar classifier like Huang *et al.* [7] but with a different methodology combining the three individual technologies; DWT, univariate randomly optimized neural network (URONN) and FL resulting in a simpler classifier with a significant increase in speed, performance and accuracy. In this methodology, the wavelet analysis is used for feature extraction due to its multiresolution nature, localized properties in time and frequency domains, fast algorithm ready for on-line implementation. On the other hand, a URONN is combined with FL to function as a classifier. A URONN has redundant networking and is very robust, providing a mathematical flexibility not available to algorithm-based classifiers. The use of rule based FL reduces the difficulties of modelling and analysis of complex systems. Rule based FL is suitable for incorporating the qualitative aspects of human experience within its mapping laws. This simple yet powerful classifier is designed using a hardware description language, VHDL hardware description language (VHDL) that is fast, simple and easy to debug or make changes in. To illustrate the classification capability of the algorithm, the algorithm is tested using software-generated events and utility sampled data and achieved 98.19% classification accuracy.

This paper is organized as follows. The design methodology is described in Section II. Section III describes in detail the implementations of neural network, while Section IV is dedicated to presenting the implementation of fuzzy logic. Section V describes the results and discussion. Finally, Section VI draws the conclusion and discusses future works.

II. DESIGN METHODOLOGY

The first step in the analysis of PQ disturbances is their detection and consists of two steps: feature extraction and classification that is performed based on the model shown in Fig. 1.

In waveform generation, five types of disturbances that include transients, sag, swell, interruption, and fluctuation, are randomly superimposed on the normal waveform at a rate in accordance with their relative occurring frequencies. In this research, the power frequency chosen was 50 Hz. The disturbance waveform consists of five cycles of samples of voltage signals (in per unit). In the classification of PQ disturbances, the following distinct features inherent to different types of PQ events have been identified: fundamental component (V_n), phase angle shift (α_n) and total harmonic distortion (THD_n). A more detailed description on these features is provided in [8].

TABLE I
SUBBANDS OF DWT TRANSFORM COEFFICIENTS

| Transform coefficients | Frequency range |
|------------------------|-------------------------------------|
| CD ₁ | 64f ₁ ~128f ₁ |
| CD ₂ | 32f ₁ ~64f ₁ |
| CD ₃ | 16f ₁ ~32f ₁ |
| CD ₄ | 8f ₁ ~16f ₁ |
| CD ₅ | 4f ₁ ~8f ₁ |
| CA ₅ | 0~4f ₁ |

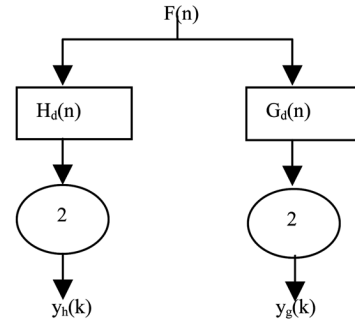


Fig. 2. Subband codification scheme of a signal.

The samples are fed in as inputs at the pre-processing stage. The pre-processing stage is used to perform feature extraction to extract the disturbance information from the PQ disturbance signal. The signals are sampled and pre-processed by a 5-level DWT for feature extraction with a sampling frequency $256f_1$ where f_1 is the power frequency, the transform coefficients contain the information about the original waveform as shown in Table I.

Wavelets can be used to implement a filter bank. The Daubechies D4 [9] wavelet is used. The Daubechies D4 transform has four wavelet and scaling function coefficients. Each step of the wavelet transform applies the scaling function and wavelet function to the input data and is limited at Nyquist frequency [9]. The localization of disturbances is fairly straight forward as it employs only filtering and decimation by factor of two. Filtering is done iteratively so that the upper half of the spectrum is analysed by the wavelet, and the lower part continues on to the next round. The lower half of the spectrum is filtered with the so-called scaling function. The wavelet acts as a high-pass filter while the scaling function acts as a low-pass filter. Consider the DWT for a signal $f(t) \in L2(R)$ as stated in (1)

$$cA_0(n) f = f(n) \quad (1)$$

where $f(n)$ is the approximation of $f(t)$ at scale-0.

The $f(n)$ signal can then be expressed with a multiresolution representation adopting a J-level DWT decomposition as stated in (2)

$$\{cA_J, \{cD_j\}_{1=j=J}\} \quad (2)$$

where $cA_J = \{cA_J(n)f\}$ and $cD_j = \{cD_j(n)f\}$, $1 \leq j \leq J$.

This procedure is performed iteratively and it moves a time domain discretized signal into its corresponding wavelet domain. This is done through a process called "subband codification" which is done through digital filter techniques [10]. This is illustrated in Fig. 2. The $f(n)$ signal is passed through a low-pass digital filter ($h_d(n)$) and a high-pass digital filter ($g_d(n)$). After that, half of the signal samples are eliminated.

Basically, the DWT evaluation has two stages. The first involves the determination of wavelet coefficients. These coefficients represent the given signal in the wavelet domain. From these coefficients, the second stage is achieved with the calculation of both the approximated and the detailed version of the original signal, in different levels of resolutions in the time domain. At the end of the first level of signal decomposition, the resulting vectors $y_h(k)$ and $y_g(k)$ are the level 1 wavelet coefficients of approximation and of detail, respectively. In fact, for the first level, these wavelet coefficients are called $cA_1(n)$ and $cD_1(n)$, respectively, as stated in (3) and (4) [11]

$$cA_i(n) = \sum_k f(n)^* h_d(-k + 2n) \quad (3)$$

$$cD_i(n) = \sum_k f(n)^* g_d(-k + 2n). \quad (4)$$

In the same manner, the calculation of the subsequent approximated ($cA_2(n)$) and the detailed ($cD_2(n)$) version associated with the level 2 is based on the level 1 wavelet coefficient of approximation ($cA_1(n)$). The process iterates, always adopting the “ $n - 1$ ” level wavelet coefficient of approximation to calculate the “ n ” level approximated and detailed wavelet coefficients. Once all the wavelet coefficients are known, the DWT in the time domain is determined. This is achieved by “rebuilding” the corresponding wavelet coefficients, along the different resolution levels. This procedure provides the approximated ($a_j(n)$) and the detailed ($d_j(n)$) version of the original signal as well as the corresponding wavelet spectrum [12].

