FUZZY NEURAL NETWORK PATTERN RECOGNITION ALGORITHM FOR CLASSIFICATION OF THE EVENTS IN POWER SYSTEM NETWORKS

A Dissertation

by

SLAVKO VASILIC

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

May 2004

Major Subject: Electrical Engineering

FUZZY NEURAL NETWORK PATTERN RECOGNITION ALGORITHM FOR CLASSIFICATION OF THE EVENTS IN POWER SYSTEM NETWORKS

A Dissertation

by

SLAVKO VASILIC

Submitted to Texas A&M University
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

Approved as to style and content by:

Mladen Kezunovic
(Chair of Committee)

Ali Abur
(Member)

Shankar Bhattacharyya
(Member)

William Lively
(Member)

Chanan Singh
(Head of Department)

May 2004

Major Subject: Electrical Engineering

ABSTRACT

Fuzzy Neural Network Pattern Recognition Algorithm for
Classification of the Events in Power System Networks. (May 2004)

Slavko Vasilic, B.S., University of Belgrade, Serbia;
M.S., University of Belgrade, Serbia;
Chair of Advisory Committee: Dr. Mladen Kezunovic

This dissertation introduces advanced artificial intelligence based algorithm for detecting and classifying faults on the power system transmission line. The proposed algorithm is aimed at substituting classical relays susceptible to possible performance deterioration during variable power system operating and fault conditions. The new concept relies on a principle of pattern recognition and detects the existence of the fault, identifies fault type, and estimates the transmission line faulted section.

The approach utilizes self-organized, Adaptive Resonance Theory (ART) neural network, combined with fuzzy decision rule for interpretation of neural network outputs. Neural network learns the mapping between inputs and desired outputs through processing a set of example cases. Training of the neural network is based on the combined use of unsupervised and supervised learning methods. During training, a set of input events is transformed into a set of prototypes of typical input events. During application, real events are classified based on the interpretation of their matching to the prototypes through fuzzy decision rule.

This study introduces several enhancements to the original version of the ART algorithm: suitable preprocessing of neural network inputs, improvement in the concept of supervised learning, fuzzyfication of neural network outputs, and utilization of on-line learning.

A selected model of an actual power network is used to simulate extensive sets of scenarios covering a variety of power system operating conditions as well as fault and disturbance events.

Simulation results show improved recognition capabilities compared to a previous version of ART neural network algorithm, Multilayer Perceptron (MLP) neural network algorithm, and impedance based distance relay. Simulation results also show exceptional robustness of the novel ART algorithm for all operating conditions and events studied, as well as superior classification capabilities compared to the other solutions. Consequently, it is demonstrated that the proposed ART solution may be used for accurate, high-speed distinction among faulted and unfaulted events, and estimation of fault type and fault section.

To My Parents for Their Love and Support

ACKNOWLEDGMENTS

This research was part of the project "Context-Dependent Network Agents: Complex Interactive Networks/Systems", supported by contract # WO 8333-05 between the Electric Power Research Institute and Carnegie Mellon University, and has been carried out by Texas A&M University under the subcontract # 542995-42590 titled "Self-Evolving Agents for Monitoring, Control and Protection of Large, Complex Dynamic Systems". Parts of this study are also funded by the Power Systems Engineering Research Center and the National Science Foundation through the Industry/University Cooperative Research Center Program.

I would like to express my sincere gratitude to my advisor, Dr. Mladen Kezunovic, for his time, guidance and support. His great technical knowledge and experience contributed enormously to this dissertation.

In addition, I am also grateful to other members of my dissertation committee: Dr. Ali Abur, Dr. Shankar Bhattacharyya and Dr. William Lively, for their precious time and valuable comments.

TABLE OF CONTENTS

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

FIGURE Page

FIGURE Page

FIGURE Page

FIGURE Page

CHAPTER I

INTRODUCTION

A. Problem Statement

The problem of detecting and classifying transmission line faults based on three-phase voltage and current signals has been known for a long time. It was solved some time ago by introducing the traditional relaying principles such as overcurrent, distance, overvoltage, undervoltage, differential, etc, implemented using either models of the transmission line or models of fault signals [1]. All of these principles are based on a comparison between the measurements and predetermined setting calculated taking into account only predetermined operating conditions and fault events. Consequently, if the actual power system conditions deviate from the anticipated ones, the measurements and settings determined by the classical relay design have inherent limitations in classifying certain fault conditions, and the performance of classical protective relays may significantly deteriorate [2]. Subsequently, a more sensitive, selective, and reliable relaying principle is needed, capable of classifying the faults under a variety of unfolding operating conditions and faults.

This dissertation introduces a new protective relaying principle for transmission lines that is based on pattern recognition instead of using traditional setting. The approach utilizes an artificial neural network algorithm where the prevailing system conditions are taken into account through the learning mechanism. Artificial neural networks generally possess an ability to capture complex and nonlinear relationships

This dissertation follows the style of *IEEE Transaction on Power Delivery.*

between inputs and outputs through a learning process. They are particularly useful for solving difficult signal processing and pattern recognition problems. The new protective relaying concept is based on a special type of artificial neural network, called Adaptive Resonance Theory (ART), ideally suited for classifying large, highly-dimensional and time-varying sets of input data [3, 4]. The new classification approach has to reliably conclude, in a rather short time, whether, where and which type of fault occurs under a variety of operating conditions [5]. Samples of current and voltage signals from three transmission line phases are recognized as the features of various disturbance and fault events in the power network, and aligned into a set of input patterns. Through combined unsupervised and supervised learning steps, neural network establishes categorized prototypes of typical input patterns. In the implementation phase, trained neural network classifies new events into one of the known categories based on an interpretation of the match between the corresponding patterns and set of prototypes using fuzzy decision rule.

B. Research Topics

Proposed neural network algorithm for relay protection has already been used in its original form and with conservative assumptions about the power network and actual operating conditions [6, 7, 8, 9]. This study introduces several enhancements of the mentioned original version of the algorithm: suitable preprocessing of neural network inputs, improvements in the supervised learning, fuzzyfication of neural network outputs, utilization of on-line learning, and extensive algorithm evaluation for several test sets, covering a variety of power system operating conditions and events.

Preprocessing of neural network input signals includes selection of measurements, analog filters, sampling frequencies, data windows and scaling methods. The preprocessing determines the number of neural network inputs and usability of constructed input patterns, and influences the trade-off between performing the classification task more accurately and making the decision quickly and in real time [10, 11]. The supervised learning phase of neural network training is redefined, allowing improved generation of pattern prototypes with more consistent matching of the density of input pattern set [12]. Original version of the algorithm had a simple interpretation of neural network outputs, and during implementation it classified a new pattern according to the most similar prototype. The new idea of interpretation of neural network outputs through a fuzzy classifier (fuzzy decision rule) is an advanced generalization concept that classifies a new pattern according to the weighted contribution from several similar prototypes [12, 11]. It provides smoother decision regions between the categories, and is especially beneficial in improving algorithm selectivity for a variety of the real events which are rather dissimilar to all events presented during training. Since proposed neural network is inherently characterized by incremental learning capability [3], an algorithm for on-line learning during varying operating conditions is enabled, to preserve attained recognition capabilities. In addition, Multilayer Perceptron (MLP) neural network algorithm and impedance based distance protection algorithms are developed and implemented for comparison with the proposed ART neural network algorithm.

Selected model of an actual power network is used to simulate extensive sets of interesting scenarios, covering a variety of power system conditions, and typical fault and no-fault events [13, 14, 15]. Implemented scenarios allow improved algorithm training and evaluation, and demonstrate the benefits of the new approach [12, 16]. Specially developed simulation environment allows feasibility of proposed comprehensive algorithm design and evaluation [17, 15, 18, 19]. In this simulation environment training and testing procedures can easily be controlled and performed through interfacing appropriate simulation programs, exchanging their simulation parameters and results, and using appropriate graphical interface.

C. Dissertation Outline

The dissertation is organized as follows. A background of power systems, protective relaying, neural networks and fuzzy logic is provided in Chapter II. Chapter III summarizes applications of neural networks and fuzzy logic to protective relaying, describes the previous relevant approaches and gives an overview of the proposed approach. The neural network algorithm training and implementation steps are explained in Chapter IV. A new technique for fuzzyfication of neural network outputs is introduced in Chapter V. Required hardware and software modules for proposed protective relaying solution, and the complex neuro-fuzzy abstraction mechanism are outlined in Chapter VI. Selected model of an actual power network and power system scenarios are presented in Chapter VII. Chapter VIII defines a new simulation environment for protective algorithm design and evaluation. Chapter IX specifies implementation steps, including scenario simulation, pattern generation, and the design of ART and MLP neural network and impedance based distance relay algorithms. The simulation results are shown and discussed in Chapter X. The conclusions are given in Chapter XI. References related to proposed work, and Appendix with power system parameters are enclosed at the end.

CHAPTER II

BACKGROUND

A. Introduction

This chapter presents background relevant to proposed research topics and is organized into five sections. First section describes a role and structure of power system, the effect of power system faults, and emphasizes the concept of protective relaying. Next section focuses on traditional protective relaying of transmission lines. The most common relaying principle, distance protection, is analyzed in third section. Artificial intelligence tools, neural networks and fuzzy logic, suitable to be used for new, pattern recognition based relaying principle, are introduced in fourth and fifth sections, respectively.

B. Power System and the Concept of Protective Relaying

A power system is a grid of electrical sources, loads, transmission lines, power transformers, circuit breakers, and other equipment, connected to provide power generation, transmission and distribution. A simple example of a typical power system is given in Fig. 1. Transmission lines link the sources and loads in the system, and allow energy transmission. The loss of one or more transmission lines is often critical for power transfer capability, system stability, and sustainability of required system voltage and frequency. Circuit breakers are devices for connecting power system components and carrying transmission line currents under normal operating conditions, and interrupting the currents under specified abnormal operating conditions. They are located at the transmission line ends and by interrupting the current they isolate the fault from the system.

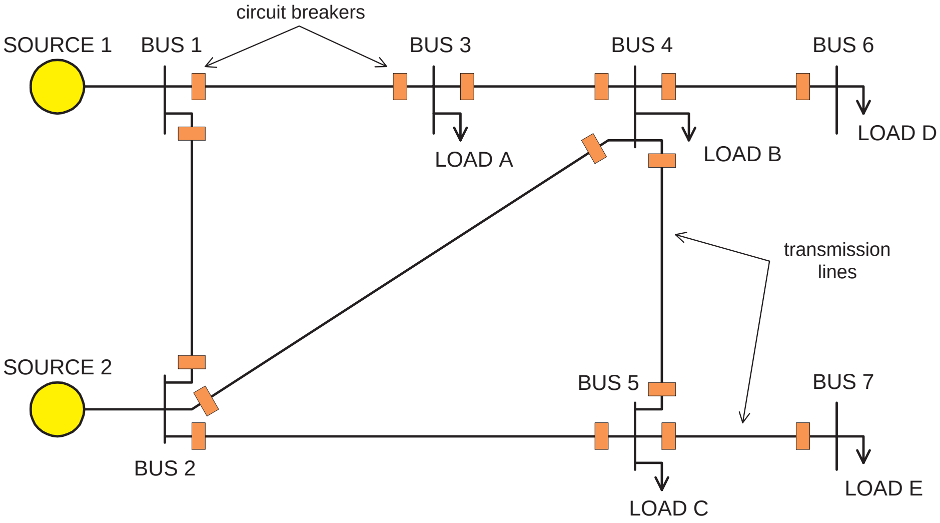


Fig. 1. Example of a power system

The purpose of protective relaying is to minimize the effect of power system faults by preserving service availability and minimizing equipment damage [20]. Since the damage caused by a fault is mainly proportional to its duration, the protection is required to operate as quickly as possible. Transmission line faults happen randomly, and they are typically an outcome of severe weather or other unpredictable conditions. Varying fault parameters as well as conditions imposed by actual network configuration and operating mode determine the corresponding transient current and voltage waveforms detected by the relays at line ends. The main fault parameters are:

- type of fault defined as single-phase-to-ground, phase-to-phase, two-phase-to-ground, three-phase, and three-phase-to-ground;
- fault distance defined as distance between the relay and faulted point;
- fault impedance defined as impedance of the earth path;

- fault incidence time defined as time instance where the fault occurs on the voltage waveform.

C. Traditional Protective Relaying for Transmission Lines

Protective relays are intelligent devices located at both ends of each transmission line, nearby network buses or other connection points [2, 21, 22, 23]. They enable high-speed actions required to protect the power network equipment when operating limits are violated due to an occurrence of a fault. The relay serves as primary protection for corresponding transmission line as well as a backup for the relays at adjacent lines, beyond the remote line end. Although a transmission line is protected by the relays in each phase, for simplicity, only one phase connection is illustrated in Fig. 2. The transmission line fault detection and clearance measurement and control chain contains instrument transformers, distance relay and circuit breaker. Instrument transformers acquire voltage and current measurements (VT and CT) and reduce the transmission level signals into appropriate lower level signals convenient to be utilized by the relay. The main role of protective relays is recognizing the faults in particular area in power network based on measured three-phase voltage and current

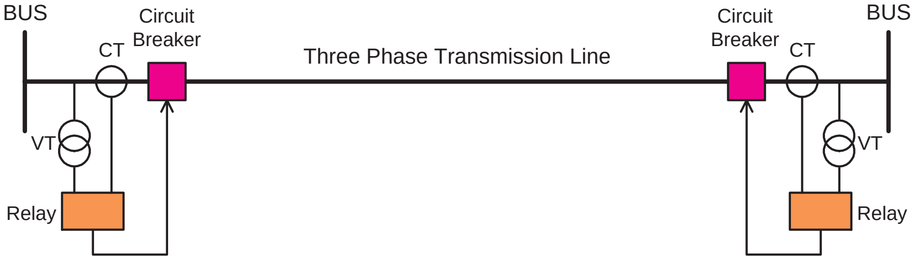


Fig. 2. Transmission line with fault detection and clearance measurement and control chain

signals. In a very short time, around 20 ms, relay has to reliably conclude whether and which type of fault occurs, and issue a command for opening the circuit breaker to disconnect faulted phases accordingly.

The relay responsibility for protection of a segment of the power system is defined by a concept of zones of protection. A zone of protection is a setting defined in terms of a percentage of the line length, measuring from the relay location. When fault occurs within the zone of protection, the protection system activates circuit breakers isolating the faulty segment of the power system defined by the zone boundaries. The zones of protection always overlap, in order to ensure back-up protection for all portions of the power system.

Fig. 3 shows a small segment of a power system with protection zones enclosed by dashed lines distinctly indicating the zones for the relays located at A, B, C and

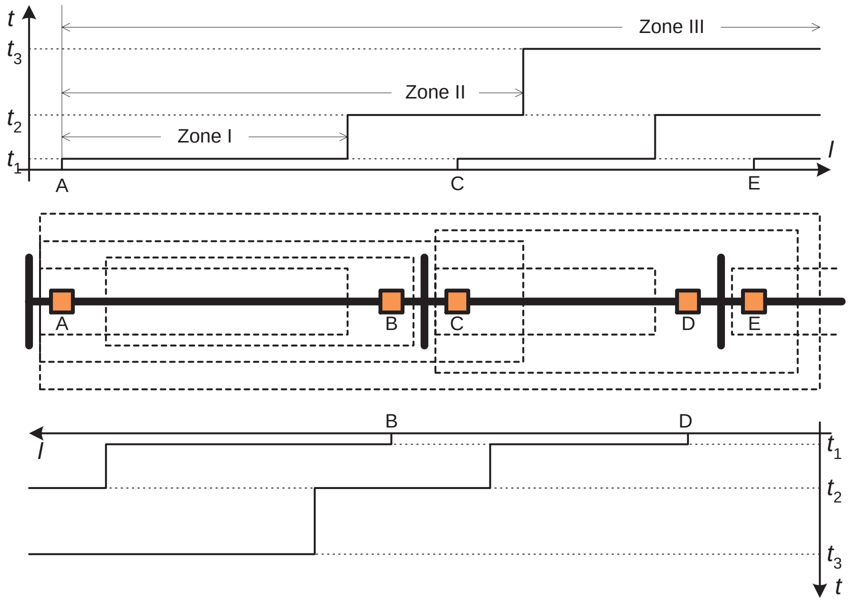


Fig. 3. The principle of relay coordination

D. For instance, relay at A has three zones of protection, where zone I is the zone of primary protection, while zones II and III are backup protections if relays at C and E missoperate, respectively for the faults in their zone I. Coordination of protective devices requires calculation of settings to achieve selectivity for faults at different locations. Zone I setting is selected for relay to trip with no intentional delay, except for the relay and circuit breaker operating time t_1. The zone I is set to underreach the remote end of the line, and usually covers around 80-85% of the line length. Entire line length cannot be covered with zone I because the relay is not capable of precisely determining fault location. Hence instant undesired operation may happen for the faults beyond remote bus. The purpose of zone II is to cover the remote end of the line which is not covered by the zone I. A time delay t_2 for the faults in zone II is required to allow coordination with the zone I of the relay at adjacent line. Zone II setting overreaches the remote line end and typically is being set to 120% of the line length. The zone III setting provides back up protection for adjacent line and operates with the time delay t_3. Typical setting for zone III is 150-200% of the primary line length, and is limited by the shortest adjacent line.

D. Distance Relaying

In the traditional protective relaying approach, a distance relay is the most common relay type for protection of multiterminal transmission lines. It operates when the apparent impedance seen by the relay decreases bellow a setting value [1, 24]. Relay settings are determined through a range of short circuit studies using the power system model and simulations of the worst-case fault conditions. Simplified examples of a single-phase electrical circuit, with voltage source, load, and pre-fault and faulted transmission line with neglected shunt susceptances are shown in Figs. 4

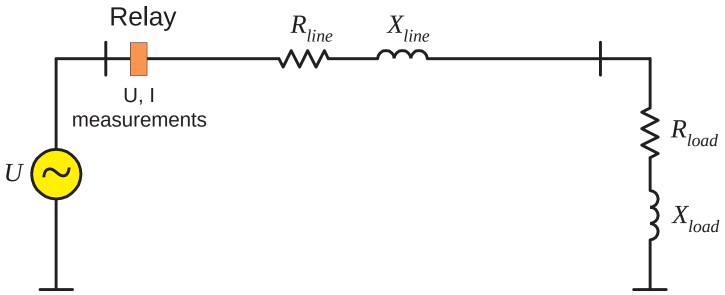


Fig. 4. Equivalent circuit of transmission line during pre-fault condition

and 5, respectively. The conditions before the fault determine the impedance ”seen” by the relay:

$$Z_{line} = R_{line} + jX_{line}, \qquad R_{line} \ll X_{line} \tag{2.1}$$

$$Z_{load} = R_{load} + jX_{load}, \qquad R_{load} > X_{load} \tag{2.2}$$

$$Z_{pre-fault} = \frac{U_{pre-fault}}{I_{pre-fault}} = Z_{line} + Z_{load} \tag{2.3}$$

The distance relay operates on the principle of dividing the voltage and current phasors to measure the impedance from the relay location to the fault. Condition in (2.1) is due to highly inductive impedance of a transmission line, while condition in (2.2)

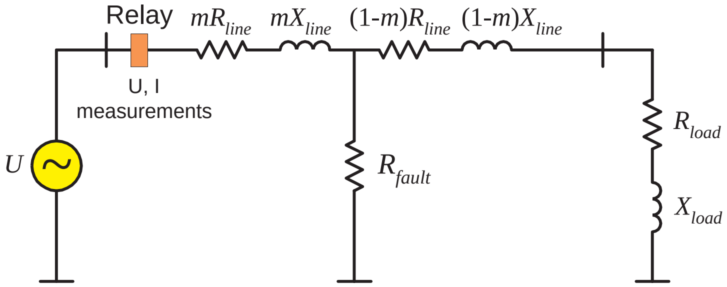


Fig. 5. Equivalent circuit of transmission line during fault condition

is due to mostly resistive load characteristic. The impedance ”seen” by the relay at the moment of the fault is given as:

$$Z_{fault} = mZ_{line} + ((1-m)Z_{line} + Z_{load}) \| R_{fault} =$$

$$= mZ_{line} + \frac{((1-m)Z_{line} + Z_{load})R_{fault}}{(1-m)Z_{line} + Z_{load} + R_{fault}} \tag{2.4}$$

$$Z_{fault} = \frac{U_{fault}}{I_{fault}} \approx mZ_{line} + R_{fault}, \qquad R_{fault} \ll R_{load} \tag{2.5}$$

The apparent impedance at the relay location is equal to the quotient of measured phase voltage and current, according to (2.3) and (2.5). Graphical representation of the measured impedance is usually shown in the complex, resistance-reactance or R-X plane given in Fig. 6, where the origin of the plane represents the relay location and the beginning of the protected line. For phase and ground faults which do not include

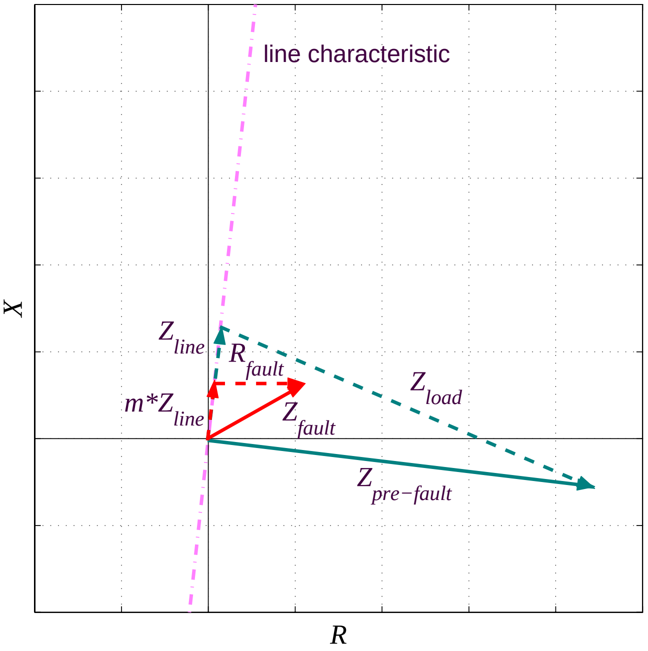


Fig. 6. Apparent impedance seen from relay location during pre-fault and fault conditions

any fault resistance, the impedance vector lies on the straight line representing the transmission line characteristic impedance.

Comparing (2.3) and (2.5), the impedance measured at the relay location under normal load conditions is generally significantly higher than the impedance during the fault, which usually includes arc and earth resistance. It depends on the fault type and location in the system where it occurs. The apparent impedance also varies with fault impedance, as well as impedance of equivalent sources and loads. However, in real situations problem of detecting apparent impedance is not as simple as the single-phase analysis given for simplified network and transmission line may suggest. The interconnection of many transmission lines imposes a complex set of relay settings [25, 26].

The impedance computed by the distance relay has to be compared against the relay settings. Settings are defined as the operating characteristics in the complex R-X plane that enclose all faults along protected section and takes into account varying fault resistance. A distance relay is designed to operate whenever estimated impedance remains within a particular area on the operating characteristic during specified time interval.

Impedance relay characteristics for three protection zones are shown in Fig. 7, where *mho* characteristic is designed for faults not including ground, while *quadrilateral* characteristic is designed for faults including ground. The *mho* operating characteristic is a circle with the center in the middle of impedance of the protected section. The *quadrilateral* operating characteristic is a polygon, where the resistive reach is set to cover the desired level of ground fault resistance, but also to be far enough from the minimum load impedance. The typical distance relaying algorithm involves three sets of three zone characteristics for each phase-to-phase and phase-to-ground fault. As long as voltages and currents are being acquired, the apparent

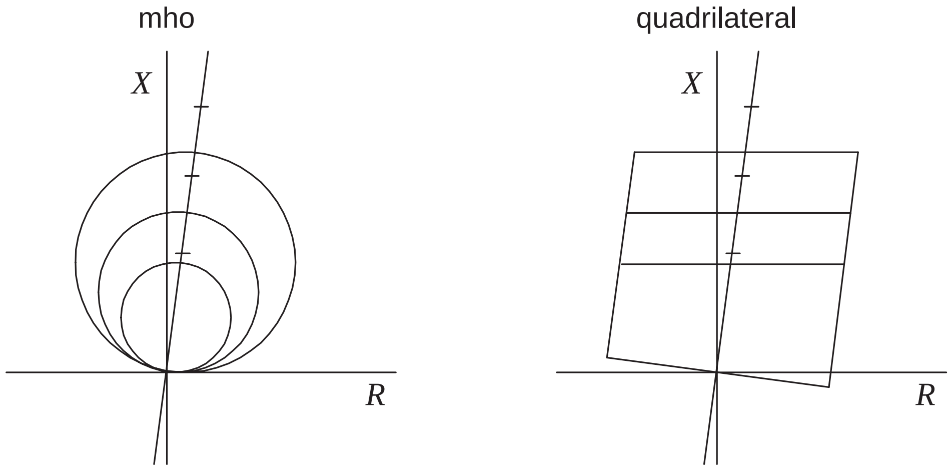


Fig. 7. Mho and quadrilateral distance relay characteristics

impedance is estimated by assuming each of possible phase-to-phase and phase-to-ground fault types. Selection of either *mho* or *quadrilateral* characteristic depends on value of zero sequence current. An existence of zero sequence current indicates ground fault and *quadrilateral* characteristic should be considered, otherwise fault does not include ground and *mho* characteristic should be examined. When the operating characteristic has been chosen, the impedance is checked against all three sets of individual settings. Depending on type and location of an actual fault, only the corresponding apparent impedances will fall inside their operating characteristics.

E. Neural Networks

Generally, power systems are highly non-linear and large-scale systems, possessing statistical nature and having subsystems whose model usually cannot easily be identified. Quite often there is no suitable analytical technology to deal with this degree of complexity and the number of computational possibilities is too high leading

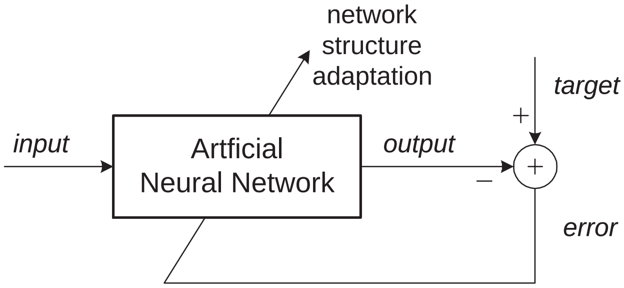


Fig. 8. The principle of Artificial Neural Networks

to unsatisfactory solutions. Presented problems can successfully be overcome using artificial neural networks (ANNs).

It is generally known that all biological neural functions, including memory, are stored in the very complex grids of neurons and their interconnections. Learning is a process of acquiring data from the environment and establishing new neurons and connections between neurons or the modifications of existing connections. An idea of artificial neural networks, shown in Fig. 8, is to construct a small set of simple artificial neurons and train them to capture general, usually complex and nonlinear, relationships among the data. Neural networks are parallel implementations of nonlinear static or dynamic systems, and their numerous independent operations can be executed simultaneously. The function of neural network is determined by structure of neurons, connection strengths, and the type of processing performed at elements or nodes. In classification tasks, the output being predicted is a categorical variable, while in regression problems the output is a quantitative variable. Neural network uses individual examples, like set of inputs or input-output pairs, and appropriate training mechanism to periodically adjust the number of neurons and weights of neuron interconnections to perform desired function. Afterwards, it possesses generalization ability, and can successfully accomplish mapping or classification of input signals that have not been presented in the training process.

The origin of neural networks theory began in the 1940s with the work of McCulloch and Pitts, who showed that networks with sufficient number of artificial neurons could, in principle, compute any arithmetic or logical function [27]. They were followed by Hebb, who presented a mechanism for learning in biological neurons [28]. The first practical application of artificial neural networks came in the late 1950s, when Rosenblatt invented the perceptron network and associated learning rule [29]. However, the basic perceptron network could solve only a limited class of problems. Few years later Widrow and Hoff introduced a new learning algorithm and used it to train adaptive learning neural networks, with structure and capability similar to the perceptron network [30]. In 1972 Kohonen [31] and Anderson [32] independently developed new neural networks that can serve as memories, by learning association between input and output vectors. Grossberg was investigating the self-organizing neural networks, capable of performing error correction by themselves, without any external help and supervision [33, 34]. In 1982, Hopfield described the use of statistical analysis to define the operation of a certain class of recurrent networks, which could be used as an associative memory [35]. The backpropagation algorithm for training multilayer perceptron neural networks was discovered in 1986 by Rumelhart and McClelland [36, 37]. Radial basis function networks was invented in 1988 by Broomhend and Lowe as an alternative to multilayer perceptron networks [38]. In the 1990s, Vapnik invented support vector machines - powerful class of supervised learning networks [39]. In the last fifteen years theoretical and practical work in the area of neural networks has been rapidly growing.

Neural networks are used in a broad range of applications, including pattern classification, function approximation, data compression, associative memory, optimization, prediction, nonlinear system modeling, and control. They are applied to a wide variety of problems in many fields including aerospace, automotive, banking, chemistry, defense, electronics, engineering, entertainment, finance, games, manufacturing, medical, oil and gas, robotics, speech, securities, telecommunications, transportation.

Generally, there are three categories of artificial neural networks: feedforward, feedback and competitive learning. In the feedforward networks the outputs are computed directly from the inputs, and no feedback is involved. Recurrent networks are dynamical systems and have feedback connections between outputs and inputs. In the competitive learning networks the outputs are computed based on some measure of distance between the inputs and actual outputs. Feedforward networks are used for pattern recognition and function approximation, recurrent networks are used as associative memories and for optimization problems, and competitive learning networks are used for pattern classification.

Pattern recognition or classification is learning with categorical outputs [40]. Categories are symbolic values assigned to the patterns and connect the pattern to the specific event which that pattern represents. Categorical variables take only a finite number of possible values. The union of all regions where given category is predicted is known as the decision region for that category. In pattern classification the main problem is estimation of decision regions between the categories that are not perfectly separable in the pattern space. Neural network learning techniques for pattern recognition can be classified into two broad categories: unsupervised and supervised learning. During supervised learning desired set of outputs is presented together with the set of inputs, and each input is associated with corresponding output. In this case, neural network learns matching between inputs and desired outputs (targets), by adjusting the network weights so as to minimize the error between the desired and actual network outputs over entire training set. However, during unsupervised learning, desired outputs are not known or not taken into account, and network learns to identify similarity or inner structure of the input patterns, by adjusting the network weights until similar inputs start to produce similar outputs.

