# POWER DISTRIBUTION SYSTEM EVENT CLASSIFICATION USING FUZZY LOGIC

A Dissertation

by

KARTHICK MUTHU MANIVANNAN

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

DOCTOR OF PHILOSOPHY

Approved by:

| | |
|---|---|
| Chair of Committee, | B. Don Russell |
| Committee Members, | Karen L. Butler-Purry |
| | Reza Langari |
| | Krishna Narayanan |
| Department Head, | Costas N. Georghiades |

December 2012

Major Subject: Electrical Engineering

# ABSTRACT

This dissertation describes an on-line, non-intrusive, classification system for identifying and reporting normal and abnormal power system events occurring on a distribution feeder based on their underlying cause, using signals acquired at the distribution substation. The event classification system extracts features from acquired signals using signal processing and shape analysis techniques. It then analyzes features and classifies events based on their cause using a fuzzy logic expert system based classifier. The classification system also extracts and reports parameters to assist utilities in locating faulty components. A detailed illustration of the classifier design process is presented.

Power distribution system event classification problem is shown to be a large scale classification problem. The reasoning behind the choice of a fuzzy logic based hierarchical expert system classifier to solve this problem is explained in detail. The fuzzy logic based expert system classifier uses generic features, shape based features and event specific features extracted from acquired signals. The design of feature extractors for each of these feature categories is explained. A new, fuzzy logic based, modified Dynamic Time Warping (DTW) algorithm was developed for extracting shape based features. Design of event specific feature extractors for capacitor problems, arcing and overcurrent events are discussed in detail. The fuzzy logic based hierarchical expert system classifier required a new fuzzy inference engine that could efficiently handle a large number of rules and rule chaining. A new fuzzy inference engine was designed for this purpose and the design process is explained in detail. To avoid information overload, an intelligent reporting framework that processes raw classification information generated by the fuzzy classifier and reports events of interest in a timely and user friendly manner was developed.

Finally, performance studies were carried out to validate the performance of the designed fuzzy logic based expert system classifier and the intelligent reporting system. The data needed to design and validate the classification system were obtained through the Distribution Fault Anticipation (DFA) data collection platform developed by Power System Automation Laboratory (PSAL) at Texas A&M University, sponsored by the Electric Power Research Institute (EPRI) and multiple partner utilities.

# DEDICATION

To my family and friends, for their love and encouragement.

# ACKNOWLEDGEMENTS

I am thankful to Dr. B. Don Russell for his patience and guidance throughout my research work. I am grateful to him for having provided me with an opportunity to work on this project, and for helping me refine this dissertation.

I wish to thank Carl L. Benner whose inputs and ideas were vital for the completion of this work. I thank my committee members, Dr. Karen L. Butler-Purry, Dr. Reza Langari and Dr. Krishna Narayanan, for their advice and time. I wish to thank Jeff Wischkaemper for his help in reviewing my dissertation.

I wish to acknowledge the Electric Power Research Institute (EPRI) and the supporting utilities for their funding of this project.

I thank my family and friends for their loving support and words of encouragement.

# TABLE OF CONTENTS

# LIST OF FIGURES

## LIST OF TABLES

# 1. INTRODUCTION

## 1.1 Introduction

The primary aim of any electricity supply system is to meet all customers' demand for energy. With increasing dependence on electricity supply, the need to achieve the highest possible reliability as economically as possible has becomes more important to customers. The large geographic areas spanned by power distribution feeders make them susceptible to factors such as: weather, disturbances caused by animals and human activity. The components of a power system include, but are not limited to, overhead lines, cables, transformers, insulators, capacitors, protective devices and the loads utilizing the power supplied. These components are susceptible to overloading, aging or defects, and damage.

Components of a power system that are either in the process of failing or that have already failed cause abnormal conditions measurable as electrical transients on the power system [1]. In some cases, a failing component may trigger the operation of a protective device, isolating that component from the system, but also creating an interruption or sustained outage. In other cases, a faulty component will continue to misoperate causing disturbances and hazardous conditions on the power system. In the former case, when a protective device operates, portions of the feeder may lose their supply of power. Such outage situations are undesirable and utilities make efforts to locate the problem and restore the supply to those parts in the shortest possible time. In the later case, a failed or misoperating component, when not isolated, may cause transients, unbalanced conditions or other power quality problems. Such disturbances may interfere with the normal operation of other components in the power system. In such situations, it is desirable to detect the faulty component and rectify or replace it before it causes any further damage to the system.

Utility companies do their utmost to keep customers served by providing them with uninterrupted supply of high quality power at competitive prices through careful planning and operation of distribution systems. Utility companies do scheduled maintenance and inspection to monitor the health of feeders. Nevertheless, failure or degradation of power system components are inevitable. Components may degrade or fail between maintenance cycles. Faulty components may have been overlooked during maintenance or healthy components may fail as a result of an operator error during a maintenance cycle. Consequently, certain situations arise and result in abnormal operating conditions. When such abnormal conditions arise, it is important to identify the cause of the problem and take measures to fix the problematic component in a timely and cost effective manner. Today, electric utilities have no automated diagnostic systems to assist in identifying, finding, and correcting degraded or failed components.

Developing a system that monitors health of feeders, detects abnormal conditions and provides information about misoperating components would provide significant benefit to the utility industry. Such system would require advanced continuous on-line monitoring to keep a utility company updated about the health of its feeders. This monitoring system, in addition to providing information about the cause of abnormal conditions, should also provide relevant parameters that will help narrow the search for the problem. Some of the advantages of such a system include:

- Utilities can resort to condition based maintenance and fix only components that have failed or started to degrade. This will allow the utility industry to save hundreds of millions of dollars in maintenance costs.

- Detecting failing components will help utility companies to fix problems before they escalate to catastrophic failures. This approach would have significant benefit to system reliability, as well as allowing utilities to fix components under optimal conditions (e.g. not in the middle of the night after an explosion, causing a large outage).

- Some components may go to failure rapidly or may not exhibit any detectable signatures during the period leading to failure. In such situations, any information about the failed component may still reduce the amount of time needed to locate the faulty component and replace it. In the absence of this information, utility companies have to manually inspect feeders based on customer calls. This is often sub-optimal, and can be expensive, time consuming, and error prone.

It is possible to monitor activity on a feeder by analyzing signals such as current and voltage waveforms corresponding to that feeder. These signals can be acquired using a data acquisition platform located at the distribution substation. Both normal and abnormal operations of power system components exhibit characteristic signatures that may be detected by analyzing the acquired signals. This analysis is quite complex and involves analyzing large volumes of data. Such analysis is expensive, time-intensive, and error prone if done manually. Developing an automated system for monitoring feeder health, that automatically reports problems in a timely and user friendly manner, is necessary. Other than the methodology presented in this dissertation, currently no other such system exists.

## 1.2 Research Objectives

This dissertation describes a new, on-line, non-intrusive, classification system for identifying and reporting normal and abnormal power system events occurring on a distribution feeder based on their underlying cause, using signals acquired at the distribution substation. In this research context, an event is any normal or abnormal observable activity on the power distribution feeder that manifests itself through measurable changes in electrical signals. Such an event classification system needs to extract features from acquired signals using signal processing and shape analysis techniques. It then must analyze features and classify events based on their cause. Ideally, the classification system should also extract and report parameters to assist utilities in locating the faulty component. The following paragraphs present key steps needed to accomplish the above goal:

***Developing a fuzzy logic based expert system:*** The primary objective of this research is the development of a fuzzy logic based expert system for classifying power system event data. A scalable and modular fuzzy hierarchical classifier is proposed. A fuzzy inference engine that supports a hierarchical structure best suited for power system event classification is needed to accomplish above objective.

***Identifying characteristic event signatures:*** Characteristic features corresponding to power system components that are either in the process of failing or have already failed must be identified. These features are needed by the fuzzy logic based expert system classifier. Some failure signatures are well understood, while others must be documented for the first time. Data should be acquired, where possible, from operational distribution feeders where events ave occured under normal operating conditions. Specialized algorithms for extracting event-specific features are needed.

***Developing RMS shape analysis algorithm:*** Manually classifying power system events primarily involves visual analysis of root mean square (RMS) current, voltage and power waveforms by a human expert. To mimic this analysis, shape recognition and segmentation algorithms are needed. A Dynamic Time Warping (DTW) based RMS shape analysis algorithm is proposed for this purpose.

***Identifying and computing important parameters:*** Once the underlying cause of a power system event is identified, appropriate parameters corresponding to the event of interest should be determined and calculated. Different power system events require different parameters to be calculated in order to estimate the physical location of the event. For example, in the case of an unbalanced capacitor operation, VAR step size and the phase involved can help determine which of several capacitor banks on a feeder are failing. In the case of a repetitive overcurrent fault, the magnitude and duration of the fault, what phases are involved, and information on any protective devices that may have operated can assist utility personnel in significantly reducing their search area. Algorithms to extract the parameters corresponding to different power system events are needed for identifying and reporting event-specific parameters.

***Intelligent reporting to avoid information overload:*** For the classification system to be usable, it cannot report every event that is classified. This is because, a majority of monitored power system transients are normal day-to-day system operation that are of little interest to utility personnel. If all these events are reported, utility personnel will be overloaded with data. They may miss the relatively few and important abnormal events (i.e. important information) that get buried among a larger number of normal events. It is important that only events of interest are presented to utility personnel along with parameters required to locate the event source. Hence, an intelligent reporting framework to process raw classification information generated by the fuzzy classifier and report events of interest in a timely and user friendly manner is also needed.

***Design of efficient classification algorithms:*** In order to report problems on distribution feeder to utility personnel in a timely fashion, all components of the expert system classifier needs to be very efficient and operate in soft real time. This design constraint is complicated by the reality that processing power available in remote monitoring units is generally quite limited. As a result, a scheme in which the fuzzy logic rule base is precompiled will be preferable, as it will reduce the time and memory complexity of all sub-components used by the classification system.

## 1.3 Dissertation Outline

This dissertation is organized as follows. Chapter 1 presents overall industry problem, the objectives of this research work, and organization of this dissertation. Chapter 2 provides a literature review of methodologies used to classify power distribution system events. Chapter 3 describes the problem of classifying power distribution system events in detail and outlines an approach used to solve the problem. Chapter 4 details shape analysis methods used for RMS shape identification. Chapter 5 identifies signatures needed to classify specific power system events and describes algorithms used to extract these signatures. Chapter 6 describes the fuzzy hierarchical classifier architecture and outlines an intelligent reporting framework. Chapter 7 describes an Intelligent Power System Event Reporting System (IPSERS) implemented based on the proposed methodology and analyzes the classification accuracy of IPSERS. Chapter 7 also provides real world examples where the proposed methodology has assisted utility personnel in locating sources of abnormal events. Chapter 8 draws conclusions and discusses possible future directions and improvements to this research.

# 2. LITERATURE REVIEW

## 2.1 Introduction

Considerable research has been done in the application of classification methods to identifying problems on power distribution systems. Most of the research falls under two categories: classifying Power Quality (PQ) disturbances and detecting specific power system events. While prior research has either focused on detecting a specific event type or on classifying events based on their nature without trying to identify their cause, this research focuses on classifying a broad range of events based on their underlying cause.