Wavelets can be chosen with a very desirable frequency and time characteristics to perform feature extraction. Feature extraction localises and discriminates a disturbance from its background signal, in different levels of resolutions. The outputs of the feature extraction are the WT coefficients (WTCs) representing the PQ disturbance signal that indicates the disturbance’s initial and end points. WTCs at several scales reveal the time-localizing information about the variation of the signal from high to low frequency bands. In addition, examining scales of the WTCs would determine the occurrence of a disturbance as well as its occurring time. PQ indices in terms of total rms, rms of individual frequency bands, duration of disturbance, and their dependent quantities such as magnitude of disturbance and harmonic distortion are measured directly from the WTCs.

Through studying the transform coefficients of variant distorted waveforms, it has been found that each disturbance may only affect one subband coefficient.

For example, transient disturbance is “fast changing” signals that have extremely short elapsed time. Therefore, it contains only the spectral contents in the very high frequency range. On the contrary, disturbances such as voltage sag, swell, fluctuation and interruption are “slow changing” disturbances that will affect the lower subband. Fig. 3 illustrates the interruption detection process where the DWT coefficients carry different types of information of the original waveform in different resolution levels. In order to obtain these coefficients, three-band decomposition was selected and Daubechies’ wavelet with four-coefficient filter is served as a wavelet paradigm [9], [13]. For more information and illustration on the feature extraction of other types of disturbances, the reader is referred to [7].

Disturbance classification is the final step of the detection and it is implemented using a classification method that combines the learning abilities of ANN and the excellent knowledge representation of FL. The processing stage is made of an ANN block

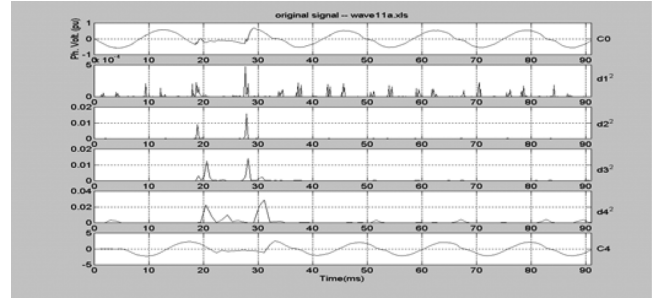


Fig. 3. Interruption detection process using DWT.

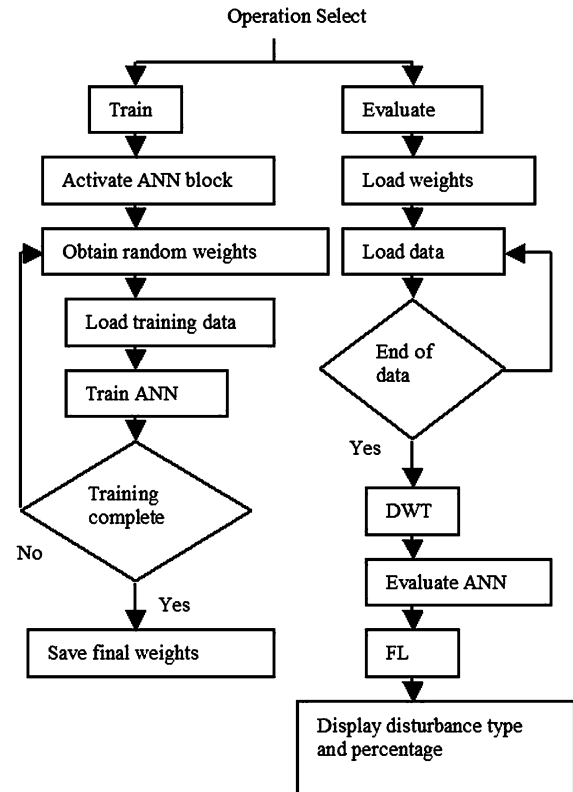


Fig. 4. Modes of operation.

trained to recognize patterns of input/output pair from the disturbance database. The patterns consist of the DWT coefficients as the input and the disturbance classes are the outputs. The ANN is trained before it is utilized in the classification process. The processing stage operates in two different modes: training and evaluation as shown in Fig. 4.

The purpose of training is to obtain the most appropriate weight values based on training data and to enable the ANN to learn from the patterns provided in the format shown in Fig. 5. The system trains the network using the provided data for a user-specified number of iterations; in this project 2000 iterations were used. The ANN is trained using a large number of training samples before they are used as part of the classifier. Thus, an input vector file containing all the training values is created. The complexity of ANN is directly related to the training time and the problem of error divergence as in (5)

$$\text{Complexity of ANN} \propto \text{training time} \propto \text{problem of error divergence.} \quad (5)$$

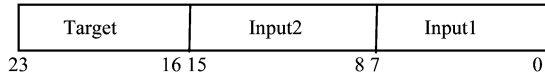


Fig. 5. Training data format.

During a training run, a random weight is generated for one weight in the ANN, and the output is evaluated. If this new weight improves the performance of the system it replaces the old weight; otherwise the original weight is kept. The training algorithm used performs this calculation for each input set six times per neuron, resulting in a final best weight for the neurons. Each input set is trained for six times and the value obtained at the sixth attempt is taken as the final weight value at the end of the training process. It is observed that there is no change in the final weight value when performing the calculation in (6) for more than six times.

When training mode is selected, DWT and FL blocks are disabled (inactive) and only the ANN block is activated. When the ANN has been trained, the final weights are sent to the output port and the training of the ANN is complete. The final weight values were assigned to each neuron when performing classification.

In Evaluation (testing) mode, the DWT block accepts PQ disturbance data as inputs, processes the inputs, and outputs the values of approximate and detail coefficients to the ANN block. The ANN block then performs pattern recognition on the coefficients and sends the evaluated output results to the FL block, which then classifies the type of disturbance together with the percentage of it belonging to a particular disturbance class.

Among different types of neural networks, the URONN [14] algorithm was chosen for the ANN implementation. It is a very advantageous approach in the implementation of an ANN because less calculation needs to be performed in each epoch as this algorithm searches for the weights that best fit the network by randomly changing them in order to minimize error. This algorithm is also robust for comparing training speeds between software and hardware.

