WAVELET BASED ANALYSIS OF CIRCUIT BREAKER OPERATION

A Thesis

by

ZHIFANG REN

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

MASTER OF SCIENCE

May 2003

Major Subject: Electrical Engineering

WAVELET BASED ANALYSIS OF CIRCUIT BREAKER OPERATION

A Thesis

by

ZHIFANG REN

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

MASTER OF SCIENCE

Approved as to style and content by:

Mladen Kezunovic
(Chair of Committee)

Ali Abur
(Member)

Krishna Narayanan
(Member)

William Lively
(Member)

Chanan Singh
(Head of Department)

May 2003

Major Subject: Electrical Engineering

ABSTRACT

Wavelet Based Analysis of Circuit Breaker Operation. (May 2003)
Zhifang Ren, B.S., Tianjin University, China
Chair of Advisory Committee: Dr. Mladen Kezunovic

Circuit breaker is an important interrupting device in a power system network. It usually has a lifetime of about 20 to 40 years. During breaker's service time, maintenance and inspection are imperative duties to achieve its reliable operation. To automate the diagnostic practice for circuit breaker operation and reduce the utility company's workload, Wavelet based analysis software of circuit breaker operation is developed here. Combined with circuit breaker monitoring system, the analysis software processes the original circuit breaker information, speeds up the analysis time and provides stable and consistent evaluation for the circuit breaker operation.

ACKNOWLEDGMENTS

The author acknowledges Dr. Mladen Kezunovic, Dr. Zijad Galijasevic, Chris Nail, Goran Latisko and CenterPoint Energy Staff members John Lucey, Don Sevcik, Wayland Cook, Hillary Vicker, and Ed Koch for their advice and contributions to the development of the software.

TABLE OF CONTENTS

Page

ABSTRACT III

ACKNOWLEDGMENTS IV

TABLE OF CONTENTS V

LIST OF TABLES VII

LIST OF FIGURES VIII

CHAPTER

I INTRODUCTION 1

- Introduction 1
- Circuit Breaker 1
- Condition Monitoring 2
- Conclusion 5

II PROBLEM DEFINITION 6

- Introduction 6
- Choice of Monitored Data 6
- Proposed Solution 10
- Conclusion 10

III WAVELET THEORY 11

- Introduction 11
- Continuous Wavelet Transform 11
- Filter Banks 13
- Filter Construction 14
- Wavelets Application 14
- Conclusion 18

IV SIGNAL PARAMETERS 19

- Introduction 19
- Features and Signal Parameters 19
- Conclusion 30

Page

CHAPTER

V FEATURE EXTRACTION 31

- Introduction 31
- Utility Functions 33
- Event Features 36
- Event #3 (DIP Time) 39
- Other Features 43
- Conclusion 49

VI SOLUTION ARCHITECTURE 50

- Introduction 50
- Functional Requirements 51
- Class Diagram 52
- Sequence Diagram 57
- Conclusion 60

VII SOLUTION EVALUATION 61

- Introduction 61
- Test Procedure 61
- Test Data 63
- Test Results 66
- Results Evaluation 72
- Conclusion 74

VIII CONCLUSION 76

- Summary of Work 76
- Contribution 76
- Future Work 77

REFERENCE 78

APPENDIX 80

VITA 83

LIST OF TABLES

TABLE Page

I Performance Indicators of Circuit Breaker 8

II Events Definition for CB Performance Indictors 20

III Features for Individual Signals 21

IV Utility Functions 32

V Extraction Functions 33

VI Inputs and Outputs for Utility Functions 36

VII Algorithms List for Event Feature Extraction 37

VIII Settings for Event #3 Feature Extraction Algorithm 42

IX Settings for Contact Feature Extraction 43

X Settings for DC Voltages Feature Extraction 45

XI Settings for Coil Current Parameter Extraction 48

XII Summary of Test Cases I 64

XIII Summary of Test Cases II 65

XIV Selected Test Cases 66

XV Results for Selected Cases 67

XVI Settings and Tolerance for Selected Cases 68

XVII Continued Results for Selected Cases 69

XVIII Test Results for 18 Types of Breakers 70

XIX Test Results for GE VIB-15.5-20000-2 Circuit Breaker 74

LIST OF FIGURES

FIGURE Page

1. Traditional maintenance strategies 3
2. Condition monitoring system for CB 4
3. Failure rate for different CB components 7
4. CB control circuit 8
5. Automated analysis system for CB operation 10
6. Example of STFT and Wavelet basis 12
7. Time-frequency window for STFT and Wavelets 13
8. Filter bank decomposition algorithm 14
9. Coil current collected from in-service CB 15
10. Signal de-noising example 16
11. Signal splitting example 18
12. Event features for a CB closing operation 20
13. Voltage drops of DC voltage 22
14. Ripple feature of DC voltage 23
15. Distortion & spike feature of DC voltage 24
16. The cooperation of coil and breaker mechanism 25
17. Events features of coil current 26
18. Choppy feature of coil current 26
19. DIP feature of contact 27
20. Noise & bounce feature of contacts 28
21. Event feature of phase currents 29

FIGURE Page

22. Restrike feature of phase current 29
23. Event features of "X" & "Y" coils 30
24. Utility and extraction functions 31
25. Trip coil current example 34
26. Flowchart for function "rise time" 35
27. Flowchart for function "ExtractTIM" 38
28. Dip features for different coil current 39
29. Use Wavelets decomposition algorithm to locate the DIP point 40
30. Flowchart for function "ExtractTIM3" 41
31. DIP time extraction 42
32. Flowchart for function "ExtractDIP" of DC voltage 44
33. Setting "Time Tolerance" 45
34. PSD for signals with ripple feature 46
35. PSD signals without ripple feature 46
36. Wavelet decomposition at level one for extracting distortion parameter 47
37. Algorithm for ExtractSPI 48
38. Initial class diagram for class "Contact" and "Signal" 53
39. Class diagram for "Contact" and "Signal" with attributes and functions 54
40. Final class diagram for the signal classes 55
41. Inheritance diagram 56
42. Composition diagram 58
43. Sequence diagram 59

FIGURE Page

44. First stage test procedure 62

45. Second stage test procedure 62

46. Results for sequence testing 68

47. Coil current of GE VIB-15.5-20000-2 circuit breaker 73

CHAPTER I

INTRODUCTION

Introduction

High voltage circuit breaker (CB) plays a crucial role in power system transmission, distribution and protection. It is a complex device; functional, mechanical and electrical descriptions are provided first in this chapter to introduce the different aspects of CB design and operation. It is a demanding device; maintenance strategies are imperative to guarantee its reliability. This chapter next reviews traditional maintenance strategies, introduces the condition monitoring concept and its present application in CB. Automated CB condition analysis is a new development in the CB maintenance area. It extends CB condition monitoring by using advanced data processing and analyzing. This chapter concludes that condition analysis will further help to build a maintenance decision-support solution.

Circuit Breaker

In the IEEE standard, circuit breaker (CB) is defined as: " A mechanical device capable of making, carrying and breaking currents under normal circuit conditions and also making, carrying for a specific time and breaking currents under specified abnormal circuit conditions such as those of short circuit [1]." This definition gives a functional description of CB. It is required that breaker not only works reliably in a normal power system condition, but also helps in performing a protective role in abnormal power system conditions.

CB reliability has explicit meaning, i.e., breaker should operate on command, and should not operate without a command. In order to achieve the requirements, CB is carefully designed to be a combination of mechanical and electrical parts. In mechanical view, CB can be simplified as a contact that can physically move to make or break a circuit current. No matter what mechanisms are used, spring, pneumatic or hydraulic,

---

This thesis follows the style and format of *IEEE Transactions on Power System*.

they all serve one purpose: to provide the means for opening and closing the contacts of circuit breaker [2]. In electrical view, CB involves two circuits of different voltage levels. At the high voltage level (primary current circuit), CB can be taken as a switch in the power system carrying alternating currents. The low voltage level circuit (secondary current circuit) refers to the control circuit that powers the mechanisms of CB so that energy can be stored and released to move the contacts. This control circuit carries direct current. A detailed functional, mechanical and electrical description of a CB can be found in the thesis of C. Nail [3].

CB usually has a lifetime about 20 to 40 years. In most of its lifetime circuit breaker remains closed and occasionally it is called upon to open and interrupt the current. Some parts of the CB will wear out due to its operation. For instance, the contact material will erode due to the arcing generated when the contact opens and closes. Some parts of the CB will deteriorate due to its inertia. For instance, the increased friction of the mechanical parts may imply lack of proper lubrication. These conditions may develop into failures that will prevent CB to perform its functions. In IEEE standard, CB failures can be classified as major failure (MF) and minor failure (mf). Major failures of a CB "cause the termination of one or more of its fundamental functions, which necessitate immediate action [4]." Minor failure refers to "any fault of a part or a sub-assembly that does not cause a major failure of a circuit breaker [4]."

Condition Monitoring

During CB's service time, maintenance and inspection are imperative duties to detect early symptom of a failure and achieve CB reliable operation. Fig. 1 shows two common maintenance strategies. Assume that the curve in the figure represents one pattern of CB deterioration rate versus time. At certain point, CB's condition exceeds its limitation, which is called breakdown. The breakdown maintenance takes place only after the major failure of CB. Under this circumstance, either an expensive overhaul or a complete replacement is inevitable. It is already concluded that such a strategy is dangerous since all the other equipments around CB maybe affected, and as a consequence may incur unbearable cost.

Fixed time interval maintenance strategy schedules frequent visit to the substation in order to detect any potential problems well before the breakdown point occurs. This preventive strategy may prolong CB's lifetime, but it does introduce unnecessary cost incurred from the increased number of the maintenance. With the increase in the number of CB installations, the aging of the in-service CBs and the decrease in the manpower, this is hard to achieve.

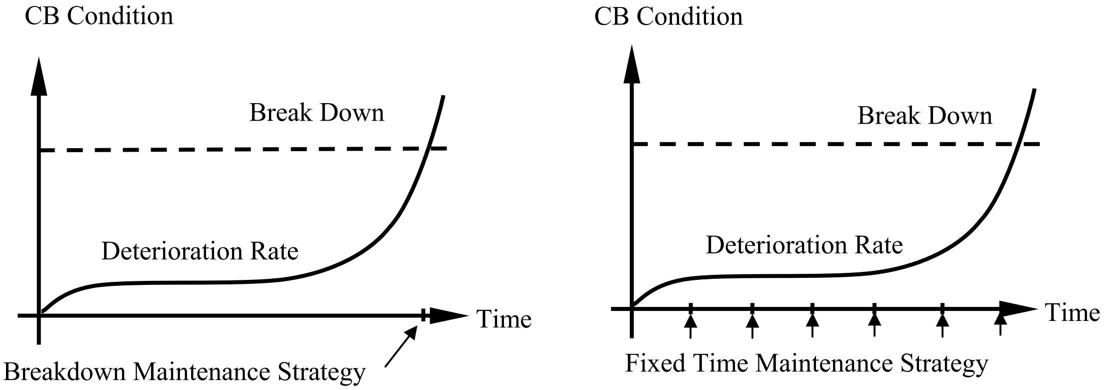
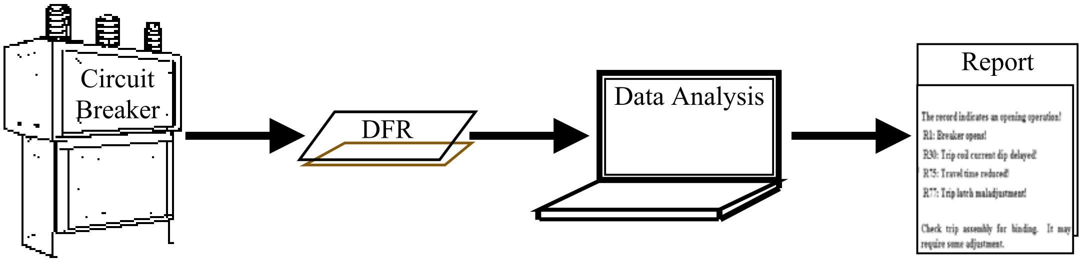


Fig. 1. Traditional maintenance strategies

The cost incurred from CB's breakdown and the cost incurred from the increased maintenance seems to be a dilemma that is hard to overcome. People are looking for an optimal solution, which can balance the costs so that the total expense is reduced to a minimum.

Condition monitoring is initiated from the fact that most equipment will have a useful life before maintenance is required. It acquires the data related to the equipment status and analyzes the data to predict the trend of the equipment deterioration. Maintenance and inspection will be performed according to the analysis result: the condition of CB. The advantages of this predictive strategy are obvious. It improves the reliability of the circuit breaker by providing just-in-time maintenance and at the same time it may reduce the maintenance costs by optimizing the schedule[5]. Of course, condition monitoring is not a free task. Data acquisition unit to collect the original data and advanced analysis system to process the data need to be developed. But these are of lesser cost compared with the cost of a high voltage CB.

Fig. 2 provides an example of a circuit breaker condition monitoring system. The system contains a circuit breaker, a digital fault recorder (DFR) used to collect data from circuit breaker, and a laptop computer used to process the data. Data analysis software resides in the computer. It acquires the original data collected by the DFR and transforms the data to readable and easily understood information. The analysis results generated by the computer will be a final report about the CB operation condition, and user can either review the results displayed on the screen or print it out.



The record indicates an opening operation!
R1: Breaker opens!
R30: Trip coil current dip delayed!
R75: Travel time reduced!
R77: Trip latch maladjustment!

Check trip assembly for binding. It may require some adjustment.

Fig. 2. Condition monitoring system for CB

Digital fault recorder (DFR) is a data acquisition device widely used in the power system [6]. DFRs are used to capture fault transients and changes in records (shots) of associated control and communication equipment. In this special application, it is connected to the control circuit of the breaker and monitors the voltages and currents in the control circuit when breaker operates. Some DFRs provide multiple communication choices such as serial ports, parallel ports, modem, phone line sharing and network protocols. The flexibility in communication choices allows users to scale each application according to individual requirements. For example, one can monitor the breaker condition either locally by connecting the DFR with a portable laptop or remotely by connecting the DFR to Ethernet.

Conclusion

Condition monitoring has been applied to transformer, circuit breaker and other power apparatus in the power industry. The obvious befits are listed below:

- Supplement inspection
- Improve maintenance and control quality
- Increase system reliability
- Provide information for the management decision
- Reduce the damage cost incurred from failures
- Reduce inspection and maintenance costs (Cost Assessment)

A cost benefit evaluation is inevitable before a full-fledged implementation.

At present, most utilities within USA have not fully implement condition monitoring into their everyday maintenance strategy although some have applied the technique to aid their routine maintenance activities [7]. One of the major reasons is that lots of judgments and experience are still needed to analyze the data being collected by a condition monitoring system.

Automated CB condition analysis is a new development in the CB maintenance area. It extends the CB condition monitoring concept by using advanced data processing and analyzing, which further helps in building a maintenance decision-support solution.

CHAPTER II

PROBLEM DEFINITION

Introduction

Condition monitoring and its application are gaining attention as a circuit breaker (CB) maintenance choice. Noteworthy efforts are devoted to developments of related techniques and devices [8][9][10]. Real experience with such development from a utility company shows the improvements in routine maintenance and inspection given that staff is properly trained [11].

Condition monitoring for CB has not been fully incorporated into the everyday maintenance procedure yet. Large number of CBs under monitoring, lack of expertise in advanced data analysis, lack of data repository to provide useful statistical results and lack of decision-making tools may all contribute to this situation. One major problem is that the original data being monitored and recorded by the system is usually hard to interpret. The traditional data analysis heavily depends on an individual's experience, and any false conclusion may lead to improper maintenance action. Furthermore, the profile of monitoring data depends upon the type of CB, operating environment and time interval since CB got into service. The data is changing gradually with the increase of CB operations, which increases the difficulty in analyzing it.

The objective of the system is to use signal processing techniques to automate the analysis process.

Choice of Monitored Data

International Council on Large Electric Systems (CIGRE) plays an important role in guiding the choice of monitored parameters. It has conducted two global-wide surveys on the reliability of circuit breakers. The results show that "the operating mechanism is the subassembly that is responsible for most of the failures. After the operating mechanism, the next most frequent subassembly reported as being responsible for the failure was the electrical control and auxiliary circuit (Fig. 3 covers the single pressure SF6 gas CBs placed after 1 January 1978 and with a rated voltage of 72.5 KV and above and service voltage greater than 63 KV) [12]."

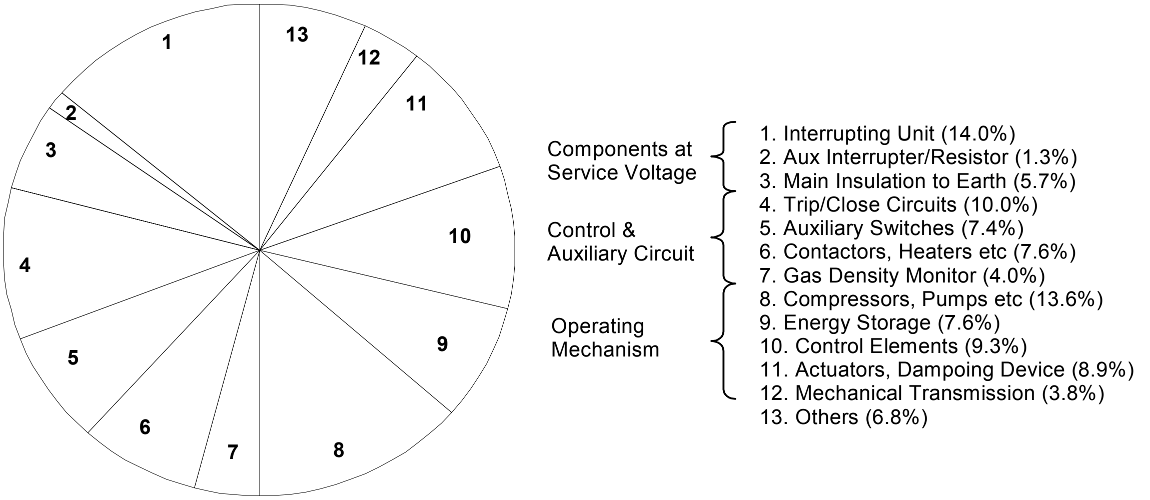


Fig. 3. Failure rate for different CB components

This thesis selects a widely used portable CB testing device as the data acquisition unit and uses the CB performance indicators as the monitored data. The selected testing device can be connected to the CB's control circuit to record analog and digital signals called performance indicators. The operator opens and closes the circuit breaker each time the test is performed and makes the recordings. The traditional analysis is done manually by overlaying traces from a "reference" case recorded earlier and making a judgment of how different the new case is.

Performance indicators contain the signals listed in Table I. They can be used to analyze the performance of the control circuit and part of the operating mechanism, whose failures account for one third of total CB failures. For CB opening operation, seven signals are used to analyze its condition while in its closing operation two more signals are needed to analyze its condition. The control circuit, shown in Fig. 4, is a simplified system and it can be divided into several sub-systems, open and close are two major sub-systems. Usually, a heater and motor will also be included. The subsystem for closing is more complex in a circuit design compared with the one for opening because it needs to prevent multiple-close attempts in one closing operation and prepare for a free opening operation after each closing operation.

TABLE I
PERFORMANCE INDICATORS OF CIRCUIT BREAKER

Open Operation	Close Operation
Trip Initiate	Close Initiate
Control DC Voltage	Control DC Voltage
Yard DC Voltage	Yard DC Voltage
A Contact	A Contact
B Contact	B Contact
Trip Coil Current	Close Coil Current
Phase Currents	Phase Currents
	X Coil
	Y Coil

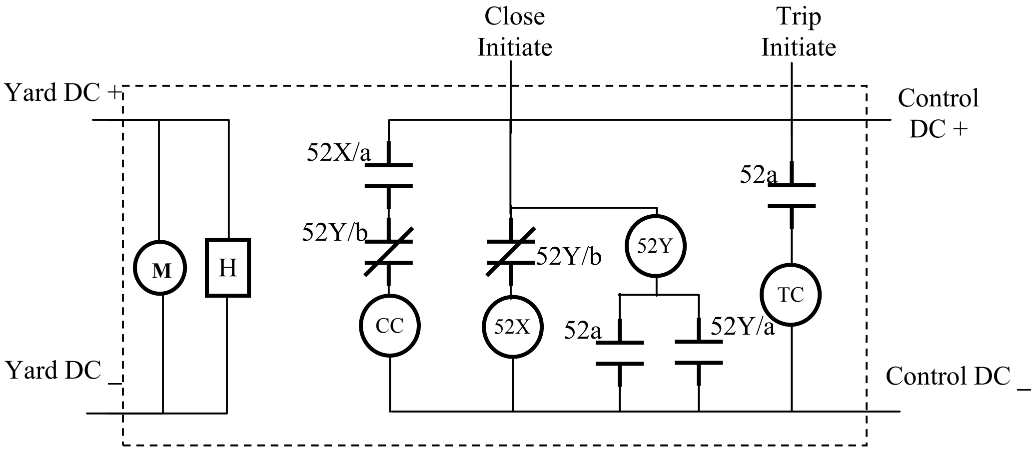


Fig. 4. CB control circuit

Control DC provides DC voltage to both "trip" and "close" subsystems (right part of Fig. 4), and yard DC provides DC voltage to the motor or heater (left part of Fig. 4). DC voltage comes from the substation batteries. Loss of battery or any other problems with the batteries may directly affect the control circuit, and consequently, breaker will not be able to operate at all.

Coming from an operator in the control house, SCADA, relay, or other source, the "initiate" signal is sent to CB to start its operation. The "initiate" signal transition from "OFF" to "ON" indicates the start of the CB operation. Loss of the "initiate" signal indicates a major problem in the sequence of relay operation.

Trip and close coil currents (TC or CC in Fig. 4) are most important signals being monitored. Each trip and close coil encloses a plunger that belongs to the operating mechanism. When trip or close coil are energized, the electric-magnetic force on the plunger initiates the movement of the whole mechanism. The movement of operating mechanism is reflected in the coil currents through the electric-magnetic effect. Depending upon the mechanical and electrical nature of CB, the current takes different waveforms. Monitoring of coil currents provides insights into the condition of both the coil and operating mechanism.

"A" and "B" contacts signals (52a and 52b) indicate the voltage across auxiliary switches, which signify the open or close status of CB. They are designed such that there is always a time difference between changing of "A" contact and "B" contact. The reciprocal of that time difference is proportional to the velocity of CB operation. A deformation of the signal may indicate a dirty contact, a binding mechanism, or a slow breaker, etc.

