Knowledge Based Fault Detection and Identification for High-Voltage Power Systems

by

Xiuping Xu

A Thesis Submitted to the Faculty of Graduate Studies
in Partial Fulfilment of the Requirements for the Degree of

Master of Science
in Computer Engineering

Department of Electrical and Computer Engineering
University of Manitoba
Winnipeg, Manitoba R3T 5V6

0-612-80093-8

THE UNIVERSITY OF MANITOBA

FACULTY OF GRADUATE STUDIES
*****
COPYRIGHT PERMISSION

KNOWLEDGE BASED FAULT DETECTION AND IDENTIFICATION FOR HIGH-VOLTAGE POWER SYSTEMS

BY

XIUPING XU

A Thesis/Practicum submitted to the Faculty of Graduate Studies of The University of Manitoba in partial fulfillment of the requirement of the degree

of

MASTER OF SCIENCE

ABSTRACT

This thesis introduces a Knowledge Based approach for High-Voltage power system faults detection and identification. Based on the feature of the typical signals obtained from the Transcan Recording System (TRS), a dual approach is pursued. Feature extraction is central to this thesis. Various features of power system signals are extracted to provide a basis for a decision support system for power system fault and identification. First, faults that have periodic signals such as phase current and 6 pulse signals, and Valve currents are analyzed using FFT and auto-correlation to identify the type of the waveform of the input signal. Second, for faults that have non-periodic signal such as pole line voltage, pole current and pole current order, a new method called Fuzzy Wavelet Analysis is introduced to determine the type of the faults. In addition, there are also some other attributes like the Ratio of Phase current and current order, ac Phase voltages Error that are analyzed using granular computing methods. Finally, we use the above attributes to set up a decision table and then use Rough Set rule generation tool called Rosetta to generate fault-classification decision rules. Performance evaluation of detectability and identifiability are defined to assist in assessing the performance that is achieved through a learning mechanism based on the detectability and identifiability measures.

Keywords: fault detection, fault identification, Fourier analysis, granulation, power system fault, rough set theory, signal analysis, wavelet analysis.

ACKNOWLEDGEMENTS

Many thanks to my supervisor Dr. J. F. Peters for his generous help and sharing his knowledge. He gave me efficient direction for the thesis.

Thanks to Liting Han for her help in this project.

Thanks for my parents and sisters' support.

Table of Contents

List of Tables

List of Figures

Nomenclature

Notation	Explanation	Section
$\underline{B}X$	lower approximation of set X relative to attributes in set B	1.1.5
$\bar{B}X$	upper approximation of set X relative to attributes in set B	1.1.5
$\mu_X^B(x)$	rough membership function evaluated at x $\in$ X relative to the set of attributes B.	1.1.8
$S(f)=\int_{-\infty}^{\infty} s(t)e^{-j2\pi ft}dt$	continuous Fourier transform	1.1.3
$Ind_A(B)$	For each $B \subseteq A$, it is associated an equivalence relation	1.1.5
c_{ij}	n x n matrix (c_{ij}) called the discernibility matrix M of S (denoted M(DT))	1.1.6
OPT(B)	For an information system S, the set of decision rules constructed with respect to a reduct R is denoted OPT(S, R)	1.1.6

1 INTRODUCTION

It is well-known that analysis and classification of power system disturbances are helpful in working towards more stability and efficiency in power delivery [4]. Recognition of a power system fault (result of some form of disturbance that causes an electrical system to have abberant behavior) can be compensated to avert system failure by switching transmission lines to supply additional current (response to increased load) or switching capacitor banks to balance increased loads. The focus of this thesis is an introduction to methodologies that can be used in classifying power system faults.

This chapter briefly presents the basic ideas underlying this thesis. The chapter is organized as follows. In Section 1.1, a brief presentation of the basic terminology used in this thesis is presented, namely, terminology from power systems, power system faults, selected transform techniques used in signal analysis (Fourier and wavelet transforms), fuzzy set theory, rough set theory, attribute reduction, decision rules, discretization, and rough membership functions. Also included in Section 1.1 is illustration of how one goes about applying rough set methods in the context of power system faults. An overview of the thesis is given in Section 1.2. The development of knowledge-based algorithms for fault detection is briefly described in Section 1.3. In the final Section of this chapter, the topical-coverage (scope) of this thesis is given.

1.1 Basic Terminology

This section briefly presents the basic terminology for this thesis.

1.1.1 Power Systems

In power electronics, a valve (also called an ideal valve) is a diode, thyristor or turn-off valve **value**. In power systems, valves are simply regarded as switches **[1]**. Thyristors were introduced in the late 1950s, and is of interest in this thesis because they are used at the Manitoba Hydro Dorsey Station. Basically, a thyristor (also called a silicon-controlled rectifier) is a four-layer, three-junction device. It has three terminals: anode, cathode and gate. This device is turned on by applying a short pulse across gate and cathode. Once the gate turns on, the gate loses its ability to turn off the device. The turnoff is achieved by applying a reverse voltage across anode and cathode. There are two types of thyristors: converter grade and inverter grade. Converter grade thyristors are used in commutation (i.e., phase-controlled) applications like high-voltage dc transmission. Inverter-grade thyristors are used in commutation applications such as dc-ac inverters. Thyristors with up to 5 KV and 3

KA capacity are available. In power system, a pole or a valve group consists of 6 vales. The pole current is the summation of each valve current or we call it phase current. Phase current is generated by opening the valve in a valve group, a positive pulse is used to open the valve. Counter is used to generate the pulse. As shown in Figure 1.1 **[28]**

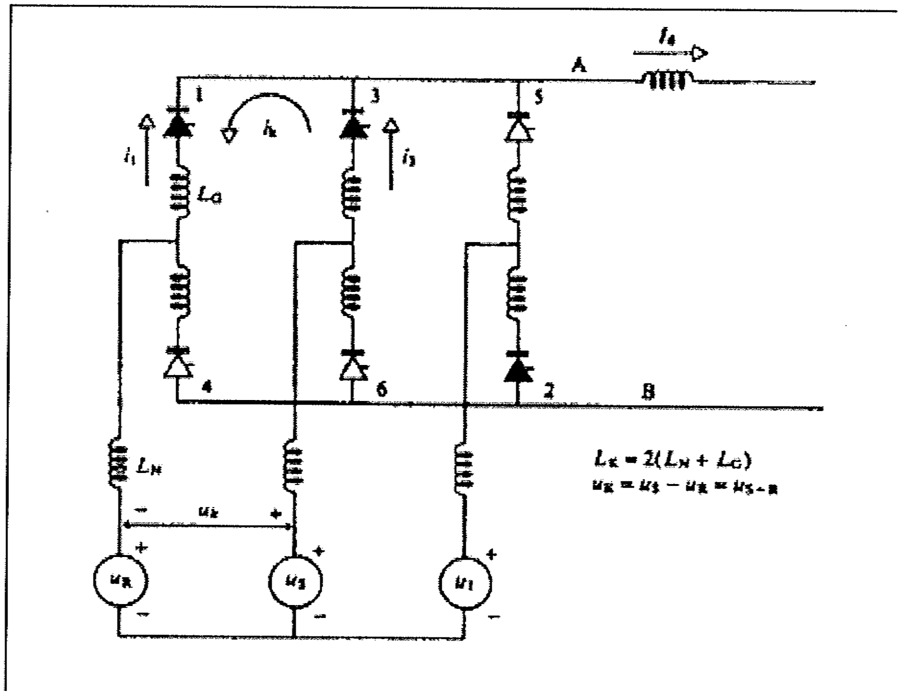


Figure 1.1 The Valve Group Diagram [28]

1.1.2 Power System Faults

A power system fault is the result of an electrical disturbance. A number of power system faults are referenced in this thesis, namely, Ring Counter Error, ac Filter Bank, 500kvClose, ac Voltage Disturbance, Pole-line Voltage Flashover, Pole-line Voltage Force Retard, Valve Asymmetrical Protection, dc line disturbance, Valve Commutation Failure and Valve Current Blip. With a Fault Ring Counter Error, the six valves are not opened in a specific designed sequence then the phase current and the pole current will sharply increase or decrease. As a result of a Fault AC Filter Bank Test, before the ac power is output for use, the power should be filtered for noise compression and cutoff the undesired frequency components. If the filter does not work well, it will cause the ac voltage energy loss or phase mix up. With a Fault 500 kv Close, the dc line is completely shut down. With a Fault AC Voltage Disturbance, the ac voltage line will be affected by the different causes such as an object (e.g., falling tree) hits the transmission line, heavy snowfall or severe wind, and sometimes radiation or magnetic field interference. With a Fault Pole-line

Voltage Flash Over, the pole voltage usually should remain at 450kv. But sometimes it will oscillate quickly. We call this event pole line voltage flash over. With a Fault Pole-line Voltage Force Retard, if the energy of a dc line decreases, then the pole line voltage will decrease slowly. Sometimes, the dc power system will restart in a short time if the control system responds quickly but usually the dc system will shut down for a long period. With a Fault Valve Asymmetrical Protection, if the pulse to open the valve arrives in abnormal sequence, this can cause more than two valves to open at the same time. Then the circuit control system will force one of the valves to close. With a DC line disturb Disturbance in the power system of Manitoba Dorsay station, the ac voltage is converted from the dc voltage. For the long distance transmission of dc voltage is easier and the interference problem can be decreased a lot. But sometimes, the dc line will be affected such as snow on the transmission line or in windy weather. With a Fault Valve Commutation Failure, sometimes although the pulse to open the valve arrives, the valve still not communicate correctly. Usually this occurs in all valves in one pole. With a Fault Valve Current Blip, sometimes only one valve in a pole increases sharply for a short period and then shuts down. This type of power system fault is caused by a short to ground.

1.1.3 Transform Techniques

Fourier methods such as the Fourier series and Fourier integral are used in analyzing continuous time signals. That is, Fourier methods are applicable in systems where there is a characteristic signal s(t) defined for all values of t in the interval [-∞,∞]. A Fourier transform decomposes a waveform into a sum of sinusoids of different frequencies [3]. The signal s(t) in the time domain is decomposed into the sum of its sinusoids S(f) in the frequency domain using the formula (1.1).

$$S(f) = \int_{-\infty}^{\infty} s(t) e^{-j2\pi ft} dt \qquad (1.1)$$

where $j = \sqrt{-1}$. In this thesis, the focus is on the application of what is known as the discrete Fourier transform that is applicable to discrete-time signals. A discrete time signal s[n] is defined for values of n in the interval [-∞,∞]. A discrete Fourier transform (DFT) is used in studying finite collections of sampled data $\{s_0, \ldots, s_{N-1}\}$ relative to the sequence $\{S_0, \ldots, S_{N-1}\}$. The DFT is computed using (1.2).

$$S_n = \sum_{k=0}^{N-1} s_0(k) e^{-j\frac{2\pi}{N}nk}, n = 0, 1, \ldots, N-1 \qquad (1.2)$$

A "fast" Fourier transform (FFT) results from the application of a particular algorithm that can compute the DFT more rapidly that other available algorithms [3].

Transient signals in a power system are non-stationary, time-varying voltage and current signals. These signals result from disturbances (faults) on transmission lines (e.g., capacitor switching, lightning strikes, short circuits) [4]. Wavelet transforms provide efficient, local analysis of non-stationary, fast transient signals. Waveform data are captured by digital transient recorders such as the Transcan Recording System used by the Manitoba Hydro Dorsey Station. The wavelet transform of an integrable function f(t) is a decomposition of f(t) into a set of basic functions denoted by $h_{s,\tau}(t)$ called wavelets. The wavelet transform is given in (1.3).

$$W_f(s,\tau) = \int s(t) h^*_{s,\tau}(t)dt \qquad (1.3)$$

where * denotes the complex conjugate, and the wavelet $h_{s,\tau}(t)$ is computed using (1.4).

$$h_{s,\tau}(t) = \frac{1}{\sqrt{s}} h\left(\frac{t-\tau}{s}\right) \qquad (1.4)$$

Where s is a scale factor, and τ is a translation factor.

1.1.4 Fuzzy Set Theory

Fuzzy set theory is concerned with granulating experimental data (i.e., identifying clusterings of data (also called information granules), approximate distributions of data values within each identified cluster, determining the degree-of-membership of each observation in a distribution). A fuzzy set itself is a pair (μ, X), where μ: X → [0,1] (degree-of-membership function) and X is a non-empty set representing domain knowledge. An information granule is defined to be a clump of objects (points) drawn together by indistinguishability, similarity, or functionality [5]. In this research, fuzzy set theory provides a convenient means of organizing and analyzing the data from power system fault files.

1.1.5 Rough Set Theory

Rough set theory offers a systematic approach to set approximation [6], and is part of an ongoing effort to use rough set methods in classifying experimental data [14]-[30]. To begin, let $S = (U, A)$ be an information system where U is a non-empty, finite set of objects and A is a non-empty, finite set of attributes, where $a{:}U \rightarrow V_a$ for every a A. For each $B \subseteq A$, there is associated an equivalence relation $\mathrm{Ind}_A(B)$ such that

$$\mathrm{Ind}_A(B) = \{(x,x') \in U^2 \mid \forall a \in B.\, \mathrm{a(x)} = \mathrm{a(x')}\} \qquad (1.5)$$

If $(x, x') \in \mathrm{Ind}_A(B)$, we say that objects x and x' are indiscernible from each other relative to attributes from B. The notation $[x]_B$ denotes equivalence classes of $\mathrm{Ind}_A(B)$. Further, partition $\mathrm{U}/\mathrm{Ind}_A(B)$ denotes the family of all equivalence classes of relation $\mathrm{Ind}_A(B)$ on U. For $X \subseteq U$, the set X can be approximated only from information contained in B by constructing a B-lower and B-upper approximation denoted by $\underline{B}X$ and $\bar{B}X$ respectively, where $\underline{B}X = \{x \mid [x]_B \subseteq X\}$ and $\bar{B}X = \{x \mid [x]_B \cap X \neq \varnothing\}$. The notation $\mathrm{POS_B(X)} = \underline{B}X$ denotes what is known as the positive region (the collection of objects that can be classified with full certainty as members of X using the knowledge represented by attributes in B.

1.1.6 Attribute Reduction and Decision Rules

An approach to finding a subset of attributes (reduct) with the same classificatory power as the entire set of attributes in an information system is briefly described in this section. This leads to a brief discussion about the derivation of decision rules with minimal descriptions in their left-hand sides. In deriving decision system rules, the discernibility matrix and discernibility function are essential. Given a decision table DT = (U, A $\cup$ {d}), the n x n matrix (c_{ij}) called the discernibility matrix M of S (denoted M(DT)) is defined in (1.6).

$$c_{ij} = \{a \in A: a(x_i) \quad a(x_j)\}, \text{ for } i, j = 1, \ldots, n. \qquad (1.6)$$

A discernibility function f_{DT} relative to discernibility matrix M for a decision table DT is a boolean function of m boolean variables $a_1^{*}, \ldots, a_m^{*}$ corresponding to attributes $a_1, \ldots, a_m$ respectively, and defined in (1.7).

$$f_{DT}\left(a_1^*, \ldots, a_m^*\right) =_{df} \wedge\left\{\vee_{ij}^* \mid 1 \leq j \leq i \leq n,\ c_{ij} \neq \varnothing\right\}, \text{ where } c_{ij}^*\left\{a^* \mid a \in c_{ij}\right\} \qquad (1.7)$$

The set of all prime implicants of f_S determines the set of all reducts of S [7]. A reduct is a minimal set of attributes B $\subseteq$ A that can be used to discern all objects obtainable by all of the attributes of an information system [8]. The reducts of an information system S correspond to the prime implicants of the discernibility function f_S [9]. That is, $\mathrm{Ind_S(B)} = \mathrm{Ind_S(A)}$. In effect, a reduct is a subset B of attributes A of information system S that preserves the partitioning of the universe U. Hence, a reduct can be used to perform the same classifications as the whole attribute set A of the information [7]. The set of all reducts of S is denoted by RED(S). Let B $\subseteq$ A. The set of all reducts in IS with attribute set B is denoted by RED(B). A method used to find a proper subset of attributes of A with the classificatory power as the entire set A has been termed *attribute reduction* [8]. Let f_M^d be a decision-relative discernibility function with respect to discernibility matrix M and decision table DT. This boolean function can be constructed from the discernibility matrix for S_d. The set of all prime implicants of f_M^d defines the set of all decision-relative reducts of the decision system S_d [10].

In other words, precise conditions for decision rules can be extracted from f_M^d derived from a discernibility matrix M as in [10]. For the decision system S_d, let $\wp(V_a)$ denote the power set of V_a, where V_a is the value set of a. For every d $\in$A - B, a decision function $d_d^B : U \rightarrow \wp(V_a)$ is defined in (1.8).

$$d_d^B(u) = \{v \in V_d \mid \exists u' \in U,\ B \subseteq A,\ (u', u) \in Ind_S(B),\ d(u') = v\} \qquad (1.8)$$

In other words, d_d^B(u) is the set of all elements of the decision column of S such that the corresponding object is a member of the same equivalence class as argument u. The next step is to determine a decision rule with a minimal number of descriptors on the left-hand side. Pairs (a, v), where a $\in$ A, v $\in$ V are called *descriptors.* A decision rule over the set of attributes A and values V is an expression of the form given in (1.9).

$$a_{i_1}(u_i) = v_{i_1} \wedge \ldots \wedge a_{i_j}(u_i) = v_{i_j} \wedge \ldots \wedge a_{i_r}(u_i) = v_{i_r} \underset{S}{\Rightarrow} d(u_i) = v \qquad (1.9)$$

where $u_i \in U$, $v_{i_j} \in V_{a_{i_j}}$, v $\in$ V_d, j = 1,...,r and r |A|. The fact that a rule is true is indicated by writing it in the form given in (1.10).

$$(a_{i_1} = v_{i_1}) \wedge \ldots \wedge (a_{i_r} = v_{i_r}) \underset{S}{\Rightarrow} (a_p = v_p) \qquad (1.10)$$

For an information system S, the set of decision rules constructed with respect to a reduct R is denoted OPT(S, R). Then the set of all decision rules derivable from reducts in RED(S) is the set in (1.11).

$$\text{OPT(S)} = \cup\{\ \text{OPT(S, R)} \mid \text{R} \in \text{RED(S)}\ \} \qquad (1.11)$$

Let S_d be a decision system with condition and decision attribute A = C $\cup$ {d} for a given set of condition attributes B $\subseteq$ C. Then define a positive region $POS_B(d)$ relative to $Ind_Q(d)$ as .

$$POS_B(D) = \bigcup\{\underline{B}X \mid X \in Ind_Q(D)\}$$

The positive region $POS_B(D)$ contains all objects in the universe U that can be classified into distinct decision classes defined by $Ind_Q(D)$. The notation $X_{S_d}(u) = \{x \in \underline{B}X \mid d(x) = d(u)\}$ denotes a decision class for any u ∈ U.

1.1.7 Discretization

Suppose that we need to obtain approximate knowledge of a continuum (e.g., behavior of a sensor signal over an interval of time) by considering parts of the continuum. Discretization of a continuum entails partition a particular interval into subintervals of reals. For example, consider the interval of reals $V_a = [v_a, w_a]$ for values of an attribute $a \in A$ in a consistent decision system $S_d = (U, A \cup \{d\})$. Discretization of V_a entails searching for a partition P_a of V_a (i.e., discovering a partition of the value sets of conditional attributes into intervals). In rough set theory, discretization leads to partitions of value sets so that if the name of the interval containing an arbitrary object is substituted for any object instead of its original value in Sd, a consistent decision system is also obtained.

1.1.8 Rough Membership Function

In this section, the traditional rough membership function introduced in [11]. A rough membership function (rm function) makes it possible to measure the degree that any specified object with given attribute values belongs to a given set X. This function μ_X^B is defined relative to a set of attributes $B \subseteq A$ in information system S = (U, A) and a given set of objects X. The equivalence class $[x]_B$ induces a partition of the universe. Let $B \subseteq A$, and let X be a set of observations of interest. The degree of overlap between X and $[x]_B$ containing x can be quantified with the rough membership function in (1.12).

$$\mu_X^B : U \to [0,1] \quad \textit{defined by} \quad \mu_X^B(x) = \frac{\left|[x]_B \cap X\right|}{\left|[x]_B\right|} \tag{1.12}$$

1.1.9 Example: Discretized Rules

In a high voltage direct current (dc) transmission system, a dc line is connected between two alternating current (ac) systems as shown in Fig. 1. Such a system has two ac converters. Converters (combinations of transformers and mercury-arc valves) are at both ends of the transmission system in Fig. 1. In the case where the flow of power is from the ac side to the dc side as in Fig. 1, then a converter acts as a rectifier in changing ac to dc. The inverter in Fig. 1 converts dc to ac. The Dorsey Station in the Manitoba Hydro system, for example, acts an inverter in converting dc to ac, which is distributed throughout the Midwest.

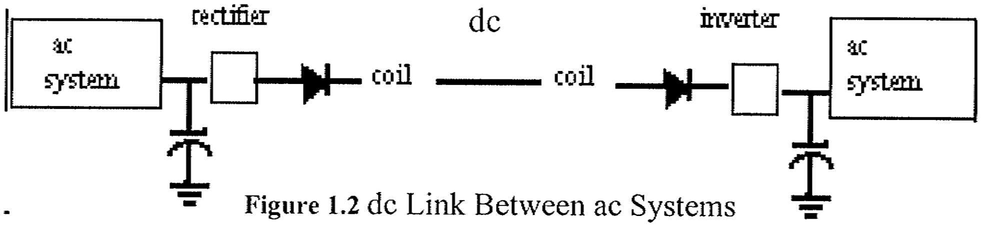


Figure 1.2 dc Link Between ac Systems

Power system faults are recorded in files. In the Manitoba Hydro Dorsay Station, a Transcan Recording system will automatically record all the status of those 27 signal into a data file whenever a fault occurs. We call this data file *.x01 file. For example, Fig. 2 shows 27 signals in a fault file which has recorded a valve ring counter error. Those 27 signals can be classified into two types: global signals which controls all signals in a valve and valve signals.

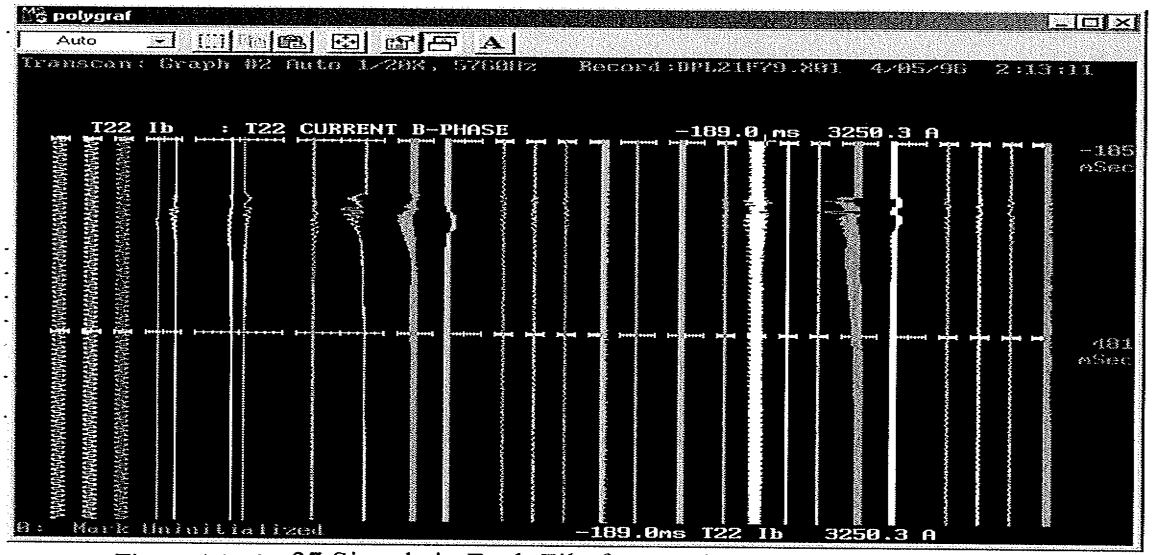


Figure 1.3 The 27 Signals in Fault File for a Valve Ring Counter Error

From left to right in the above graph, the first 9 signals are global signals, and the remaining 18 signals are valve signals:

global signals = {AC phase A, B, and C voltages, pole current order, pole Alpha order, pole 1 {2) current, pole 1{2) voltage}

group 11, 12, 13 value signals = {6 pulse voltage, Alpha response, valve current (A {B, C} phase), start pulse}

For simplicity, we illustrate the classification of the waveforms of transmission system faults relative to valve commutation failure (i.e., failure to transfer of current from one circuit to another correctly). Sometimes when the pulse to open a valve arrives, the valve fails to communicate current correctly and a commutation failure occurs. A decision (d) to classify a waveform for a power transmission fault as a commutation failure depends on an assessment of phase current (pc), current setting (cs), maximum phase current (max pc), ac voltage error (acve), pole line voltage (plvw) and phase current (pcw) waveforms. A sample commutation failure decision table is given next. In Table 1.1, d = 1 {0} indicates a fault representing {not representing} a commutation failure.