For two (or more) hidden layers, the backpropagation and competitive learning network like LVQ loses its computational simplicity thus for such algorithms, the strategic search can be competitive in speed [14]. The URONN algorithm learns to associate successful outputs to the corresponding inputs after repeated learning attempts, through trial and error. The algorithm loops through the process of generating and testing weights, ending once a weight value that produces the least error has been obtained. The rate of change of error to rate of change of weights is used by the network as a measure in changing and adapting to a weight value. The weights are changed through learning based on the problem at hand. The URONN algorithm utilizes an adaptive network to change the weight by an amount proportional to the difference between the desired output and the actual output. The inputs and outputs of URONN can be described as given in (6) [14]

$$\begin{aligned}
 \text{Inputs : } & [N, M, J, Q; \text{exemplar training pairs} \\
 & \{(x^{(q)}, t^{(k(q))})\}; \text{sample size } I_s; \\
 & \text{number of outer iterations } I; \\
 & \text{and } b \equiv \text{bound of weight range}] \\
 \text{Outputs : } & [\text{trained weights } (w, u)] \quad (6)
 \end{aligned}$$

where N is the number of input features, M is the number of neurodes in the hidden layer, J is the number of neurodes in the output layer, and Q is the number of exemplar pair instances to be used for training.

In addition, this algorithm utilizes 8-bit signed integers as values for weights and inputs instead of 32-bit floating-point values. Arithmetic operation using floating-point operands is significantly more complex than when using signed integer operands [14]. In addition, for hardware modelling, each weight is typically implemented as n -bit signed integer, where n is typically 8 in the literature [15], and the weights are usually stored in an SRAM-Array. This algorithm offers several other advantages such as the fact that a large-scale of feedforward standard ANN can be trained independently of parametric settings for which no information is available, the reasonably good initial rate of convergence obtained when using a small sample size, the generalized learning and ease at which the ANN moves out of the well of shallow local minima to deeper local or global minima [14].

Finally, the post-processing stage groups the output data and form decisions. This stage is important because the output results from the ANN block may contain some imprecision. The neural approach suffers difficulty in estimating how well a given training set reflects an unknown underlying distribution of points and whether the neural networks encode the original structure [16]. For the available training samples, inaccuracies or even errors are unavoidable. Without a rule, when classifiers such as ANN make a mistake, there is no explanation as to why the mistake is being made. All these factors add uncertainties to the input-output relationship described by the training set. Another advantage of using FL is to cater for cases where there are multiple types of disturbances occurring at the same time. The use of FL will provide more accurate, more comprehensive and more useful information on the type of power disturbances. Thus, the output results are then fed into the FL block to further classify the output in one of the five disturbance types based on a set of predefined codeword. The rule-based approach is extensively used in artificial intelligence and expert systems. The main aim is to enhance the transparency of the expert systems by tuning rules and membership functions [17]. A number of standard rules are used to mimic experienced PQ engineers thought processes. The rule-based post processing attempts to correct wrong decisions made by classifiers and thus improve the accuracy of the classification results. They have inherent abilities to deal with imprecise or noisy data, making them suitable for fault diagnosis [18]. FL uses a rule-based or heuristic reasoning process to analyze PQ data, just as a PQ expert would. In addition, FL allows a system to be more understandable to a non-expert operator. In this way, FL is used as a general methodology to incorporate knowledge of PQ to assist in decision-making where the PQ knowledge is presented as a series of rules.

The system presented in this work is designed to recognize five types of PQ disturbances. Taking readings of the voltage values (in per unit) of each cycle in the time domain identifies the disturbance. The values are then compared against the preset characteristic of a particular PQ disturbance as defined by the IEEE 1159.2 working group. The voltage values (in per unit) are then fed into the DWT block to obtain the coefficient values that carries the disturbance information of the original PQ disturbance signal. The ANN is trained with these coefficients cov-

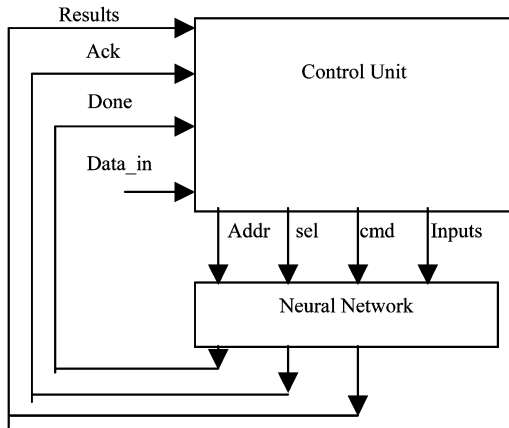


Fig. 6. ANN design.

ering samples of disturbance data from each class. Fuzzy rules were established to perform the classification task. Using the same principles, the classification of other types of PQ disturbances such as capacitor switching, notching, impulse, etc., can be easily achieved.

III. IMPLEMENTATION OF ARTIFICIAL NEURAL NETWORK

In realistic applications, the design of neural network system is a complex, usually iterative, and interactive task. Although it is impossible to provide an all-inclusive algorithmic procedure, the following inter-related skeletal steps reflect typical efforts and concerns.

- Step 1: Study the classes of measurements/patterns under consideration to develop possible (quantitative) characterizations. This includes assessment of (quantifiable) structures, probabilistic characterization and exploration of possible classes of similarity/dissimilarity measures. In addition, possible deformations or invariant properties and characterization of “noise” sources should be considered at this point.
 - Step 2: Determine the availability of measurements (input) or feature (pre-processed) data.
 - Step 3: Consider constraints on the desired system performance and computational sources.
 - Step 4: Consider the availability and quantity of training and test data.
 - Step 5: Consider the availability of suitable and known ANN system structures.
 - Step 6: Develop an ANN simulation.
 - Step 7: Train the ANN system.
 - Step 8: Simulate the ANN system performance-using test set.
 - Step 9: Iterate until the desired performance is achieved.
- The implementation of neural network is made of ANN and Control Unit block as depicted in Fig. 6.

A. ANN

This module is one of the more important modules as it performs the classification task. It consists of neurons and their connections into a network. An ANN consists of simple processing elements known as neurons. The input of a neuron is usually an input column-vector $x = [x(1), \dots, x(p)]^T$ of a

pre-processed signal. The n th element of the input vector, $x(n)$, is connected to a neuron k by a weight factor $m(k, n)$. The weight factor then forms a weight vector for the neuron k , $m_k = [m(k, 1), \dots, m(k, p)]$. The output of the neuron is simply a linear combination of the input vector x with the weight vector m_k , as shown in (7)

$$u_k = m_k x^T = \sum m(k, i)x(i) \text{ for } i \text{ from } 1 \text{ to } P. \quad (7)$$

The Neuron architecture is modeled using the Carry-Save-Adder prototype as its multiplier component for its high speed and regular structure. The datapath is made up of three layers, two for the hidden layer and one for the output layer which is connected in the traditional feed-forward architecture. For this implementation, input neurons were not considered, as their purpose is to merely distribute the inputs among the neurons at the hidden layer. Thus, in the design, the inputs come directly from the data bus into the neurons at the hidden layer.