Measurements of phase currents interrupted by the CB are the only signals not collected from the control circuit. They show exactly whether breaker makes or breaks the currents. This seemingly redundant information is actually very useful. Putting it together with the other signals from control circuit (e.g., "A" and "B" contact, trip and close coil currents) allows a confirmative check. Any inconsistency may indicate a wrong cable connection or a problem with the control circuit, or operating mechanism. If phase currents show that CB operated correctly, but one of the contacts is not sensing the correct status of CB, then one can conclude that there is a problem with the contact.

Only the closing operation has "X" and "Y" coil signals. As mentioned before they are used to prevent multiple-close attempts in one closing operation. "X" and "Y" coil have also only "ON" or "OFF" status. Loss of "X" or "Y" coil signal will require a check of them and the circuit around them.

The efforts of this thesis are to design algorithm to analyze the mentioned performance indicators, and detect problems with CB from the signal analysis results.

Proposed Solution

The proposed automated condition analysis solution can be divided into two sub-systems shown in Fig. 5. The first sub-system uses data processing techniques to "translate" the original data into certain input data that an expert system can understand. The second sub-system then reads the input data, analyzes it and summarizes the results into a language a human being can understand. The overall solution is in a sense analogous to the human analysis. Instead of human observation, the data pre-processing sub-system (first box in Fig. 5) utilizes signal-processing techniques like Wavelets to extract important information from the data. Instead of human decision, the diagnostic sub-system (second box in Fig. 5) utilizes Expert System to process the information received from pre-processing and provides a concise conclusion about the circuit breaker operational condition.

The aim of the work reported in the thesis is to develop data pre-processing module while the expert system development has already been covered in reference [3].

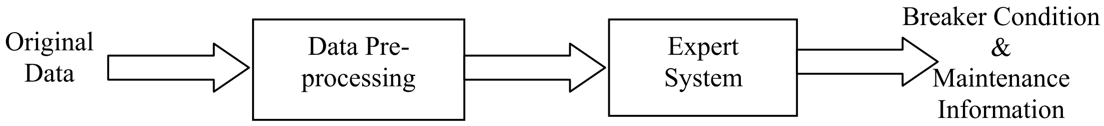


Fig. 5. Automated analysis system for CB operation

Conclusion

This chapter points out the efforts of this thesis, which is to design algorithm to analyze the mentioned performance indicators and detect problems with CB from the signal analysis results, and proposes the architecture for the automated CB condition analysis solution.

CHPATER III

WAVELET THEORY

Introduction

As mentioned above, the original data collected from the circuit breaker (CB) cannot be directly presented to "any learning, discovering, or visulizing algorithm [12]" such as expert system, neural network before it is pre-processed. Data pre-processing is an important step and it is the focus of this work.

In this data pre-processing application, Wavelets algorithm helps to obtain useful information from the original data collected from CB control circuit. It is beneficial to review the theory in order to understand fully how it works. Instead of providing an overall discussion of Wavelets, the following sections cover only basic yet important ideas in the theory and application. The discussion also tries to introduce the Wavelets from different points of view.

Continuous Wavelet Transform

Wavelets theory has been developed as a unifying framework in mid-eighties, although similar ideas and constructions took place as early as the beginning of the century. Researchers lead by Morlet, Grossmann, and Meyer, built strong mathematical foundations around the subject and named their work "Ondelettes" (Wavelets) [13].

Wavelets theory is easier to understand when being compared to Fourier analysis. The following three equations are Fourier Transform, Short-Time Fourier Transform and Wavelet Transform respectively. These transforms are similar in their definition. That is, they are the convolutions (inner products) between the original function $f(t)$ and their respective base function, such as $e^{j\omega t}$.

$$\hat{f}(\omega) = \left\langle f(t), e^{j\omega t} \right\rangle = \int_{-\infty}^{+\infty} f(t) \cdot e^{-j\omega t} dt$$

$$G_{\phi} f(b,\xi) = \left\langle f(t), \phi_{b,\xi}(t) \right\rangle = \int_{-\infty}^{+\infty} f(t) \cdot \phi(t-b) \cdot e^{j\xi t} dt$$

$$\Psi_{b,a}(b,a) = \left\langle f(t), \psi_{b,a}(t) \right\rangle = \int_{-\infty}^{+\infty} f(t) \cdot \frac{1}{\sqrt{a}} \psi\left(\frac{t-b}{a}\right) dt$$

The difference of three transforms lies in the base functions. Base function $e^{j\omega t}$ is a global function in that it is defined on the entire real axis ($t \in (-\infty, +\infty)$). Base functions involved in STFT and Wavelet Transform are called local bases since they exist in a finite interval of the real axis ($t \in (x, y)$). For example, STFT is also called windowed Fourier Transform because its base function is obtained by multiplying $e^{j\xi t}$ with a window function $\phi(t)$ that is finite in real axis. "To represent a global function $f(t)$, $t \in (-\infty, +\infty)$ with a local basis $\psi(t)$, $t \in (x, y)$, functions that exist outside the finite interval must be represented by integer shift of the base function along the real axis [14]." "b" in the equations is such an integer shift. The base functions for Wavelet Transform are not only shifted over the time domain, but also dilated over the frequency domain. $\psi(t)$ is usually called mother wavelet, and it is shifted and dilated to $\frac{1}{\sqrt{a}}\psi\left(\frac{t-b}{a}\right)$.

Fig. 6 shows one example of the base function for STFT and Wavelet analysis. The window function used for STFT is $\chi_{[-\tau,\tau]}(t)$ and the mother Wavelet is Morlet Wavelet [15]. It is observed that the STFT bases have the fixed length in time domain, however the length of the Wavelet bases is decreasing with the frequency increasing.

The advantage of Wavelet analysis over Fourier analysis can also be observed from Fig. 7 where time-frequency window radius "increases in time (reduces in frequency) while resolving the low-frequency contents, and decreases in time (increases in frequency) while resolving the high-frequency contents of a signal [16]."

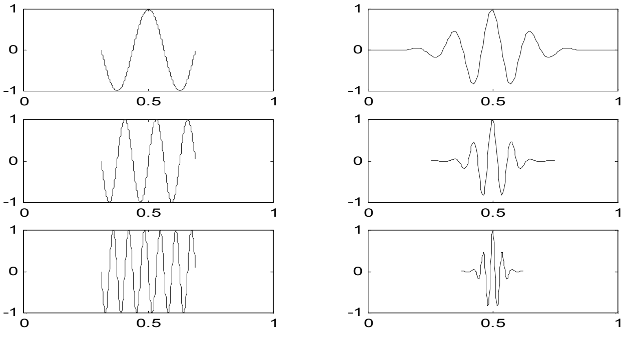


Fig. 6. Example of STFT and Wavelet basis

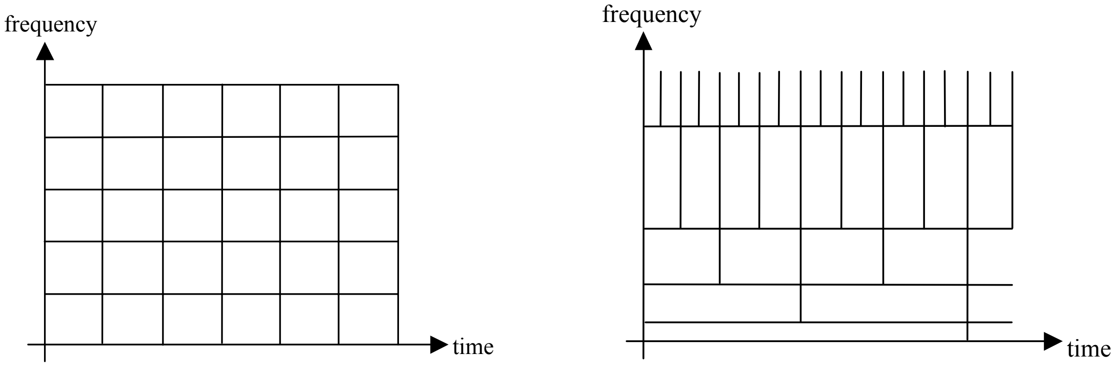


Fig. 7. Time-frequency window for STFT and Wavelets

Filter Banks

Besides mathematical view, the Wavelet theory has also been enriched through other fields like physics and engineering. In the late-eighties Daubechies and Mallat made their contribution to the theory of Wavelets and "established connections to discrete signal processing results [13]". "With the multiresolution work of S. Mallat and the work of I. Daubechies [17]", the Wavelet theory has led to an interesting connection with filter banks.

The filter bank idea provides perfect and fast algorithms to decompose and reconstruct the original signal with Discrete Wavelet Transform. De-composition algorithm can be represented by the tree structure shown in Fig. 8. The original signal comes from the left, and it is fed into two filters, low pass filter (h_0) and high pass filter (h_1). The low pass filter allows the low frequency part of the signal to go through, down-samples it by 2 and obtains the approximation of the signal a_0. For the high pass filter, it does a similar job and gets the wavelet coefficients w_0. More filter banks can be connected in series to achieve lower order approximation of the signal. The reconstruction algorithm is a reverse process of a de-composition algorithm and it uses different filter banks g_0 and g_1.

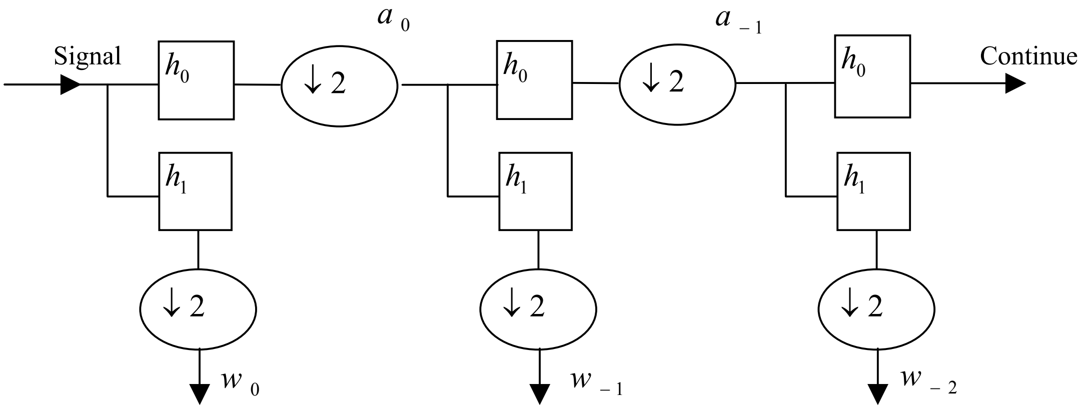


Fig. 8. Filter bank decomposition algorithm

Filter Construction

Before the implementation of the decomposition and reconstruction algorithm, one needs to construct the filter bank through Wavelets. Multi-resolution analysis (MRA) forms the most important building block for this construction. Detail equation derivation based on MRA can be found in reference [14].

In reality, lots of math tools are available for generating the filter coefficients. For example, MATLAB Wavelets toolbox provides a group of functions to generate filter coefficients, such as biorthogonal spline wavelets (function “biorwavf”), coiflet wavelets (function “coiwavf”), and the famous daubechies wavelets (function “dbaux”)[15], etc.

Wavelets Application

This subsection introduces the Wavelets application in one-dimensional digital signal processing. Decomposition and reconstruction algorithm discussed in former section will be used to process the data collected from CB.

De-noising

The original data comes from two CBs of the same type, from the same manufacturer, and located in the same substation. Fig. 9 picks only one channel of the signals, the current in the trip coil. Before any further analysis, we assume that data gathered from these two CBs should have similar waveform envelopes since they are of the same type and work in a similar environment. We differentiate two CBs through their device identifications.

The first trip coil current (Fig. 9) comes from CB “01A0” (ID). Trip coil encloses a plunger that is connected to the opening mechanism. Once the open command is sent to the control circuit, the trip coil current increases gradually until enough energy is stored to push the plunger out of the coil. The coil current appears to saturate when the trip lever is forced to release the trip latch connected with the plunger. The dip next to the saturation indicates the starting of mechanism movement that will eventually open the main contacts of the CB. The coil current then continues to increase until an auxiliary contact “a” opens the current path to the trip coil. Auxiliary “a” contact is designed to open when the main contacts begin to open. Trip coil quickly de-energizes and the current returns to zero.

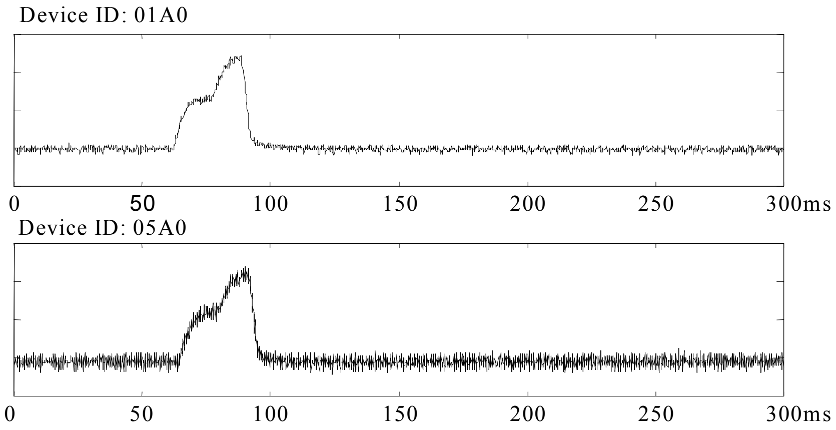


Fig. 9. Coil current collected from in-service CB

Usually the instrumentation noise appearing on the signal is acceptable. It won’t affect the measurements of the initial point, the dip point and the de-energizing point of the coil current. However, the noise shown in the second of data (Fig. 9) collected from CB “05A0” is heavy enough to mess up the measurements. Due to the unavoidability and unpredictability of the instrumentation noise, proper de-noising technique must be used to clean the data and help to get precise measurements.

Fig. 10 shows the second trip coil current that is burdened with noise, and the two-step signal decomposition by using wavelet ‘db1’ (Daubechies wavelet at order one). Fig. 10 uses the same notation as Fig. 8. If the input signal in Fig. 8 is the signal shown in the upper-left box in Fig. 10, then the wavelet coefficients and signal approximations can also find their parallel in two figures. The approximations retain all the important information like the dip point. The details take most of the instrumentation noise from the original signal. Therefore, one can use the signal approximations to measure the events.

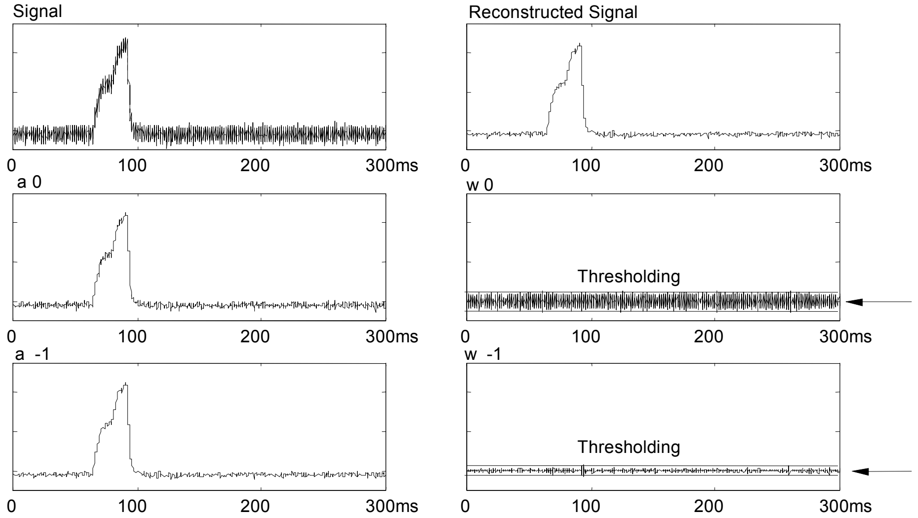


Fig. 10. Signal de-noising example

Decomposition seems already powerful in this example, but there is no guarantee that approximation can keep all the important information, and sometimes, signal detail may have the information that one don’t want to drop. This is highly depending upon the wavelets, their order and the decompose level one chooses. Another de-noising way is to refine the signal detail, and reconstruct the signal by using both the approximation and refined detail. Further analysis or calculation will be performed on the reconstructed signals. Thresholding can be used to refine the signal. The most commonly used in one-dimensional signal processing are hard and soft thresholding. Both methods choose a gate value, and any value smaller than the gate value will be set to zero. For hard thresholding the value above the gate will remain untouched, while for soft thresholding the value above the gate will be subtracted by the gate value. In Fig. 10, a reconstructed signal using the soft thresholding is set beside the original signal. If you compare it with the 2nd level approximation, slight difference can be observed because the second approximation does not contain any detail information.

Splitting Signal Components

In the next example, yard DC voltage that is supplied to the motor and heater in the control circuit is selected to illustrate the Wavelets application. Yard DC usually remains at a constant voltage. Any extreme voltage drop will be considered as a potential problem. Special events take place during the circuit breaker operation may affect the voltage, however, if the voltage still remains within the threshold, the change in the voltage is acceptable.

Three special events that may affect the DC voltage are examined here. The first event happens when either the trip or the close coil is energized. The coil current goes up during the energizing, which drags the DC voltage down. The second event happens near the end of a circuit breaker closing operation. Some types of breakers use yard DC for close energy, and voltage will be dragged down slowly near the end of the signal where the charging motor prepares for a free opening action. The third event covers the whole time span of the circuit breaker operation. For certain types of breakers, small sinusoidal waveforms will be added to the DC voltage during its operation.

In order to measure the voltage changes incurred by these three different events, we use “db1” wavelets to decompose the signal into 6 levels. The voltage drop due to the coil energizing and motor charging can be measured at the 6th level signal approximation and the amplitude of sinusoidal waveform can be measured at the 3rd level signal detail. With the help of decomposition algorithm, we are able to separate different events (shown in Fig. 11) and measure each of them respectively.

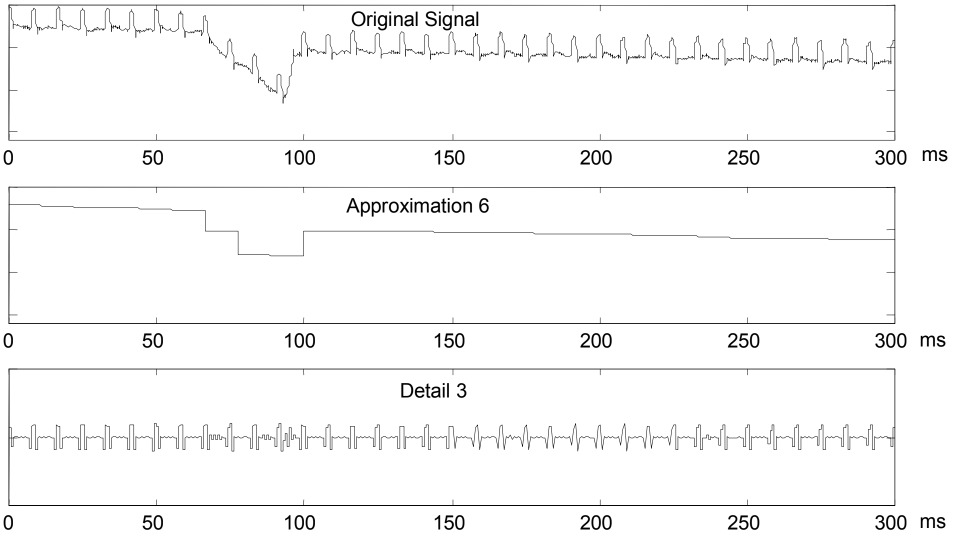


Fig. 11. Signal splitting example

Conclusion

This chapter first provides a mathematical view of the continuous wavelet transform. CWT is compared with FT and STFT, and the advantage of CWT can be easily observed through the comparison. DWT is widely used in the engineering area since it reduces the computation load and eliminates redundant information that exists in CWT. The two-channel filter bank algorithm (decomposition and reconstruction) introduced in the third section is an equivalent to DWT. In the end, we demonstrate how to apply Wavelets algorithm in the digital signal processing by using the data collected from in-service CB.

CHAPTER IV

SIGNAL PARAMETERS

Introduction

After collecting the data from circuit breaker (CB), an expert usually takes a look at the newly recorded performance indicators, compares them with old records, and draws a conclusion based on the overall information. An automated analysis system works differently. In order to use information from the original data, it must be able to describe the information quantitatively. A feature is a piece of information that needs to be and can be quantified. A signal parameter is used to quantify the feature. The role of the pre-processing system described in this thesis is to extract the pertinent information (features) in the form of signal parameters. This process is also called feature extraction, which is "a mapping of a data vector X into a lower dimensional feature vector Y (where length(Y) << length(X)) " [12].

This chapter describes the features and defines signal parameters for the performance indicators. They are selected based on the manufacturer's manuals, relevant books, as well as the discussion with experienced maintenance crews who have been using performance indicators to predict CB operating condition for years.

Features and Signal Parameters

The important features for the CB condition analysis fall into two categories: Events features and features that describing the smoothness or special shape of the individual signal.

An event refers to a state transition or an unusual change in the waveform profile. The time when the event happens and sequence of events are of interest for analyzing the CB condition. A maximum of ten events have been identified in Table II. Not all of these events will take place in a CB operation. For example, events related to "X" and "Y" coil will only appear in a closing operation for certain types of CB. The first seven events are expected to show up in every data record as indicated in Fig. 12. Three phase currents usually make at the same time and break at slightly different time. To fit the phase currents events into the whole sequence, only one signal parameter is provided in Table II. In real analysis, three signal parameters $T7_A$, $T7_B$ and $T7_C$ are used to describe the change in three phase currents separately.

TABLE II
EVENTS DEFINITION FOR CB PERFORMANCE INDICTORS

Event #	Event Description	Signal Parameters
1	Trip or close operation is initiated. (Trip or close initiate signal changes from "LOW" to "HIGH")	T1
2	Coil current picks up	T2
3	Coil current dips after saturation	T3
4	Coil current drops off	T4
5	B contact breaks or makes (a change of status from "LOW" to "HIGH" or vise versa)	T5
6	A contact breaks or makes	T6
7	Phase currents breaks or makes	T7
8	X coil current picks up	T8
9	X coil current drops out	T9
10	Y coil current picks up	T10

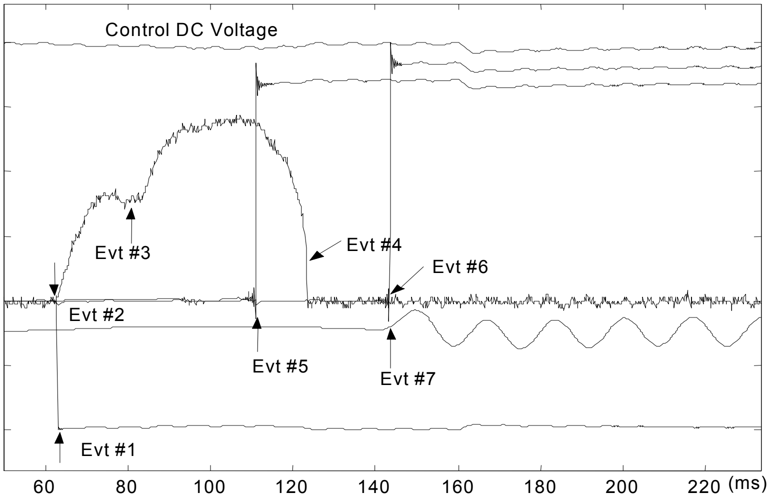


Fig. 12. Event features for a CB closing operation

Signals are not always as "clean" as shown in Fig. 12. Coil currents may be loaded with heavy noise that makes it hard to find where the dip is located. The energized coil current may appear choppy. There might be a spike, ripple or distortion observed on the DC voltage. Even a contact signal may be distorted with noise or bounce. These features are useful in telling the condition of certain parts of CB. Signal parameters such as SPI (spike), NOI (noise), RIP (ripple), etc, are defined in Table III to describe the features for different signals.