Little prior work exists in the area of power distribution system event classification as described in this dissertation. This is largely due to diverse nature of power distribution systems. The development of generalized classification algorithms that work equally well on feeders operating under different conditions, across multiple utilities, serving customers with varying needs is difficult. This research attempts to fill that gap by developing a generalized fuzzy logic based expert system architecture that is targeted towards power system event classification problem. One significant reason for the lack of prior art is the absence of field data that document the wide range of event signatures occurring on operational systems with the fidelity needed to design reliable classification routines. This research uses data collected using advanced monitoring units developed as a part of the Distribution Fault Anticipation (DFA) project sponsored by the Electrical Power Research Institute (EPRI). DFA monitoring units have been installed on sixty feeders belonging to eleven utilities across the North America and have been gathering data for the past seven years. Data collected using these devices represents the largest known library of events and failure signatures on distribution feeders. Many of the events and failures documented in the DFA database had not been documented before the DFA project began. Understanding gained from analysis of this library of events has served as the basis for developing the fuzzy logic based classification methodology put forth in this dissertation. In spite of significant differences between the proposed research and currently available techniques, it is worth reviewing PQ disturbance classification and specific event identification methodologies. These methods provide valuable insight into existing power system signal analysis techniques and classification approaches.

Power quality may be defined as delivering electrical energy with characteristics required for the proper operation of various components and loads that make up the power system without significantly affecting their performance or reliability. A PQ disturbance may be described as deviation of voltage signals from their ideal values [2]. PQ disturbances include transients, long duration voltage variations (under-voltage, over-voltage, sustained interruptions), short duration voltage variations (swells, sags, oscillations), voltage imbalance, harmonic distortions and power frequency variations [3]. PQ disturbance classification methods focus on identifying the nature of disturbances and seldom try to identify the component that caused the disturbance. Another drawback of PQ disturbance classification methods is that they mostly restrict themselves to analysis of voltage signals. These methods overlook critical information contained in current signals. For example, an overcurrent fault will cause a dip in voltage signals irrespective of its position relative to the measurement point. Based on observing voltage signals alone, it is not possible to determine if a fault was caused by a problem upstream or downstream of a measurement point. Similar arguments can be made for other event types too. Knowing the position of a problematic component relative to the measurement point is essential to narrow down the location of the problem. To accomplish this, voltage signals by themselves are not sufficient, and current signals, along with values calculated from a combination of current and voltage, such as real and reactive power, need to be analyzed. Methods for specific event identification may do a better job of identifying event cause and position, but have their drawbacks.

Specific event identification methods, as the name suggests, are designed to detect and/or locate specific problems. Some examples of specific event identification include overcurrent fault location, high-impedance fault detection, transformer fault detection, inrush detection, capacitor identification. The major drawback of these methods is that they do not provide a general framework for identifying a broad range of events. It is not practical to combine these different methods to form a hybrid classification system since these specific event identification methods often do not share a common input interface. Some specific event identification methods require that measurements be made at a close proximity or even inside the equipment of interest. Such methods can be very expensive and cumbersome if they have to be installed at a utility-wide level. For wide-scale deployment, a classification method to determine the health of components and equipment connected to a feeder based solely on measurements taken directly at a distribution substation. The primary objectives of this research is to develop one such method.

The following sections review and summarize existing approaches to power system event classification with focus on PQ disturbance classification and specific event identification.

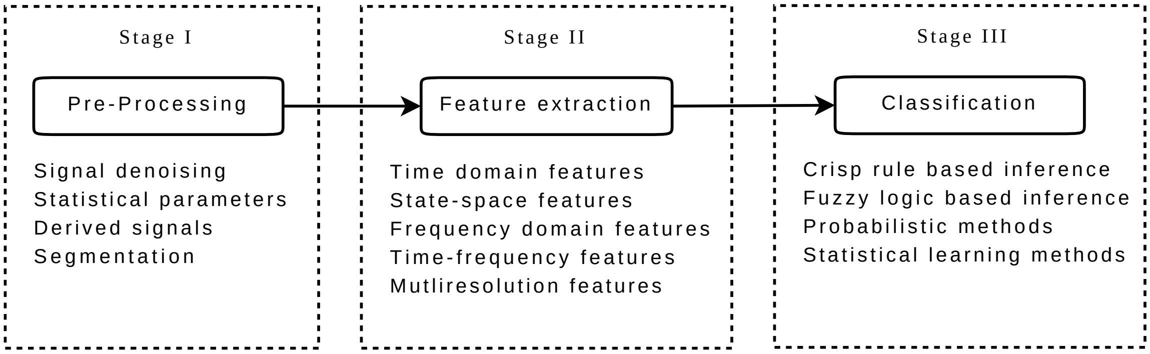


Figure 2.1: General Structure of Disturbance Classifiers

## 2.2 Literature Review of PQ Disturbance Classification Methods

Power quality disturbance classification methods have a general structure as outlined in Figure 2.1. In the figure, Stage I represents the pre-processing stage. During pre-processing, operations like signal denoising, computing statistical parameters and computing derived signals may be performed. Some approaches use segmentation during pre-processing to isolate data corresponding to a disturbance and to extract pre-disturbance and post-disturbance parameters. Stage II represents the feature extraction stage. Feature extraction is essential to concisely and accurately present sufficient data to identify a disturbance without overwhelming the classifier. The quality of the features to a large extent determine accuracy of classification. Choice of features and feature extraction techniques are influenced by the nature of the signal being analyzed, domain knowledge and the processing power available. Features serve as inputs to Stage III, the classification stage. This stage is responsible for analyzing features, discriminating different disturbance types and assigning a class label that associates the data with a specific disturbance type. The choice of classifier is heavily influenced by the nature of features used.

Several PQ disturbance classification methods have been proposed. These methods vary in their choice of feature extraction techniques and classification techniques but conform to the general structure outlined above. Some of these methods use real field data but most of them use simulated data as inputs. Each of these methods have their own advantages and disadvantages, however there is no empirical data or comparisons available in the literature that will help to prove superiority of one method over the other. The following sections will discuss feature extraction techniques and classifiers that have been used for PQ disturbance classification.

### *2.2.1 Feature Extraction Techniques*

A variety of feature extraction techniques have been applied to PQ disturbance analysis. These techniques can be broadly classified into time domain analysis, state-space analysis, frequency domain analysis, time-frequency analysis, and mutliresolution analysis.

#### *2.2.1.1 Time domain analysis*

Digital filters may be used extract information contained in certain frequency bands by passing a time domain signal through the digital filter. More than one such filter may be used to extract features corresponding to different frequency bands. Chen [4] proposed a method that used a series of digital filters to extract features from voltage signals. Digital filters were used to obtain signals corresponding to specific frequency bands. Then the deviation of the energy content in these different frequency bands from their nominal values was used to determine the type of PQ disturbance. Lu [5] proposed a different time domain approach based on mathematical morphology [6]that uses shapes as probes to extract features from time domain signals. Lu claims that good performance under noisy conditions and computational efficiency as the main advantages of this approach. His analysis, however, is limited to voltage signals and does not provide an automatic classification strategy.

#### *2.2.1.2 State-space analysis*

A widely used state-space approach in power quality is Kalman Filter. Kalman filter is a recursive estimator that uses state estimates from a previous time step and the measurements from the current time step to estimate the current state values. Typically for power system signals, amplitude and phase of fundamental frequency, and the amplitude and phase for a set of harmonics serve as the states. The estimated state values are compared with actual observed values and this information is used for disturbance analysis. Styvaktakis [7] used Kalman filter for disturbance detection, Styvaktakis also computed a disturbance index based on the deviation of estimated values and actual voltage measurements. These were used as inputs to an expert system based disturbance classifier. Since Kalman filter assumes that the system being modeled is a linear dynamic system and power system disturbances and transients are highly non-linear, Zhang [8] proposed an approach based on unscented Kalman filter which is better suited for non-linear systems. Zhang only outlined a method to detect power quality problems but did not provide a means of classifying power quality disturbances.

#### *2.2.1.3 Frequency domain analysis*

Representing power system signals in the frequency domain by applying a Discrete Fourier Transform (DFT) is a well established tool for feature extraction. Application of the DFT provides a means of estimating the amplitude and phase angles of the fundamental component and its harmonics. Further, values corresponding to a subset of these harmonics may be chosen to achieve feature reduction. The choice of the harmonics may be either based on expert knowledge, or general dimensionality reduction techniques such as Principal Component Analysis (PCA) or Independent Component Analysis (ICA) may be applied [9]. Amplitude and phase estimation may be improved by applying windowing such as Hanning, Hamming, etc. The main disadvantages of DFT-based approaches is the inverse relationship between the frequency resolution and window size. A single window size may not be suited for analyzing both disturbances that manifest as short duration transients that last a fraction of a cycle and those that manifest as slow changes lasting several seconds.

Typically, Fast Fourier Transform (FFT) which is a computationally efficient implementation of DFT is used. DFT-based features have been used with classifiers such as neural networks, Bayesian networks and rule based systems for PQ disturbance classification. Chai [10] proposed a method that used amplitude estimates of 20 spectral components obtained using FFTs. These features served as inputs to a Bayes linear classifier. Chai limited the analysis to voltage signals. Dash [11] outlined an approach that used Fourier coefficients obtained from voltage signals and a linear combiner based on Adaline neuron [12] to extract features. These features were then processed by a fuzzy rule based system to classify disturbances. Liao [13] described a method for voltage disturbance analysis that used DFT based features and wavelet coefficients as inputs to a fuzzy rule based expert system.

#### *2.2.1.4 Time-frequency analysis*

Time-frequency analysis have gained popularity in recent years because they offer time localization of different frequency components present in the signal being analyzed. With DFTs no time domain information is provided. This is especially useful when the component frequencies of the signal being analyzed change over time and for transient detection. Time-frequency analysis techniques such as Short Term Fourier Transforms (STFT), Continuous Wavelet Transforms (CWT), Gabor transforms and Wigner-ville distribution provide a way to compute time-frequency distributions that helps to track different frequency components over time [14]. Most of these techniques work by allowing the window over which different frequency components are computed to be changed. One main drawback of time-frequency based methods over DFTs is increased computational complexity.

A time-frequency analysis technique that has often been used for PQ disturbance analysis is the S-Transform [15]. S-Transform is an extention of CWT. Reddy [16] recommended the use of S-Transforms for PQ disturbance analysis and provided proof of their superiority over STFT and Wavelet Transforms based on visual analysis. Reddy, did not, however describe a method for automatic classification. Chilukuri [17] proposed a method based on S-Transforms. Chilukuri used features derived from time-frequency matrices obtained by applying a discrete version of a S-Transform on voltage signals. These features were analyzed by a fuzzy rule based classifier to automatically classify PQ disturbances. Shangwei [18] presented an approach that used a S-Transform for feature extraction. Shangwei used Support Vector Machine (SVM) to analyze the features and classify PQ disturbances.

#### *2.2.1.5 Multiresolution analysis*

Multiresolution analysis (MRA), as the name suggests, works by decomposing a signal into various levels of approximation or resolutions [19]. MRA allows the signal to be studied at various resolutions without losing time domain information. Rapid changes in the signal can be studied at higher resolutions while slow changes may be studied at lower resolutions. Wavelet decomposition is the primary tool used in MRA. An efficient implementation of wavelet decomposition, Discrete Wavelet Transform (DWT) is used for practical applications. DWT uses filter banks and decimation to recursively decompose signals into frequency bands. High frequency bands have higher resolution (i.e. sampling rate) while lower frequency bands have lower resolution. Features obtained from decomposed signals (also called coefficients) have been used as inputs to PQ disturbances classifiers. The filter banks used in DWT are derived based on a mother wavelet that will be used for analysis. One of the challenges in using DWT is the choice of mother wavelet best suited for the problem at hand. A danger in this approach is the possibility of selecting a mother wavelet too specific to a particular dataset.