The most commonly used type of feedforward neural networks is Multilayer Perceptron (MLP) [41]. It has multiple layers of parallel neurons, typically one or more hidden layers with inner product of the inputs, weights and biases and nonlinear transfer functions, followed by an output layer with inner product of the inputs, weights and biases and linear/nonlinear transfer functions, shown in Fig. 9. Use of nonlinear transfer functions allows the network to learn complex nonlinear relationships between input and output data.

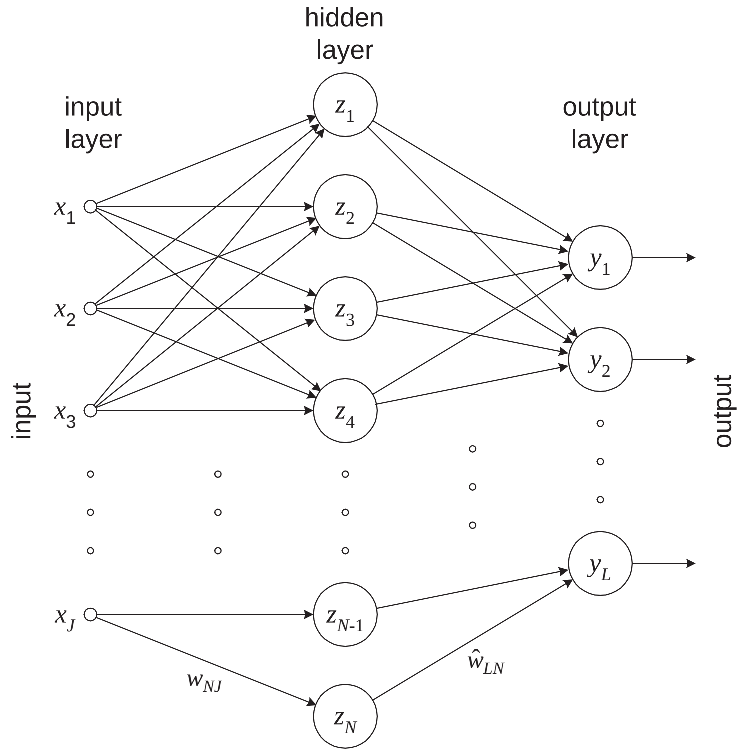


Fig. 9. The structure of Multilayer Perceptron (MLP) neural network

Competitive or Self-Organized neural networks try to identify natural groupings of data from a large data set through clustering. The aim of clustering is to allocate input patterns into much smaller number of groups called clusters, such that each pattern is assigned to unique cluster. Clustering is the process of grouping the data into groups or clusters so that objects within a cluster have high similarity in comparison to one another, but are dissimilar to objects in other clusters. Two widely used clustering algorithms are K-means [42] where the number of clusters is given in advance, and ISODATA where the clusters are allocated incrementally [43]. The Competitive networks learn to recognize the groups or clusters of similar input patterns, each having members that are as much alike as possible. The similarity between input patterns is estimated by some distance measure, usually by the Euclidean distance. Neural network neurons are cluster centers defined as prototypes of input patterns encountered by the clusters. During learning, each prototype becomes sensitive and triggered by a different domain of input patterns. When learning is completed a set of prototypes represents the structure of input data [44, 45, 46]. For supervised classification, each cluster belongs to one of existing categories, and number of neural network outputs corresponds to desired number of categories, determined by the given classification task.

The Competitive Learning network consists of two layers. The first layer performs a correlation between the input pattern and the prototypes. The second, output layer performs a competition between the prototypes to determine a winner that indicates which prototype is best representative of the input pattern. The associate learning rule [32, 31] is used to adapt the weights in a competitive network, and typical examples are the Hebb [28] in (2.6) and Kohonen rule [47] in (2.7). In Hebb rule, winning prototype $\mathbf{w}_{ij}$ is adjusted by moving toward the product of input $\mathbf{x}_i$

and output $\mathbf{y}_j$ with learning rate α

$$\mathbf{w}_{ij}(k) = \mathbf{w}_{ij}(k-1) + \alpha \mathbf{x}_i(k) \mathbf{y}_j(k) \tag{2.6}$$

In Kohonen rule, the winning prototype $\mathbf{w}_{ij}$ is moved toward the input $\mathbf{x}_i$

$$\mathbf{w}_{ij}(k) = \mathbf{w}_{ij}(k-1) + \alpha(\mathbf{x}_i(k) - \mathbf{w}_{ij}(k-1)), \quad \text{for} \quad i = \min_i \|\mathbf{x}_i(k) - \mathbf{w}_{ij}(k-1)\| \tag{2.7}$$

Typical example of Competitive Learning networks is Vector Quantization (VQ) network [48, 49], where the prototype closest to the input pattern is moved toward that pattern. Competitive Learning networks are efficient adaptive classifiers, but they suffer from certain problems. The first problem is that the choice of learning rate forces a trade-off between the speed of learning and stability of the prototypes. The second problem occurs when clusters are close together such that they may overlap and encompass patterns already encountered by other clusters. The third problem is that occasionally an initial prototype is located so far from any input pattern that it never wins the competition, and therefore never learns. Finally, a competitive layer always has as many clusters as neurons. This may not be acceptable for some applications when the number of clusters cannot be estimated in advance. Some of the problems discussed can be solved by the Self-Organized-Feature-Map (SOFM) networks [50, 51, 47, 52, 48] and Learning Vector Quantization (LVQ) networks [53, 54]. Self-Organized-Feature-Maps learn to capture both the distribution, as competitive layers do, and topology of the input vectors. Learning Vector Quantization networks try to improve Vector Quantization networks by adapting the prototypes using supervised learning. Mentioned Competitive Learning networks, like many other types of neural network learning algorithms, suffer from a problem of unstable learning, usually called the stability/plasticity dilemma. If a learning algorithm is plastic or sensitive to novel inputs, then it becomes unstable and with certain degree forgets prior learning. Stability-plasticity problem has been solved by the Adaptive Resonance Theory (ART) neural networks, which adapts itself to new inputs, without destroying past training.

ART neural network is a modified type of competitive learning, used in the Grossberg network [33, 34], and has unique concept in discovering the most representative positions of prototypes in the pattern space [3, 55, 56, 57]. Similarly to SOFM and LVQ networks, the prototype positions are dynamically updated during presentation of input patterns [48]. However, contrary to SOFM and LVQ, the initial number of clusters and cluster centers are not specified in advance, but the clusters are allocated incrementally, if presented pattern is sufficiently different from all existing prototypes. Therefore, ART networks are sensitive to the order of presentation of the input patterns. The main advantage of ART network is an ability to self-adjust the underlying number of clusters. This offers flexible neural network structure that can handle an infinite stream of input data, because their cluster prototype units contain implicit representation of all the input patterns previously encountered. Since ART architectures are capable of continuous learning with non-stationary inputs, the on-line learning feature may be easily appended.

The ART architecture, shown in Fig. 10, consists of input, hidden, and output layers and their specific interconnections: hidden to output layer activations, output to hidden layer expectations, the mismatch detection subsystem and gain control. The key innovation of ART is the use of expectation or resonance, where each input pattern is presented to the network and compared with the prototype that it most closely matches. When an input pattern is presented to the network, it is normalized and multiplied by cluster prototypes, i.e. weights in the hidden-output layer. Then, a competition is performed at output layer to determine which cluster prototype is closest to the input pattern. The output layer employs a ”winner-take-all” compe-

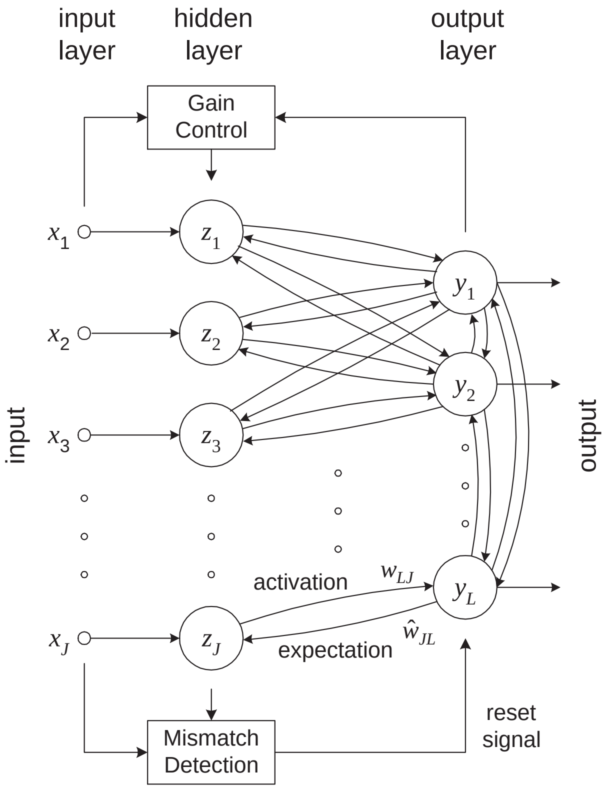


Fig. 10. The structure of Adaptive Resonance Theory (ART) neural network

tition, leaving only one unit with non-zero response. Thus, input pattern activates one of the cluster prototypes. When a prototype in output layer is activated, it is reproduced as an expectation at the hidden layer. Hidden layer then performs a comparison between the expectation and the input pattern, and the degree of their match is determined. If match is adequate (resonance does occur) the pattern is added to the winning cluster and its prototype is updated by moving toward input pattern. When the expectation and the input pattern are not closely matched (resonance does not occur), the mismatch is detected and causes a reset in the output layer. This reset signal disables the current winning cluster, and the current expectation is removed to allow learning of new cluster. In this way, previously learned prototypes are not affected by new learning. The amount of mismatch required for a reset is determined by the controlled gain or vigilance parameter that defines how well the current input should match a prototype of the winning cluster. The set of input patterns continues to be applied to the network until the weights converge, and stable clusters are formed. Obtained cluster prototypes generalize density of input space, and during implementation have to be combined with the K-Nearest Neighbor (K-NN) or any other classifier for classification of new patterns [40, 58].

There are many different variations of ART available today. ART1 performs unsupervised learning for binary input patterns [3], while ART2 is modified to handle analog input patterns [55]. ART-2A is version of the ART2 with faster learning enabled [59]. ART3 performs parallel searches of distributed prototypes in a hierarchical network structure [56]. ARTMAP is a supervised version of ART1 [57]. There are also many other neural networks derived from mentioned basic ART structures.

F. Fuzzy Logic

Power systems are large, complex, broadly distributed systems, influenced by unexpected events. The mathematical formulations of power systems and related events are derived under restrictive assumptions and allow many uncertainties. The system model is not often available, and its operations are based on many vague constraints and multiple conflicting objectives. Computational burden for solving such system using conventional techniques becomes significant. The experience of human experts can be represented as functional knowledge in terms of heuristic rules. Consequently, fuzzy logic is the technique which can successfully handle imprecise, vague of "fuzzy" information present in power systems.

The term fuzzy logic has been used in two different senses. In a narrow sense, fuzzy logic is a logical system that generalizes classical two-valued logic, which is an extension of multivalued logic for reasoning under uncertainty. In a broad sense, fuzzy logic is the theory of fuzzy sets as a generalization of conventional set theory, by introducing graded membership and relates to sets with soft boundaries [60].

Fuzzy set theory derives from the fact that almost all natural classes and concepts are fuzzy rather than crisp in nature. Fuzzy logic systems provide an excellent framework to model uncertainty and imprecision in human reasoning with the use of linguistic variables with membership functions.

Fuzzy logic was motivated by two objectives. First, it aims to mitigate difficulties in developing and analyzing complex systems encountered by conventional mathematical tools. Second, it is motivated by observing that human reasoning can utilize concepts and knowledge that do not have well-defined, sharp boundaries. The concept of fuzzy-set theory can serve as an interface between a linguistic (qualitative) and numeric (quantitative) variables. It provides an efficient solution for modeling complex systems that involve numeric variables and qualitative description.

With conventional probabilistic and deterministic classifiers, the features characterizing the input data are quantitative (numerical) in nature. Impreciseness or ambiguity in such data may arise from various sources, like measuring instrument error, noise, etc. For these reasons, it may become convenient to use linguistic variables in order to describe feature information. In such cases, it is not appropriate to give an exact numerical representation to uncertain feature data. Rather, it is reasonable to represent uncertain feature information by fuzzy subsets. Fuzzy set theory uses linguistic variables, rather than quantitative variables, to represent uncertainty. A linguistic variable enables its value to be described both qualitatively by a linguistic term and quantitatively by a corresponding membership function. Fuzzy logic can be used to easily translate heuristic reasoning and knowledge from qualitative descriptions to quantitative descriptions. It has two important components: fuzzy rules and membership functions.

Fuzzy logic is conceptually easy to understand, because the mathematical concepts of fuzzy reasoning are very simple. It is tolerant of imprecise data and can handle ambiguity. It can model nonlinear functions of arbitrary complexity. Fuzzy logic can be built using the experience of experts, and can resolve conflicting objectives. It is flexible and is relatively easy to implement.

The idea of fuzzy set theory was introduced in 1965 by Zadeh as a new approach to represent vagueness in everyday life [61]. The concept of fuzzy sets is the idea of grade of membership. The first work on the application of fuzzy set theory to pattern classification was given in 1966 by Belman, Kalaba and Zadeh, where the problem of pattern classification was formulated as the problem of interpolation of the membership function of a fuzzy set [62]. A cluster analysis based on concept of a fuzzy partition was given in 1970 by Rupini [63]. Fuzzy clustering algorithms, the Fuzzy ISODATA, and the Fuzzy C-Means algorithms, were introduced by Dunn in 1973 [64]. During the same year Bezdek introduced the first cluster validity measure for fuzzy partitions generated by Fuzzy C-Means [65]. The fuzzy K-Nearest Neighborhood algorithm was developed in 1985 by Keller [66]. In 1989, Gath and Geva developed an approach for assessing the number of clusters using a performance based validity criteria [67]. The theory of fuzzy sets has been applied to ART neural networks to develop their fuzzy analogies, including Fuzzy ART [68] and Fuzzy ARTMAP [69].

Fuzzy models attempt to capture and quantify nonrandom imprecision. The sets of realistic objects do not have precisely defined membership. Conventional or crisp sets contain objects that satisfy precise properties required for membership. Fuzzy sets, on the other hand, contain objects that satisfy imprecise properties to varying degrees. Fuzziness describes event ambiguity and is a type of deterministic uncertainty. It measures the degree to which an event occurs, not whether it occurs.

A fuzzy set is a set without a crisp, clearly defined boundary, but with smooth boundaries. Fuzzy set theory generalizes classical set theory to allow partial or gradual membership. Fuzzy sets provide a natural way of dealing with problems in which the source of imprecision is the absence of sharply defined criteria of class membership rather than the presence of random variables. It directly addresses the limitation of classical sets that always have sharp boundaries, by allowing membership in a set to be a matter of degree. The notion of membership in fuzzy sets thus becomes a matter of degree, which is a number between 0 and 1. A membership degree of 0 represents complete nonmembership, while membership degree of 1 represents complete membership. This way, a smooth and gradual transition from the regions outside the set to those in the set can be described.

A membership function is a curve that defines how each point in the input space is mapped to a membership value between 0 and 1. The membership function value

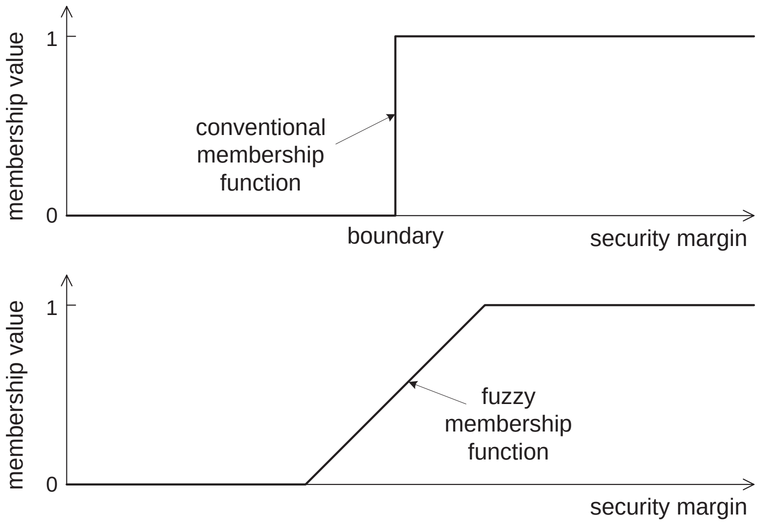


Fig. 11. Membership functions: conventional and fuzzy

measures the degrees to which objects satisfy imprecisely defined properties. Fig. 11 clearly illustrates the membership functions for a crisp set and for a fuzzy set. Mostly used shapes for membership functions are triangular, trapezoidal, piecewise linear and Gaussian.

Let U be the universal set. Each crisp set A of U can be described by a characteristic function of a membership function

$$\mu_A : U \rightarrow \{0, 1\},\ u \in U \tag{2.8}$$

where μ_A is 1 if $u \in A$, and 0 otherwise. Conventional set theory allows the statement to have only one of two truth values: true or false, usually written 1 or 0. A fuzzy set B of U is described by its membership function

$$\mu_B : U \rightarrow [0, 1],\ u \in U \tag{2.9}$$

where μ_B expresses the degree in which the element u belongs to B, and is called a degree of membership. Fuzzy set theory is a natural generalization of conventional set theory by allowing variables to have degree of truth. A fuzzy set is characterized by a membership function $\mu_F(x)$ which takes only values in an interval. Crisp set is a special case of fuzzy set where the degrees of membership of the elements take only values 0 and 1.

Conventional logic uses intersection, union and complement operations, and any relationship can be described by combinations of these operators. However, in fuzzy logic, union, intersection and complement are defined in terms of their membership functions.

A fuzzy logic system usually maps crisp inputs into crisp outputs, and in such case contains four major components: fuzzification, rules, inference engine and defuzzification. Fuzzification converts input data into suitable linguistic values which may be viewed as labels of fuzzy sets. If-then rule statements are used to formulate the conditional statements that comprise fuzzy logic. The rules are a set of linguistic statements based on expert knowledge, including experience and heuristics, instead of detailed mathematical model. In the fuzzy inference engine, fuzzy logic principles are used to combine fuzzy rules into a mapping from fuzzy input sets to fuzzy output sets. The process of fuzzy inference involves: membership functions, fuzzy logic operators, and if-then rules. Mathematically, fuzzy rule-based inference can be viewed as an interpolation scheme because it enables the fusion of multiple fuzzy rules when their conditions are all satisfied to a degree. Defuzzification produces a crisp output from the fuzzy set that is the output of the inference engine.

The concept of fuzzy set fits very naturally into the framework of pattern recognition and copes with uncertainty in the feature space and classification space. The concept of fuzzy sets can be used at the feature level for representing input data as an array of membership values denoting the degree of possession of certain properties. At the classification level, it can be used for representing category membership of objects, and for providing a representation of missing information in terms of membership values. Therefore, fuzzy set theory can be used for handling of uncertainties in various stages of pattern recognition. The central problem of pattern classification is the problem of interpolation of the membership functions and corresponds to the problem of learning from examples in neural network theory.

Uncertainty in classification or clustering of patterns may arise from the overlapping nature of the various categories. In conventional classification techniques, the process consists of assigning a pattern to belong to only one category, which is neither true physically nor mathematically. Thus, feature vectors and the objects they represent can and should be allowed to have degrees of membership in more than one category. In the fuzzy-set approach, the category membership of a pattern is a fuzzy set, and a pattern does not necessarily belong to just one category. The concept of fuzzy subsets offers special advantages over conventional clustering by allowing algorithms to assign each object a partial or distributed membership in each of the clusters. This way, a smooth and gradual transition from the region inside the set to those outside the set can be described.

A fuzzy classifier is any classifier which uses fuzzy sets either during its training or during its operation. In the classifier design, various situations can be present: input data having fuzzy labels, classifier outputs having fuzzy labels, and models that are fuzzy producing crisp output labels.

Fuzzy classifiers are needed because in some problems there is insufficient information to properly implement classical methods. Furthermore, not only the category label of an object may be needed but also some additional information. The characteristics of objects or category labels are conveniently represented in terms of fuzzy sets. Moreover fuzzy set theory gives a mathematical tool for including and processing expert opinions about classification decisions, features and objects.

Fuzzy logic technique may be used for introduction of Fuzzy K-Nearest Neighbors classifier that generalizes the simple non-fuzzy K-Nearest Neighbors classifier, used so far. Fuzzy classifier can be combined with ART neural network and applied for improved interpretation of neural network outputs.

G. Summary

The protective relaying in power systems aims to recognize the faults in power network and initiate actions required to clear the fault and protect the power network equipment. The protective relaying problems were solved in the past using conventional techniques with many limitations. This dissertation proposes a new approach based on pattern recognition and utilization of artificial intelligence techniques: neural networks and fuzzy logic. Neural networks and fuzzy systems are similar in the sense that they both gain their generalization ability by learning from prior experience. Neural networks acquire knowledge through training, while fuzzy system collects the knowledge through heuristic reasoning. Stored knowledge is settled as a non-parametric model for estimating input-output functions, and produces correct outputs despite severe variations of input signals.

CHAPTER III

PREVIOUS WORK AND PROPOSED APPROACH

A. Introduction

This chapter addresses previous work of particular interest for the new protection approach proposed by this dissertation, and briefly outlines new approach as well. First section summarizes neural network and fuzzy logic applications developed so far for protective relaying of transmission lines. In the next section one peculiar protection approach based on special type of neural network is particularly emphasized, and used as a foundation for the novel approach that has to be developed. The design steps of that new approach are given in the last section, and will be studied in detail in following chapters.

B. Neural Networks and Fuzzy Logic Applications for Protective Relaying

The use of artificial intelligence techniques for protective relaying has been very attractive in recent years. Neural networks and fuzzy logic [70, 71, 72, 73, 74, 75, 76, 77] have been used to solve complex power system protection problems, particularly those where traditional approaches have difficulty in achieving the desired speed, accuracy and selectivity. Numerous applications of neural networks were used in the past intended to partially improve and substitute some of the standard relaying functions used in protection of transmission lines, originally realized by conventional techniques [78]. Typical functions that are comprised in the most important studies are fault detection and classification [79, 8, 80, 81, 82, 83, 84, 85, 86], fault direction discrimination [87, 88, 89], fault characterization (section estimation) [90, 82, 91, 92], adaptive relaying [93, 94], fault location [95, 96], autoreclosing [97, 98], fault analysis

[99, 100, 101], high impedance fault detection, alarm processing, disturbance detection and classification, etc. The applications are mainly based on widely used Multilayer Perceptron (MLP) feed-forward networks [102, 99, 79, 87, 90, 97, 95, 98, 80, 93, 81, 82, 101, 83, 96, 84, 91, 85], and in recent years on Radial Basis Function (RBF) [103, 99, 81, 85, 86, 92], Self-Organizing Maps (SOM) [48, 81, 94], Learning Vector Quantization (LVQ) [48, 81], Adaptive Resonance Theory (ART) [55, 8, 100], Recurrent [104, 88], Counterpropagation [105, 81], and Finite Impulse Response neural networks [106, 89]. Moreover, various fuzzy logic techniques have been used for fault detection and classification [107, 82], fault characterization, fault location [108], fault analysis [109], and disturbance detection and classification. An overview of neural networks and fuzzy logic utilization for main power system protection and monitoring functions is given in Table I.

The benefit of all utilized techniques in improving protection performance is demonstrated mostly only through the simulations. The development of their practical hardware and software implementation is still at the beginning. A comparison of proposed approach with the classical relaying techniques is rarely provided, even at the simulation level. Consequently, further studies are needed to enhance theoretical and application sides of the artificial intelligence concepts used for solving power system protection problems.

C. Previous Protection Approach

The specific nature of the real-time transmission line fault detection, classification and characterization tasks imposes important conditions for selection of neural network structure and learning strategies, because of several reasons:

Table I. Overview of neural networks and fuzzy logic applications for protective relaying

Function	Neural Networks					Fuzzy Logic
	MLP	RBF	SOFM, LVQ	ART	Others	
Fault Detection and Classification	[110] [111] [112] [113] [114] [79] [84] [117] [118] [119] [122] [80] [123] [124] [85] [125] [126] [83] [127] [81] [128] [129] [130] [120] [82]	[92] [85] [86] [81]	[115] [81] [120]	[100] [116] [121] [7] [9] [8]	[89] [81]	[111] [84] [107] [119] [123] [82]
Fault Direction Discrimination	[131] [132] [126] [87] [133] [135] [101]		[131]		[89] [134] [88]	
Fault Characterization	[136] [137] [90] [117] [119] [126] [140] [141] [142] [91]	[138] [139] [143]		[7] [9]		[138] [139] [143]
Adaptive Relaying	[94] [144] [145] [93] [146] [149] [150] [151] [152] [153]		[94] [147] [154]		[148]	
Fault Location	[155] [156] [96] [157] [84] [159] [119] [160] [158] [161] [95] [162] [91] [163]	[92] [158] [86]	[158]			[84] [108] [119]
Autoreclosing	[98] [79] [97] [164]				[165] [89] [166]	
High Impedance Fault Detection	[167] [168]					
Fault Analysis	[169] [99] [170] [128]	[99]		[121] [7] [9] [8]		[109] [171]
Alarm Processing	[172] [173] [174]					
Disturbance Detection and Classification	[175] [176]		[177] [178]	[177]		[178]

- large number of neural network inputs due to high dimensionality of pattern vectors constructed by taking samples of 3-phase voltages and currents in a given data window;

- large number of neural network outputs due to the given classification task of detecting the exact fault type and protection section where fault has occurred;

- large number of training patterns due to a variety of power network fault and disturbance events, and operating modes.

Whenever the number of neural network inputs, outputs and training patterns is significantly high as in this particular case, typically used MLP neural networks face a problem of selecting number of hidden layers and neurons in each layer, and selecting the appropriate learning rate. Training of these networks is computationally demanding and time consuming, and very easily converges to a local minimum, usually very far from the global minimum. Furthermore, retraining of this type of network with new training data is rather involved, and requires repetition of entire training procedure for updated input set.

Instead of using MLP neural network, a special type of self-organized neural network that belongs to the group of Adaptive Resonance Theory (ART) networks was proposed. ART neural network technique had been implemented for power system security assessment purposes [166]. Later, the same idea was utilized for transmission line fault analysis, including fault classification and characterization [7, 8], and post-mortem study of fault events [9].

Although used version of the ART algorithm offered many advantages, certain deficiencies have been observed. They include relatively small number of training and test patterns that correspond only to basic fault events and narrow choice of fault parameters, insufficient and not tuned preprocessing of input signals, somewhat ineffective supervised learning, simple interpretation of neural network output signals, and absence of self-learning capabilities after completed off-line training procedure.

D. Proposed Protection Approach

New approach relies on the same basic principle as the previous approach [7, 8, 9], where ART neural network is applied to the samples of measured signals. However, previous approach needs to be improved by specifying and implementing advanced concepts in algorithm off-line and on-line learning, input signal preprocessing and fuzzyfication of neural network outputs. Moreover, comprehensive algorithm training and evaluation has to be performed for realistic sets of various power system scenarios, and compared with classical relaying solution, and other alternative solutions based on neural network. The new approach can be summarized through the following steps:

- Develop a new neuro-fuzzy based pattern recognition approach applied directly to the samples of measurement signals to produce fault detection, classification and characterization in real-time;

- Implement training and test procedures of ART neural network algorithm using suitable software tool for numerical modeling and simulation;

- Select and implement suitable power network model and relevant set of the power system fault and disturbance events using appropriate software for simulating power system transients;

- Propose a new interactive simulation environment for interfacing neural network based relay model and complex power network scenarios, beforehand realized in different simulation programs;

- Perform algorithm training and testing based on power system simulation results and assigned classification tasks;

- Study algorithm sensitivity versus various neural network input signal preprocessing steps: selection of measurement quantities, sampling frequency, data window, and scaling ratio between voltage and current samples;

- Implement interpolation among the outputs of trained neural network by using pertinent fuzzy decision rule and optimizing the adjustable parameters;

- Improve the concept of supervised learning during training procedure to remove sporadically biased evolving of neural network structure;

- Implement neural network algorithm on-line learning through dynamical adaptation to different operating modes of power system;

- Develop and implement protection algorithms based on previous ART approach, widely used MLP neural network and impedance based distance relay to be used for comparison with the proposed ART solution.

E. Summary

Numerous applications of neural networks and fuzzy logic have been implemented so far for protective relaying of transmission lines. The applications cover all relaying functions, and use various types of neural networks and fuzzy logic methods as well. One specific approach, based on ART neural network and implemented for fault detection, classification and characterization, has been found to be particularly suitable to serve as a foundation of new approach. Newly defined approach introduces several advanced concepts to overcome deficiencies observed in the previous approach, and is expected to significantly improve performance of the most significant relaying functions.

CHAPTER IV

NEURAL NETWORK ALGORITHM

A. Introduction

This chapter presents ART neural network pattern recognition algorithm. First section discusses very important feature of proposed neural network - the inherent adaptivity of its structure. Moreover, an extensive description of a training of neural network by utilizing unsupervised and supervised learning stages is provided in second and third sections. The training steps are illustrated in fourth section using suitable *Training Demo* applied to a simplified real problem. Fifth section explains the implementation of trained neural network, and specifies the decision rule for interpreting neural network outputs.

B. The Adaptive Neural Network Structure

The proposed neural network does not have a typical predetermined structure with specified number of neurons, but rather an adaptive structure with self-evolving neurons. The structure depends only upon the characteristics and presentation order of the patterns in input data set. The diagram of complete procedure of neural network training is shown in Fig. 12.