TABLE III
FEATURES FOR INDIVIDUAL SIGNALS

Signal Name	Feature Description	Signal Parameters
Control DC or Yard DC Voltage	Voltage drop (in percentage) from the normal value when the coil current is activated.	DIP1
Control DC or Yard DC Voltage	Voltage drop (in percentage) from the normal value when the coil current is deactivated	DIP2
Control DC or Yard DC Voltage	Voltage rise or drop (in percentage) from the normal value of a spike	SPI
Control DC or Yard DC Voltage	Voltage deviation (in percentage) from the normal value of ripples added in the whole signal channel	RIP
Control DC or Yard DC Voltage	Voltage deviation (in percentage) from the normal value of the distortion added in the partial channel	DIS
A Contact or B Contact	Excessive drop (in percentage) from the normal value when the contact is in the high status	DIP
A Contact or B Contact	Excessive noise appears when the contact changes its status	NOI
A Contact or B Contact	Bounce appears when the contact changes its status	BOU
Coil Current	Choppy waveforms after coil current saturates	NOI
X Coil, Y Coil or Trip, Close Initiate	Excessive drop (in percentage) from the normal value when the signal is in the high status	DRP
Phase Currents	Phase current re-strikes	RES

The "Initiate" Signal

The "initiate" signal usually makes a transition from "OFF" to "ON" during CB operation, and is expected to remain "ON" status until the end of the record. Since this transition is the first event that happens in the record, it is used to indicate the start of a sequence of events.

Signal parameters for the "initiate" signals include: T1 and DRP.

Signal parameter T1 is used to represent the time instant when the "initiate" signal makes its transition. Signal parameter DRP is used to indicate whether or not the "initiate" signal drops out during the operation.

Control and Yard DC Voltage

Signal parameters for the control and yard DC voltages include: DIP1, DIP2, RIP, SPI and DIS.

DC voltages usually remain relatively constant. However, other parts of the circuit may affect the DC voltage. In both opening and closing operation, when the trip or close coil is energized, it draws the control DC voltage down a little bit. This slight drop in voltage correlated with coil current is shown in Fig. 13.

Some CBs use yard DC voltage for closing energy, and slight variations may be seen near the end of the test record where the charging motor pulls the voltage down (Fig. 13).

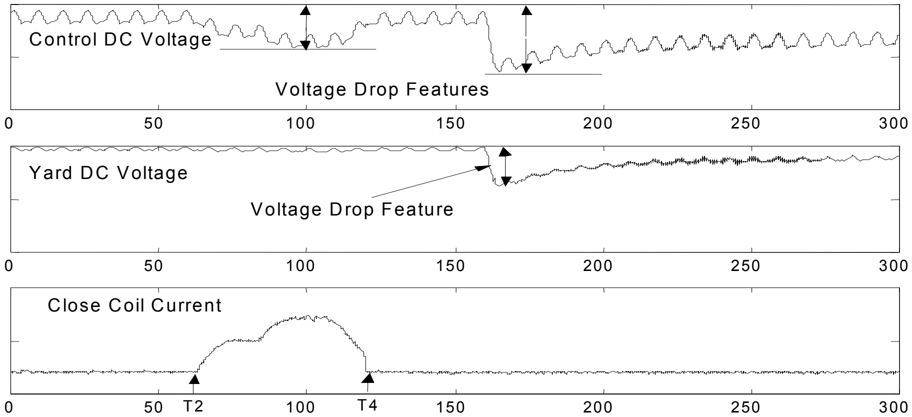


Fig. 13. Voltage drops of DC voltage

Assume that $D_1 = 1 - \min(V(t)) / avg(V) < S_1, T_2 < t < T_4$

$$D_2 = 1 - \min(V(t)) / avg(V) < S_2, t > T_4$$

In the equations above, D_1 is the maximum voltage drop originated from coil energizing during $T_2 < t < T_4$. D_2 is the maximum voltage drop caused by motor charging when $t > T_4$, i.e., when the breaker finished its mechanical movement. These two features needs to be differentiated. The settings for these two features are also different. S_1 is usually smaller than S_2. Any enormous voltage drop exceeding the settings will be considered as an alarm for the DC battery condition. Signal parameter DIP1 equals D_1, and DIP2 equals D_2.

Besides the voltage amplitude features, three more features need to be examined and observed in both DC voltages. These features are ripple, spike and distortion.

Ripple refers to small sinusoidal waveforms added on the DC voltage. Some types of CBs operate on the magnetic force produced by the load current [2]. The sinusoidal current has also been reflected in the DC voltage, which is usually hard to observe. By zooming in the vertical axis, the amplitude of DC voltage is magnified, and ripples appear on the voltage. Fig. 14 shows both a magnified and normal view for DC voltage. The ripple is measured by calculating its amplitude shown in Fig. 14. If the ripple amplitude exceeds certain expected value, signal parameter RIP will be set to TRUE, and then the analysis system will signify the problem —a failed SCR in the station battery charger [18].

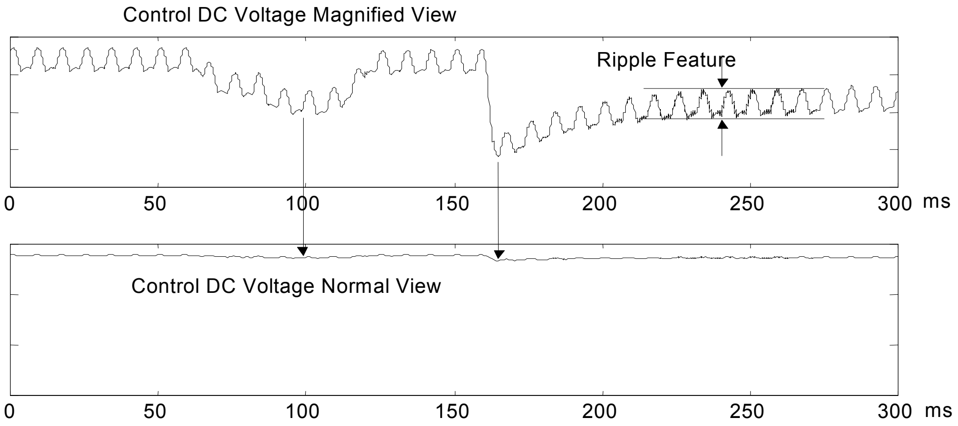


Fig. 14. Ripple feature of DC voltage

While ripples exist over the whole span of the DC voltage, the distortion feature only appears on a part of DC voltage. It looks like a saw blade in Fig. 15. If the distortion amplitude exceeds certain expected value, signal parameter DIS will be set to TRUE, and the analysis system will signify the problem: a failed SCR in the station battery charger [18].

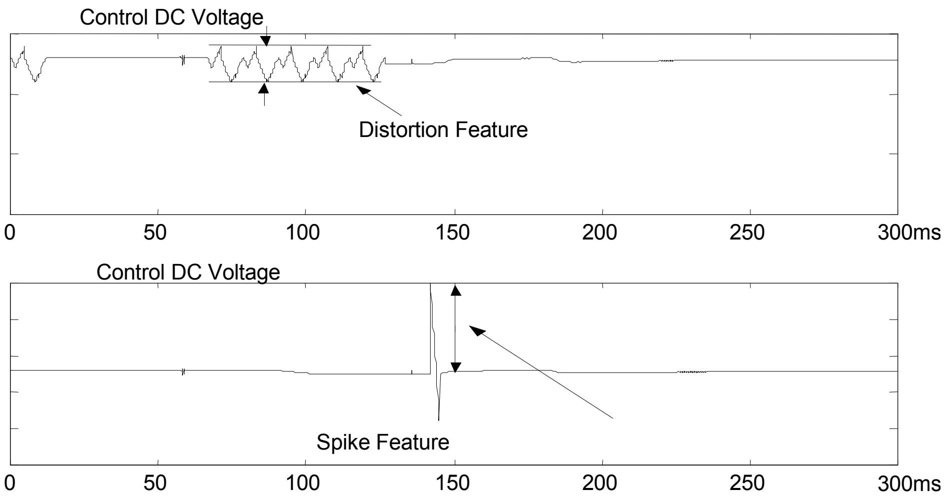


Fig. 15. Distortion & spike feature of DC voltage

While the ripple describes a global feature of the DC voltage, spike can be classified as a local feature, which can be modeled as a Kronecker delta function δ. $\delta_{k,l} = \begin{cases} A_s, k = l \\ 0, \ k \neq l \end{cases}, k, l \in t$. Assume that the amplitude of DC voltage is "A", and spike can be calculated as $K = \frac{A_s}{A}\%$. Signal parameter SPI equals K.

Coil Currents

Signal parameters for the coil currents include: T2, T3, T4, and CHP.

The trip coil current usually picks up (Event #2 at T2 in Fig. 17) immediately after the "trip initiate" signal is activated (changes status from "OFF" to "ON"). The close coil current may start with a certain delay. The next observation, after the coil current picks up, is the point in which the armature free travel is completed. The coil current attempts to saturate as the coil tries to turn the trip shaft and release the latch. A small dip after the saturation indicates where the armature is moving again and at this point (Event #3 at T3) the mechanism is in motion (as shown in Fig. 16). The time period between the moment when current picks up (T2) and the moment when dip occurs (T3) in the coil current curve is the free travel time (equals T3 – T2) or trip latch time. The coil current event also needs to correlate with the event of “A” or “B” contact. The time period between the dip and the operation of either “A” or “B” contact is the mechanism travel time (equals T6 – T3 & T5 – T3 in Fig. 17). Any friction added in the coil, rollers or mechanism bearings reflect in lengthening of the free travel time or mechanism travel time. Coil currents drops off at T4, which happens anywhere between T5 and T6 for most CB types.

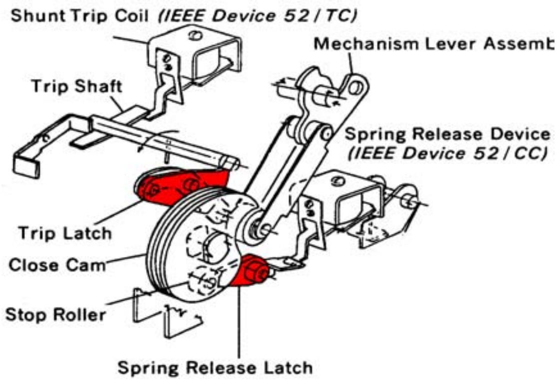


Fig. 16. The cooperation of coil and breaker mechanism

Using equations to summarize the above description:

1 $|T2 - T1| < \varepsilon$ (opening operation, ε is a small value)

2 $|T2 - T1 - E| < \varepsilon$ (closing operation, E is the expected time)

3 Free Travel Time $|T3 - T2 - E| < \varepsilon$ (E is the expected travel time)

4 Mechanism Travel Time = $|T5 - T3 - E| < \varepsilon$ (closing operation, E is the expected mechanism travel time)

5 Mechanism Travel Time = $|T6 - T3 - E| < \varepsilon$ (opening operation)

6 T6 < T4 < T5 (opening operation for most CB types)

7 T5 < T4 < T6 (closing operation for most CB types)

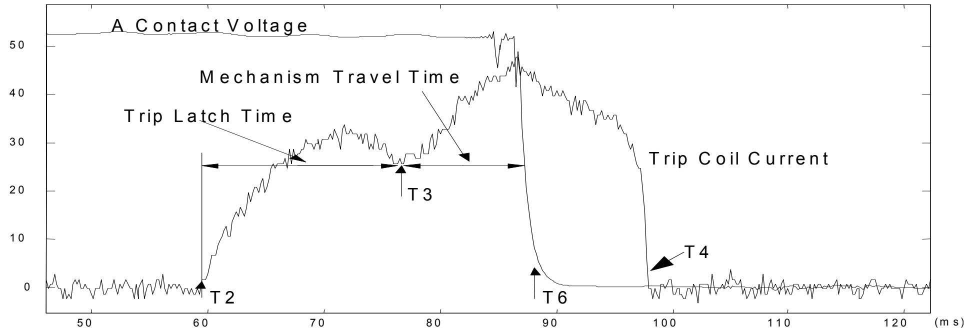


Fig. 17. Events features of coil current

Besides the event features, choppy feature are described as well. The saturation line of the coil current is expected to be smooth, but in Fig. 18 the line is choppy. This feature discloses the possibility of a bad coil. Signal parameter CHP is set to be TRUE when a choppy feature is detected.

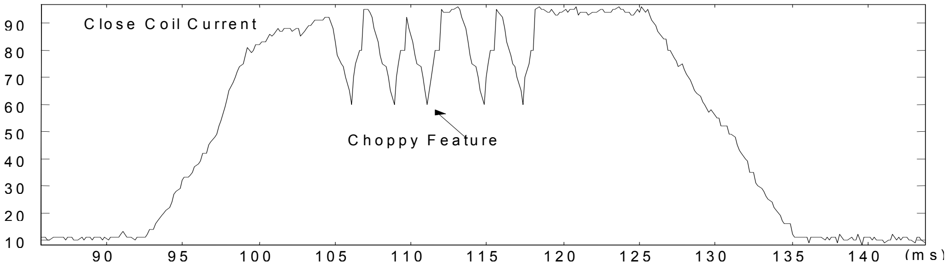


Fig. 18. Choppy feature of coil current

A and B Contact

Signal parameters for the “A” and “B” contact include: T5, T6, DIP, NOI and BOU.

When CB operates, auxiliary contacts “A” and “B” change their open/close status according to the status change of main contacts. The status of “A” and “B” contact is measured by voltage, and the voltage changes either from "LOW" to "HIGH" or from "HIGH" to "LOW". The auxiliary contacts are designed such that there is always a time difference between the changing of "A" contact and that of "B" contact. The reciprocal of the time difference is proportional to the velocity of CB operation, which is used to measure the speed of operation.

T6 is used to designate event #6 of "A" contact and T5 is used to designate event #5 of "B" contact. Therefore the CB velocity can be defined as "1/|T5 – T6|". Usually time difference |T5 – T6| can be used to tell the CB velocity. If it is longer than the expected value E, the velocity is said to be decreased, and vise versa. The requirement can be expressed as $\left||T5 - T6| - E\right| < \varepsilon$, where E stands for the expected value.

Other features of "A" and "B" contact other than the time feature are the dip, noise, and bounce. Signal parameter DIP (Fig. 19) for contact is defined in a similar way as parameters DIP1 and DIP2 for DC voltage. DIP equals to D_1 in a closing operation, and equals to D_2 in an opening operation.

$$D_1 = 1 - \min(V(t)) / avg(V(t)), \quad 0 < t < T_5 (or \quad T_6)$$

$$D_2 = 1 - \min(V(t)) / avg(V(t)), \quad t > T_5 (or \quad T_6)$$

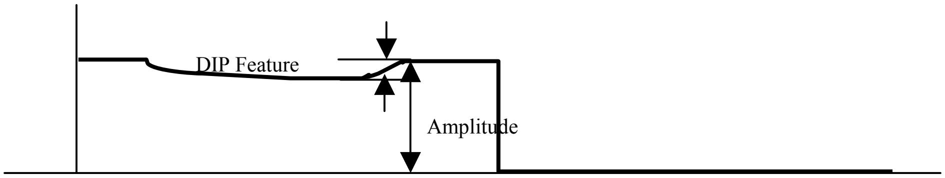


Fig. 19. DIP feature of contact

If there is lots of noise when contact changes its state, the contact may be dirty and needs cleaning. The bounce of a contact in Fig. 20 shows a defective contact buffer. Signal parameter NOI and BOU will be set to TRUE if either condition happens.

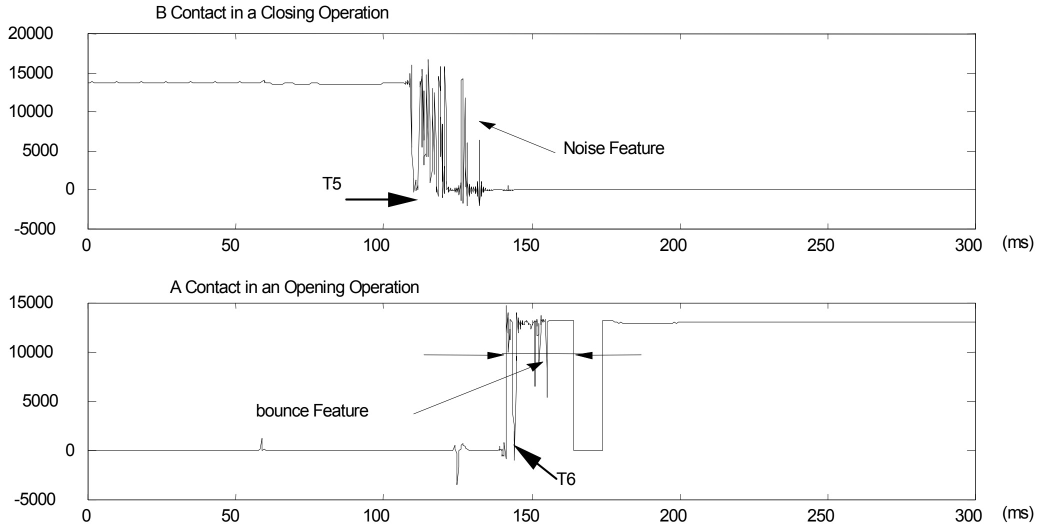


Fig. 20. Noise & bounce feature of contacts

Phase Currents

Signal parameters for the phase currents include: T7 and RES.

To fit into the overall sequence of operation, phase currents could make in a closing operation anywhere after the event #3 coil current dip time (T3 < T7). It is also required that all three phase-currents must break or make within certain period of time, usually within a cycle. Otherwise there will be a phase current time violation, and CB pole alignment needs to be checked (Fig. 21). This feature can be represented as $\max(T7_A, T7_B, T7_C) - \min(T7_A, T7_B, T7_C) < S$.

There should be no re-strikes (shown in Fig. 22) or continued phase current indications in an opening operation. Signal parameter RES is used to indicate the presence of a re-strike.

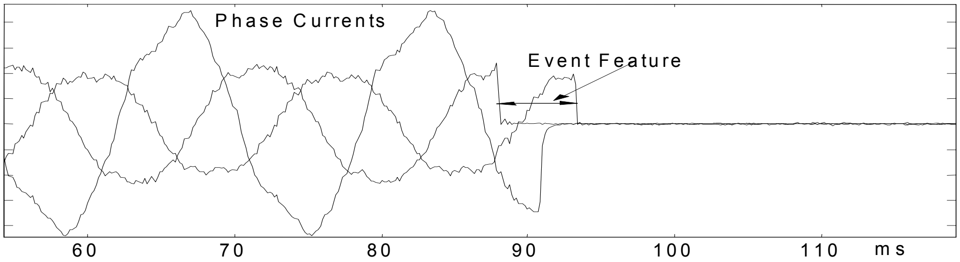


Fig. 21. Event feature of phase currents

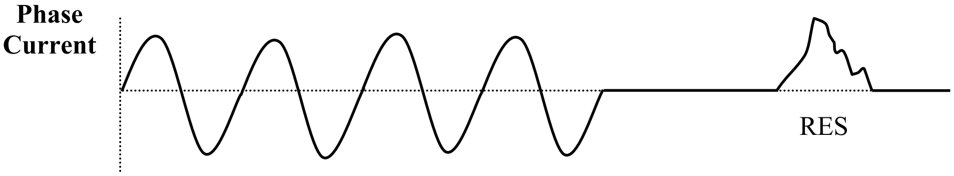


Fig. 22. Restrike feature of phase current

X and Y coil

Signal parameters for the “X” Coil and “Y” Coil signals include: T8, T9, T10 and DRP.

“X” and “Y” Coil are only involved in closing operation. Their currents can also be simplified as having only two statuses: energized and de-energized. “X” coil current is expected to appear at T8 in Fig. 23 immediately after the “close initiate” signal starts at T1 ($|T8 - T1| < \varepsilon$) since the “close initiate” signal is directly applied to x coil in the closing circuit. It should appear well before the close coil current rises at T3. The reason is that “X” coil needs some time to be energized, and until then it can close the contact 52X/a to establish an electric path to energize the close coil. Therefore, it conforms to T2 > T8.

If the breaker is equipped with “Y” coil or anti-pump relay, the distance between “Y” energizing at T10 and “X” de-energizing at T9 is the “Y” coil transition time. This timing is critical to prevent anti-pumping in the closing operation. It is required that T10 < T9.

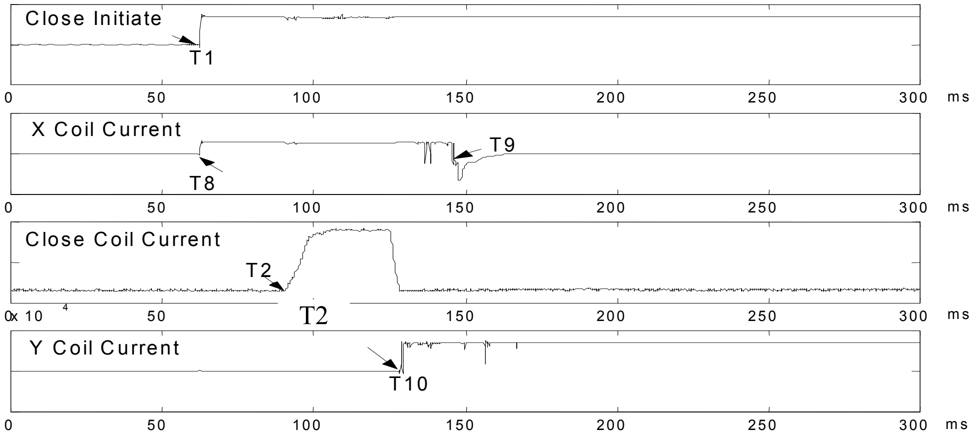


Fig. 23. Event features of "X" & "Y" coils

Conclusion

Performance indicators, their features and signal parameters have been discussed in this section. One can easily discover that performance indicators are the input and the signal parameters are the output data for the pre-processing module respectively, and now we are well prepared to discuss the pre-processing module in the next section.