Wavelet transforms have been used in combination with different classification techniques for PQ disturbance classification. Wavelet transforms have been found most successful in detecting transients and short duration PQ disturbances. Santoso [20] showed that MRA can be used for PQ disturbance identification. He used Daubechies wavelet for the analysis. Santoso's work laid the foundation for further research using DWT as a feature extraction technique for PQ disturbance classification. Santoso [21, 22] also developed a classification approach that used wavelets for feature extraction and neural networks [23] for classification. Santoso used only voltage waveforms collected from monitoring units and restricted the analysis to capacitor transients, impulse transients, sags and interruptions. Elmitwally [24] used wavelet based features as inputs to a neuro-fuzzy classifier [25]. He used simulated data for testing the classification scheme and restricted the analysis to voltage signal. Chung [26] used wavelet packet based decomposition, which is an extension of DWT [27] for feature extraction. Chung then used Hidden Markov Modeling (HMM) [28] to determine the likelihood of a disturbance type based on the features. Chung used, wavelet and HMM based classifier only for capacitor and impulse transients. He used RMS features for sags and interruptions. Abdel-Galil [29] also proposed a similar approach based on wavelets and HMM. Zang [30] and Kocaman [31] proposed using DWT as feature extractors for Support Vector Machine (SVM) based classifier. Intrestingly Kocaman restricted the analysis to slow voltage changes such as sags and swells.

Recently, several methods that use wavelet-based features as inputs to fuzzy rule based classifiers [32] have been proposed [13, 33, 34, 35]. All these methods analyzed voltage signals, but differed in their choice of wavelets, how the wavelet coefficients are processed to obtain features, and the rule base. In each case, the rules were framed based on expert knowledge. Abdel-Galil [36], took a different approach, using C4.5, a decision tree based inductive learning method to generate rules. His analysis still used used wavelets to extract features and limited the analysis to voltage signals.

It can be seen that a wide range of feature extraction techniques have been applied for power disturbance classification. In most of these approaches, the choice of features have been based on expert knowledge. Almost all the features have been extracted from voltage signals. Different classification techniques have been used with the above feature extraction techniques. Following section will discuss these classification techniques.

### *2.2.2 Classification Techniques*

Classification techniques that have been used for PQ disturbance analysis fall under three broad categories; logic based methods (e.g., rule based approaches), probabilistic methods (e.g., naive Bayes classifier, hidden Markov models) and statistical learning methods (neural networks, support vector machines, decision trees). These methods try to mimic the human abilities of reasoning, problem solving, learning and decision making.

#### *2.2.2.1 Logic based methods*

Logic based methods use If-Then rules to represent expert knowledge. User inputs and/or features extracted from measurements serve as inputs. An inference engine then classifies data based on rules and inputs. Such methods are also called expert systems. Kazibwe [37] proposed an expert system based on If-Then rules that diagnosed power quality problems using user inputs combined with voltage and current measurements. Kazibwe's method's reliance on human input represented a significant shortcoming. Styvaktakis [7] developed an expert system to automatically classify power system events and offer information in terms of power quality. The expert system used voltage measurements only and was capable of distinguishing different types of voltage dips, voltage steps and interruptions. Styvaktakis used Kalman filtering to detect and quantify changes in the three-phase voltages by assigning a detection index. These detection indices served as inputs to a rule based classification module that classified the event. Styvaktakis's method was limited to analyzing voltage disturbances. While this method could identify voltage disturbances caused by overcurrents, inrushes, large motor starts, and voltage steps, it was not capable of detecting arcing and capacitor related problems. Styvaktakis's method did not provide event specific information such as current magnitudes, load lost and VAR step sizes that are needed to identify the source of the disturbance.

Recently, several methods of PQ disturbance classification based on fuzzy expert systems have been proposed [25, 11, 24, 17, 33, 13, 34, 35]. Fuzzy logic based expert system is a generalization of simple If-Then rule based systems that use binary logic. In fuzzy logic, truth is not restricted to discrete true (1) and false (0) values, but allows truth values to be in the closed interval [0,1]. Such expert systems help to better model expert knowledge and human approximate reasoning. None of these methods try to classify power system events based on their underlying cause. Moreover, these methods employ a few simple fuzzy rules, and do not provide the necessary foundation required to design a large scale classifier such as the one proposed in this dissertation.

#### *2.2.2.2 Probabilistic methods*

Power system event analysis often requires reasoning under incomplete or uncertain information. Incomplete information may be caused by faulty sensors, or because some signals were not measured. Uncertainties can be introduced by noise inherent to the power system and sensors or due to the dynamics of the power system. Probabilistic methods are well suited for reasoning under uncertainty [38]. Bayesian methods and hidden Markov models are probabilistic classification methods that have been used for PQ disturbance classification. Bayesian methods use Bayes theorem to determine the probability with which an observed event belongs to a disturbance category based on extracted features. Chai [10] used Bayes theorem to generate a linear discrimination function for voltage disturbance classification with features extracted using FFTs. Wang [39] designed a classifier based on Bayes theorem and used wavelet packet decomposition for extracting input features for the classifier. Wang assumed an underlying Gaussian distribution and used only voltage signals. Unlike Bayesian networks, Bayes methods do not allow causal relationship between the features and the disturbance categories to be explicitly specified. Bayes methods rely on empirical measurements made from training data to estimate conditional probabilities. For this purpose they often assume an underlying probability distribution. Such assumptions may not always be correct or optimal.

Hidden Markov model (HMM) is another probabilistic classification method that has been used for PQ disturbance analysis. HMMs have been proved to be effective in temporal pattern classification [28]. HMMs assume the underlying process to be a Markov process with hidden states and only the outputs that are dependent on the hidden states are visible. Typically HMM based models are constructed for each PQ disturbance class. Observed PQ measurements are compared to those generated by the HMM based models. The PQ disturbances are then assigned the class of the model that most accurately matches the measurements. Abdel [29] proposed a HMM based approach for PQ disturbance classification using FFT and DWT to extract features vectors. These feature vectors represented the observable outputs. Abdel used Vector Quantization (VQ) to map the feature vectors into more manageable discrete output states, and constructed HMM-based models based on the feature vectors during the training stage. Abdel then compared new observations those generated by the models and used the model with closest match to label the PQ disturbance. Chung [26] used a similar approach based on Wavelet Packet (WP) coefficients and HMM. He used this approach only for disturbances involving high frequencies, however. Both Abdel and Chung restricted their analysis to voltage signals from simulated data and did not try to classify the disturbances based on their cause.

#### *2.2.2.3 Statistical learning methods*

Statistical learning methods can be considered as supervised learning procedures that group observations based on one or more characteristic features of the observation. To accomplish this, they use a training set that comprised of observations that have already been labeled as belonging to a group. Statistical learning methods such as neural networks, decision trees, and support vector machines have been used for PQ disturbance classification. Artificial neural networks (ANN) are simple yet powerful tools for pattern classification. Feed-forward neural networks can be used to model complex and non-linear input-output relationships. This, combined with their relative ease of implementation has made them a popular choice for pattern classification. Neural networks have typically been combined with a feature extraction technique such as wavelets for PQ disturbance identification. Santoso [21, 22] used a Learning Vector Quantization network (LVQ) as a pattern classifier for PQ disturbances. LVQs are a supervised version of self organizing maps. Santoso used wavelets for feature extraction, and these features served as inputs to the LVQ network. Reaz [34] used DWT for feature extraction with a Univariate Randomly Optimized Neural Network (URONN) and fuzzy logic for PQ disturbance classification. URONN is a variation of an adaptive neural network that uses randomized weight selection based on classification error. Reaz used URONN to tune membership parameters of a fuzzy rule based system. Elmitwally [24] used a similar method for PQ disturbance classification except that he used a backpropagation instead of randomized weight selection for training the neural network. Even though neural networks have been preferred for their ease of implementation, their black box nature makes it very hard to either encode or infer knowledge from them. Another big disadvantage of using neural networks is that they are prone to overfitting. Considerable effort and a very large training set would be required to reliably use them for large scale power system event classification. Support Vector Machines (SVM) have been used for PQ disturbance classification to overcome some of the disadvantages of neural networks.

SVMs [40] are another supervised learning technique that are good at solving classification problems that require non-linear decision boundaries. They project features into higher dimensional feature spaces and construct linear decision boundaries in this higher dimensional feature space. This approach allows them to achieve global optima as they are always searching a convex surface. Janik [41] used DWT for feature extraction and constructed a Gaussian kernel based SVM to classify simulated PQ disturbance signals. Zang [30] and Kocaman [31] also proposed similar methods based on DWT and SVM for PQ disturbance classification. While SVM based classifiers are good at handling non-linear classification problems, they have been used primarily for classification problems with small samples. Their computational complexity makes them difficult to use for large scale classification problems [42]. SVMs also have the disadvantage of being a black box approach, making it hard to decode knowledge from trained SVMs.

While ANN and SVM approaches employ a black box approach, decision trees are a straightforward and easily understood alternative where decision trees are constructed from training data. Decision trees have been used for pattern classification for their simplicity and easy to interpret nature. Abdel-Galil [36] used C4.5, a decision tree based algorithm for classification of PQ disturbance. Decision tree algorithms are better suited for categorical data and are not very good for classifying continuous data as often encountered in PQ domain. Decision trees are also prone to instability and can produce widely varying trees based on the order in which training samples are presented.

It is clear from prior literature that PQ disturbance classification systems seldom try to identify the component or components that caused the PQ disturbance. To address this issue, methods to identify a specific problematic component or specific type of disturbance based on its cause have been proposed.

## 2.3 Literature Review of Specific Event Identification Methods

In contrast to PQ disturbance classification methods, specific event identification methods focus on a specific problematic component or a specific type of disturbance. There are numerous power system components such as capacitors, Load Tap Changers (LTCs), transformers, insulators, conductors etc. Multiple methods to identify problems in these components can be found in literature. It is not possible to review each of these methods. Since this research focuses on non intrusive, substation based measurement and analysis of electrical signals, only a subset of specific event identification methods that use non-intrusive substation based electrical measurements are reviewed here. The following specific event identification methods are discussed here:

- Overcurrent (OC) identification.
- Arcing and High Impedance Fault (HIF) identification.
- Capacitor switching identification.

The above specific event identification methods together make up the majority of specific event identification methods for power distribution systems found in the literature. Each of these methods are reviewed further in the following paragraphs.

### *2.3.1 Overcurrent Identification*

Out of all specific event identification methods, overcurrent fault (bolted fault) identification has gained the most attention. One of the reasons for this is that overcurrents identification algorithms are an integral part of protective relaying. Recently, research on overcurrent identification and location has gained traction from a reliability viewpoint. Timely detection and location of overcurrent events help reduce outage duration and possibly avoid the recurrence of overcurrents if a faulty component can be identified. Since this research focuses on identifying events based on their cause and providing information that helps to locate problematic components, overcurrent identification and location algorithms that fit this profile, will be reviewed here.