The training consists of numerous iterations of alternating unsupervised and supervised learning stages, suitably combined to achieve maximum efficiency. Groups of similar patterns are allocated into clusters, defined as hyper-spheres in a multi-dimensional space, where the space dimension is determined by the length of input patterns. The neural network initially uses unsupervised learning with unlabelled input patterns to form fugitive clusters. It tries to discover pattern density by their

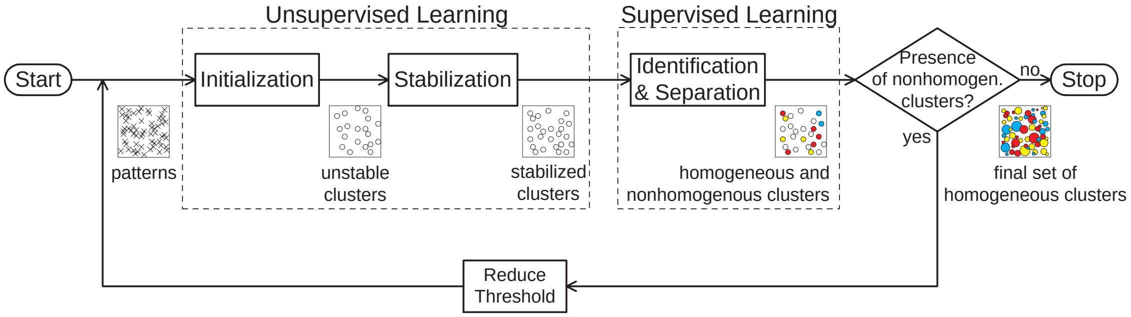


Fig. 12. Combined unsupervised and supervised neural network learning

groupings into clusters, and to estimate cluster prototypes that can serve as prototypes of typical input patterns. The category labels are then assigned to the clusters during the supervised learning stage. The tuning parameter called threshold parameter, controls the cluster's size and hence the number of generated clusters, and is being consecutively decreased during iterations. If threshold parameter is high, many different patterns can then be incorporated into one cluster, and this leads to a small number of coarse clusters. If threshold parameter is low, only very similar patterns activate the same cluster, and this leads to a large number of fine clusters.

After training, the cluster centers serve as neural network neurons and represent typical pattern prototypes. The structure of prototypes solely depends on the density of input patterns. Each training pattern has been allocated into a single cluster, while each cluster contains one or more similar input patterns. A prototype is centrally located in the respective cluster, and is either identical to one of the actual patterns or a synthesized prototype from encountered patterns. A category label that symbolizes a group of clusters with a common symbolic characteristic is assigned to each cluster, meaning that each cluster belongs to one of existing categories. The number of categories corresponds to the desired number of neural network outputs, determined by the given classification task. During implementation of trained network, distances between each new pattern and established prototypes are calculated, and using K-Nearest Neighbor classifier the most representative category amongst nearest prototypes is assigned to the pattern [58].

C. Unsupervised Learning

The initial data set, containing all the patterns, is firstly processed using unsupervised learning realized as a modified ISODATA clustering algorithm. During this stage patterns are presented without their category labels. The initial guess of the number of cluster and their positions is not specified in advance, but only a strong distance measure between cluster prototypes is specified using the threshold parameter. Unsupervised learning consists of two steps: initialization and stabilization phases.

The initialization phase, shown in Fig. 13, incrementally iterates all the patterns and establishes initial cluster structure based on similarity between the patterns. The entire pattern set is presented only once. Since the number of cluster prototypes is not specified, training starts by forming the first cluster with only first input pattern assigned. A cluster is formed by defining a hyper-sphere located at the cluster center, with radius equal to the actual value of threshold parameter. New clusters are formed incrementally whenever a new pattern, rather dissimilar to all previously presented patterns, appears. Otherwise, the pattern is allocated into cluster with the most similar patterns. The similarity is measured by calculating the Euclidean distance between a pattern and existing prototypes. Presentation of each input pattern updates a position of exactly one cluster prototype. Whenever an input pattern is presented, cluster prototypes that serve as neurons compete among themselves. Each pattern is compared to all existing prototypes and winning prototype with minimum

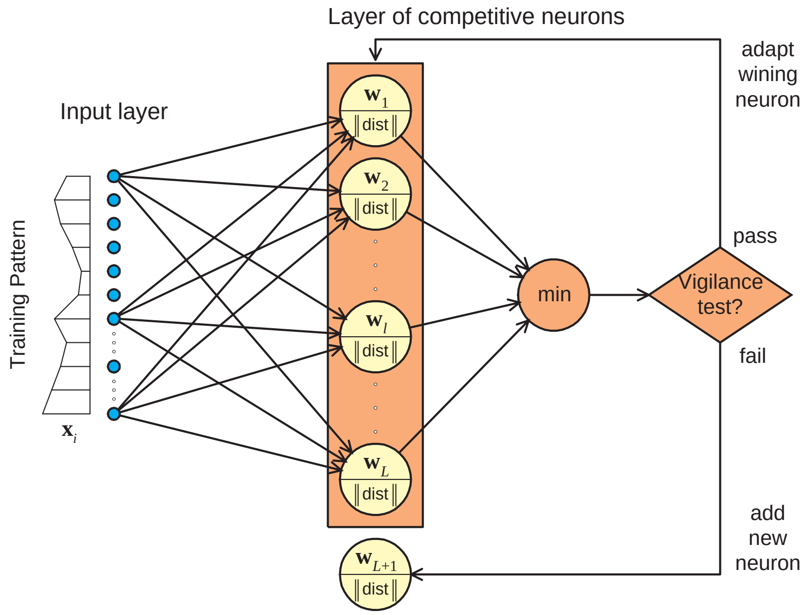


Fig. 13. Unsupervised learning - initialization phase

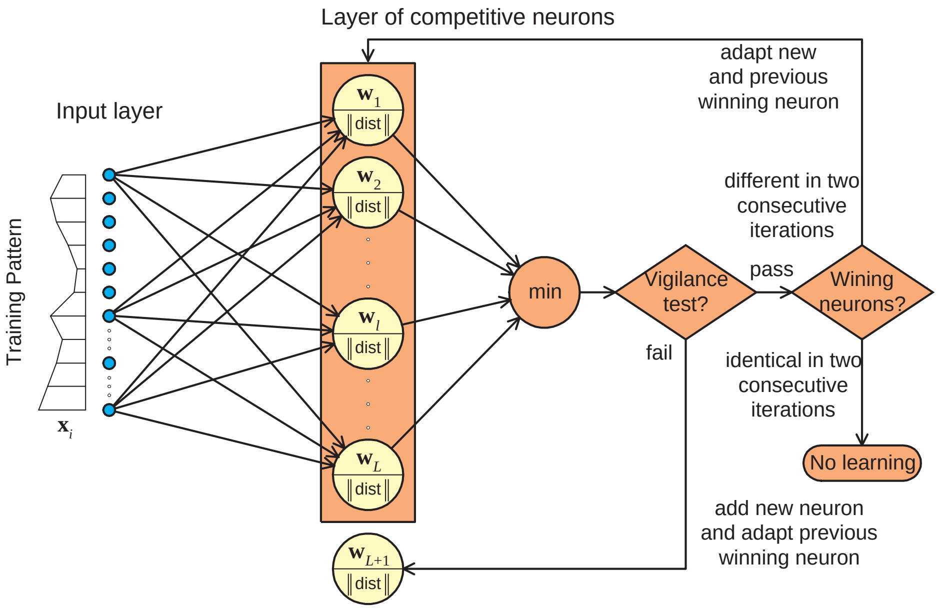


Fig. 14. Unsupervised learning - stabilization phase

distance to the pattern is selected. If winning prototype passes vigilance or similarity test, meaning that it does satisfactory match the pattern, the pattern is assigned to the cluster of winning prototype and cluster is updated by adding that pattern to the cluster. Otherwise, if winning prototype fails vigilance test, a new cluster with prototype identical to the pattern is being added. The entire procedure continues until all patterns are examined. Initialization phase does not reiterate, and although the clusters change their positions during incremental presentation of the patterns, already presented patterns are not able to change the clusters. Consequently, the final output of the initialization phase is a set of unstable clusters. Since initialization phase does not reiterate the patterns, stabilization phase is needed to refine the number and positions of the clusters.

Stabilization phase shown in Fig. 14 is being reiterated numerous times until a stable cluster structure is obtained, when none of the patterns exchange the clusters during single iteration. Stabilization phase starts with presenting all the patterns again. Each pattern is compared again to all existing prototypes, and winning prototype is selected. If for an actual pattern a winning prototype fails vigilance test, a new cluster is formed and the previous winning prototype is updated by erasing the pattern from that cluster. If for the pattern happens that the winning prototype is identical in two consecutive iterations and passes vigilance test, the learning does not occur since the pattern has not recently changed the cluster. Otherwise, if winning prototypes are different in the current and previous iteration, the pattern is moved to the current winning cluster and its prototype is updated by adding the pattern, while the previous winning prototype is updated by erasing the pattern from corresponding cluster. The stabilization phase is completed when all clusters retain all their patterns after single iteration.

Unsupervised learning produces a set of stable clusters, including homogenous

Table II. Unsupervised learning

Step	Action
0	J is the length of input pattern, I is the number of training patterns, L is the number of clusters or neurons, n_l is the number of patterns that belong to the cluster l. $\mathbf{x}_i = [x_{i1}, x_{i2}, \ldots, x_{iJ}]$ is i-th input pattern and x_{ij} is j-th feature of the i-th input pattern, and $\mathbf{w}_l = [w_{l1}, w_{l2}, \ldots, w_{lJ}]$ is a prototype of l-th cluster.
1	Set: $i = 1$, $L = 0$, $n_l = 0$.
2	Form new cluster: $\mathbf{w}_{L+1} = \mathbf{x}_i$, $n_{L+1} = 1$, $L = L + 1$.
3	Set: $i = i + 1$, If $i > I$ go to Step 7; If $i \leq I$ go to Step 4.
4	Compute the Euclidean distance d_l between pattern i and prototype of the cluster l: $d_l = \sqrt{(\mathbf{w}_l - \mathbf{x}_i)(\mathbf{w}_l - \mathbf{x}_i)^T} \quad \text{for } l = 1, 2, \ldots, L;$ Find winning prototype p for which $d_p = \min_l d_l$.
5	Compare minimum distance d_p to threshold parameter ρ: If $d_p > \rho$ go to Step 2; If $d_p \leq \rho$ go to Step 6.
6	Pattern i is assigned to the cluster p, and winning prototype is updated: $\mathbf{w}_p = \dfrac{n_p}{n_p + 1}\mathbf{w}_p + \dfrac{1}{n_p + 1}\mathbf{x}_i$, $n_p = n_p + 1$, go to Step 3.
7	Set $i = 0$, every pattern is presented again.
8	Set: $i = i + 1$, If $i > I$ go to Step 13; If $i \leq I$ go to Step 9.
9	Pattern i currently belongs to the cluster q. Compute the Euclidean distance d_l between pattern i and prototype of the cluster l: $d_l = \sqrt{(\mathbf{w}_l - \mathbf{x}_i)(\mathbf{w}_l - \mathbf{x}_i)^T} \quad \text{for } l = 1, 2, \ldots, L;$ Find winning prototype p for which $d_p = \min_l d_l$.
10	If $d_p > \rho$ go to Step 11 to form new cluster; If $p \neq q$ and $d_p \leq \rho$ go to Step 12 to change the cluster for pattern i; If $p = q$ and $d_p \leq \rho$ go to Step 8, because learning does not occur for pattern i.
11	Form new cluster $L + 1$, and update previous winning prototype q: $\mathbf{w}_{L+1} = \mathbf{x}_i$, $n_{L+1} = 1$, $L = L + 1$; $\mathbf{w}_q = \dfrac{n_q}{n_q - 1}\mathbf{w}_q - \dfrac{1}{n_q - 1}\mathbf{x}_i$, $n_q = n_q - 1$, go to Step 13.
12	Change cluster for pattern i and update new and previous winning prototypes p and q: $\mathbf{w}_p = \dfrac{n_p}{n_p + 1}\mathbf{w}_p + \dfrac{1}{n_p + 1}\mathbf{x}_i$, $n_p = n_p + 1$; $\mathbf{w}^q = \dfrac{n_q}{n_q - 1}\mathbf{w}_q - \dfrac{1}{n_q - 1}\mathbf{x}_i$, $n_q = n_q - 1$.
13	If in Steps 8-12 any pattern has changed its cluster membership go to Step 7, otherwise unsupervised learning is completed.

clusters, containing patterns of the identical category and non-homogenous clusters, containing patterns of two or more categories. It requires relatively high computation time due to a large number of iterations needed for convergence of the stabilization phase. The steps of unsupervised learning, performed through initialization and stabilization phases, are defined through mathematical expressions in Table II.

D. Supervised Learning

During supervised learning, shown in Fig. 15, the category label is associated with each input pattern allowing identification and separation of homogenous and non-homogenous clusters produced by unsupervised learning. Category labels are assigned to the homogeneous clusters, and they are being added to the memory of

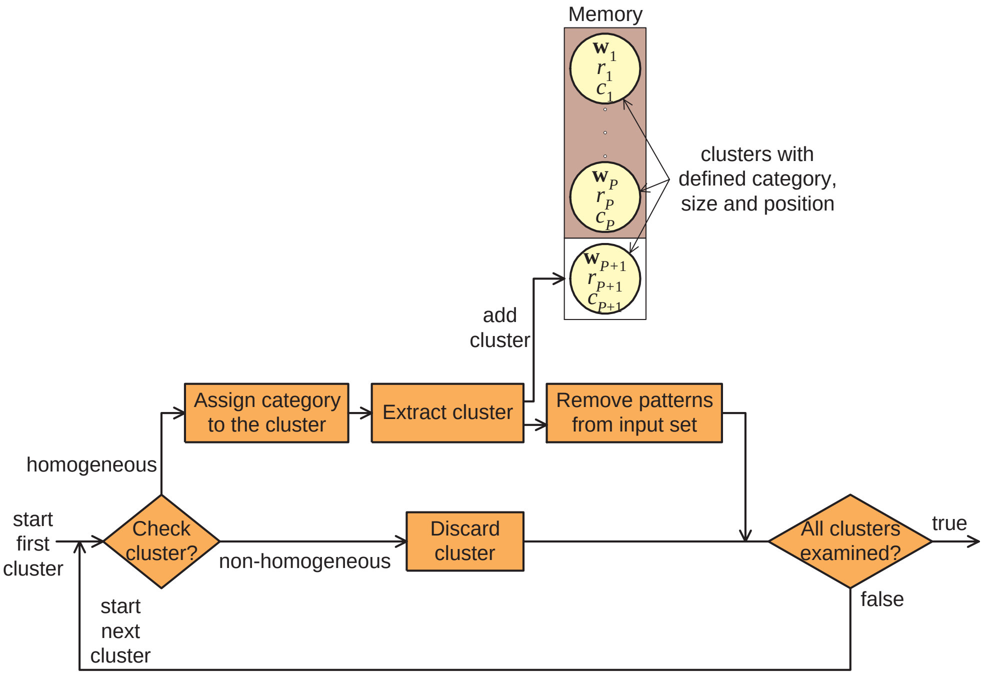

Memory
$\mathbf{w}_1$ r_1 c_1
$\mathbf{w}_P$ r_P c_P
$\mathbf{w}_{P+1}$ r_{P+1} c_{P+1}
clusters with defined category, size and position
add cluster
Assign category to the cluster
Extract cluster
Remove patterns from input set
homogeneous
start first cluster
Check cluster?
non-homogeneous
Discard cluster
All clusters examined?
true
false
start next cluster

Fig. 15. Supervised learning

stored clusters, including their characteristics: prototype position, size, and category. The patterns from homogeneous clusters are removed from further unsupervised-supervised learning iterations. Set of remaining patterns, present in non-homogenous clusters is transformed into new, reduced input data set and used in next iteration. The convergence of learning process is efficiently controlled by the threshold parameter, being slightly decreased after each iteration. The learning is completed when all the patterns are grouped into homogeneous clusters. Contrary to slow-converging unsupervised learning, supervised learning is relatively fast because it does not need to iterate.

One interesting phenomenon has been observed during supervised learning stage. Whenever allocated clusters with different categories mutually overlap, certain number of their patterns may fall in overlapping regions. In previous ART version, here named ART^1 [7, 8, 9, 14, 11, 10], although each of such patterns has been nominally assigned to the nearest cluster, their presence in clusters of other category leads to questionable validity of those clusters. One typical example of a small set of training patterns and obtained clusters is given in Fig. 16a, where many of the patterns are members of two clusters with different categories. The ambiguity can be solved by redefining supervised learning and introducing ART^2 with imposed restricted condition for identification of homogeneous clusters [12]. ART^2 revises supervised learning by requiring the homogeneous cluster to encompass patterns of exactly one category. Therefore, both subsets of patterns, one assigned to the cluster, and other encompassed by the cluster although assigned to any other cluster, are taken into account during supervised learning. Consequently, ART^2 allows overlapping between the clusters of different categories only if there are no patterns in overlapping regions. For the same set of training patterns, ART^2 produces finer graining of cluster structure, noticeable in Fig. 16b.

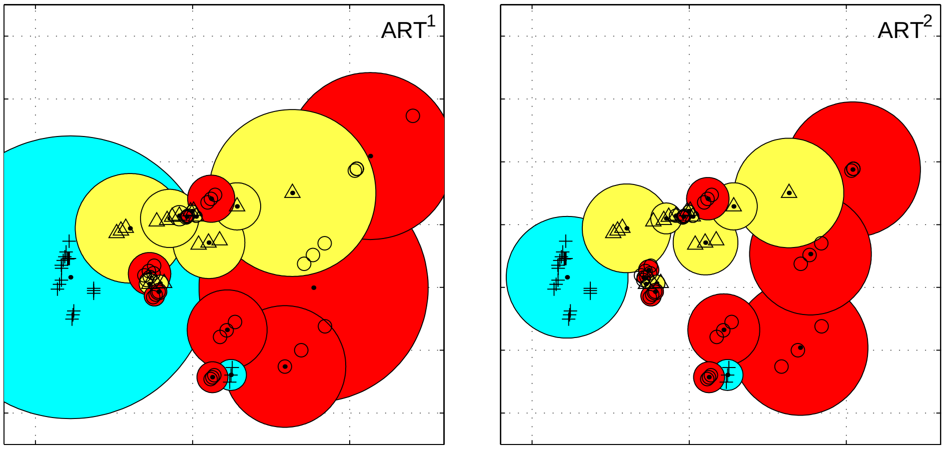


Fig. 16. Comparison of cluster structures generated by ART^1 and ART^2

E. Training Demo

The transformation of input patterns into clusters during unsupervised and supervised learning procedures will be demonstrated using convenient *Training Demo* applied on a simplified real problem, but with retention of all its main characteristics. Simplification is based on reduced number of training patterns, which belong to only three categories. The pattern length is selected to be only two allowing the pattern and cluster representation in a two dimensional graph. Training patterns, shown in Fig. 17, are generated by simulating three types of ground faults, combined with six values of fault distance, three values of fault impedance and fifteen values of fault inception angle. Total number of presented patterns is $3 \times 6 \times 3 \times 15 = 810$. Patterns are built by using one sample per phase from two phase currents, few sampling time steps after fault has occurred. The symbols used for showing the patterns correspond to different types of the fault.

The outcome of the *Training Demo* after single initialization phase is shown

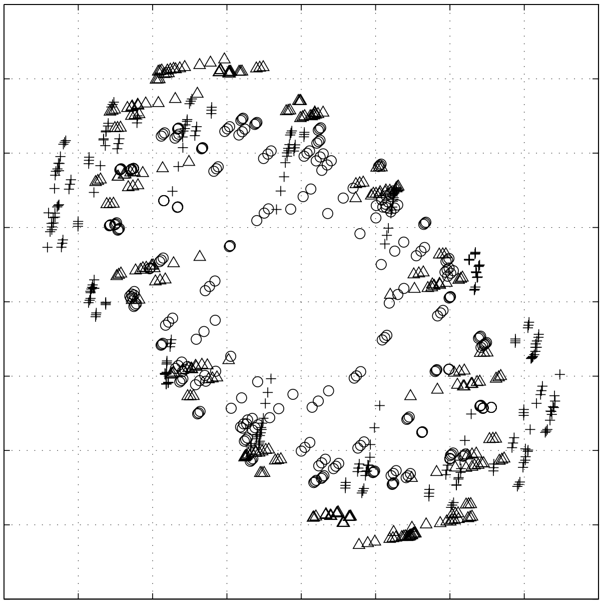


Fig. 17. Training Demo: set of categorized training patterns

in Fig. 18. Since pattern categories are not relevant during unsupervised learning stage, patterns are shown using identical symbols. Cluster's radii are determined by the current value of the threshold parameter in a particular iteration. Most of the patterns are grouped into clusters, although some patterns lie outside clusters. Cluster prototypes move during presentation of input patterns. In Fig. 18, the traces of moving cluster prototypes are also shown. The outcome of the *Training Demo* after single stabilization phase is shown in Fig. 19. The positions of existing clusters have been refined, and few new clusters have been formed. Now, all patterns are grouped into stable structure of clusters. The outcome of the *Training Demo* after single supervised learning stage applied on the structure of clusters obtained by the previous single unsupervised learning stage is shown in Fig. 20. Totally three clusters have been identified as homogeneous, and they belong to two different categories.

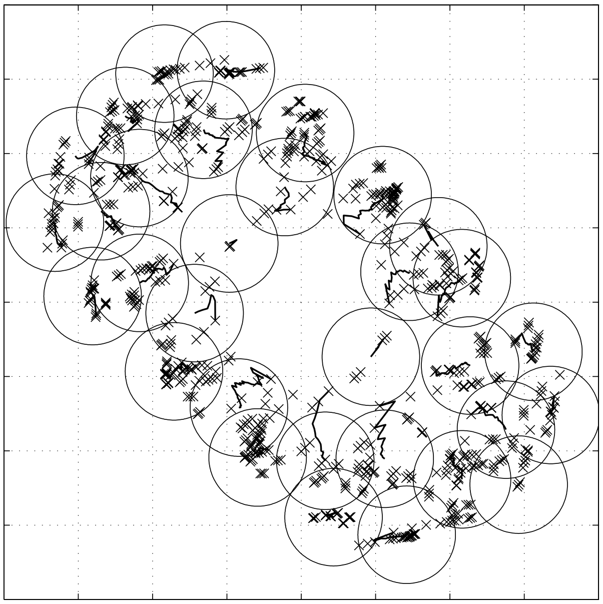


Fig. 18. Training Demo: unsupervised learning - outcome of single initialization phase

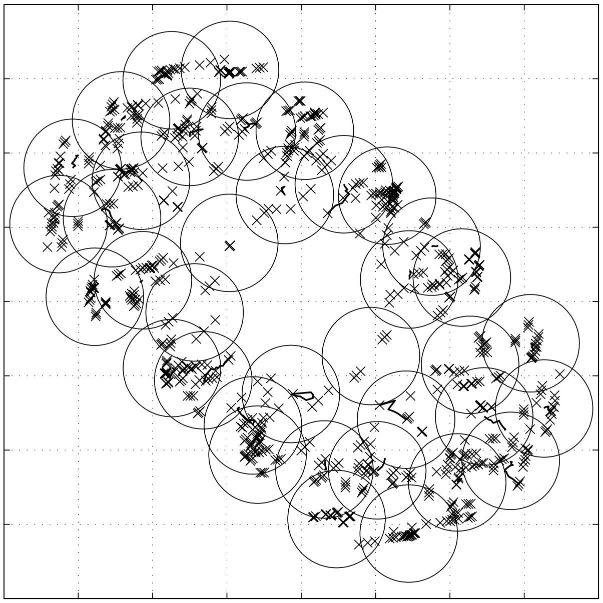


Fig. 19. Training Demo: unsupervised learning - outcome of single stabilization phase

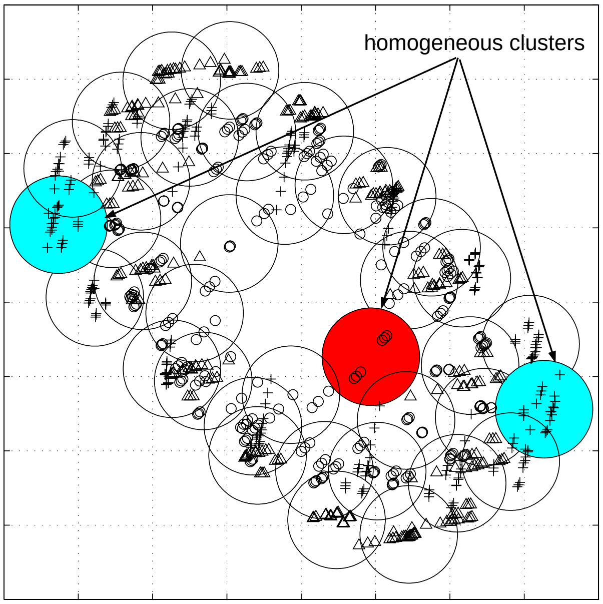


Fig. 20. Training Demo: outcome of single unsupervised-supervised learning stage

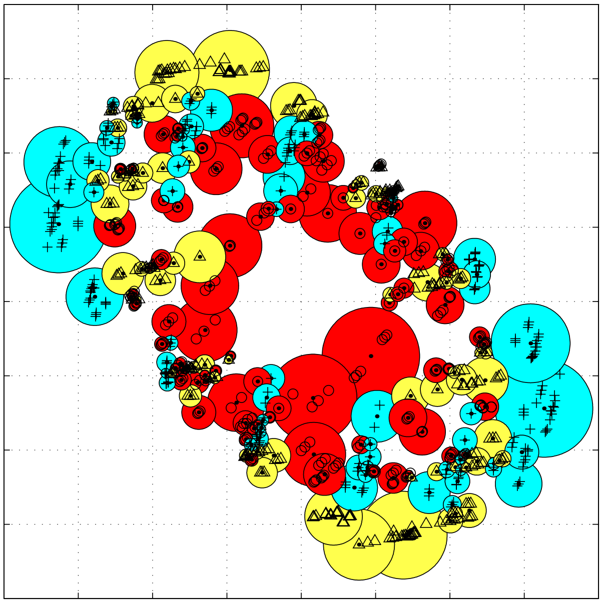


Fig. 21. Training Demo: final outcome of consecutive unsupervised-supervised learning stages

After numerous unsupervised-supervised learning iterations, the final structure of homogeneous clusters and their prototypes is generated and shown in Fig. 21. Established clusters have different size, location, and category. During neural network implementation the obtained cluster structure has to be employed as initial abstraction tool for the classification of new patterns.

The outcomes of the *Training Demo* with applied previous and new condition for supervised learning are shown in the Fig. 22. Introducing ART^2 an overlapping between the clusters is allowed only if there are no patterns in overlapping regions. Comparing to ART^1, ART^2 is expected to produce more clusters, slightly reduced in size, and with more consistent matching of the input space density.

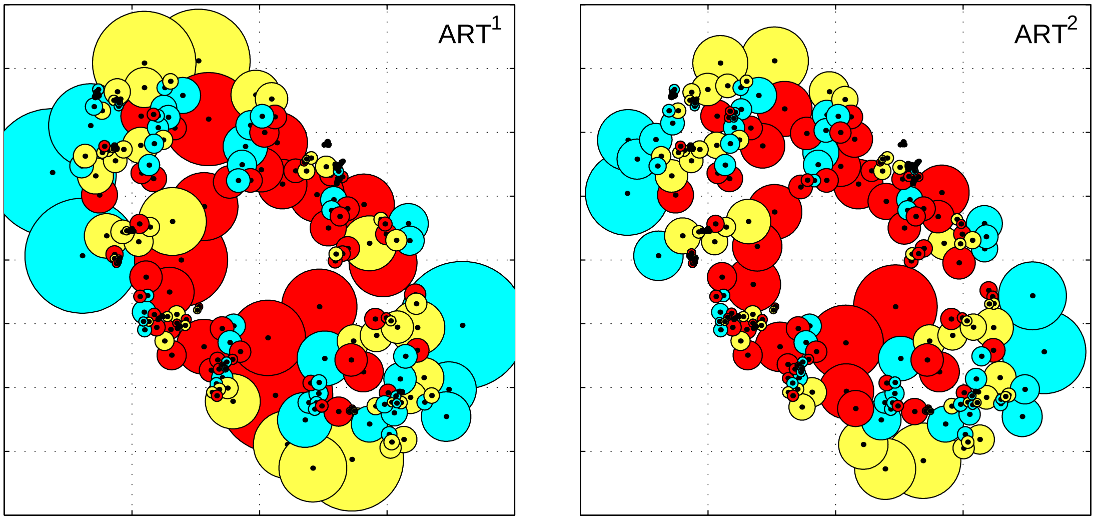


Fig. 22. Training Demo: comparison of cluster structures obtained by ART^1 and ART^2

F. Implementation

During implementation or testing of the trained neural network, new patterns are classified according to their similarity to the cluster prototypes generated during training. Classification is performed by interpreting the outputs of trained neural network through K-Nearest Neighbor (K-NN) classifier [58]. The K-NN classifier determines the category of a new pattern based on the majority of represented categories in prespecified small number of nearest clusters retrieved from the cluster structure established during training. It requires only the number K that determines how many neighbors have to be taken into account. K-NN classifier seems to be very straightforward and is reasonably employed since the number of prototypes is significantly smaller then the number of patterns in the training set.

Fig. 23 shows an implementation of the trained neural network with the simplest Nearest Neighbor classifier, which is identical to K-NN for $K = 1$. With such choice

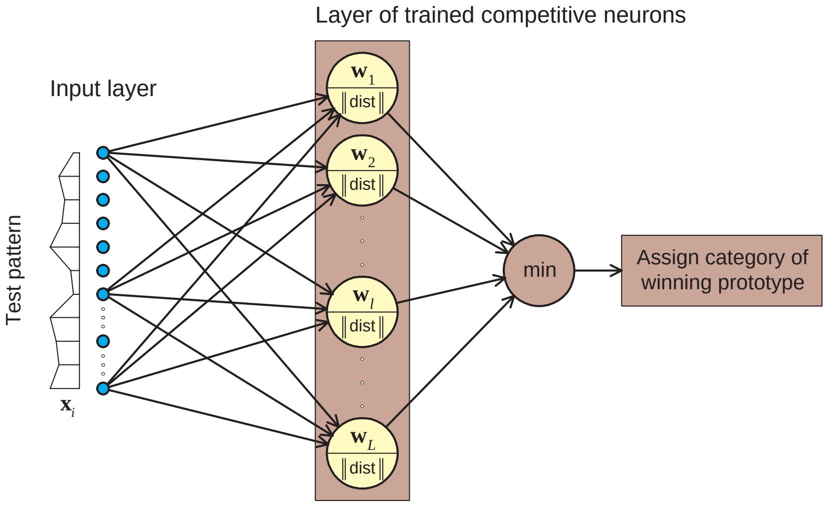


Fig. 23. Implementation of trained neural network with Nearest Neighbor classifier

of K, the category of nearest prototype is assigned to the pattern, and that prototype completely determines the pattern classification. Presented technique has been widely used so far for interpreting the outputs of ART neural network [7, 8, 9, 15, 14, 10].