CHAPTER V

FEATURE EXTRACTION

Introduction

Functions are used to contain the algorithm for feature extraction. It is preferable that the algorithm within the function implements only one simple task. Communication among the functions can be achieved through understandable function interface such as a self-explaining function name and well-documented input and output parameters.

Following the same design consideration, two classes of functions, extraction functions and utility functions, are identified to extract the feature-describing signal parameters. Utility functions do not calculate a signal parameter directly; rather, they provide a service to other functions. As shown in Fig. 24, utility function 3 provides service to utility function 1 and utility function 1 provides service to an extraction function 1. In this way, a utility function can be called upon repeatedly. Both the inputs and the outputs of an extraction function are listed in Table VI. Each extraction function may use the service provided by utility functions. In other words, the extraction function may be built by using the utility functions.

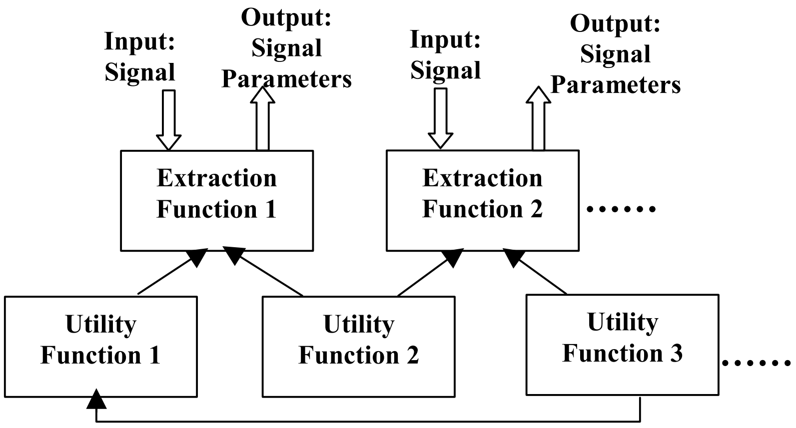


Fig. 24. Utility and extraction functions

Table IV and Table V provide a list of utility functions and a list of extraction function respectively. The name for the extraction function is generated through combining the word “Extract” and name of the extracted signal parameters, for example, “ExtractSPI” is the combination of “Extract” and signal parameter “SPI” describing the spike on DC Voltage. In one special case, both DC voltage and contact have a signal parameter named “DIP” and they are used to describe similar features. Signal parameters are renamed to “ExtractDIP1” and “ExtractDIP2” to differentiate them in the table. Actually this renaming is unnecessary when using special techniques in the higher-level software implementation.

TABLE IV
UTILITY FUNCTIONS

No.	Function Name	Function Description
1	trimavg	Calculate the signal average that is insensitive to unexpected large disturbance and noise.
2	max	Calculate the maximum value of a signal
3	min	Calculate the minimum value of a signal
4	falltime	Calculate the time when the signal changes its status from “HIGH” to “LOW”.
5	risetime	Calculate the time when the signal changes its status from “LOW” to “HIGH”.
6	derivative	Calculate the derivative of the signal
7	pburg	Calculate the power spectral density of the signal
8	decompose	Wavelet decompose method
9	denoise	Wavelet denoise method
10	getcoeff	Get the coefficients of specified level from decomposed signal
11	Threshold	Calculate the amplitude threshold or the gate value

Before the discussion of individual function, it is necessary to discuss a general pre-processing procedure: three-level feature extraction including de-noising, splitting and signal parameter calculation. De-noising is used to suppress the excessive instrumentation noise and reveal the features that may be distorted by the noise. Soft thresholding is used in de-noising to preserve the desired signal features. The Splitting process is expected to separate the features to facilitate the calculation of the signal parameters for individual features. Signal parameters are calculated after de-noising and splitting process. In reality, some signal parameters can be easily calculated without undergoing the denoising and splitting steps.

TABLE V
EXTRACTION FUNCTIONS

No.	Function Name	Function Description
1	ExtractDIP1	Extract Control and Yard DC Voltage Dip Feature
2	ExtractSPI	Extract Control and Yard DC Voltage Spike Feature
3	ExtractDIS	Extract Control and Yard DC Voltage Distortion Feature
4	ExtractRIP*	Extract Control and Yard DC Voltage Ripple Feature
5	ExtractNOI	Extract A and B Contact Noise Feature
6	ExtractDIP2	Extract A and B Contact Dip Feature
7	ExtractCHP	Extract Trip and Close Coil Current Choppy Feature
8	ExtractSUP	Extract Trip and Close Coil Current Suppression Feature
9	ExtractDRP	Extract X, Y Coil and Trip, Close Initiate Drop Out Feature
10	ExtractRES	Extract Phase Currents Re-strike Feature
11	ExtractTIM*	Extract event features

* Source code in MATLAB has been provided in Appendix I

Utility Functions

Utility functions are introduced first because they will be frequently mentioned in the discussion of the extraction functions.

Function “trimavg” differs from average or mean of signal we usually use in that it uses a certain percent of the signal samples to calculate the average. The signal samples are first sorted from small to large. If 20 percent samples are to be used, 10% smallest samples and 10% largest samples are put aside, leaving the 80% samples in the middle for averaging. Using this function avoids the sensitivity of the average function to an unexpected noise or distortion. For example in Fig. 25, a trip coil current of type FLUARC from manufacturer Square D has suppression when the current drops off. The average for the overall signal sample is 12.6175 while using the 80 percent samples the average is 6.6162. Obviously, suppression contributes to the increase of the overall average that nearly doubles the trimmed average.

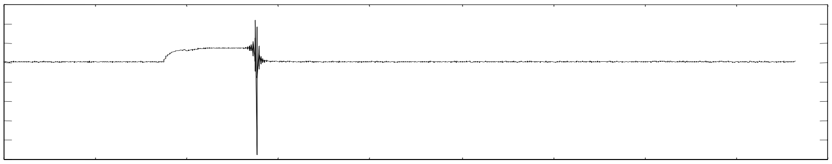


Fig. 25. Trip coil current example

Trimmed average is used to construct an amplitude threshold when extracting the event features. A signal status change is detected when the signal sample goes above or below the threshold from its original status, and event will be located at the point when the signal sample crosses the threshold.

Due to the sampling taking place by DFR, continuous circuit signals are converted into discrete signals, and time is converted to a sampling point or index (i) through the equation $i = t/f$, where t is the time point and f is the sampling frequency. When collecting the performance indicators, the sampling frequency of DFR is configured to 5760Hz so that the first sample corresponds to 0.01736ms in time.

Functions “rise time” and “fall time” are used to locate the time points when the signal picks up and drops out. The inputs are the start index, end index and threshold of the signal. The default index starts at 0 and ends at the last signal sample. Threshold is obtained by scaling up the trimmed average of the signal with a setting. For function “rise-time”, the signal is checked sample by sample until its value is larger than the threshold. Since there is always some noise when the signal changes its status, a setting “Stable Count” is introduced to make sure that the signal reaches its steady state after the detected sample. If “Stable Count” equals 10, the subsequent 10 samples will be checked. If all of them are larger than the threshold, then the signal is considered reaching its steady state. Otherwise, the detected sample is dropped and the checking continues until another steady state is detected or the last sample of the signal is reached. Fig. 26 summarizes the algorithm for locating the time point when signal changes from status “LOW” to “HIGH”. Souce code for this function is provided in Appendix I.

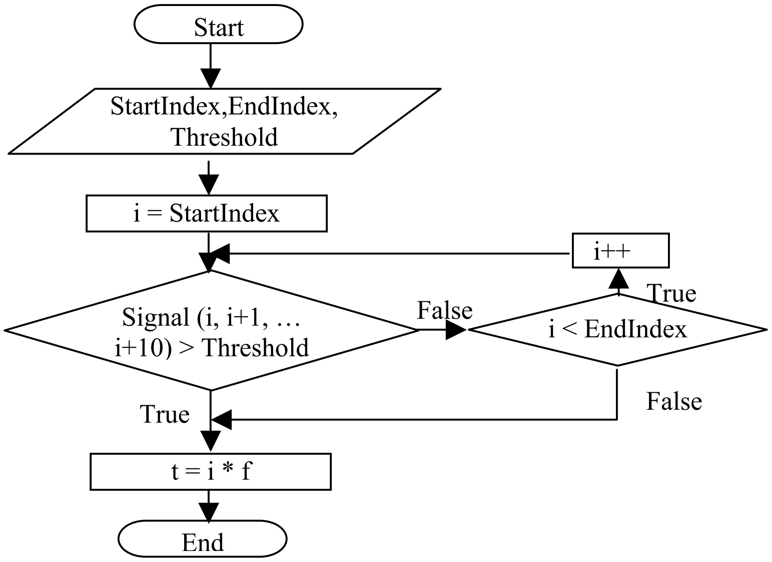


Fig. 26. Flowchart for function “rise time”

Function “fall-time” is designed in the same way except that the signal is checked whether it is smaller than the threshold.

Function “decompose” implements Wavelets decomposition algorithm, which has already been given in chapter III. The inputs of the “decompose” function include the original signal, decompose level and filter order. Daubechies scaling and wavelets decompose filters are used in the algorithm. The maximum decompose level is ten. Two outputs are variable-length vectors. The first output “C” stores all the detail coefficients up to the decompose level specified in the input and the approximation coefficients of the coarsest level. The second output “L” stores the length of detail and approximation coefficients in order to facilitate the retrieval of a given detail or approximation.

Function “getcoeff” returns the desired-level wavelet approximation or detail coefficients from the outputs “C” and “L” of the previous function “decompose”.

Function “denoise” decomposes signal, performs soft thresholding to the detail components and reconstructs the signal using the approximation and the refined details.

Function “Pburg” filters the signal using Burg’s method, and using the FFT of the filtered signal to calculate the one-sided PSD (power spectrum density) with full power .

Functions “max”, “min” and “derivative” output the maximum, the minimum and the derivative of the input signal.

To close the discussion of utility functions, the data type of the inputs and outputs are summarized in Table VI.

TABLE VI
INPUTS AND OUTPUTS FOR UTILITY FUNCTIONS

No.	Function Name	Input Data Type	Output Data Type
1	trimavg	vector <double> Signal	double
2	max	vector<double> Signal	double
3	min	vector<double> Signal	double
4	falltime	vector<double> Signal, double (StartIndex, EndIndex, Threshold)	float
5	risetime*	vector<double> Signal, double (StartIndex, EndIndex, Threshold)	float
6	derivative	vector <double> Signal	double
7	pburg	vector <double> Signal	float
8	decompose	vector <double> Signal	vector <double> C vector <double> L integer (wavelet level) integer (wavelet order)
9	denoise	vector <double> Signal	vector <double>
10	getcoeff	vector <double> C vector <double> L	vector <double> integer (coefficients level)
11	threshold	vector <double> Signal	double

* Source code in MATLAB has been provided in Appendix I

Event Features

The event features for the signals listed in the second column of the Table VII will be discussed in this section. All these signals have two statuses: “ON” or “OFF”, “LOW” or “HIGH”, energized or de-energized and activated or de-activated. The event feature refers to the time point when the signal makes transition from one status to another, which is usually represented by a significant change in the waveforms. This characteristic allows a uniform algorithm design for these signals.

Fig. 27 starts with a signal as the input and ends with one or two time signal parameters as the outputs. During the process, the algorithms call a utility function “Threshold” to prepare one of the inputs for functions “rise time” or “fall time”. They call utility functions “rise time” or “fall time” to locate the events of the signal. Signal in the parentheses share the same algorithm with the signal before it.

TABLE VII
ALGORITHMS LIST FOR EVENT FEATURE EXTRACTION

Algorithm #	Input Signals	Events Of the Signal	Extracted Signal Parameters
1	Close (Trip) Initiate	#1	T1
2	X (Y) Coil	#8, #9 (#10, #11)	T8, T9 (T10, T11)
3	B (A) Contact, (Phase Current)	#5 (#6), (#7)	T5 (T6), (T7)
4	Trip (Close) Coil Current	#2, #4	T2 (T4)

Algorithm #1 is designed for detecting the trigger point of “close (trip) initiate” signal. “Close (trip) initiate” changes its status from “LOW” to “HIGH” upon triggering; therefore the function “rise time” is called to locate the trigger point.

Algorithm #2 is designed for detecting the activation and deactivation point of the “X” (“Y”) coil signal. The “X” (“Y”) coil changes its status from “LOW” to “HIGH” during activation and from “HIGH” to “LOW” during deactivation; therefore two functions “rise time” and “fall time” are called to locate the activation point and deactivation point respectively.

Algorithm #3 is designed for detecting the status change of the “B” (“A”) contact. “B” (“A”) contact changes status from “LOW” to “HIGH” during a CB closing operation and from “HIGH” to “LOW” during an opening operation. Based on the CB operation, a decision to choose correct utility functions must be made. For closing operation, function “rise time” is called while function “fall time” is called for the opening operation. Algorithm #3 is also used to detect when the phase currents drop out or pick up.

Algorithm #4 is designed for detecting the energizing and de-energizing point of the trip (close) coil current. Because a coil has indirect connection to the circuit breaker operation mechanism, the coil current is easily affected by the instrumentation noise. For this reason utility function “denoise” is called to alleviate the noise effect. Upon energizing trip (close) coil current changes gradually from “LOW” to “HIGH”, and upon de-energizing trip (close) coil current changes gradually from “HIGH” to “LOW”. Functions “rise time” and “fall time” are called consecutively to locate the beginning of the energizing and the end of de-energizing.

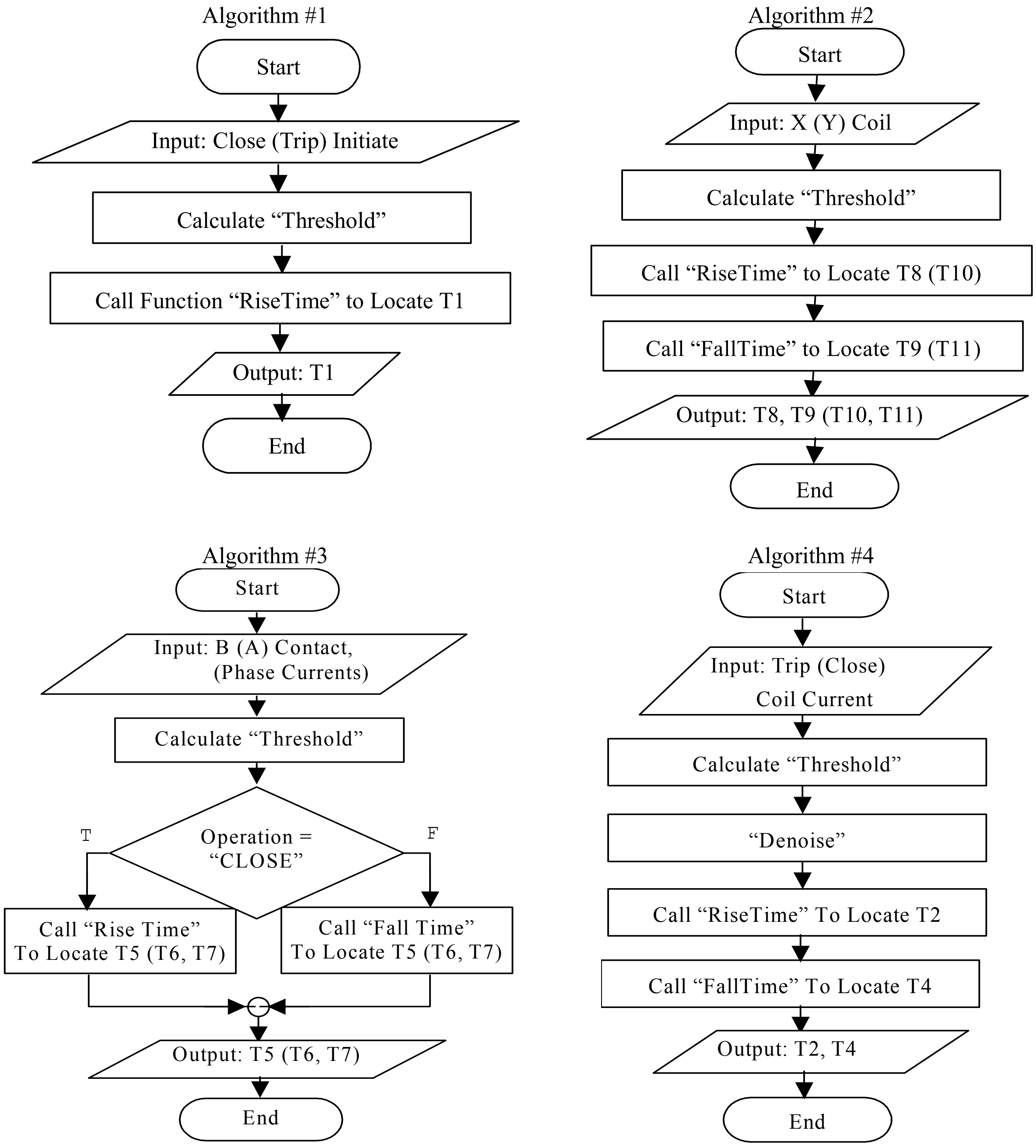


Fig. 27. Flowchart for function “ExtractTIM”

Event #3 (DIP Time)

Event #3 refers to the dip event of the trip (close) coil current. Fig. 28 provides four close coil currents of different breaker types. DIP1 represents a common dip pattern, DIP2 represents a significant dip pattern, DIP3 represents an insignificant dip pattern and DIP4 represents a no dip or hardly-observable dip pattern. A common dip is what has been described before; it is formed when the coil current drops slightly after the saturation. A significant dip shows a drastic drop in the current, which can be easily confused with the situation when the coil current drops out. An insignificant dip shows a minor drop in the current, which can be easily covered by the noise. For some type of breaker, dip can hardly be differentiated from the noise, which makes the event extraction process extremely hard.

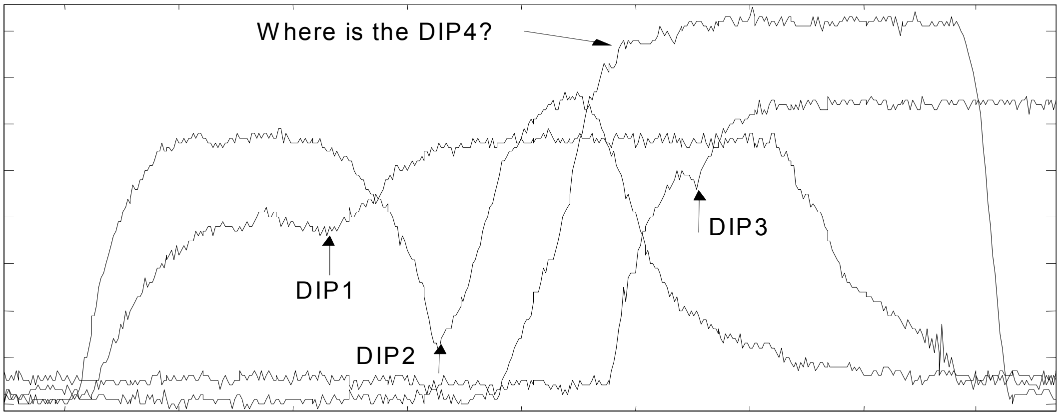


Fig. 28. Dip features for different coil current

Fig. 29 shows an original trip coil current of a GE breaker, its 6th level approximation (using 'db1') and its 4th level detail signal. The 6th level approximation has such a low resolution that the waveform takes the shape in steps. The 4th level detail preserves all the event information such as the beginning time, the dip time and the de-energizing time of the coil current. The goal here is to pick the dip time from the 4th level detail. When making a correlation between the approximation and detail signal, it is not hard to find that the dip time falls into the most significant stair in the approximation signal. The most significant stair covers a limited time interval, which can be used to find the dip time in the detail since the dip time is the most significant point within that time interval.

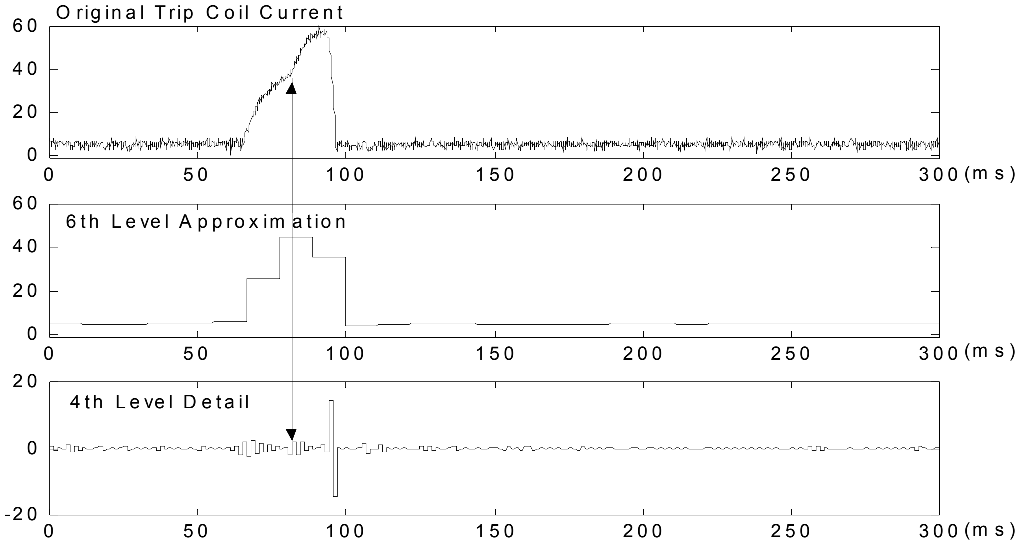


Fig. 29. Use Wavelets decomposition algorithm to locate the DIP point

The sampling frequency of the original signal is 5760 Hz, and in total 1728 samples are taken within 300 ms. Each time the signal is fed into the decomposition filter, the signal is down-sampled by 2. The 6th approximation is obtained through 6 decomposition filters, thus the signal samples in the approximation has been reduced to 27 ($1728 \div 2^6 = 27$). It is quite easy to find the maximum value (most significant value) and its index in the approximation sequence. "[index-1, index]" defines the most significant stair in the approximation signal. Next it needs to be converted to an interval for the 4th level detail signal. Since the detail has been down-sampled four times, the new interval is $[(index-1)\times 2^{6-4}, index\times 2^{6-4}]$. The following task is to find the maximum value and its index in this interval, which only contains 4 (2^{6-4}) samples. The dip time is obtained by multiply the new index by 0.0028 ($2^4 \div 5760$) seconds.