Table 1.1 Commutation Failure Decision Table

	Acve	pc/cs	plvw	pcw	cs	max pc	d
file1	0.059	0.0697	0	0.01875	0	0	0
file2	0.059	0.0697	0	0.01875	0.1667	0	0
file3	0.059	0.0697	1	0.01875	0.1667	0.0856	1
file4	0.059	0.0697	0.5	0.01875	0.054	0.0856	1
file5	0.059	0	0	0	0	0	0
file6	0.059	0	0.5	0	0	0	0
file7	0.059	0.0697	0.5	0.01875	0.054	0.0856	0
file8	0.059	0.0697	1	0.01875	0.1667	0.0956	1

Signal data needed to construct the condition granules in Table 1 come from files specified in column 1 of the table. Rosetta is a public domain toolset that makes it possible to derive a reduced set of decision rules based reducts and discretization (see http://www.idi.ntnu.no/~aleks/rosetta/). The notation max-pc[*, 0.043], for example, specifies that the maximum phase current in contained in the interval (-∞, 0.043), i.e., -∞ < max-pc < 0.043. Sample discretized rules derived from Table 1 using Rosetta are given in (1.13).

(1.13)

plvw([*, 0.750)) AND cs([0.111, *)) AND max-pc([*, 0.043)) => d(no)
plvw([0.750, *)) AND cs([0.111, *)) AND max-pc([0.043, *)) => d(yes)

Let PLF denote a pole line fault in a high voltage power system. The set F = {x | PLF(x) = yes} consists of pole line fault readings which are judged to be commutation failures. Notice that there is some uncertainty concerning the waveform represented by file4 and file7 (yes/no decision values in Table 1). Let A be the set of attributes represented in Table 1. Then from Table 1, we obtain approximation regions $\underline{A}F = \{\{file3, file8\}\}$, $\overline{A}F = \{file3, file4, file8\}$, and boundary region $BF_A(F) = \{file4\}$, The classification PLF represented by the decision column labeled d in Table 1 is rough, since the boundary region is not empty. Next, we relax the requirement that a rm function be defined for equivalence classes relative to IND, and consider approximation regions such as $\underline{A}F$ and $\overline{A}F$. Assume that [f]$_A$ = {equivalence class consisting of files with the same outputs of attribute B in the universal fault file space} = {file4, file7, file9, file10}, which includes files not considered in Table 1. Then consider, for example, the degree of overlap between $\overline{A}F$ and [f]$_A$ (see Fig. 3).

$$\mu^{A}_{\overline{AF}}(f) = \frac{\left|\overline{AF} \cap [f]_A\right|}{\left|[f]_A\right|} = \frac{1}{3}$$

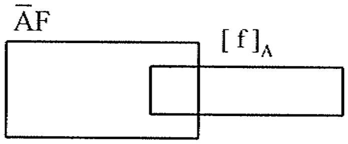


Figure 1.4 The Degree of Overlap

1.2 Overview

Manitoba Hydro Dorsey Station currently uses the Transcan Recording System (TRS) as a tool to record and monitor the HVDC Converter Station and related power system. This system has the ability to perform fairly intelligent signal processing and can be configured to operate at either Auto-call mode or Auto-poll mode to record the data. The Auto-call mode enables a recorder subsystem, i.e., a remote recorder is enabled to call the master automatically whenever a fault **occurs**. The Auto-poll mode enables the master to poll the subsystem according to the requirements of the user to check if there is any fault during a certain period of time. This thesis introduces possible algorithms that can be embedded in the TRS to give the system the capability to classify the type of the faults that occur and also to give preliminary assessment of the possible cause of certain types of faults.

Fault diagnosis, or equally fault detection and identification (FDI), is a mature field with contributions ranging from model-based techniques to data-driven configurations that capitalize upon soft computer and other "intelligent" technologies [1][2]. Recently some strategic issues and approaches about fault detection and identification have been addressed by several investigators [12, 19, 23, 27, 28].

In this project, a fault model was set up as a database to store the rules of different types of faults. These features are obtained through digital signal processing (DSP) and feature extraction methods on the data from the TRS. In many practical situation, uncertainty about the nature of a detected fault can hamper decision-making of a plant engineering and as a result, perhaps, affect the performance of the system significantly. This realization provides the motivation for a possible application of a number of classical as well new technologies (fuzzy logic, rough set theory) in the analysis of faults identified by the TRS. The new fault classification system has the ability to directly describe the potential faults. In this new system, the fast Fourier transform (FFT) low-pass filters construct and one type of wavelets are combined with degree-of-membership functions from fuzzy set theory [3] in decision tables from rough set theory [insert ref.] that are used in classifying the type of the fault. A major innovation in the proposed work relates to the utility of wavelets, in a fuzzy logic rule base setting, for fault-classification purposes.

1.3 The Development of Knowledge based algorithms for fault Detection

Fault detection and identification (FDI) is of interest in a wide variety of applications such as power system, control system, image analysis, analysis of radar signals, smart sensors, texture analysis, medicine, and industry. Typically, an FDI system entails the following components:

- Monitoring and reporting the presence of faults or failures. A power system fault is associated with some form of electrical disturbance (e.g., sudden increase in load or sudden increase in reactance of a circuit) that affects the stability of a power system [see Kimbark, 5]. If not interrupted quickly, fault current can severly damage conductors and equipment [see Broadwater, 1167]. In case of abnormalities in the system under observation, possible faults are not only reported but also verified by additional processing.
- Classifying faults. The FDI algorithm decides upon the type of the faults including no fault condition.
- Identifying the origin of a detected fault. This includes differentiating between a system failure and a functional failure. A system failure is a degradation of performance of the hardware of the system while a functional failure refers to a condition of the system state variable resulting in an unwanted operating mode such as instability, and so on. Many functional faults can eventually lead to system failure.

With the availability of powerful computing platforms, feature processing in classification theory has become an important part of many applications. Intelligent processing tools like fuzzy logic, neural networks and optimization techniques aim at accommodating large grain uncertainty while utilizing all available information in classifying observed behavior patterns (waveforms) of an electrical system. Due to the wide range of time constants, some of the waveforms belong to the same type but because of the analysis only in time or frequency domain alone is not sufficient to capture features (e.g, when two waveforms actually belong to the same type of waveform, but they have a phase shift in time domain). It turns out that the distortion is still not zero and sometimes the distortions are even bigger than comparing with two different waveform. That leads to the mistaken classification of the fault type. So that is why we translate data into the frequency domain to calculate the correlation.

1.4 Scope of this Thesis

In this thesis, the acronyms WT1, WT2 denote the wavelet translate and fuzzy computing output of the 2 pole line voltage. The acronym FFT denotes fast Fourier transform. Briefly, a FFT is used to decompose a waveform into a sum of sinusoids of different frequencies [3]. The signal in the time domain is decomposed into the sum of its sinusoids in the frequency domain using the formula (1). FFT6P denotes the FFT for the 6 pulse voltages (3 poles FFT6P1,FFT6P2,FFT6P3) The acronym Error denotes the ac voltage phase disturbance; it is calculated by phase shifting and error calculation This research also employs wavelets. Briefly, a wavelet is a family of signals, where signals are scaled by a single function called Mother wavelet. Wavelets are useful in power system fault classification because for most of the signals in power system like current and voltage, the frequency is usually 60 HZ, and for the constant signals they only have sharp oscillations when there is fault happened. Wavelet transformation and coefficient analysis are used in this study. The term wavelet transformation means decompose the signals into a sum of wavelet family with different scales and translation factors. (see details in Chapter 4.1.1)
In sum, this thesis treats the following topics as part of a study of power system faults.

- Data Discovery and Preprocessing on the data from TRS (real-time signals, unreadble)
- Wavelet Analysis on Pole-line Voltages (WT1,WT2)
- FFT and IFFT, Low-pass filter for 6 Pulse Voltages(FFT6p1,FFT6p2,FFT6p3)
- Ratio of Phase Current and Current Order(Ratio)
- Distortion of AC Phase Voltages (Error)
  The algorithms used to get the above attributes are:
- Wavelet Transformation and it's Coefficient Analysis to get the feature of the wavelet
- FFT , IFFT and Low-pass filter
- Fuzzy set theory and granular computing
- Error detection
- Rough Set theory and its application in classifying power system faults.

2 PREPROCESSING

Before we do the further processing on the data of the signals we have to do some preprocessing and also find out the characteristics for the different signals. In preprocessing, we first recovery the data from binary to ASCII (American Standard Code for Information Interchange) format, and then separate the signals into different group based on the physical type of them (Current, Voltage, Current Order). After that we have to do some reduction on the data to get the part that is really useful for feature extraction. In section 2.2 we will introduce some characteristics of the signals, based on the characteristics we select different processes to get the features.

2.1 Preprocessing

In this section we will introduce the 3 steps for signals preprocessing, Data recovery, Signal separation and Information deduction.

2.1.1 Data Recovery

When we get the data from the TRS, they are binary format and compressed as *.x01 files. Those files are unreadable to us. That means, from the original data we can not figure out the information it is carrying. But together with the .x01 files TRS also provide us the *.scf files. The *.scf file give us the information about how to recovery the data. Selected sample lines from such a file are given next

It tells us how many channels have been scanned. For instance, the recovered file has 48 analog and 4 digital channels. The *.scf file also indicates that the scanning order and the physical name of each channel. In the .x01 file, the first 52 binaries are used for the file name and date. The following 8 binaries are used for sampled data and the last 4 bits are used to indicate the channel number.

A C++ program was designed by Liting Han to recover the binary format data (.x01) into ASCII format data which is readable (*.dat) [23]. This program has been optimized for to complete the work for this thesis. One .x01 file can be recovered into 48 *.dat files. Among these files, 27 are used to represent the 27 signals in the power system. From the 27 signals, we can figure out the characteristics of different types of faults and determine the possible cause of that.

2.1.2 Signal Separation based on the type of the signal

In order to set up the fault model for the system, we used 56 *.x01 files as the raw data and do the processing on them to find the rules, and then use 25 another *.x01 files to do the training and testing. The training and testing part is used to modify the system and also detect whether the whole system is reliable. After we recovery the data into readable format, they are separated into different groups according to the physical properties of the signals. We can distinguish the signals from their number in names. For example, usually the file with the name dpl12*101.dat is phase current signal, dpl22*51.dat represent the 6 pulse voltage. After the separation of the data, further operation can be done on the different type of data.

2.1.3 Information Deduction

In each of the files, there are a lot of data to describe the signal. For example, for each signal of phase current there are over 7000 data. This can lead excessively processing on all the data. What interested us most is the part of the signal data where faults happened. Hence, we need to monitor the signal and find out the location where a fault starts and also the location of the end of the fault, then we can just process on the fault part. This approach will reduce the operation time a lot.

2.2 Characteristics of the signals

Among the 27 recovered signals, we need to identify normal conditions, i.e. when no fault occurs. Some of them are constant signals and some of them are periodic. When we detect or identify the faults, first we have to know the properties of these signals in normal conditions. This section is used to introduce the normal properties.

2.2.1 Constant signals

In the 26 signals recovered from each .x01 file, there are some constant signals like pole-current order, alpha order, pole current, pole-line voltage. Under normal conditions, these have the values given in Table 2:

Table 2.1 Constant signals in the recovered 27 signals

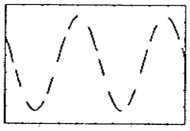
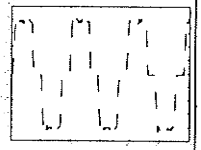
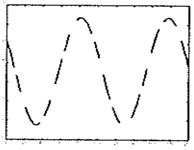
pole current order	alpha order	pole current	pole line voltages
±1400 amps	150 degrees	±1400 amps	±450kV

The first row is the name of the constant signals in the 27 files and the second row in Table 2 contains constant signal values in the normal conditions.

2.2.2 Periodic Signals

In the 27 signals there are also some periodic signals. For Periodic Signals like AC Phase Voltage, Phase Current and 6 pulse Voltage, their waveform and values are shown in the Table 3.

Table 2.2 periodic signals in the 27 recovered signals

	AC Phase Voltage	Phase Current	6 Pulse Voltage
Amplitude(peak to peak)	27KV	1400amps	27KV
Waveform			

The first row in Table 3 gives the names of the periodic signals and the second row gives the values of those signals in normal conditions. The third row gives the typical waveforms for periodic signals.

3 FAULT DETECTION

The 26 signals contains AC Voltage, Pole Current Order, Alpha order, pole current, pole line voltage, 6 pulse voltage, Valve currents, start pulse. Among these signals, AC voltage, pole current and voltages are sinusoidal signals, pole current and voltages are constant when no faults happened. For the phase current, it is still periodic but not sinusoidal as they are determined by the position of the start pulse. The signal modeling for AC phase voltages, pole currents and voltages is easier to realize than that of the phase current, since the AC phase voltages are typical sinusoidal signals, and both of the pole currents and voltages should remain constant as well when the power system operates normally. The phase currents, however, are a little complicated, as they will be determined by the position of the start pulse as shown in Figure1 from **[1].**

According to the [1], we can get the reference signal for the phase current. Before the commutation, its formula is shown in (3.1), during the commutation is shown in (3.2) and after the commutation is shown in (3.3).

$$i_k = 0 \qquad (3.1)$$

$$i_k = \sqrt{2}U(\cos\alpha + \cos v)/(w \cdot L_k) \qquad (3.2)$$

$$i_k = I_d \qquad (3.3)$$

Where, i_k : the phase current

U : the rms value of the phase-to-phase voltage

α : the delay of the start pulse

v: the phase of the phase current signal

I_d : the stable current after current commutation

$i_k : 0 \rightarrow I_d$; while $v : \alpha \rightarrow \mu$ (μ is the interval for commutation)

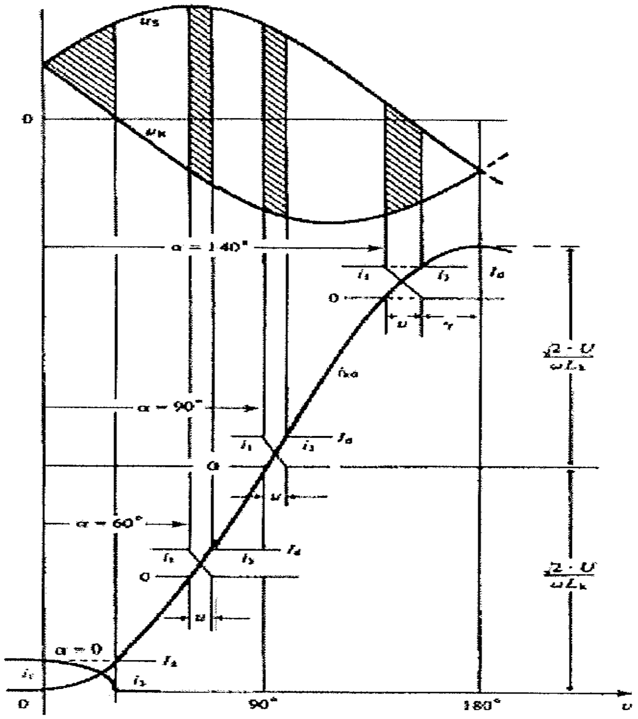


Figure 3.1 The Waveform of Phase Current

Figure 3.1 shows how the position of the start pulse determines the shape of phase currents.

3.1 Abnormal Signal Detection

As we said above, different signals have different features when a fault occurs. For those constant signals, such as pole current order, alpha order, pole current and voltages, it is easy to detect the fault because those signals have fixed value when they are normal. If a fault occurs, we can compare the abnormal value to the normal one to get the error signal based on **Table 2.1.** Also when we read the data in each file, we find out that even in normal conditions the constant signals are not exactly "constant", i.e., they have some small differences not easy to detect in the waveforms. So we have to define some threshold for the fault detection, i.e. only when the value of the signal disturbance over this threshold, we can say that a fault has occurred, otherwise we ignore them.

3.2 Error Detective Filter

In this section, we take the phase current signal as the example. We can see that for different types of faults the phase current distortions are different. In Fig. 5, the error signal of the some phase current is shown. For the three ac Phase signals, there are 120° phase differences between the A-phase and B-phase, B-phase and C-phase, A-phase and C-phase. From the data file we can find out that every 96 samples represent one period of the phase current. So we shift the B-phase by 32 samples and C-phase for 64 samples. If no error has occurred, these signals should be exactly the same shapes. Otherwise, they have some error output.

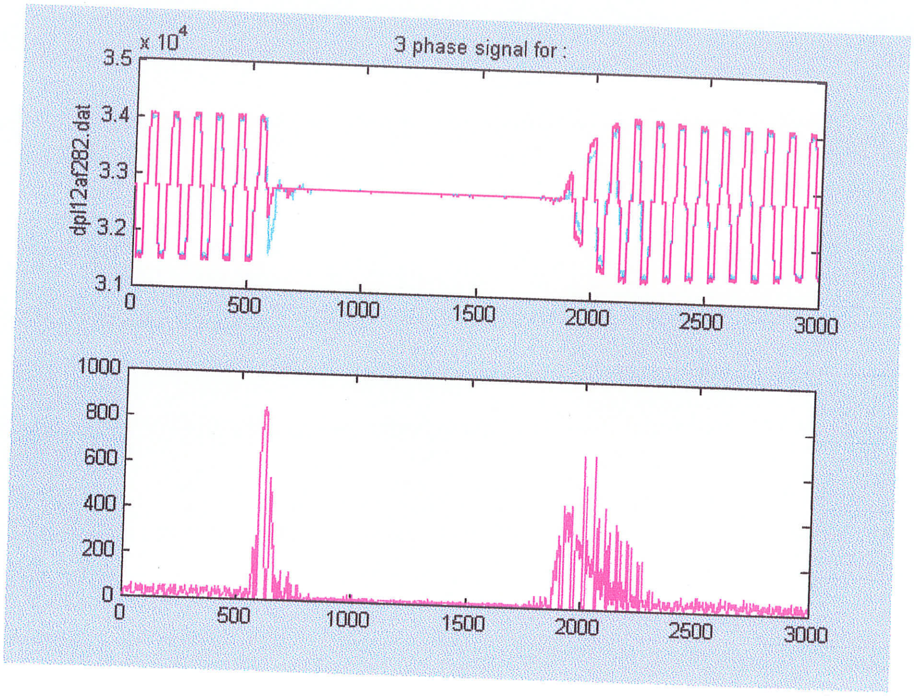


Figure 3.2(a) Pole-line Flash Over

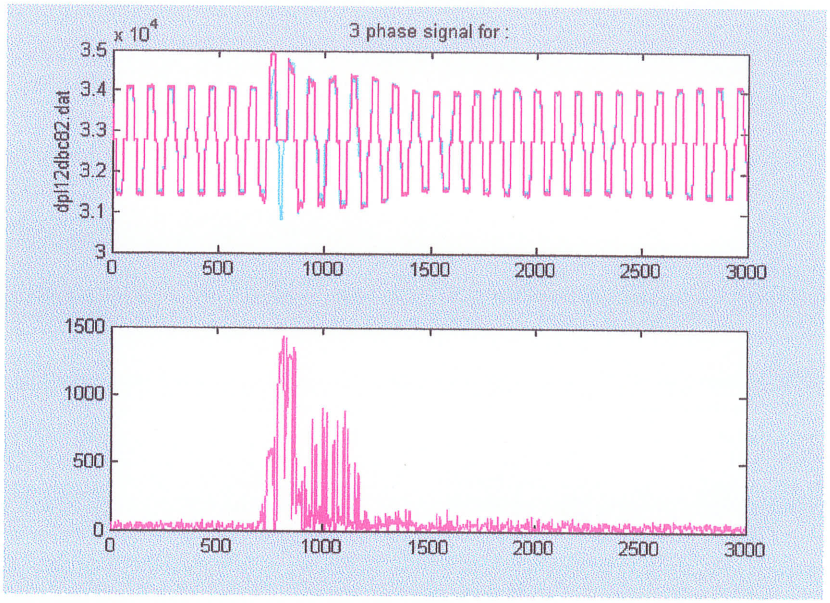


Figure 3.2(b) Commutation

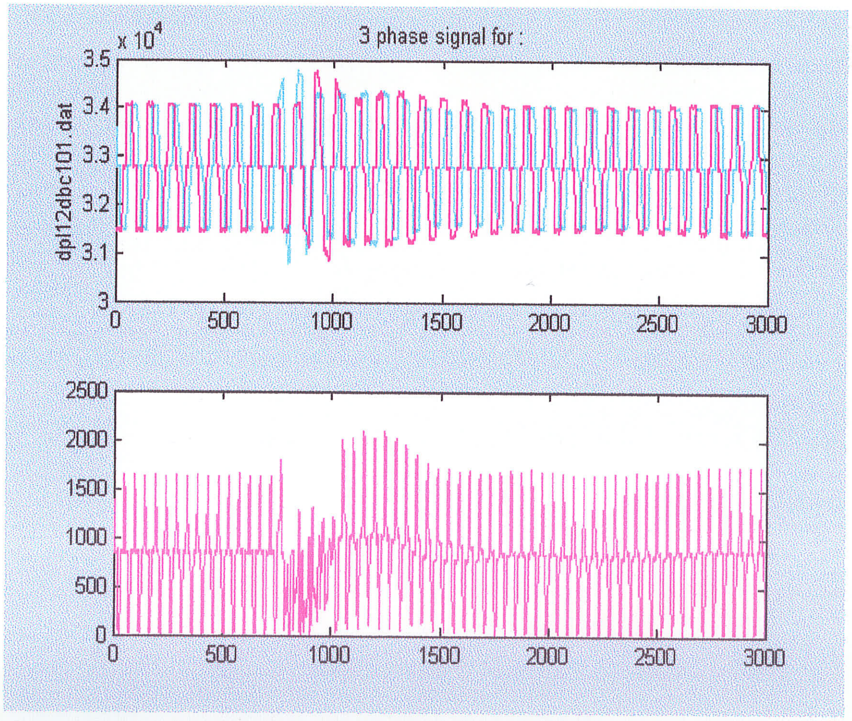


Figure 3.2 (c) AC Disturbance

In Figure 6, it can be seen that for different types of faults the distortions of phase current signal are different. For Pole-line Flash Over fault, there are two peaks, for Commutation fault there is one peak and for AC Disturbance, the fault signals exist almost all the time.

Table 3.1 Some property values of the error signal

Fault type	Counter	distance	average
Pole Line flashover	2	1273	66.732
Pole Line retard	1	0	500.231
AC Disturbance	1	0	60.524

In **Table 3.1**, the first column is the types of the faults, column 2 counter denotes the number of the peaks in the error and the distance is used to denote the how many points between the two peaks. Average in column 4 in Table 4 is the average error value of the 3 phase signals.

3.3 System for Fault Detection and Classification

In this project we set up a system for fault detection and identification (FDI). The FDI system is based on the analysis of the data from Transcan Recording System (TRS). This system has the ability to detect and classify the faults. It gives the indication of the type of the faults and also generates the rules for the fault classification. The following is the flowchart for this system.

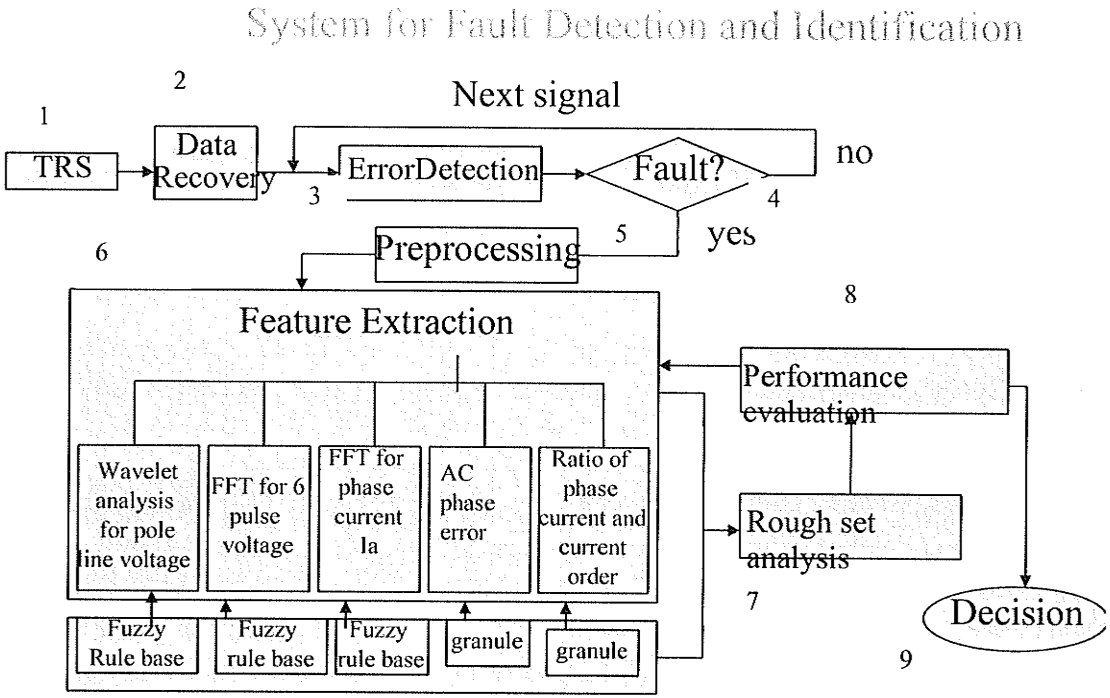


Figure 3.3 Fault Detection and Identification System

Inside the **Figure 3.** we have some numbered blocks. Each block as a particular some function in the detection and identification of power system faults. Block 1 is the TRS, where we get the original data. Here the data are is unreadable to us, as we have introduced in 2.1.1. Block 2 is used for recovering data, a C++ program is set up to recover the data into *.dat ASCII format, as mentioned in 2.1.1. Block 3 and 4 are for error detection. Before we do some process on the signals we have to detect the error signals to see whether a fault happened, if "no" that means no fault happened, we can go back to the TRS to detect the next signal, otherwise, we do the further processing. as mentioned in3.1. Block 6 is Feature Extraction, it is the key part of the whole system. It is used to get the features of the signals based on the characteristic analysis in 2.2. For pole-line voltages we use the wavelet transform and the fuzzy computing algorithm to get the fuzzified decision of the signals. For 6 pulse voltages, we use FFT, a low-pass filter and fuzzy computing to get the results. For phase current, we took the signals of phase A as the example, and we also use FFT, low-pass filter and fuzzy computing to do the analysis. For ac voltages, first we use the equation **(4.16)** to calculate the error of the 3 phase signals and then use the granule algorithm to make the decision. Also we calculate the ratio of phase current and current order, and use the granule algorithm to separate the faults. After the feature extraction, we got 11 attributes and the value based on each of the attributes of different faults. Then set up an information table for the next part. Block 7 is the implementing of rough set analysis, in this part we take the information table which is obtained from part 6 as the input table of rough set. Then use the rough set algorithms on the information table, which is implemented by the tool ROSETTA to generate a collection of classification decision rules. Block 8 is the performance evaluation, we recovery another 55 *.x01 files to do the training of the system. This also provides a feedback to the feature extraction and Rough Set Analysis block, and helps us to optimize the FDI system.