B. Control Unit

The control unit is used to control the ANN. It consists of five sub modules modeled in a top-down design approach as described in the following sections.

1) *Trainer*: This module basically utilizes state machines, which governs control signals for the ANN. The trainer consists of a state machine that traverses the states and sends and receives control signals to the ANN. When the trainer has completed the specified number of training run, it enters the idle state awaiting the next instruction.

2) *Bus Master*: This component is designed to get and set the weights, and to activate the ANN during operational mode. The data bus controller controls the transfer of data through the ANN, and supplies the network with new random weights as well as the inputs. The datapath entity is instantiated for each neuron. Subsequently, each neuron has a unique identifier corresponding to the number in which it was instantiated. A comparator uses this identifier to ensure that the value being passed in on the address bus lines matches the identifier of the neuron. If a match is detected, the particular neuron is then used. Along with the address bus and the data bus, there exists a command bus and select bus. The command bus provides the neuron with one of four commands utilizing 2 bits, which determine the type of operation to be performed.

- i) “00” is to read the weights of the neurons.
- ii) “01” is to write weights of the neurons, i.e., take from datapath and store.
- iii) “10” enables the idle state and.
- iv) “11” is to begin forward calculation.

The select bus selects the weight for the neuron. The combinational logic determines which weight is being used and what operation is to be performed. The bi-directional 8-bit data bus exists to provide the weight storage units with their values during storage and to take the values from the storage units when reading. The address bus is used to inform the ANN as to which neuron should be performing a read or write operation.

3) *Error Calculator*: Error calculator keeps track of the number of times that the network misclassifies each input data for the network, which instructs the control unit to either keep or discard a new weight. This module compares the output evaluated by the ANN from the inputs provided with the current weights, and compares it with the expected target output. If the

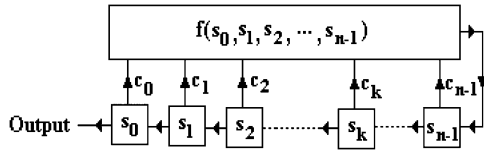


Fig. 7. Linear feedback shift register.

evaluated output is closer to the target value, the tested current random weight replaces the previous weight value.

Let t_k be the k th target or desired output and Z_k be the k th computed output with $k = 1, \dots, K$ and w represents all the weight in the network. The training error is calculated using (8)

$$\text{Error} = \frac{\sum_{n=1}^N \sum_{k=1}^K (Z_{kn} - t_{kn})^2}{NK} \quad (8)$$

where N = number of pattern and K = number of output.

4) *Pseudo Random Number Generator*: The random number generator generates random weights. It employs the linear feedback shift register (LFSR) that consists of the input sequence (initialisation vector), the feedback (tap sequence), and the output. An LFSR is a mechanism for generating a sequence of binary bits. The register consists of a series of cells. A clock regulates the behaviour of the register and at each clock instant, the contents of the cells of the register are shifted right by one position, and the exclusive-or of a subset of the cell contents is placed in the leftmost cell. A sample for LFSR of a generic length, n is shown in Fig. 7.

The feedback gives a linear relationship between the input and the output. Let the input sequence of length n be $(s_0, s_1, \dots, s_{n-1})$. The feedback is thus a linear function $f(s_0, s_1, \dots, s_{n-1})$ defined by (9)

$$f(s) = \sum_{i=0}^{n-1} c_i s_i \quad (9)$$

where c_0, c_1, \dots, c_n are constant coefficients. The output of the LFSR is determined by the initial values s_0, s_1, \dots, s_{n-1} and the linear recursion relationship as shown in (10) and (11)

$$s_{k+1} = \left(\sum_{i=0}^{n-1} c_i s_{i+k} \right) \quad k \geq 0 \quad (10)$$

or equivalently

$$\sum_{i=0}^n c_i s_{i+k} \quad k \geq 0 \quad (11)$$

where $c_n = 1$ by definition.

5) *RAM Interface*: This module handles the storage of the training data into RAM. It also recalls the training data set for the control unit for subsequent training epochs. The values are stored as 32-bit binary.

IV. IMPLEMENTATION OF FUZZY LOGIC

Both ANN and FL systems realize complex non-linearity by combining and interpolating membership functions, sigmoid

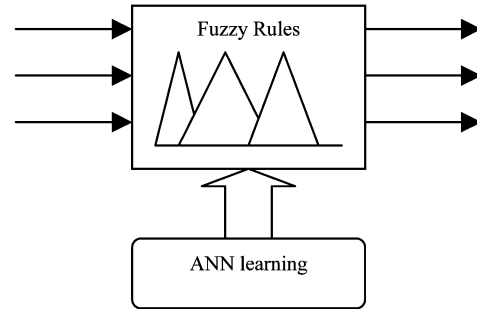


Fig. 8. FL and ANN interaction.

functions, and other base functions. Since ANN and FL systems have different advantages in their learning function and in the explicit knowledge expression of fuzzy rules, respectively, they can complement each other. Fig. 8 shows both technologies connected in cascade. In this configuration, the ANN configures fuzzy reasoning rules where a feedforward ANN is employed to handle fuzzy values and adjusts the parameters of the membership functions [19], [20]. This is performed through learning and by adjusting weights among the neurons for ANN. A fuzzy classifier consists of three parts, the fuzzification of the inputs, the defuzzification of the outputs and the rule-base.

Other evolutionary methods such as Genetic Algorithm (GA) have demonstrated to be a powerful tool to perform tasks such as generation of fuzzy rule base, optimization of fuzzy rule bases, generation of membership functions, and tuning of membership functions [21]. All these tasks can be considered as optimization or search processes. Fuzzy system generated or adapted by GA is called Genetic Fuzzy Systems [22]. GA can be used for local tuning of fuzzy membership functions. For this purpose, the local adjustment is employed on the initial membership functions. GA is also used to investigate discrete points that will be modified on the membership functions [23], [24].