The algorithm is summarized in the flowchart in Fig. 30. The algorithm first selects a group of settings including decomposition level (app_level), detail level (det_level) and the stair in the approximation that the dip shall correspond to based on the breaker type. Table VIII lists the settings for different breaker types. The default settings are used for new breaker types. Next, the algorithm decomposes the signal and gets the approximation and detail coefficients of the desired level. The coeffcients are sorted and calculated to locate the dip time in the end. Output of the extraction is the signal parameter T3. Souce code for this function is provided in Appendix I

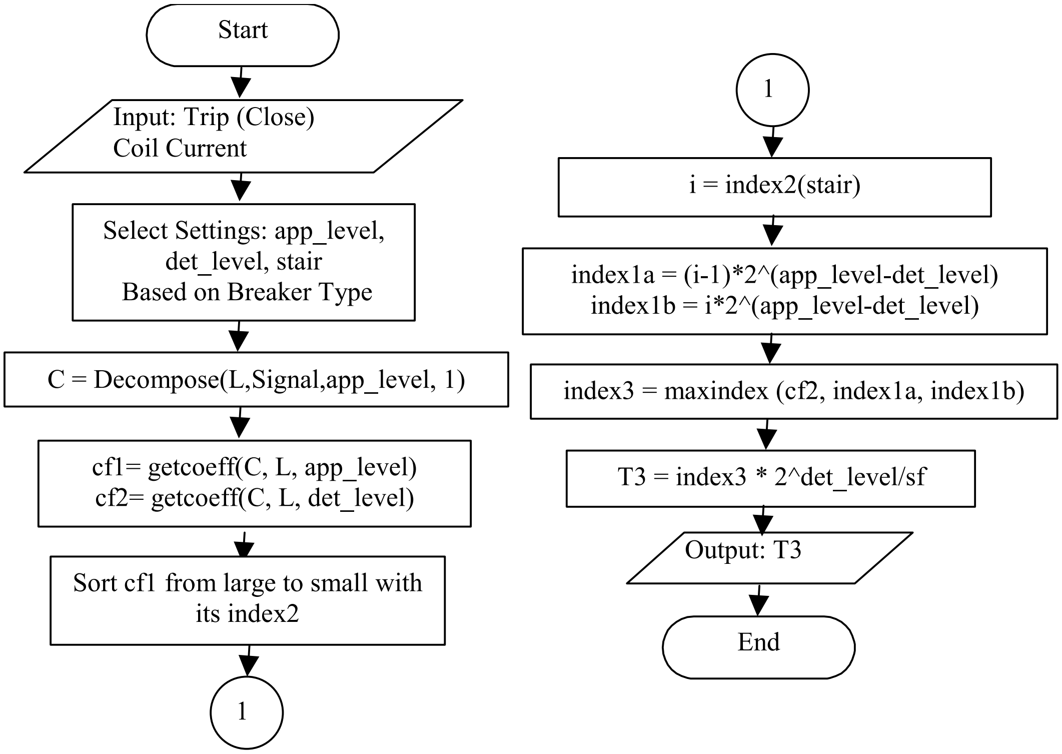


Fig. 30. Flowchart for function "ExtractTIM3"

Returning to the initial four dip patterns, when the algorithm is applied to the signals, the results shown in Fig. 31 are the dip points in milliseconds. For the hardly-detectable dip pattern the location effort is dropped.

TABLE VIII
SETTINGS FOR EVENT #3 FEATURE EXTRACTION ALGORITHM

Manufacturer	Breaker Type	app_level	det_level	stair
ABB	15GHN750	6	3	4
ABB	R4	6	3	2
GE	FKD-38-12000-1	7	4	2
GE	VIB-15-500	6	4	1
GEC	OMX-15	6	4	3
ITE	VBK	6	4	2
S&C	FVR-1121125A	6	3	3
S&C	SD	6	4	3
SA	BCM-15-1	7	2	4
SA	SDO-15-500	7	3	4
SD	VACARC	7	5	2
Default	Default	6	4	1

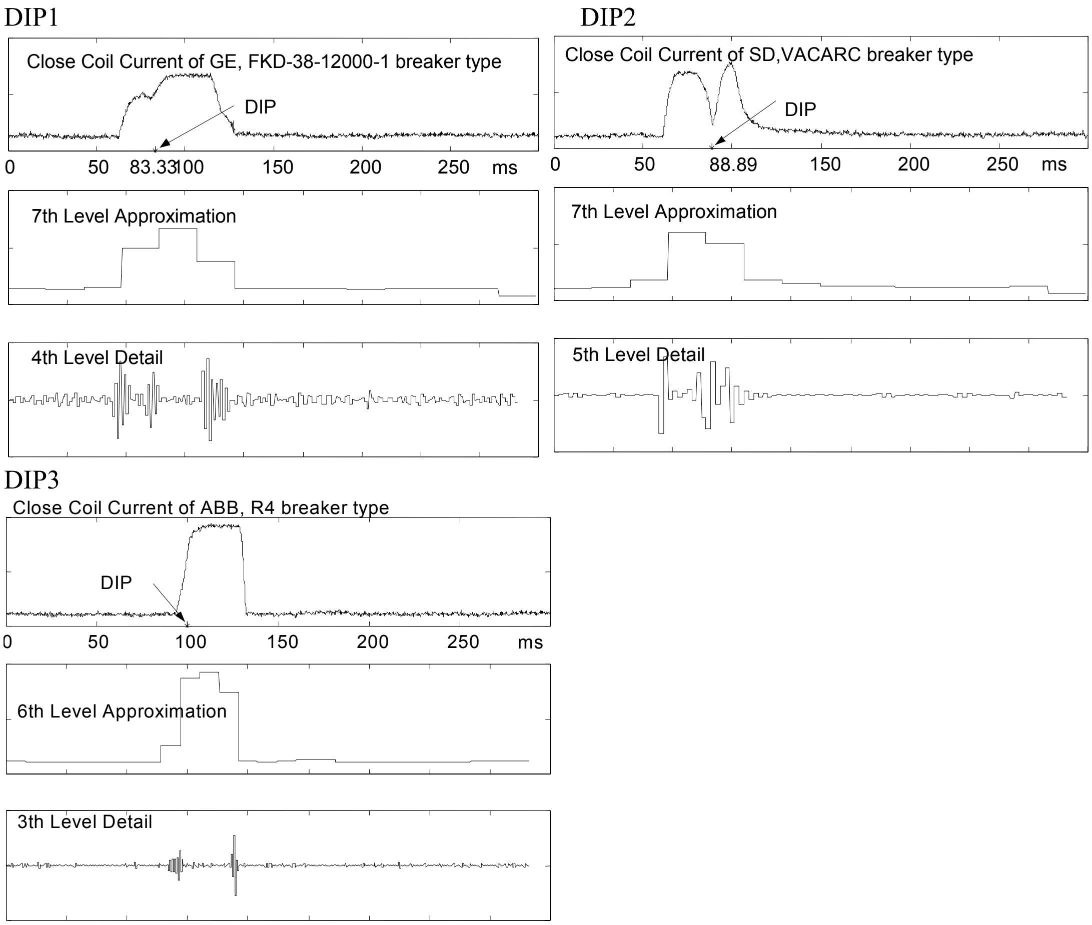


Fig. 31. DIP time extraction

Other Features

A (B) Contact

Function “ExtractDIP” is responsible for calculating the average depth of the maximum dip during “HIGH” status and returning the ratio of the depth and amplitude

Function “ExtractNOI” is responsible for signifying whether there is noise during the transition period or not. Function “derivative” is called to obtain the signal derivative, and the power of the derivative is calcualted to measure the level of noise. If the result exceeds the setting “NOISE_LEVEL”, a boolean flag is set to be true, otherwise the flag is set to be false. The function will return this Boolean flag.

Function “ExtractBOU” is responsible for determining whether there is bounce on the “Contacts” or not. Potential Bounce is detected when more than one Events are detected for the “Contacts”. Next the width of Bounce is measured and compared against a setting “BOUNCE_WIDTH”. If the width is larger than the setting, the output signal parameter BOU will be set to TRUE (BOU default value is FALSE). All the settings are listed in Table IX

TABLE IX
SETTINGS FOR CONTACT FEATURE EXTRACTION

Setting Name	Description	Default Value
NOISE_LEVEL	Power of signal derivatives that measure the noise level.	200
BOUNCE_WIDTH	Width of a Bounce measured by time	5 ms

DC Voltage

DC Voltage differs from the former signals we have already discussed in that it has no apparent status change and shall remain in “HIGH” status all the time. When the trip or close coil is energized, the coil current goes up and draws the DC voltage down a little bit. This corresponding feature of the DC voltage is what an engineer wants to look at. A reasonable drop during the coil current activity is usually acceptable. Any significant drop, however, needs to be reported. In order to calculate the depth of the voltage drop, the beginning and end of the drop need to be detected first. Utility functions “rise time” and “fall time” are called in the “ExtractDIP” algorithm (Fig. 32) repeatedly in order to catch as many voltage “DIP” cases as possible.

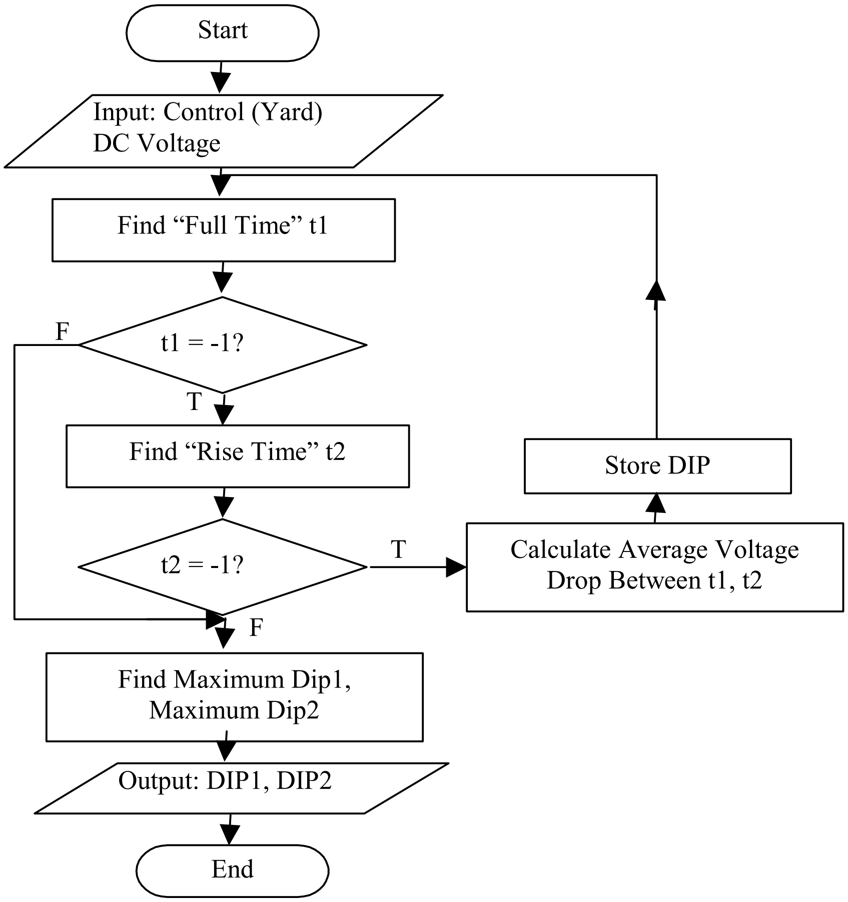


Fig. 32. Flowchart for function “ExtractDIP” of DC voltage

Each time a time slot [t1, t2] is detected, the voltage drop in the DIP time slot is calculate and stored. The maximum voltage drop both inside and outside the coil current (CC) activity time are obtained by comparing all the drop values. In order to tell whether a “DIP” happens within the CC activity time slot, a setting called “Time Tolerance” is introduced. As shown in Fig. 33, if the “DIP” falls into the CC activity time slot plus the tolerance setting, it will be considered as a DIP within the CC activity time slot. All the settings for DC voltage are listed in Table X.

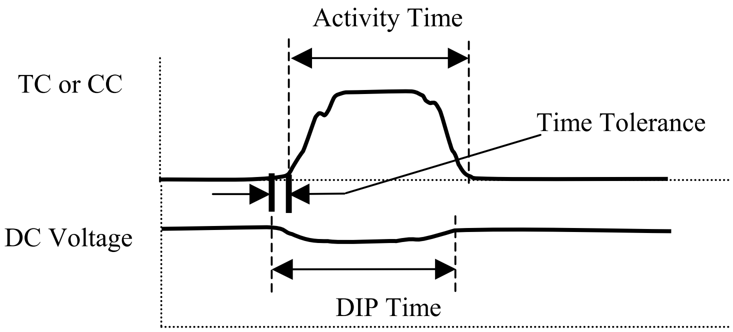


Fig. 33. Setting "Time Tolerance"

TABLE X
SETTINGS FOR DC VOLTAGES FEATURE EXTRACTION

Setting Name	Description	Typical Value
Time Tolerance	Allowed time difference between detected DIP time slot and Coil Current activity time slot.	10 samples
PSD Setting	Power spectrum density setting	10 decibel

Extracting ripple parameter "RIP" begins with a signal decomposition that is splitting the ripple feature from other features in DC voltage. This process has been discussed and it was concluded on page 25 that detail signal component at level 3 can be used to represent the ripple feature. Further calculation on ripple feature is taking place in the frequency domain. Power spectrum density (PSD) estimate is made via Burg's method. Especially, PSD at 60Hz, 120Hz and 240Hz are evaluated since the ripple feature is generally composed of system frequency and its multipliers. Fig. 34 and Fig. 35 show the PSD for DC voltage with or without ripple feature. If there is no outstanding ripple feature, the PSD of the signal either has no peak value at a multiple of system frequency or the value is comparatively small. If the peak values at the multiple system frequencies are larger than the PSD setting (listed in Table X), two cycles of signals are selected to calculate the signal parameter “RIP”, which equals max(S)-min(S), i.e., the amplitude of the ripple. Souce code for this function is provided in Appendix I

In the algorithm for extracting distortion parameter “ExtractDIS”, wavelet decomposition at level one and four are used. Fig. 36 gives out the wavelet decomposition coefficients at level one for the following cases: normal, with excessive ripple, with distortion and with spike.

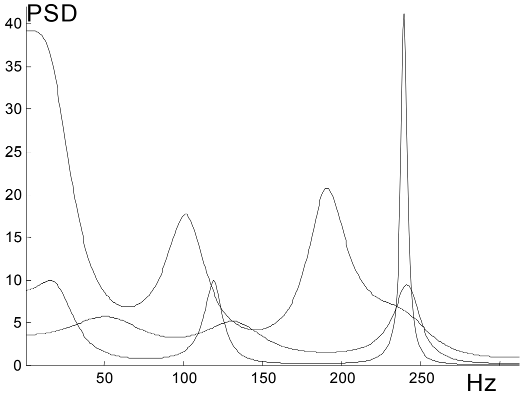


Fig. 34. PSD for signals with ripple feature

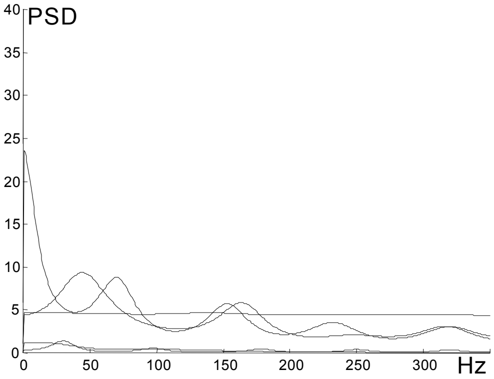


Fig. 35. PSD signals without ripple feature

Usually the power of decomposed-coefficients at level one is enough to tell the distortion from the normal case. In the following four pictures one can notice that for distorted signal (the third picture), the amplitude of coefficients is larger and the density is higher comparing with the coefficients for other conditions. Furthermore, the median absolute deviation (MAD) of coefficients is calculated and assigned to parameter DIS, which is used to distinguish the distortion.

In some cases, the calculated parameter DIS for a signal with excessive ripple may be large enough for the expert system to falsely identify a ripple as a distortion abnormality. In order to avoid such a mistake, WAVELET decomposition coefficients at level four are also calculated. It is observed that for a ripple, the coefficients at level four are larger than the coefficients at level one while for a distortion case the coefficients at level one are larger than the coefficients at level four. In this way, one can tell the ripple from the distortion and assign zero to parameter DIS when excessive ripple is detected through comparation of WAVELET decomposing coefficients at different levels.

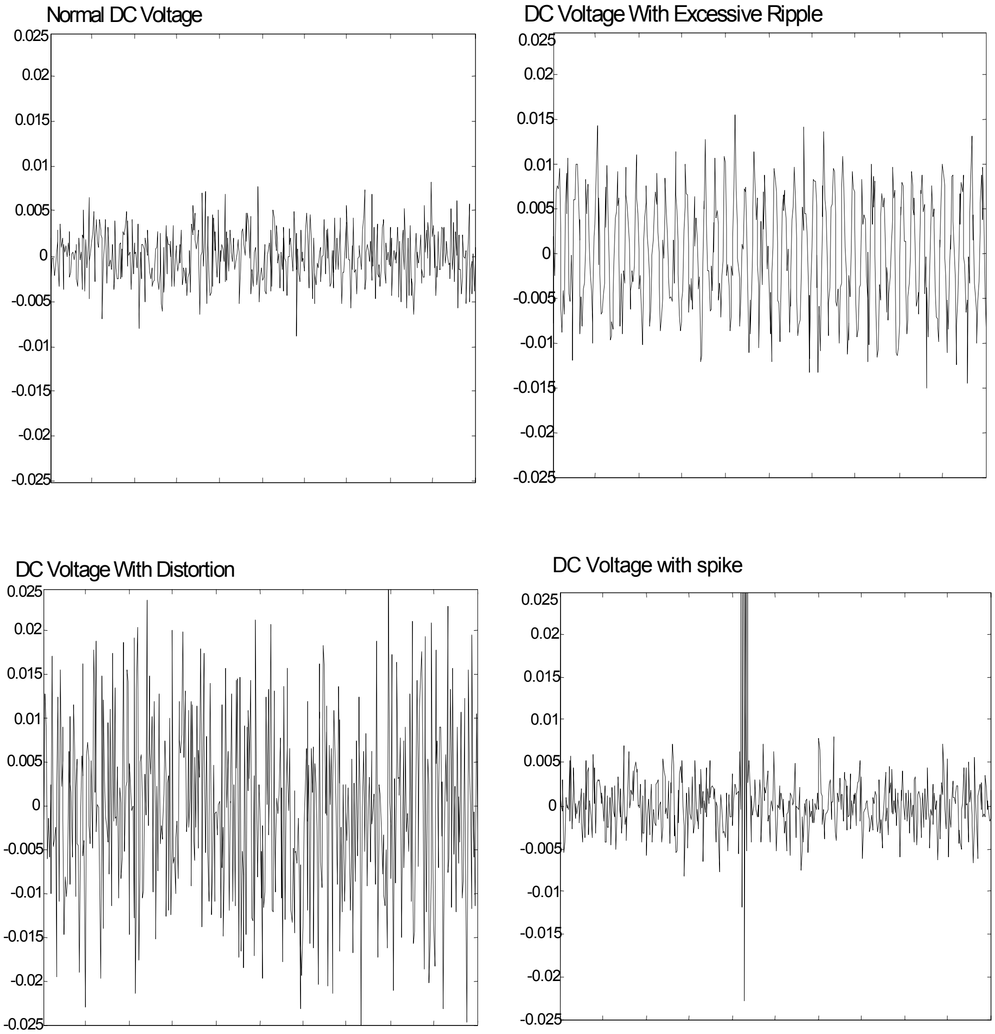


Fig. 36. Wavelet decomposition at level one for extracting distortion parameter

The algorithm for extracting spike related parameter SPI is similar to that of DIP. The signal mean represented by the horizontal line in the middle of Fig. 37 is calculated first, and two boundaries around the mean are set. During the detection, boundaries are moving toward the middle line until any one of them reaches the signal. The length of the excessive drop is checked here again to tell a spike from a dip, which has already been discussed before.

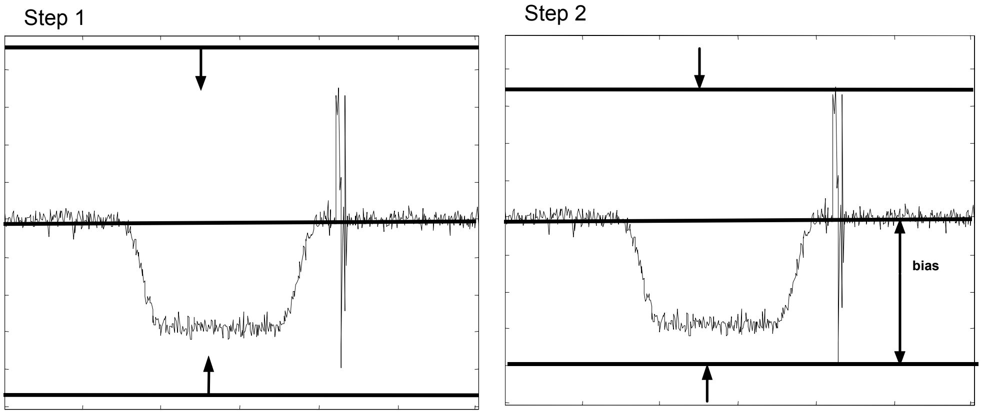


Fig. 37. Algorithm for ExtractSPI

Coil Current

After the event feature extraction is performed for coil current, events T2, T3, T4 are located in the waveforms. Function "ExtractCHP" is called to calculates signal parameter NOI, which is proportional to the noise energy in the original signal before the denoising process. The first step in this calculation requires that a wavelet decomposition be performed to the level 2 with the Daubechies #2 (Db2) wavelet. All of the detail coefficients from both levels are squared and summed up. The resulting number is multiplied by a scaling factor and saved as the energy parameter. This energy parameter is then compared to an energy threshold shown in Table XI to determine if there is choppy-like noise on the signal.

TABLE XI
SETTINGS FOR COIL CURRENT PARAMETER EXTRACTION

Setting Name	Purpose	Typical Value
Energy Threshold	Used to determine if there is excessive noise in the signal.	15

Phase Currents

In the case of a opening operation, function “ExtractRES” is called to detect a restrike feature on the signal. A restrike is detected when the function finds that the signal crosses back above the threshold after the first transition to zero. Utility function “rise time” is called to detect the restrike.

The output is signal parameter RES, a boolean value that indicates the existence of a restrike.

Conclusion

During the development of the pre-processing algorithms, features are identified and feature extraction algorithms are designed in the form of functions. Each extraction function outputs a pre-defined feature.