Most fault location schemes assume that an event has already been identified as overcurrent. To truly automate fault location and reporting, a classifier that can differentiate overcurrent events from other power system events should also be incorporated into the fault location scheme. Algorithms that can both determine disturbances to be overcurrent and subsequently attempt to locate them are difficult to find in the literature. For example, impedance based fault location schemes like the one proposed by Girgis [43] required the fault and the fault type to be identified prior to being used for fault location. Girgis's fault location scheme also requires that the feeder topology, load and line impedances to be known. Use of power quality monitors (PQ monitors) has gained popularity and hence methods that use the data collected from these monitors such as [44] have also been proposed. Sabin [45] demonstrated a impedance based fault location method on Consolidated Edison underground primary distribution system. Sabin differentiated overcurrents from inrushes using harmonics. He used data from PQ monitors and also built a detailed model of the distribution system. Obtaining and maintaining feeder topology information and impedance information is big draw back of impedance based fault location schemes. It is both tedious and expensive to gather such data for large scale use of these algorithms.

Traveling wave based fault location schemes for distribution systems such as the one proposed by Thomas [46] do not need feeder topology or impedance information. These methods use fault recorders that record voltage and current samples at a very high sampling rate (around 1.0Mhz). Then they use information contained in the high frequency components of incident and reflected waves to determine fault location. Like the impedance based approach, these algorithms do not try to distinguish between transients caused by faults and those caused by other factors such as capacitor switching. Traveling wave based methods are more common in transmission systems that exhibit simple topologies and are harder to use with distribution systems which exhibit a branching topology. Other than these conventional approaches, expert system based methods [47, 48, 49], fuzzy logic based methods [50, 51, 52, 53], probabilistic reasoning based methods [54, 55, 56] and statistical learning based [57, 58, 59, 60] methods have also been proposed for fault location. These methods also suffer the shortcomings of conventional methods. The shortcomings being, they either assume that that an overcurrent event has been detected or they rely on feeder specific information such as feeder topology, line impedances or information from SCADA or AMR being available.

### *2.3.2 Arcing and High Impedance Fault (HIF) Identification*

Arcing faults have always been a hard to detect problem on electrical power systems. Unlike bolted overcurrent faults, arcing faults are often intermittent, and seldom cause protective devices to operate. Arcing may continue indefinitely without progressing to a sustained fault that draws high enough current to operate a traditional protective device, or to be noticed by the public. Such incipient arcing conditions are a safety hazard to utility crews and the public. Definitive characterization of the nature of downed conductor arcing arcing faults on medium voltage feeders was performed in the late 1970s through 1990s by Russell, et al. [61, 62, 63, 64, 65] . The extensive research performed to characterize and detect arcing faults on medium voltage systems has been summarized by Sedighizadeh, et al [66]. One important distinction between these arcing fault detection methodologies and the methodology outlined in this research [67] is that, the methodology outlined in this research has a wider scope, and is capable of detecting other abnormal conditions such as, but not limited to, recurrent overcurrents and capacitor related problems.

### *2.3.3 Capacitor Bank Switching Identification*

Capacitors banks are used to provide voltage and VAR support on distribution system feeders, and are common devices in power distribution feeders. Due to their ubiquitous nature, capacitor banks are also most prone to failures [68]. Capacitor banks can exhibit a variety of abnormal conditions such as unbalanced operations, restrikes, switch bounce, arcing and repeated cycling due to faulty controllers. However, there is no single method available for detecting all of these conditions. For example, Lee [69] and Santoso [70] provided methods for detecting unbalanced capacitor operations. However, these methods did not detect any other abnormal condition. Socholuliakova [71] presented a method for locating capacitor banks based on voltage transients. However, this method did not differentiate transients caused by normal capacitor switching form those caused by abnormal switching. Khani [72] proposed a method for location and position position identification of capacitor banks based on current and voltage measurements. However, this method did provide a mechanism for detecting abnormal switching of capacitor banks. This research proposes a methodology for detecting various abnormal capacitor operations and also providing information that may help in locating a problematic capacitor bank.

## 2.4 Chapter Summary

In this chapter, the lack of prior art for identifying power system events based on their underlying cause was shown. Existing power quality disturbance classification methods were reviewed. In addition, the shortcomings of existing PQ disturbance classification methods were pointed out. Existing methods for identifying overcurrent, arcing and capacitor switching events were reviewed, and their shortcomings were also discussed. This dissertation addresses the shortcomings of current methodologies and presents a new methodology to classify and report a wide variety of power system events based on their underlying cause.

# 3. PROBLEM FORMULATION

## 3.1 Introduction

The main objective of an electricity supply system is to to supply power that is both reliable and economical, while keeping up with the ever increasing demand for energy. Electrical power originates at a generation station and is transmitted to a load by a system of conductors and other equipment that make up an electrical power system. Transmission lines carry bulk power from generators to bulk power, higher voltage substations.

The electric power distribution system is the part of an electric utility system between the bulk power source and the consumers' point of delivery [73]. Figure 3.1 shows a simplified diagram of a typical distribution system. Distribution systems can be divided in various parts, namely, subtransmission circuits, distribution substations, distribution or primary feeders, distribution transformers, secondary circuits and service drops. The subtransmission circuits deliver energy from the bulk power sources to the various distribution substations located in the load area. The subtransmission voltage is usually between 12.5 and 245 kV. The distribution substation consists of one or more power transformer banks together with the necessary equipment and switchgear to reduce the subtransmission voltage to a lower primary system voltage for local distribution.

Each distribution substation serves its own load area. The area served by the distribution substation is subdivided and a primary feeder, usually operating in the range of 4.6 to 34.5kV, supplies each subdivision. The primary feeder normally consists of either a three-phase, three-wire or a three-phase, four-wire main that runs from the substation to the load center where it branches into three-phase sub-

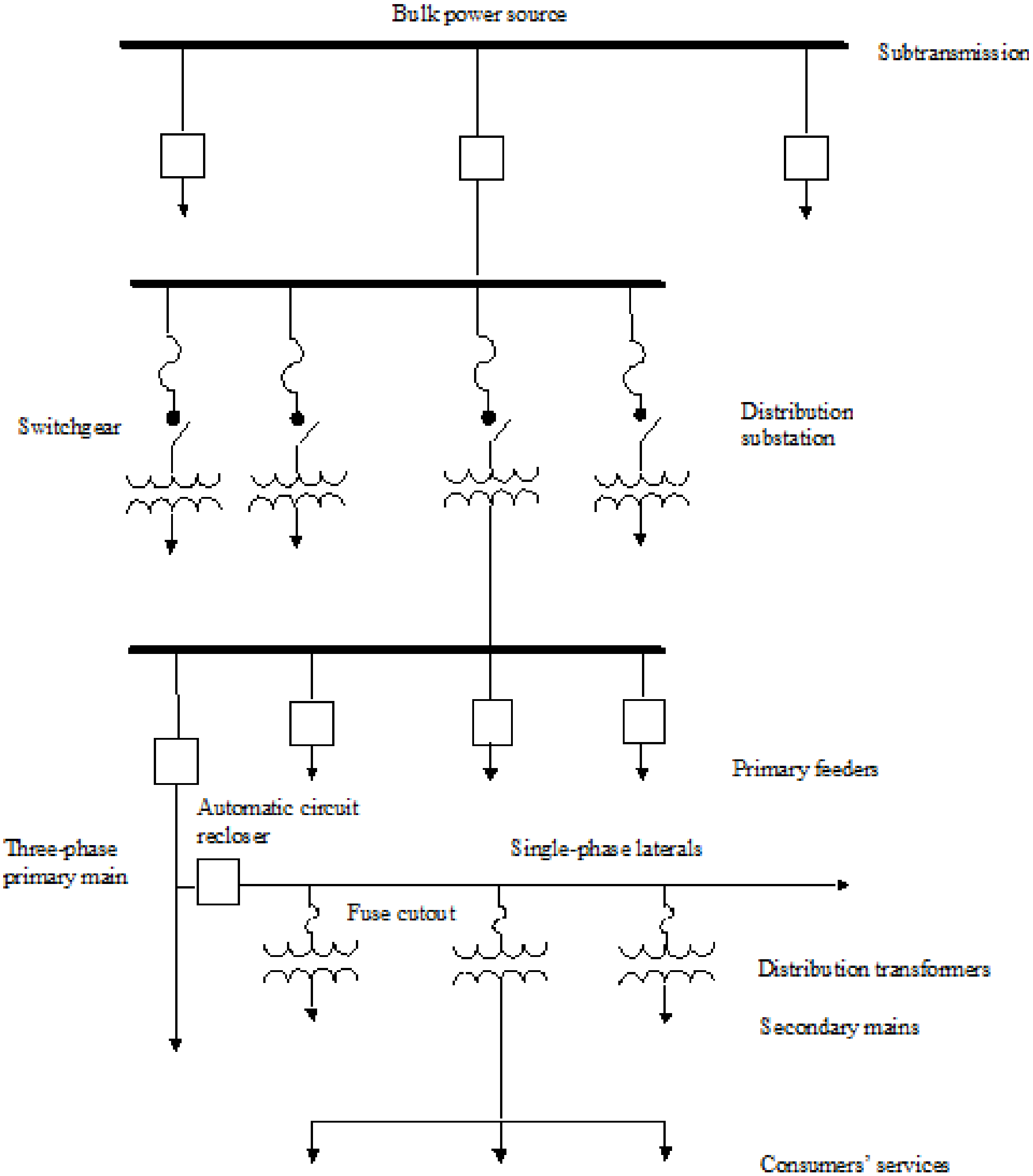


Figure 3.1: One-line Diagram of a Typical Distribution System [73]

feeders and single-phase laterals. The distribution transformers are connected to the primary feeders, subfeeders, and laterals usually through fused cutouts, and supply the radial secondary circuits to which the consumers' services are connected.

The equipment, components, and other apparatus that makes up a power distribution system includes wires, cables, switches, reclosers, insulators, capacitors, etc. As the result of causes such as contact of a conductor by vegetation or due to aging, apparatus can fail or cease to operate normally. Sometimes a failure of apparatus causes an abnormally high current that can further damage the distribution system, injure members of the public or utility personnel, or damage customer devices and equipment. In addition to physical damage, failures often result in degraded power quality, or loss of electric service.

Distribution systems are designed with components, known as protective devices, which interrupt high current flows, preventing damage to the system and potential safety hazards. For example, a relay or recloser is a type of protective device which monitors a circuit and opens a circuit breaker when the current magnitude exceeds a specified threshold for a set duration. When the thresholds are met, the device operates, and power is interrupted to end users.

In many cases, the device recloses to restore power after a predetermined time period. If the failure is still present, power may be interrupted again. In this manner, several power interruptions can result from the same cause. These interruptions may happen either in close succession or spread over an indeterminate period of time. In other cases, failure of an apparatus may cause a power outage. In some other cases, the faulty apparatus may continue to operate abnormally fashion without causing any interruptions but may cause severe disturbances that may affect other equipment present in the power distribution system.

When an outage occurs, failing or failed apparatus often must be repaired or replaced before normal, reliable operation can be restored. When sustained interruptions occur, electrical power system operators such as utility companies must make repairs. Sustained interruptions often occur during “non-business hours,” often increasing interruption duration and the expense of restoration. Repairs made in this manner are often more troublesome and expensive than if they could have been planned in advance. In addition, if repairs can be made before a power outage occurs, overall service is more reliable and of higher quality, and inconveniences and economic losses to customers may be avoided. Therefore, it is preferable to identify failures, incipient failures, and other improper or suboptimal operating conditions of power system apparatus before these conditions affect power quality or cause momentary interruptions or sustained power outages.