4 FEATURE EXTRACTION

In this chapter we will introduce the key part of the FDI system: Feature Extraction. A fault feature is also called the signature of the fault, which is a distinct pattern of data (signal) that is associated with a particular fault. If a fault signature is detected in the input data, the presence of the particular defect is very likely. In a big system sometimes we often need quite a few attributes to describe the faults. In this section first we use a two-prong approach for analysis of the signals **Figure 4.**. The approach is based on the characteristic of the periodic signals and constant signals. For periodic signals we use the FFT and fuzzy computing algorithm, while for the constant signals we use wavelet transform and also fuzzy computing to get the feature output. In section 4.2 we introduce the algorithm of calculating the ac voltage-phase error and implement granule theory to get the definition. In section 4.3 a new attribute is introduced as the ratio of the phase current and current order.

For each of the attributes in this fault detection and identification system, we use the following structure to determine the kind of fault (see **Figure 4.1**). The fuzzy diagnostic system takes features as inputs and then outputs any indications that fault mode may have occurred in the plant.

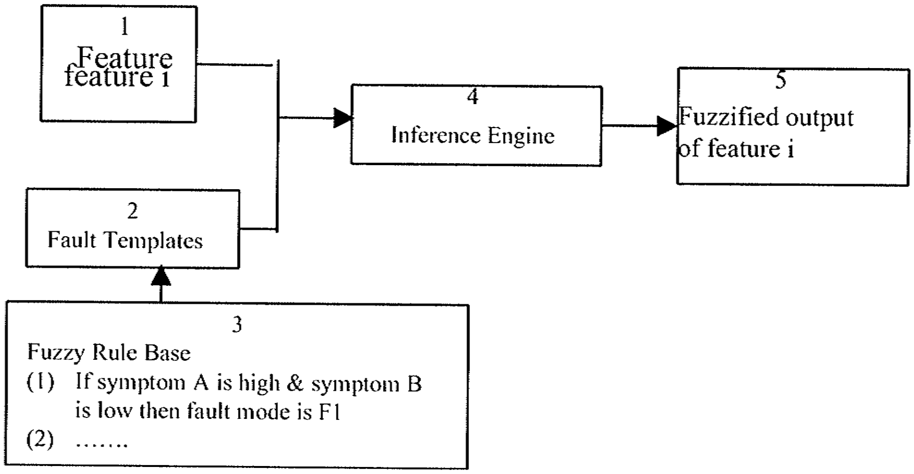

1
Feature
feature i
2
Fault Templates
3
Fuzzy Rule Base
(1) If symptom A is high & symptom B is low then fault mode is F1
(2) …….
4
Inference Engine
5
Fuzzified output of feature i

Figure 4.1 Fuzzy Diagnostic System

The above figure is to show the fuzzy diagnostic system that is used for feature extraction in the system. Part 1 is used to get the feature of the signal such as WT or FFT6P or Ratio, ERROR. Part 2 is the fault templates that include the rule of the typical faults based on this feature i. We got the rules from a study of different types signals. Part 4 is Inference Engine that compares the feature with the templates stored in part 3. Finally, Part 5 is the output of the fuzzy computing, it is the fuzzified output of the feature. The following figure suggests the basic idea of fuzzy computing.

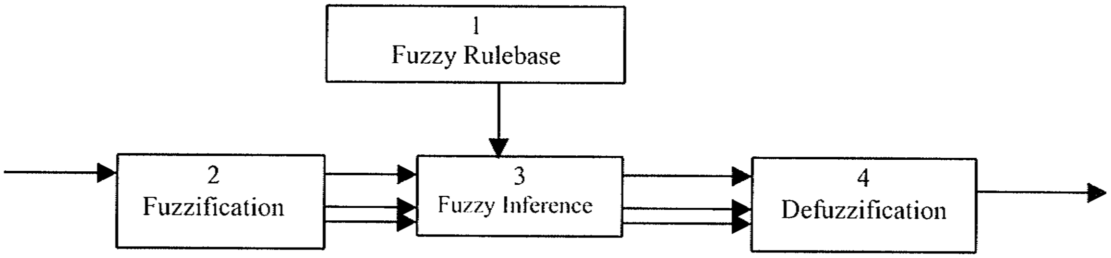


Figure 4.2 The Fuzzy Logic Diagnostic System

The fuzzy diagnostic system takes features as input and then outputs any indications that a fault mode may have occurred in the plant. The fuzzy logic system structure is composed of four blocks: fuzzification (block 2), the fuzzy inference engine (block 3), the fuzzy rule base (block 1) and the defuzzification (block 4) as shown in **Figure 4.2**. The fuzzification block converts features to degree of membership in a linguistic label set such as low, high, etc., and the fuzzy rule base is constructed from symptoms that indicate a potential fault mode. The fuzzy rule base can be developed directly from user experience, simulated models, or experimental data. Fuzzy outputs are aggregated (maximum method) through the fuzzy inference engine to determine a degree of fulfillment for each rule corresponding to each fault mode. Finally, in the last step, the system defuzzifies the resulting output (this is not used in this project).

4.1 A Two-prong Approach for the Periodic Signals and Constant Signals

Based on the characteristics of the signals we have mentioned in chapter 2, we use a two-prong approach to get the feature of the signals. In this section we introduce wavelet theory **(4.1.1)** and also the give a brief introduction to the typical wavelets. In 4.1.3, we implement the wavelet transform on the pole-line voltages. Section 4.1.4, we is to give the fuzzified decision of each type of the faults based on the selected feature. Section 4.1.5 presents the FFT for the 6 pulse signals, where a co-relation calculation is also introduced Section 4.1.6 is the FFT for phase current. The typical 9 types waveform of the phase current will be shown and we also implement fuzzy computing. Till now we have identified 8 attributes of the information system.

The following is the idea of the two-prong approach for feature extraction

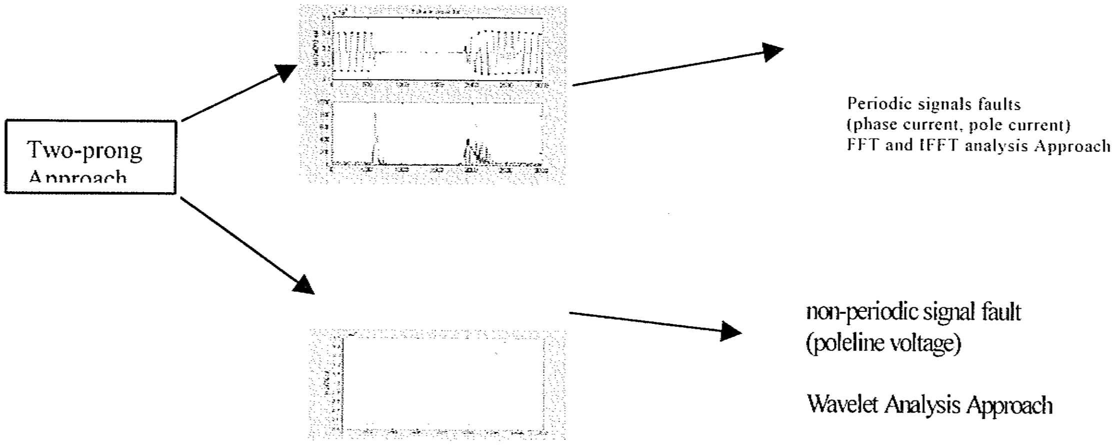

Two-prong
Approach
Periodic signals faults
(phase current, pole current)
FFT and IFFT analysis Approach
non-periodic signal fault
(poleline voltage)
Wavelet Analysis Approach

Figure 4.3 The Two-prong Approach for Signals

In **Figure 4.**, we show that for periodic signals like the phase current, 6 pulse voltages , etc, we use the Fast Fourier Transform (FFT) and Inverse Fast Fourier Transform (IFFT) analysis approach, while for the non-periodic signals we use the wavelet analysis on them.

For the periodic signals we use the FFT together with a hamming window low-pass filter [33] following by a 64 points inverse FFT and a Hamming window filter in **(4.1)**.

$$window(i) = 0.54 + 0.46\cos\left(\frac{2\pi(i-1)}{2048}\right) \qquad (4.1)$$

For the constant signals we use the wavelet transform. Wavelets are scaled waveforms that measure signal variations. By traveling through scales, zooming procedures provide powerful characterizations of signal structure such as singularities. Time varying harmonics are detected from the position and the scale of high amplitude wavelet coefficients.

4.1.1 Wavelet Theory

A wavelet Ψ is a function of zero average (see (4.2)2).

$$\int_{-\infty}^{+\infty} \psi(t)dt = 0, \qquad (4.2)$$

Imposing $\|\psi\| = 1$ implies that $\|\psi_{a,b}\| = 1$. A wavelet $\psi_{a,b}$ has an energy in time that is centered at a over a domain proportional to b.

Let $x(t) \in L^2(R)$ be the input signal to be analyzed. A family of signals is chosen, called wavelets $\{\psi_{a,b}\} \in L^2(R)$, for different values of a and b, given by (4.3).

$$\psi_{a,b} \equiv |a|^{-\frac{1}{2}} \psi(\frac{t-b}{a}) \; \forall a, b \in R \tag{4.3}$$

$\psi_{a,b}$ is a real wavelet. b can be thought of as translation factor and a is the scaling factor (dilation or compression; a notion of frequency). $\psi(t)$ is called the mother wavelet and should satisfy (4.4).

$$C_{\psi} = 2\pi \int \frac{\tilde{\psi}(\omega)}{|\omega|} d\omega < \infty \tag{4.4}$$

where $\tilde{\psi}(\omega)$ is the Fourier Transform of $\psi(t)$. Equation **(4.4)** is called the admissibility condition for the mother wavelet.

The following **Figure 4.** is about the wavelet family of Symmelets at various scales and locations.

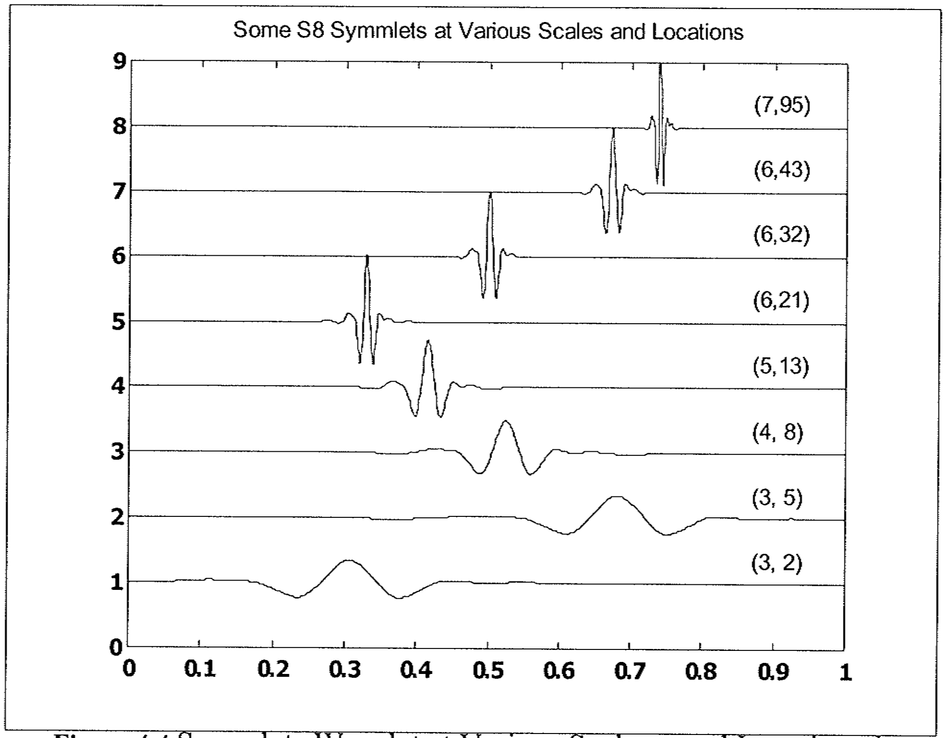


Figure 4.4 Symmlets Wavelet at Various Scales a and Locations b

The following figure is to show the wavelet and its Fourier transform.

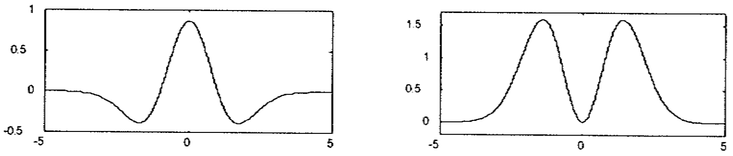


Fig. 12 (a) Mexican hat wavelet

Fig. 12 (b) Fourier transform of (a)

Figure 4.5 Mexican hat wavelet for $\alpha = 1$ and its Fourier transform

The coefficients for the WT for some a and b are defined as the inner product in $L^2(R)$ of $x(t)$ and $\psi_{a,b}(t)$ as in (4.5).

$$c_{a,b} = \langle f, \psi_{a,b} \rangle = \int_{-\infty}^{\infty} x(t)\psi_{a,b}(t)dt \qquad (4.5)$$

Since the input signal is normally available in the form of sampled data, the discrete version of the above ideation is given as in (4.6).

$$c'_{a,b} = \sum_{v=0}^{N} x(v)\psi_{a,b}(v) \tag{4.6}$$

N is the number of samples for which $\psi_{a,b}$ is non-zero. Since most of the features produce a signature in a wide range of frequencies that is spread over a range of time (space), m>0 number of coefficients c'_{ai} are stored. To increase the usefulness of these coefficients, we apply a transformation χ to get a trend of the feature signature in **(4.7)**

$$c_{ji} = \chi(c'_{ai}, b_j) \quad i = 1,......n \tag{4.7}$$

where n is the number of the wavelet scale used. The transformation χ varies from application to application. It can be envelope extraction, magnitude of a complex value, etc. The coefficients $c_{j,i}$ are stacked in a matrix arrangement which we refer to as the *Information Matrix A*, represented as follows in **(4.8)**.

$$A = \underset{(Timeb\ )}{\downarrow} \begin{vmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ . & . & . & . \\ c_{n1} & c_{n2} & \cdots & c_{nn} \end{vmatrix} \tag{4.8}$$

The matrix A with elements c_{ji} has the following characteristics:

- For a fixed $j=u$, the c_{ui}'s give the frequency response of the input signal at a particular time instant.
- For a fixed $I=v$, the c_{jv}'s give the relative level of a particular frequency over a period of time(or space).
- Each column of the matrix A is referred to as $A_i (i = 1,....n)$ and is comparable to a bandpassed version of the signal.

A is normalized column-wise so that $c_{ji} \in [0,1]$. The values of the scales a_i are calculated by an optimization process which is a part of off-line learning algorithm.

4.1.2 Four specific Mother Wavelets

There are several types of wavelets that have been used in different problem areas: Haar, D4, S8, Coiflet C3 Dubuc-Deslauriers Wavelets, Average-Interpolating Wavelets, Meyer Wavelets. In this section, four kinds of popular wavelet families are introduced as shown in **Figure 4.** They are Haar wavelet, D4 wavelet, Coiflet C3 wavelet, S8 Symmlets wavelet. In the upper left-hand corner is a square-wave wavelet. It is the first wavelet. In the upper right-hand corner is Daubechies D4 wavelet, it is the first continuous compactly supported orthonormal wavelet family. They are minimal phase filters that generate wavelets have a minimal support for a given number of vanishing moments. Then the lower left-hand of the figure is another orthonormal wavelts system where both father and mother have special vanishing moments properties. The last on in the lower right-hand corner is the Symmlet wavelet, which are also wavelets within a minimum size support for a given number of vanishing moments, but they are as symmetrical as possible, as opposed to Daubechies filters which are highly asymmetrical (see **Figure 4.**).

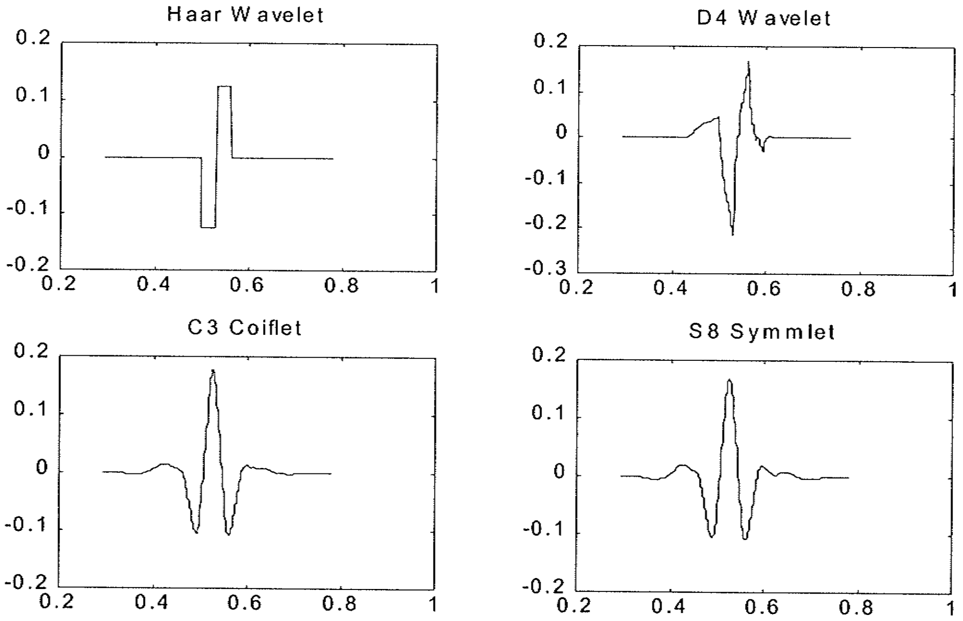


Figure 4.6 Four type of Mother Wavelet

4.1.3 Wavelet Analysis on Pole-line Voltages

In this section we introduce the procedure of extracting the feature of Pole-line voltages using wavelets and also show the result of the typical types of the signals. Then we create a fuzzy template based on the typical type of the wavelets and give the fuzzified result of the fuzzy computing.

For different type of faults, the wavelet coefficients of the pole-line voltages are different. Figure 7 shows how to get the feature using different wavelet. Here we use the third type of the wavelet, Coiflet C3. First we extract the signals to make them have the length n=4096 points that are dyadic (i.e. n=2^J). And then we generate an Orthonormal quadrature mirror filter for wavelet transform usage. Then we can make the wavelet transform of the input signals. Here we tried to use the four typical types for the mother wavelet, but from experimental results we decided to use the Coiflet, since it is better for separating the faults. The others may also be useful in this area, but the Coiflet has already worked well in my case.

The following is the system for wavelet feature extraction on the pole-line voltage.

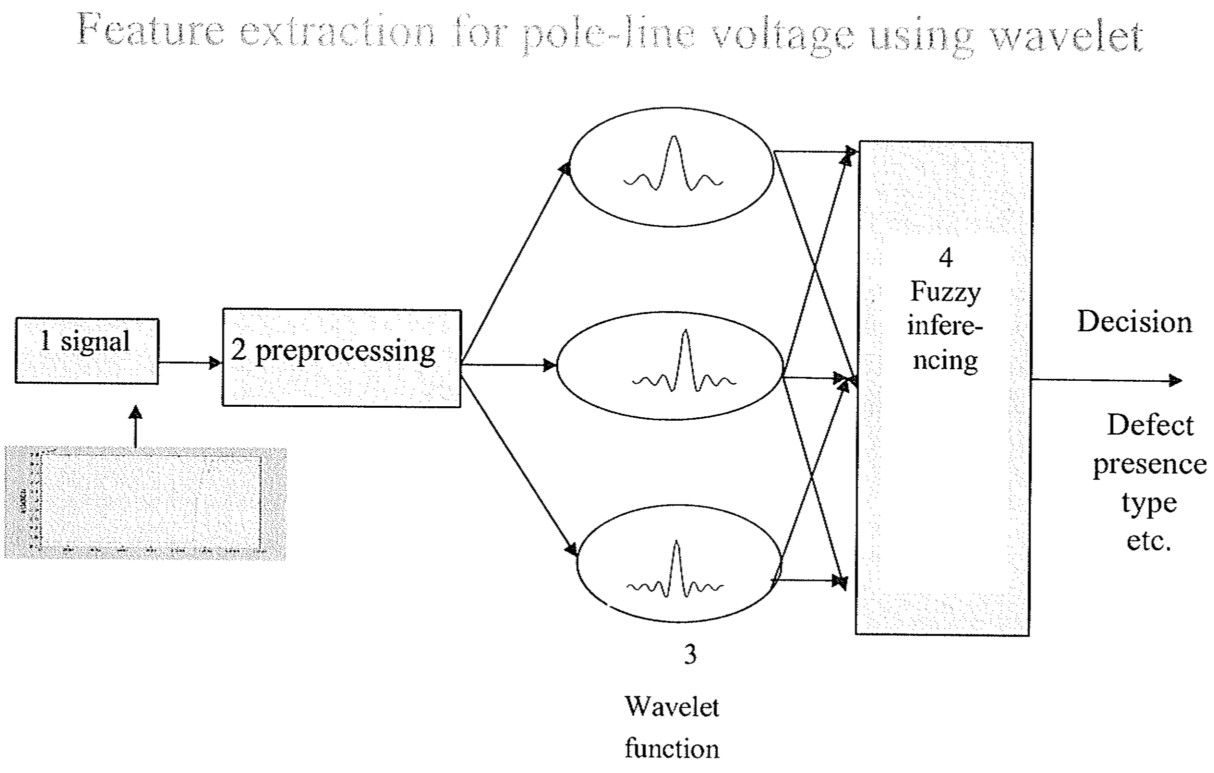

Feature extraction for pole-line voltage using wavelet
1 signal
2 preprocessing
3
Wavelet function
4
Fuzzy infere-ncing
Decision
Defect presence type etc.

Figure 4.7 Feature extraction using different wavelets

In Fig.4.7, first we get the signals after recovery and detection and then do preprocessing on the data which we have already mentioned in Chapter 2. After that the wavelet functions are used on the signals. Finally we use the fuzzy inferencing to compare the feature with the rules in the fuzzy template and calculate the fuzzy output.

In this project we use this system on the pole-line voltage signals. First, we use it on the 56 learning signals to extract the typical 10 wavelets output as the rules and store them in the fuzzy template. In Fig. 8, the 10 typical wavelets output form of the pole-line voltages are shown.