G. Summary

Presented ART neural network based algorithm for pattern classification is characterized with self-organized structure and unique relationship between unsupervised and supervised learning strategies. Unsupervised learning uses uncategorized input patterns to form a set of fugitive clusters whose centers represent typical pattern prototypes. During supervised learning, categories of patterns within clusters are analyzed, homogeneous clusters are identified and added to the permanent set of prototypes. However, previous version of supervised learning has been reasonably modified to enhance generalization ability. During training, unsupervised and supervised learning phases alternate unless all patterns are classified into homogeneous clusters. Consequently, during implementation, new patterns are classified by interpreting the match between a pattern and established structure of prototypes using K-Nearest Neighbor classifier.

CHAPTER V

FUZZYFICATION OF THE NEURAL NETWORK

A. Introduction

This chapter introduces new method, called classifier, for interpreting neural network outputs using fuzzy logic. First section studies the deficiency of previously used non-fuzzy classifier, and necessity for applying fuzzy logic. Mathematical description of the non-fuzzy classifier is provided in the second section. Non-fuzzy classifier is gradually extended by inserting the effects of weighted distances and cluster size in the third and fourth sections, respectively. In the fifth section, both extensions are merged and a new fuzzy classifier is synthesized. Finally, in the sixth section the concept of fuzzy classifier is demonstrated and discussed using a *Fuzzyfication Demo*.

B. The Necessity for Applying Fuzzy Logic

The main advantage of K-NN classifier introduced in Chapter IV is its computational simplicity, but substantial disadvantage is that each of the K neighboring clusters is considered equally important in determining the category of the pattern being classified, regardless of their size and distances to the pattern. Using such classifier, smooth and reliable boundaries between the categories cannot be established. For instance, an unambiguous situation exists whenever a new pattern is very close to only one of the prototypes, and intuitively has to be classified to the category of that prototype. However, in reality the events are quite diverse. The corresponding patterns might appear in unlabeled space between the clusters or in their overlapping regions and be more or less similar to several prototypes located nearby and possibly labeled with different categories. One such example is shown in Fig. 24, where a small

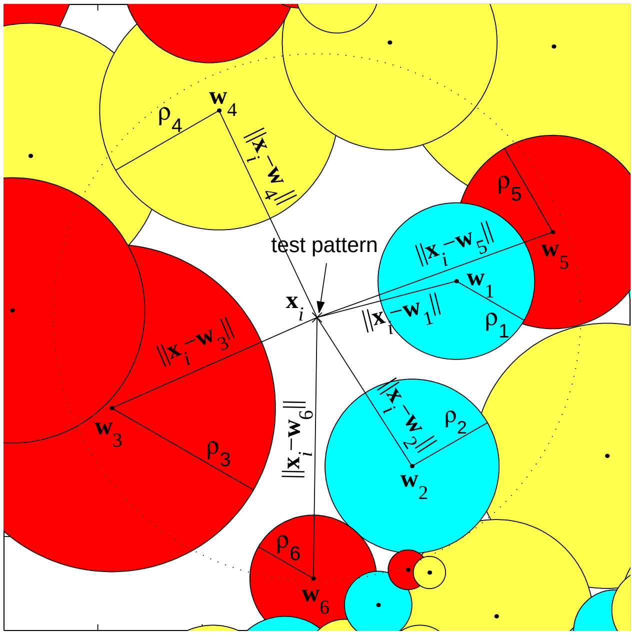

$\mathbf{w}_4$
ρ_4
$\|\mathbf{x}_i - \mathbf{w}_4\|$
ρ_5
test pattern
$\|\mathbf{x}_i - \mathbf{w}_5\|$
$\mathbf{w}_5$
$\mathbf{w}_1$
$\mathbf{x}_i$
$\|\mathbf{x}_i - \mathbf{w}_1\|$
ρ_1
$\|\mathbf{x}_i - \mathbf{w}_3\|$
ρ_2
$\|\mathbf{x}_i - \mathbf{w}_2\|$
$\mathbf{w}_3$
ρ_3
$\|\mathbf{x}_i - \mathbf{w}_6\|$
$\mathbf{w}_2$
ρ_6
$\mathbf{w}_6$

Fig. 24. Test pattern and the nearest clusters with different category label, radius, and distance to the pattern

portion of the cluster structure shown in Fig. 21 is enlarged.

Obviously, K-Nearest Neighbor classifier used so far needs to be improved to achieve better generalization for the patterns that correspond to a new set of events, previously unseen during training, and conceivably dissimilar to any of existing prototypes. The classification of a new pattern may be redefined, not only to be the simple function of categories of the nearest clusters, but complex function composed of their category, size, and distance to the pattern [11, 12]. A new approach can be proposed by introducing the theory of fuzzy sets into the classifier concept to develop its fuzzy version [66], shown in Fig. 25. New classifier is supposed to provide more realistic classification of new patterns.

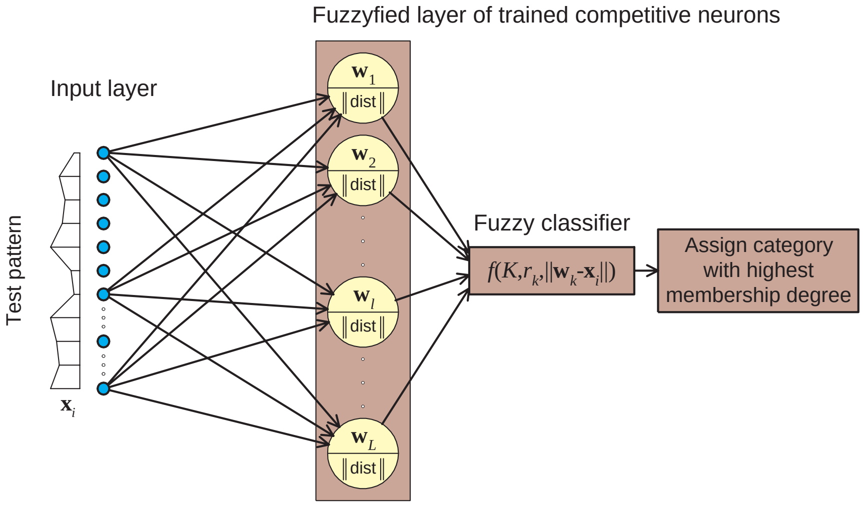


Fig. 25. Implementation of a trained neural network with Fuzzy K-Nearest Neighbor classifier

C. Crisp K-Nearest Neighbor Classifier

Given a set of categorized clusters, the crisp or non-fuzzy K-NN classifier determines the category of a new pattern $\mathbf{x}_i$ based only on the categories of the K nearest clusters

$$\mu_c(\mathbf{x}_i) = \frac{1}{K}\sum_{k=1}^{K}\mu_c(\mathbf{w}_k) \tag{5.1}$$

where $\mathbf{w}_k$ is a prototype of cluster k, $\mu_c(\mathbf{w}_k)$ is membership degree of cluster k belonging to category c, $\mu_c(\mathbf{x}_i)$ is membership degree of pattern $\mathbf{x}_i$ belonging to category c. $i = 1, \ldots, I$; $c = 1, \ldots, C$; $k = 1, \ldots, K$; where I, K, and C are the numbers of patterns, nearest neighbors, and categories, respectively. Given classifier allows $\mu_c(\mathbf{w}_k)$ to have only crisp values 0 or 1, depending on whether or not a cluster k belongs to category c

$$\mu_c(\mathbf{w}_k) = \begin{cases} 1 & \text{if cluster } k \text{ belongs to category } c \\ 0 & \text{otherwise} \end{cases} \tag{5.2}$$

If two or more of K nearest clusters have the same category, then they add membership degrees to the cumulative membership value of that category. Finally, when the contributions of all neighbors are encountered, the most representative category is assigned to the pattern

$$g(\mathbf{x}_i) = \max_c \mu_c(\mathbf{x}_i) \tag{5.3}$$

where $g(\mathbf{x}_i)$ is the category assigned to pattern $\mathbf{x}_i$, and $c = 1, \ldots, C$. Thus the outputs of a given neural network reflect different categories of input events.

D. The Effect of Weighted Distances

Initial enhancement of K-NN classifier is introduction of weighted contribution of each of the K neighbors according to their distance to a pattern, giving greater weight to the closer neighbors. The distance $d_k(\mathbf{x}_i)$ is generally selected to be the weighted Euclidean distance between pattern $\mathbf{x}_i$ and prototype $\mathbf{w}_k$

$$d_k(\mathbf{x}_i) = \|\mathbf{x}_i - \mathbf{w}_k\|^{\frac{2}{m-1}} \tag{5.4}$$

where the parameter $m \geq 1$ is fuzzyfication variable and determines how heavily the distance is weighted when calculating each neighbors' contribution to the pattern category membership. For choice of $m = 2$, calculated distance is identical to Euclidean distance. Moreover, as m increases toward infinity, the term $\|\mathbf{x}_i - \mathbf{w}_k\|^{\frac{2}{m-1}}$ approaches one regardless of the distance, and the neighbors are more evenly weighted. However, as m decreases toward one, the closer neighbors are weighted more heavily than those further away. If $m \approx 1$ the algorithm behaves like crisp K-NN classifier for $K = 1$.

E. The Effect of Cluster Size

Next improvement of K-NN classifier is to insert fuzzy membership degree as a measure of the cluster belonging to its own category. Apparently, there is no meaningful reason why membership value $\mu_c(\mathbf{w}_k)$ must retain only a crisp value. This dissertation proposes an original extension of crisp K-NN by considering size of generated clusters in an original way. Since each cluster belongs exactly to one of the categories, membership value $\mu_c(\mathbf{w}_k)$ may be redefined to reflect the relative size of the actual cluster k

$$\mu_c(\mathbf{w}_k) = \begin{cases} \rho_k & \text{if cluster } k \text{ belongs to category } c \\ 0 & \text{otherwise} \end{cases} \tag{5.5}$$

where $\mu_c(\mathbf{w}_k)$ is the membership degree of cluster k belonging to category c, and is selected to be proportional to the radius of cluster k. The outcome is that the larger neighboring clusters would contribute more than the smaller ones.

F. Fuzzy K-Nearest Neighbor Classifier

The extensions proposed in (5.4) and (5.5) can now be used to define Fuzzy K-NN classifier that generalizes crisp K-NN classifier given in (5.1). A new pattern has to be classified based on the categories of K nearest clusters, their relative size and weighted distances to the pattern. The Fuzzy K-NN classifier calculates, using superposition, a set of membership values $\mu_c(\mathbf{x}_i)$ of input pattern $\mathbf{x}_i$ belonging to all categories present in the K nearest clusters based on the following formula

$$\mu_c(\mathbf{x}_i) = \frac{\displaystyle\sum_{k=1}^{K} \mu_c(\mathbf{w}_k) \frac{1}{\|\mathbf{x}_i - \mathbf{w}_k\|^{\frac{2}{m-1}}}}{\displaystyle\sum_{k=1}^{K} \rho_k \frac{1}{\|\mathbf{x}_i - \mathbf{w}_k\|^{\frac{2}{m-1}}}} \tag{5.6}$$

where $\mu_c(\mathbf{w}_k)$ is given in (5.5). Finally, the pattern is assigned to the category with the highest membership degree according to (5.3). The denominator in (5.6) uses ρ_k for all K neighbors to normalize total membership value which must be equal to one. Adaptive behavior of the classifier can be attained by playing with parameters K and m and selecting the best pair of values for a particular implementation. For $K = 1$, both fuzzy and non-fuzzy classifiers become identical.

G. Fuzzyfication Demo

Step-by-step evolution from a non-fuzzy to fuzzy classifier is presented in Figs. 26-28 for the set of clusters obtained by the *Training Demo* in Chapter IV and shown in Fig. 21. In given figures, the effects of weighted distances and clusters' size are gradually introduced into fuzzy membership degree. The clusters are shown in their original two-dimensional space, while their impact modeled through membership function is represented in a third dimension.

First of all, Fig. 26 shows clusters' impact without included effects of cluster size and distance to the points in the pattern space. In this case the parameters of fuzzy classifier are $m = 1$ and $\mu_c(\mathbf{w}_k) = 1$, and such choice apparently corresponds to a non-fuzzy classifier. The effect of weighted distances is introduced in Fig. 27, but cluster size is not yet considered. In this situation fuzzy classifier has parameters $m > 1$ and $\mu_c(\mathbf{w}_k) = 1$ and acts as weighted non-fuzzy classifier. Merged effects of clusters' size and distances to the points in a pattern space are shown in Fig. 28. The figure shows the essence of intrinsic fuzzy classifier, since the values of the parameters are $m > 1$ and $\mu_c(\mathbf{w}_k) = \rho_k$. The decision regions between the categories are defined by superposition of clusters' membership functions using expressions (5.6) and (5.3). The decision regions for a choice of parameter values $K = 4$ and $m = 1.50$ are

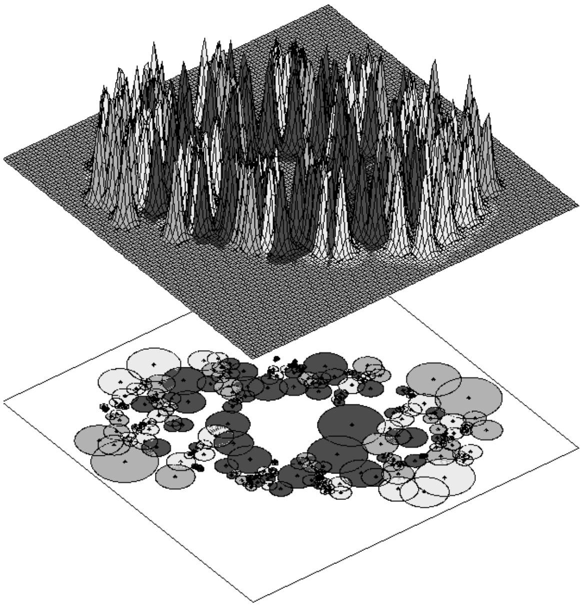


Fig. 26. Fuzzyfication Demo: K-Nearest Neighbor classifier with equal contribution of all prototypes

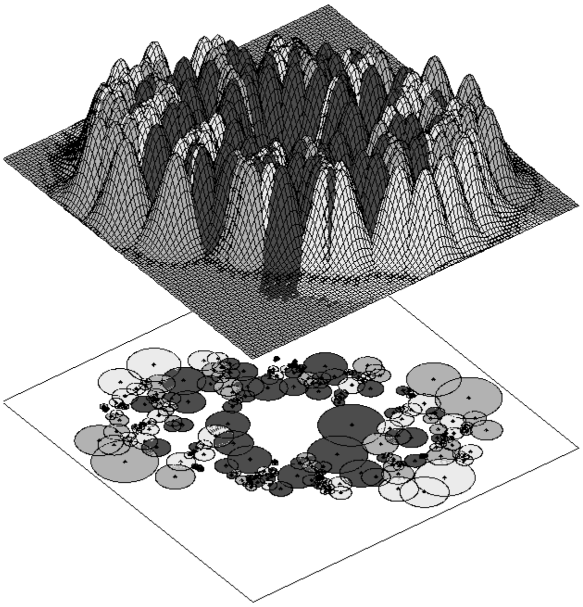


Fig. 27. Fuzzyfication Demo: extended K-Nearest Neighbor classifier with included contribution of weighted distance from the prototypes

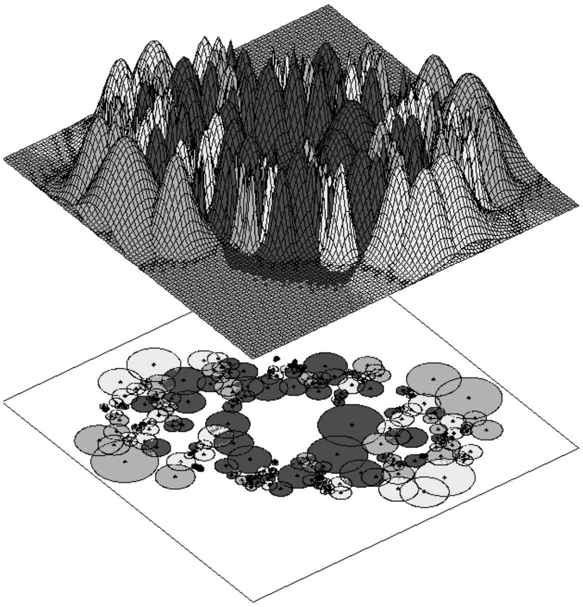


Fig. 28. Fuzzyfication Demo: Fuzzy K-Nearest Neighbor classifier with included contribution of cluster size and weighted distance from the prototypes

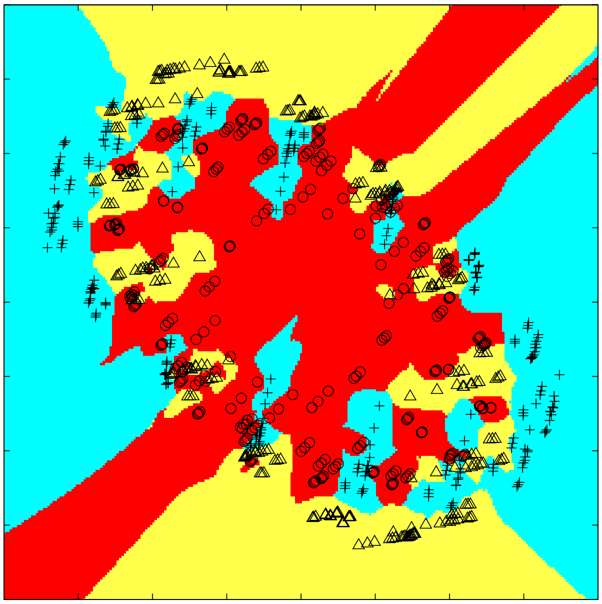


Fig. 29. Fuzzyfication Demo: Category decision regions established by Fuzzy K-Nearest Neighbor classifier

presented in Fig. 29. As expected, cluster's impact is substantial in their proximity, having bigger clusters heavily involved in creating decision boundaries, while smaller clusters react only in a narrow area nearby.

H. Summary

A new fuzzy method for interpretation of neural network outputs that substitutes the classical method has been developed. It represents an advanced concept that classifies new patterns according to weighted contribution of few similar pattern prototypes established during training. The fuzzy classifier takes into account the distance among a pattern and selected number of nearest prototypes, as well as the size of the clusters related to those prototypes. Fuzzy classifier with properly tuned parameters may result in very precise category-decision boundaries and will improve generalization capabilities accordingly.

CHAPTER VI

PROPOSED HARDWARE AND SOFTWARE SOLUTION

A. Introduction

Complete hardware and software solution of a new artificial intelligence based digital relaying concept is proposed in this chapter. First section shows and explains an architecture of the new relaying solution. Second section describes data acquisition and signal preprocessing steps. The procedures of neuro-fuzzy protective relaying algorithm design and implementation are summarized in the third section. This section also illustrates the abstraction mechanism developed for solving the pattern recognition problem. The execution of the relay output command is addressed in the fourth section.

B. Relay Architecture

A functional block diagram of proposed hardware and software solution for protective relaying based on combined neuro-fuzzy approach is shown in Fig. 30. Given relaying principle, as in any other digital relay generally comprises three fundamental hardware subsystems: a signal conditioning subsystem, a digital processing subsystem and command execution subsystem.

The main part of the relay is software realization of protective algorithm. The algorithm is a set of numerical operations used to process input signals to identify and classify the faults, and subsequently initiate an action necessary to isolate the faulted section of the power system. The protective relay must operate in the presence of a fault that is within its zone of protection, and must restrain from operating in the absence of a fault, or in case of faults outside its protective zone.

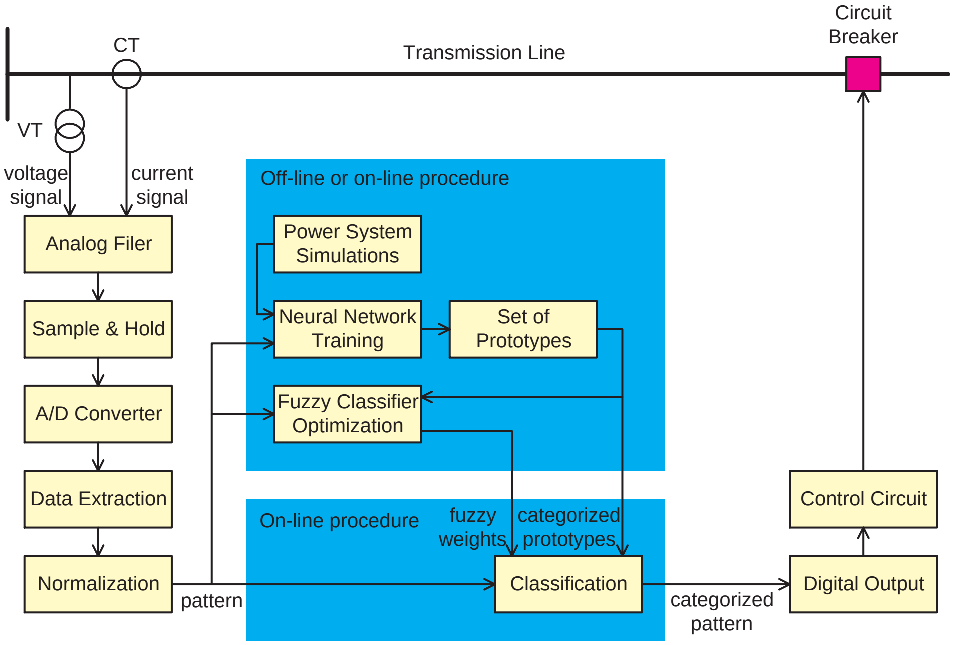


Fig. 30. Proposed solution for protective relaying based on neural network and fuzzy logic

C. Data Acquisition and Signal Preprocessing

The training of neural network based protective algorithm can be performed either on-line using measurement data directly taken from the field, or off-line by accessing historical record of fault and disturbance cases. Since interesting cases do not happen frequently, initial off-line training becomes inevitable, and sufficient training data are provided using simulations of relevant power system scenario cases. Various fault and disturbance events and operating states need to be simulated, by changing power network topology and parameters.

Transmission line current and voltage signal levels at relay location are usually very high (kV and kA ranges). They are measured with current and voltage transformers (CT and VT) and reduced into lower operating range typical for A/D converters. The process of pattern extraction or forming neural network input signals from obtained measurements depends on several signal preprocessing steps [10, 11]. Attenuated, continuously varying, three-phase current and voltage sinusoidal signals are filtered by low-pass analog antialiasing filter to remove noise and higher frequency components. According to the sampling theorem requirement, the ideal filter cut-off frequency has to be less or equal one-half of the sampling rate used by the A/D converter. Furthermore, filtered analog signals are sampled by A/D converter with specified sampling frequency, and converted to its digital representation.

Selection of sampled data for further processing generally includes both the three phase currents and voltages, but also may include either three phase currents or three phase voltages. The samples are extracted in a dynamic data window with desired

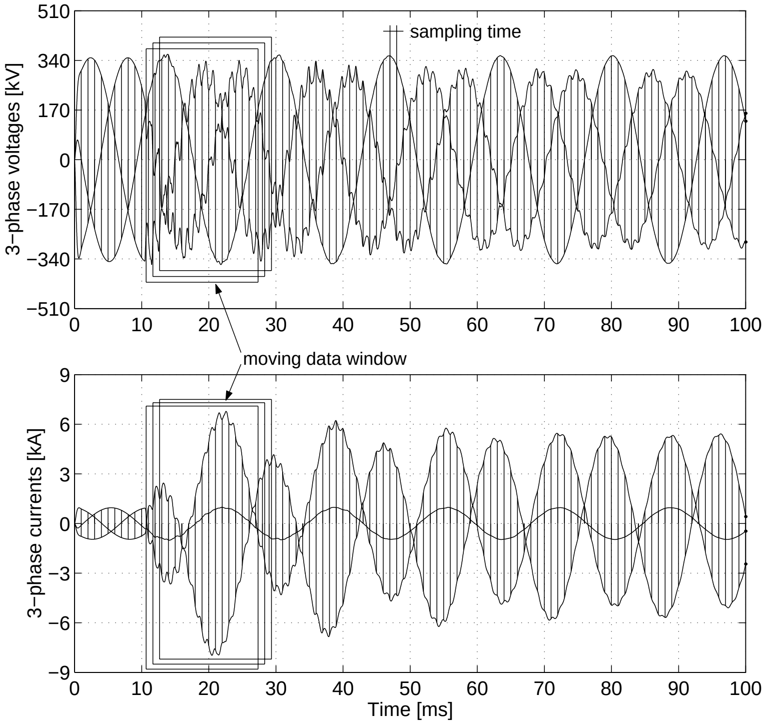


Fig. 31. Moving data window for voltage and current samples

length (Fig. 31), normalized, and aligned together to form a common input vector of pattern components. Since a pattern is composed of voltage and current samples that originate from two different quantities with different levels and sensitivities of signals, independent scaling of two subsets of samples present in a pattern may improve algorithm generalization capabilities. Using this value, the current contribution to the relay decision may be increased over the voltage contribution or vice versa.

Specified conditioning of input signals determines the length and characteristics of input patterns and influences the trade-off between performing the fault classification more accurately and making the real time decision sooner. The values of preprocessing parameters adversely affect the algorithm behavior during training as well as performance during implementation, and should be optimized in each particular situation whenever either relay location, classification tasks or expected scenarios are different.

D. Neuro-Fuzzy Protective Algorithm

The main function assigned to the relay is detection of the faults in the designated area of relay responsibility, and classification of fault type. Supplemental function is recognition of the zone of relay protection where fault has occurred, and is called fault characterization.

ART neural network, explained in Chapter IV, is applied to the patterns extracted from voltage and current measurements. Neural network training is complex incremental procedure of adding new and updating existing pattern prototypes. The outcome of training process is a structure of labeled prototypes where each belongs to one of the categories. A category represents combination of a fault type and a section of relay protection where fault has occurred.

During real-time (on-line) implementation, trained neural network possesses generalization ability and is expected to successfully classify new patterns that have not been presented during training process. The prototypes established during training are dynamically compared with new patterns extracted from the actual measurements. A pattern similarity to all the prototypes is calculated, and a subset of the most resembling prototypes is retrieved. Finally, a pattern is being classified by interpreting relationship between the pattern and chosen prototypes using fuzzy decision rule defined in Chapter V. If the pattern is classified to the category of unfaulted or normal state, then the input data window is shifted for one sample, pattern vector is updated and the comparison is performed again, until a fault is detected and classified.

Illustration of complete steps of the input space mapping into category deci-

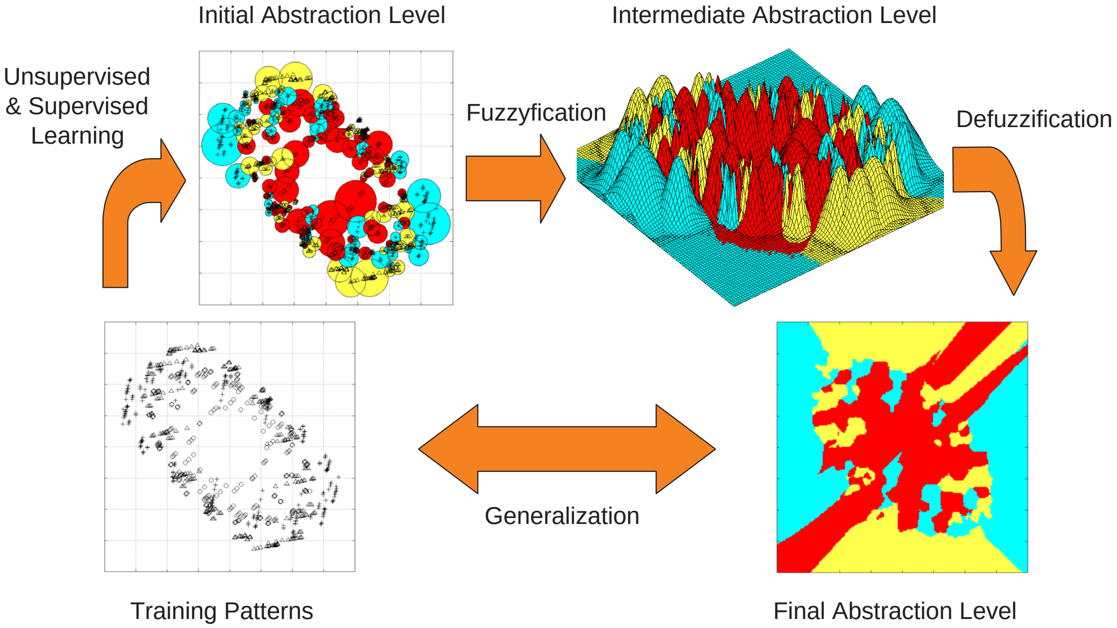


Fig. 32. Mapping of the pattern space into category decision regions using unsupervised and supervised learning, fuzzyfication and defuzzyfication

sion regions using proposed neural network - fuzzy logic based approach is shown in Fig. 32. Using unsupervised/supervised learning, the space of training patterns is transferred into initial abstraction level, containing set of clusters with corresponding prototypes, size and categories. Moreover, the clusters are fuzzyfied and transformed into intermediate abstraction level. Final abstraction level is attained when, using defuzification, category decision regions with smooth boundaries are established.