Utilities may utilize a variety of methods to reduce the number of power interruptions and outages that occur. Conventional methods vary by utility, but generally fall into two categories: a) physically examining and/or testing individual apparatus periodically in an effort to determine whether they are likely to fail and cause a power outage or power quality problems and b) replacing apparatus according to a predetermined schedule. Problems exist with both approaches. One significant problem is that examining, testing, and/or replacing large number of individual devices can be time-consuming, expensive, and difficult to schedule without interruption power to the end users. Another problem with the second approach is that, when apparatus are replaced according to a predetermined schedule, they may be replaced when they are functioning normally, well before any failure actually begins. Alternatively, they may not be replaced in sufficient time to avoid an outage. Further periodic maintenance or replacements actually can inadvertently introduce problems that did not previously exist. Clearly it would be preferable to know when apparatus are beginning to deteriorate or operate improperly or sub-optimally, so repairs or replacements could be made prior to actual failure.

In an effort to identify failing or failed devices, utilities monitor the operation of distribution circuits by monitoring one or more signals, such as current and voltages. However, as outlined in the previous chapter, most methods currently used for monitoring distribution systems are unable to determine cause of a disturbance. Hence, they are unable to identify the root-cause of the disturbance, such as devices on a circuit that are failing or that have failed. One reason is that majority of conventional methods rely on voltage measurements alone. Failing apparatus often produce small changes in current waveforms, and virtually no change in voltage waveforms. As such, the majority of methods for detecting power system are not sensitive to the primary electrical evidence generated by most failing apparatus.. Another problem with conventional monitoring methods is that human experts skilled in the art of using measured quantities to identify failed or failing devices must analyze real-time or near real-time data obtained from a power system. These experts may not be available for analysis when needed, and manual analysis can be expensive and prone to error. Yet another problem with conventional methods is that much of the data collected may correspond to normal day-to-day operations of the power system, while only a small fraction of collected data corresponds to abnormal operations caused by failing or failed apparatus. It is not feasible to manually analyze such huge amounts of data in a timely and reliable fashion. This research attempts to address these short comings of methods presently used in the utility industry for monitoring the health of distribution systems.

This research work uses data collected by advanced monitoring units developed as a part of Distribution Fault Anticipation (DFA) project, sponsored by the Electrical Power Research Institute (EPRI). Classification algorithms developed developed for this research are currently being utilized by the DFA project. The DFA project and DFA data collection platform will be reviewed in the following sections before proceeding to problem formulation.

## 3.2 Distribution Fault Anticipation Background

Over the past decade, Power System Automation Laboratory (PSAL) at Texas A&M University has developed and applied intelligent systems for monitoring and diagnosing failures and incipient failures of components on distribution systems, sponsored by the Electric Power Research Institute (EPRI) and multiple partner utilities [74]. PSAL began its first formal EPRI-funded project in the area of fault anticipation in 1997. This was Phase I of DFA project or proof-of-concept phase. The goal of that project as stated by Dr. B. Don Russell was to determine the validity of the initial, fundamental hypothesis:

*"Equipment often degrades slowly over time. As it does, it produces measurable electrical changes. Recognizing these changes provides the basis for 'anticipating' faults, thereby avoiding full-blown failures, faults and outages."*

The project involved the design and installation of equipment to perform substation-based monitoring of three feeders, one at each of three utility companies. Data collection was done through continuous monitoring, to detect changes in a variety of electrical parameters and record high-fidelity data when such changes occurred. Dial-up modems were used to retrieve newly captured waveform data. Each captured event waveform was manually analyzed to determine the likely cause on the power system. The Phase I proof-of-concept project provided encouraging results [1]. Over a period of approximately two years, the project documented several examples of line apparatus exhibiting detectable changes in electrical parameters, often well before the utility company or its customers experienced problems. A primary limitation of this research was that its scope was limited to three feeders. A further limitation was that the process of retrieving and processing data was manual and time intensive. Additionally, the number of incidents of incipient failures on three feeders over a nominal two-year period was limited. The incidents that were discovered, however, encouraged EPRI and its members to expand the scope in a Phase II project.

Phase II of DFA project was called 'Field Data Collection and Algorithm Development' phase. PSAL undertook the second phase of the DFA project in the year 2000. This effort expanded the number of monitored feeders significantly, increased the number of utility companies involved in data collection, and increased the level of interaction between the research team and utility engineers. PSAL designed a prototype data collection system with capabilities that were enhanced in comparison to the systems used in Phase I. This system made it feasible for utility companies to purchase prototype systems and instrument more feeders. Approximately eleven utility companies installed prototype monitoring systems in fourteen substations across North America. Data retrieval and analysis procedures were performed manually in Phase I. PSAL recognized that these manpower-intensive procedures would not be feasible for the expanded number of sites and feeders monitored in Phase II. Therefore the prototype system was designed to automate many of these processes in this second phase. The system took advantage of the availability of high-speed Internet for data retrieval. Data retrieval was automated, with waveform files being automatically downloaded to utility owned master stations, and to a "Master Master" station at Texas A&M.This made data collection more efficient and reduced manpower requirements. After installing prototype equipment, participating utilities were responsible for investigating interruptions, outages, and other abnormal occurrences. They also were responsible for determining which recorded data was associated with these events. The expanded effort resulted in a large volume of data and the characterization of numerous normal and abnormal feeder events. At the outset of the project, there was little knowledge about apparatus failure modes and even less knowledge about how progressive failures manifest themselves electrically over time. Because incipient faults do not occur with great frequency, it was important that detection thresholds be configured sensitively enough to capture these infrequent events when they do occur. As a result of these sensitivity considerations and tradeoff analysis a bias was maintained toward high sensitivity, to avoid missing important events. This meant numerous, normal-system events were captured, analyzed and archived. This resulted in considerable knowledge about the signals that occur as various apparatus begin to deteriorate. Utility engineers provided much-needed feedback to determine details about failures that occurred. The expanded data-collection effort of Phase II documented numerous failures and failure precursors [75].

The large-scale data collection activity of Phase II project demonstrated the ability to detect, characterize, and many types of normal and abnormal behavior on distribution feeders. The field devices served primarily as data-capture devices. Master station computers at each host utility company and at PSAL retrieved captured event data from these field units for analysis and processing. The author developed algorithms for characterizing captured events, and created processes for running these algorithms on captured data automatically as they occurred and were captured. Over time, the continued processing of new events allowed the author to evaluate the efficacy of various algorithms and to refine them where appropriate. To facilitate a reasonable mechanism for developing, testing, and refining algorithms for processing data coming from more than 70 feeders, algorithms initially were implemented in MatLab$^{TM}$ and were run on a computer server system at PSAL. This required all recorded data to be retrieved to PSAL's server, via Internet connections to the substations. The volume of data was immense, but manageable for a research project with more than 70 feeders. While this approach was acceptable for a research project, it would not be feasible to retrieve, store, process or archive this level of data on an ongoing basis, for the hundreds of feeders system-wide deployment at a typical utility would require. In addition, this processing hierarchy requires high-speed data transfer to transfer all data back to a centralized location for processing. Phase II purposely relied on centralized data retrieval, processing, and archiving, to facilitate learning for the first time what kinds of electrical signals incipient failures produce, and to allow development and testing of methods for characterizing these signals and signatures. Some of the processes were not fully automated and would be unreasonably cumbersome in full deployment. Communications, storage, and processing limitations dictated that a significantly different architecture would be needed to make DFA technology practical. Considerable work was needed, to develop appropriate architectures and determine requirements for system integration. That work has been the focus of the Phase III project, which is the current phase of the DFA project [76]. An obvious part of the solution is to move as much of the data handling and processing burdens to the lowest possible level in the system hierarchy, which is the feeder device itself. This is the approach taken with the revised platform that is being used for pilot demonstrations.

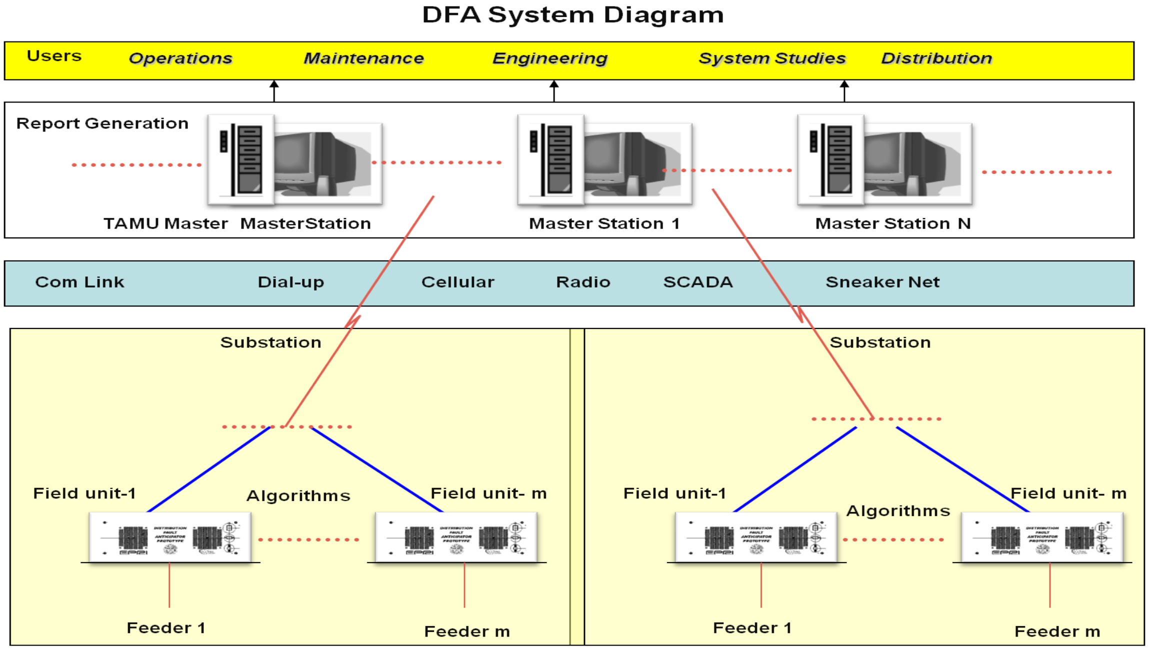


Figure 3.2: DFA System Diagram

### 3.3 DFA Data Collection Platform Description

The DFA data acquisition platform was designed to measure electrical phenomena occurring on 15-kV-class distribution feeders. Measurements are based on passive monitoring of waveforms from the secondary terminals of conventional substation-based current transformers (CTs) and potential transformers (PTs). The DFA platform has evolved over the past decade and has gone through three hardware revisions namely Phase I (proof of concept), Phase II (prototype), and Phase III (system integration).

This research uses data from both Phase II and Phase III platforms collected over the past five years. Both the prototype and commercial pilot platforms share similar data acquisition circuitry, and can be considered to produce more-or-less identical results when observing the same event. Figure 3.2 shows the system diagram of the present DFA data collection platform.