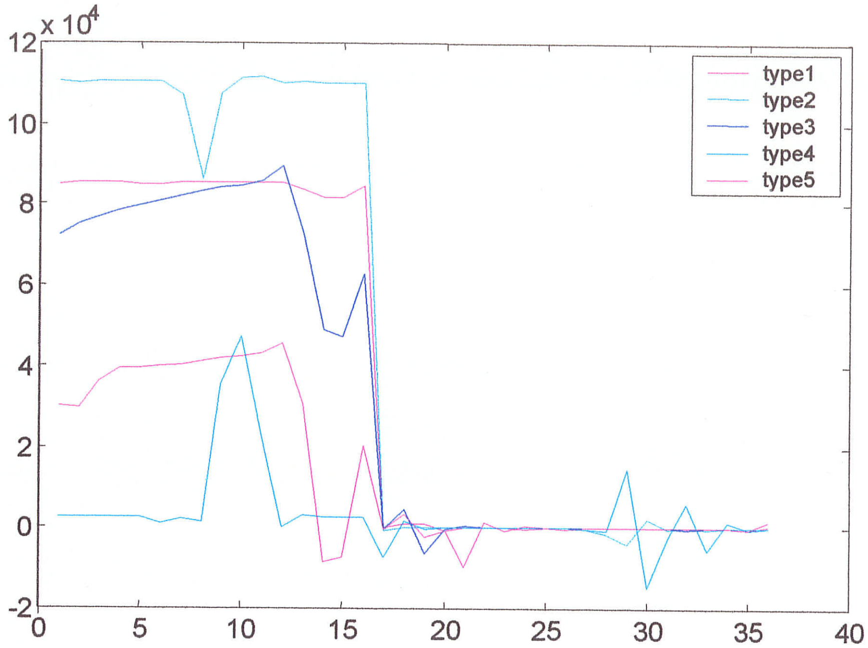


Figure 4.8 (a) first 5 typical types wavelet output of pole-line voltage

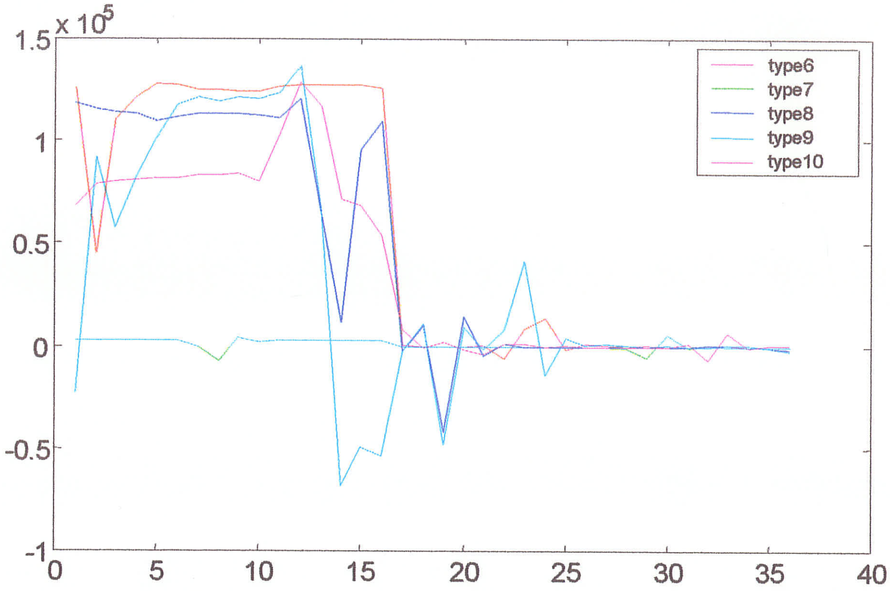


Figure 4.8 (b) Another 5 typical types wavelet output of pole-line voltage

In **Figure 4.8** give the output of the 10 typical wavelet transform of the pole-line voltage fault signals.

Next we give the inverse wavelet transform of the 10 typical wavelet transform output above. It is shown in Fig. 15.

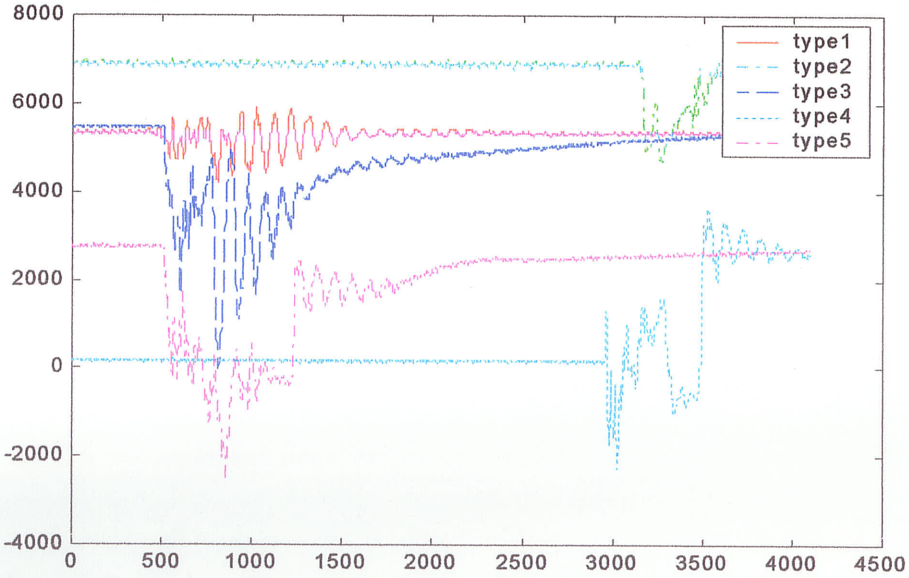


Figure 4.9 (a) the inverse wavelet transform of the first 5 types typical signals

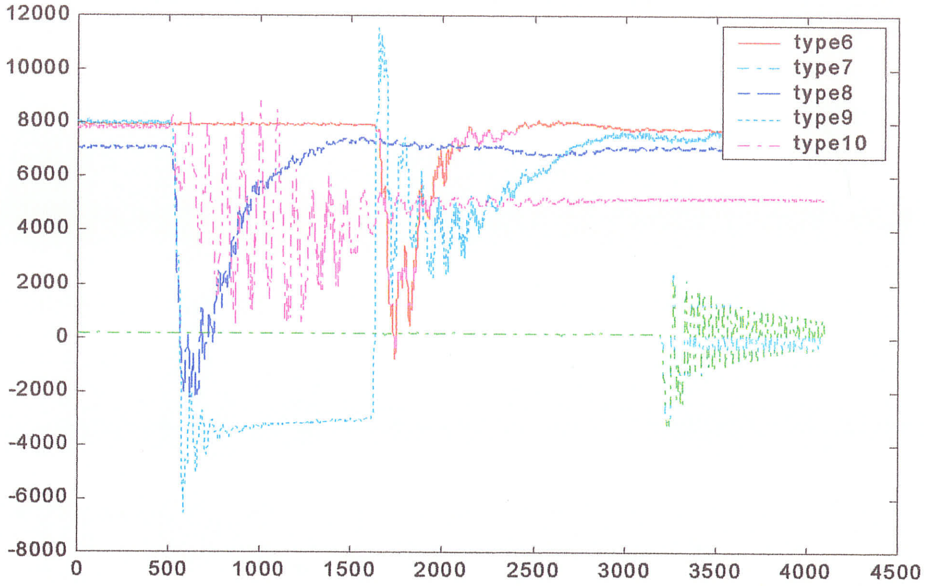


Figure 4.9 (b) the inverse wavelet transform of another 5 types typical signals

As we have mentioned above, we set up the 10 types of the typical wavelet as the fuzzy rule base, and use it to fuzzify the incoming new signals.

Although several measures of similarity and distance are available for fuzzy set [1], a new measure is defined as it provides additional control for adaptive noise suppression.

Let $X,Y \in [0.1]$ be two sets with elements $x_i$and y_i, then a measure of similarity is defined by (4.9) and (4.10).

$$\tilde{A} = \{(x, \mu_{\tilde{A}}(x)) \mid x \in X\} \tag{4.9}$$

$$sim(X,Y) = \sum_i \frac{1-|x_i - y_i|}{1+\alpha|x_i - y_i|} \tag{4.10}$$

where $\alpha > -1$ is a predetermined constant. In this project we set $\alpha = 1$.

4.1.4 Intelligent Decision Making

Decision making is performed by comparing the fuzzified features with the templates stored in the Knowledge-base. The Knowledge base $B \subset R^m$ exploits the information already available about the system and its features through mathematical models, experience, heuristics or any other sources. $\beta_{a_i,k} \in B$ represent the trends for scale a_i(frequency) for the kth feature(k=1.....M), where M is the total number of the anticipated features. We will use '~' to represent a fuzzy set. Let $x \subseteq R^m$. A fuzzy a set $\underset{\sim}{A} \in X$ is set of ordered pairs.

The entity $\mu_{\underset{\sim}{A}}(x)$ is the membership function, the value of which is the grade of membership of x in $\underset{\sim}{A}$. Consider, for example, the membership function **(4.11).**

$$\mu_{\underset{\sim}{A}}(x) = sim(A, x) \tag{4.11}$$

where $A \in X$ is a crisp vector whose value is decided a priori. Based on the definition in Equation (26), the grade of membership of the vector A_i in each $\underset{\sim}{\beta}_{a_i,k}$ is calculated by the membership function $\mu_{\underset{\sim}{\beta}_{a_i,k}}(A_i)$ and used for inferencing via a set of if-then rules. For example, in this project we set up a knowledge base for pole-line voltages. From the experimental we got 10 typical types wavelet transform outputs and store them as the rules inside the base. These rules are some if-then forms. If we want to decide the coming signal's wavelet transform type we do the fuzzy computing with the rules in the knowledge base and give the decision of the type.

After we do this on some of the learning signals we get the following table:

Table 4.1 the fuzzified output of the feature of the different signals

filename	type1	type2	type3	type4	type5	type6	type7	type8	type9	type10	Determine
dpl21f7962	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	[1]
dpl21f7992	0.000	0.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	[3]
dpl21f7a62	1.000	0.026	0.060	0.008	0.012	0.015	0.008	0.019	0.010	0.044	[1]
dpl21f7a92	0.173	0.051	1.000	0.024	0.038	0.035	0.023	0.048	0.031	0.123	[3]
dpl21f7b62	1.000	0.042	0.104	0.014	0.020	0.025	0.013	0.031	0.017	0.078	[1]
dpl21f7b92	0.211	0.061	1.000	0.029	0.045	0.041	0.027	0.056	0.036	0.145	[3]
dpl21f7c62	1.000	0.083	0.204	0.027	0.039	0.050	0.025	0.059	0.033	0.151	[1]
dpl21f7c92	0.169	0.056	1.000	0.028	0.044	0.038	0.026	0.054	0.035	0.137	[3]
dpl21f9662	1.000	0.083	0.232	0.028	0.042	0.050	0.027	0.062	0.035	0.170	[1]
dpl21f9692	0.163	0.052	1.000	0.025	0.040	0.035	0.024	0.046	0.032	0.138	[3]
dpl21f9562	1.000	0.101	0.286	0.035	0.052	0.061	0.033	0.075	0.043	0.202	[1]
dpl21f9592	0.154	0.050	1.000	0.025	0.040	0.035	0.024	0.049	0.032	0.123	[3]
dpl218a62	1.000	0.069	0.157	0.022	0.032	0.041	0.021	0.049	0.027	0.121	[1]
dpl21f8a92	0.000	0.000	0.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	[4]

From the above table we can decide on kind of wavelet transform output of the input signal. We should notice that the value of the type is not a real one. Rather, it is a kind of degree of matchness of input signal and those stored in the fuzzy rule base. For example, "1" here does not mean the two signal is exactly the same. Rather, it means they have the biggest degree of matchness.

4.1.5 FFT for 6 Pulse Voltages

Six pulse voltages form is a sinusoidal signal. To determine the waveform of a new incoming 6 pulse signal, we take the FFT of the signal and use a low-pass filter to get rid of the high frequency elements. Here we use 4096 points FFT and 64-points IFFT on the signals. From the experimental result we also set up a rule base for the typical waveforms. Then use calculate the correlation of the new coming signal and ones in the rule base by this relation:

$$y[n]=\frac{1}{2M+1}\sum_{l=-M}^{M} r[l]\bullet x[l+n] \qquad (4.12)$$

We do this in the frequency domain in case a phase shift happened in the time domain. So here both r and x are in the frequency domain.

Also we set up a rule base for the typical 7 types of waveform for the 6 pulse signals. This is got from the experimental result (see **Figure**).

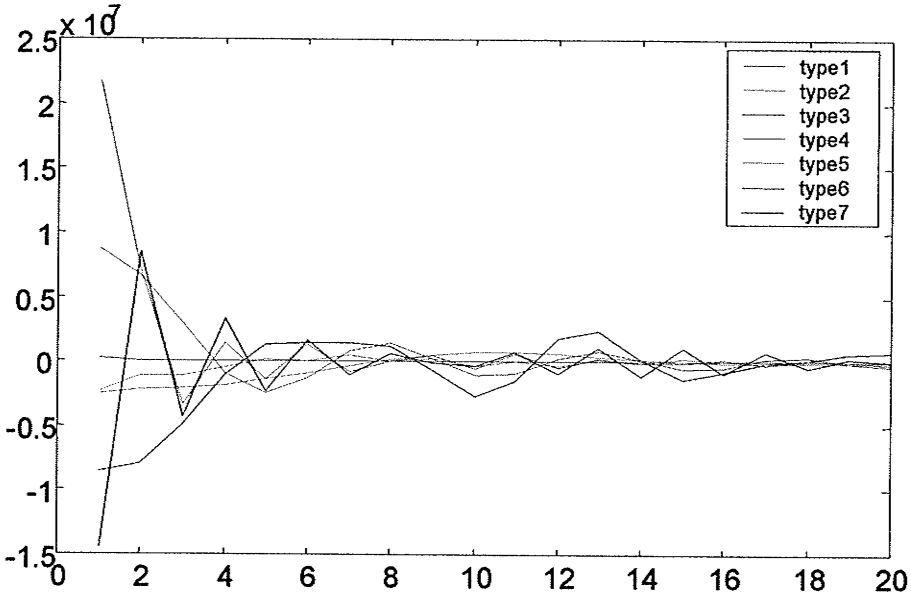


Figure 4.10 types of typical FFT transform of 6 pulse voltage

From the Figure 10 we can see that the 7 types typical signals have different shapes in frequency domain. Then we do the inverse FFT and a low-pass filter on the FFT output shown above, and we can see the waveform of the signals in the time domain as following:

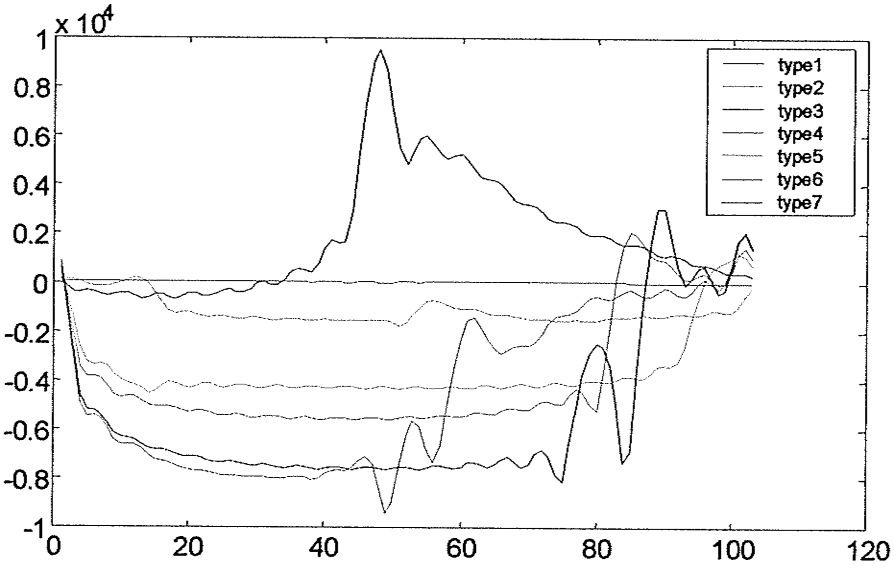


Figure 4.11 the inverse FFT for the 7 types typical 6 pulse signals

So in **Figure 4.11** we can find out that if the signals have dramatic difference in the frequency domain, their waveform in the time domain will also have big difference. That is why we can transform data into the frequency domain to do the identification. While it is also true that if the signals have close FFT output in the frequency domain, their waveform in the time domain should be similar to each other. For example, the following 4 signal's FFT have something in common: the maximum valves and minimum values of these signals in the frequency domain have the same location.

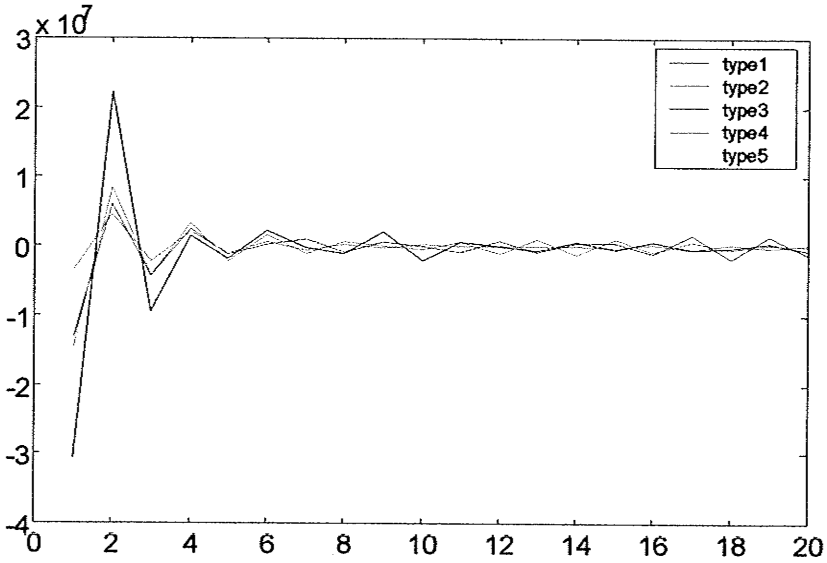


Figure 4.12 (a) 4 pulse voltage signals have similar FFT output

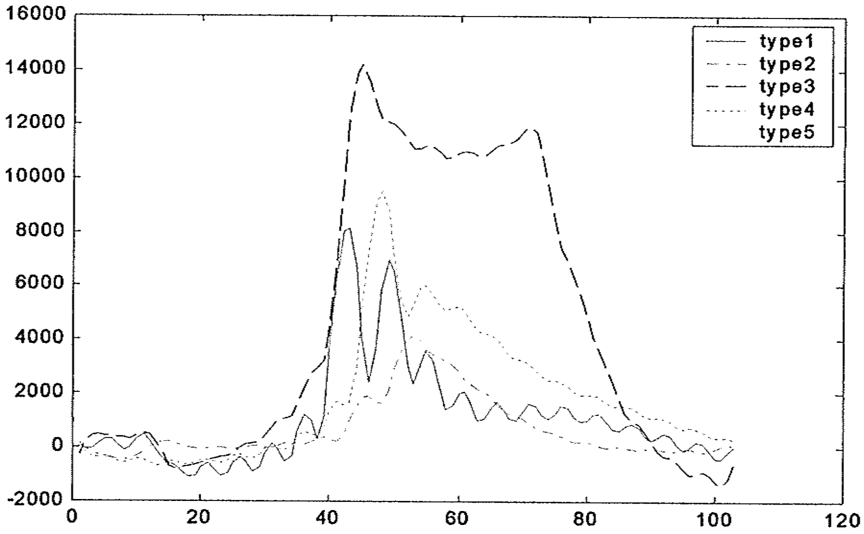


Figure 4.12 (b) the inverse output of the 4 FFT output above

Based on the experimental result we find out the common properties of those similar signals. They are also some other signals similar to each other and have some other properties. We store the properties in if-then form in the rule base for future signal classification. One of the rules property (for example, for the above 4 signals in **Figure 4.12**), we create a table about the max and min value location of the 4 signals shown in **Figurc 4.12**:

Table 4.1 the locations of the maximum and minimum FFT output value for the 4 signals in **Figure 4.**

file	file 1	file 2	file 3	file 4
maxLocation	2	2	2	2
minLocation	1	1	1	1

In **Table 4.1** the first row is the 4 signal files and the second row give the location of the maximum FFT output, the third row is the location of the minimum FFT output in the frequency domain.

Continuing in this manner, we can set up the rule base for the 6 pulse voltage signals and compare the other learning signals of six pulse voltage with the rule base and then we can get the result as shown in Table 4.2.

Table 4.2 Waveform determination of 6 pulse voltages

filename	type1	type2	type3	type4	type5	type6	type7	type8	type9
dpl21f79111.dat		0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
dpl21f79112.dat		0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
dpl21f7951.dat	0.000		0.0166	0.0032	0.0044	0.0086	0.0033	0.0030	0.0072
dpl21f7a111.dat		0.000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
dpl21f7a112.dat		0.000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
dpl21f7a51.dat	0.0000		0.0173	0.0034	0.0046	0.0090	0.0034	0.0031	0.0075
dpl21f7b111.dat		0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
dpl21f7b112.dat		0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
dpl21f7b51.dat	0.0000		0.0371	0.0073	0.0099	0.0193	0.0075	0.0068	0.0162
dpl21f7c111.dat		0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
dpl21f7c112.dat	0.0000	0.0000		0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
dpl21f7c51.dat	0.0000		0.0230	0.0044	0.0059	0.0119	0.0045	0.0041	0.0099
dpl21f96111.dat		0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
dpl21f96112.dat		0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
dpl21f9651.dat	0.0000		0.0141	0.0027	0.0037	0.0073	0.0028	0.0025	0.0061
dpl21f95111.dat		0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
dpl21f95112.dat	0.0000		0.4357	0.1240	0.1778	0.3162	0.1386	0.1269	0.2668
dpl21f9551.dat	0.0000		0.0141	0.0027	0.0037	0.0073	0.0028	0.0025	0.0061
dpl21f8a111.dat	0.0000		0.0314	0.0062	0.0084	0.0163	0.0063	0.0057	0.0137
dpl21f8a112.dat	0.0000	0.0000	0.0000		0.0000	0.0000	0.0000	0.0000	0.0000
dpl21f8a51.dat	0.0000		0.0152	0.0029	0.0040	0.0079	0.0030	0.0027	0.0066
dpl21f8b111.dat	0.0000	0.0000	0.0000	0.0000		0.0000	0.0000	0.0000	0.0000
dpl21f8b112.dat	0.0000	0.0000	0.0000	0.0000	0.0000		0.0000	0.0000	0.0000
dpl21f8b51.dat	0.0000		0.0145	0.0028	0.0038	0.0075	0.0029	0.0026	0.0063

We know that in **Table 4.2** most of the signals are Ring Counter fault except the last 6 signals. From the table we can see for the Ring Counter fault, most of the FFT output for pole1 and pole2 are type1 and pole3 is type 2. This observation makes it possible to separate this kind of fault from others.

4.1.6 FFT for Phase Current

In this section we use FFT and low-pass filter on the phase current signals. First we give a brief introduction of FFT theory. FFT(X) is the discrete Fourier transform (DFT) of vector X. If the length of X is a power of two, a fast radix-2 fast-Fourier transform algorithm is used. If the length of X is not a power of two, a slower non-power-of-two algorithm is employed. For matrices, the FFT operation is applied to each column. For N-D arrays, the FFT operation operates on the first non-singleton dimension (also see **1.1.3** for more information), for length N input vector x, the DFT is a length N vector X, with elements N as in (4.13).

$$X(k) = \sum_{k=1}^{N} x(n) \cdot e^{(-j2\pi(k-1)\cdot(n-1)/N)} \qquad n=1 \tag{4.13}$$

The inverse DFT (computed by IFFT) is given in **(4.14)**.

$$x(n) = \frac{1}{N}\sum_{n=1}^{N} X(k) \cdot e^{(j2\pi(k-1)\cdot(n-1)/N)} \qquad k=1 \tag{4.14}$$

The relationship between the DFT and the Fourier coefficients *a* and *b* is shown in (4.15).

$$x(n) = a_0 \sum a(k) \cdot \cos(2\pi kt(n)/Ndt) + b(k) \cdot \sin(a\pi kt(n)/Ndt) \tag{4.15}$$

Where x is a length N discrete signal sampled at times t with spacing dt. After we do the FFT of the phase current signals we use a hamming window low-pass filter on the FFT output. We also set up a rule base for the phase current signals. These rules come from the 56 learning signals. In the experiment we take the current of phase A for example. There are a total of 9 types of typical FFT output which are shown in Fig.19.