E. Relay Command Execution

A pattern categorized by protection algorithm is converted into digital output and by control circuit transformed into a corresponding signal for circuit breaker activation. During normal operation, the circuit breaker is closed and currents are flowing through the transmission line. Depending on the recognized event and selected designer's logic, circuit breaker either trips faulted phases within proposed time period and removes fault from the rest of the system, or stays unaffected.

Whenever a possibility of temporary faults exists, additional reclosing relays are usually used to automatically reclose the circuit breaker after the fault has been isolated for a short time interval, and restore the normal status of the power system.

F. Summary

The new neuro-fuzzy relaying solution that relies on concept of pattern recognition has been developed. Proposed solution can be used for reliable distinction among faulted and unfaulted events, and selective fault type and fault section estimation. Neural network training data are acquired either using actual and recorded relay measurements, or through power system simulations. The signals are collected, filtered, sampled and extracted in dynamic data window. The patterns obtained possess inherited characteristics of power system events, and constitute a basis for developing complex abstraction mechanism for classification. The protective approach is based on ART neural network that establishes the prototypes of typical input patterns, and afterwards employs the prototypes for classification of new inputs using fuzzy decision rule. The output of the relaying algorithm is a command for circuit breaker activation.

CHAPTER VII

POWER SYSTEM MODELING AND SCENARIO CASES

A. Introduction

This chapter introduces a power network model and related power system scenario cases used for the design and evaluation of proposed protective relaying algorithm. A detailed and accurate model of the real power network which is very suitable to imitate realistic power system operation is outlined in the first section. The most suitable sets of fault, disturbance, and non-fault scenarios relevant for studying relay operation, are determined in the second, third and fourth sections, respectively.

B. Power Network Model

The specially developed power network model for relay testing and simulation studies represents an actual 345kV power system section, called STP-SKY. The data is obtained from Center Point Energy in Houston [13] and is enclosed in Appendix. The network model is unique since it has the configuration that allows one to simulate many normal as well as fault conditions that are rather difficult for the traditional relaying to deal with. The electrically remote parts of the system were modeled using Thevenin equivalents, while other components in the studied section were modeled in detail. The reduced network equivalent was obtained by using calculations based on load flow and short circuit data. It has nine buses and contains eleven short and long transmission lines, represented by lumped and distributed parameter models. Lines were modeled as frequency independent and mutual coupling between parallel lines was neglected. The model was verified using both the steady state and transient state results by using recordings captured during actual fault events in the system.

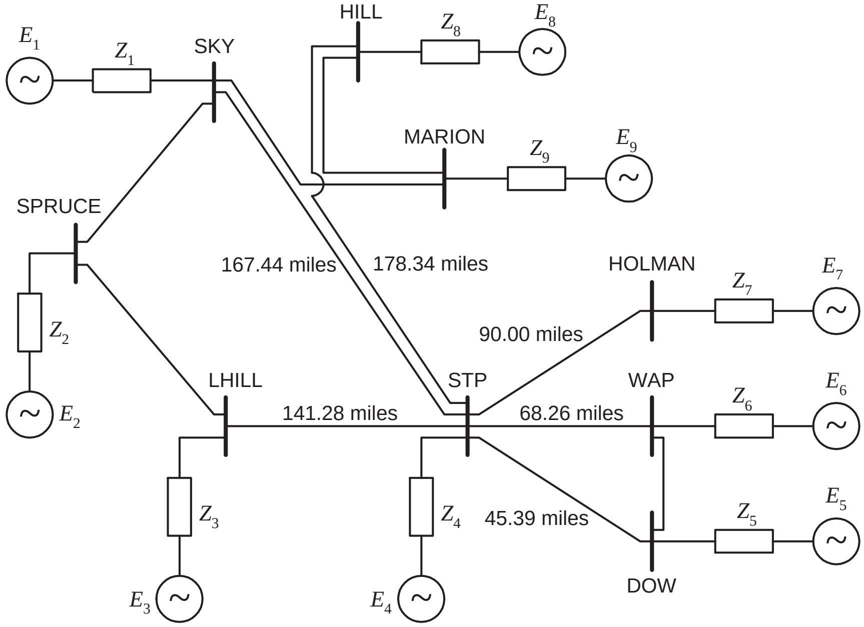


Fig. 33. Center Point Energy STP-SKY power network model

The one-line diagram of developed reduced section model is shown in Fig. 33. Among other transmission lines, STP-SKY line has been recognized as a line of particular interest, and transient signals obtained at one line end, bus SKY, will be utilized for implementing and evaluating relay design.

C. Fault Scenario Cases

The faults on the transmission line have the most significant impact on relay setting and evaluation. Generally, the simulation of the transmission line faults depends on four main fault parameters whose varying values outline the set of typical transient waveforms captured at the relay location:

- type of the fault: single-phase-to-ground, phase-to-phase, two-phase-to-ground, three-phase, and three-phase-to-ground;

- distance between relay and faulted point: $0 \div 100\%$ of the line length;
- impedance between faulted point and ground: $> 0\ \Omega$;
- fault inception angle (time) relative to the period of sinusoidal signal: $0 \div 360°$ ($0 \div 16.67$ ms).

Besides faults on the transmission line, there are many other interesting scenarios including disturbances and non-fault events that may perturb the relay measurements [13, 15].

D. Disturbance Scenario Cases

The diversity of disturbances that may affect relay operation includes various switching events and changes in power system operating conditions. This dissertation identifies several factors which change power system conditions and may cause severe impact on the relay performance. Previously defined fault scenarios originally simulated for nominal power system conditions can be repeated during existence of selected off-nominal power system conditions. A set of elaborated disturbance conditions includes:

- weak infeed caused by some lines or sources being out of service;
- source and load variations from their nominal values;
- system frequency variation from its nominal value.

Weak infeed is present in the network whenever an imbalance between equivalent impedance of the sources in the network exists, and can be created by disconnecting some of the equivalent sources. One typical situation of the weak infeed present at the bus SKY is introduced when the sources E1 and E9 are disconnected.

Source and load variations can occur due to: various changes in the power system network configuration, voltage control, changes in power consumption, fault at the remote part of the network, equipment outages, etc. Since the sources and loads in the network model are represented as equivalent network sources, their changes are considered as single events. The source E1 is selected to be the varying source, because changes of its phase angle have been observed to have the most significant impact on the relay performance, both the impedance and neural network based.

System frequency variation is caused by certain transient events in the network, or by imbalance between the generated active power and load demand. System frequency variations affect all equivalent sources in the network.

E. Non-Fault Scenario Cases

Additional, non-fault scenarios convenient for studying relay operation in the given power network are:

- line closing with remote line end open and grounded;
- line closing with permanent fault present on the line and the remote line end open.

Line closing onto grounded line is introduced by switching SKY end of the SKY-STP line while STP end is open and grounded. Line closing onto permanent fault is introduced by switching SKY end of the SKY-STP line while the permanent fault is present and STP line end is open.

F. Summary

The carefully developed model of a real power network has been presented in this chapter. One of the transmission lines has been recognized to be of particular interest, and measurements from one line end will be used to perform relay design and evaluation. Moreover, the set of applicable scenarios of special interest for protection of selected line were chosen and described. The scenarios encompass sets of appropriate fault and non-fault events, during nominal and disturbed operating conditions. The simulation outputs have to be used for training, tuning and testing the proposed protective relaying algorithm.

CHAPTER VIII

SIMULATION ENVIRONMENT

A. Introduction

This chapter proposes a novel interactive simulation environment that contains software modules for developing and evaluating complex protection algorithms. First section provides a summary of available software tools for power system modeling and simulation, and relaying algorithm design and testing. Second section describes software modules for independent power network and protective relay implementation, and required interfacing. Scenario setup and automatic generation of scenario cases are explained in third section.

B. Software Tools for Power System Simulation and Protective Algorithm Design

Modeling and simulation of complex power networks require availability of diverse software tools. The use of Electromagnetic Transient Program (EMTP) [179] and Alternative Transients Program (ATP) [180] for simulating power system and producing voltage and current transient waveforms has been known for a long time. More recently, general purpose modeling and simulation tool MATLAB with its toolbox SimPowerSystems has been used for the power network modeling [181, 182].

The relay models for protective relaying simulation may be implemented in either EMTP/ATP or in standard programming languages. Existing modeling approaches in EMTP/ATP do not facilitate interfacing sophisticated relay models to the power network models. Proposed simulation method requires relay models to be embedded into EMTP/ATP by using either TACS or MODELS facilities [180, 183]. It is still rather difficult to implement sophisticated and detailed relay models in these programs, due to the lacks of necessary mathematical tools and suitable graphical user interfaces. Moreover, automatic simulation of a large number of scenarios is practically impossible and is a limiting factor as well. Since simulation outputs have to be used as a signal generator for the protective algorithm training and evaluation, the model interfacing for large-scale simulations is critical for study of the new relaying approach. The relays can be more accurately and more efficiently modeled by using either C language or a commercial software package like MATLAB with pre-defined libraries of specialized functions [184, 185, 186, 187, 188]. In recent years, few new approaches have been proposed to connect the electromagnetic transient programs with MATLAB [189, 190, 191].

C. Model Interfacing

The simulation environment based on MATLAB software package is selected as the main engineering tool for performing modeling and simulation of power systems and relays, as well as for interfacing the user and appropriate simulation programs [189, 17, 15, 18, 19]. ATP is used for detailed modeling of power network and simulation of interesting events. It possesses excellent power networks modeling capabilities, exceptional libraries of elements and provides fast and accurate simulation results. Scenario setting and neural network relaying algorithm are implemented in MATLAB and interfaced with the power network model implemented in ATP. MATLAB has been chosen due to availability of the powerful set of programming tools, signal processing and numerical functions, and convenient user-friendly interface.

In this specially developed simulation environment, design and evaluation procedures can be easily controlled and performed. This is done through interfacing programs for power system simulation and relay modeling, and exchanging their sim-

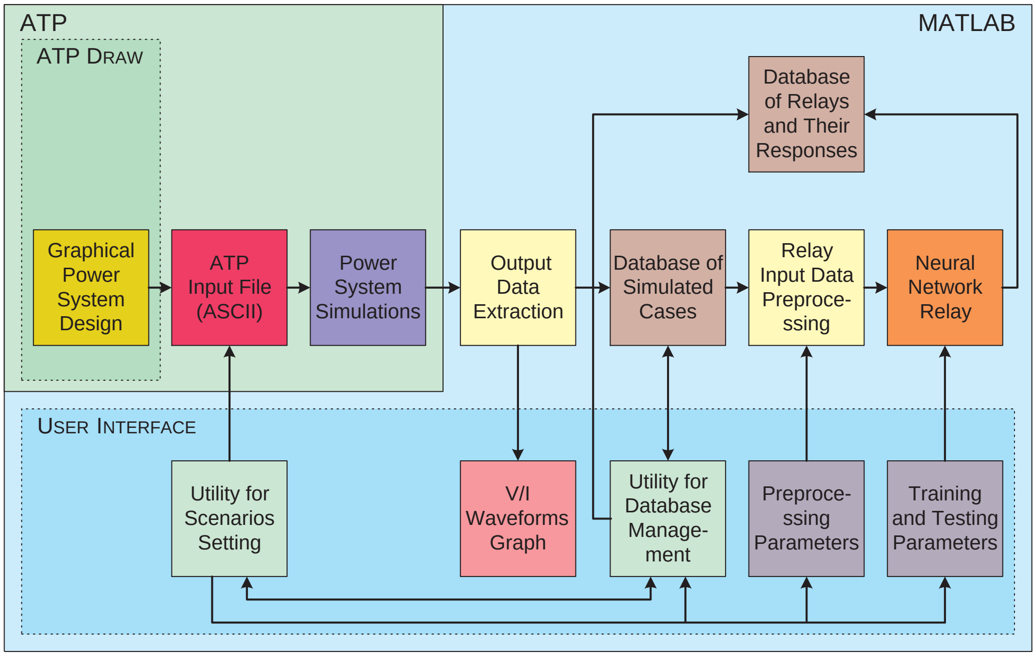

ATP
ATP Draw
MATLAB
Database of Relays and Their Responses
Graphical Power System Design
ATP Input File (ASCII)
Power System Simulations
Output Data Extraction
Database of Simulated Cases
Relay Input Data Preprocessing
Neural Network Relay
User Interface
Utility for Scenarios Setting
V/I Waveforms Graph
Utility for Database Management
Preprocessing Parameters
Training and Testing Parameters

Fig. 34. Integrated software tools for power system simulation by using MATLAB and ATP

ulation parameters and results (Fig. 34). MATLAB user interface provides user access to create the scenarios and control the simulation settings. Specially developed utility in MATLAB enables running automatically simulations for a large number of scenario cases. The user is allowed to specify all desired modifications in the network topology and parameters, as well as their combinations to create a set of scenarios. The ATP simulations are automatically initiated and controlled for each case of proposed scenario set, and simulation outputs are interactively converted into appropriate format and used as a signal generator for the neural network algorithm training and evaluation, also implemented in MATLAB.

User interface facilitates scenario formulation by selecting combinations of events and setting their parameters. It also manages database of simulated cases, relay models and relay responses. It is used for relay input data preprocessing, and to set up relay design parameters.

D. Generating Simulation Cases

Through a MATLAB graphical user interface the user can select sequences of individual and combined changes in the network topologies and corresponding parameters described in Table III. Control of transmission line fault simulations is based on changing the line section lengths to model varying fault distance, changing opening and closing time instances of phase-to-phase and phase-to-ground switches to create a specific fault type with desired inception time, and changing fault impedance. Line switching is controlled through circuit breaker opening and closing times, corresponding to various fault inception times. Source/load changes are established through

Table III. Accessible parameters and corresponding actions for updating power network model

Parameter	Action
Fault type	Adjusting opening/closing times of the phase-to-phase and phase-to-ground switches covering all fault types and various fault angles inside one cycle
Fault inception time	
Fault location	Adjusting line section lengths from both line ends to the faulted point and covering the faults along entire line
Fault impedance	Adjusting fault impedance
Line out of service	Holding line breakers permanently open
Line switching	Adjusting opening/closing time instances for the circuit breakers and covering various switching angles inside one cycle
Source/load variation	Adjusting voltage amplitude, phase angle and impedance
System frequency variation	Adjusting system frequency and frequency of equivalent sources
Relay location	Activating specific measurement points in the network
Integration time step	Adjusting general simulation parameters
Simulation end time	

varying voltage source amplitude, phase angle and impedance. Changes of power system frequency affect frequency dependent elements of the system, including voltage sources and impedances of frequency dependent lines. In addition to mentioned power network parameters, few other general simulation parameters can be controlled as well. They encompass integration time step and simulation end-time, and provide capability to select desired time frame, speed and accuracy of the simulation.

Systematized procedure allows user to specify a set of hundreds or thousands of scenario cases within few simple steps. For each individual case, MATLAB program automatically updates related data fields in the ATP's power network description file, and initiates ATP simulation for each case. Therefore, required changes in the power network model, typical for the event that has to be simulated, are updated between consecutive simulations. After execution of each simulation case, results are extracted and converted into a data format understandable by MATLAB, and recorded for further processing.

E. Summary

Evaluation of any protective relaying solution requires modeling of the power network and protective relays, and simulating the interactions for a variety of the power system events. The new MATLAB based software environment brings exceptional flexibility in the design of new relays by allowing protective relay modeling to be implemented independently from the power network modeling. Electromagnetic transient program ATP is used for modeling and simulating the power network model. That model is interfaced to MATLAB software package for the setup and automatic simulation of a large number of scenarios. Simulation outputs are used as a signal generator for a protective algorithm design and testing, implemented in MATLAB.

CHAPTER IX

ALGORITHM IMPLEMENTATION

A. Introduction

This chapter focuses on the implementation steps for protective algorithm design. Simulation of specified fault and no-fault events during nominal or off-nominal power system operating conditions is explained in first section. Second section describes utilization of obtained simulation outputs for generation of training and test patterns. Three different relaying algorithms are designed for fault detection, classification, and characterization (section estimation). ART neural network algorithm training and tuning is presented in third section. For comparison with ART algorithm, MLP neural network algorithm and impedance based distance relay are designed in fourth and fifth section, respectively.

B. Simulation of Scenario Cases

The scenarios for algorithm off-line training are generated during nominal power system operating conditions by specifying several values for each of the four fault parameters and combining these values to cover diversity of fault events. The example how the fault parameters affect relay measurements is shown in Figs. 35-38, for different fault type, distance, impedance and inception angle, respectively. Parameters used for generation of the training cases are:

- fault type: all eleven types of fault and no-fault state;
- fault distance: from 5 up to 95% of the line length, in increments of 10%;
- fault resistances for ground faults: 3, 11, and 19 Ω;

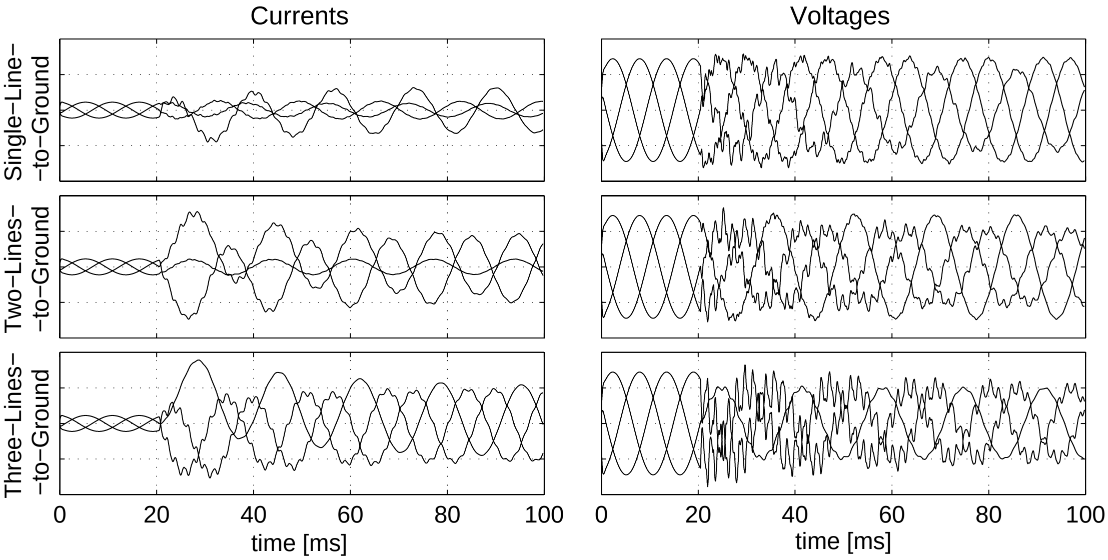


Fig. 35. Example of current and voltage waveforms for three types of fault

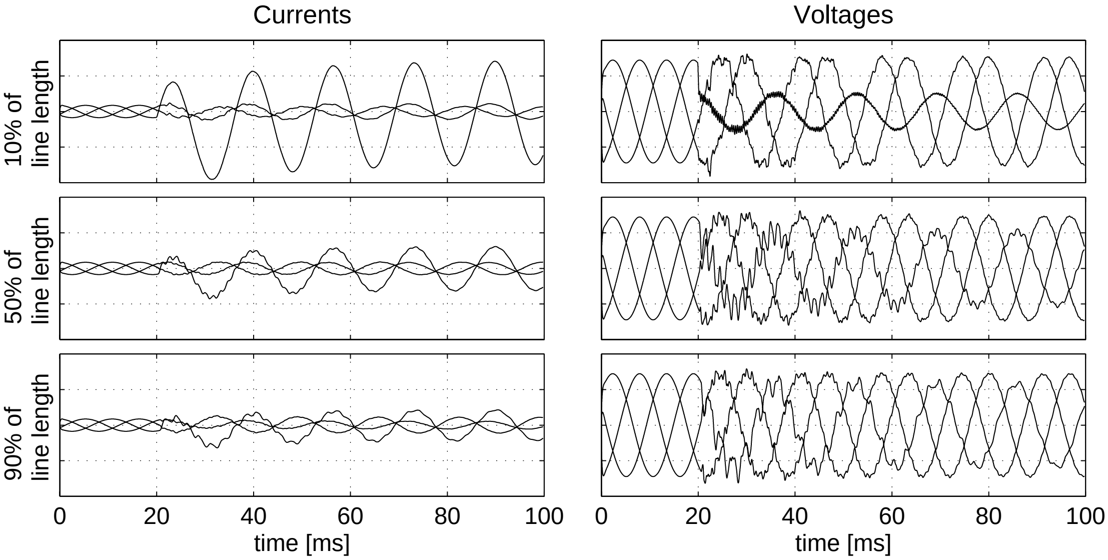


Fig. 36. Example of current and voltage waveforms for single-phase-to-ground fault at three fault locations

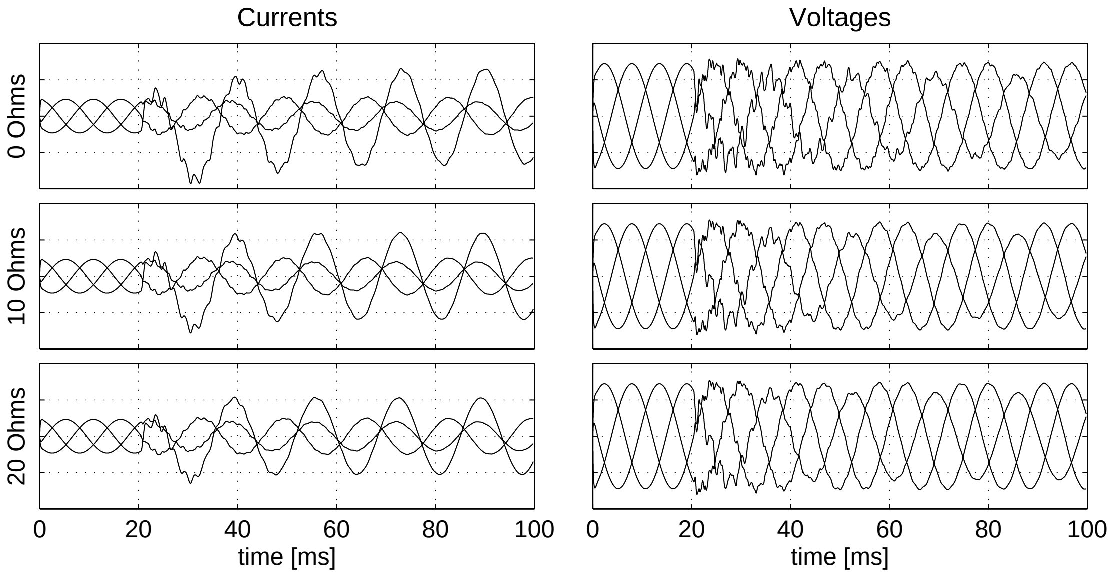


Fig. 37. Example of current and voltage waveforms for single-phase-to-ground fault with three values of fault resistance

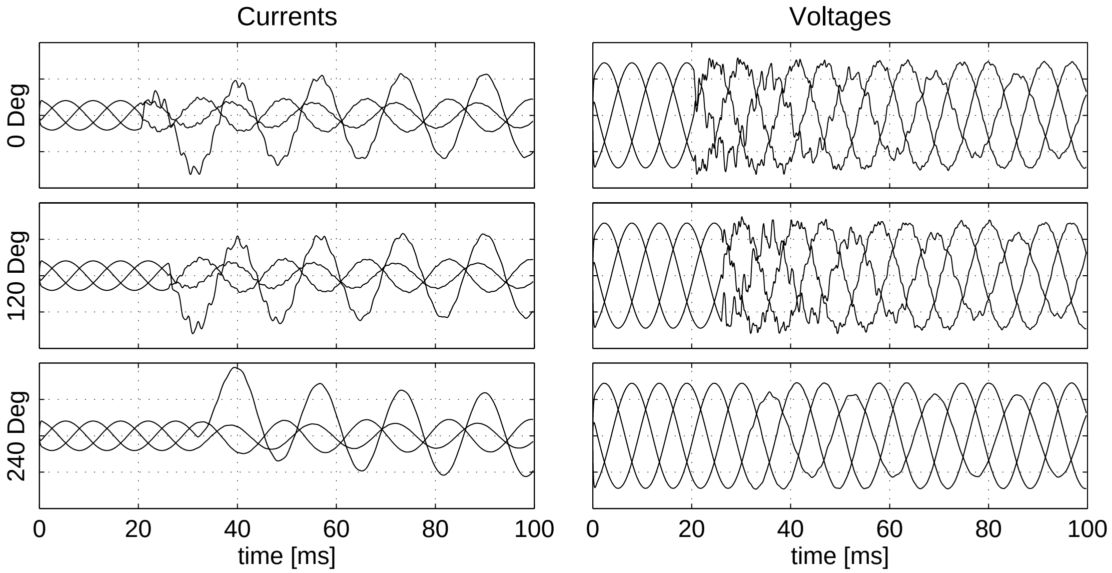


Fig. 38. Example of current and voltage waveforms for single-phase-to-ground fault for three values of fault incident angle

- fault angle: from 0 up to 336°, in increments of 24°.

Total number of patterns in the training set, obtained by combining selected values of all parameters, is 3315. For algorithm on-line learning, a total of 13260 patterns is obtained by repeating the faults for nominal system, and three off-nominal system conditions: weak infeed, source/load variations and system frequency variation.

Test scenarios for algorithm evaluation in heuristic, previously unseen situations are selected to be diverse and statistically independent from the training scenarios. Fault parameters used for generation of test scenarios are randomly selected using uniform distribution of: fault type, fault distance between 0 and 100% of the line length, fault angle between 0 and 360°; and normal distribution of fault resistance, with mean 0 Ω and standard deviation 10 Ω, taking into account only positive values. Total of six sets of test cases are generated, where first four sets are implemented for the faults on the transmission line of interest for the following network conditions:

- nominal system - 5000 patterns;
- weak infeed in the network, due to disconnected sources E1 and E9 - 5000 patterns;
- off-nominal voltage source E1 with phase angle shift of $-30°$ - 5000 patterns;
- off-nominal system frequency of 59 Hz - 5000 patterns.

Two other sets of test cases are implemented for primary line switching onto faulted line and line switching onto grounded line:

- closing the line with remote end open and grounded - 100 patterns;
- closing the line with present persistent fault on the line and with remote end open - 1000 patterns.

There is also another set of scenarios called validation set, used to tune the parameters of fuzzy classifier, and as early stopping condition during training of MLP neural network. Although validation set is generated using faults on primary line and nominal network conditions, it is statistically independent from the foremost set of test scenarios. Total number of patterns in the validation set is 4000.

C. Generation of Training and Test Patterns

Current and voltage measurements acquired through simulations are used for forming the patterns for neural network algorithm training and testing. Selected quantities are either three phase currents, or three phase voltages, or both three phase currents and voltages. Antialiasing filter is a second order low-pass Butterworth filter with the cut-off frequency equal to a half of selected sampling frequency. Sampling frequencies of 1.92 kHz (32 samples/cycle) and 3.84 kHz (64 samples/cycle), and dynamic data windows of 8.33 ms (half cycle) and 16.67 ms (one cycle) are implemented. Combining the values for sampling frequency and data window, four cases of pattern length with 96, 192, 192 and 384 components are obtained and illustrated in Fig. 39. Afterwards, all patterns in a given training set are normalized by scaling all their features to have zero mean, and to be in the range $[-1, 1]$. An identical scaling ratio is used later for normalization of patterns in a test set. Whenever patterns with both current and voltage samples are involved, heuristic rescaling of only current portion of the patterns is performed several times for each individual training case until the best scaling ratio offering lowest predictive error is discovered and subsequently adopted. The effect of the use of various scaling ratios for scaling current portion of the pattern is shown in Fig. 40.

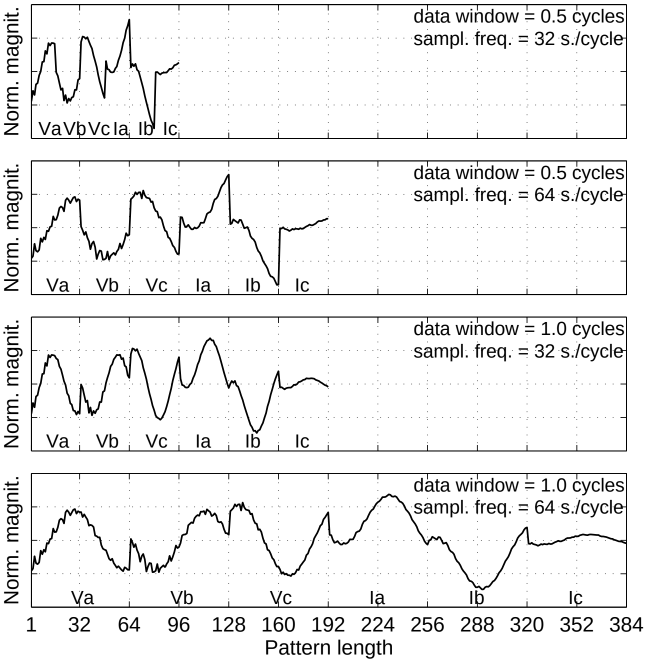


Fig. 39. Example of patterns for different values of data window and sampling frequency

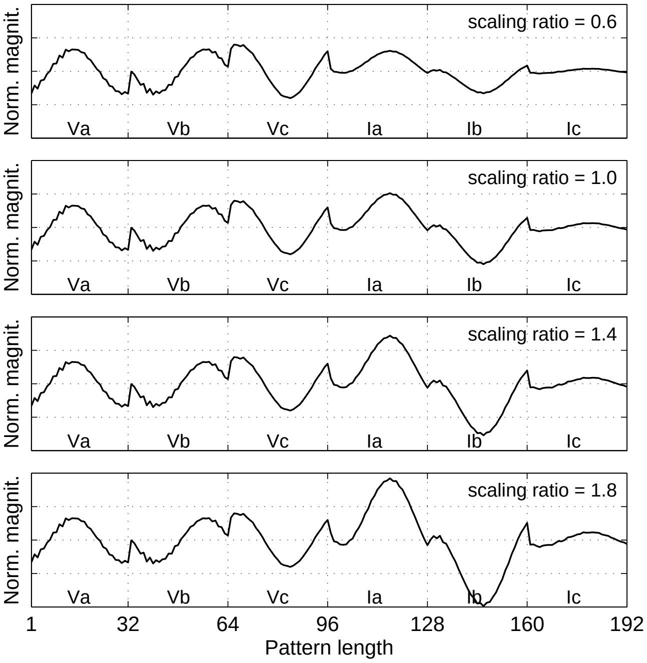


Fig. 40. Example of the patterns for different current/voltage scaling ratios

D. Design of Adaptive Resonance Theory Neural Network

For each input set, two types of classifications are implemented, and the total number of output categories depends on the assigned classification task. For the first classification task, fault type detection, there are eight categories: seven for fault types and one for no-fault situation, where phase-to-phase and phase-to-phase-to-ground fault with identical phases involved are treated as a unique fault type. For the other more challenging classification task, simultaneous fault type and fault section detection, there are fifteen categories, comprising seven fault types in each of two sections and no-fault situation. Boundary distance between protection sections is set to be 80% of the line length.