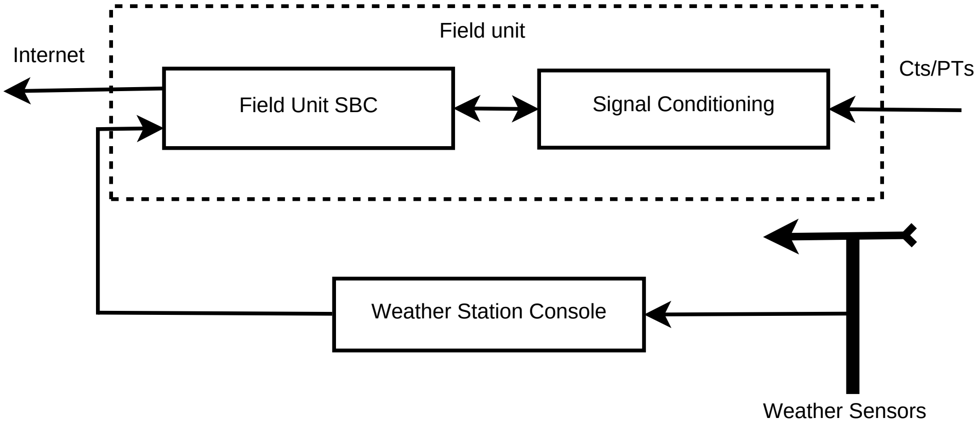


Figure 3.3: Field Unit

The DFA data collection platform consists of two main components: the DFA Field Unit and the DFA Master Station. In order to avoid verbosity, the term DFA Field Unit often is shortened to Field Unit (FU) and the term DFA Master Station is shortened to Master Station (MS).

### *3.3.1 DFA Field Unit*

Each field unit is designed to monitor the current and voltage signals for one feeder. Each field unit provides four current inputs (five-amp nominal) and three voltage inputs (120-volt nominal), for connection to the secondary windings of conventional current and potential transformers (CTs and PTs). Field unit hardware consists of two main components (Figure 3.3). The first is a custom-designed signal-conditioning module (Figure 3.4) and the second is a Single Board Computer (SBC).

The signal-conditioning module (SCM) is responsible for analog conditioning of the feeder's three voltage inputs and four current inputs and for converting each of these inputs to digital format. The SCM interfaces with the CT and PT inputs, provides appropriate signal conditioning, and converts the analog signals to digital samples at a rate of 15,360 samples per channel per second (256 samples

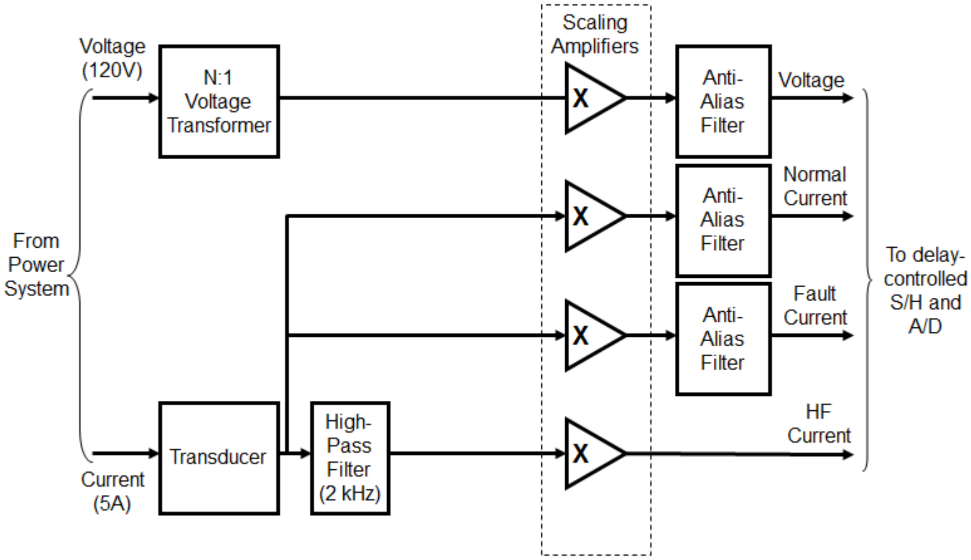


Figure 3.4: Analog Signal Conditioning

per cycle at 60 Hertz). SCM outputs a total of 15 sampled channels, as listed in Table 3.1.

The analog processing of the voltages is significantly different than that of the currents, but both types share the basic conditioning and conversion of the high power inputs from the electric power system to electronic level signals and limiting the signals' frequencies appropriately.

**Voltage Channels:** The nominal input for each of the three voltage inputs is 120V AC (alternating current) volts. The input transformer for each voltage channel consists of a standard N:1 signal voltage transformer. The upper signal flow path in Figure 3.4 shows the analog processing of one voltage channel. The SCM scales each voltage input so that it uses at least 90 percent of the ADC's range. Each voltage channel has an anti-alias filter.

**Current Channels:** The nominal input for each of the four current channels is 5 AC amperes. Analog processing of the current channels is more complex than that of the voltage channels. The lower signal flow path in Figure 3.4 illustrates the

Table 3.1: List of high-speed SCM waveform channels

| **Signal** | **Description** |
|---|---|
| $\mathrm{V_A}$ | Voltage, Channel A |
| $\mathrm{V_B}$ | Voltage, Channel B |
| $\mathrm{V_C}$ | Voltage, Channel C |
| $\mathrm{INorm_A}$ | Normal Current, Channel A |
| $\mathrm{INorm_B}$ | Normal Current, Channel B |
| $\mathrm{INorm_C}$ | Normal Current, Channel C |
| $\mathrm{INorm_N}$ | Normal Current, Channel N |
| $\mathrm{IFault_A}$ | Fault Current, Channel A |
| $\mathrm{IFault_B}$ | Fault Current, Channel B |
| $\mathrm{IFault_C}$ | Fault Current, Channel C |
| $\mathrm{IFault_N}$ | Fault Current, Channel N |
| $\mathrm{IHF_A}$ | High Frequency Current, Channel A |
| $\mathrm{IHF_B}$ | High Frequency Current, Channel B |
| $\mathrm{IHF_C}$ | High Frequency Current, Channel C |
| $\mathrm{IHF_N}$ | High Frequency Current, Channel N |

analog processing for one current channel. The initial stage for each current channel uses a transducer to create a voltage waveform signal proportional to the input current waveform. After this conversion to a proportional voltage waveform, the signal's path is split into multiple paths and subsequent analog processing results in three distinct outputs, each providing particular frequency components, dynamic ranges, and resolutions. These signal paths are described below.

1. Normal Current Channel: The intent of the Normal Current channel is to provide the fundamental-frequency current and its harmonics during normal conditions (i.e., the majority of conditions, with the exception of overcurrent faults). The SCM scales each Normal Current signal so that it uses at least 90 percent of the ADC's range when five RMS AC amperes are applied to the corresponding input. Each Normal Current channel has an anti-alias filter in its path before the ADC. The ADC converts this signal at a rate of 15,360 samples per second (i.e., 256 samples per fundamental-frequency cy-

cle).

2. Fault Current Channel: The intent of the Fault Current channel is to provide the fundamental-frequency current and its harmonics during overcurrent conditions. Its processing is identical to that of the Normal Current channel, except that its scaling is such that it uses at least 90 percent of the ADC's range when 100 RMS AC amperes are applied to the corresponding input. Each Fault Current channel has an anti-alias filter in its path before the ADC. The ADC converts this signal at a rate of 15,360 samples per second (i.e., 256 samples per fundamental-frequency cycle).

3. High-Frequency Current Channel: The intent of the High-Frequency Current channel is to represent the level of energy for current signals in the frequency range between 2,000 and 10,000 Hertz. The signal passes through a high-pass filter with a corner frequency of 2,000 Hertz. The output of this high pass filter then passes through a low-pass filter with a corner frequency of 10,000 Hertz. The ADC converts this signal at a rate of 15,360 samples per second (i.e., 256 samples per fundamental-frequency cycle). The low-pass filter's corner frequency specification violates the Nyquist criterion. This was deemed acceptable because the primary purpose of this channel is to measure the amount of energy in the current's high-frequency range, not necessarily to extract its frequency content with high precision.

The SCM interfaces with adjust single-board computer (SBC) via PCI-104 bus. The SBC has on board RAM and flash based storage. The feeder module's SBC communicates with its signal-conditioning module via PCI-104 bus. This allows the feeder module's SBC to adjust various settings on the signal-conditioning module and allows the signal-conditioning module to send its digitized samples to the field unit's SBC for further processing.

The field unit's SBC performs the following functions:

1. Performs real-time numerical operations on the data, including computation of standard power system quantities, frequency components, etc.

2. Monitors electrical parameters based on real time computations, and triggers high-speed waveform captures, storing them to waveform files.

3. Computes the following quantities from sampled channels and optionally storing them to waveform files:

   (a) The RMS of each of the sampled signals.

   (b) The RMS of a point-by-point subtraction from previous cycle of normal current, fault current, high-frequency current, and voltage waveforms.

   (c) The individual components of a 2-cycle FFT of normal current, up to 960Hz.

   (d) Cycle-by-cycle values of real power (P), reactive power (Q), complex power (S), and power factor.

   (e) Phase angles of the 60Hz component of for both voltage and normal current waveforms.

4. Calculating various statistical values that are periodically recorded stored in a database. These statistical values include maximum, minimum, average and standard deviation computed for both electrical quantities and weather data. In this dissertation, this statistical data will be referred to as 'interval data.'

5. Managing data storage, and ensuring disk space availability for continued data collection.

6. Optionally running classification algorithms to classify captured data. Classification results and a subset of extracted features are stored in a database to aid further processing.

7. Optionally running clustering algorithms that use classification results and features extracted by classification algorithms to group related data captures. Results of clustering algorithms are also written to a database.

8. Communicating with Master Station(s) for the following purposes:

   (a) Obtaining software updates from Master Station.

   (b) Obtaining settings information from Master Station.

   (c) Transferring waveform data to Master Station.

   (d) Synchronizing local database with Master Station database(s) to transfer classification and aggregation results.

An optional component of the Field Unit is a weather station. The weather station is a commercially available, off-the-shelf component. It provides a suite of weather sensors that connect to a weather station console residing in the substation control house. This console provides a serial connection that allows the Field Unit's SBC to poll it for real-time weather values. Measured weather parameters include temperature, relative humidity, wind speed and direction, and rainfall.

### *3.3.2 DFA Master Station*

The master station became part of the DFA platform in the second phase of DFA project. During the second phase, the prototype system was redesigned to automate data retrieval. Data was automatically retrieved to master stations. This made data collection more efficient and reduced manpower requirements.

Each FU communicates with a master station. This communication is performed over a TCP/IP connection.The master station is responsible for data retrieval and storage, as well as updating settings and firmware for the device in the field. During each communication interval, the master station does the following:

1. Synchronizes database records on the field with those on the master station.

2. Retrieves all available waveform files.

3. Retrieves all interval data.

4. Updates any system settings that have changed (e.g. triggering thresholds, CT/PT settings, etc),.

5. Performs software updates to the field device, if any are available.

The master station serves as a common collection point, maintaining a full database of all events recorded and retrieved. Data contained at the master station can be opened by custom software (DFAGui) which allows for visual analysis of large numbers of waveform files retrieved from field, as well as viewing interval data.

DFAGui allows for viewing both classification results provided by automatic classification algorithms, or manual classifications assigned by a user. DFAGui also allows the user to view classifications that have been already assigned to a waveform record. DFAGui allows for viewing both classification results provided by automatic classification algorithms, or manual classifications assigned by a user. Manual classifications serve as a means for validating classification performance of automatic classification algorithms developed as a part of this research work.