A. Fuzzifier

The function of fuzzifier is to transform crisp inputs into fuzzy inputs. Crisp inputs are 8-bit hexadecimal values representing input values from the real world. These values are referred to as ‘‘crisp’’ values since they are represented as a single number, not a fuzzy one. In order for the fuzzy system to understand the inputs, the crisp input has to be converted to a fuzzy number. This process is called fuzzification. To get fuzzy inputs, the crisp input is compared with membership function parameters. The parameters are the value of centre, width, slope and type of membership function as shown in Fig. 9. In this design, three types of membership function, i.e., left inclusive, right inclusive and symmetrical inclusive membership function have been used.

B. Rule Evaluation

Rule evaluation is the second step of the FL process, and determine what control action should occur in response to a given set of input values. The rule evaluation method used in this design is called ‘min-max’ inferences [15], since it takes the minimum of the antecedents to determine rule strength and the maximum of the rule strengths for each consequent to determine fuzzy outputs. Six rules were used as shown below.

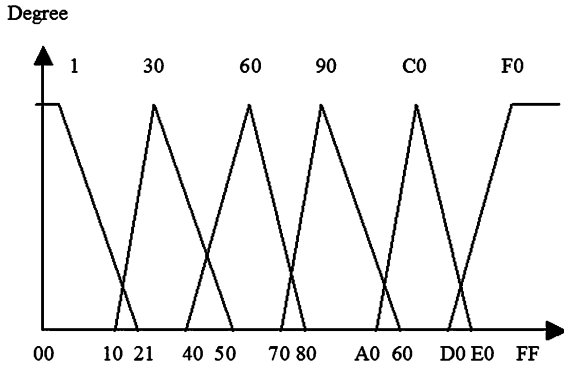


Fig. 9. Fuzzy membership functions.

- Rule 1: IF Input1 is High THEN Class is Transient or a high level of Input1 often indicates transient.
- Rule 2: IF Input1 is Low and Input2 is A_1 THEN Class is Sag.
- Rule 3: IF Input1 is Low and Input2 is A_2 THEN Class is Swell.
- Rule 4: IF Input1 is Low and Input 2 is A_3 THEN Class is Interruption.
- Rule 5: IF Input1 is Low and Input2 is A_4 THEN Class is Fluctuation.
- Rule 6: IF Input1 is A_5 THEN Class is Normal.

In the above rules, A_1, A_2, A_3, A_4, A_5 , High and Low are the membership functions for the input patterns. Taking Rule 1 as an example, the two uncertainties to be modeled are “often” and “high”, which are most easily represented as a fuzzy measure and fuzzy set, respectively.

Consider two fuzzy sets represented by $U = [\mu_s^1, \mu_s^2, \dots, \mu_s^L]$ and $V = [\mu_c^1, \mu_c^2, \dots, \mu_c^L]^T$. The association can be implemented by building up a rule matrix to recall V from U using (12)

$$V = XoU \tag{12}$$

with X as an $L \times L$ matrix. The max-min composition relation is denoted by the composition operator “o” where each entry of $V\mu_c^i$ is evaluated by taking the fuzzy inner product of a row of X with the input vector U using (13)

$$\mu_c^i = \max_j \min \{x_{ij}, \mu_s^j\} \tag{13}$$

with x_{ij} denoting the (i, j) th element of X . It is seen that the fuzzy vector-matrix multiplication resembles classical vector-matrix operation except for replacing pair-wise multiplication with pair-wise minima and replacing row (column) sums with row (column) maxima. The rule-matrix X is obtained by evaluating the fuzzy outer-product using (14)

$$X = V * U^T \tag{14}$$

with $x_{ij} = \min \{\mu_c^i, \mu_s^j\}$. It appears that each entry of $V = [\mu_c^1, \mu_c^2, \dots, \mu_c^L]^T$ can be utilized for determining how well the input matches the corresponding disturbance type since the μ_c^i is basically proportional to the similarity measure μ_s^i . Consequently, the V is used as the belief degree as well as the class indicator in the classification.

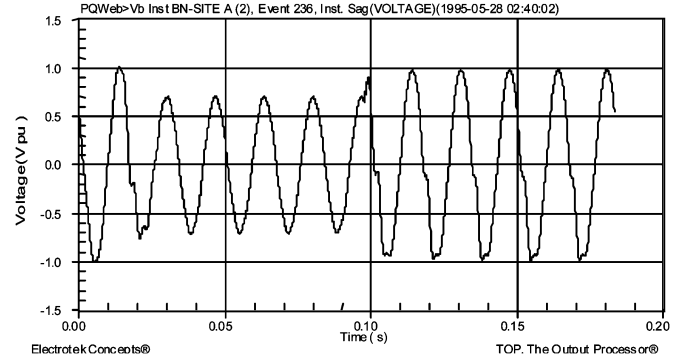


Fig. 10. Sag.

C. Defuzzifier

Defuzzification is the last step in the FL process, which transforms the fuzzy output to crisp output based on output membership function. In defuzzification, all significant fuzzy output is combined into a specific, comprehensive result to get the crisp output.

Center of Gravity (COG) or centroid method [25] is used where output membership function is determined using singletons. A singleton membership function is defined by a single 8-bit value in the knowledge base. Defuzzification uses (15) to determine the crisp output.

$$\text{Output} = \frac{(\sum \text{fuzzy output} * \text{singleton})}{(\sum \text{fuzzy output})} \tag{15}$$

V. RESULTS AND DISCUSSION

A. Data Collection

Simulation of PQ disturbances was performed by third-party tools (Alternative Transient Program, Matlab and COMTRADE). Samples of software-generated signals are shown in Fig. 10. A total of 10 200 examples (1700 examples per class) were used to test the classification method. The starting time, duration and distortion magnitude were generated randomly. This makes the testing results more reliable because none of these are fixed for real power system disturbance events. Collaboration was made with related parties from Tenaga Research Centre’s PQ Lab. Field data was collected from six different substations located throughout Malaysia including Bayan Lepas, Bukit Raja, Serdang, Skudai, Telekom Bangsar, and TNB Headquarters. All events that occurred during a specific date and time interval were given by the event tolerance summary that was collected from the power utility. A sample reading of an event tolerance summary of a substation is shown in Table II. The reading was taken from Phase A voltage signal where Number refers to a bus number, Event Type refers to the point at which the event was captured, amplitude refers to the lowest or highest voltage reading, duration refers to the length corresponding to the particular amplitude and date/time corresponds to the date/time when the event was captured. As the possibility of more than one event occurring at any time exists, the event tolerance summary was used to extract the event type showing the readings for the individual phase of a particular disturbance type. A sample is illustrated in Fig. 11. This is followed by the feature extraction process by the DWT.