During algorithm training, both preceding and recent concept for identifying

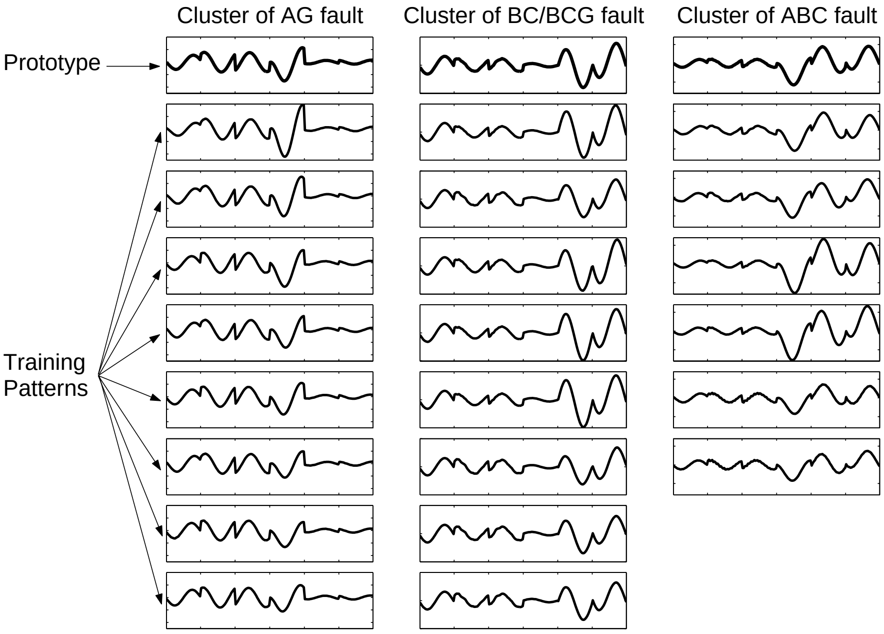


Fig. 41. Examples of cluster prototypes and encountered training patterns

homogeneous clusters are implemented to allow comparison of related classification results. The decreasing rate of the threshold parameter is selected to be 0.95. An example of a time-domain representation of three typical clusters with allocated training patterns and constructed prototypes is shown in Fig. 41. It demonstrates similarity among all the patterns encountered by each cluster and corresponding synthesized prototype.

Non-fuzzy and Fuzzy K-NN classifiers are implemented in parallel for all proposed test cases. Non-fuzzy K-NN is always settled with the most reasonable value $K = 1$, widely used so far. However, tuning of Fuzzy K-NN classifier needs proper selection of used number of neighbors K and the weight m for each particular combination

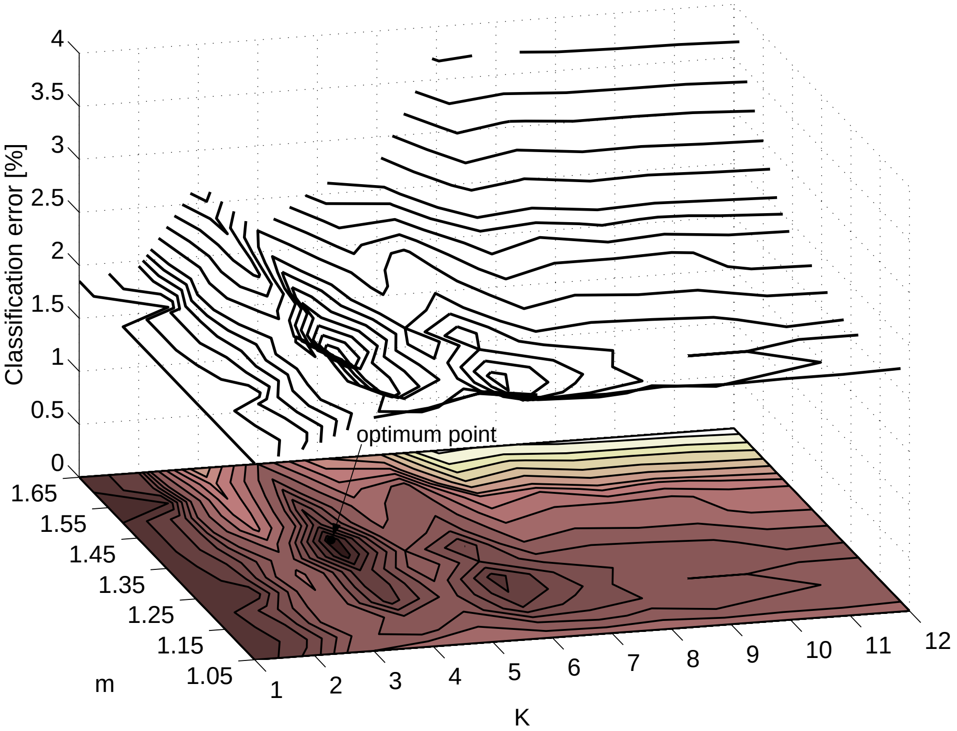


Fig. 42. Fuzzy classifier parameter optimization through minimization of predictive classification error

of the training set and classification task. For illustration, typical mesh and contour plots for pairs of values of these two independent variables are shown in Fig. 42. In the given case, lowest predictive classification error over validation set of patterns is achieved for the number of neighbors $K = 4$, and distance weight $m = 1.40$.

Fig. 43 shows example of the classification for three test patterns using discovered nearest prototypes and fuzzy classifier, for the values $K = 5$ and $m = 1.50$. In first two cases, superimposed impact of dominant neighbors, first and second, and first and fourth, respectively, easily assigns pattern to the right category. Third case shows more complex situation where the second and third neighbors of one category overcome the first and fifth neighbors of other category and again classify the pattern correctly.

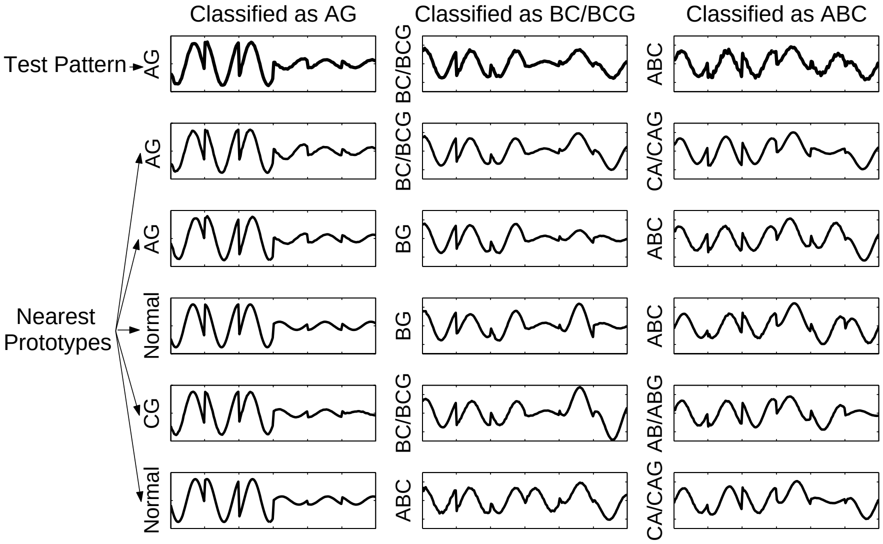


Fig. 43. Examples of fuzzyfied classification of test patterns

E. Design of Multilayer Perceptron Neural Network

Generally, the MLP neural network can represent any nonlinear or linear relationship between input and output vectors if the hidden layer or layers have enough neurons. The best number of hidden neurons depends in a complex way on the number of inputs and outputs, the number of training cases, the complexity of the function or classification to be learned, the network architecture, the type of hidden neuron activation function, the training algorithm and regularization. Selecting too few hidden units leads to high training error and high generalization error due to underfitting. Too many hidden units leads to low training error but high generalization error due to overfitting. Learning is a standard error-back-propagation algorithm, in which the network weights and biases are iteratively adjusted to minimize the performance function, defined as an average squared error between the actual network outputs and desired or target outputs [37, 36, 102].

Typical, single layer MLP neural network is utilized as a protective algorithm alternative to ART algorithm. The number of neural network inputs is 192, corresponding to sampling frequency of 32 samples/cycle and data window of one cycle. Number of neural network outputs is 8 when only fault type needs to be classified, or 15 when both fault type and section have to be recognized. Selected learning algorithm is a resilient propagation [192], convenient for high volume of processed data. The description of the algorithm is enclosed in Appendix. Adopted learning rate is 0.1. The training has been repeated numerous times for the networks related to both classification tasks, by changing the number of hidden neurons until optimal number of hidden neurons offering the lowest predictive error with respect to the validation set is discovered. Validation set is also used as an early stopping criterion to prevent overfitting. The training procedure terminates whenever validation error starts to increase in several consecutive epochs. The optimal number of hidden neurons discovered and subsequently applied is 45 for classifying fault type, and 75 for classifying fault type and section. For chosen number of hidden neurons, neural network with various initial network weights are trained and a case having the lowest training error is selected as an optimal solution. Learning terminated after 275 epochs for the neural network for classifying fault type, and after 465 epochs for the neural network for classifying fault type and section. Training and validation errors for both training cases are shown in Fig. 44.

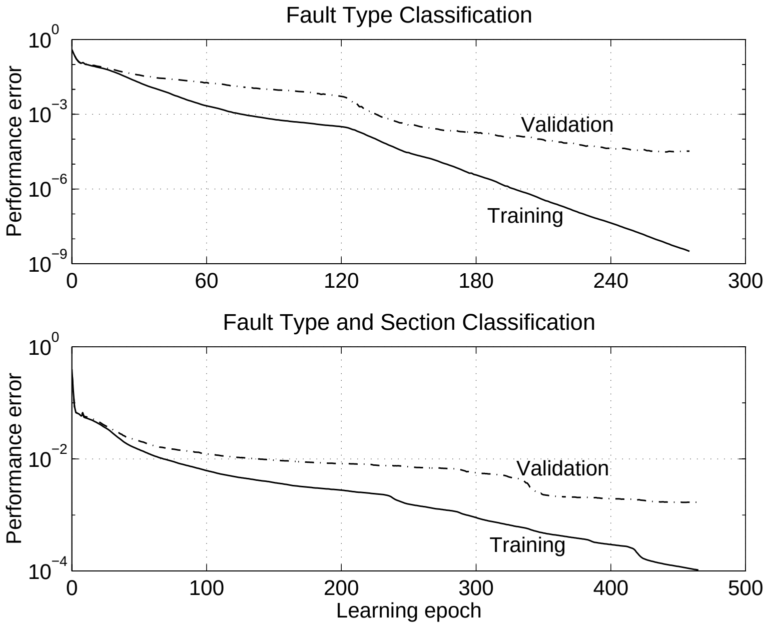


Fig. 44. Training procedure of MLP neural network for both classification tasks, with utilized training and validation sets

F. Design of Impedance Based Distance Relay

A model of typical impedance based distance relay is designed for comparison between traditional protection concept and new neural network algorithms. Impedance principle of distance relaying requires only the fundamental frequency components of obtained measurements. The sampling frequency of 32 samples/cycle and data window of one cycle are selected. Actual voltage and current amplitudes and phase angles are estimated using full cycle Fourier algorithm that rejects DC component [21]. Furthermore, the relay detects grounded or ungrounded faults, calculate the trajectory of apparent impedance according to (2.5) and compare with selected operating characteristics. The output of relaying algorithm is an indication of detected fault type and fault section.

The relay settings are calculated using a manual of a typical distance relay and proposed protection solution with *mho* and *quadrilateral* operating characteristics [193]. The relay is selected to have two zones of protection. Required reach for zone I is 80% of the line length. Zone II reach is set to be 150% of protected line length, to assure that all faults along the selected transmission line are detectable, even with strong infeed from the remote terminals. Three *mho* elements for phase-to-phase faults have been implemented for each zone. Impedance reach for zones I and II is determined within given ranges of line impedance. Three *quadrilateral* elements have been implemented for phase-to-ground faults. *Quadrilateral* reactive reach is based upon the same premises used for *mho* elements. However, the maximum resistive reach that determines sensitivity for high resistance faults is calculated using estimated load and line impedance. It is set to cover the desired level of ground fault resistance, but to be far enough from minimum load impedance, and is calculated using maximum primary voltage and current values, and transformer ratios. Distinction between grounded and ungrounded faults is achieved by calculating zero sequence current and comparing it to the threshold dependent on a minimal expected zero sequence current during grounded faults. All relay settings are defined in secondary quantities, obtained by dividing primary impedances by the quotient of the voltage and current transformer ratios. The description of this implementation is enclosed in Appendix.

G. Summary

The design implementation steps of the protection algorithms have been addressed in this chapter. Model of the given power network has been used for simulating various power system scenarios. Procedures of generating scenarios cases are performed attentively by varying fault and network parameters to roughly cover all possible occasions of events. The simulation outputs, three-phase voltage and current waveforms, are used for forming the sets of training and test patterns submitted as inputs to the ART and MLP neural network algorithms, or for forming the set of test inputs for the distance relay algorithm. Training procedures of both ART and MLP neural network algorithms are optimized to enhance classification capabilities for a variety of previously unseen patterns. The distance relay settings have been calculated according to proposed protection approach with *mho* and *quadrilateral* operating characteristics.

CHAPTER X

SIMULATION RESULTS

A. Introduction

This chapter presents comparative classification results for ART and MLP neural network based, and impedance based protection algorithms. The characteristics of generated training and test sets are outlined in the first section. Second section provides test results for fault events during nominal operating conditions for previous and novel ART algorithm - ART^1 and ART^2, respectively. Third section illustrates test results for fault events during disturbances for ART^1, ART^2, ART^2 with on-line learning, MLP and distance algorithm. In the fourth section, test results for non-fault events are shown for ART^1, ART^2, MLP and distance algorithm. The discussion of obtained results is given in fifth section.

B. Overview of the Training and Test Sets

The summary of the training, test and validation sets, previously defined in Chapter IX, is provided in Tables IV and V. Off-line training set is based on transmission line faults during nominal operating conditions, while on-line training set is based on transmission line faults during nominal and all adopted off-nominal operating conditions. Four test sets are based on line faults during either nominal or particular off-nominal conditions. Two additional test sets are based on non-fault events during nominal conditions. Validation set is based on line faults during nominal conditions and is used for tuning the fuzzy classifier.

Table IV. List of implemented training scenarios

		Set #	
		1	2
Training type	off-line	+	
	on-line		+
Network conditions	nominal	+	+
	weak infeed		+
	source deviation		+
	frequency deviation		+
Line fault	fault type	all 11 types and normal state	
	fault distance	$5 \div 95\%$, in increments of 10%	
	fault resistance	$3, 11, 19\Omega$	
	fault angle	$0 \div 336°$, in increments of $24°$	
Cases		3315	13260

Table V. List of implemented test and validation scenarios

		Set #						
		1	2	3	4	5	6	7
Type	test	+		+	+	+	+	+
	validation		+					
Network conditions	nominal	+	+					
	weak infeed			+				
	source deviation				+			
	frequency deviation					+		
	persistent line fault						+	
	remote end grounded							+
Line fault	fault type	all 11 types and normal state						
	fault distance	$0 \div 100\%$						
	fault resistance	$\sigma = 10\Omega$						
	fault angle	$0 \div 360°$						
Line switching	switching type						closing	
	switching angle						$0 \div 360°$	
Cases		5000	4000	5000	5000	5000	1000	100

C. Classification of Fault Events During Nominal Operating Conditions

Implemented ART algorithms are ART^1, characterized with original definition of supervised learning and non-fuzzy classifier, and ART^2 with advanced supervised learning and introduced fuzzy classifier. Tables VI and VII provide summaries of ART^1 and ART^2 training and test results for nominal power system, with various sets of training patterns used and classification tasks proposed. Patterns are gener-

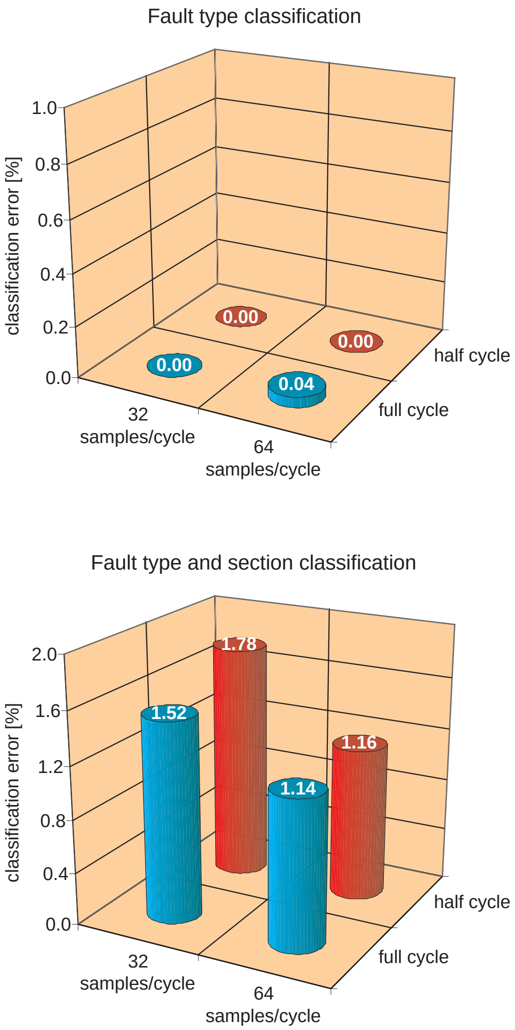
Table VI. Summary of ART^1 and ART^2 fault type classification results for nominal power system

			Case #					
			1	2	3	4	5	6
Input data	currents		+		+	+	+	+
	voltages			+	+	+	+	+
Pattern extraction	data window		1.0	1.0	0.5	0.5	1.0	1.0
	sampl. freq.		32	32	32	64	32	64
	length		96	96	96	192	192	384
Training type	off-line		+	+	+	+	+	+
	on-line							
Number of prototypes	ART^1		131	284	265	339	202	205
	ART^2		242	538	343	354	309	610
Classifier	ART^1	K	1	1	1	1	1	1
	ART^2	K	1	3	10	7	5	3
		m	-	1.20	1.45	1.50	1.50	1.55
Testing error [%]	ART^1		0.02	7.36	0.46	0.38	1.08	0.82
	ART^2		0.00	3.80	0.00	0.00	0.00	0.04

Table VII. Summary of ART^1 and ART^2 fault type and fault section classification results for nominal power system

			Case #						
			1	2	3	4	5	6	7
Input data	currents		+		+	+	+	+	+
	voltages			+	+	+	+	+	+
Pattern extraction	data window		1.0	1.0	0.5	0.5	1.0	1.0	1.0
	sampl. freq.		32	32	32	64	32	64	32
	length		96	96	96	192	192	384	192
Training type	off-line		+	+	+	+	+	+	
	on-line								+
Number of prototypes	ART^1		563	579	644	648	725	716	1887
	ART^2		877	826	750	834	916	934	2599
Classifier	ART^1	K	1	1	1	1	1	1	1
	ART^2	K	5	1	4	1	4	1	4
		m	1.75	-	1.30	-	1.40	-	1.40
Testing error [%]	ART^1		4.30	10.30	2.84	2.26	2.34	2.14	3.22
	ART^2		2.90	8.08	1.78	1.16	1.52	1.14	2.34

ated using different measurement quantities, sampling frequencies and data windows. Classification tasks are recognition of either fault type or both fault type and fault section. The number of pattern prototypes, classifier parameters and classification error are given for each trained solution.


Fault type classification

Fault type and section classification

Fig. 45. Classification error of ART^2 depending on used sampling frequency and data window

First observation is related to classification error of ART^2 for nominal system shown in Fig. 45, with respect to changing data window and sampling frequency used for forming the patterns. ART^2 provides very good classification of fault type, since error is zero or almost zero in all cases. For fault type and section classification, error is obviously lower with longer data window and especially with higher sampling frequency.

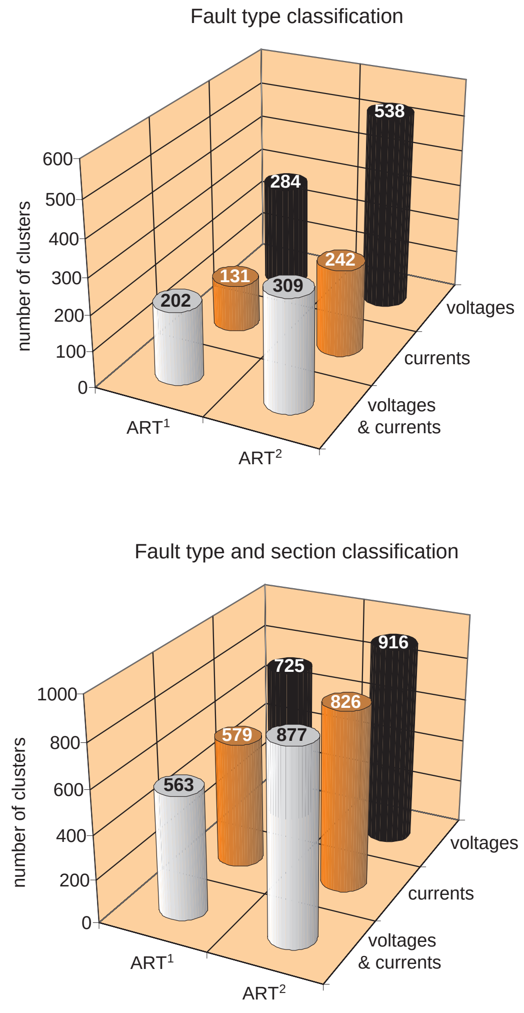
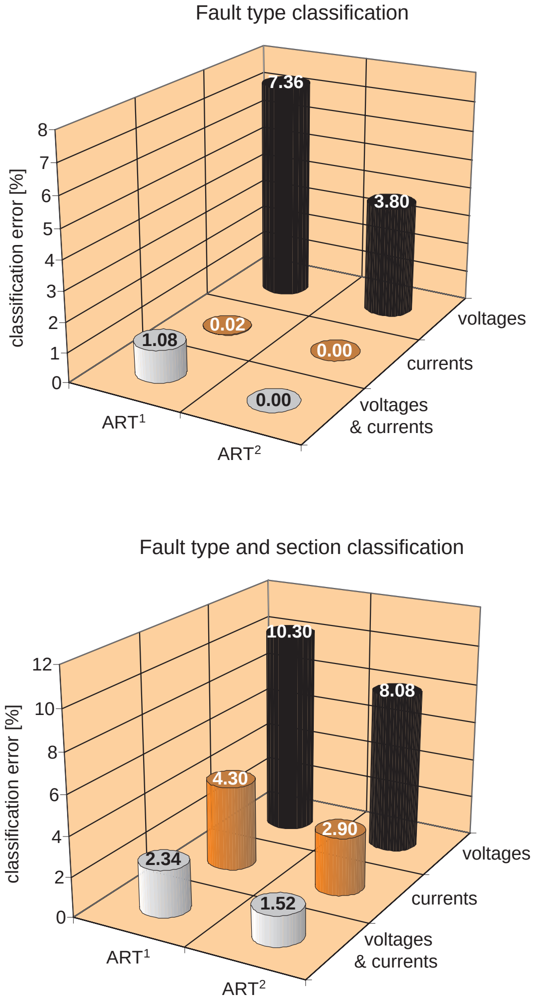


Fig. 46. Number of generated clusters of ART^1 and ART^2 depending on measurement quantities used

Fig. 46 shows comparison of the number of prototypes generated by ART^1 and ART^2, depending on sampling frequency and data window used. Comparing to ART^1, ART^2 produces more prototypes due to stronger condition for supervised learning, which generally accepts lower number of clusters per single iteration, and forces more clusters with smaller size. Since patterns based on voltages do not significantly vary during the faults, the number of clusters for fault type classification is higher because clusters are generally with smaller size. The patterns based on currents are very dissimilar to each other, and larger clusters are easily accepted during supervised learning. Mixed, current and voltage patterns and corresponding clusters inherit characteristics of both mentioned situations. For fault type and section classification case, the number of prototypes is few times higher than for fault type classification, because proper classification of faults around the boundary between two protection


Fault type classification

Fault type and section classification

Fig. 47. Classification error of ART^1 and ART^2 depending on measurement quantities used

sections requires more algorithm iterations until many small, fine-grained clusters, very close to each other in the pattern space are obtained. In this case there is no significant difference in number of prototypes between the use of current, voltage and combined current-voltage patterns.

The comparison of classification error of ART^1 and ART^2 for different type of measurements is given in Fig. 47. Obviously, results show the advantage of ART^2 comparing to ART^1. Generally, patterns based only on the voltage samples are not sufficient for successful classification, because they do not preserve enough qualitative information about characteristics of simulated events. However, ART^2 algorithm with patterns based on combination of currents and voltages guarantees very low classification error for both classification tasks.

D. Classification of Fault Events During Disturbances

Fig. 48 shows comparative evaluation for faults on the selected transmission line during nominal operating conditions, weak infeed, voltage source variation and system frequency variation, for implemented distance relay, MLP, ART^1 and ART^2 algorithms. ART^2 classifies fault type almost perfectly for all nominal and off-nominal network conditions and comparing to ART^1 and distance relay, extraordinary improvements have been observed. However, MLP possesses performance similar to ART^2 in classifying fault type. For fault type and section classification, besides algorithms already studied, an additional algorithm - ART^2 with on-line learning is introduced. Classification results show that ART^2 is somewhat sensitive to variations of voltage source and system frequency, and more sensitive for weak infeed. Repeatedly, it is in some degree better than ART^1 for all network conditions, reducing the classification error between $16 - 35\%$. Although practically insensitive

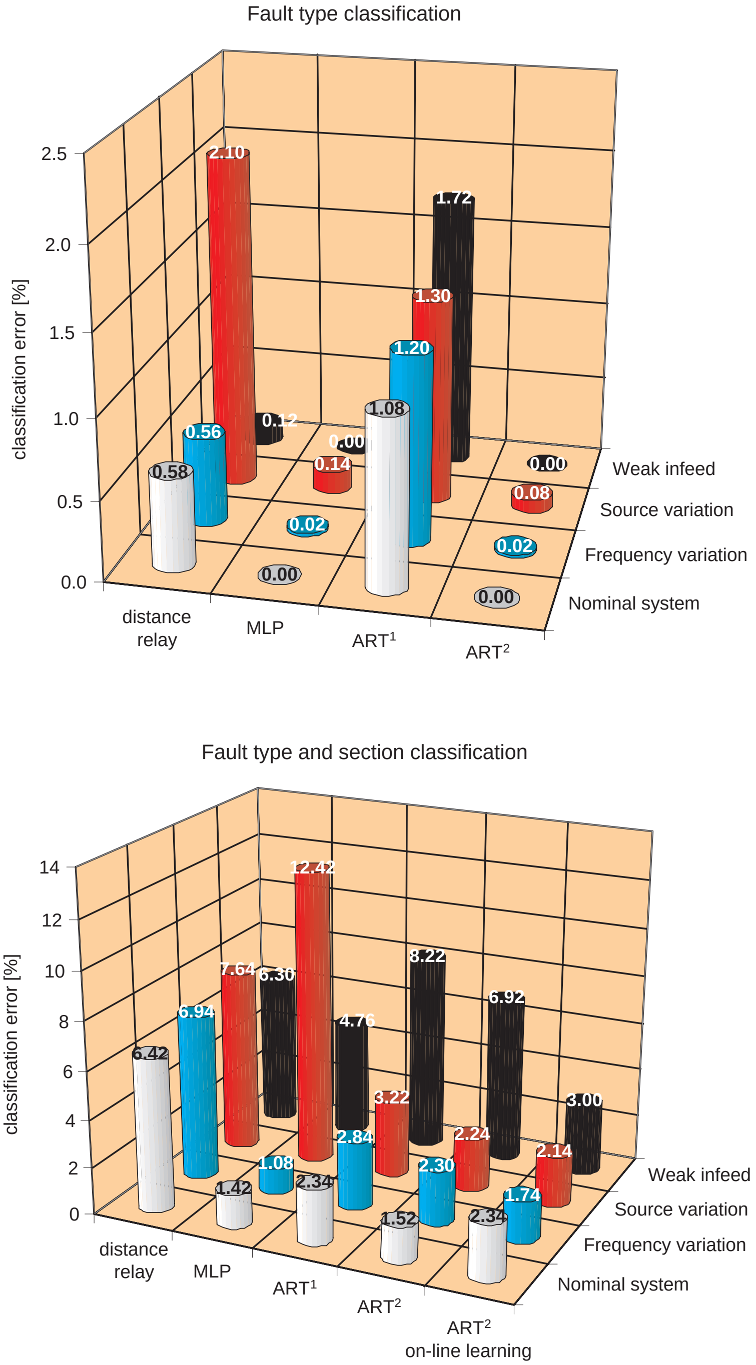


Fig. 48. Classification error of distance relay, MLP, ART^1, ART^2, and ART^2 with on-line learning, for faults during nominal and off-nominal power system conditions

for weak infeed, and not too much sensitive for other disturbances, distance relay model is generally not very accurate in classifying the fault type and section. ART^2 is significantly better than the distance relay model in fault section classification for all network conditions, except for weak infeed where it is slightly worse. This can be explained as a superior selectivity of ART^2 for the faults around the boundary between two zones of protection. Comparison of ART^2 with MLP shows that ART^2 is superior for source variation, where MLP is extremely poor. However, MLP is slightly better for frequency variation and weak infeed, while the responses for nominal system are similar. ART^2 with on-line learning is significantly better than ART^2 and the distance relay model for all off-nominal system conditions, and better than MLP except for frequency variation. As expected, ART^2 with on-line learning almost equally responds to nominal and off-nominal network conditions, although not being exactly good for nominal system as ART^2.