CHAPTER VI

SOLUTION ARCHITECTURE

Introduction

Previous chapter discussed the feature extraction algorithms by functions. Although we know that one function may use the service of another, a clear relationship among functions, between functions and data were not established. Furthermore, the description only provides a static view of each function, and the interaction among the functions and data is missing. These functions act like building blocks, and this chapter is responsible for organizing them into a whole solution.

An object-oriented pre-processing model is defined and implemented in the visual c++ [19]. Object-oriented approach is a well-established concept in software engineering. It uses techniques such as encapsulation, inheritance and polymorphism to achieve readability and reusability so that the source code is easy to maintain and extend.

Nothing is more natural than the way we look at the world around us. We easily identify different objects, trees, a bank, a company, family members, etc. Object-oriented technology borrows the idea, and maps the objects from the real world into the software application.

All object share two common things: they possess attributes and provide services. Attributes are “some variables (data or information) for which each object has its own value” [20]. Services are “the advertised or public work that an object is willing to perform when requested by another object” [20]. A group of objects that share the same attributes and services are generalized as a class. A class encapsulates data attributes that can be read or written as well as functions that can provide services to the external world. A class diagram is used to account for different classes, their attributes, and functions. Besides the encapsulation, the class uses inheritance to address the relationship among different object. A new class can be derived from an existing class, and the new class is called a child of the class it is derived from. A class may be derived from more than one class. When a class is derived from another class, it may inherit its parent class member functions and data members. By deriving classes from other classes, it allows the application's developers to reuse existing code. An inheritance diagram can be used to describe the relationship among different classes.

Besides the concepts of objects and classes, the interactions among different objects are called message passing. When a function is called, a message is sent to the class the function belongs to. During the execution of this function, more functions may be called, messages generated and then sent to other classes. Sequence diagram are used to trace the interactions (or behaviors) of the classes.

Pre-processing modeling in this chapter follows the procedure provided below.

- Define Functional Requirements
- Generalize Classes Using Class Diagram and Inheritance Diagram
- Describe Interactions Using Sequence Diagram

Functional Requirements

The pre-processing module has to be capable of reading original data samples acquired by the data acquisition unit connected to the circuit breaker. The format of the collected raw data is assumed to be IEEE COMTRADE binary data format [21]. Based on this assumption, the following requirements can be defined:

Reading configuration file: Configuration file needs to be read in order to determine the number, allocation and characteristics of the channels of the data acquisition system.

Reading data samples file: Data sample file needs to be read in order to obtain samples for each channel of the data acquisition system.

The pre-processing module has to be capable of extracting features from the original set of data samples. The choice and accuracy of feature (or signal parameters) calculation may profoundly affect the results of the analysis. Some advanced processing tools that may be used for feature extraction including wavelet analysis, Fourier analysis, and digital filtering.

Wavelet Analysis: Wavelet analysis has an important role in signal processing discussed in this document. Its ability to provide time-frequency localization corresponds very well to the type of signal processing that needs to be performed. Use of Discrete Wavelet Transform (DWT) is expected. In our case, Daubechies family of wavelets is used.

Fourier Analysis: Fourier analysis is a classical tool in signal processing and analysis. Its use is related to obtaining information on frequency spectrum of the signal. Implementation of the Fast Fourier Transform (FFT) is needed.

Filtering: Digital filters can remove or extract certain signal components of the signal frequency spectrum. Common use of filters is to eliminate measurement noise. Low-pass and possibly other types of IIR filters will be needed. Certain other smoothing techniques could also be needed depending on the exact signal features.

Since features may vary from type to type, a mechanism called settings is created to customize the processing for different types of breakers.

Class Diagram

A basic method to identify the classes in the system is to write down functional requirements for the system and pick up all the nouns as the classes, attributes and function candidates such as DFR, original data sample, signal processing, circuit breaker, etc. Obviously there is no single best solution in the design of the classes.

Another method of identifying the classes is to group the functions that have been developed in the last chapter, and generalize the classes by one common reason why the functions can be organized into a group. For example, to put all the extraction functions belonging to the same signal together, and generalize the signals as classes. Fig. 38 identifies a class named “Contact” with four functions that are used to extract four features for “A” and “B” Contacts. In this way six classes can be identified from the table listing all the signal channels:

- Initiate: Close initiate, trip initiate
- DCVoltage: Control DC Voltage, Yard DC Voltage
- Contact: A Contact, B Contact
- XYCoil: X Coil, Y Coil
- PhaseCurrent: Phase A, Phase B, Phase C Currents
- CoilCurrent: Trip Coil Current, Close Coil Current

Utility functions also need to be grouped. Because the extraction functions belonging to different signal classes share the services provided by utility functions, it is not appropriate to put the utility functions in any of the signal classes defined above. We define a new general class called “signal” to contain all the utility functions. To preserve the encapsulation and enable access to the utility functions, general class “signal” is defined as the parent of the other specified signal class such as “Contact” shown in Fig. 38. Therefore all the utility functions can be inherited by the child classes, which means child class can use utility functions directly without redefining the functions in its class again.

Class Name: Contact
Attributes List:
Function List: 1. ExtractTIM 2. ExtractDIP 3. ExtractBOU 4. ExtractNOI

Class Name: Signal	
Attributes List:	
Function List:	
1. trimavg	6. derivative
2. risetime	7. decompose
3. falltime	8. denoise
4. max	9. getcoeff
5. min	10. pburg

Fig. 38. Initial class diagram for class “Contact” and “Signal”

The attribute list is still missing in the class diagram. Each specified signal has its own signal parameters and settings, and they can be put into the attributes of the specified signal classes. All the signals have the original data samples and they can be put into the attributes of the general signal class so that all the specified signal classes can inherit it. Some settings used by the utility functions should also be put into the general signal class. Fig. 39 places one specified signal class “Contact” and the general signal class in parallel to allow easy comparison between them.

Even though the attributes and functions are organized into classes and relationship among different classes is established, the encapsulation process has not been finished yet. Besides the meaning of containing, encapsulation also has an implication of hiding. Attributes are usually hidden to prohibit access from an outside class and to protect the data. Functions are hidden for the purpose of simplifying the interface and releasing the burden of knowing the detail implementation. Hiding is realized by assigning different access privileges “Public”, “Protected” and “Private” to attributes and functions. “Public” attributes and functions allow non-limited access, they can be reached both inside and outside the class (access from other classes). “Protected” attributes and functions allow limited access, they can be reached only from inside the class or from their inherited classes. “Private” attributes and functions allow no access outside its class.

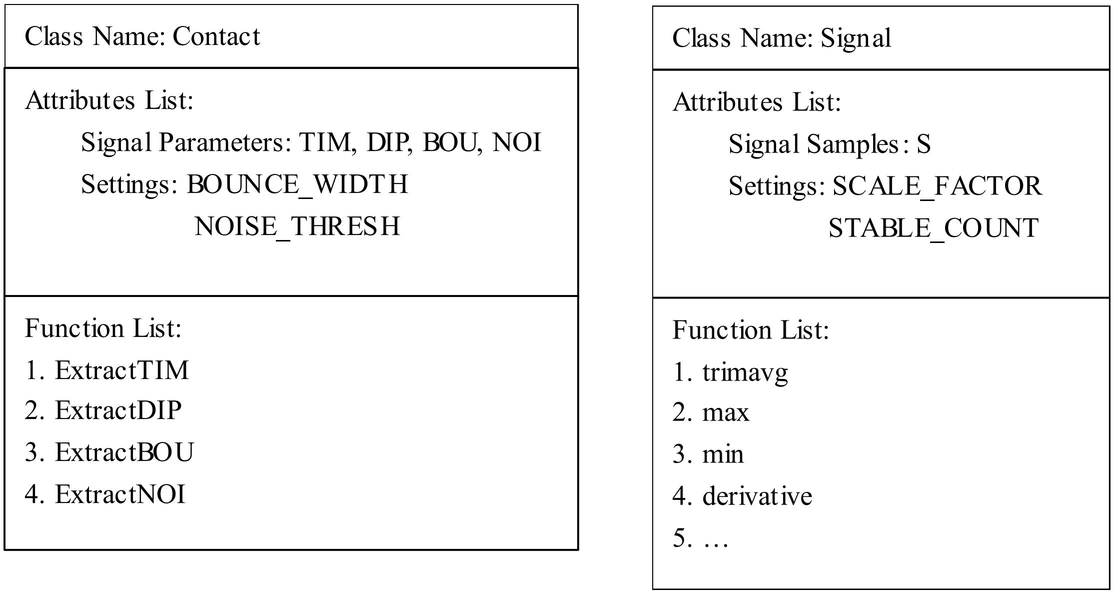
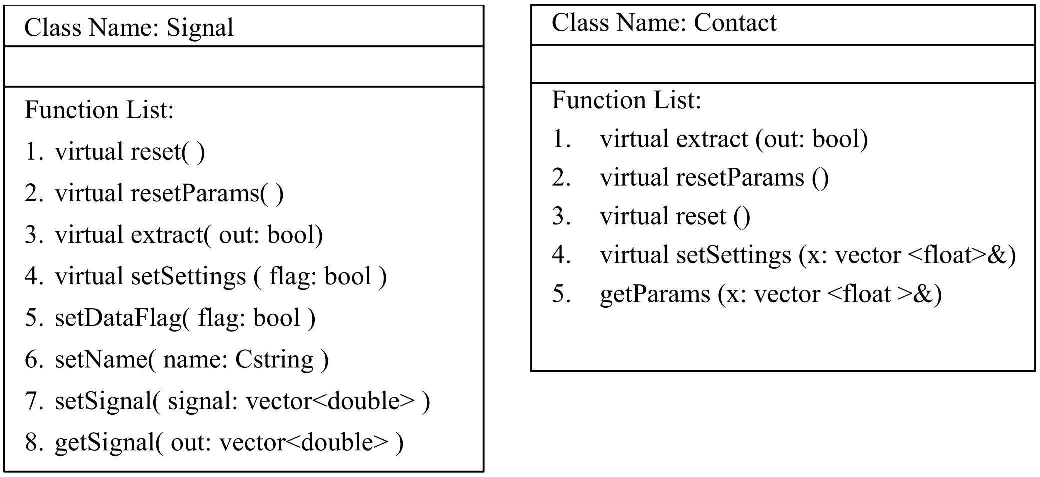


Fig. 39. Class diagram for “Contact” and “Signal” with attributes and functions

To further encapsulate the classes shown in the above figure, the attributes and extraction functions of the specified signal classes are assigned to the “private” privilege. The attributes and utility functions of the general signal class are assigned to the “protected” privilege so that all its child classes can reach the data and use the services. After the hiding process, the attributes and functions list of both general class “Signal” and specified signal class example “Contact” are left empty. The next task is to define functions that provide the interface. Interface functions will be given “Public” access privilege. They define how to access the class data and use the class services from outside world. For example, other classes may want to get the original signal samples to draw figures, they may want to set the settings, and most importantly they may want to initiate the signal processing and obtain the extracted signal parameters. Fig. 40 gives the final class diagram for the signal classes. All the private and protected attributes and functions are hidden from an outside access. Only one child class “Contact” is given in the figure since others are defined in the same way.



Class Name: Signal

Function List:
1. virtual reset( )
2. virtual resetParams( )
3. virtual extract(out: bool)
4. virtual setSettings (flag: bool)
5. setDataFlag(flag: bool)
6. setName(name: Cstring)
7. setSignal(signal: vector<double>)
8. getSignal(out: vector<double>)

Class Name: Contact

Function List:
1. virtual extract (out: bool)
2. virtual resetParams ()
3. virtual reset ()
4. virtual setSettings (x: vector <float>&)
5. getParams (x: vector <float >&)

Fig. 40. Final class diagram for the signal classes

Both parent and child classes have to share the same function names with a designation “virtual”. These functions are in fact implemented differently, which is called polymorphism. The mechanism allows a parent pointer to be initialized with any child object. When a virtual function in the parent is called through the pointer, the executed function has the same name as the virtual function belonging to the initialized child.

To summarize the discussion of the class generalization, Fig. 41 provides the inheritance diagram for the overall signal-processing module. Many of the base classes shown in the figure are MFC classes from the MFC class library.

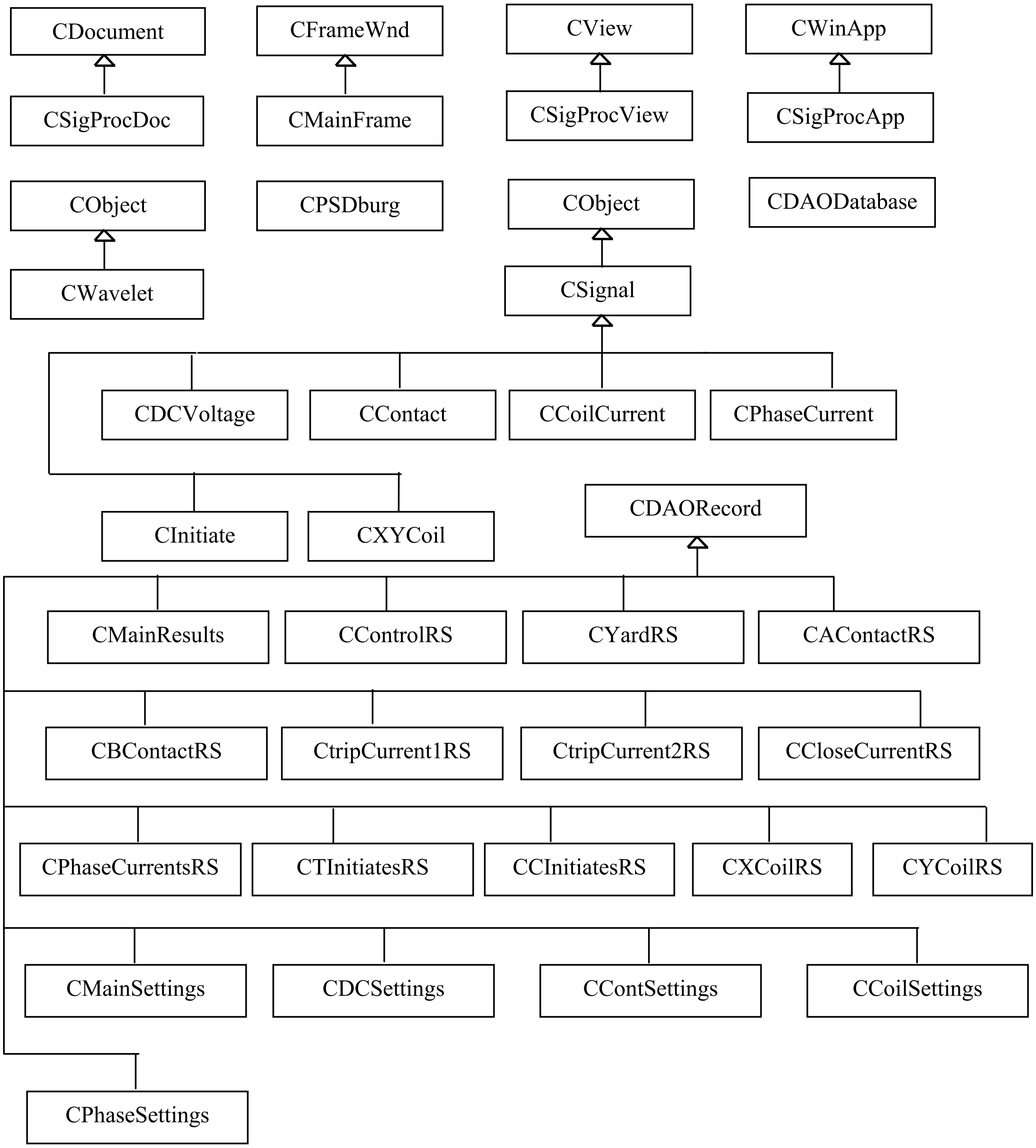


Fig. 41. Inheritance diagram

MFC AppWizard creates the following classes automatically: CSigProcDoc, CMainFrame, CSigProcView, and CSigProcApp [19]. They contain all the necessary code for implementing a simple interface and initializing the program. Think of an ordinary windows program, it basically contains three elements including the framework, the document and the view. All the framework elements such as menu, toolbar and status bars are organized in the class “CMainFrame”. Class “CSigProcDoc” provides access to the data and documents, therefore, all signal classes are defined within. Class “CSigProcView” provides viewing functions, for example, showing the waveform or the content of a document. Class “CSigProcApp” is responsible for initiating the application.

The classes CWavelet and CPSDburg are called utility classes because they consist solely of member functions and do not contain any data members. They are used for performing wavelet analysis and power spectral density analysis respectively. The classes derived from CSignal are used to represent all the signals that the circuit breaker monitoring system analyzes. All the classes that are derived from CDAORecordset are used to retrieve and store information in the database. Some of them are used to access the settings of the database and others are used to access the results of the database.

A composition diagram typically shows how all the objects are organized within the application. Fig. 42 shows the composition diagram for the signal processing module. Note that a majority of the objects are contained within the CSigProcDoc class as data members. The diamond symbol shown in the diagram indicates that an object or a group of objects is contained within a class. The numbers located next to each class indicate the number of objects of that class that are contained within the CSigProcDoc class. For example, the number two next to the CContact class indicates that there are two objects of the CContact class (“A” Contact and “B” Contact) contained within the CSigProcDoc class. The figure also shows that there is one wavelet object contained within the CSignal class. Since the classes CContact, CCoilCurrent, CPhaseCurrent, CInitiate, CXYCoil, and CDCVoltage are all derived from the CSignal class, each of those classes has access to that wavelet object and its associated member functions.

Sequence Diagram

The interactions among different functions and classes are initiated to extract signal parameters from the signals. System sequence diagram is used to account for the whole interaction. On the top of the system sequence diagram (Fig. 43) there are five objects including “:CMainFrame”, “:CSigProcDoc”, “:CSignal”, “:CDaoRecordset”,

":CSigProcView". These objects, enclosed within a rectangular box, are instances of their respective classes. Notice that the representation of an object is different from that of a class, and it includes underlined class name and a colon before it. The vertical line beginning from the bottom of an object box represents the object interface. The horizontal arrow line indicates message that is sent from one interface to the other. The leftmost vertical arrow line shows that the time sequence begins from the top and ends at the bottom.

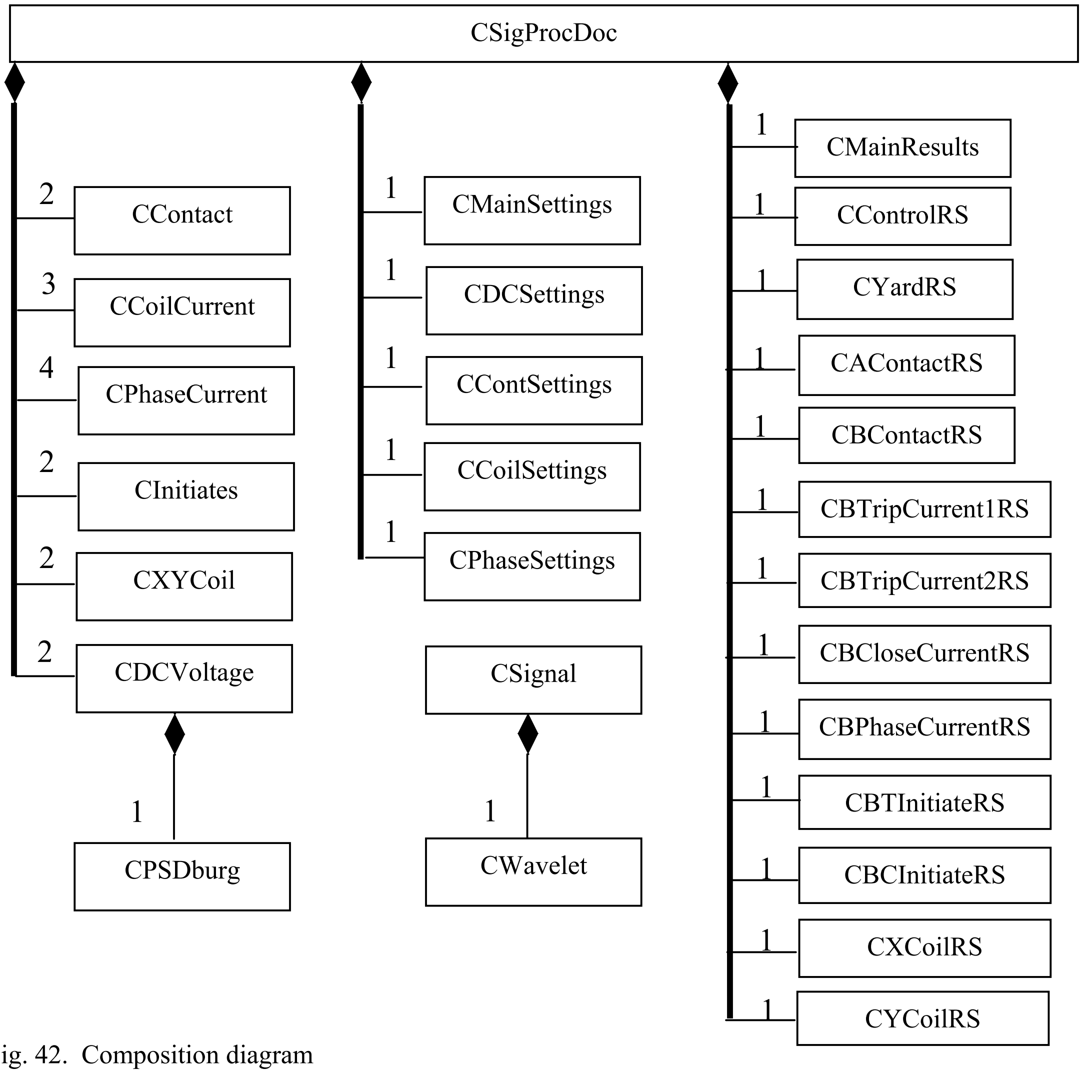


Fig. 42. Composition diagram

To begin with, an OnExtractStart message is sent to object CMainFrame checking the availability of the signal record. If the signal record file is not found or the settings for the signal record are missing, this message returns a null. In that case, the process exits with an indication of the failure.