Since the master station at Texas A&M serves as a central repository of data collected from field units across multiple utilities, it lends itself well to the following functions:

1. Running automatic classification algorithms on waveform data collected from all field units. Waveform data will further be discussed in the following section 3.3.3.

2. Generate automatic reports and alerts based on the results of running automatic classification algorithms. These reports and alerts are then made available through a web page.

### *3.3.3 Waveform Files*

The main source of data for this research are waveform files. Waveform files provide a snapshot of waveform data recorded during a power system disturbance. Each FU continually monitors both current and voltage waveforms looking for power system disturbances. The majority of such disturbances or transients are due to normal system events, and are not of any special interest.

Information written into the recorded waveform files is configurable. For this research, FUs were configured to record all available channels. This included the fifteen high-speed channels in Table 3.1, as well as all harmonic and other calculated parameters as described in Section 3.3.1.

## 3.4 Power Distribution System Event Classification Problem Formulation

The primary objective of this research is to develop a new on-line, non-intrusive classification system for identifying and reporting normal and abnormal power system events on a distribution feeder based on their underlying cause. In this research context, an event or disturbance is any observed activity on the power distribution feeder that manifests itself as a measured changes in electrical signals.

Power distribution system event classification offers many challenges. For it to be practical and to be used in the field, distribution power system classification algorithms should be able to meet the following requirements:

1. Classify both normal and abnormal power system events accurately based on their underlying cause. The majority of power system events are normal system operations. The classification algorithm should be able to identify normal system events (such as motor starts, load switching and normal capacitor switching) so that they can safely be ignored. The classification algorithm should also be able to detect abnormal system operations that generally make up less than one percent of all power system events. Some misclassifications within the normal event category may be acceptable (ex: labeling a motor start as load switching). However, labeling a normal system event as an abnormal system event or vice-versa is a more serious error. This is because utility personnel are typically interested in abnormal system events. Misclassifications of abnormal system events (i.e. false positives or false negatives) will cause utility personnel not to trust the classification system. This will undermine usefulness of the classification system.

2. Provide information about a problematic component to help locate the component. When power system components misoperate or fail causing abnormal system events, simply identifying the problem is not always sufficient. If the classification algorithm provides details of the specific component, it

helps utility personnel to quickly locate the source of the problem. For example, consider an unbalanced capacitor operation. If the algorithm classifies the event and reports it a unbalanced capacitor, then the utility personnel may have to look at all the capacitor banks on the feeder for possible problem. However if the algorithm also reports the size of the capacitor and the phase that did not switch, it may help narrow the search to a specific capacitor bank.

3. Classify events under varying operating conditions and noise levels. Power disturbances signatures observed on these systems vary for many reasons, such as the following:

    (a) Distribution feeders have differing characteristics based on utility operating procedures and the types of customers they serve. For example, different utilities may operate their distribution systems at different voltages (7KV, 24KV etc.).

    (b) Distribution systems span different geographical regions and hence are affected by environmental factors such as weather and vegetation.

    (c) Components making up the distribution systems vary. For example some utilities may decide to use ungrounded capacitor banks while most use grounded capacitor banks.

    (d) Sensors used to record power system disturbances may be different or the sensors may be configured differently. For example, some utilities may choose to connect potential transformers (PTs) phase-to-phase voltage instead of phase-to-ground.

4. Classify events when one or more signals are missing. Based on utility pol-

icy, utilities may decide to instrument only a subset of voltage or current phases. Alternately, one or more channels may be missing because of faulty sensors or they may have been intentionally removed from waveform files to reduce disk space. Classification algorithm must be able to deal with such missing data channels.

5. Classify in 'soft' real-time fashion. To help utility personnel quickly identify, remediate and decrease outage times, the classification algorithm needs to be online and provide classification results in a timely fashion. Utilities may prefer certain activities such as overcurrent faults that caused an outage to be reported as soon as possible (within a few minutes), while some abnormal operations such as unbalanced capacitors may be reported once a day.

Designing a classifier that meets all the above requirements is challenging. It is preferable to break down the power system event classification problem into manageable tasks. To this end, the power system event classification problem is presented as a collection of sub problems. Then, a fuzzy logic based classification scheme is presented as a proposed solution.

### *3.4.1 Multi-phase Power System Event Classification Problem Statement*

Power system event classification algorithms designed to work with power distribution system data need to handle mutli-phase data. This is because most distribution power systems are three phase in nature. Input signals to these algorithms are three phase in nature and their outputs need to include phase involvement and ground involvement as two attributes besides the class label. In this research, it is also desired to provide some event specific information that will help narrow the location of the component causing the event. As a result, the power system event classification problem can be formulated as follows.

Let $x$ represent waveform data instance as shown in equation below:

$$x = \left(x_{samp}, x_{cyc}\right) \tag{3.1}$$

$$x_{samp} = (V, VPD, I, IPD, IHF) \tag{3.2}$$

$$x_{cyc} = \left(V_{RMS}, VPD_{RMS}, I_{RMS}, IPD_{RMS}, P_{phasor}, Q_{phasor}\right) \tag{3.3}$$

where:

$x_{samp}$represents high speed data sampled at a rate of 15360 samples/second.

$x_{cyc}$ represents computed waveforms that have a sample rate of 60 samples/second also referred to as per-cycle waveforms.

$V$ represents three phase high speed voltage samples recorded over the duration of the event.

$I$ represents three phase current and neutral current samples recorded over the duration of the event.

$VPD$ represents three phase high speed phasor differenced[1] voltage samples computed after removing ambient load over the duration of the event.

$IPD$ represents three phase phasor differenced current and neutral current samples computed after removing ambient load over the duration of the event.

[1] Phasor differencing is a technique developed at Texas A&M's Power System Automation Laboratory; it is used for removing an estimated steady-state load component from a sampled signal in the presence of phase drifts introduced either due to sampling or due to changes in power system frequency [77].

$IHF$ represents three phase high frequency current and high frequency neutral current samples recorded over the duration of the event.

$V_{RMS}$ represents three phase RMS voltage computed over the duration of the event.

$I_{RMS}$ represents three phase RMS current and RMS neutral current computed over the duration of the event.

$VPD_{RMS}$ represents three phase RMS phasor differenced voltage computed over the duration of the event.

$IPD_{RMS}$ represents three phase RMS phasor differenced current and neutral current computed over the duration of the event.

$P_{phasor}$ represents three phase real phasor power computed over the duration of the event.

$Q_{phasor}$ represents three phase reactive phasor power computed over the duration of the event.

Let $y$ represent classifier output as shown in equation below:

$$y = \left(y_e, y_p, y_{Grnd}, y_{Pos}\right) \tag{3.4}$$

where $y_e \in \{y_{e1}, ..., y_{eL}\}$ represents power system event class labels, $y_p \in \{A, B, C, AB, BC, CA, ABC\}$ represents phase labels and $y_{Grnd} \in \{0, 1\}$ represents whether or not ground was involved and $y_{Position} \in \{0, 1\}$represents whether or not the event was caused by an activity down stream of the monitoring unit.

Let $i$ represent event specific output information as shown in equation below:

$$i = (z_1, \cdots, z_F), \ F \in \mathbb{N} \tag{3.5}$$

where $i$ represents event specific information that can be used to locate a power system component causing the event.

Then, the multi-phase power system event classification problem can be stated as: Given a desired mapping $g : X \to Y \times I$ that correctly maps waveform data instances $x \in X$ to output $(y, i), \ y \in Y, \ i \in I$ and given manually labeled data $D_M = \{(x(1), y(1), i(1)), \cdots, (x(m), y(m), i(m))\}$, design an algorithm that produces the mapping $h : X \to Y \times I$ that closely approximates the correct mapping $g$.

In the above formulation, the requirement '$h$ closely approximate the correct mapping $g$' needs further attention. Let $h$ map waveform data instances $x \in X$ to output $(\hat{y}, \hat{i}), \ \hat{y} \in Y, \ \hat{i} \in I$. Since $g$ is unknown, let us assume that the desired mapping $g \in G$, be a function in the hypothesis space $G$. Then $g$ can be represented in terms of a 'correctness' function $f : X \times Y \times I \to \mathbb{R}$ as

$$g(x) = arg \underset{(y,i)}{max} \{f(x, y, i)\} \tag{3.6}$$

such that $g$ always returns $(y, i)$ that gives maximum correctness value as perceived by an expert. The requirement "$h$ closely approximate the correct mapping $g$" can be interpreted to mean reducing perceived error (as seen by an expert) between the output of $h$ with respect to the output $g$. In the context of power system event classification, it is more intuitive to interpret the requirement to mean reducing a perceived cost (as seen by utility personnel) due to differences in the output of $h$ and $g$ . This is because, when a classification algorithm misclassifies and reports incorrect information to utility personnel, it could potentially result in wasted time and effort or it could undermine the usefulness of the algorithm. Let $J\left((y,i),(\hat{y},\hat{i})\right)$ , $J : X \times Y \times X \times Y \rightarrow \mathbb{R}$ represent such a perceived cost function. For notational simplicity, let $J\left((y,i),(\hat{y},\hat{i})\right) = J_{g,h}$. Clearly, perceived cost of misclassification $J_{g,h}$ is subjective and may even vary based on utility policies, it will be hard to rigorously derive a mathematical expression for it. Without any loss of generality, for the purpose of this research $J_{g,h}$ is represented as:

$$J_{g,h} = j(M_e, M_p, M_{Grnd}, M_{pos}, M_i),\ j : \mathbb{R}^5 \rightarrow \mathbb{R} \tag{3.7}$$

where:

$M_e = m_e\left(y, \hat{y}\right),\ m_e : Y \times Y \rightarrow \mathbb{R}^{\geq 0}$ represents the cost of misclassifying event type $y_e$ as $\hat{y}_e$. $m_e\left(y, \hat{y}\right) = 0$ when $y_e = \hat{y}_e$.

$M_p = m_p\left(y, \hat{y}\right),\ m_p : Y \times Y \rightarrow \mathbb{R}^{\geq 0}$ represents the cost of misclassifying phase type $y_p$ as $\hat{y}_p$. $m_p\left(y, \hat{y}\right) = 0$ when $y_p = \hat{y}_p$.

$M_{Grnd} = m_{Grnd}\left(y, \hat{y}\right),\ m_{Grnd} : Y \times Y \rightarrow \mathbb{R}^{\geq 0}$ represents the cost of misidentifying ground involvement $y_{Grnd}$ as $\hat{y}_{Grnd}$. $m_{Grnd}\left(y, \hat{y}\right) = 0$ when $y_{Grnd} = \hat{y}_{Grnd}$.

$M_i = m_i\left((y,i),(\hat{y},\hat{i})\right),\ m_i : Y \times I \times Y \times I \rightarrow \mathbb{R}^{\geq 0}$ represents the cost of errors in estimating event specific parameters $i$ as $\hat{i}$. $m_i\left((y,i),(\hat{y},\hat{i})\right) = 0$ when $i = \hat{i}$.