E. Classification of Non-Fault Events

Fig. 49 shows the classification results for switching onto grounded line and switching onto faulted line. ART^2 and distance relay model are insensitive for switching onto grounded line, ART^1 is very sensitive if trained to recognize only fault type, while MLP is completely useless. For switching onto persistent fault, distance relay model is better than other algorithms. ART^1 and ART^2 are similar between each other and worse than the distance relay model, while MLP response is again unacceptable.

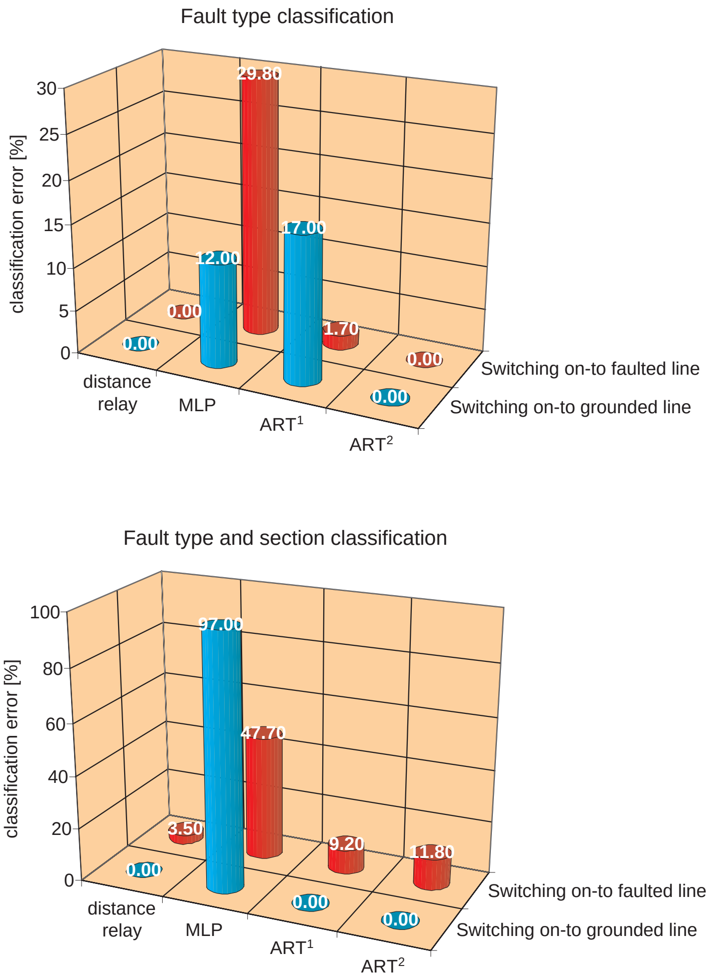


Fig. 49. Classification error of distance relay, MLP, ART^1 and ART^2, for switching on-to grounded and faulted line

F. Discussion

Several remarks can be provided regarding the results. First of all, simultaneous classification of fault type and section has been determined to be much more difficult task than classifying only fault type. ART^2 shows better recognition capabilities than ART^1 for all anticipated scenarios. It is better than the MLP algorithm and distance relay model for almost all scenarios, with significantly smaller classification error.

Therefore, ART^2 with the fuzzy classifier and improved supervised learning provides remarkable recognition capabilities, and is fairly robust for all studied off-nominal operating conditions. With on-line learning it becomes an exceptional solution for unpredictable deviations of power system operating conditions. The classification results are satisfactory for all implemented values of sampling frequency and data window, that correspond to typical values used in protective relaying.

G. Summary

This chapter provides comprehensive evaluation of proposed ART^2 neural network protection algorithm and its comparison with other relevant algorithms, ART^1 as a previous version of ART algorithm, MLP neural network algorithm and distance relay model. All implemented algorithms have been designed and tested for two tasks: a) fault detection and classification; and b) fault detection, classification and characterization. Test scenarios have included line faults and non-fault events during nominal and perturbed operating conditions. Simulation results show satisfactory responses for ART^2 algorithm for each of the implemented scenarios, and superior recognition capabilities comparing to ART^1 algorithm as well as MLP algorithm and distance relay model.

CHAPTER XI

CONCLUSION

A. Summary of the Achievements

The Dissertation presents a new solution for high-speed protective relaying using artificial intelligence and pattern recognition. The approach utilizes self-organized, Adaptive Resonance Theory (ART) neural network, based on combined use of unsupervised and supervised learning methods. Training of the neural network translates set of input events into a smaller set of their typical prototypes. During implementation of the neural network, real events are associated to the generated prototypes, and classified by fuzzy decision rule. Proposed neuro-fuzzy relaying solution may successfully substitute conventional relays vulnerable to occasional missoperation due to their heavy dependency on varying system and fault conditions.

This Dissertation has introduced significantly advanced ART algorithm comparing to the previous solution, with several important improvements. Input signal preprocessing have been applied and optimized, and enables finer selection of the most representative input data subset used to establish pattern prototypes during training, or to match existing prototypes during implementation. Supervised learning has been redefined and allows improved prototype generation, and their better utilization during implementation. Furthermore, previously used simple, non-fuzzy classifier has been substituted with advanced, fuzzy classifier that provides interpolation technique for interpreting the neural network outputs. Moreover, on-line learning has been introduced and guarantees well-tuned pattern recognition capabilities during prevailing operating conditions. The algorithm has been designed and evaluated in the new simulation environment by using extensive sets of training and test patterns obtained by simulating convenient power network model and a variety of fault and disturbance scenarios.

B. Research Contribution

Simulation results obtained show exceptional agility and robustness of the novel ART neural network algorithm for nominal and off-nominal power system conditions. The algorithm possesses superior classification capabilities compared to previous version of ART algorithm, MLP neural network algorithm, and impedance based distance relay.

Presented solution, properly implemented and tuned, assures reliable fault detection, and selective fault type and fault section estimation. Combined use of neural and fuzzy techniques leads to a sophisticated reasoning that may improve the ability of recognizing a variety of real events usually dissimilar to the events anticipated during training. This method is robust to untypical and noisy inputs, and is quite effective whenever sufficiently large set of relevant input events is available. The interaction between dedicated programs for independent power system modeling and protective algorithm design leads to significantly improved relay design implementation and performance evaluation.

The robust pattern recognition approach developed in this dissertation can be utilized in many other applications, whenever high amount of diverse input data is present, numerous output categories are required and complex decision making process is expected.

C. Directions for Future Research

Future work related to proposed neuro-fuzzy pattern recognition approach may take several directions. It may continue with the study of algorithm dependence on input signal preprocessing steps, by anticipating the use of unequal data windows for voltage and current samples, independent scaling of pattern components, feature extraction, etc. Furthermore, the supervised learning can be repeatedly revised to be more restrictive, by completely preventing the overlapping among the clusters with different categories. Finally, the modifications of fuzzy classifier may be studied, to allow more controllable influence of cluster size on a membership function of each category.

REFERENCES

[1] W. A. Lewis and S. Tippet, "Fundamental basis for distance relaying on 3-phase systems," *AIEE Trans.*, vol. 66, pp. 694–708, Feb. 1947.

[2] IEEE PES Power Systems Relaying Committee, "IEEE guide for protective relay applications to transmission lines," IEEE Standard C37.113-1999, Feb. 2000.

[3] G. A. Carpenter and S. Grossberg, "A massively parallel architecture for a self-organizing neural pattern recognition machine," *Computer Vision, Graphics, and Image Processing*, vol. 37, no. 1, pp. 54–115, Jan. 1987.

[4] Y. H. Pao, *Adaptive Pattern Recognition and Neural Networks*, Reading, MA: Wesley, 1989.

[5] Working Group D5 of the Line Protection Subcommittee of the IEEE Power System Relaying Committee, "Proposed statistical performance measures for microprocessor-based transmission line protective relays, Part I: Explanation of the statistics, Part II: Collection and uses of data," *IEEE Trans. Power Delivery*, vol. 12, no. 1, pp. 134–156, Jan. 1997.

[6] I. Rikalo, "Neural net application to transmission line fault detection and classification," M.S. thesis, Texas A&M Univ., College Station, TX, Dec. 1994.

[7] M. Kezunovic, I. Rikalo, and D. Sobajic, "High-speed fault detection and classification with neural nets," *Electric Power Systems Research*, vol. 34, no. 2, pp. 109–116, Aug. 1995.

[8] M. Kezunovic and I. Rikalo, "Detect and classify faults using neural nets," *IEEE Computer Applications in Power*, vol. 9, no. 4, pp. 42–47, Oct. 1996.

[9] M. Kezunovic, I. Rikalo, and D. Sobajic, "Real-time and off-line transmission line fault classification using neural networks," *Engineering Intelligent Systems*, vol. 4, no. 1, pp. 57–63, Mar. 1996.

[10] S. Vasilic, M. Kezunovic, and D. Sobajic, "Optimizing the performance of transmission line relaying algorithm implemented using an adaptive, self-organized neural network," in *Proc. 14th Power System Computation Conf.*, Seville, Spain, June 2002, ses. 42, no. 3, pp. 1–6.

[11] S. Vasilic and M. Kezunovic, "An improved neural network algorithm for classifying the transmission line faults," in *Proc. IEEE PES Winter Meeting*, New York, Jan. 2002, vol. 2, pp. 918–923.

[12] S. Vasilic and M. Kezunovic, "Fuzzy ART neural network algorithm for classifying the power system faults," *IEEE Trans. Power Delivery*, accepted.

[13] D. Ristanovic, S. Vasilic, and M. Kezunovic, "Design and implementation of scenarios for evaluating and testing distance relays," in *Proc. North American Power Symp.*, College Station, TX, Oct. 2001, pp. 185–192.

[14] S. Vasilic and M. Kezunovic, "New design of a neural network algorithm for detecting and classifying transmission line faults," presented at the IEEE PES Transmission and Distribution Conf., Atlanta, GA, Oct. 2001.

[15] M. Kezunovic and S. Vasilic, "Design and evaluation of context-dependent protective relaying approach," in *Proc. IEEE Porto Power Tech*, Porto, Portugal, Sep. 2001, vol. 4, pp. 1–6.

[16] M. Kezunovic, S. Vasilic, and F. Gul-Bagriyanik, "Advanced approaches for detecting and diagnosing transients and faults," in *Proc. 3rd Mediterranean*

Conf. on Power Generation, Transmission, Distribution and Energy Conversion, Athens, Greece, Nov. 2002, no. 177, pp. 1–6.

[17] M. Kezunovic and S. Vasilic, "Advanced software environment for evaluating protection performance during power system disturbances using relay models," in *Proc. CIGRE SC 34 Colloq.*, Sibiu, Romania, Sep. 2001, no. 116, pp. 1–6.

[18] M. Kezunovic, S. Vasilic, and D. Ristanovic, "Interfacing protective relays and relay models to power system modeling software and data files," in *Proc. Intl. Conf. Power System Technology*, Kunming, China, Oct. 2002, vol. 1, pp. 253–259.

[19] M. Kezunovic and S. Vasilic, "Analysis of protective relaying operation and related power system interaction," in *Proc. IFAC Symp. Power Plants and Power Systems Control*, Seoul, Korea, Sep. 2003, pp. 399–404.

[20] P. Anderson, *Power System Protection*, New York: McGraw-Hill, 1999.

[21] A. G. Phadke and J. S. Thorp, *Computer Relaying for Power Systems*, Taunton, United Kingdom: Wiley, 1988.

[22] M. S. Sachdev (editor), "Microprocessor relays and protection systems," IEEE Tutorial Course Text 88EH0269-1-PWR, Feb. 1988.

[23] G. D. Rockefeller, "Fault protection with a digital computer," *IEEE Trans. Power App. Syst.*, vol. 88, pp. 438–462, Apr. 1969.

[24] V. Cook, *Analysis of Distance Protection*, New York: Wiley, 1985.

[25] Working Group of the Line Relay Protection Subcommittee of the AIEE Relays Committee, "Protection of multiterminal taped lines," *AIEE Trans., Part III. Power Aparatus and Systems*, vol. 80, pp. 55–65, Apr. 1961.

[26] Working Group of the Line Protection Subcommittee of the IEEE Power Systems Relaying Committee, "Protection aspects of multi-terminal lines," Research Report TH0056-2-PWR, New York, Feb. 1979.

[27] W. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *Bullet. Mathematical Biophysics*, vol. 5, pp. 115–133, 1943.

[28] D. O. Hebb, *The Organization of Behavior: A Neuropsychological Theory*, New York: Wiley, 1949.

[29] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," *Psychological Review*, vol. 65, no. 6, pp. 386–408, Nov. 1958.

[30] B. Widrow and M. E. Hoff, "Adaptive switching circuits," in *Conv. Record IRE Western Electronic Show and Convention*, Los Angeles, CA, Aug. 1960, vol. 4, pp. 96–104.

[31] T. Kohonen, "Correlation matrix memories," *IEEE Trans. Computers*, vol. 21, no. 4, pp. 353–359, Apr. 1972.

[32] J. A. Anderson, "A simple neural network generating an interactive memory," *Mathematical Biosciences*, vol. 14, no. 3–4, pp. 197–220, Aug. 1972.

[33] S. Grossberg, "Adaptive pattern classification and universal recording: I. Parallel development and coding of feature detectors," *Biological Cybernetics*, vol. 23, no. 3, pp. 121–134, July 1976.

[34] S. Grossberg, "Adaptive pattern classification and universal recording: II. Feedback, expectation, olfaction, and illusions," *Biological Cybernetics*, vol. 23, no. 4, pp. 187–202, Aug. 1976.

[35] J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proc. Natl. Academy of Sciences*, vol. 79, pp. 2554–2558, Apr. 1982.

[36] D. E. Rumelhart and J. L. McClelland (editors), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Volume I: Foundations*, Cambridge, MA: MIT Press, 1986.

[37] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagation errors," *Nature*, vol. 323, pp. 533–536, Oct. 1986.

[38] D. S. Broomhead and D. Lowe, "Multivariable functional interpolation and adaptive networks," *Complex Systems*, vol. 2, no. 3, pp. 321–355, June 1988.

[39] V. Vapnik, *The Nature of Statistical Learning Theory*, New York: Springer, 1998.

[40] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, New York: Wiley, 2001.

[41] F. Rosenblatt, *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanism*, Buffalo, NY: Cornell Aeronautical Laboratory, 1961.

[42] S. P. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Information Theory*, vol. 28, no. 2, pp. 129–137, Mar. 1982.

[43] G. H. Ball and D. J. Hall, "A clustering technique for summarizing multivariate data," *Behavioral Science*, vol. 12, pp. 153–155, Mar. 1967.

[44] R. Didday, "A model of visuomotor mechanisms in the frog optic tectum," *Mathematical Bioscience*, vol. 30, no. 1–2, pp. 169–180, July 1976.

[45] F. Montalvo, "Consensus versus competition in neural networks: A comparative analysis of three models," *Intl. Jour. Man-Machine Studies*, vol. 7, no. 3, pp. 333–346, May 1975.

[46] S. Amari and M. A. Arbib, "Competition and cooperation in neural nets," in *Systems Neuroscience*, J. Metzler, Ed. New York: Academic Press, 1977, pp. 119–165.

[47] T. Kohonen, *Self-organization and Associative Memory*, New York: Springer, 1989.

[48] T. Kohonen, *Self-Organizing Maps*, New York: Springer, 2001.

[49] R. Hecht-Nielsen, *Neurocomputing*, Reading, MA: Wesley, 1990.

[50] T. Kohonen, "Clustering, taxonomy, and topological maps of patterns," in *Proc. 6th Intl Conf. Pattern Recognition*, Munich, Germany, Oct. 1982, vol. 1, pp. 114–128.

[51] T. Kohonen, "Self-organizing formation of topologically correct feature maps," *Biological Cybernetics*, vol. 43, no. 1, pp. 59–69, Jan. 1982.

[52] H. Ritter, T. Martinez, and K. Schulten, *Neural Computation and Self-Organizing Maps: An Introduction*, Reading, MA: Wesley, 1992.

[53] T. Kohonen, "An introduction to neural computing," *Neural Networks*, vol. 1, no. 1, pp. 3–16, 1988.

[54] T. Kohonen, "Learning vector quantization," in *Proc. 1st Intl. Neural Network Society Meeting*, Boston, MA, Sep. 1988, pp. 303–305.

[55] G. A. Carpenter and S. Grossberg, "ART2: Self-organization of stable category recognition codes for analog input patterns," *Applied Optics*, vol. 26, no. 23, pp. 4919–4930, Dec. 1987.

[56] G. A. Carpenter and S. Grossberg, "ART3: Hierarchical search using chemical transmitters in self-organizing pattern recognition architectures," *Neural Networks*, vol. 3, no. 2, pp. 129–152, 1990.

[57] G. A. Carpenter, S. Grossberg, and J. Reynolds, "ARTMAP: Supervised real-time learning and classification of nonstationary data by a self-organizing neural network," *Neural Networks*, vol. 4, no. 5, pp. 565–588, 1991.

[58] T. M. Cover and P. E. Hart, "Nearest neighbor pattern classification," *IEEE Trans. Information Theory*, vol. 13, pp. 21–27, Jan. 1967.

[59] G. A. Carpenter, S. Grossberg, and D. B. Rosen, "ART2-A: An adaptive resonance algorithm for rapid category learning and recognition," *Neural Networks*, vol. 4, no. 4, pp. 493–504, 1991.

[60] J. Yen and R. Langari, *Fuzzy Logic: Intelligence, Control, and Information*, Upper Saddle River, NJ: Prentice Hall, 1998.

[61] L. A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, no. 3, pp. 338–353, June 1965.

[62] R. E. Bellman, R. Kalaba, and L. A. Zadeh, "Abstraction and pattern classification," *Jour. Mathematical Analysis and Applications*, vol. 13, no. 1, pp. 1–7, Jan. 1966.

[63] E. H. Ruspini, "A new approach to fuzzy clustering," *Information Sciences*, vol. 2, no. 3, pp. 319–350, July 1970.

[64] J. C. Dunn, "A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters," *Jour. Cybernetics*, vol. 3, no. 3, pp. 32–57, July/Sep. 1973.

[65] J. C. Bezdek, "Fuzzy mathematics in pattern classification," Ph.D. dissertation, Cornell Univ., Ithaca, NY, Sep. 1973.

[66] J. Keller, M. R. Gray, and J. A. Givens, "A fuzzy K-nearest neighbor algorithm," *IEEE Trans. Systems, Man and Cybernetics*, vol. 15, no. 4, pp. 580–585, July/Aug. 1985.

[67] I. Gath and A. B. Geva, "Unsupervised optimal fuzzy clustering," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 773–781, July 1989.

[68] G. A. Carpenter, S. Grossberg, and D. B. Rosen, "Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system," *Neural Networks*, vol. 4, no. 6, pp. 759–771, 1991.

[69] G. A. Carpenter, S. Grossberg, N. Markuzon, J. Reynolds, and D. Rosen, "Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps," *IEEE Trans. Neural Networks*, vol. 3, no. 5, pp. 698–713, Sep. 1992.

[70] M. A. El-Sharkawi and D. Niebur (editors), "Application of artificial neural networks to power systems," IEEE Tutorial Course Text 96TP112-0, Piscataway, NJ, 1996.

[71] R. Aggarwal and Y. Song, "Artificial neural networks in power systems: Part I - General introduction into neural computing," *Power Engineering Jour.*, vol.

11, no. 3, pp. 129–134, June 1997.

[72] R. Aggarwal and Y. Song, "Artificial neural networks in power systems: Part II - Types of artificial neural networks," *Power Engineering Jour.*, vol. 12, no. 1, pp. 41–47, Feb. 1998.

[73] R. Aggarwal and Y. Song, "Artificial neural networks in power systems: Part III - Examples of applications in power systems," *Power Engineering Jour.*, vol. 12, no. 6, pp. 279–287, Dec. 1998.

[74] D. Srinivasan, A. C. Liew, and C. S. Chang, "Applications of fuzzy systems in power systems," *Electric Power Systems Research*, vol. 35, no. 1, pp. 39–43, Oct. 1995.

[75] Y.-H. Song and A. T. Johns, "Applications of fuzzy logic in power systems: Part I - General introduction to fuzzy logic," *Power Engineering Jour.*, vol. 11, no. 5, pp. 219–222, Oct. 1997.

[76] Y.-H. Song and A. T. Johns, "Applications of fuzzy logic in power systems: Part II - Comparison and integration with expert systems, neural networks and genetic algorithms," *Power Engineering Jour.*, vol. 12, no. 4, pp. 185–190, Aug. 1998.

[77] Y.-H. Song and A. T. Johns, "Applications of fuzzy logic in power systems: Part III - Example applications," *Power Engineering Jour.*, vol. 13, no. 2, pp. 97–103, Apr. 1999.

[78] M. Kezunovic, "A survey of neural net applications to protective relaying and fault analysis," *Engineering Intelligent Systems for El. Engr. and Comm.*, vol. 5, no. 4, pp. 185–192, Dec. 1997.

[79] T. Dalstein and B. Kulicke, “Neural network approach to fault type classification for high speed protective relaying,” *IEEE Trans. Power Delivery*, vol. 10, no. 2, pp. 1002–1011, Apr. 1995.

[80] W. W. L. Keerthipala, C. T. Wai, and W. Huisheng, “Neural network based classifier for power system protection,” *Electric Power Systems Research*, vol. 42, no. 2, pp. 109–114, Aug. 1997.

[81] Y. H. Song, Q. X. Xuan, and A. T. Johns, “Comparison studies of five neural network based fault classifiers for complex transmission lines,” *Electric Power Systems Research*, vol. 43, no. 2, pp. 125–132, Nov. 1997.

[82] H. Wang and W. W. L. Keerthipala, “Fuzzy-neuro approach to fault classification for transmission line protection,” *IEEE Trans. Power Delivery*, vol. 13, no. 4, pp. 1093–1104, Oct. 1998.

[83] A. Poeltl and K. Frohlich, “Two new methods for very fast fault type detection by means of parameter fitting and artificial neural networks,” *IEEE Trans. Power Delivery*, vol. 14, no. 4, pp. 1269–1275, Oct. 1999.

[84] P. K. Dash, A. K. Pradhan, and G. Panda, “A novel fuzzy neural network based distance relaying scheme,” *IEEE Trans. Power Delivery*, vol. 15, no. 3, pp. 902–907, July 2000.

[85] W.-M. Lin, C.-D. Yang, J.-H. Lin, and M.-T. Tsay, “A fault classification method by RBF neural network with OLS learning procedure,” *IEEE Trans. Power Delivery*, vol. 16, no. 4, pp. 473–477, Oct. 2001.

[86] A. K. Pradhan, P. K. Dash, and G. Panda, “A fast and accuarte distance relaying scheme using an efficient radial basis function neural network,” *Electric*

Power Systems Research, vol. 60, no. 4, pp. 1–8, Nov. 2001.

[87] T. S. Sidhu, H. Singh, and M. S. Sachdev, "Design, implementation and testing of an artificial neural network based fault detection discriminator for protecting transmission lines," *IEEE Trans. Power Delivery*, vol. 10, no. 2, pp. 697–706, Apr. 1995.

[88] M. Sanaye-Pasand and O. P. Malik, "Implementation and laboratory test results of an Elman network-based transmission line directional relay," *IEEE Trans. Power Delivery*, vol. 14, no. 3, pp. 782–788, July 1999.

[89] A. L. O. Fernandez and N. K. I. Ghonaim, "A novel approach using a FIRANN for fault detection and direction estimation for high-voltage transmission lines," *IEEE Trans. Power Delivery*, vol. 17, no. 4, pp. 894–900, Oct. 2002.

[90] T. Dalstein, T. Friedrich, B. Kulicke, and D. Sobajic, "Multi neural network based fault area estimation for high speed protective relaying," *IEEE Trans. Power Delivery*, vol. 11, no. 2, pp. 740–747, Apr. 1996.

[91] F. Zahra, B. Jeyasurya, and J. E. Quaicoe, "High-speed transmission line relaying using artificial neural networks," *Electric Power Systems Research*, vol. 53, no. 3, pp. 173–179, Mar. 2000.

[92] P. K. Dash, A. K. Pradhan, and G. Panda, "Application of minimal radial basis function neural network to distance protection," *IEEE Trans. Power Delivery*, vol. 16, no. 1, pp. 68–74, Jan. 2001.

[93] A. G. Jongepier and L. van der Sluis, "Adaptive distance protection of double-circuit lines using artificial neural networks," *IEEE Trans. Power Delivery*, vol. 12, no. 1, pp. 97–105, Jan. 1997.

[94] R. K. Aggarwal, Q. Y. Xuan, R. W. Dunn, A. T. Johns, and A. Bennet, "A novel fault classification technique for double-circuit lines based on a combined unsupervised/supervised neural network," *IEEE Trans. Power Delivery*, vol. 14, no. 4, pp. 1250–1256, Oct. 1999.

[95] D. Novosel, B. Bachmann, D. Hart, Y. Hu, and M. M. Saha, "Algorithms for locating faults on series compensated lines using neural network and deterministic methods," *IEEE Trans. Power Delivery*, vol. 11, no. 4, pp. 1728–1736, Oct. 1996.

[96] Z. Chen and J.-C. Maun, "Artificial neural network approach to single-ended fault locator for transmission lines," *IEEE Trans. Power Systems*, vol. 15, no. 1, pp. 370–375, Feb. 2000.

[97] D. S. Fitton, R. W. Dunn, R. K. Aggarwal, A. T. Johns, and A. Bennett, "Design and implementation of an adaptive single pole autoreclosure technique for transmission lines using artificial neural networks," *IEEE Trans. Power Delivery*, vol. 11, no. 2, pp. 748–756, Apr. 1996.

[98] Z. Q. Bo, R. K. Aggarwal, A. T. Johns, H. Y. Li, and Y. H. Song, "A new approach to phase selection using fault generated high frequency noise and neural networks," *IEEE Trans. Power Delivery*, vol. 12, no. 1, pp. 106–115, Jan. 1997.

[99] D. K. Ranaweera, "Comparison of neural network models for fault diagnosis of power systems," *Electric Power Systems Research*, vol. 29, no. 2, pp. 99–104, Mar. 1994.

[100] K. L. Butler, J. A. Momoh, and D. J. Sobajic, "Field studies using a neural-net-based approach for fault diagnosis in distribution networks," *IEE Proc.*

Generation, Transmission and Distribution, vol. 144, no. 5, pp. 429–436, Sep. 1997.

[101] S. Websper, R. W. Dunn, R. K. Aggarwal, A. T. Johns, and A. Bennett, "Feature extraction methods for neural network-based transmission line fault discrimination," *IEE Proc. Generation, Transmission and Distribution*, vol. 146, no. 3, pp. 209–216, May 1999.

[102] B. Widrow and M. Lehr, "30 years of adaptive neural networks: perceptron, madaline, and backpropagation," *IEEE Proc.*, vol. 78, no. 9, pp. 1415–1442, Sep. 1990.

[103] M. J. D. Powell, "Radial basis functions for multivariable interpolation: A review," in *Algorithms for the Approximations of Functions and Data*, J. C. Mason and M. G. Cox, Eds. Oxford, United Kingdom: Clarendon Press, 1987, pp. 143–167.

[104] J. L. Elman, "Finding structure in time," *Cognitive Science*, vol. 14, no. 4, pp. 179–211, Apr./June 1990.

[105] R. Hecht-Nielsen, "Counterpropagation networks," *Applied Optics*, vol. 26, no. 23, pp. 4979–4984, Dec. 1987.

[106] E. A. Wan, "Temporal backpropagation for FIR neural networks," in *Proc. Intl. Joint Conf. Neural Networks*, San Diego, CA, June 1990, vol. 1, pp. 575–580.

[107] A. Ferrero, S. Sangiovanni, and E. Zappitelli, "A fuzzy-set approach to fault-type identification in digital relaying," *IEEE Trans. Power Delivery*, vol. 10, no. 1, pp. 169–175, Jan. 1995.

[108] P. Jarventausta, P. Verho, and J. Partanen, "Using fuzzy sets to model the uncertainty in the fault location process of distribution networks," *IEEE Trans. Power Delivery*, vol. 9, no. 2, pp. 954–960, Apr. 1994.

[109] H. Monsef, A. M. Ranjbar, and S. Jadid, "Fuzzy rule-based expert system for power system fault diagnosis," *IEE Proc. Generation, Transmission and Distribution*, vol. 144, no. 2, pp. 186–192, Mar. 1997.

[110] W. M. Al-hassawi, N. H. Abbasi, and M. M. Mansour, "A neural-network based approach for fault classification and faulted phase selection," in *Proc. Canadian Conf. Electrical and Computer Engineering*, Calgary, Canada, May 1996, vol. 1, pp. 384–385.

[111] Z. Q. Bo, G. S. Wang, F. Jiang, P. J. Moore, and A. T. Johns, "Distinguish between fault and switching operation generated transients using a fuzzy neural network," in *Proc. 2nd Intl. Conf. Energy Management and Power Delivery*, Singapore, Mar. 1998, vol. 1, pp. 117–121.

[112] Z. Chen, A. Kalam, and A. Zayegh, "Advanced microprocessor based power protection system using artficial neural networks techniques," in *Proc. Intl. Conf. Energy Management and Power Delivery*, Singapore, Nov. 1995, vol. 1, pp. 439–444.

[113] D. V. Coury and D. C. Jorge, "Artificial neural network approach to distance protection of transmission lines," *IEEE Trans. Power Delivery*, vol. 13, no. 1, pp. 102–108, Jan. 1998.

[114] D. V. Coury and M Oleskovicz, "Applying pattern recognition in distance protection," in *Proc. Intl. Conf. Power System Technology*, Beijing, China, Aug. 1998, vol. 2, pp. 1148–1152.

[115] B. H. Chowdhury and K. Wang, “Fault classification using Kohonen feature mapping,” in *Proc. Intl. Conf. Intelligent System Applications to Power Systems*, Orlando, FL, Jan./Feb. 1996, pp. 194–198.