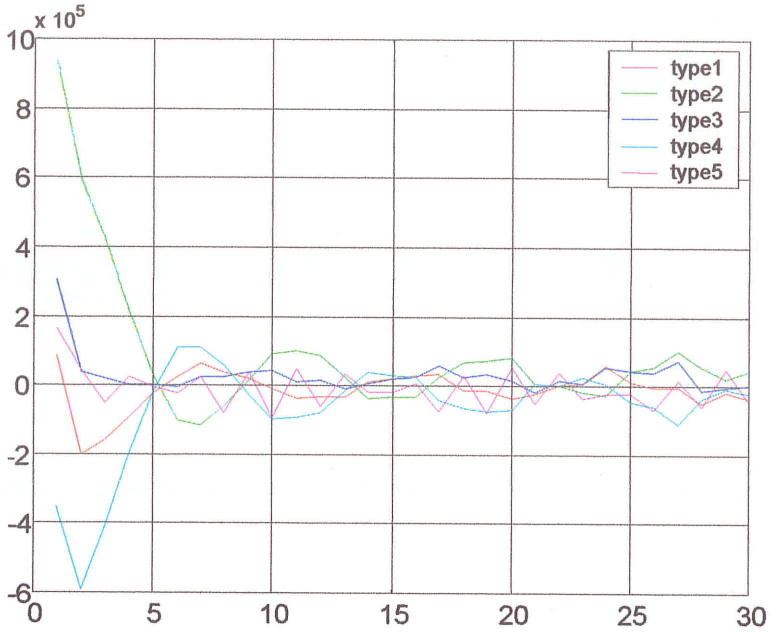


Figure 4.13 (a) the first 5 types of typical FFT for A phase current

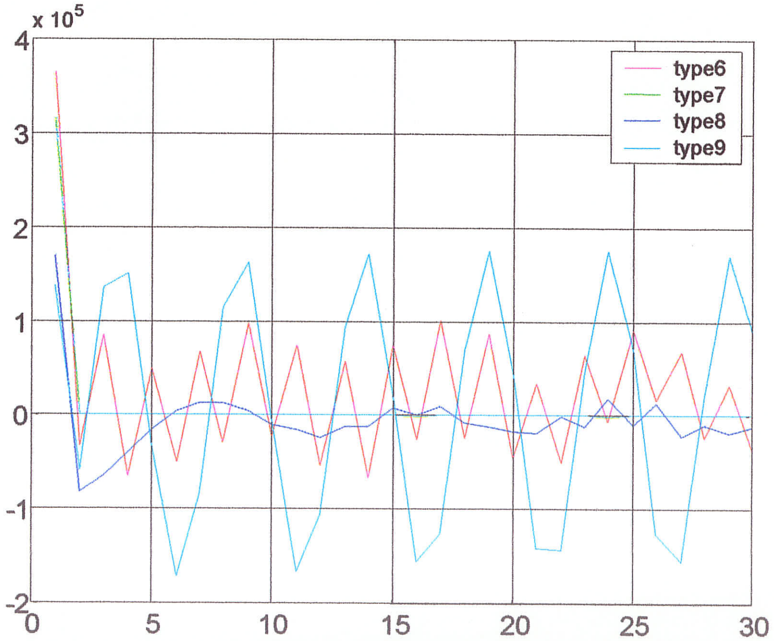


Figure 4.13 (b) another 4 types typical FFT of A phase current

The above **Figure 4.13** is the 9 types of typical FFT output of the phase current. We display them in different colors. From it we can see that the FFT output is different for the 9 signals.

Now we use the low-pass filter on the result in **Figure 4.1** and also do inverse FFT to get the waveforms of the 9 signals in the time domain as shown in Fig. 20.

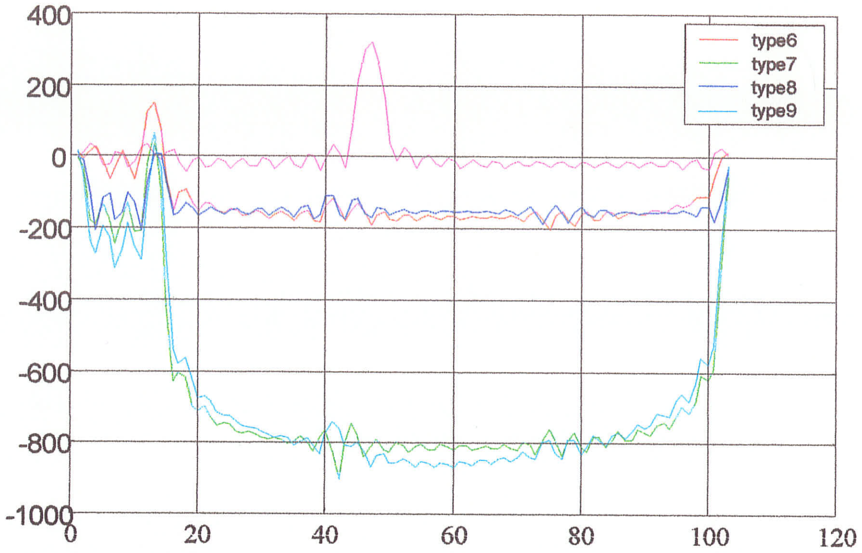


Figure 4.14 (a) inverse FFT of the first 5 A-phase current signals

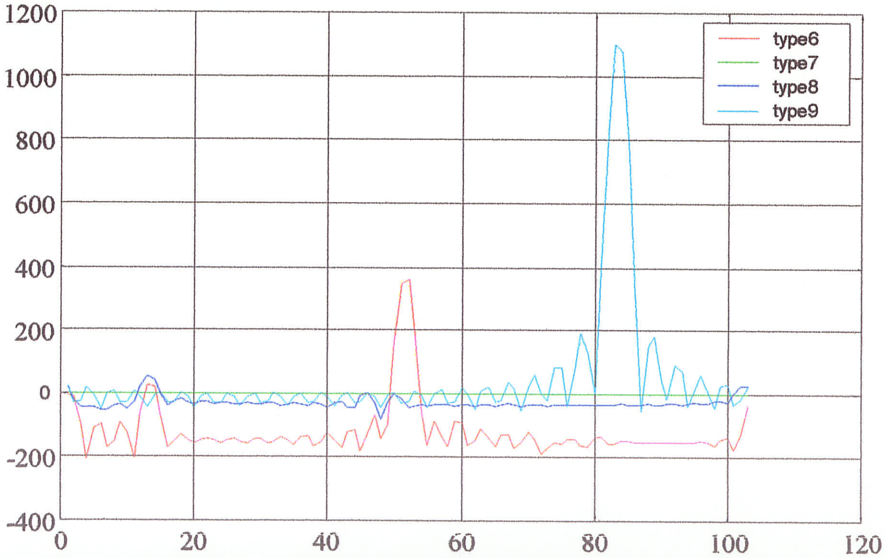


Figure 4.14 (b) another 4 types of the typical inverse FFT for the A phase current

From the above analysis on the signals in Figure 4., we can find out that if the waveforms are different in the time-domain the output also not the same in the frequency domain. So we do the correlation calculation in the frequency domain can deduct the influence of phase shift at the same time make the size of the data smaller. The following is the result of the analysis on the A phase current of all the learning signals. We can get the fuzzified decision of all the signals as shown in **Table 4.3**.

Table 4.3 the waveform decision of Phase Current (phase A)

filename	type1	type2	type3	type4	type5	type6	type7	type8	type9	Decision
dpl12af2102.dat	1.000	0.257	0.539	0.965	0.594	0.463	0.660	0.819	0.339	1
dpl12af2121.dat	1.000	0.255	0.536	0.952	0.593	0.461	0.656	0.815	0.339	1
dpl12af271.dat	0.000	0.000	0.000	1.000	0.000	0.000	0.000	0.000	0.000	4
dpl12afc102.dat	0.107	1.000	0.154	0.072	0.129	0.132	0.144	0.125	0.095	2
dpl12afc121.dat	0.100	1.000	0.141	0.068	0.120	0.124	0.134	0.117	0.089	2
dpl12afc71.dat	0.116	1.000	0.164	0.080	0.143	0.146	0.161	0.138	0.104	2
dpl12cb5102.dat	0.251	1.000	0.366	0.165	0.308	0.324	0.366	0.307	0.211	2
dpl12cb5121.dat	0.217	1.000	0.312	0.144	0.264	0.279	0.312	0.264	0.184	2
dpl12cb571.dat	0.231	1.000	0.332	0.154	0.283	0.295	0.338	0.283	0.196	2
dpl12cbe102.dat	0.386	1.000	0.612	0.239	0.435	0.499	0.591	0.484	0.302	2
dpl12cbe121.dat	0.345	1.000	0.563	0.213	0.420	0.463	0.528	0.428	0.272	2
dpl12cbe71.dat	0.450	1.000	0.676	0.282	0.543	0.566	0.668	0.568	0.358	2
dpl12cd7102.dat	0.141	1.000	0.195	0.097	0.171	0.180	0.193	0.168	0.125	2
dpl12cd7121.dat	0.142	1.000	0.195	0.098	0.171	0.180	0.192	0.168	0.127	2
dpl12cd771.dat	0.141	1.000	0.193	0.099	0.174	0.177	0.195	0.169	0.126	2
dpl12cd782.dat	0.271	0.100	0.180	1.000	0.188	0.159	0.199	0.226	0.130	4
dpl12da1102.dat	0.446	1.000	0.656	0.284	0.529	0.539	0.596	0.516	0.373	2
dpl12da1121.dat	0.343	1.000	0.484	0.231	0.438	0.425	0.481	0.412	0.297	2
dpl12da171.dat	0.305	1.000	0.464	0.195	0.396	0.377	0.449	0.372	0.250	2
dpl12dba102.dat	0.335	1.000	0.525	0.211	0.419	0.426	0.528	0.426	0.267	2
dpl12dba121.dat	0.302	1.000	0.468	0.192	0.377	0.383	0.470	0.382	0.243	2
dpl12dba71.dat	0.346	1.000	0.524	0.223	0.444	0.435	0.551	0.443	0.279	2
dpl12dbb102.dat	0.292	0.106	0.189	1.000	0.203	0.168	0.206	0.241	0.142	4
dpl12dbb121.dat	0.198	0.070	0.126	1.000	0.134	0.114	0.139	0.162	0.093	4
dpl12dbb71.dat	0.437	0.162	0.290	1.000	0.294	0.250	0.307	0.349	0.212	4
dpl12dbc102.dat	0.200	0.098	0.237	0.111	1.000	0.175	0.304	0.285	0.108	5
dpl12dbc121.dat	0.677	0.337	0.796	0.393	0.584	0.621	1.000	0.871	0.366	7
dpl12dbc71.dat	0.430	0.178	0.300	1.000	0.368	0.274	0.327	0.365	0.235	4
dpl12dbd102.dat	0.611	0.243	0.424	1.000	0.452	0.387	0.447	0.514	0.317	4
dpl12dbd121.dat	0.208	0.071	0.129	1.000	0.133	0.118	0.143	0.168	0.093	4
dpl12dbd71.dat	0.603	0.226	0.410	1.000	0.377	0.361	0.418	0.485	0.298	4
dpl12dd2102.dat	0.362	0.136	0.238	1.000	0.242	0.221	0.257	0.297	0.177	4
dpl12dd2121.dat	0.350	0.129	0.228	1.000	0.232	0.211	0.246	0.285	0.167	4
dpl12dd271.dat	1.000	0.092	0.275	0.172	0.210	0.189	0.292	0.457	0.114	1
dpl12de2102.dat	0.969	0.220	0.472	1.000	0.436	0.395	0.525	0.673	0.284	4
dpl12de2121.dat	1.000	0.220	0.477	0.954	0.439	0.398	0.531	0.684	0.285	1
dpl12de271.dat	0.265	0.115	1.000	0.103	0.250	0.249	0.795	0.452	0.107	3
dpl12dfb102.dat	0.446	0.193	0.571	0.230	0.548	0.386	1.000	0.853	0.209	7
dpl12dfb121.dat	0.413	0.187	0.559	0.217	0.506	0.375	1.000	0.766	0.197	7
dpl12dfb71.dat	0.611	0.181	0.353	1.000	0.386	0.305	0.410	0.495	0.240	4

4.2 ac Disturbance

In this section we first introduce the formula of calculating the ac voltage phase error and then use the granule algorithm for the error definition.

4.2.1 ac voltage phase error calculation

We know that 3 phase voltages, the A-phase, B-phase and C-phase have fixed phase difference between each other. That is, 120°, it is 1/3 of 360°. From the data files we can find out that one period of ac voltage is represented by 96 point of data. So if we shift B-phase by 1/3 of one period which means 32 points and shift C-phase by 64 points then there should be no phase difference among the 3 phases. If no fault has occurred, the error of the 3 phase should be very small, otherwise it will be large. From the experimental results we obtained, we found that for different types of fault the values of the disturbance are in different intervals, so we take this as another attribute to decide the type of faults. The disturbance of the 3-phase signals is calculated by the following formula **(4.16):**

$$error = \frac{|Phase_A - Phase_B| + |Phase_B - Phase_C| + |Phase_C - Phase_A|}{3} \qquad (4.16)$$

Now we take one file as the example, and the analysis results are shown below:

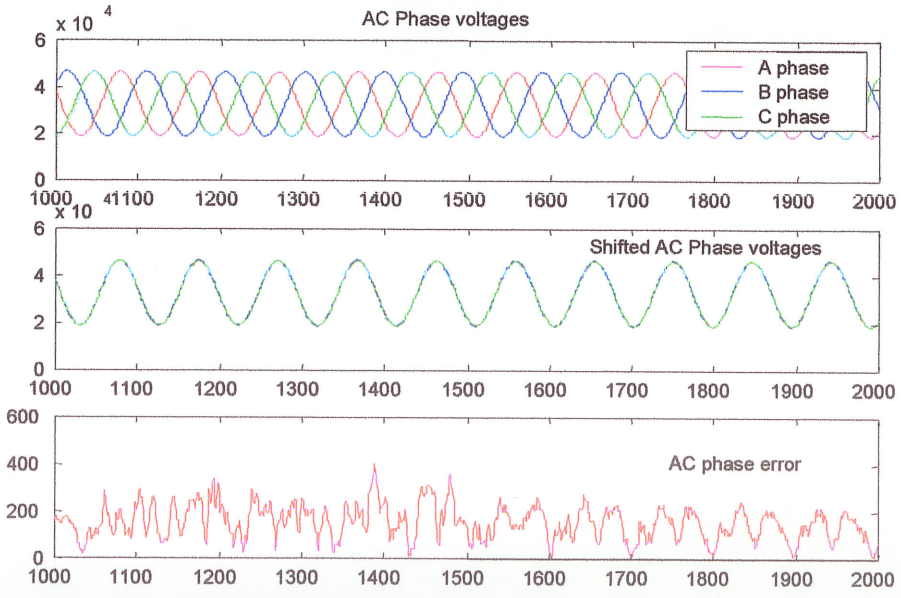


Figure 4.16 Analysis of AC 3-phase voltages by phase shifting method

In **Figure 4.1** the first part is the original 3 phases voltage. The signal in red represents the A-phase, the signal in blue is the B-phase and the signal in green is the C-phase. We can see clearly that there are phases between the A-B-C phase. Then the second part is the shifted output, they are till in the same color as in part one. We can find out that the 3-phase signal is almost the same, except in some areas where a fault or disturbance has occurred. The last part in the figure is the error calculation output using formula **(4.16).** Now we get the result of the ac error for each of the 3-phase voltage learning signals as shown in **Table 6.1**

.

Table 4.4 ac error calculation for different type of faults.

AC Error	Fault type
137.2993	RingCounter
134.6926	RingCounter
137.2309	RingCounter
144.8651	RingCounter
134.3129	FilterBank
137.6172	FilterBank
139.6347	FilterBank
139.1256	FilterBank
223.6835	ValveCAB
197.9716	ValveCAB
141.6358	ValveCAB
177.8086	500kv Close
173.8517	500kv Close
406.2066	AC Disturb
358.9058	AC Disturb
583.374	AC Disturb
341.6262	AC Disturb
408.805	AC Disturb
254.8043	PoleFlashOver
261.1932	PoleFlashOver
255.7666	PoleFlashOver
257.3749	PoleFlashOver
258.5361	PoleFlashOver
184.3592	PoleRetard
243.3629	PoleRetard
239.9993	PoleRetard
233.4423	PoleRetard
115.8183	Aysm Pro
185.428	Aysm Pro
189.891	Aysm Pro
165.2361	Aysm Pro
229.2396	DC Disturb
226.0509	DC Disturb
132.7284	DC Disturb
224.9378	Commutation
224.5581	Commutation
225.4212	Commutation
147.2432	CurrentBlip
149.4044	CurrentBlip
144.7698	CurrentBlip
139.9294	CurrentBlip

In the above table, the first column is the ac error values and the second column is the types of the faults in the 56 learning signals. When we do the analysis on the error values we notice that for the same type of fault generally the value falls into an interval that can be separated from the other type of fault. So we use the granule algorithm to group the values. This will be introduced in the following section.

4.2.2 Granule algorithm for Error definition

After we got this feature (ac phase Disturbance), we found out for the same kind of faults the values of the errors are close to each other. So we can estimate the dynamic range of the errors and separate them into different intervals. A granule is a kind of grouping where the elements of the grouping are in some way similar. We use Radial Basis Functions to determine the distance between the input vector and a prototype vector.

Radial basis function methods have their origins in techniques for performing exact interpolation of a set of data points in a multi-dimensional or one-dimensional space. The exact interpolation problem requires every input vector to be mapped exactly onto the corresponding target vector and forms a convenient starting point for our discussion of radial basis functions.

The radial basis function approach introduces a set of N *basis functions*, one for each data point, which take the form $\phi(\|x-x^n\|)$ where $\phi(\bullet)$ is some non-linear function and the nth such function depends on the distance$\|x-x^n\|$, usually taken to be Euclidean between x and x^n. The output of the mapping is then taken to be a linear combination of the basis function **(4.17):**

$$h(x)=\sum_n w_n\phi(\|x-x^n\|) \qquad (4.17)$$

Both theoretical and empirical studies show that, in the context of the extract interpolation problem, many properties of the interpolating function are relatively insensitive to the precise form of the non-linear function $\phi(\bullet)$. Several forms of basis function have been considered. In this project, we use the Gaussian in **(4.18):**

$$\phi_j(x)=\exp(-\frac{\|x-\mu_j\|^2}{2\sigma_j^2}) \qquad (4.18)$$

where x is the input vector(AC Voltages Error, Ratio of the Phase Current and Current Order) with elements x_j and μ_j is the vector determining the center of basis function ϕ_j. The basis function parameters should be chosen to form a representation of the probability density of the input data. This leads to an unsupervised procedure for optimizing the basis function parameters which depends only on the input data from the training set and ignores any target information (decision). The basis function centres μ_j can then be regarded as *prototypes* of the input vectors. There are several ways for selecting the basis function centresμ_j. One is to set them equal to a random subset of the input vectors from the training set. But this is not an optimal procedure so far as density estimation is concerned, and may also lead to the use of an unnecessarily large number of basis functions in order to achieve adequate performance on the training data. Another approach is to start with all data points as basis functions centers and then selectively remove centers in such a way as to have minimum disruption on the performance of the system.

There are also some other procedures to choose the width parameters σ_j. One heuristic approach is to choose all the σ_j to be equal and to be given by some multiple of the average distance between the basis function centers. This ensures that the basis functions overlap to some degree and hence give a relatively smooth representation of the distribution of training data. The optimal width may be different for basis functions in different regions of input data. For instance, the widths may be determined from the average distance of each basis function to its *L* nearest neighbors where *L* is typically small. Based on the granule theory and the training information table we got for the ac Phase Error, and I did some experiments using this approach to set up the basis function for this system as in (4.19)

$$ErrorVeryLow = e^{(-(error-137.00)^2/72.00)} \tag{4.19 a}$$

$$ErrorLow = e^{(-(error-160.00)^2/200.00)} \tag{4.19 b}$$

$$ErrorHigh = e^{(-(error-245.00)^2/6050.00)} \tag{4.19 c}$$

For each of the error value we got by the file *granule.m* (in Appendix) we calculate the output of ErrorVeryLow(error), ErrorLow(error) and ErrorHigh. The biggest output for a certain input data means the range that the input data falls in. It is shown as in **Figure 4.17**

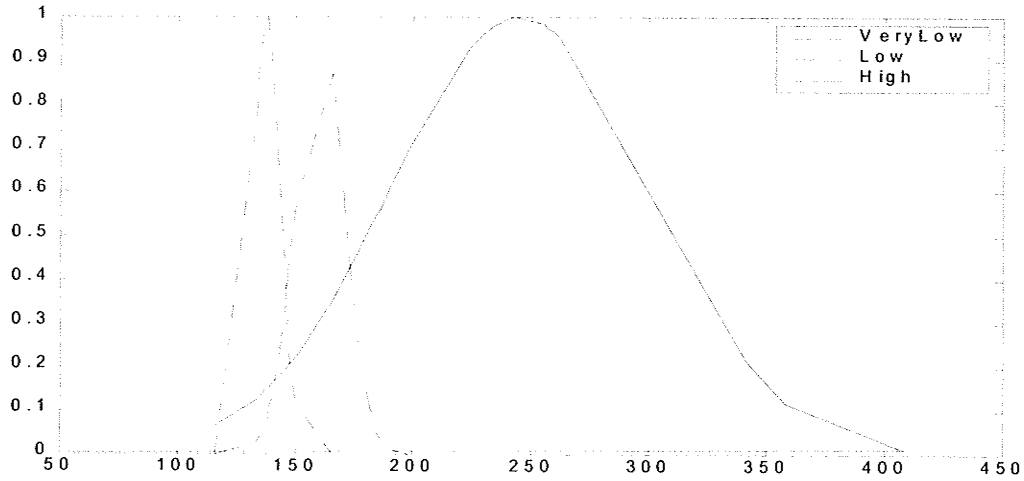


Figure 4.17 Granule Output for AC Voltage Phase Error

The basis function in red is the *ErrorVeryLow*, in green is the basis function for *ErrorLow*, while the blue one is the basis function for *ErrorHigh*. Based on the above figure, we can calculate the granule output of a new coming signal. For example, after we do the 3-phase error calculation on a new ac voltage we get the error value, it is 170.00, then we calculate the *ErrorVeryLow* use **(4.19 a)**, *ErrorLow* use **(4.19 b)**, *ErrorHigh* use the formula **(4.19 c).** The result is 0.0269 for *ErrorVeryLow*, 0.6065 for *ErrorLow* and 0.3947 for *ErrorHigh*. So that means the biggest possibility of this error value is *ErrorLow*.

We use this algorithm on all the ac voltages phase error and then we can get the information table as **Table 4.**

Table 4.5 the error output and granule

filename	ac phase Error	Granule	Fault type
dpl21f79113.dat	137.299	VL	RingCounter
dpl21f7a113.dat	134.693	VL	RingCounter
dpl21f7b113.dat	137.231	VL	RingCounter
dpl21f95113.dat	144.865	VL	RingCounter
dpl21f8a113.dat	134.313	VL	FilterBank
dpl21f8c113.dat	137.617	VL	FilterBank
dpl21f8d113.dat	139.635	VL	FilterBank
dpl21f8e113.dat	139.126	VL	FilterBank
dpl2200b113.dat	223.684	H	ValveCAB
dpl227ce113.dat	197.972	H	ValveCAB
dpl226e8113.dat	141.636	VL	ValveCAB
dpl21f2a113.dat	177.809	L	500kv Close

dpl21f2b113.dat	173.852	L	500kv Close
dpl12e3363.dat	406.207	VH	AC Disturb
dpl224a5113.dat	358.906	VH	AC Disturb
dpl224d0113.dat	583.374	VH	AC Disturb
dpl12cbe63.dat	341.626	VH	AC Disturb
dpl12e3d63.dat	408.805	VH	AC Disturb
dpl12afc63.dat	254.804	H	PoleFlash
dpl12dfe63.dat	261.193	H	PoleFlash
dpl12dfb63.dat	255.767	H	PoleFlash
dpl12e4463.dat	257.375	H	PoleFlash
dpl2344e113.dat	258.536	H	PoleFlash
dpl225c4113.dat	184.359	H	PoleRetard
dpl12af263.dat	243.363	H	PoleRetard
dpl12dd263.dat	239.999	H	PoleRetard
dpl12de263.dat	233.442	H	PoleRetard
dpl2240563.dat	115.818	VL	Aysm Pro
dpl224d9113.dat	185.428	L	Aysm Pro
dpl22697113.dat	189.891	L	Aysm Pro
dpl2288b113.dat	165.236	L	Aysm Pro
dpl12dba63.dat	229.24	H	DC Disturb
dpl12e2a63.dat	226.051	H	DC Disturb
dpl2249c113.dat	132.728	VL	DC Disturb
dpl1305463.dat	224.938	H	Commutation
dpl1306263.dat	224.558	H	Commutation
dpl1307163.dat	225.421	H	Commutation
dpl226cc113.dat	147.243	VL	CurrentBlip
dpl226d5113.dat	149.404	VL	CurrentBlip
dpl226e5113.dat	144.77	VL	CurrentBlip
dpl226e7113.dat	139.929	VL	CurrentBlip

In **Table 4.** the first column is the names of the phase A ac voltage signals, column 2 is the ac phase error from **(4.18),** the third column is the granule output of all the ac phase error and finally in column 4 is the types of the faults.

4.3 Ratio for Phase-current and Current-order

When we working on the data from the TRS we try to find out more attributes that can decide the type of the faults which means for different kind of faults the attribute has different description. The more "useful" attributes we have the more accurate for the decision. That is why the ratio of phase current and current order have been used. The displays in Fig. 23 show the results of the analysis of the phase current signal in different faults. We calculated the ratio and found out that for most of the Valve Ring Counter fault, Valve CAB fault and AC Filter Bank Testing the ratio is bigger than 1 in a certain period of the fault duration. And the ratio is close to 1 in the Pole-line Flashover fault, Pole-line Voltage Force Retard fault and 500kv Close fault.