TABLE II
PHASE A EVENT TOLERANCE SUMMARY FEBRUARY 9, 2002 6 EVENTS PMU
SKUDAI 22 KV (TX_No_6 02/02/02 12:35) (CBEMA)

| Number | Type | Amplitude | Duration | Date and Time |
|--------|------|-----------|----------|-------------------------|
| 1 | 25 | 13477.4 | 0.95 | Feb 02 2002 12:35:02 |
| 10 | 30 | 12989.07 | 0.04 | Feb 03 2002 20:44:38 |
| 50 | 30 | 12842.581 | 0.06 | Feb 04 2002 12:54:58 |
| 160 | 26 | 6763.108 | 0.04 | Feb 04 2002 15:45:30 |
| 40 | 25 | 13111.151 | 5.36 | Feb 04 2002 20:44:38 |
| 380 | 26 | 8838.43 | 0.12 | Feb 05 2002 01:43:56 |

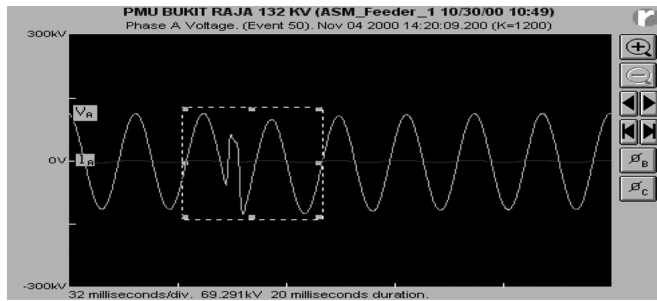


Fig. 11. Fluctuation event on Phase A.

In this work, samples of PQ disturbance events occurring on all the three phases of a disturbance signal were collected individually. However, in the feature extraction and classification process, each phase of a voltage signal is utilized independent of each other. At the moment, the proposed system classifies one type of disturbance at any one time and further research and enhancement will be made to cater for composite events.

B. Simulation and Results

The system is modeled using VHDL, where max + PlusII and Quartus II softwares were used for simulation. Testbenches were used to perform extensive tests on each individual component. Performance on the test sets of the entire system combining WT, ANN, and FL reached an average of 98.19% accuracy. The combination of WT and ANN only produced a classification accuracy of 96.17%.

The results of classification using the test and training set for all five disturbances are shown in Table III. The table is organized as follows: in each row the performance of the network is represented by a specific PQ disturbance type, while each column represents the results of the classification. The percentage of correct classification is calculated using (16)

$$\% \text{correct classification rate} = \frac{\text{Total correct classified disturbance events}}{\text{total disturbance events}} \quad (16)$$

C. Comparison

In order to examine and analyze the effectiveness and feasibility of the proposed approach, a comparison in terms of percentage of accuracy is made and presented in Table IV.

TABLE III
PERFORMANCE OF THE SYSTEM ON THE TEST AND TRAINING DATA (C-CLASS, 1-TRANSIENT, 2-VOLTAGE SAG, 3-VOLTAGE SWELL, 4-INTERRUPTION, 5-FLUCTUATION, 6-NORMAL, CORRECT-PERCENTAGE OF CORRECT IDENTIFICATION, MI—PERCENTAGE OF MISTAKE IDENTIFICATION TO CLASS 1)

| C | Correct | M1 | M2 | M3 | M4 | M5 | M6 |
|---|---------|------|------|------|-----|------|------|
| 1 | 97.24 | - | 0.6 | 0.8 | - | 1.1 | 0.26 |
| 2 | 97.14 | 1.15 | - | - | 0.1 | 1.26 | 0.35 |
| 3 | 98.87 | 0.4 | - | - | - | 0.73 | - |
| 4 | 98.20 | 0.03 | - | - | - | 1.54 | 0.23 |
| 5 | 97.73 | 1.27 | 0.2 | 0.8 | - | - | - |
| 6 | 99.98 | - | 0.01 | 0.01 | - | - | - |

TABLE IV
PERFORMANCE COMPARISONS

| Method | Accuracy |
|--|----------|
| Wavelet Transform and Neural Fuzzy (Huang et al, 2002) | 96.5% |
| Wavelet-based Neural Classifier (Surya et al. 2000) | 94.37% |
| Proposed | 98.19% |
| Neural Network (Mladen and Rikalo, 1996) | 95.93% |
| Wavelet and Fuzzy Logic (Jaehak et al, 2002) | 98.7% |

Method by Jaehak *et al.* [5] employing wavelet and fuzzy to classify PQ disturbances produced the highest result at 98.7%, followed by the proposed method in this paper employing DWT, URONN, and FL at 98.19%. Method by Huang [7] employing DWT, FSCL-LVQ, and FL and Mladen and Rikalo [4] using neural net achieved a classification accuracy of 96.5% and 95.93% respectively. Lastly, PQ disturbance waveform recognition combining wavelet and ANN method by Surya *et al.* [6] obtained an accuracy of 94.37% which is also lower than the accuracy of the combination of WT and ANN (URONN) that produced a classification accuracy of 96.17%. Based on the comparison made in Table IV with existing ANN methodologies, it is proven that this new approach provided a higher recognition rate and also confirm that a combination of more than one method yields a better result.

The method by Jaehak *et al.* [5] yielded the highest result. However, the response time of a fuzzy based classifier for classifying disturbance signals slows down, as the number of rules grows larger. Rules based expert systems are highly dependent on “if...then” clauses. If a large number of event types or features were analyzed, the expert system would become more complicated and the risk of losing selectivity would increase (ambiguity). It also takes time to refine the rules and membership values. This limitation is found in similar works in FL by Dash *et al.* [26] and Mladen *et al.* [27], where the classification time taken are 0.02 seconds and 15 milliseconds respectively. This will pose a problem for future enhancement to the system where the classification time will continue to increase as more rules are added in order to classify various other types of PQ disturbances. In comparison, this work yields a shorter classification time of 16 microseconds using the same data set. The classification time was not provided in the paper by Jaehak *et al.* [5] and thus comparison with the paper’s method could not be done.