[116] K. L. Butler and J. A. Momoh, “A neural net based approach for fault diagnosis in distribution networks,” in *Proc. IEEE PES Winter Meeting*, Singapore, Jan. 2000, vol. 2, pp. 1275–1278.

[117] H. Jiali, D. Yuqian, L. Yongli, W. Gang, and L. Shanshan, “Distance relay protection based on artificial neural network,” in *Proc. 4th Intl. Conf. Advances in Power System Control, Operation and Management*, Hong Kong, Nov. 1997, vol. 2, pp. 515–520.

[118] N. Kandil, V. K. Sood, K. Khorasani, and R. V. Patel, “Fault identification in an AC-DC transmission system using neural networks,” *IEEE Trans. Power Systems*, vol. 7, no. 2, pp. 812–819, May 1992.

[119] H. Kanoh, M. Kaneta, and K. Kanemaru, “Fault location for transmission lines using inference model neural network,” *Electrical Engineering in Japan*, vol. 111, no. 7, pp. 84–93, Dec. 1991.

[120] X.-R. Wang, S.-T. Wu, and Q.-Q. Qian, “Neural network approach to power transmission line fault classification,” in *Proc. Intl. Conf. Information, Communications and Signal Processing*, Singapore, Sep. 1997, pp. 1737–1740.

[121] M. Kezunovic, I. Rikalo, and D. Sobajic, “Neural network applications to real-time and off-line fault analysis,” in *Proc. Intl. Conf. Intelligent System Applications to Power Systems*, Montpellier, France, Sep. 1994, vol. 1, pp. 29–36.

[122] B. Kasztenny, "Artificial neural network approach to power system protection," in *Proc. Intl. Conf. Intelligent System Applications to Power Systems*, Montpellier, France, Sep. 1994, vol. 2, pp. 665–671.

[123] W. W. L. Keerthipala, W. Huisheng, and C. T. Wai, "On-line testing of a fuzzy-neuro based protective relay using a real-time digital simulator," in *Proc. IEEE PES Winter Meeting*, Singapore, Jan. 2000, vol. 3, pp. 1917–1922.

[124] S. A. Khaparde, P. B. Kale, and S. H. Agarwal, "Application of artificial neural network in protective relaying of transmission lines," in *Proc. 1st Intl. Forum Application of Neural Networks to Power Systems*, Seattle, WA, July 1991, pp. 122–125.

[125] M. Oleskovicz, D. V. Coury, and A. C. P. L. F. de Carvalho, "Artificial neural network applied to power system protection," in *Proc. 5th Brazilian Symp. Neural Networks*, Belo Horizonte, Brazil, Dec. 1998, pp. 247–252.

[126] M. Oleskovicz, D. V. Coury, and R. K. Aggarwal, "A complete scheme for fault detection, classification and location in transmission lines using neural networks," in *Proc. 7th Intl. Conf. Developments in Power System Protection*, Amsterdam, Netherlands, Apr. 2001, pp. 335–338.

[127] Y. H. Song, A. T. Johns, Q. Y. Xuan, and J. Y. Liu, "Genetic algorithm based neural networks applied to fault classification for EHV transmission lines with a UPFC," in *Proc. 6th Intl. Conf. Developments in Power System Protection*, Nottingham, United Kingdom, Mar. 1997, pp. 278–281.

[128] K. S. Swarup and H. S. Chandrasekharaiah, "Fault detection and diagnosis of power systems using artiicial neural networks," in *Proc. 1st Intl. Forum*

Applications of Neural Networks to Power Systems, Seattle, WA, July 1991, pp. 102–106.

[129] E. Vazquez, H. J. Altuve, and O. L. Chacon, "Neural network approach to fault detection in electric power systems," in *Proc. IEEE Intl. Conf. Neural Networks*, Washington, DC, June 1996, vol. 4, pp. 2090–2095.

[130] R. Venkatesan and B. Balamurugan, "A real-time hardware fault detector using an artificial neural network for distance protection," *IEEE Trans. Power Delivery*, vol. 16, no. 1, pp. 75–82, Jan. 2001.

[131] R. K. Aggarwal, A. T. Johns, and Q. Y. Xuan, "Performance evaluation of different neural networks as applied to directional discrimination in faulted teed transmission circuits," in *Proc. 13th Power Systems Computation Conf. - PSCC'99*, Trondheim, Norway, June/July 1999, vol. 2, pp. 815–819.

[132] T. Dalstein, D. J. Sobajic, B. Kulicke, and Y. H. Pao, "Neural network approach to fault direction identification in electric power systems," in *Proc. North American Power Symposium*, Washington, DC, Oct. 1993, pp. 290–299.

[133] T. S. Sidhu, H. Singh, and M. S. Sachdev, "An artificial neural network for directional comparison relaying of transmission lines," in *Proc. 6th Intl. Conf. Developments in Power System Protection*, Nottingham, United Kingdom, Mar. 1997, pp. 282–285.

[134] M. Sanaye-Pasand and O. P. Malik, "Performance of a recurrent neural network-based power transmission line fault directional module," *Engineering Intelligent Systems for El. Engr. and Comm.*, vol. 5, no. 4, pp. 221–228, Dec. 1997.

[135] T. S. Sidhu, M. S. Sachdev, and I. Mital, "A technique for analyzing artificial neural network based protective relays," in *Proc. 7th Intl. Conf. Developments in Power System Protection*, Amsterdam, Netherlands, Apr. 2001, pp. 438–441.

[136] Z. E. Aygen, S. Seker, M. Bagriyanik, F. G. Bagriyanik, and E. Ayaz, "Fault section estimation in electrical power systems using artificial neural network approach," in *Proc. IEEE Transmission and Distribution Conf.*, New Orleans, LA, Apr. 1999, vol. 2, pp. 466–469.

[137] Z. Q. Bo, H. Y. Li, R. K. Aggarwal, A. T. Johns, and P. J. Moore, "Non-communication protection of transmission line based on genetic evolved neural network," in *Proc. 6th Intl. Conf. Developments in Power System Protection*, Nottingham, United Kingdom, Mar. 1997, pp. 291–294.

[138] T. Bi, Y. Ni, and F. F. Wu, "Advanced fault selection estimation system for power networks based on hybrid fuzzy system and radial basis function neural network," in *Proc. North American Power Symp.*, College Station, TX, Oct. 2001, pp. 99–104.

[139] T. Bi, Y. Ni, C. M. Shen, and F. F. Wu, "An on-line distributed intelligent fault section estimation system for large-scale power networks," *Electric Power Systems Research*, vol. 62, no. 3, pp. 173–182, July 2002.

[140] M. Sanaye-Pasand, H. Khorashadi-Zadeh, and O. P. Malik, "High speed accurate transmission line distance protection using ANNs," in *Proc. IEEE PES Summer Meeting*, Vancouver, Canada, July 2001, vol. 1, pp. 127–132.

[141] H.-T. Yang, W.-Y. Chang, and C.-L. Huang, "A new neural networks approach to on-line fault section estimation using information of protective relays and

circuit breakers," *IEEE Trans. Power Delivery*, vol. 9, no. 1, pp. 220–230, Jan. 1994.

[142] H.-T. Yang, W.-Y. Chang, and C.-L. Huang, "Power system distributed on-line fault section estimation using decision tree based neural nets approach," *IEEE Trans. Power Delivery*, vol. 10, no. 1, pp. 540–546, Jan. 1995.

[143] M. Togami, N. Abe, T. Kitahashi, and H. Ogawa, "On the application of a machine learning technique to fault diagnosis of power distribution lines," *IEEE Trans. Power Delivery*, vol. 10, no. 4, pp. 1927–1936, Oct. 1995.

[144] S. K. Chakravarthy, C. V. Nayar, and N. R. Achuthan, "Applying pattern recognition in distance relaying: Part 1 - Concept; Part 2 - Feasibility," *IEE Proc. C*, vol. 139, no. 4, pp. 301–314, July 1992.

[145] A. G. Jongepier and L. van der Sluis, "Adaptive distance protection of a double-circuit line using artificial neural nets," in *Proc. 5th Intl. Conf. Developments in Power System Protection*, York, United Kingdom, Mar./Apr. 1993, pp. 157–160.

[146] S. A. Khaparde, N. Warke, and S. H. Agarwal, "Adaptive realying using artificial neural network," in *Proc. 2nd Intl. Forum Applications of Neural Networks to Power Systems*, Yokohama, Japan, Apr. 1993, pp. 447–452.

[147] S. Skok and A. Marusic, "The self-organizing neural network applied to adaptive distance protection," in *Proc. IEEE PES Winter Meeting*, Singapore, Jan. 2000, vol. 3, pp. 1956–1960.

[148] Q. Y. Xuan, R. K. Aggarwal, and A. T. Johns, "Counterpropagation network based fault classification for double-circuit lines," in *Proc. 8th Mediterranean*

Electrotechnical Conf., Industrial Applications in Power Systems, Computer Science and Telecommunications, Bari, Italy, May 1996, vol. 2, pp. 657–660.

[149] S. A. Khaparde, N. Warke, and S. H. Agarwal, "An adaptive approach in distance protection using an artificial neural network," *Electric Power Systems Research*, vol. 37, no. 1, pp. 39–44, Apr. 1996.

[150] K. K. Li, L. L. Lai, and A. K. David, "Stand alone intelligent digital distance relay," *IEEE Trans. Power Systems*, vol. 15, no. 1, pp. 137–142, Feb. 2000.

[151] W. Qi, G. W. Swift, P. G. McLaren, and A. V. Castro, "An artificial neural network application to distance protection," in *Proc. Intl. Conf. Intelligent Systems Applications to Power Systems*, Orlando, FL, Jan.–Feb. 1996, pp. 226–230.

[152] W. Qi, G. W. Swift, P. McLaren, and A. Castro, "Distance protection using an artificial neural network," in *6th Intl. Conf. Developments in Power System Protection*, Nottingham, United Kingdom, Mar. 1997, pp. 286–290.

[153] W. Xiaoru, Q. Qingquan, and W. Sitao, "Applying neural networks in adaptive distance protection," in *Proc. Intl. Conf. Power System Technology*, Beijing, China, Aug. 1998, vol. 2, pp. 1158–1162.

[154] S. Skok and A. Marusic, "Distance protection of a double-circuit line based on Kohonen neural network considering different operating modes," in *Proc. IEEE Porto Power Tech*, Porto, Portugal, Sep. 2001, vol. 4, pp. 1–5.

[155] M. A. Al-shaher, M. M. Sabry, and A. S. Saleh, "Fault location in multi-ring distribution network using artificial neural network," *Electric Power System Research*, vol. 64, no. 2, pp. 87–92, Feb. 2003.

[156] Z. Chen and J.-C. Maun, "Artificial neural network approach to single-ended fault locator for transmission lines," in *Proc. IEEE Power Industry Computer Applications Conf.*, Columbus, OH, May 1997, pp. 125–131.

[157] P. K. Dash, D. P. Swain, H. P. Khincha, and A. C. Liew, "Digital protective relaying using an adaptive neural networks," in *Proc. Intl. Conf. Energy Management and Power Delivery*, Singapore, Nov. 1995, vol. 2, pp. 684–689.

[158] A. J. Mazon, I. Zamora, J. Gracia, K. J. Sagastabeitia, and J. R. Saenz, "Selecting ANN structures to find transmission faults," *IEEE Computer Applications in Power*, vol. 14, no. 3, pp. 44–48, July 2001.

[159] S.-H. Kang, K.-H. Kim, K.-R. Cho, and J.-K. Park, "High speed offset free distance relaying algorithm using mulitlayer feedforward neural networks," in *Proc. Intl. Conf. Intelligent Systems Applications to Power Systems*, Orlando, FL, Jan./Feb. 1996, pp. 210–214.

[160] B. Kasztenny, E. Rosolowski, M. Lukowicz, and J. Izykowski, "Current related relaying algorithms immune to saturation of current transformers," in *Proc. Intl. Conf. Developments in Power System Protection*, Nottingham, United Kingdom, Mar. 1997, pp. 365–368.

[161] A. J. Mazon, I. Zamora, J. F. Minambres, M. A. Zorrozua, J. J. Barandiaran, and K. Sagastabeitia, "A new approach to fault location in two-terminal transmission lines using artificial neural networks," *Electric Power Systems Research*, vol. 56, no. 3, pp. 261–266, Dec. 2000.

[162] Y. H. Song, "Accurate fault location scheme based on neural networks applied to EHV transmission systems," in *Proc. Intl. Conf. Power System Technology*, Beijing, China, Oct. 1994, vol. 2, pp. 1028–1031.

[163] D. Zhou, K. Yasuda, and R. Yokoyama, "Real time fault location in power system combining neocognitron and BP neural network," in *Proc. Intl. Conf. Power System Technology*, Beijing, China, Oct. 1994, no. 2, pp. 1032–1036.

[164] A. F. Sultan, G. W. Swift, and D. J. Fedirchuk, "Detection of high impedance arcing faults using a multi-layer perceptron," *IEEE Trans. Power Delivery*, vol. 7, no. 4, pp. 1871–1877, Oct. 1992.

[165] H.-J. Koglin and T. Lobos, "Recognition of arcing faults on transmission lines using a neural network," in *Proc. 12th Power Systems Computation Conf.*, Dresden, Germany, Aug. 1996, vol. 2, pp. 688–693.

[166] Y.-H. Pao and D. J. Sobajic, "Combined use of unsupervised and supervised learning for dynamic security assessment," *IEEE Trans. Power Systems*, vol. 7, no. 2, pp. 878–884, May 1992.

[167] S. Ebron, D. L. Lubkeman, and M. White, "A neural network approach to the detection of incipient faults on power distribution feeders," *IEEE Trans. Power Delivery*, vol. 5, no. 2, pp. 905–914, Apr. 1990.

[168] A. M. Sharat, L. A. Snider, and K. Debnath, "A neural network based back error propagation relay algorithm for distribution system high impedance fault detection," in *Proc. 2nd Intl. Conf. Advances in Power System Control, Operation and Management*, Hong Kong, Dec. 1993, vol. 2, pp. 613–620.

[169] Y Fukuyama and Y. Ueki, "Fault analysis system using neural networks and artificial intelligence," in *Proc. 2nd Intl. Forum Applications of Neural Networks to Power Systems*, Yokohoma, Japan, Apr. 1993, pp. 20–25.

[170] Y. Shimakura, J. Inagaki, S. Fukui, and S. Hori, "An artificial neural network

and knowledge-based method for reasoning causes of power network faults," in *Proc. 2nd Intl. Forum Applications of Neural Networks to Power Systems*, Yokohama, Japan, Apr. 1993, pp. 265–269.

[171] J. C. Tan, P. A. Crossley, and P. G. McLaren, "Fuzzy expert system for on-line fault diagnosis on a transmission network," in *Proc. IEEE PES Winter Meeting*, Columbus, OH, Jan./Feb. 2001, vol. 2, pp. 775–780.

[172] E. H. P. Chan, "Application of neural network computing in intelligent alarm processing," in *Proc. Power Industry Computer Application Conf.*, Seattle, WA, May 1989, pp. 246–251.

[173] T. N. Nagabhushana and H. S. Chandrasekharaiah, "Fault diagnosis of electrical power systems using incremental radial basis function nets," in *Proc. Intl. Conf. Energy Management and Power Delivery*, Singapore, Nov. 1995, vol. 2, pp. 560–564.

[174] C. Rodriguez, J. I. Martin, C. Ruiz, A. Lafuente, S. Rementeria, J. Perez, and J. Muguerza, "Fast and reliable fault analysis in complex power systems," in *Proc. 2nd Intl. Forum Applications of Neural Networks to Power Systems*, Yokohama, Japan, Apr. 1993, pp. 14–19.

[175] A. M. Gaouda, S. H. Kanoun, M. M. A. Salama, and A. Y. Chikhani, "Pattern recognition applications for power system disturbance classification," *IEEE Trans. Power Delivery*, vol. 17, no. 3, pp. 677–683, July 2002.

[176] A. K. Ghosh and D. L. Lubkeman, "The classification of power system disturbance waveforms using a neural network approach," *IEEE Trans. Power Delivery*, vol. 10, no. 1, pp. 109–115, Jan. 1995.

[177] D. L. Lubkeman, C. D. Fallon, and A. A. Girgis, "Unsupervised learning strategies for the detection and classification of transient phenomena on electric power distribution systems," in *Proc. 1st Intl. Forum Applications of Neural Networks to Power Systems*, Seattle, WA, July 1991, pp. 107–111.

[178] J. Huang, M. Negnevitsky, and D. T. Nguyen, "A neural-fuzzy classifier for recognition of power quality disturbances," *IEEE Trans. Power Delivery*, vol. 17, no. 2, pp. 609–616, Apr. 2002.

[179] *Electromagnetic Transient Program (EMTP) - Rule Book, EPRI EL 6421-1*, Electric Power Research Institute, Palo Alto, CA, June 1989.

[180] *Alternative Transients Program (ATP) - Rule Book*, CanAm EMTP User Group, Portland, OR, Sep. 1992.

[181] *Using MATLAB*, The MathWorks Inc., Natick, MA, Aug. 2002.

[182] *SimPowerSystems User's Guide*, The MathWorks Inc., Natick, MA, July 2002.

[183] *MODELS in ATP - Rule Book*, CanAm EMTP User Group, Portland, OR, Aug. 1995.

[184] *Signal Processing Toolbox User's Guide*, The MathWorks Inc., Natick, MA, Dec. 2002.

[185] *Neural Network Toolbox User's Guide*, The MathWorks Inc., Natick, MA, Jan. 2003.

[186] *Fuzzy Logic Toolbox User's Guide*, The MathWorks Inc., Natick, MA, July 2002.

[187] *Statistics Toolbox User's Guide*, The MathWorks Inc., Natick, MA, Feb. 2003.

[188] *Using Simulink*, The MathWorks Inc., Natick, MA, Sep. 2003.

[189] M. Kezunovic and Q. Chen, "A novel approach for interactive protection system simulation," *IEEE Trans. Power Delivery*, vol. 12, no. 2, pp. 668–674, Apr. 1997.

[190] J. Mahseredjian and F. Alvarado, "Creating an Electromagnetic Transient Program in MATLAB: MatEMTP," *IEEE Trans. Power Delivery*, vol. 12, no. 1, pp. 380–388, Jan. 1997.

[191] J. Mahseredjian, G. Benmouyal, X. Lombard, M. Zouiti, B. Bressac, and L. Gerin-Lajoie, "A link between EMTP and MATLAB for user-defined modeling," *IEEE Trans. Power Delivery*, vol. 13, no. 2, pp. 667–674, Apr. 1998.

[192] M. Riedmiller and H. Braun, "A direct adaptive method for faster backpropagation learning: the RPROP algorithm," in *Proc. IEEE Intl. Conf. Neural Networks*, San Francisco, CA, Mar./Apr. 1993, vol. 1, pp. 586–591.

[193] *SEL-321 Phase and Ground Distance Relay, Directional Overcurrent Relay, Fault Locator: Instruction Manual*, Schweitzer Engineering Laboratories Inc., Pullman, WA, Mar. 1998.

APPENDIX A

STP-SKY POWER SYSTEM MODEL DATA

Table VIII. External equivalent source parameters

Source number	Source voltage E		Positive-sequence impedance $Z_1(\Omega)$		Zero-sequence impedance $Z_0(\Omega)$	
	$\|E\|(V)$	$\angle E$(deg)	R_1	X_1	R_0	X_0
1	340642.65	-53.35	1.512	37.1316	2.135	41.2228
2	359296.80	-45.32	0.988	22.5195	0.381	13.2594
3	347349.45	-57.39	1.883	26.8844	1.868	15.0786
4	386993.40	-15.36	0.345 0.950	17.4962 438.0000	0.272 0.950	15.0786 438.0000
5	350889.15	-40.89	1.463	18.5016	1.626	21.5459
6	353932.05	-39.00	0.207	3.9483	0.621	5.6827
7	347418.45	-36.49	1.070	12.7048	4.495	25.2697
8	344796.45	-54.64	1.488	30.6251	3.303	40.0388
9	347287.35	-45.54	1.738	16.8539	4.975	31.0060

Table IX. Line parameters

Line	Positive-sequence impedance $Z_1(\Omega)$		Zero-sequence impedance $Z_0(\Omega)$	
	R_1	X_1	R_0	X_0
SKY-MARION	1.071	9.403	5.475	36.064
MARION-HILL	1.785	16.430	9.522	62.840
SPRUCE-SKY	0.952	12.022	11.308	41.896
SPRUCE-LHILL	7.500	68.440	66.300	218.050
DOW-WAP	3.214	33.327	21.540	96.291

Line	Length (miles)	Resistance (Ω/mile)		Reactance (Ω/mile)		Susceptance (Ω/mile)	
		R_1	R_0	ωL_1	ωL_0	ωC_1	ωC_0
STP-WAP	68.26	0.06134	0.38980	0.56640	2.05080	7.60150	3.83770
STP-HOLMAN	90.02	0.06134	0.38980	0.56640	2.05080	7.60150	3.83770
STP-LHILL	141.28	0.06134	0.38980	0.56640	2.05080	7.60150	3.83770
STP-DOW	45.39	0.03092	0.40502	0.29693	1.63593	14.73960	5.99290
STP-SKY	167.44	0.06134	0.43594	0.56640	2.00988	7.62447	4.37250
STP-HILL	178.34	0.06134	0.43594	0.56640	2.00988	7.62447	4.37250

APPENDIX B

MULTILAYER PERCEPTRON ALGORITHM

Resilient propagation (RPROP) algorithm for supervised learning in multi-layered feedforward neural networks:

```
For all weights and biases {
    if ($\frac{\partial E}{\partial \omega_{ij}}(t-1) \cdot \frac{\partial E}{\partial \omega_{ij}}(t) > 0$) then {
        $\Delta_{ij}(t) = \min(\Delta_{ij}(t-1) \cdot \eta^{+}, \Delta_{max})$
        $\Delta\omega_{ij}(t) = -\mathrm{sign}(\frac{\partial E}{\partial \omega_{ij}}(t)) \cdot \Delta_{ij}(t)$
        $\omega_{ij}(t+1) = \omega_{ij}(t) + \Delta\omega_{ij}(t)$
    }
    else if ($\frac{\partial E}{\partial \omega_{ij}}(t-1) \cdot \frac{\partial E}{\partial \omega_{ij}}(t) < 0$) then {
        $\Delta_{ij}(t) = \max(\Delta_{ij}(t-1) \cdot \eta^{-}, \Delta_{min})$
        $\Delta\omega_{ij}(t) = -\Delta\omega_{ij}(t-1)$
        $\omega_{ij}(t+1) = \omega_{ij}(t) + \Delta\omega_{ij}(t)$
        $\frac{\partial E}{\partial \omega_{ij}}(t) = 0$
    }
    if ($\frac{\partial E}{\partial \omega_{ij}}(t-1) \cdot \frac{\partial E}{\partial \omega_{ij}}(t) = 0$) then {
        $\Delta_{ij}(t) = \Delta_{ij}(t-1)$
        $\Delta\omega_{ij}(t) = -\mathrm{sign}(\frac{\partial E}{\partial \omega_{ij}}(t)) \cdot \Delta_{ij}(t)$
        $\omega_{ij}(t+1) = \omega_{ij}(t) + \Delta\omega_{ij}(t)$
    }
}
```

where E is error function, ω_{ij} is weight from neuron j to neuron i, Δ_{ij} is individual update value for each weight, η^{-} is decreasing factor and η^{+} is increasing factor.

APPENDIX C

DISTANCE RELAY ALGORITHM

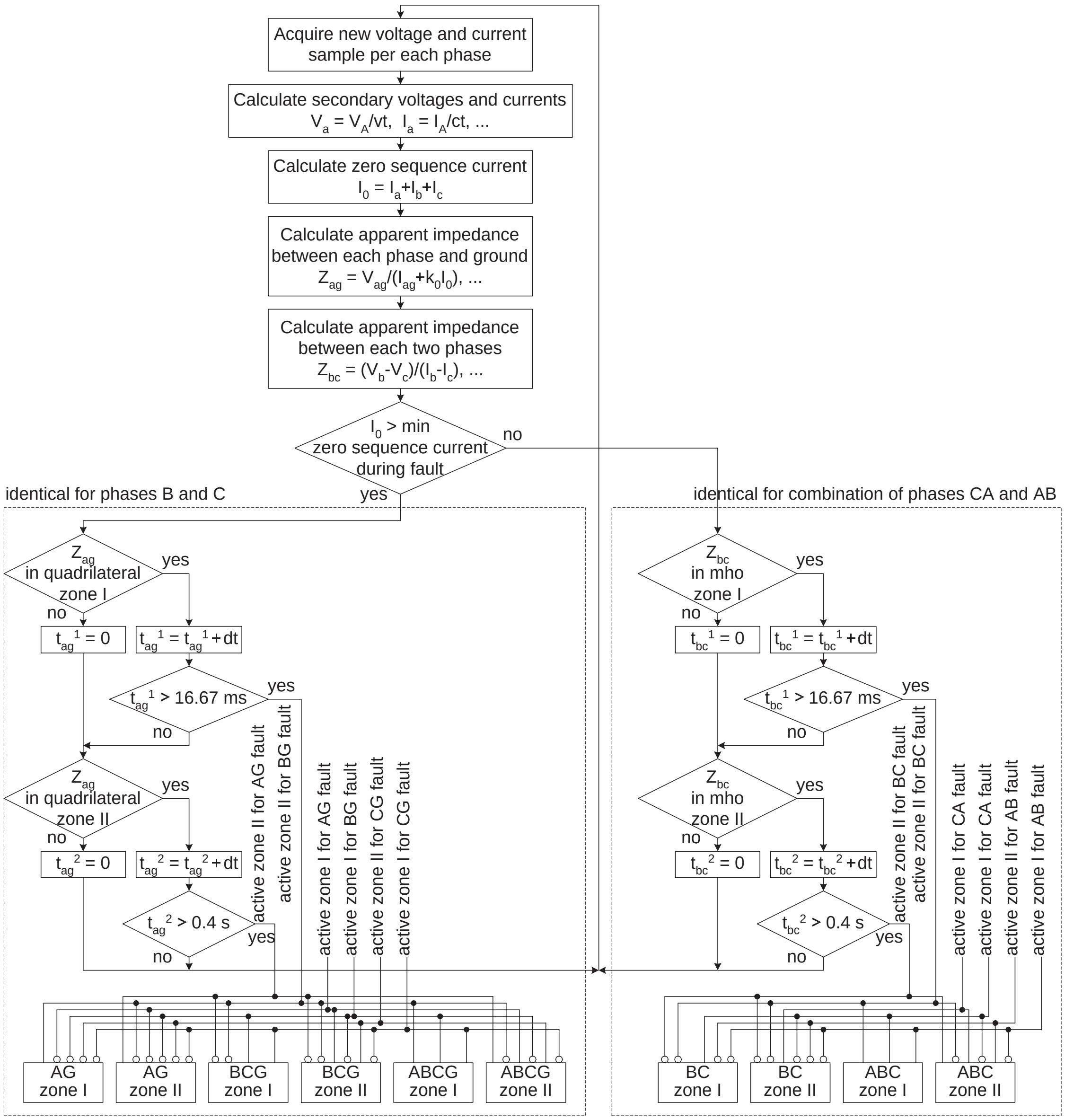

Acquire new voltage and current sample per each phase
Calculate secondary voltages and currents $V_a = V_A/vt$, $I_a = I_A/ct$, ...
Calculate zero sequence current $I_0 = I_a+I_b+I_c$
Calculate apparent impedance between each phase and ground $Z_{ag} = V_{ag}/(I_{ag}+k_0I_0)$, ...
Calculate apparent impedance between each two phases $Z_{bc} = (V_b-V_c)/(I_b-I_c)$, ...
I_0 > min zero sequence current during fault
no
yes
identical for phases B and C
identical for combination of phases CA and AB
Z_{ag} in quadrilateral zone I
yes
no
$t_{ag}^1 = 0$
$t_{ag}^1 = t_{ag}^1+dt$
$t_{ag}^1 > 16.67$ ms
yes
no
Z_{ag} in quadrilateral zone II
yes
no
$t_{ag}^2 = 0$
$t_{ag}^2 = t_{ag}^2+dt$
$t_{ag}^2 > 0.4$ s
yes
no
active zone II for AG fault
active zone II for BG fault
active zone I for AG fault
active zone I for BG fault
active zone II for CG fault
active zone I for CG fault
AG zone I
AG zone II
BCG zone I
BCG zone II
ABCG zone I
ABCG zone II
Z_{bc} in mho zone I
yes
no
$t_{bc}^1 = 0$
$t_{bc}^1 = t_{bc}^1+dt$
$t_{bc}^1 > 16.67$ ms
yes
no
Z_{bc} in mho zone II
yes
no
$t_{bc}^2 = 0$
$t_{bc}^2 = t_{bc}^2+dt$
$t_{bc}^2 > 0.4$ s
yes
no
active zone II for BC fault
active zone II for BC fault
active zone I for CA fault
active zone I for CA fault
active zone II for AB fault
active zone I for AB fault
BC zone I
BC zone II
ABC zone I
ABC zone II

Fig. 50. Impedance based distance relay algorithm

VITA

Slavko Vasilic received his B.S. and M.S. degrees in electrical engineering from University of Belgrade in 1993 and 1999, respectively.

Since January 2000, he has been a graduate research assistant in the Department of Electrical Engineering, working on artificial intelligence and signal processing applications for power system analysis, control and protection. From 1994 to 1999 he was a research assistant in the Department of Electrical Engineering, University of Belgrade, working on the development of robust multivariable control algorithms and their implementation for power electronics devices.

His industrial experience includes Energoproject-Hydroengineering, Belgrade, Serbia from 1994 to 1999 as a systems engineer for hardware configuration and software applications for monitoring, control and supervisory systems for power generation and transmission. His previous industrial experience includes working in the Technical Overhauling Department, Cacak, Serbia from 1993 to 1994 as a design engineer for object targeting and fire control systems.

His research interests are signal processing, modern control systems, pattern recognition, numerical modeling and optimization, parameter and state estimation, system identification, machine learning, neural networks, fuzzy logic and statistical analysis.

His permanent address is Dr. Misovica 157, 32000 Cacak, Serbia.