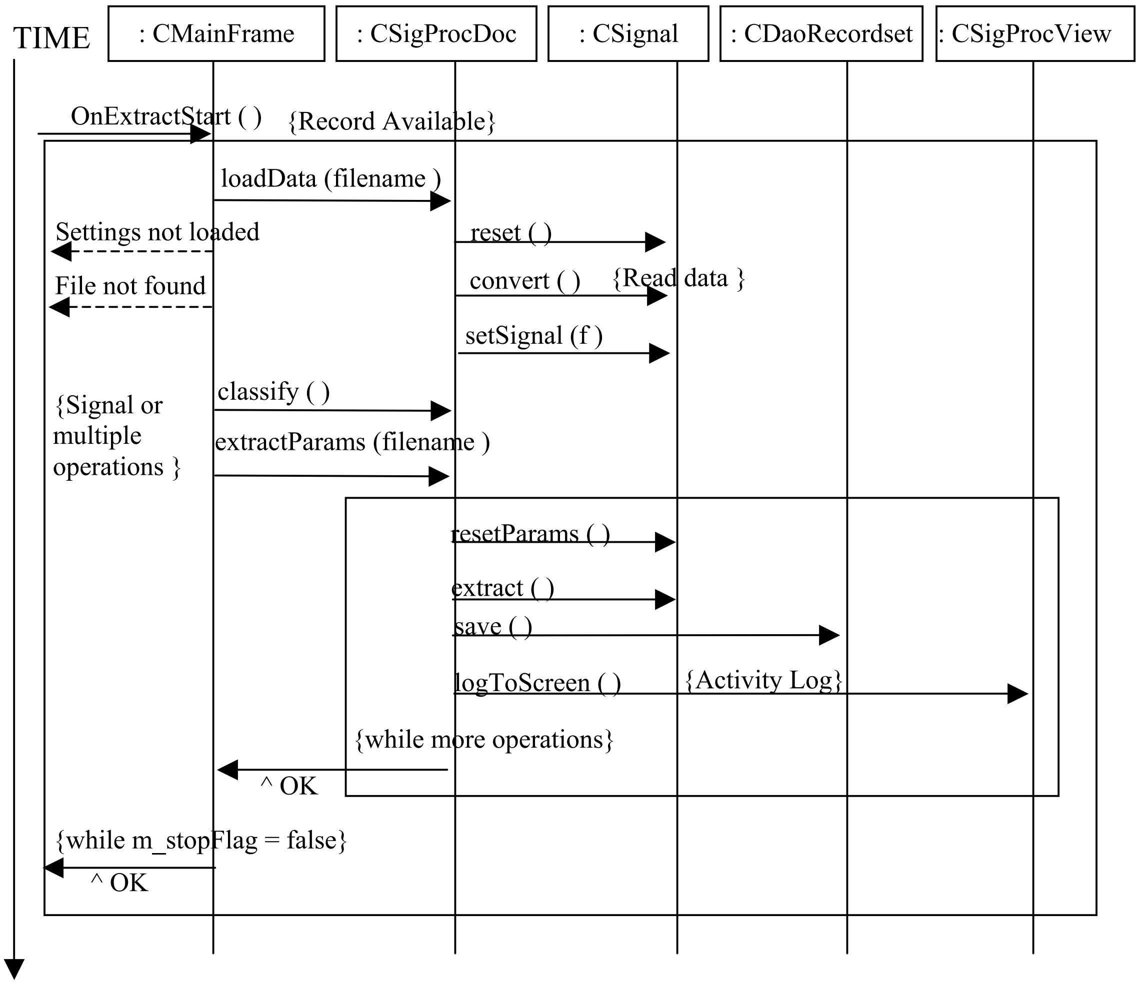


Fig. 43. Sequence diagram

If both the record and settings are available, three more messages are sent from CMainFrame object to CSigProcDoc object. These messages in turn evoke other messages like a chain reaction. For instance, upon receiving the loadData(filename) message, CSigProcDoc will sent out the “reset”, “setSignal” and “convert” messages in sequence to different derived objects of CSignal.

The “classify message” step asks the CSigProcDoc object to find out how many operations are in the record. Since operation is supposed to be extracted separately, a loop structure is taken to process all the operations residing within one record. The extract process is started by the extractParams message and continued until there is no more operation. The loop structure is enclosed within a rectangular box.

After the extract process is completed, a success message is sent back from CSigProcDoc object to CMainFrame object. In our application, 12 channels of signal records are waiting for processing, and the whole extract process is within a repetitive structure.

Conclusion

The architecture for data pre-processing in the automated circuit breaker condition analysis system is implemented by the visual C++, which is an object-oriented programming language. A requirement description is provided at first to clearly state the objective of the design and implementation. Major classes are identified step by step, and important object-oriented mechanisms including encapsulation, inheritance and polymorphism are introduced during the class identification process. Class diagrams and hierarchy inheritance diagram summarize the static design of the overall signal pre-processing system. In the end, a dynamic system view is provided through the system sequence diagram.

Both design and implementation of the pre-processing architecture take the advantage of the object oriented software engineering technology. For the design part, unified modeling language (UML [20]) is used to generate the class diagrams and sequence diagrams. For the implementation part, Microsoft fundamental classes (MFC [19]) are used to set up the software framework. These ingredients help to build a stable, readable and extendable software in the end.

CHAPTER VII

SOLUTION EVALUATION

Introduction

In order to evaluate the analysis solution, a two stage testing is launched first. Solution testing is a critical phase because it helps to evaluate the work including system requirements, designs, and implementation. The test consists of test data, test procedure and test results. The automated condition analysis system for circuit breaker (CB) is designed for solving the real world problem, and it is preferred that real data collected directly from CB be used as the system inputs. Test procedure needs to be carefully designed so that all parts of the source codes are fully tested. Test results need to be studied and insights of the results should be provided in order to evaluate the overall performance of the system.

Test Procedure

The test procedure for this system is divided into two stages. The test goal in the first stage is to find whether every feature is extracted correctly from the original data. A backwards-linking method is used here. Features are first predicted or calculate manually by an expert. The prediction and calculation are based on what is observed in the original data and expert's personal experience. Then the data is fed into the analysis system, and the extraction outcome will be compared with the known features. Any discrepancies between the two results may justify a review of the design and implementation of the feature extractor.

A detailed procedure for first stage testing is shown in Fig. 44.

In the object-oriented module defined in the previous chapter, feature extractors are organized as functions within different signal classes. For example, the ripple extractor for the DC voltage takes the form of a private function "ExtractRIP" belonging to the class "CDCVoltage". To test each function within user-defined classes is equivalent to testing each feature extractor. Therefore, both the test data and results are organized and presented in the same way.

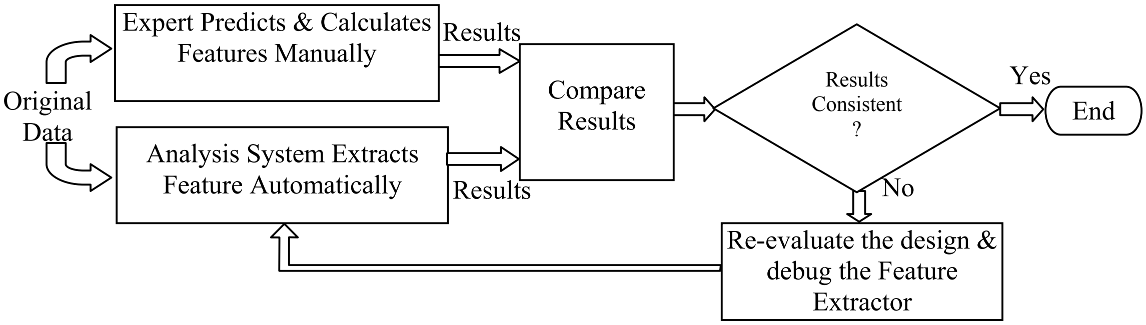


Fig. 44. First stage test procedure

The test goal in the second stage is to build up settings for different types of breaker, thus the data are organized according to the breaker type. The test method used here are forward-linking method. Certain amounts of data for each type of breakers are collected. Before testing, features are unknown to the people involved in the testing. A detailed procedure for the second stage testing is shown in Fig. 45.

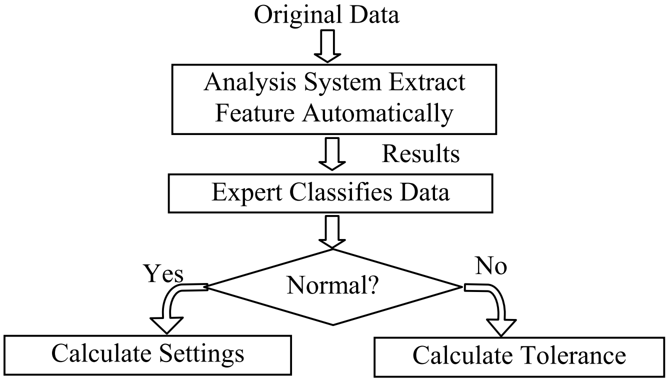


Fig. 45. Second stage test procedure

The data is directly fed into the automated analysis and the results will be compared with other data of the same breaker type. Expert then classifies the data into two categories: normal or abnormal. It is expected that most of the data will fall into the normal category, and the signal parameters will be used to establish the settings. The abnormal data possess one or more features that are either much larger or much smaller than the settings. Their values are used to limit the tolerance so that all the abnormal feature values will fall outside the tolerance.

Test Data

This section provides basic information on the collection of data that are being used to test CB analysis system. Since the test procedure has been divided into two stages, the data collection efforts are also split into two parts.

The first effort is narrowed down to only two types of CBs: Westinghouse R3 and AA10-80. Westinghouse is the manufacturer, "R3", and"AA10-80" are CB types. This is due to the difficulty to find all features from in-service breakers within a short period of time. Instead, two CBs (one of type R3 and the other AA10-80) are taken into a laboratory, and features are created deliberately. Table XII organizes the collected test cases according to the signal type and the functions under testing. In summary, all the test cases meet the following requirements:

1) Belongs to R3 or AA10 breaker type from manufacturer Westinghouse;
2) Follows COMTRADE file Format [21];
3) Must have desired signal features.

The second effort is to collect at least ten records for each type of CB, five for opening operation and five for closing operation with the assumption that most in-service breaker are in good condition. Altogether 454 records of 11 known breaker manufacturers and 31 breaker types are collected for testing. All these records are collected from the in-service breakers in 16 different substations. Table XIII organizes the collected test cases according to the breaker types. In summary, all the test cases in this table meet the following requirements:

1) Collect at least five open-operation and five close-operation cases from different breaker types;
2) Follows COMTRADE file Format [21];
3) Must be collected from in-service breaker.

In Table XIII there are 15 breaker types that do not have enough test cases. More time and efforts are needed for collecting them.

TABLE XII
SUMMARY OF TEST CASES I

Signal Type	Feature Parameter	Functions Under Test	Features Descriptions	Minimum Cases
Trip or Close Initiate	T1	ExtractTIM	Resets Prematurely	1
	DRP	ExtractDRP	Drops Out During Operation	1
X or Y Coil	T8, T9, T10	ExtractTIM	No Activation	1
			Activity Delayed	1
			No Deactivation	1
			Deactivation Delayed	1
			Deactivation Premature	1
	DRP	ExtractDRP	Drops Out When Energized	1
Contact	T5, T6	ExtractTIM	Contact Flat	1
			Contact Premature	1
			Contact Delayed	1
	DIP	ExtractDIP	Excessive Dip (Contact Unstable)	1
	BOU	ExtractBOU	Contact Bounce	1
	NOI	ExtractNOI	Noisy on Contact	1
DC Voltage	DIP1	ExtractDIP ExtractTIM	Excessive Dip within Coil Current Activity Period (DC Unstable)	1
	DIP2	ExtractDIP ExtractTIM	Excessive Dip outside the Coil Current Activity Period (DC Unstable)	1
	RIP	ExtractRIP	DC Rippled	1
	DIS	ExtractDIS	DC Distorted	1
	SPI	ExtractSPI	Spike on DC	1
Coil Current	T2	ExtractTIM	Coil Current Flat	1
			Pickup Premature	1
			Pickup Delayed	1
	T3	ExtractTIM3	Dip Delayed	1
	NOI	ExtractCHP	Coil Current Distorted	1
	T4	ExtractTIM	Coil Current No Drop	1
Phase Current	T7	ExtractTIM	No Drop in Trip operation	1
			No Rise in Close operation	1
			Delayed	1
	RES	ExtractRES	Re-strikes	1
Total Test Cases Needed				29

TABLE XIII
SUMMARY OF TEST CASES II

Manufacturer	Breaker Type	Num of Open Cases	Num of Close Cases
ABB	15GHN750	5	5
	R4	14	13
	V	10	10
GE	FK-69-7500	2	2
	FKD-38-12000-1	17	18
	VIB-15.5-20000-2	24	25
	VIB-15.5-500	2	2
GEC	OMX 15	1	1
HITACHI	MFPTB-300-63LA	8	8
ITE	VBK	3	3
POWELL VAC	15PV750-2	7	8
SA	BCM-15-1	2	2
	SDO-15-500	5	5
	SE-3B	2	2
SIEMENS	GM1	1	1
	SDV	8	8
	SP-72.5-43-3	1	1
	3AF-TF3C	1	1
SD	FLUARC	21	22
	VACARC	6	6
	FLUARC-FVBS1121120	9	9
	FLUARC-SF6	2	1
S&C	FVR 1121125A	6	6
	SD	5	5
WH	R2	6	6
	R3	36	39
	R4	9	9
	AA10	3	4
	245SP1500	1	0
	345SP1500	4	3
	V	2	2
	V6	2	2
Total Received	454	225	229

Test Results

First Stage Testing

Four test cases are selected (Table XIV), each representing a typical CB problem such as slow breaker, stuck breaker, velocity decreased and trip latch maladjustment. Several features may appear in the same case, some of them make contribution to the major problems mentioned above and others contribute to minor problems, such as a noise contact or an unstable DC voltage.

TABLE XIV
SELECTED TEST CASES

Test Case	Breaker Type	Features Description
1. Slow Breaker Closing	R3	1) Breaker closes! 2) Control voltage unstable! 3) Control voltage spike! 4) 'A' contact noise! 5) 'A' contact delayed! 6) 'B' contact delayed! 7) Close coil current drop delayed! 8) Phase Current transition delayed! 9) Y Coil no activation!
2. Stuck Breaker	R3	1) Breaker opens! 2) 'A' contact flat! 3) 'B' contact flat! 4) Trip coil current not drop! 5) Phase A Current did not drop! 6) Phase B Current did not drop! 7) Phase C Current did not drop!
3. Velocity Decreased	R3	1) Breaker closes! 2) 'A' contact delayed! 3) 'B' contact premature!
4. Trip latch maladjustment	AA10	1) Breaker opens! 2) Trip coil current dip delayed! 3) Travel time reduced!

Table XV gives the results of signal parameters that measure a sequence of events. If a signal or a feature of the signal does not exist, “–2” will be used as the default value for the signal parameter that measures the non-existing feature. For example, a Westinghouse R3 breaker does not have the dip event during the energizing of coil current. The signal parameter T3 that is used for measuring this event is set to "–2" for case #1, #2 and #3. If an event does not happen, "–1" will be used as the default value for the signal parameter. For example, CB in case #2 is a stuck breaker. Because the phase currents, "A", "B" contacts and trip coil current do not make their expected status transition during its operation, the related signal parameters are set to "–1". Since signal parameter "T1" measures the trigger point, it is always set to 0.

TABLE XV
RESULTS FOR SELECTED CASES

Signal Name	Signal Parameter	Case #1	Case #2	Case #3	Case #4
Close (Trip) Initiate	T1	0.00	0.00	0.00	0.00
Close (Trip) Coil Current	T2	0.03073	0.00069	0.03108	0.00069
	T3	-2	-2	-2	0.01892
	T4	0.09288	-1	0.08351	0.04201
B Contact	T5	0.07726	-1	0.05503	0.06476
A Contact	T6	0.12431	-1	0.09583	0.02951
Phase Currents	T7	0.12621	-1	0.09201	0.03073
X Coil	T8	0.00000	-2	0.00000	-2
	T9	0.08837	-2	0.07813	-2
Y Coil	T10	-1	-2	0.07986	-2

In order to tell whether the measured signal parameters is delayed or premature, settings and tolerances are provided in Table XVI. Settings define the expected value of the signal parameter and tolerances define the deviation from the settings. If the signal parameter value falls into the area defined by the setting and tolerance, it will be considered as normal. If the signal parameter value falls out of the defined area, it will be considered as either a delay or a premature event. All the values including settings, tolerances and signal parameters are provided in time unit of second.

To visualize the event sequence, event-time plots for each case are provided in Fig. 46. A horizontal line marks 0 second as the trigger line. Each star in the figure represents an event, and each bar represent the area defined by the settings and tolerances. The first event "T1" is drawn on the trigger line.

TABLE XVI
SETTINGS AND TOLERANCE FOR SELECTED CASES

Signal Para.	Settings for Case #1, #3	Tolerance for Case #1, #3	Settings for Case #2	Tolerance for Case #2	Settings for Case #4	Tolerance for Case #4
T1	0.00	N/A	0.00	N/A	0.00	N/A
T2	0.03120	0.0055	0.00052	0.001	0.00087	0.001
T3	-2	N/A	-2	N/A	0.01040	0.0025
T4	0.08300	0.008	0.03784	0.005	0.04201	0.005
T5	0.06589	0.01	0.05530	0.01	0.06476	0.01
T6	0.08324	0.01	0.02840	0.01	0.02950	0.01
T7	0.08500	0.01	0.03100	0.008	0.02934	0.008
T8	0.00000	0.008	**N/A**			
T9	0.07800	0.008				
T10	0.08170	0.008				

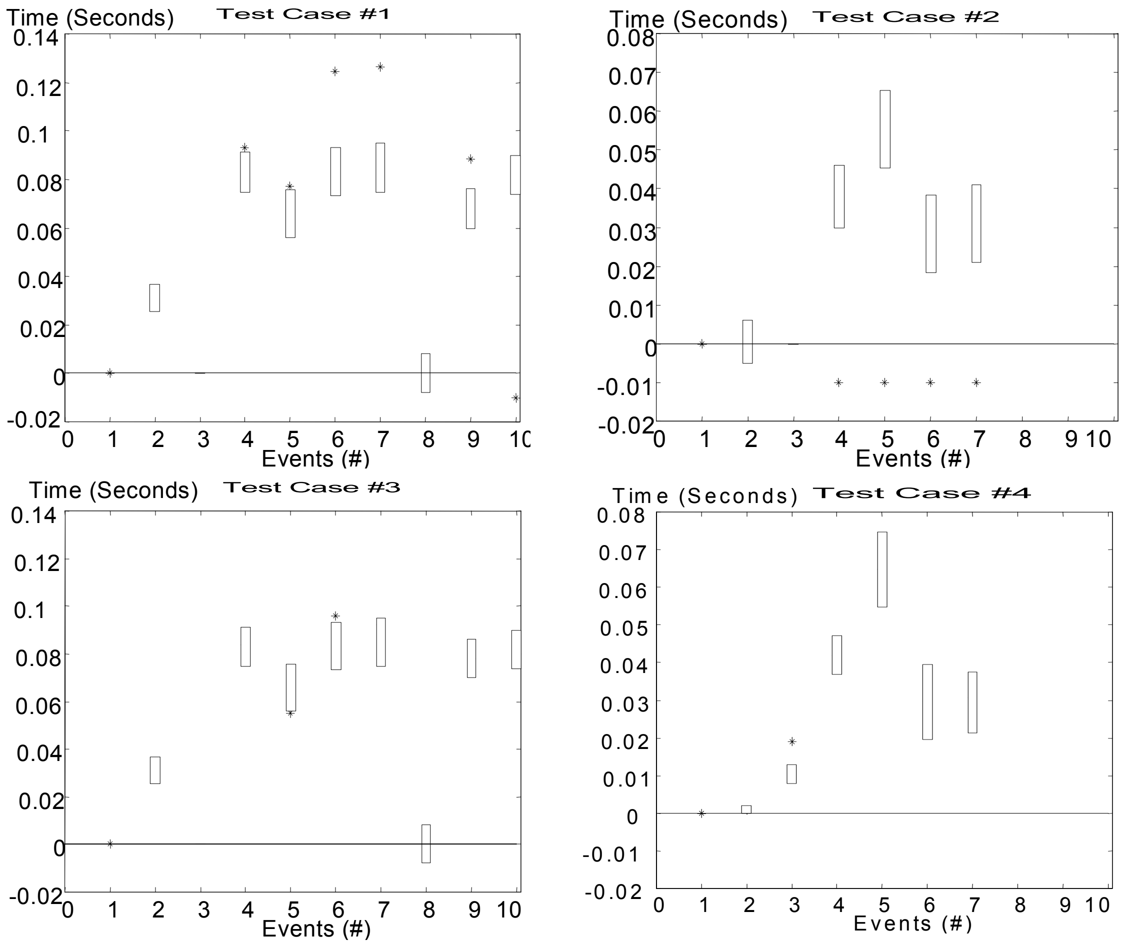


Fig. 46. Results for sequence testing

Any event (cross in Fig. 46) drawn below the trigger line indicates that the event did not happen. Test case #2 for a stuck breaker is such a good example. For a given event, a bar defines the normal area that the event shall fall into. If the event is shown outside that area, it is an abnormal case indicating a CB problem. For the slow breaker in case #1, most events appear above the bar revealing the delay in the CB closing. For the breaker in case #3, event #5, "B" contact transition time, is premature; in event #6, "A" contact transition time, is delayed. The increase in time difference between these two events shows a decreased velocity. CB in case #4 has a delayed dip (event #3), which reveals a problem with the trip latch.

The signal parameters results that are not related with event sequence are shown in Table XVII.

TABLE XVII
CONTINUED RESULTS FOR SELECTED CASES

Signal Name	Signal Parameter	Case #1	Case #2	Case #3	Case #4
Control DC Voltage	DIP1	4.17	0.00	4.50	0.00
	DIP2	**20.89**	3.62	3.28	0.00
	RIP	0.04	0.00	0.04	0.05
	DIS	0.00	0.00	0.00	0.00
	SPI	**37.56**	0.00	6.64	0.00
Yard DC Voltage	DIP1	3.60	0.00	4.07	**-2**
	DIP2	4.13	3.62	4.40	**-2**
	RIP	0.05	0.00	0.05	**-2**
	DIS	0.00	0.00	0.00	**-2**
	SPI	5.19	0.00	6.66	**-2**
A Contact	DIP	0.00	0.00	0.00	0.00
	BOU	False	False	False	False
	NOI	**True**	False	False	False
B Contact	DIP	0.00	0.00	0.00	0.00
	BOU	False	False	False	False
	NOI	False	False	False	False
Close or Trip Coil Current	SUP	False	False	False	False
	NOI	11.95	0.00	12.03	4.27
Phase Currents	RES	False	False	False	False
Trip or Close Initiate	DRP	False	False	False	False
X Coil	DRP	False	False	False	False
Y Coil	DRP	False	False	False	False

The results show that case #1 also has problems like unstable control voltage (determined by signal parameter DIP2), control voltage spikes (determined by signal parameter SPI), and a noise "A" contact (determined by signal parameter NOI). For case #4, all the signal parameters of yard DC voltage are assigned to "–2" because the AA10 breaker does not have yard DC voltage.

Second Stage Testing

Table XIII lists the typical time sequences for 18 types of CB, and each has at least 5 test cases for each operation. The time unit is second.