TABLE V
COMPARISON ON THE SPEED AND ACCURACY OF SOFTWARE ALGORITHMS FOR
EXPERIMENTATION USING 5000 ITERATIONS

| | VHDL | Matlab | C |
|----------------------------|----------|--------|-------|
| Correct Classification (%) | 98.19 | 71.4 | 69.15 |
| Time (s) | 0.000016 | 135 | 5 |

In this work, instead of using conventional programming languages such as C, Matlab or Java, the new method models the design using Very High Speed Integrated Circuits Hardware Description Language or VHDL. This is a hardware independent language, which supports a design to be modified easily to suit research requirement. The use of VHDL helps in speeding the execution process as shown in Table V. In addition, designs can be developed and tested efficiently and in a shorter design cycle time.

The performance of the classifier of the proposed method is encouraging. The results reveal that the proposed approach is computationally simple, accurate and exhibits a good balance of flexibility, speed, size and design cycle time. These important characteristics are needed to design a good classification system. Comparison and results presented validate the successful classification of PQ disturbance.

Based on the comparisons made above, the feasibility and the useful features of the new approach are further summarized and justified as follows.

- 1) For feature extraction, the proposed method uses DWT that enables signal representation with a few terms and can effectively remove redundancy of time domain data and therefore reduce the size of a classifier.
- 2) For the classification method, the proposed method combines the learning abilities of ANN and the excellent knowledge representation of FL. This produces a more intelligent system, with recognition performance better than those obtained by the individual technologies
- 3) ANN is used for its pattern recognition capabilities. However, its ability to perform well is greatly influenced by the weight adaptation algorithm. Thus a unique algorithm, *uron* was chosen for the implementation of a ANN. This algorithm is a very advantageous approach in the implementation of a ANN as explained earlier as it simplifies calculation, minimizes error, reduces training time, and provide a significant increase in the accuracy of the classification [14].
- 4) As there exists uncertainty in the training set and in the subsequent pattern recognition, FL is used to determine the final output rather than taking the output of the ANN as the final classification, improving robustness in the system and producing a more accurate result.
- 5) The VHDL model provides a systematic approach and helps in speeding the execution process.
- 6) The system has the potential of being extended to classify other kinds of PQ disturbances.
- 7) This novel combination of methods shows promise for further development of a fully automated PQ monitoring system.

VI. CONCLUSIONS AND FUTURE RESEARCH

The advantages of a combination of WT, ANN, and FL are obvious. It therefore appears that a judicious integration of the

merits of these three technologies can provide a more intelligent system in terms of generic advantages like parallelism, fault tolerance, adaptivity, and uncertainty management by evolving synergism between them, to handle real life ambiguous situations and recognition problems.

A new approach of PQ disturbance classifier utilizing WT, ANN, and FL is proposed. The initial research involved determining the functions of the system and designing the individual modules in VHDL to perform these functions. The functionality of the modules was successfully verified. Future work will include the detection and classification of composite/polyphase events.

ACKNOWLEDGMENT

The authors wish to thank Tenaga Nasional Research and Development Centre, Malaysia, for providing valuable assistance in obtaining power disturbance field data from the various substations throughout the country.

REFERENCES

- [1] M. Kezunovic, "A survey of neural net applications to protective relaying and fault analysis," *Eng. Intell. Syst.*, vol. 5, no. 4, pp. 185–192, 1997.
- [2] A. M. Gaouda, M. M. A. Salama, M. R. Sultan, and A. Y. Chikhani, "Power quality detection and classification using wavelet-multiresolution signal decomposition," *IEEE Trans. Power Del.*, vol. 14, no. 4, pp. 1469–1476, Oct. 1999.
- [3] H. Shyh-Jier, Y. Tsai-Ming, and H. Jiann-Tseng, "FPGA realization of wavelet transform for detection of electric power system disturbances," *IEEE Trans. Power Del.*, vol. 17, no. 2, pp. 388–394, Apr. 2002.
- [4] M. Kezunovic and I. Rikalo, "Detect and classify faults using neural nets," *IEEE Comput. Appl. Power*, vol. 9, no. 4, pp. 42–47, Oct. 1996.
- [5] J. Chung, E. J. Powers, W. M. Grady, and S. C. Bhatt, "Power disturbance classifier using a rule-based method and wavelet packet-based hidden Markov model," *IEEE Trans. Power Del.*, vol. 17, no. 1, pp. 233–241, Jan. 2002.
- [6] S. Santoso, E. J. Powers, W. M. Grady, and A. C. Parsons, "Power quality disturbance waveform recognition using wavelet-based neural classifier—Part 2: Application," *IEEE Trans. Power Del.*, vol. 15, no. 1, pp. 229–235, Jan. 2000.
- [7] H. Jiansheng, M. Negnevitsky, and D. T. Nguyen, "A neural-fuzzy classifier for recognition of power quality disturbances," *IEEE Trans. Power Del.*, vol. 17, no. 2, pp. 609–616, Apr. 2002.
- [8] M. Kezunovic, "Advanced assessment of the power quality events," in *Proc. 9th Int. Conf. Harmonics and Quality of Power*, Orlando, FL, Oct. 2000.
- [9] I. Daubechies, *Ten Lectures on Wavelets*. Philadelphia, PA: SIAM, 1992, vol. 61.
- [10] Y. Meyer, *Wavelets: Analysis and Applications*. Philadelphia, PA: SIAM, 1993.
- [11] D. C. Robertson, O. I. Camps, and J. S. Mayer, "Wavelets and electromagnetic power system transients," *IEEE Trans. Power Del.*, vol. 11, no. 2, pp. 924–930, Apr. 1996.
- [12] C. Penna, "Detection and classification of power quality disturbances using the wavelet transform," M.Sc. thesis, Faculty Elect. Eng., Univ. Federal Uberlandia, Minas Gerais, Brazil, 2000.
- [13] I. Daubechies, "Wavelet transform, time-frequency localization and signal analysis," *IEEE Trans. Inf. Theory*, vol. 36, no. 5, pp. 961–1005, Sep. 1990.
- [14] Looney and Carl, *Pattern Recognition Using Artificial Neural Networks: Theory and Algorithms for Engineers and Scientists*. New York: Oxford Univ. Press, 1997, ch. 3, 4, and 6.
- [15] S. Collis, "The requirements for a VHDL-based custom IC, ASIC or FPGA design process," in *Inst. Elect. Eng. Colloq. VHDL—Applications and CAD Advances*, 1993, pp. 7/1–7/7.
- [16] B. Kosko, *Neural Networks and Fuzzy Systems: A Dynamical System Approach to Machine Intelligence*. Englewood Cliffs, NJ: Prentice-Hall, 1992, pp. 299–337.
- [17] A. L. Dexter, "Fuzzy model based fault diagnosis," *Proc. Inst. Elect. Eng.*, vol. 146, no. 6, pp. 545–550, 1995.
- [18] R. Babuska, *Fuzzy Modeling in Control*. Norwell, MA: Kluwer.