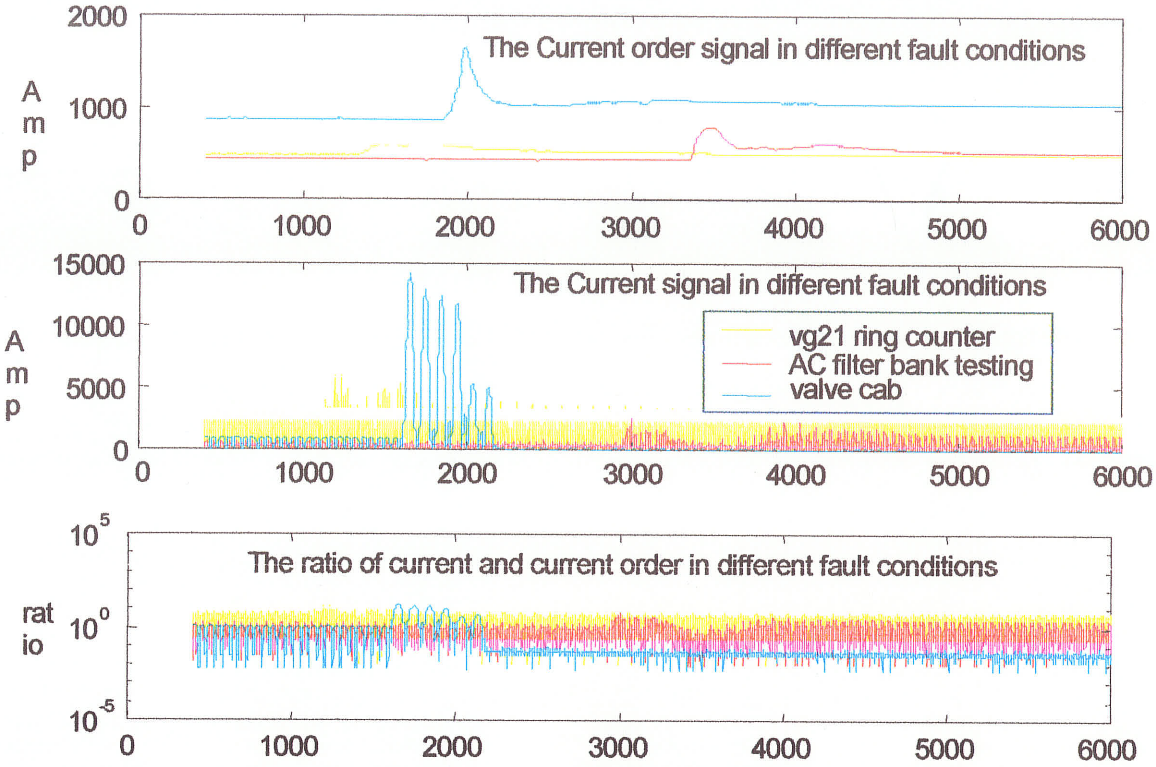


Figure 4.18 Ratio in the Valve Ring Counter, Valve CAB and AC Filter Bank Faults

In Fig.4.18, the signals in yellow is the Ring Counter fault, in red is the Filter Bank Testing fault and that in blue is the Valve CAB fault. The first part of the figure shows the current order signal in the 3 faults and the second part is the current order signals. The third part is the ratio of the current order and the current in the 3 faults respectively.

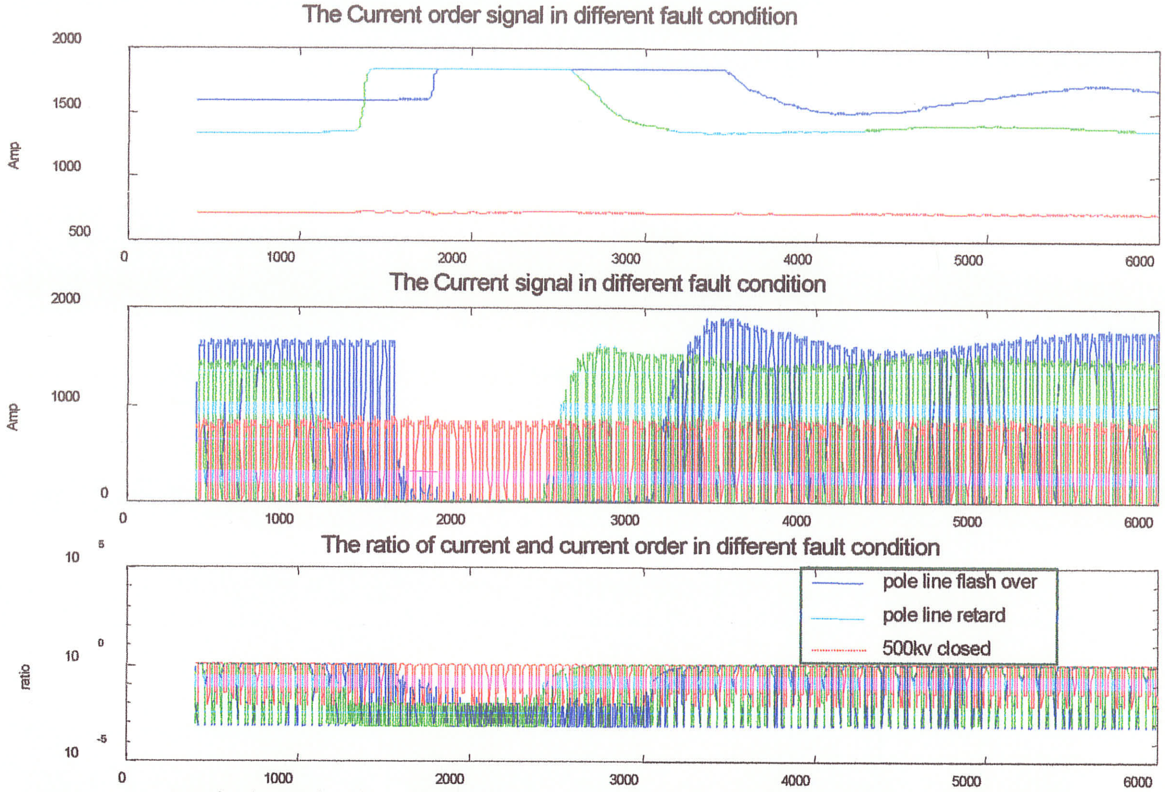


Figure 4.19 Ratio in Pole-line Flash Over, Pole-line Retard and 500kv Close faults

In **Figure 4.119** the signals in blue is the pole-line flash over fault, in green is the pole-line Retard fault and that in red is the 500KV close fault. The first part of the figure is showing the current order signal in the 3 faults and the second part is the current order signals. The third part is the ratio of the current order and the current in the 3 faults respectively.

Here we also use the granule algorithm for the Ratios. From the Ratio value of all the phase current and current order in the learning signals, we find out that the maximum ratio is in the fault of Valve CAB, which is 16.26. We set up the following formulae to classify the ratios into three ranges: Normal, Mediate and High. The selection of the parameters like σ_j and μ_j are based on the basis function theory **(4.18)** and to modify these formulae to be adaptive to the experimental need of our data, it is shown as follows in **(5.1)**.

$$RatioNormal = e^{(-(Ratio-1)^2/1)} \qquad (4.20\text{ a})$$

$$RatioMedate = e^{(-(Ratio-4)^2/10)} \qquad (4.20\text{ b})$$

$$RatioHigh = e^{(-(Ratio-12)^2/36)} \qquad (4.20\text{ c})$$

And the corresponding result of this grouping output for all the ratios is shown in **Figure 4.**.

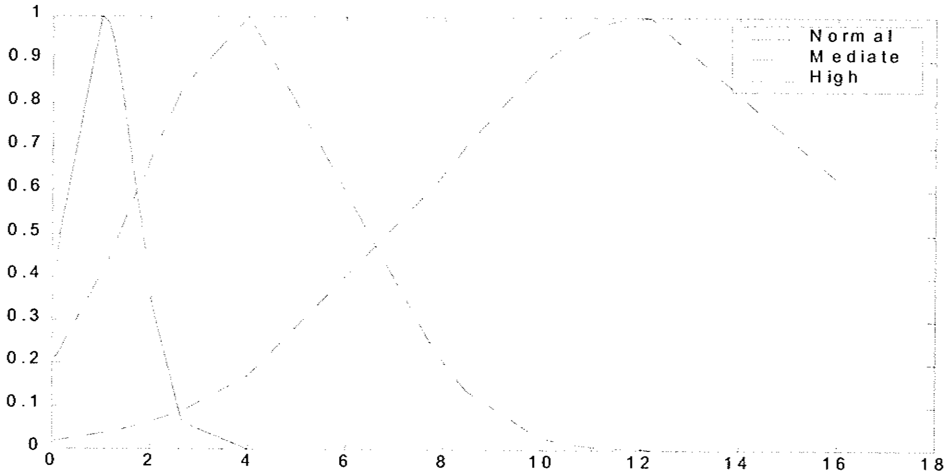


Figure 4.20 Ratio Granule Output for all the learning signals

In **Figure 4.**, the basis function in red is the *RatioNormal*, and green is the basis function for *RatioMediate*, while the blue plot is for the basis function for *RatioHigh.* Based on the above figure, we can calculate the granule output of a training signal's ratio of phase current and current order. For example, after we do the 3-phase error calculation on a new ratio we is 2.3, then we calculate the *RatioNormal* use **(4.20 a)** *RatioMediate* use **(4.19 b)**, *RatioHigh* use the formula **(4.19 c).** The result is 0.1845 for *RatioNorma,* 0.7490 for *RatioMediate* and 0.0733 for *RatioHigh.* So that means the biggest possibility of this error value is *RatioMediate.*

Now we use this granule algorithm on all the ratios we got from the learning signals, we can get the result table as **Table 4.**:

Table 4.6 the ratio of current and current order, and granule ratio result

String	Float	String	String
21f79	12.24	H	RingCounter
21f7a	11.6	H	RingCounter
21f7b	7.71	H	RingCounter
21f7c	10.47	H	RingCounter
21f96	0.056	VL	RingCounter
21f95	0.079	VL	RingCounter
21f8a	5.94	H	FilterBank
21f8b	12.54	H	FilterBank

21f8c	11.17	H	FilterBank
21f8e	11.23	H	FilterBank
21f8d	9.93	H	FilterBank
2200b	16.26	H	ValveCAB
227ce	0.057	VL	ValveCAB
226e8	0.174	VL	ValveCAB
21f2a	1.24	VL	500kvClose
21f2b	1.31	VL	500kvClose
12e33	2.077	M	ACDisturb
224a5	2.059	M	ACDisturb
224d0	2.1	M	ACDisturb
12cbe	2.15	M	ACDisturb
12da1	1.85	M	ACDisturb
12dbb	1.65	M	ACDisturb
12e3d	1.73	M	ACDisturb
12dbc	1.73	M	ACDisturb
12dbd	1.57	M	ACDisturb
2344e	1.84	M	PoleLineFlashover
22355	1.42	VL	PoleLineFlashover
12afc	1.15	VL	PoleLineFlashover
12dfe	1.135	VL	PoleLineFlashover
12dfb	1.19	VL	PoleLineFlashover
12e44	1.116	VL	PoleLineFlashover
22424	1.159	VL	PoleLineFlashover
225c4	0.039	VL	PoleLineRetard
12af2	1.1411	VL	PoleLineRetard
12dd2	1.2141	VL	PoleLineRetard
12cb5	1.11	VL	PoleLineFlashover
12cd7	1.1	VL	PoleLineFlashover
12dfb	1.19	VL	PoleLineFlashover
12dfe	1.135	VL	PoleLineFlashover
12e0e	1.15	VL	PoleLineFlashover
22405	1.513	M	Aysm.Prot
224d9	1.49	M	Aysm.Prot
22697	1.689	M	Aysm.Prot
22880	1.77	M	Aysm.Prot
22884	1.77	M	Aysm.Prot
2288b	2.67	M	Aysm.Prot
12dba	1.177	VL	DCDisturb
12e2a	1.2	VL	DCDisturb
13054	1.75	M	Commutation
13062	1.92	M	Commutation
13071	1.548	M	Commutation
226cc	7.747	H	ValveBlip
226db	8.165	H	ValveBlip
226e0	8.48	H	ValveBlip
226e5	11.651	H	ValveBlip
226e7	0.175	H	ValveBlip

In the table, the first column is the fault learning signals and the second column is the ratio of the phase current and current order in each of the fault signals. The third column is the result obtained using the granule algorithm to group the ratios, and the last column is about the types of the signals.

4.4 Fault type classification based on Rough Set

In this section, first we set up a information table based on all the attributes we have got in Chapter **4,** so far we have 10 attributes: WT1,WT2, FFT6P1, FFT6P2, FFT6P3, FFTT1, FFTT2, FFTT3, RATIO, ERROR.

4.4.1 The information Table

Based on the fault signal pre-processing and feature extraction, we got some attributes that are useful for classifying the types of the faults. We can set up an information table for the system, and this information table includes all the attributes we have obtained and also the decision (type of the fault). We use rough set theory to analyze the table (e.g., upper and lower approximations) and use Rosetta to derive fault classification rules. Finally, we can get the rule for each fault. In the information table 11, we have 10 attributes and one decision.

Table 4.7 the information table of all the attributes

files	WT1	WT2	FFT6P1	FFT6P2	FFT6P3	FFTT1	FFTT2	FFTT3	ERROR	RATIO	DECISION
String	Integer	Integer	Integer	Integer	Integer	Integer	Integer	Integer	String	String	String
21f79	1	3	1	2	3	1	7	7	VL	H	RingCounter
21f7a	1	3	1	2	3	1	7	5	VL	H	RingCounter
21f7b	1	3	1	2	3	4	7	4	VL	H	RingCounter
21f7c	1	3	4	2	3	4	7	7	VL	H	RingCounter
21f96	1	3	1	2	3	1	7	5	VL	VL	RingCounter
21f95	1	3	1	2	3	2	7	7	VL	VL	RingCounter
21f8a	1	4	3	5	3	7	7	7	VL	H	FilterBank
21f8b	1	5	6	7	3	7	7	7	VL	H	FilterBank
21f8c	1	5	2	8	3	7	7	7	VL	H	FilterBank
21f8e	1	3	9	2	3	7	7	7	VL	H	FilterBank
21f8d	1	5	9	8	3	2	7	7	VL	H	FilterBank
2200b	6	7	3	4	8	1	4	7	H	H	ValveCAB
227ce	6	7	3	3	8	1	7	7	H	VL	ValveCAB
226e8	2	7	3	3	3	8	7	7	VL	VL	ValveCAB
21f2a	1	1	3	3	12	7	7	2	L	VL	500kvClose
21f2b	1	1	3	3	12	7	2	2	L	VL	500kvClose
12e33	8	8	1	1	1	7	2	2	VH	L	ACDisturb
224a5	8	8	2	2	2	4	4	4	VH	L	ACDisturb
224d0	8	8	2	2	2	4	4	4	VH	L	ACDisturb
12cbe	8	8	3	2	2	2	1	11	VH	L	ACDisturb
12da1	8	2	1	1	1	2	2	2	VH	L	ACDisturb
12dbb	8	2	4	2	2	4	4	4	VH	L	ACDisturb

12e3d	8	8	2	2	2	4	4	4	VH	L	ACDisturb
12dbc	8	2	1	2	2	7	4	5	VH	L	ACDisturb
12dbd	8	8	4	4	3	4	4	4	VH	L	ACDisturb
2344e	1	7	8	8	8	1	1	7	H	L	PoleLineFlashover
22355	9	6	8	8	8	3	4	10	H	VL	PoleLineFlashover
12afc	9	2	1	1	1	3	3	1	H	VL	PoleLineFlashover
12dfe	9	2	8	8	8	2	3	3	H	VL	PoleLineFlashover
12dfb	9	2	8	8	8	4	7	7	H	VL	PoleLineFlashover
1.2e44	2	2	8	8	8	2	3	3	H	VL	PoleLineFlashover
22424	7	2	8	8	8	7	7	1	H	VL	PoleLineFlashover
225c4	7	7	3	3	3	7	7	7	H	VL	PoleLineRetard
12af2	7	7	10	10	10	1	4	1	H	H	PoleLineRetard
12dd2	8	2	11	11	11	1	4	4	H	VL	PoleLineRetard
12cb5	7	2	8	8	8	2	2	2	H	VH	PoleLineFlashover
12cd7	10	2	8	8	8	2	2	2	H	L	PoleLineFlashover
12dfb	9	2	11	8	8	2	2	4	H	H	PoleLineFlashover
12dfe	9	2	1	1	1	3	4	2	H	H	PoleLineFlashover
12e0e	9	2	10	10	11	2	2	2	H	H	PoleLineFlashover
22405	6	10	8	2	2	1	1	4	L	H	Aysm.Prot
224d9	2	10	8	2	2	2	2	2	L	H	Aysm.Prot
22697	6	10	5	4	2	7	8	7	L	H	Aysm.Prot
22880	2	10	11	4	2	2	2	2	L	H	Aysm.Prot
22884	2	10	12	4	7	7	8	7	L	H	Aysm.Prot
2288b	6	10	8	2	2	1	1	1	L	H	Aysm.Prot
12dba	2	2	4	4	12	2	2	2	H	VH	DCDisturb
12e2a	2	2	4	3	3	2	3	3	H	H	DCDisturb
13054	2	5	12	4	3	4	7	7	H	H	Commutation
13062	2	7	12	13	3	2	2	2	H	VH	Commutation
13071	2	7	12	7	3	4	7	7	H	VH	Commutation
226cc	2	5	3	3	3	2	8	7	VL	VH	ValveBlip
226db	6	7	3	3	3	7	7	7	VL	VH	ValveBlip
226e0	6	7	3	3	3	7	3	7	VL	VH	ValveBlip
226e5	2	7	3	3	3	7	7	7	VL	H	ValveBlip
226e7	6	7	3	3	3	7	7	7	VL	H	ValveBlip

In **Table 4.7**, we give all the values of each of the attributes and we will use the rough set theory to analyze the attribute values. So the second row of the table gives the type of the value, and it means the output value is a string or integer or float. This is one of the formats of information table for the rough set tool ROSSETA explained in Chapter **5.**

5 Rough Set Analysis

In the Chapter, first we will give a brief introduction to the rough set theory and then make further analysis on the information table (Table 4.7) to generate the decision partitions and the lower (upper) approximations for this table 5.1.2. We also introduce the membership function of the rough set and use it on the 10 attributes (5.1.3). In section 5.2 we implement the algorithms in the rough set tool ROSSETA to analyze the information and finally get the rules of the system.

5.1 Introduction of Rough Set theory

A rapid growth of interest in rough set theory and its applications has been seen lately. Let IS = (U, A) be an information system where U is set of objects (universe) and A is a set of attributes (e.g., let FFT ∈ A denote a fast Fourier transform). Recall that each attribute a ∈ A is a mapping of the form a: U→ Va. Rough set methodology is based on concept (set) approximations constructed from available background knowledge represented in information system. Each set of attributes B ⊆ A (called a feature set) that is selected reflects our background knowledge (features of experimental data that we know about). In an information system IS, a parameterized family of concept approximations is built. Then by tuning of the rough set model underlying the approximation spaces, improvements in the concept approximations can be obtained. Rough set theory was proposed by Zdzislaw Pawlak as a new approach to knowledge discovery from incomplete data **[6]**. Its approach to processing of incomplete data is based on the lower and the upper approximation of a set. The rough set is defined as the pair of two crisp sets corresponding to these approximations. If both approximations of a given subset of the universe are exactly the same, then one can say an approximated set is completely definable with respect to available information.

5.1.1 Information Systems

A data set is represented as a table. In such a table, each row represents a case, an event, or simply an object. In the information table we got from the FDI system, the objects are the files from the TRS. Every column represents an attribute (a variable, an observation, a property, etc.) that can be measured for each object. In our case they are the characteristics we got from the feature extraction such as the like wavelet type of all the objects and waveform of all the objects. This table is called an *Information System* (which we have created above). In general, an information system is a pair A $= (U, A)$, where U is a non-empty finite set of *objects* called the *universe* and A is a non-empty finite set of *attributes* such that $a: U \rightarrow V_a$ for every $a \in A$. The set V_a is called the *value set* of a.

In our information table, we have 10 attributes, namely, wavelet form of Pole-line Voltages (2 poles), FFT transform of 6 pulse voltages (3 poles), FFT transform of phase current of I_a and also the AC Phase error, Ratio of phase current and current order. So in this research, the set of attributes A={WT1, WT2, FFT6P1, FFT6P2, FFT6P3, FFT1, FFT2, FFT3, ERROR, RATIO}. We have 58 objects that are recovered from the original .x01 file to be detected, U={21f79, 21f7a, 21f7b,etc.}. In the *decision* so far we have 11 type of faults $r(d) = 11$, where r means the rank of the decision. The decision d defines a partition:

$$\{X_1, X_2, X_3 ....X_{11}\} \text{ of } U$$

Where

$X_1 = \{21f79,21f7a,21f7b,21f7c,21f95,21f96\}$, //Fault1

$X_2 = \{21f8a,21f8b,21f8c,21f8d,21f8e\}$ //Fault2

$X_3 = \{2200b,227ce,226e8\}$ // Fault3

$X_4 = \{21f2a,21f2b\}$ //Fault4

$X_5 = \{12e33,224a5,224d0,12cbe,12da1,12dbb,12e3d,12dbc,12dbd\}$ //Fault5

$X_6 = \{23440e,22355,12afc,12dfe,12e33,22424,121cb5,12cd7,12dfb,12dfe,12e0e\} // Fault6$

$X_7 = \{225c4,12af2,12dd2\} // Fault7$

$X_8 = \{22405,224d9,22697,22880,22884,2288b\} // Fault8$

$X_9 = \{12dba,12e2a\} // Fault9$

$X_{10} = \{13054,13062,13071\} // Fault10$

$X_{11} = \{226cc,226db,226e0,226e5,226e7\}$ //Fault11

5.1.2 Lower and Upper Approximations Based on the Information System

So for each type of fault we calculate the lower and upper approximation based on each of the attributes we have. Generally speaking rough set is set up based on a long- term data collection information system. Now we just assume that the information system we have got above has already covered all kinds of fault events and all kinds of outputs of those attributes in different type of faults. In fact that is not the fact because there are some more kinds of faults have occurred, and in this thesis we just give the idea of this algorithm and point out that the approach is also useful for the other signals. In this section we will take one example to discuss how to construct *lower* and *upper* approximation of the rough set. Let us consider the wavelet type of the Pole-line voltage as the example attribute and decision of fault Pole-line Flashover. The construction of lower and upper approximation will take the following 3 steps:

step1: set up a universal set for each type of fault for B(Attribute1): Wavelet

$BDX_{B:wavelet}^{fault1}$ = {x|x= v_a of fault1 based on attribute B:wavelet}={13,13,13,13,13,13}

$\Rightarrow$ equivilence set{13}

$\Rightarrow$ equivilence class ={21f79,21f7a,21f7b,21f7c,21f95,21f96}

$BDX_{B:wavelet}^{fault2}$ = {x|x= v_a of fault2 based on attribute B: wavelet}={14,15,15,13,15}

$\Rightarrow$ equivilence set{13,14,15}

$\Rightarrow$ equivilence class ={21f8a,21f8b,21f8c,21f8d,21f8e}

$BDX_{B:wavelet}^{fault3}$ = {x|x= v_a of fault3 based on attribute B: wavelet}={67,67,27}

$\Rightarrow$ equivilence set{67,27}

$\Rightarrow$ equivilence class ={2200b,227ce,226e8}

$BDX_{B:wavelet}^{fault4}$ = {x|x= v_a of fault4 based on attribute B: wavelet}={11,11}

$\Rightarrow$ equivilence set{11}

$\Rightarrow$ equivilence class={21f2a,21f2b}

$BDX_{B:wavelet}^{fault5}$ = {x|x= v_a of fault5 based on attribute B: wavelet}={88,88,88,88,82,82,88,82,88}

$\Rightarrow$ equivilence set{88,82}

$\Rightarrow$ equivilence class ={12e33,
224a5,224d0,12cbe,12da1,12dbb,12e3d,12dbc,12dbd}

$BDX_{B:wavelet}^{fault6}$ = {x|x= v_a of fault6 based on attribute B:wavelet}={17,96,92,92,92,22,72,72,102,92,92,92}

$\Rightarrow$ equivilence set{17,96,92,72 102}

$\Rightarrow$ equivilence class
={2344e,22355,12afc,12dfe,12dfb,12e44,22424,12cd5,12cd7,12dfb,12dfe,12e0e}

$BDX_{B:wavelet}^{fault7}$ = {x|x= v_a of fault7 based on attribute B: wavelet}={77,77,82}

$\Rightarrow$ equivilence set{77,82}

$\Rightarrow$ equivilence class ={225c4,12af2,12dd2}

$BDX_{B:wavelet}^{fault8}$ = {x|x= v_a of fault8 based on attribute B: wavelet}={610,210,610,210,210,610}

$\Rightarrow$ equivilence set{210,610}

$\Rightarrow$ equivilence class ={22405,224d9,22697,22880,22884,2288b}

$BDX_{B:wavelet}^{fault9}$ = {x|x= v_a of fault9 based on attribute B: wavelet}={22,22}

$\Rightarrow$ equivilence set{22}

$\Rightarrow$ equivilence class ={12dba,12e2a}

$BDX_{B:wavelet}^{fault10}$ = {x|x= v_a of fault10 based on attribute B: wavelet}={25,27,27}

$\Rightarrow$ equivilence set [x]={25,27}

$\Rightarrow$ equivilence class ={13054,13062,13071}

$BDX_{B:wavelet}^{fault11}$ = {x|x= v_a of fault11 based on attribute B: wavelet}={25,67,67,27,67}

$\Rightarrow$ equivilence set[x]={25,27,67}

$\Rightarrow$ equivilence class={226cc,226db,226e0,226e5,226e7}

step 2: Lower and Upper approximation construction in equivalent set measure:

We take the decision of fault 5 and the attribute B: wavelet as the example

Lower approximation in the equal set measure:

$$\underline{BD}X_{B:wavelet}^{fault5} = [x]_{B:wavelet}^{D:fault5} - \sum_{\substack{i=1 \\ i\neq 5}}^{11} ([x]_{B:wavelet}^{D:fault5} \cap [x]_{B:wavelet}^{D:faulti})$$

$$= \{88\} \cup \{82\} - \{82\}$$

$$= \{88\} \text{// only overlap is in fault 7}$$

Upper approximation in the equal set measure:

$$\overline{BD}X_{B:wavelet}^{D:fault5} = \underline{BD}X_{B:wavelet}^{D:fault5} \cup \sum_{\substack{i=1 \\ i\neq 5}}^{11} ([X]_{B:wavelet}^{D:fault5} \cap [X]_{B:wavelet}^{D:faulti})$$

$$= \{82\} \cup \{88\}$$

$$\textit{yes/no set in equal measure} = \sum_{\substack{i=1 \\ i\neq 5}}^{11} ([X]_{B:wavelet}^{D:fault5} \cap [X]_{B:wavelet}^{D:faulti})$$

$$= \{82\}$$

Step3 Recover the lower and upper approximation by using all elements

lower approximation:

$$\underline{BD}X_{B:wavelet}^{fault5} = \{88\}$$

Upper approximation:

$$\overline{BD}X_{B:wavelet}^{fault5} = \{88,82\}$$

Using the 3 steps above, we set up the lower and upper approximation for all the files. This is very helpful for the decision-making, since we can see the *lower approximation* is the character that only this kind of fault has. So if a signal's attribute value belongs to this approximation we can give the fault classification decision right away, otherwise we have to check the other attributes.