TABLE XVIII
TEST RESULTS FOR 18 TYPES OF BREAKERS

Signal Parameters	Manufacturer & Breaker Type					
	ABB 15GHN750		ABB V		ABB R4	
	OPEN	CLOSE	OPEN	CLOSE	OPEN	CLOSE
T2	0.00153	0.00219	0.00160	0.02558	0.00315	0.02639
T3	0.01548	0.01222	-2	-2	0.03062	0.04797
T4	0.03820	0.05920	0.02398	0.05787	0.05597	0.09687
T5	0.05868	0.04727	0.02867	0.05446	0.04408	0.07943
T6	0.03042	0.07809	0.02338	0.06250	0.03761	0.08695
T7	0.03564	0.07501	0.02500	0.05800	0.03800	0.08900
T8	-2	-2	-2	0.0	-2	0.00000
T9	-2	-2	-2	0.05490	-2	0.07917
T10	-2	-2	-2	0.06490	-2	0.09693

Signal Parameters	Manufacturer & Breaker Type					
	GE FKD-38-12000-1		GE VIB-15.5-20000-2		SIEMENS SDV	
	OPEN	CLOSE	OPEN	CLOSE	OPEN	CLOSE
T2	0.00109	0.00126	0.00162	0.00107	0.00170	0.00229
T3	0.01753	0.01541	0.01473	0.01310	0.01649	0.01781
T4	0.07157	0.06083	0.03028	0.03470	0.03649	0.04299
T5	0.08351	0.05255	0.03424	0.05872	0.03802	0.03800
T6	0.06189	0.07170	0.02820	0.06891	0.03400	0.04600
T7	0.03073	0.09355	0.02700	0.06374	0.03337	0.04312
T8	-2	-2	-2	0.02610	-2	-2
T9	-2	-2	-2	0.05000	-2	-2
T10	-2	0.06225	-2	0.02610	-2	0.06000

TABLE XVIII
CONTINUED

Signal Parameters	Manufacturer & Breaker Type					
	S&C FVR 1121125A		S&C SD		WH AA10	
	OPEN	CLOSE	OPEN	CLOSE	OPEN	CLOSE
T2	0.00083	0.00204	0.00042	0.00219	0.00087	0.03281
T3	0.01847	0.01016	0.02358	0.01194	0.01040	0.03400
T4	0.03024	0.05278	0.02927	0.06753	0.04201	0.25590
T5	0.03163	0.04714	0.02517	0.05700	0.06476	0.19271
T6	0.02903	0.05556	0.02188	0.07100	0.02950	0.24028
T7	0.03200	0.05000	0.02239	0.06582	0.02934	0.24618
T8	-2	-2	-2	-2	-2	0.0
T9	-2	-2	-2	-2	-2	0.25500
T10	-2	0.05800	-2	0.07000	-2	0.24510

Signal Parameters	Manufacturer & Breaker Type					
	SD VACARC		SD FLUARC		SD FLUARC-FVBS1121120	
	OPEN	CLOSE	OPEN	CLOSE	OPEN	CLOSE
T2	0.0	0.0	0.0	-2	0.0	-2
T3	0.01170	0.02558	0.01060	-2	0.01000	-2
T4	0.02347	0.04705	0.04032	-2	0.04000	-2
T5	0.03475	0.03782	0.05530	0.06927	0.04500	0.08325
T6	0.02350	0.05304	0.02840	0.08737	0.03500	0.13000
T7	0.02825	0.04696	0.03471	0.08300	0.03000	0.14000
T8	-2	0.00000	-2	-2	-2	-2
T9	-2	0.06600	-2	-2	-2	-2
T10	-2	0.01100	-2	-2	-2	-2

Signal Parameters	Manufacturer & Breaker Type					
	WH R2		WH R3		WH R4	
	OPEN	CLOSE	OPEN	CLOSE	OPEN	CLOSE
T2	0.00052	0.02937	0.00052	0.03120	0.00061	0.02972
T3	0.01229	0.03979	-2	-2	-2	-2
T4	0.03757	0.07389	0.03784	0.08300	0.02457	0.06529
T5	0.04000	0.06200	0.05530	0.06589	0.03064	0.06156
T6	0.03300	0.07500	0.02840	0.08324	0.02798	0.06994
T7	0.02912	0.07149	0.03100	0.08500	0.02470	0.07000
T8	-2	0.00000	-2	0.00000	-2	0.0
T9	-2	0.09500	-2	0.06800	-2	0.07020
T10	-2	0.07500	-2	0.08170	-2	0.07150

TABLE XVIII
CONTINUED

Signal Parameters	Manufacturer & Breaker Type					
	HITACHI MFPTB-300-63LA		POWELL VAC 15PV750-2		SA SDO-15-500	
	OPEN	CLOSE	OPEN	OPEN	CLOSE	OPEN
T2	0.00052	0.02528	0.00145	0.00052	0.02528	0.00145
T3	0.00600	0.02689	0.01591	0.00600	0.02689	0.01591
T4	0.02250	0.07046	0.04242	0.02250	0.07046	0.04242
T5	0.02598	0.09110	0.03362	0.02598	0.09110	0.03362
T6	0.02337	0.10425	0.03021	0.02337	0.10425	0.03021
T7	0.01500	0.11287	0.02655	0.01500	0.11287	0.02655
T8	-2	-2	-2	-2	-2	-2
T9	-2	-2	-2	-2	-2	-2
T10	-2	0.06966	-2	-2	0.06966	-2

Results Evaluation

During the first stage of testing and evaluation, there is bias in selecting abnormal cases as test cases, the algorithms are expected to produce stable results no matter what abnormalities the signal might have.

During the second stage of testing and evaluation, test cases are randomly collected. Usually, several test cases of one CB type are selected to build the breaker settings. These cases are from different CBs, and their operation condition may be slightly different. Test cases may be collected at different times and locations, and environmental influence on the test cases, for example, the instrumentation noise, may be different.

Fig. 47 puts the coil currents of 18 test cases in parallel for comparing. Even though all the test cases belong to the same breaker type and come from the same manufacturer, the difference in the waveforms can still be clearly observed. The fourth case in the left column shows no obvious waveform at all. This might be attributed to a bad connection between the DFR data recorder and the CB’s control circuit. The eighth and ninth cases in the left column have been both affected by strong noise. However, the noise takes different forms. For the eighth case, the noise is heavily added to the part when the coil current has already de-energized. For the case below it, the noise is evenly added to the whole signal channel.

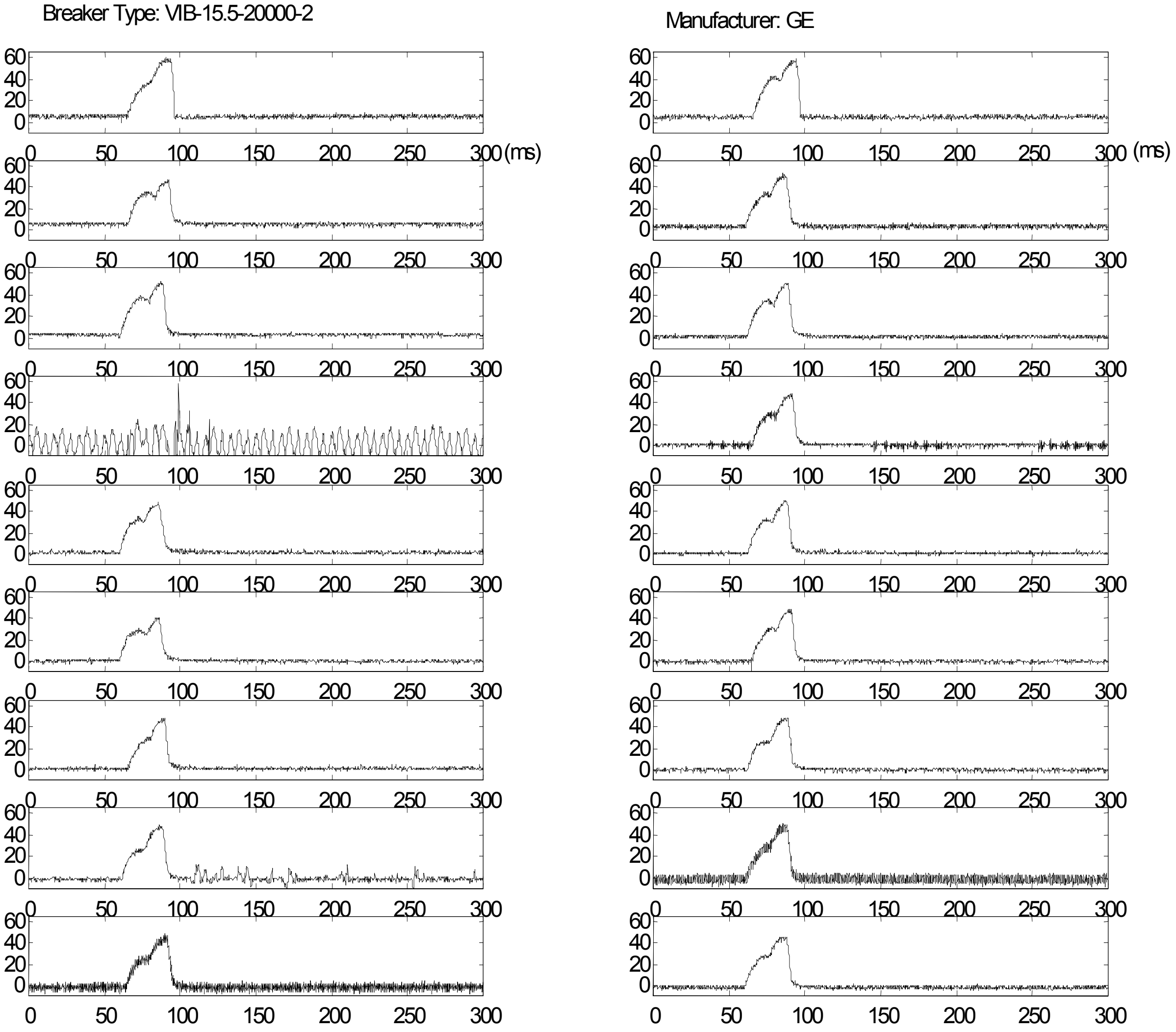


Fig. 47. Coil current of GE VIB-15.5-20000-2 circuit breaker

Algorithm is expected to provide correct results for all the normal cases, that is, all the cases except the fourth case in the example (results for the dip time are provided in Table XIX). The result for the abnormal case could be meaningless, but that is not the problem with the pre-processing algorithm, nor will the pre-processing algorithm be responsible for detecting the abnormal case. Instead, expert system will be responsible for this task. Expert system is able to make a decision by combining all the information provided by different signal parameters. For the fourth case, the expert system will first find that both signal parameters T2 (Event #2 Coil current picks up) and T4 (Event #4 Coil current drops off) equal to –1, which implies that coil current did not rise at all. Then expert system will summarize the condition disregarding the result of the signal parameter T3 (Event #3 Coil current dip time).

TABLE XIX
TEST RESULTS FOR GE VIB-15.5-20000-2 CIRCUIT BREAKER

Case No.	DIP Time (ms)	Case No.	DIP Time (ms)
1	80.5556	10	77.7778
2	83.3333	11	77.7778
3	83.3333	12	80.5556
4	77.7778	13	80.5556
5	80.5556	14	77.7778
6	80.5556	15	77.7778
7	69.4444	16	77.7778
8	80.5556	17	80.5556
9	77.7778	18	77.7778

The results of pre-processing can be used to generate settings for the expert system. For the test cases provided in Fig. 47, the average dip time is 79.58ms. The tolerance can be decided by getting the maximum deviation of the dip times from the setting 79.58, which is 3.76 ms in our example.

The performance consistency shall also hold for different CB types. The example given in Chapter V demonstrates the performance consistency of the "ExtractDIP" algorithm for three different CB types.

Conclusion

During the first stage testing, the selection for test cases is in favor of abnormal cases, that is, the cases that have certain signal features and can fire certain expert system rules. Testing results show that the analysis software can produce expected results given that features and signal parameters are defined.

During the second stage testing, test cases are randomly collected. Features are unknown before the testing, and any undefined features may also appear in the test cases. For example, excessive noise, garbage waveforms or even missing signal channels due to an incorrect connection of test device. The analysis system is able to ignore these problems unrelated with the CB operation and provides stable results. The testing also covers a great number of different CB types.

It is also determined that the analysis system can only detect predefined features and fire existing expert system rules. If there are other features in the data, then the system will simply not recognize them. Extra development effort would be required for the system to be able to detect and classify new features. More effort on data collection and testing is necessary for CBs that do not have enough test cases as well as for new types of CB.

CHAPTER VIII

CONCLUSION

Summary of Work

The work of this thesis is to solve a real world problem facing utility companies everyday, i.e., how to analyze the condition of CB effectively. An automated analysis solution is provided in this thesis. The solution is designed specifically for analyzing signals from the CB control circuit. Details of the work have been covered in Chapter 4, 5, 6 and 7 respectively. The analysis software generates stable and consistent results. The following work is accomplished:

- Features are identified in the signals, and signal parameters are defined to quantifies the features;
- Feature extraction algorithms are designed and implemented by using Wavelets and other signal processing techniques;
- Overall pre-processing system is build up in an objected-oriented architecture;
- Test cases of different breaker types, and different abnormalities are used to test the viability of the system.

Contribution

Automated CB condition analysis is a new development in the CB maintenance area. It extends CB condition monitoring by using advanced data processing and analyzing, which will further help to build a maintenance decision-support solution. This work improves CB operation analysis by:

- Automating the analysis process,
- Accelerating the analysis speed,
- Providing consistent analysis calibrations,
- Allowing compatibility with different CB types,
- Enabling user-friendly access to the CB operation.

Future Work

Future work on introducing the probabilistic model into the system is suggested. With more time and effort this approach could possible eliminate the need for collecting empirical data, and the system could be more robust in design.

REFERENCE

[1] *IEEE Standard Definitions For Power Switchgear*, ANSI/IEEE C37.100b-1986, 1981.

[2] R. D. Garzon, *High Voltage Circuit Breakers: Design and Applications*, New York: Marcel Dekker, 1997, pp. 214-222.

[3] C. Nail, "Automated Circuit Breaker Analysis", M.S. thesis, Dept. Electrical Engineering, Texas A&M University, College Station, August 2002, pp. 10-18.

[4] *IEEE Guide for Diagnostics and Failure Investigation of Power Circuit Breakers*, IEEE Std C37.10-1995, 23 Sept. 1996.

[5] M. Kezunovic, C. Nail, Z. Ren, D. R. Sevcik, S. Lucey, W.E. Cook, E.A. Koch, "Automated Circuit Breaker Monitoring and Analysis", IEEE PES summer meeting, Chicago, July 2002.

[6] *TR-100 Digital Fault Recorder*, Specifications by Rochester Scientific Columbus, 2002.

[7] J. Endrenyi, S. Aboresheid, R. N. Allan, G. J. Anders, S. Asgarpoor, R. Billinton, N. Chowdhury, E. N. Dialynas, M. Fipper, R. H. Fletcher, C. Grigg, J. McCalley, S. Meliopous, T. C. Mielnik, P. Nitu, N. Rau, N. D. Reppen, L. Salvaderi, A. Schneider, and Ch. Singh, "The Present Status of Maintenance Strategies and the Impact of Maintenance on Reliability", IEEE Transactions on Power Systems, vol. 16, No. 4, 638 – 646, November, 2001

[8] *RTR-84 Circuit Breaker Response Recorder*, Hathaway Systems Corporation, Belfast, Ireland, Hathaway, 1999.

[9] J. Cosgrave, S. Strangherlin, J. Humphries, G.R. Jones, C. Piazza, S. Volponi, L. Sfondrini, "Intelligent Optical Fiber Monitoring of Oil-Filled Circuit Breakers," *IEE Proceedings of Generation, Transmission and Distribution,* vol. 146, no. 6, pp. 557 –562, Nov. 1999.

[10] H. K. Høidalen, M. Runde, O. Haugland, G. Ottesen, and M. Öhlén, "Continuous Monitoring of Circuit Breakers Using Vibration Analysis," *in Proc. 1999 Eleventh International Symposium on High Voltage Engineering*. pp. 102 –106.

[11] L.T.Rumfield, “Circuit Breaker Diagnostic Response Testing Using the RTR-84 Response Time Recorder”, presented at the Sixty-First Annual International Conference of Doble Clients, June 1994.

[12] P. L. Fletcher, W. Degen, “A Summary of The Final Results and Conclusions of The Second International Enquiry on The Reliability of High Voltage Circuit-Breakers”, *Second International Conference on the Reliability of Transmission and Distribution Equipment*, 1995, pp. 24 –30.

[13] O. Rioul, M. Vetterli, *Wavelets and Signal Processing*, IEEE SP MAGAZINE, pp. 14-38, Oct. 1991.

[14] J. C. Goswami, A. K. Chan, *Fundamentals of Wavelets: Theory, Algorithms, and Applications*, New York: Wiley-Interscience, 1999, p. 13.

[15] M. Misiti, Y. Misiti, G. Oppenheim, J. Poggi, *Wavelet Toolbox For Use with MATLAB*, The MathWorks, 1996.

[16] J. C. Goswami, A. K. Chan, *Fundamentals of Wavelets: Theory, Algorithms, and Applications*, New York: Wiley-Interscience, 1999, p. 57.

[17] M. B. Ruskai, *Wavelets and Their Applications*, Boston, London: Jones and Bartlett Publishers, 1992, p. 18.

[18] G. K. Nelson, “Circuit Breaker Response Time Testing”, Presented to EEI – Electrical Equipment and Systems Committee, Phoenix, Arizona, February 17-20, 1991.

[19] *Microsoft mastering. MFC development using Microsoft Visual C++ 6.0/Microsoft Corporation*, Redmond, WA, Microsoft Press, 2000.

[20] R. D. Lee, W. M. Tepfenhart, *UML and C++: A Practical Guide To Object-Oriented Development*, Upper Saddle River, N.J., Prentice Hall, 2001.

[21] *IEEE Standard Common Format for Transient Data Exchange (COMTRADE) for Power Systems*, IEEE Std. C37.111-1999.

APPENDIX

SOURCE CODE

```
function [t] = risetime(sig, startIndex, endIndex, set_1, set_2)

% Function: Detect the Signal Status Change from Low to High

% sig: Original signal
% startIndex
% endIndex
% set_1, Amplitude Threshold
% set_2, Stable Count

sig2 = diff(sig); sig2 = [sig2;sig2(end)];
detection = 0;
count = 0;
failflag = 1;
riseIndex = -1;

if startIndex == -1
   t = -1;
   return
end

if endIndex == 0
   endIndex = length(sig);
end

for i = startIndex:endIndex-1
   if (sig(i)>set_1 & sig2(i)>=0)
      if detection == 0
         riseIndex = i;
         detection = 1;
      end
      count = count + 1;
   end
   if (detection == 1 & sig(i)>set_1 & count >= set_2)
      failflag = 0;
      break
   end
   if (detection ==1 & sig(i)<set_1 & count <= set_2)
      detection = 0;
      count = 0;
   end
end

if failflag ~= 0
   riseIndex  = -1;
end

t = riseIndex;
```

```
function [TIM3]=ExtractTIM3(sig, Type, Operation)

% Function: Detect the dip time T3 of the coil current

% sig: original signal
% Type: Breaker type
% Operation: Close or open operation


% Default Approximation level, Detail level, sampling frequency
app_level = 6;
det_level = 4;
sf = 5760; % sampling frequency


% Select approximation and detail levels for the specific breaker type
[app_level, det_level, stair] = select(Type, Operation);


% Decompose
[C,L] = wavedec(sig,app_level,'db1');


% Get Approximation
A = appcoef(C,L,'db1',app_level);


% Get Detail
D = detcoef(C,L,det_level);


% Sort Approximation Sequence
[Y,I]=sort(A);


% Obtain index zone in which the coil dip is located
IndexZone = I(end - stair + 1);


% Convert index zone to detail level
Y = [IndexZone-1, IndexZone].* 2^(app_level-det_level);


% Locate the Dip point & convert it from index to time
X = abs(D(Y(1)+1:Y(2)-1));
[Y2,I2] = sort(X);% maximum value
T3 = (I2(end)+ Y(1)-1)* 2^det_level/sf;


% Return the output
TIM3 = T3;
```

```
function [RIP]=ExtractRIP(sig, set_1)

% Function: Extract the ripple feature of the dc voltage

% sig: Original signal
% set_1: Power Spectral Density settings

% Decompose
[C,L]= wavedec(x,5,'db1');

% Get Detail
det_level = 3;
D = detcoef(C,L,det_level);

% Get PSD parameters
sf = 5760; %sampling frequency
ORDER = fix(log2(length(sig)))+1;NFFT = 2^ORDER;Fs=fix(sf/det_level);

% Get PSD via burg’s method
[Pxx,F] = pburg(D,ORDER,NFFT,Fs);


% Obtain PSD at 60, 120, 240 Hz
for j = 1:length(F)
   if (F(j)>=60)
      P60 = Pxx(j);
      break;
   end
end
jj = j;
for j = jj:length(F)
   if (F(j)>=120)
      P120 = Pxx(j);
      break;
   end
end
jj = j;
for j = jj:length(F)
   if (F(j)>=240)
      P240 = Pxx(j);
      break;
   end
end

% Calculate the RIP amplitude
if P60 > set_1 | P60 > set_1 | P60 > set_1
   sig2 = sig(1:Fs/sf);
   RIP = (max(sig2)-min(sig2))/mean(sig2);
else
   RIP = 0;
end
```

VITA

Zhifang Ren
No. 19 Yue-Tan-Nan Street, Apt. 906 of No.3 Bldg.
Xichen District, Beijing, 100045, China

Educational Background

Zhifang Ren received her Bachelor of Science Degree in Electrical Engineering from Tianjin University in 1998. She earned her Master of Science Degree in Electrical Engineering from Texas A&M University in May 2003.

Publications

[1] M. Kezunovic, C. Nail, Z. Ren, D.R. Sevcik, J. Lucey, W. Cook, E. Koch, "Automated Circuit Braker Monitoring and Analysis," IEEE PES Summer Meeting, Chicago, July 2003.

[2] M. Kezunovic, Zhifang Ren, Goran Latisko, D.R. Sevcik, J. Lucey, W. Cook, E. Koch, "An Expert System for Automated Analysis of Circuit Breaker Operation," 12th ISAP (The 'Intelligent Systems Application to Power Systems') Conference, Greece, June 2003.

[3] M. Kezunovic, Zhifang Ren, Goran Latisko, D.R. Sevcik, J. Lucey, W. Cook, E. Koch, "Automated Analysis of Circuit Breaker Operations," EPRI Substation Equipment Diagnostics Conference, New Orleans, February 2003.