- [19] A. Kawamura, N. Watanabe, H. Okada, and K. Asakawa, "A prototype of neuro-fuzzy cooperation system," in *Proc. IEEE Int. Conf. Fuzzy Systems*, San Diego, CA, 1992, pp. 1275–1282.
- [20] H. Takagi, T. Kouda, and Y. Kojima, "Neural-networks designed on approximate reasoning architecture," *J. SOFT*, vol. 3, no. 1, pp. 133–141, 1991.
- [21] C. Ahmet, S. S. Vaidy, A. G. Dick van, M. A. S. Peter, and J. D. Jack, "A method for local tuning of fuzzy membership functions," in *Proc. Int. Conf. Computational Science*, 2005, vol. 3516, pp. 1143–1147.
- [22] O. Cordón, F. Herrera, F. Hoffmann, and L. Magdalena, "Genetic fuzzy systems. Evolutionary tuning and learning of fuzzy knowledge bases," *Adv. Fuzzy Syst.*, 2001.
- [23] O. Cordón, F. Herrera, F. Gomide, F. Hoffmann, and L. Magdalena, "Ten years of genetic-fuzzy systems: A current framework and new trends," in *Proc. Joint 9th IFSA World Congr. 20th NAFIPS Int. Conf.*, Vancouver, BC, Canada, 2001, pp. 1241–1246.
- [24] A. Arslan and M. Kaya, "Determination of fuzzy logic membership functions using genetic algorithms," *Fuzzy Sets Syst.*, vol. 118, no. 2, pp. 297–306, 2001.
- [25] K. Asai, *Fuzzy Systems for Information Processing*. Tokyo, Japan: Ohmsha, 1995.
- [26] Dash, P. K. S. Mishra, M. M. A. Salama, and A. C. Liew, "Classification of power system disturbances using a fuzzy expert system and a fourier linear combiner," *IEEE Trans. Power Del.*, vol. 15, no. 2, pp. 472–477, Apr. 2000.
- [27] M. Kezunovic, I. Rikalo, C. W. Fromen, and D. R. Sevcik, "New automated fault analysis approaches using intelligent system technologies," in *Proc. Int. Conf. Power System Technology*, Beijing, China, Oct. 1978, pp. 6–6.



Mamun Bin Ibne Reaz was born in Bangladesh in December 1963. He received the B.Sc. and M.Sc. degrees in applied physics and electronics from the University of Rajshahi, Bangladesh, in 1985 and 1986, respectively.

He has had research experience in Norway, Ireland, and Malaysia. Currently he is a Lecturer with the Department of Electrical and Computer Engineering, International Islamic University Malaysia, Malaysia, involving in teaching, research and industrial consultation. He has published extensively

in the area of IC Design and Biomedical application IC. He is author and co-author of more than 80 papers in design automation and IC design for biomedical applications.



Florence Choong was born in Kuala Lumpur, Malaysia, in 1980. She received the B.Eng. (Hons.) degree in electronics engineering in computers and the Meng.Sc. degree in electronics engineering from Multimedia University, Cyberjaya, Malaysia, in 2002 and 2005, respectively.

She is presently Lecturer at the Faculty of Engineering, Multimedia University Malaysia. She is the main author of two international conference papers and three journal papers. Her main research interests are in the area of intelligent system

applications and VLSI design.



Mohd Shahiman Sulaiman received the B.A.Sc. degree in electrical engineering and the M.A.Sc. degree in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada.

He has worked in the area of low-power high-speed mixed-signal IC design since 1998. He is currently a Lecturer with the Faculty of Engineering, Multimedia University, Malaysia. He has been a research associate for Intel Corporation (Malaysia) and Matsushita Electric Industrial Co., Ltd., Japan. He has authored/co-authored more than 30 international

conference/journal papers on integrated circuit design and design automation.



Faisal Mohd-Yasin (M'96) received the B.Sc. degree in electrical engineering, the M.Sc. degree in telecommunications and computers, and the M.Sc. degree in computer engineering from The George Washington University, Washington D.C.

Currently, he is a Senior Lecturer with the Faculty of Engineering with the Multimedia University, Cyberjaya, Malaysia. He is also an Associate Research Fellow with the Institute of Microengineering and Nanoelectronics of the National University of Malaysia, and an Associate with the Malaysian

Industry-Government Group for High Technology. His current research interests are in the area of radio-frequency identification (RFID) circuit design and noise in microelectromechanical systems (MEMS). He has written or coauthored many international journal and conference papers.

Mr. Mohd-Yasin is a member of Tau Beta Pi, Golden Key National Honor Society and National Society of Collegiate Scholars.



Masaru Kamada was born in Ibaraki, Japan, in 1962. He received the B.S., M.S., and Ph.D. degrees in engineering in 1984, 1986, and 1988, respectively, from the University of Tsukuba, Tsukuba, Japan.

He was with the University of Tsukuba from 1988 to 1992. He was an Associate Professor from 1992 to 2005 and is currently a Professor with the Department of Computer and Information Sciences, Ibaraki University, Hitachi, Japan. He was an Academic Guest at the Swiss Federal Institute of Technology, Zurich, Switzerland, from 1993 to 1995. He served the engineering science society of IEICE as an Associate Editor of its Japanese transactions (1997–2000), Publications Secretary (1998–1999), and Webmaster (1999).

He was Treasurer (2000–2001) with the Japan chapter of the IEEE CAS society and a student paper contest co-chair of ISCAS 2005. He has been serving *Sampling Theory in Signal and Image Processing* as Secretary since 2003. His current research interests include signal/image processing algorithms and hardware implementation.

Dr. Kamada is a member of IEICE and EURASIP.