5.1.3 The Membership function of Rough Set

In classical set theory, either an element belongs to a set or it does not. The corresponding membership function is the characteristic function for the set, and this function takes values 1 and 0,respectively. In the case of rough sets, the degree of relative overlap between the set X and the equivalence $[x]_B$ class to which x belongs is computed using a rough membership function defined in (5.1). **[4]**.

$$\mu_X^B : U \rightarrow [0,1] \text{ and } \mu_X^B(x) = \frac{|[x]_B \cap X|}{|[x]_B|} \tag{5.1}$$

For example, here we calculate the membership of attribute B: wavelet and the output is 88 in the rough set of $BDX^{D:fault5}_{B:wavelet}$

Step1 find out all elements that are represented by equivalence classes $[x=82]_{B:wavelet}$, here the element refers to the fault file whose output of attribute B is 82

$$\{\text{all elements in } [x=82]_{B:wavelet}\}=\{x|B(x)=82\}=\{82,82\}$$
$$\Rightarrow \{\text{fault\_12da1.fault\_12dd2}\}$$

Step2 calculate the membership of rough set: the upper approximation of rough set

$$BDX^{fault5}_{B:wavelet} : \overline{BDX}^{D:fault5}_{B:wavelet} = \{88,82\}$$

For the output of B(x) with value of 82, its membership with respect to the rough set $BDX^{fault5}_{B:wavelet}$ is given in (5.2).

$$\mu_x^B : U \to [0,1] \tag{5.2}$$

$$\mu_x^B(x) = \frac{card(\{[x=82]_{B:waveli}\} \cap \overline{BDX}^{D:fault5}_{B:wavelet})}{card(\{[x=82]_{B=wavelet}\})} = \frac{card(\{82\})}{card(\{82,82\})} = 0.5$$

Based on the analysis above, we can take fault1 as an example, and Table 12 gives the membership of fault 1 with respect to all the attributes:

Table 5.1 membership of fault1 with respect to all the attributes

Membership of Fault 1 Based on The Different Attributes

	WT1	WT2	FFT6P1	FFT6P2	FFT6P3	FFTT1	FFTT2	FFTT3	ERROR	RATIO	decision
Fault1	0.0714	0.4128	0.01	0.0625	0.0454	0.1	0.0476	0.0434	0.0588	0.0434	1
Fault1	0.0714	0.4128	0.01	0.0625	0.0454	0.1	0.0476	0.33	0.0588	0.0434	1
Fault1	0.0714	0.4128	0.01	0.0625	0.0454	0.1	0.0476	0.11	0.0588	0.0434	1
Fault1	0.0714	0.4128	0.02	0.0625	0.0454	0.1	0.0476	0.0434	0.0588	0.0434	1
Fault1	0.0714	0.4128	0.01	0.0625	0.0454	0.1	0.0476	0.33	0.0588	0.099	1
Fault1	0.0714	0.4128	0.01	0.0625	0.0454	0.066	0.0476	0.0434	0.0588	0.099	1
Fault1	0.0714	0	0	0	0.0454	0	0.0476	0.0434	0.0588	0.0434	0
Fault1	0.0714	0	0	0	0.0454	0	0.0476	0.0434	0.0588	0.0434	0
Fault1	0.0714	0	0	0	0.0454	0	0.0476	0.0434	0.0588	0.0434	0
Fault1	0.0714	0.4128	0	0.0625	0.0454	0.066	0.0476	0.0434	0.0588	0.0434	0
Fault1	0.0714	0	0	0	0	0	0.0476	0.0434	0.0588	0.0434	0
Fault1	0	0	0	0	0	0.1	0	0.0434	0	0.0434	0
Fault1	0	0	0	0	0.0454	0.1	0.0476	0.0434	0.0588	0	0
Fault1	0.0714	0	0	0	0	0	0.0476	0.0434	0	0	0
Fault1	0.0714	0	0	0	0	0	0.0476	0	0	0	0

Fault1	0	0	0.01	0	0	0	0	0	0	0	0
Fault1	0	0	0	0.0625	0	0.1	0	0.11	0	0	0
Fault1	0	0	0	0.0625	0	0.1	0	0.11	0	0	0
Fault1	0	0	0	0.0625	0	0.066	0	0	0	0	0
Fault1	0	0	0.01	0	0	0.066	0	0	0	0	0
Fault1	0	0	0.02	0.0625	0	0	0	0.11	0	0	0
Fault1	0	0	0	0.0625	0	0	0	0.11	0	0	0
Fault1	0	0	0	0.0625	0	0	0	0.33	0	0	0
Fault1	0	0	0.02	0	0.0454	0.1	0	0.11	0	0	0
Fault1	0.0714	0	0	0	0	0.1	0	0.0434	0	0	0
Fault1	0	0	0	0	0	0	0	0	0	0	0
Fault1	0	0	0.01	0	0	0	0	0	0	0	0
Fault1	0	0	0	0	0	0.066	0	0	0	0	0
Fault1	0	0	0	0		0.1	0.0476	0.0434	0	0	0
Fault1	0	0	0	0	0	0.066	0	0	0	0	0
Fault1	0	0	0	0	0	0	0.0476	0	0	0	0
Fault1	0	0	0	0	0.0454	0	0.0476	0.0434	0	0	0
Fault1	0	0	0	0	0	0.1	0	0	0	0	0
Fault1	0	0	0	0	0	0.1	0	0.11	0	0	0
Fault1	0	0	0	0	0	0.066	0	0	0	0	0
Fault1	0	0	0	0	0	0.066	0	0	0	0	0
Fault1	0	0	0	0	0	0.066	0	0.11	0	0.0434	0
Fault1	0	0	0	0	0	0	0	0	0	0.0434	0
Fault1	0	0	0	0	0	0.066	0	0	0	0.0434	0
Fault1	0	0	0	0.0625	0	0.1	0	0.11	0	0.0434	0
Fault1	0	0	0	0.0625	0	0.066	0		0	0.0434	0
Fault1	0	0	0	0	0	0	0	0.0434	0	0.0434	0
Fault1	0	0	0	0	0	0.066	0		0	0.0434	0
Fault1	0	0	0	0	0	0	0	0.0434	0	0.0434	0
Fault1	0	0	0	0	0	0.1	0	0	0	0.0434	0
Fault1	0	0	0.02	0	0	0.066	0	0	0	0	0
Fault1	0	0	0.02	0	0.0454	0.066	0	0	0	0.0434	0
Fault1	0	0	0	0	0.0454	0.1	0.0476	0.0434	0	0.0434	0
Fault1	0	0	0	0	0.0454	0.066	0	0	0	0	0
Fault1	0	0	0	0	0.0454	0.1	0.0476	0.0434	0	0	0
Fault1	0	0	0	0	0.0454	0.066	0	0.0434	0.0588	0	0
Fault1	0	0	0	0	0.0454	0	0.0476	0.0434	0.0588	0	0
Fault1	0	0	0	0	0.0454	0	0	0.0434	0.0588	0	0
Fault1	0	0	0	0	0.0454	0	0.0476	0.0434	0.0588	0.0434	0
Fault1	0	0	0	0	0.0454	0	0.0476	0.0434	0.0588	0.0434	0

In **Table 5.1** we use **(5.2)** to get the membership of fault1 for all the attributes. We can notice that only when all the membership are not zero, the decision is "1", here "1" means true.

For each of the fault which we have detected in the system we set up this kind of rough set and calculate the membership based on all the attributes and at the same time give the decisions. That means so far we have computed 11*10 rough sets for the DFI system described in this thesis.

5.2 The Model Process of Rough Set

Using rough set methods, approximation descriptions of concepts can be constructed from some primitive concepts. It is furthermore well-known that target concept (e.g. decision classes) descriptions defined directly by Boolean combinations of descriptors of the form $a = v$(when a is an attribute and $a \in V_a$) are often not of good approximations quality.

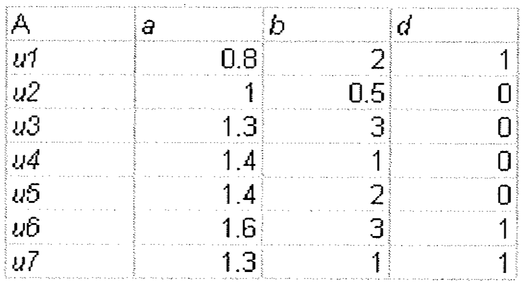
In this section we introduce the algorithms in the rough set tool ROSSETA. These algorithms are used on the information table. First we introduce the idea of discretization. It determines how coarsely we want to view the whole information table (**5.2.1**). After the discretization, we use attribute reduction to get the minimum number of useful attributes that are helpful in making decisions. Finally we use ROSSETA to generate the rules for the whole system.

5.2.1 Information table Discretization

The discretization step determines how coarsely we want to view the whole information table. For each of the attribute which is usually measured in real numbers, this attribute can be discretized into two, three or more, but finitely many, intervals. We can easily see that the selection of appropriate intervals and partitioning of attribute value sets is a complex problem and its complexity can grow exponentially in the number of attributes to be discretized **[6] [7].**

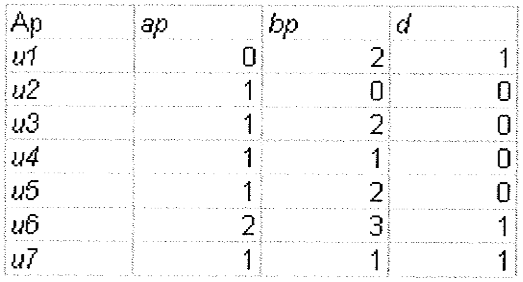
A number of successful approaches to the problem of finding effective heuristics for real value attributes quantization (discretization) has been focused on discretization and symbolic attribute value grouping. What we used in this project is the Boolean Reasoning method on the information table. Discretization problems and symbolic value partition problems are of high computational complexity. So we have to design efficient heuristics. We will concentrate on the basic discretization methods based on the rough set and Boolean Reasoning approaches. In discretization of decision table (Table 13) $\mathrm{A} = (U, A \cup \{d\})$ where $V_a = [v_a, \omega_a)$ is an interval of reals, we search for a partition $P_a$of V_a for any $a \in \mathrm{A}$. Any partition of V_a is defined by sequence of so-called cuts$v_1 < v_2 < v.... < v_k$ from V_a. Hence, any family of partitions $\{P_a\}_{a \in \mathrm{A}}$ can be identified with a set of cuts. In the discretization process we search for a set of cuts satisfying the conditions. This is shown in **Table 5.2.**

Table 5.2 (a) the original decision system



A	a	b	d
u1	0.8	2	1
u2	1	0.5	0
u3	1.3	3	0
u4	1.4	1	0
u5	1.4	2	0
u6	1.6	3	1
u7	1.3	1	1

Table 5.2 (b) P-discretization of A



Ap	ap	bp	d
u1	0	2	1
u2	1	0	0
u3	1	2	0
u4	1	1	0
u5	1	2	0
u6	2	3	1
u7	1	1	1

Where P= {(a,0.9),(a,1.5),(b,0.75),(b,1.5)}

In **Table 5.2 (a)** the first column is the objectand the second and third columns are the attributes' values. The last column is the decision based on the values of the attributes. In **Table 5.2 (b)** the first column is the same as that in (a), the second and third columns are the p-discretization of A, and the last column is the decision.

The set of possible values of *a* and *b* are defines by:

$$V_a = [0,2) ; \quad V_b = [0,4)$$

The set of possible values of *a* and *b* on objects form *U* is given by:

a(U) = {0.8, 1, 1.3, 1.4, 1.6};
b(U) = {0.5, 1, 2,3},

respectively.

We will describe a discretization process that returns a partition of the value sets of conditional attributes into intervals. The partition is done in such a way that if the name of the interval containing an arbitrary object is substituted for any object instead of its original value in A, a consistent decision system is also obtained. In this way the size of value attribute sets in a decision system is reduced. In the above example, the following intervals for condition attributes are obtained:

$$[0.8,1);\ [1,1.3);\ [1.3,1.4);\ [1.4,1.6) \text{ for } a$$
$$[0.5,1);\ [1,2);\ [2,3) \text{ for } b$$

The intervals are defined by the objects in decision system. Cuts are pairs $(a,\ c)$ where $c \in V_a$. We will restrict our considerations for cuts defined by the middle points of the intervals defined above. The following cuts are obtained:

$$(a,\ 0.9);\ (a,1.15);(a,1.35);(a,1.5)$$
$$(b,0.75);\ (b,1.5);(b,2.5).$$

Any cut defines a new conditional attribute with binary values. We use the Johnson strategy. Using this strategy one can look for a cut discerning the maximal number of object pairs (with different decisions), and next one can eliminate all already discerned object pairs and repeat the procedure until all object pairs are discerned. The Boolean reasoning algorithm discretizes the numerical attributes in A according to the discernibility-based multivariate procedure. This produces a set of interval boundaries *Cuts* as a side-effect.

$$Cuts_i = \{(a,\ c) \mid \text{c is a cut for attribute } a \text{ computed by } D_j\}$$

The following **Table 5.3** is the discretized information table:

Table 5.3 Discretized Information Table using Rosetta

files	WT1	WT2	FFT6P1	FFT6P2	FFT6P3	FFTT1	FFTT2	FFTT3	ERROR	RATIO	decision
21f79	[*, 2)	[*, 4)	[*, 5)	2	3	[*, 2)	7	7	{VL}	H	RingCounter
21f7a	[*, 2)	[*, 4)	[*, 5)	2	3	[*, 2)	7	5	{VL}	H	RingCounter
21f7b	[*, 2)	[*, 4)	[*, 5)	2	3	[2, 8)	7	4	{VL}	H	RingCounter
21f7c	[*, 2)	[*, 4)	[*, 5)	2	3	[2, 8)	7	7	{VL}	H	RingCounter
21f96	[*, 2)	[*, 4)	[*, 5)	2	3	[*, 2)	7	5	{VL}	VL	RingCounter
21f95	[*, 2)	[*, 4)	[*, 5)	2	3	[2, 8)	7	7	{VL}	VL	RingCounter
21f8a	[*, 2)	[4, *)	[*, 5)	5	3	[2, 8)	7	7	{VL}	H	FilterBank
21f8b	[*, 2)	[4, *)	[5, *)	7	3	[2, 8)	7	7	{VL}	H	FilterBank
21f8c	[*, 2)	[4, *)	[*, 5)	8	3	[2, 8)	7	7	{VL}	H	FilterBank
21f8e	[*, 2)	[*, 4)	[5, *)	2	3	[2, 8)	7	7	{VL}	H	FilterBank
21f8d	[*, 2)	[4, *)	[5, *)	8	3	[2, 8)	7	7	{VL}	H	FilterBank
2200b	[4, *)	[4, *)	[*, 5)	4	8	[*, 2)	4	7	{H}	H	ValveCAB
227ce	[4, *)	[4, *)	[*, 5)	3	8	[*, 2)	7	7	{H}	VL	ValveCAB
226e8	[2, 4)	[4, *)	[*, 5)	3	3	[8, *)	7	7	{VL}	VL	ValveCAB
21f2a	[*, 2)	[*, 4)	[*, 5)	3	12	[2, 8)	7	2	{L, VH}	VL	500kvClose
21f2b	[*, 2)	[*, 4)	[*, 5)	3	12	[2, 8)	2	2	{L, VH}	VL	500kvClose
12e33	[4, *)	[4, *)	[*, 5)	1	1	[2, 8)	2	2	{L, VH}	L	ACDisturb
224a5	[4, *)	[4, *)	[*, 5)	2	2	[2, 8)	4	4	{L, VH}	L	ACDisturb
224d0	[4, *)	[4, *)	[*, 5)	2	2	[2, 8)	4	4	{L, VH}	L	ACDisturb
12cbe	[4, *)	[4, *)	[*, 5)	2	2	[2, 8)	1	11	{L, VH}	L	ACDisturb
12da1	[4, *)	[*, 4)	[*, 5)	1	1	[2, 8)	2	2	{L, VH}	L	ACDisturb

12dbb	[4, *)	[*, 4)	[*, 5)	2	2	[2, 8)	4	4	{L, VH}	L	ACDisturb
12e3d	[4, *)	[4, *)	[*, 5)	2	2	[2, 8)	4	4	{L, VH}	L	ACDisturb
12dbc	[4, *)	[*, 4)	[*, 5)	2	2	[2, 8)	4	5	{L, VH}	L	ACDisturb
12dbd	[4, *)	[4, *)	[*, 5)	4	3	[2, 8)	4	4	{L, VH}	L	ACDisturb
2344e	[*, 2)	[4, *)	[5, *)	8	8	[*, 2)	1	7	{H}	L	PoleLineFlashover
22355	[4, *)	[4, *)	[5, *)	8	8	[2, 8)	4	10	{H}	VL	PoleLineFlashover
12afc	[4, *)	[*, 4)	[*, 5)	1	1	[2, 8)	3	1	{H}	VL	PoleLineFlashover
12dfe	[4, *)	[*, 4)	[5, *)	8	8	[2, 8)	3	3	{H}	VL	PoleLineFlashover
12dfb	[4, *)	[*, 4)	[5, *)	8	8	[2, 8)	7	7	{H}	VL	PoleLineFlashover
12e44	[2, 4)	[*, 4)	[5, *)	8	8	[2, 8)	3	3	{H}	VL	PoleLineFlashover
22424	[4, *)	[*, 4)	[5, *)	8	8	[2, 8)	7	1	{H}	VL	PoleLineFlashover
225c4	[4, *)	[*, 4)	[5, *)	8	8	[2, 8)	2	2	{H}	VH	PoleLineFlashover
12af2	[4, *)	[*, 4)	[5, *)	8	8	[2, 8)	2	2	{H}	L	PoleLineFlashover
12dd2	[4, *)	[*, 4)	[5, *)	8	8	[2, 8)	2	4	{H}	H	PoleLineFlashover
12cb5	[4, *)	[*, 4)	[*, 5)	1	1	[2, 8)	4	2	{H}	H	PoleLineFlashover
12cd7	[4, *)	[*, 4)	[5, *)	10	11	[2, 8)	2	2	{H}	H	PoleLineFlashover
12dfb	[4, *)	[4, *)	[*, 5)	3	3	[2, 8)	7	7	{H}	VL	PoleLineRetard
12dfe	[4, *)	[4, *)	[5, *)	10	10	[*, 2)	4	1	{H}	H	PoleLineRetard
12e0e	[4, *)	[*, 4)	[5, *)	11	11	[*, 2)	4	4	{H}	VL	PoleLineRetard
22405	[4, *)	[4, *)	[5, *)	2	2	[*, 2)	1	4	{L, VH}	H	Aysm.Prot
224d9	[2, 4)	[4, *)	[5, *)	2	2	[2, 8)	2	2	{L, VH}	H	Aysm.Prot
22697	[4, *)	[4, *)	[5, *)	4	2	[2, 8)	8	7	{L, VH}	H	Aysm.Prot
22880	[2, 4)	[4, *)	[5, *)	4	2	[2, 8)	2	2	{L, VH}	H	Aysm.Prot
22884	[2, 4)	[4, *)	[5, *)	4	7	[2, 8)	8	7	{L, VH}	H	Aysm.Prot
2288b	[4, *)	[4, *)	[5, *)	2	2	[*, 2)	1	1	{L, VH}	H	Aysm.Prot
12dba	[2, 4)	[*, 4)	[*, 5)	4	12	[2, 8)	2	2	{H}	VH	DCDisturb
12e2a	[2, 4)	[*, 4)	[*, 5)	3	3	[2, 8)	3	3	{H}	H	DCDisturb
13054	[2, 4)	[4, *)	[5, *)	4	3	[2, 8)	7	7	{H}	H	Commutation
13062	[2, 4)	[4, *)	[5, *)	13	3	[2, 8)	2	2	{H}	VH	Commutation
13071	[2, 4)	[4, *)	[5, *)	7	3	[2, 8)	7	7	{H}	VH	Commutation
226cc	[2, 4)	[4, *)	[*, 5)	3	3	[2, 8)	8	7	{VL}	VH	ValveBlip
226db	[4, *)	[4, *)	[*, 5)	3	3	[2, 8)	7	7	{VL}	VH	ValveBlip
226e0	[4, *)	[4, *)	[*, 5)	3	3	[2, 8)	3	7	{VL}	VH	ValveBlip
226e5	[2, 4)	[4, *)	[*, 5)	3	3	[2, 8)	7	7	{VL}	H	ValveBlip
226e7	[4, *)	[4, *)	[*, 5)	3	3	[2, 8)	7	7	{VL}	H	ValveBlip

In **Table 5.3** there are 4 attributes be discretized, they are WT1, WT2, FFT6P1 FFTT1. Note that discretization only used on the integer and float, but not on the strings. The ROSSETA Johnson algorithm generate the intervals for the 4 attributes as follows:

WT1 {[*, 2), [2,4), [4, *)}
WT2 {[*, 4), [4, *)}
FFT6P1 {[*, 5), [5, *)}
FFTT1 {[*, 2), [2,8), [8, *)}

At the same time we can notice that the Johnson algorithm also does the reduction on the information table. There are only 5 attributes being used for the decision-making and the others have been ignored. That means those attributes have no help in decisions. Now we use the Reduction algorithm to get the table as we get the reduced information like the following sample:

Table 5.4 Rosetta Reduction Algorithm to get the Reducted Attributes

Reduct	Support	Length
{WT1,WT2,FFT6P1,FFTT1,ERROR}	100	5

The first parameter "Reduct" in **Table 5.4** gives the list of the attributes after the reduction and the second parameter "support" indicates how many decisions have been made correctly based on the attributes in "Reduct", and 100% means we can make the complete decision using these attributes. And the last parameter indicates how many attributes being used.

5.2.2 Rule Generation use Rosetta

After we do the discritization and reduction on the original information table, we can use the rough set tool Rosetta **[6] [17] [18]** for the rule generating.

In table **Table 5.6** it generates the rules for the fault identification. For example, we do the feature extraction on an incoming signal to decide the type of the faults. Now we get the values of the 5 attributes like this **Table 5.5**:

Table 5.5 example attributes values for a signal

WT1	WT2	FFT6P1	FFTT1	ERROR
1	3	4	1	VL

When we use the rules to decide the type of fault of this signal, we find out that its attribute values fall into the rule1, so the fault is Ring Counter. We store the rules for the whole system for fault identification.

Table 5.6 the FDI decision table (Rosetta)

```
WT1([*, 2)) AND WT2([*, 4)) AND FFT6P1([*, 5)) AND FFTT1([*, 2)) AND ERROR({VL}) => DECISION(RingCounter)
WT1([*, 2)) AND WT2([*, 4)) AND FFT6P1([*, 5)) AND FFTT1([2, 8)) AND ERROR({VL}) => DECISION(RingCounter)
WT1([*, 2)) AND WT2([4, *)) AND FFT6P1([*, 5)) AND FFTT1([2, 8)) AND ERROR({VL}) => DECISION(FilterBank)
WT1([*, 2)) AND WT2([4, *)) AND FFT6P1([5, *)) AND FFTT1([2, 8)) AND ERROR({VL}) => DECISION(FilterBank)
WT1([*, 2)) AND WT2([*, 4)) AND FFT6P1([5, *)) AND FFTT1([2, 8)) AND ERROR({VL}) => DECISION(FilterBank)
WT1([4, *)) AND WT2([4, *)) AND FFT6P1([*, 5)) AND FFTT1([*, 2)) AND ERROR({H}) => DECISION(ValveCAB)
WT1([2, 4)) AND WT2([4, *)) AND FFT6P1([*, 5)) AND FFTT1([8, *)) AND ERROR({VL}) => DECISION(ValveCAB)
WT1([*, 2)) AND WT2([*, 4)) AND FFT6P1([*, 5)) AND FFTT1([2, 8)) AND ERROR({L, VH}) => DECISION(500kvClose)
WT1([4, *)) AND WT2([4, *)) AND FFT6P1([*, 5)) AND FFTT1([2, 8)) AND ERROR({L, VH}) => DECISION(ACDisturb)
WT1([4, *)) AND WT2([*, 4)) AND FFT6P1([*, 5)) AND FFTT1([2, 8)) AND ERROR({L, VH}) => DECISION(ACDisturb)
WT1([*, 2)) AND WT2([4, *)) AND FFT6P1([5, *)) AND FFTT1([*, 2)) AND ERROR({H}) => DECISION(PoleLineFlashover)
WT1([4, *)) AND WT2([4, *)) AND FFT6P1([5, *)) AND FFTT1([2, 8)) AND ERROR({H}) => DECISION(PoleLineFlashover)
WT1([4, *)) AND WT2([*, 4)) AND FFT6P1([*, 5)) AND FFTT1([2, 8)) AND ERROR({H}) => DECISION(PoleLineFlashover)
WT1([4, *)) AND WT2([*, 4)) AND FFT6P1([5, *)) AND FFTT1([2, 8)) AND ERROR({H}) => DECISION(PoleLineFlashover)
```

WT1([2, 4)) AND WT2([*, 4)) AND FFT6P1([5, *)) AND FFTT1([2, 8)) AND ERROR({H}) => DECISION(PoleLineFlashover)
WT1([4, *)) AND WT2([4, *)) AND FFT6P1([*, 5)) AND FFTT1([2, 8)) AND ERROR({H}) => DECISION(PoleLineRetard)
WT1([4, *)) AND WT2([4, *)) AND FFT6P1([5, *)) AND FFTT1([*, 2)) AND ERROR({H}) => DECISION(PoleLineRetard)
WT1([4, *)) AND WT2([*, 4)) AND FFT6P1([5, *)) AND FFTT1([*, 2)) AND ERROR({H}) => DECISION(PoleLineRetard)
WT1([4, *)) AND WT2([4, *)) AND FFT6P1([5, *)) AND FFTT1([*, 2)) AND ERROR({L, VH}) => DECISION(Aysm.Prot)
WT1([2, 4)) AND WT2([4, *)) AND FFT6P1([5, *)) AND FFTT1([2, 8)) AND ERROR({L, VH}) => DECISION(Aysm.Prot)
WT1([4, *)) AND WT2([4, *)) AND FFT6P1([5, *)) AND FFTT1([2, 8)) AND ERROR({L, VH}) => DECISION(Aysm.Prot)
WT1([2, 4)) AND WT2([*, 4)) AND FFT6P1([*, 5)) AND FFTT1([2, 8)) AND ERROR({H}) => DECISION(DCDisturb)
WT1([2, 4)) AND WT2([4, *)) AND FFT6P1([5, *)) AND FFTT1([2, 8)) AND ERROR({H}) => DECISION(Commutation)
WT1([2, 4)) AND WT2([4, *)) AND FFT6P1([*, 5)) AND FFTT1([2, 8)) AND ERROR({VL}) => DECISION(ValveBlip)
WT1([4, *)) AND WT2([4, *)) AND FFT6P1([*, 5)) AND FFTT1([2, 8)) AND ERROR({VL}) => DECISION(ValveBlip)

So far we have generated the rules for the fault detection and identification system. Those rules are generated from the 56 learning signals that we have recovered from the TRS of Manitoba Hydro Dorsey Station.

6 PERFORMANCE EVALUATION

Performance evaluation is carried out by measuring the detectability and identifiability of the FDI scheme. It aims at minimizing the error of the system. A C++ program is generated based on the rules. When we do the performance evaluation, first we get the raw data from the TRS and use the FDI system **Figure 3.** to do the type detection. The more signals we use for training, the better result we can get. If the accuracy (identifiability) of one type of fault identification is less than 50%~60%, we have to go back to modify the attributes or try to get more attributes for fault identification.

Performance Evaluation can enhance the knowledge-based system and is helpful in the detection and identification process. The identifiability I_K of a particular fault k depends on how clearly it relates to the rule base and how different it is from other faults. We recovered another 55 faults from the TRS and use them as the training signals. First use the FDI system to get the value of every attributes v_a and then use the rule got from the ROSETTA to make the decision.

Table 6.1 Performance evaluation of 60 training signals

fault type	# of correct	# of incorrect	accuracy(I_k)
Ring counter	5	1	0.83
Filter Bank	4	1	0.8
Valve CAB	4	2	0.67
500kv close	3	1	0.75
AC Disturbance	5	1	0.83
Pole-line Flashover	6	1	0.85
Pole-line Retard	2	1	0.66
Aysm.Prot	4	2	0.67
DC Disturbance	2	1	0.66
Commutation	3	1	0.75
Valve Blip	4	1	0.80

From the above **Table 6.1** we can find out that if we use more signals for learning generally, we can get better results. For example, in the learning step (generating the rules) we use a greater number of signals for the Ring Counter, Filter Bank, ac Disturbance, Pole-line Flash Over, Valve Blip faults compared with the number of signals used for the other types of faults. So the result is better. That tells us that when we generate the rules, we should recover as many signals as possible for learning to optimize the system. The results obtained so far are summarized in Fig. 26.

The **Figure 6.1** is the accuracy output for the system.

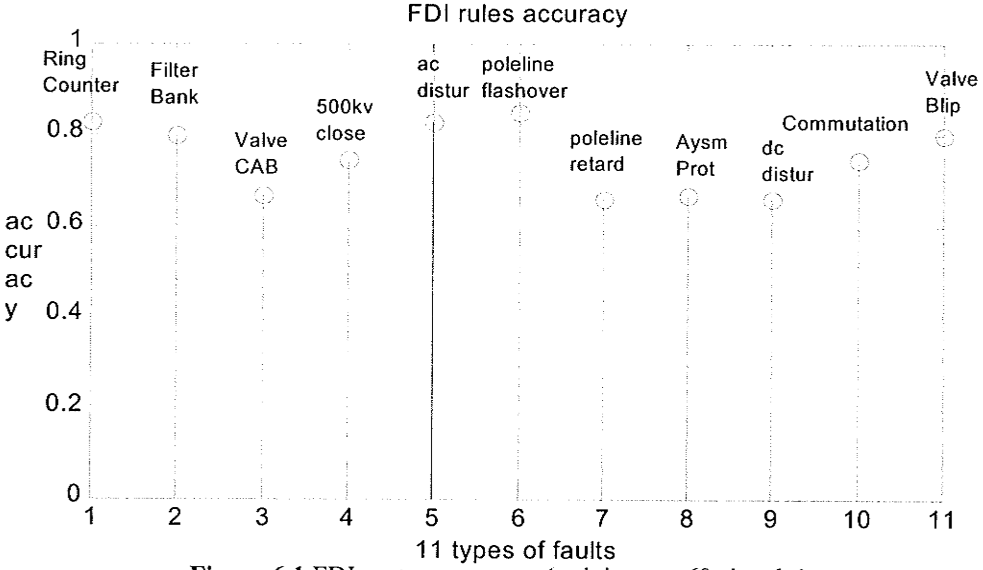


Figure 6.1 FDI system accuracy (training use 60 signals.)

7 USER INTERFACE

For the convenience of the user to use the FDI system, we set up an user interface to implement the system. The interface has the ability to get the data from the TRS of Manitoba Hydro Dorsey Station and then recover the data from binary format into ASSCII format. It also can do the preprocessing on the recovered data. The key part is that it can recall the feature extraction algorithm and finally use the rough set rules to give the decision of which type of fault for a new signal.

Table 7.1 manual for implement the interface

Feature	Explanation	Visual
open	Read raw data from selected directory	**open**
change	recovery the data from binary format(*.x01) to ASSCII format (*.dat)	**change**
check	whether there is fault happened	**check**
Feature Extraction	recall the algorithms to extract the features of the signal	**Feature**
View	view the rules in the system	**view**
Decision	give the decision of the fault type and the FDI system accuracy for this type of fault	**Decision**

When we use the interface, we should get the data first and then recovery them. After that we do the checking of the faults. We integrate the preprocessing procedure inside the Feature Extraction Part. Click on the Feature Extraction button we can get the output value for each of the attribute according to the properties of the signals like (phase current, pole-line voltage, current order and so on). This button will recall another interface for the features, which we will show later. After we get the features of the signal we can click on the View button, this one is used to recall the rules that have already been stored in the system. The rules are those we got using the rough set tool Rosetta as shown in **Table 5.6**. This one helps the user to check the decisions. The Decision button in the interface is used to give the decision of the fault type. At the same time we give the FDI accuracy for this type of fault. The interface is

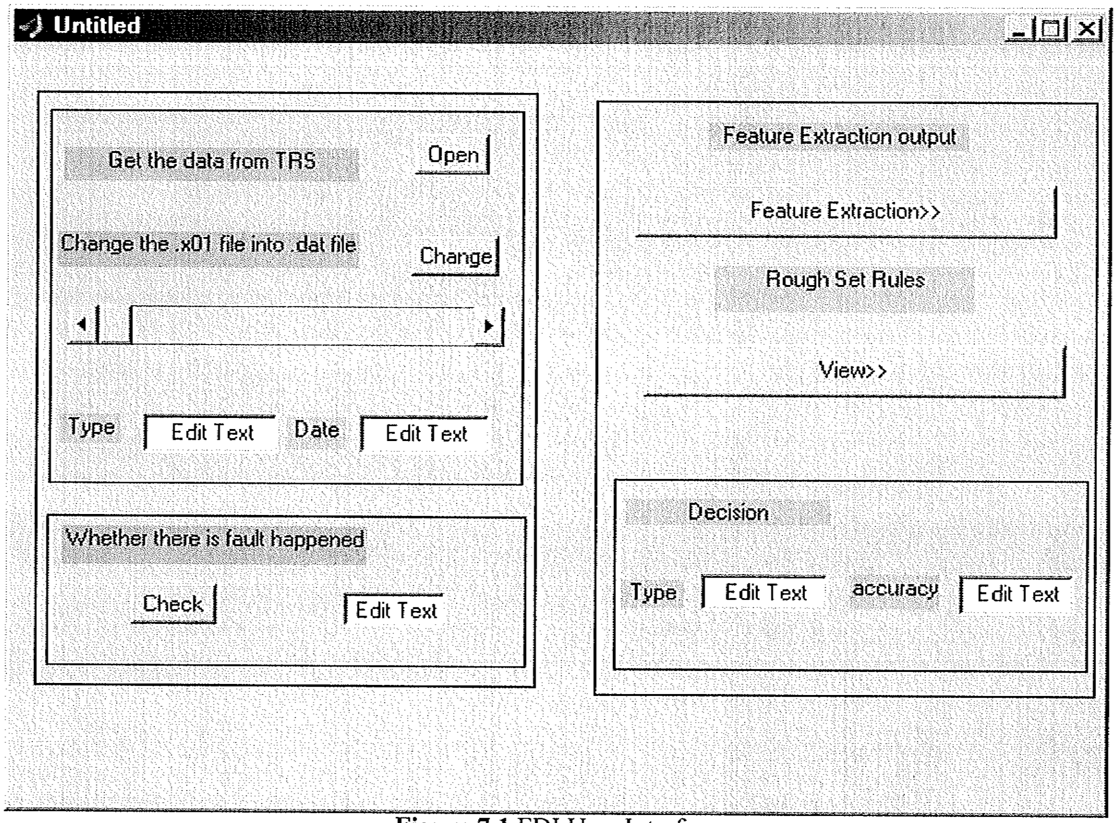


Figure 7.1 FDI User Interface

Now we click on the Open button a list box will show up for the user to choose the file for analysis. **Figure 7.2**

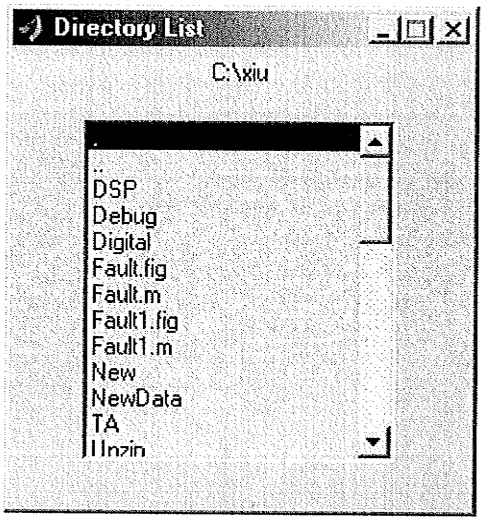


Figure 7.2 Directory list for the data

Then the user selects the file from the directories. Double-click is used to open the file. But in our case, usually we select the *.x01 file for analysis, since it is unreadable to the user. Then click on the Change button, and it will recall the function to recover the data from binary format into ASCII format. For each of the file it will need almost 15 minutes for recovery . After we do the data recovery, the *.dat files will be generated in the same directory with the *.x01 file. Then we check whether there is a fault in this signal using the Check button. If there is a fault, "1" will be returned in the text area, otherwise return "0". If the "0" appears, the user can go back to the directory to select another file to do the detection and identification. If "1" appears, we will go to the feature extraction part of the DFI system. Click on the Feature Extraction button and a new screen will show up as **Figure 7.3**:

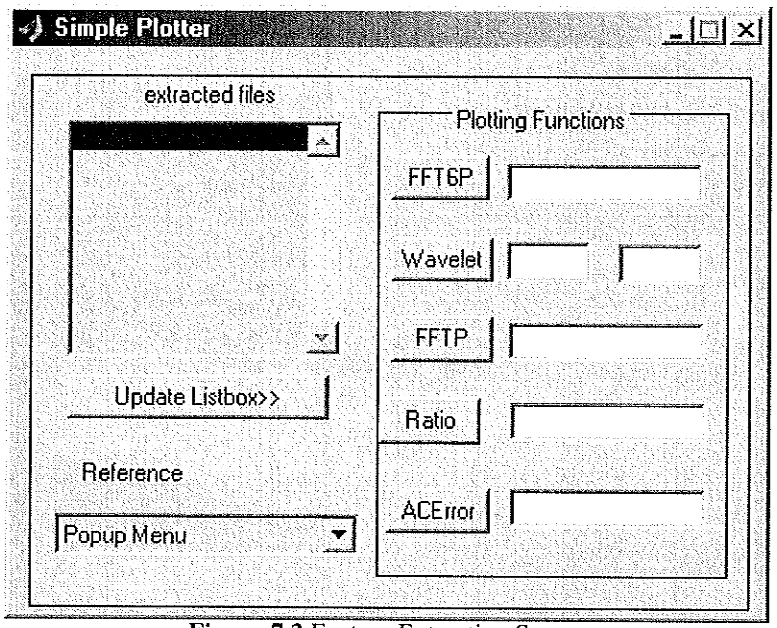


Figure 7.3 Feature Extraction Screen

Here is the manual for the feature extraction screen

Table 7.2 manual for Feature extraction screen

Feature	Explanation	Visual
extracted files	list of the files that have been recovered by the Change function	**Change**
update list	every time when the file been recovered this one is used to update the list box	**change**
Reference:	this tells the user how to choose the functions of different signals	**Popup**
Plotting	do the feature extraction and at the	

Function	same time plot the related output	
FFT6P	FFT for 6 pulse voltages. This plot gives the typical types of FFT in the rule base and also plots the FFT output for the signal you have selected. Return the type.	**FFT6P**
Wavelet	Wavelet transform for the signal. This will return WT1 and WT2	**Wavelet**
FFTP	FFT for phase current a-phase signal. This will plot the typical types of FFT in the rule base and also the FFT for the phase current signal in this file. Return the type in the text area	**FFTP**
Ratio	calculate the ratio of phase current and current order for this file and return the ratio (granule output)	**Ratio**
ac error	calculate the ac voltage phase error of the file and return the error (granule output)	**ACError**

]

After we get all the features we need for the file, we will go back to the main interface **Figure 7.1**. Now click on the View button to check the rules generated by Rosetta, and then click on the Decision button. It will show the decision of the type of the fault and also the accuracy of the FDI system on this type of fault.

8 CONCLUSION

In this thesis, first we present the way to analyze and recover the binary format data of the TRS of Manitoba Hydro Dorsey Station. After an explanation of recovery of the data, we introduced some algorithm for the detection and identification of the faults such as wavelet transform algorithm for the pole-line voltages signals, FFT and low-pass filters used to get the feature of the 6 pulse voltages and the phase current signal. We call this a two-prong approach because we use the wavelet transform on the constant signals while we use the FFT and low-pass filter on the periodic signals. Fuzzy computing is also used in decision-making associated with the wavelet transform and FFT output types. We also calculate the ratio of phase current and current order as well as the ac voltage phase error. Then a granulation algorithm is used for grouping the output values of the ratio and ac error. We "granulate" (group) the data in setting up an information table in what we call a Fault Detection and Identification (FDI) system. Rough set theory has been used to analyze the information table. We also use the rough set tool called Rosetta to generate the rules for the FDI system. We use 60 fault files to do the performance evaluation and determine the accuracy of the system. Finally, a user interface was created for user to implement the system. We have also discovered that the more fault files we use for training the FDI, the greater the accuracy of the classification system.

9 REFERENCES

[1] K. Thorborg, Power Electronics. London: Prentice-Hall, 1988.

[2] K.S. Rajashekara, Power Electronics. In: In: R.C. Dorf (Ed.), The Engineering Handbook. Boca Raton, FL: CRC Press, 1996, 1254-1268.

[3] E. O. Brigham, The Fast Fourier Transform and Its Applications. NJ: Prentice-Hall, 1988.

[4] Y. Sheng, Wavelet transform. In: A.D. Poularikas (Ed.), The Transforms and Applications Handbook. Boca Raton, FL: CRC Press, 1996, 747-828.

[5] L.A. Zadeh, Fuzzy logic = Computing with words, IEEE Trans. on Fuzzy Systems, vol. 4, 1996, 103-111.

[6] Z. Pawlak, Rough Sets: Theoretical Aspects of Reasoning About Data. Boston, MA, Kluwer Academic Publishers, 1991.

[7] J. Komorowski, Z. Pawlak, L. Polkowski, A. Skowron, Rough sets: A tutorial. In: S.K. Pal, A. Skowron (Eds.), Rough Fuzzy Hybridization: A New Trend in Decision-Making. Berlin: Springer-Verlag, 1999, 3-98.

[8] R.W. Swiniarski, L. Hargis, Rough sets as a front end of neural networks texture classifiers, Neurocomputing, vol. 36, Feb. 2001, 85-103.

[9] A. Skowron, R.W. Swiniarski, Information granulation and pattern recognition. In L. Polkowski, A. Skowron, Rough-Neuro Computing. Berlin: Physica-Verlag, 2002 [to appear].

[10] A. Skowron, C. Rauszer, The discernibility matrices and functions in information systems. In: Intelligent Decision Support, Handbook of Applications and Advances of the Rough Sets Theory, Slowinski, R. (Ed.), Dordrecht, Kluwer Academic Publishers, 1992, 331-362.

[11] Z. Pawlak, A. Skowron, Rough membership functions. In: R. Yager, M. Fedrizzi, J. Kacprzyk (Eds.), Advances in the Dempster-Shafer Theory of Evidence, NY, John Wiley & Sons, 1994, 251-271.

[12] E.W. Kimbark, Power System Stability, vol. 1: Elements of Stability Calculations. NY: John Wiley & Sons, 1948.

[13] R. Broadwater, A. Sargent, R.E. Lee, Power distribution. In: R.C. Dorf (Ed.), The Engineering Handbook. Boca Raton, FL: CRC Press, 1996, 1167-1175.

[14] J.F. Peters, A. Skowron, J. Stepaniuk, Information granule decomposition, *Fundamenta Informatica*, 2001 [to appear].

[15] J.F. Peters, A. Skowron, Z. Suraj, W. Pedrycz, N. Pizzi, S. Ramanna, Classification of meterological weather data: A rough set approach, *Pattern Recognition Letters*, 2001 [to appear].

[16] J.F. Peters, S. Ramanna, M. Borkowsky, A. Skowron, Z. Suraj, Sensor, filter and fusion models with rough Petri nets, *Fundamenta Informatica*, vol. 34, 2001, 1-19.

[17] J.F. Peters, A. Skowron, A rough set approach to knowledge discovery, *International Journal of Intelligent Systems*, 2001 [to appear].

[18] J.F. Peters, A. Skowron, A rough set approach to reasoning about data. *International Journal of Intelligent Systems*, vol. 15, 2001, 1-2.

[19] J.F.Peters, L.Han, S.Ramanna, Rough neural computing in signal analysis, *Computational Intelligence*, vol. 1, no. 3, 2001, 493-513.

[20] A. Skowron, J. Stepaniuk, J.F. Peters, Discovering patterns in information granules. In: S. Hirano, M. Inuiguchi, S. Tsumoto (Eds.), *Rough Sets and Granular Computing*. Berlin: Physica Verlag, 2002 [to appear].

[21] A. Skowron, J. Stepaniuk, J.F. Peters, Extracting patterns using information granules: A brief introduction. In: S. Hirano, M. Inuiguchi, S. Tsumoto (Eds.), *Lecture Notes in Artificial Intelligence*, 2001 [to appear].

[22] S.K. Pal, L. Polkowski, J. Peters, A. Skowron, Rough neurocomputing: An Introduction. In: S. Pal, A. Skowron (Eds.), *Rough-Neuro Computing*, ch. 7. Berlin: Springer Verlag [to appear].

[23] L. Han, J.F. Peters, S. Ramanna, R. Zhai, Classifying faults in high voltage power systems: A rough-fuzzy neural computational approach. In: N.Zhong, A. Skowron, S. Ohsuga (Eds.), New Directions in Rough Sets, Data Mining, and Granular-Soft Computing, *Lecture Notes in Artificial Intelligence*, vol. 1711. Berlin: Springer-Verlag, 1999, 47-54.

[24] J. F. Peters, W. Pedrycz, Computational Intelligence. In: J.G. Webster (Ed.), *Encyclopedia of Electrical and Electronic Engineering.* 22 vols.NY, John Wiley & Sons, Inc., 1999.

[25] Z. Suraj, J.F. Peters, W. Rzasa, A comparison of different decision algorithms used in volumetric storm cell classification. In: In: H.D. Burkhard, L. Czala, H.S. Nguyen, P. Starke (Eds.), *Proc. of the Workshop on Concurrency, Specification and Programming*, Oct. 2001, Warsaw, Poland [to appear].

[26] A. Skowron, J. Stepaniuk, J.F. Peters, Extracting patterns using information granules. In: S. Hirano, M. Inuiguchi, S. Tsumoto (Eds.), *Proc. of Int. Workshop on Rough Set Theory and Granular Computing* (RSTGC'01), Matsue, Shimane, Japan, 20-22 May 2001, 135-142.

[27] L. Han, R. Menzies, J.F. Peters, L. Crowe, High Voltage Power Fault-Detection and Analysis System: Design and Implementation. In: *Proc. Canadian Conf. on Electrical and Computer Engineering* (CCECE'99),, Edmonton, Alberta, May, 1999.

[28] L. Han, R. Menzies, J.F. Peters, L. Crowe, High Voltage Power Fault-Detection and Analysis System: Design and Implementation. In: *Proc. Canadian Conf. on Electrical and Computer Engineering* (CCECE'99),, Edmonton, Alberta, May, 1999.

[29] A. Skowron, R.W. Swiniarski, Information granulation and pattern recognition. In: L.Polkowski, A. Skowron (Eds.), *Rough-Neuro Computing*. Berlin: Springer-Verlag, 2002.

[30] Skowron, J. Stepaniuk, J.F. Peters, Approximation of information granule sets. In: W. Ziarko, Y. Yao (Eds.), *Rough Sets and Current Trends in Computing* (RSCTC'2000). Banff, Alberta, Canada, 2000, 33-39.

[31] Mufi Mudi, Vachtsevanos George, Automated Fault Detection and Identification Using a Fuzzy Analysis Technique.

[32] Rosetta knowledge-discovery tool, http://www.idi.ntnu.no/~aleks/rosetta/).

[33] A. Antoniou, Digital Filters: Analysis, Design, and Applications. Bombay: McGraw-Hill, 1993.