Based on Equation 3.7, the multi-phase power system event classification problem can be restated as: Given a desired mapping $g : X \rightarrow Y \times I$ that correctly maps event data instances $x \in X$ to output $(y, i)$, $y \in Y$, $i \in I$ and given manually labeled data $D_M = \{(x(1), y(1), i(1)), \cdots, (x(m), y(m), i(m))\}$, design an algorithm that produce the mapping $h : X \to Y \times I$ that minimizes total perceived cost of misclassification $C(D_M)$:

$$C(D_M) = \sum_{k=1}^{K} j(M_{e,k}, M_{p,k}, M_{Grnd,k}, M_{pos,k}, M_{i,k}) \tag{3.8}$$

where:

$M_{e,k} = m_e(y(k), \hat{y}(k))$, $M_{p,k} = m_p(y(k), \hat{y}(k))$, $M_{Grnd,k} = m_{Grnd}(y(k), \hat{y}(k))$ and $M_{i,k} = m_i((y(k), i(k)), (\hat{y}(k), \hat{i}(k)))$.

$h(x(k)) = (y(k), i(k))$ and $g(x(k)) = (\hat{y}(k), \hat{i}(k))$.

The above formulation looks very similar to a supervised learning problem. However, the intention here is not to impose the requirement of $h(x)$ being a supervised learning algorithm. Parameters of $h(x)$ need not be automatically tuned using the objective function in Equation 3.8. Instead, it is possible to design $h(x)$ as a 'handcrafted' expert system based classifier. This is an appealing option for the following reasons:

1. Years of experience gained by manually analyzing waveform data (Section 3.2) can be directly coded into the expert system instead of having to be learned automatically.

2. It will be shown in Section D.5 that power system event classification problem is a large scale classification problem. Knowing that this is a large scale classification problem, it will be both challenging and counter productive to try and design a classifier based on machine learning techniques; especially when expert knowledge can be used to simplify the problem [78].

3. Using handcrafted expert system does not impose any constraints on the type of objective function that can be used to evaluate classifier performance. Machine learning techniques use cost functions to tune classifier parameters. Using nonlinear, non smooth or complex logical reasoning within the cost function may severely limit the tools that can be used to train the classifier.

For the above reasons, an expert system based classifier was chosen. For an expert system, the total perceived cost $C(D_M)$ may be implicit and can be used for making initial adjustments to expert system as a part of designing the classifier. Total perceived cost $C(D_M)$ may also be used for regression testing. For example, let $h(x)$ be modified either to accommodate new knowledge or to fix an existing deficiency. Let $h'(x)$ be the modified version. Then Equation 3.8 provides a way to quantify the performance of $h'(x)$ against $h(x)$.

### 3.4.2 Segmentation - Problem of Detecting Sub-events

Waveform data instances $x$ (Equation 3.1) may contain data corresponding to one or more related or unrelated power system events. An expert looking at waveform data will be able to identify and process these distinguishable power system activity. Such related or unrelated and distinguishable power system activity within waveform data instances are called sub-events. The classification problem formulated in the previous section does not explicitly account for the possibility of sub-events, the problem formulation is modified to accommodate sub-events. Let waveform data instance $x$ be a vector of sub-event data as shown in equations below:

$$x = (x_1, \cdots, x_S),\ S > 0 \tag{3.9}$$

$$x_s = \left(x_{samp}(s), x_{cyc}(s)\right) \tag{3.10}$$

where:

$S$ is the number of sub-events in waveform data instance $x$.

$x_s$ represents sub-event data for sub-event $s$.

$x_{samp}(s)$ represents high speed data within a sub-event $s$.

$x_{cyc}(s)$ represents computed 60 samples per second quantities within a sub-event $s$.

When waveform data instance is broken in to sub-events, each sub-event $s$ can have its' own output labels and event specific information. If there was only one sub-event, or each of the sub-events were related and were caused by the same power system component, then it would be possible to assign a single event category to the whole waveform data instance. However if there were more than one unrelated sub-events within a waveform data instance, there are two ways it could be handled:

1. Allowing each waveform data instance to be assigned more than one event category type.
2. Allowing each waveform data instance to be assigned only one event category. This could be done by choosing an event category type that would be perceived as the most beneficial to utility personnel.

For example, consider the scenario where a waveform data instance contained two unrelated sub-events. Let one sub-event correspond to a motor start and a second sub-event correspond to an arcing capacitor switch. It is then possible to assign both motor start and arcing capacitor switch class labels to the waveform data instance. Alternately, it is possible to assign only arcing capacitor switch label; as it will be perceived to be more important category. As a matter of practicality, and to keep the system simple, it is preferable to do the latter (i.e., assign only arcing capacitor switch label). This is the approach taken in this research. It is assumed that the desired mapping $g$ (Equation 3.6) always returns a single set of output class labels and event specific information $(y, i)$ that correspond to a sub-event which will be perceived as the most important to be reported.

Designing a segmentation algorithm to break waveform data instance to sub-events is one of the sub-problems involved in designing a power system event classification algorithm. Now, the segmentation problem can be stated as: Design a segmentation algorithm $T_{Seg}$ as part of the algorithm that minimizes total perceived cost (Equation 3.8), where:

$$T_{Seg}(x) = (x_1, \cdots, x_S)\,,\ S > 0 \tag{3.11}$$

### *3.4.3 Feature Extraction - Problem of Extracting Event Characteristics*

It will be difficult to design a classifier that directly uses raw waveform data $x$ (Equation 3.1). This is because the dimension of the input space for such a classifier will be too large and non deterministic. It is common to transform raw data into a more manageable feature space through a feature extraction process. Let $x_{Fea} = T_{Fea}(x)$, $T_{Fea} : X \to X_{Fea}$, $x_{Fea} \in X_{Fea}$ represent such a transformation. where $x_{Fea}$ represents feature vector and $X_{Fea}$ represents a feature space which has much reduced dimensionality, when compared to the input waveform data space $X$. Choice of the feature space $X_{Fea}$ is affected by many factors, most important of which is domain knowledge. Based on the knowledge obtained from others skilled in this art and based on the author's experience gained by manually analyzing power system waveform data , the following representation is chosen for feature vector and feature space:

$$x_{Fea} = \left(x_{Shape}, x_{Generic}, x_{Specific}\right)\ ,\ X_{Fea} = X_{Shape} \times X_{Generic} \times X_{Specific} \tag{3.12}$$

where:

$x_{Shape} \in X_{Shape}$ represents shape based features observed in per-cycle waveforms..

$x_{Generic} \in X_{Generic}$ represent features derived using simple mathematical operations on waveform data including, but not limited to, finding mean, maximum, minimum values.

$x_{Specific} \in X_{Specific}$ represent characteristic signature of specific power system event type. Dedicated algorithms are needed for extracting these event specific features.

Section 3.4.6 will further explain the rationale behind choosing a feature space as described above. Accounting for the possibility of the waveform data to be segmented into sub-events, each of the three feature vectors $x_{Shape}$, $x_{Generic}$ and $x_{Specific}$ can be expressed as follows:

$$x_{Shape} = \left(x_{Shape}(1), \cdots, x_{Shape}(S)\right)\ ,\ S > 0 \tag{3.13}$$

$$x_{Generic} = \left(\left(x_{Generic}(1), \cdots, x_{Generic}(S)\right), \bar{x}_{Generic}\right)\ ,\ S > 0 \tag{3.14}$$

$$x_{Specific} = \left(\left(x_{Specific}(1), \cdots, x_{Specific}(S)\right), \bar{x}_{Specific}\right)\ ,\ S > 0 \tag{3.15}$$

where:

$S$ is the number of sub-events in waveform data .

$x_{Shape}(s)$ represents shape based features extracted from sub-event data $x_s$.

$x_{Generic}(s)$ represents generic features extracted from sub-event data $x_s$.

$\bar{x}_{Generic}$ represents a sub-event independent component of the generic features derived the whole of waveform data $x$.

$x_{Specific}(s)$ represents event specific features extracted from sub-event data $x_s$.

$\bar{x}_{Specific}$ represents a sub-event independent component of specific features extracted from the whole of waveform data $x$.

Designing feature extractors $T_{Shape}(x) = x_{Shape}$, $T_{Generic}(x) = x_{Generic}$ and $T_{Specific}(x) = x_{Specific}$ is one of the sub-problems involved in designing a power system event classification algorithm. The next section discusses shape based feature extractors $T_{Shape}$ and event specific feature extractors $T_{Specific}$.

#### 3.4.3.1 *Shape based feature extractors*

Shape based feature extracts, extract shape related information from waveform data. Shape based feature extractors are required to do the following:

1. Detecting characteristic shapes including, but not limited to step changes, dips, surges in per-cycle waveforms. In this sense it is a classifier by itself and the class labels correspond to shapes seen in per-cycle waveforms.

2. Estimating parameters associated with the shapes. For example, if the feature extractor detects a step change, then it will also estimate the size of the step change.

Based on the above requirements, designing the feature extractor $T_{Shape}$ is equivalent to designing an underlying cycle-by-cycle waveform shape classifier $C_{Shape}$ and shape dependent parameter estimator $P_{shape}$ as summarized below:

$$T_{Shape}\left(x\right) = \left(x_{Shape}(1), \cdots, x_{Shape}(S)\right),\ S > 0 \tag{3.16}$$

$$x_{Shape}(s) = \left(L_{V_{RMS}(s)}, L_{VPD_{RMS}(s)}, L_{I_{RMS}(s)}, L_{IPD_{RMS}(s)}, L_{P_{phasor}(s)}, L_{Q_{phasor}(s)}\right) \tag{3.17}$$

$$L_{Sig(s)} = \left(c_{Sig(s)}, p_{Sig(s)}\right),\ Sig \in \left\{V_{RMS}, VPD_{RMS}, I_{RMS}, IPD_{RMS}, P_{phasor}, Q_{phasor}\right\} \tag{3.18}$$

$$c_{Sig(s)} = C_{Shape}\left(Sig\left(s\right)\right),\ 1 \leq s \leq S,\ c_{Sig(s)} \in \left\{Shape_1, \cdots, Shape_{N_s}\right\} \tag{3.19}$$

$$p_{Sig(s)} = P_{Shape}\left(Sig\left(s\right), c_{Sig(s)}\right),\ 1 \leq s \leq S,\ p_{Sig(s)} \in \left\{p_1, \cdots, p_{N_p}\right\} \tag{3.20}$$

where:

$T_{Shape}\left(x\right)$ is the shape based feature extractor that needs to be designed.

$S$ is the number of sub-events in waveform data .

$x_{Shape}(s)$ represents shape based features extracted from sub-event data $x_s$ corresponding to sub-event $s$.

$L_{Sig(s)}$ represents shape based features extracted from the signal data $Sig\left(s\right)$ within the sub-event $s$.

$c_{Sig(s)}$ represents the shape detected in signal data $Sig\left(s\right)$ within the sub-event $s$.

$p_{Sig(s)}$ represents the shape dependent parameters extracted for signal data $Sig\left(s\right)$ within the sub-event $s$.

#### 3.4.3.2 *Event specific feature extractors*

There are some features needed for power system event classification that are not shape based features. They do not fall into the category of generic features either.These features are grouped under event specific features. For example, consider the following requirements:

1. Finding certain signature within the waveform data can sometimes have a major influence on deciding whether or not the data belongs to a certain event category. It is possible to design algorithms to detect the presence or absence these features. These algorithms can be treated as two class classifier whose output is either zero or one based on the presence or absence of a signature.

2. Estimating certain features from waveform data, which by themselves may not have a major influence on the classification result; but when used with other features, may help to improve classification accuracy for some event categories.

3. One of the requirements placed on the classification algorithm is to extract event specific information or features $\hat{\imath}$ (Section 3.4.1). These features could aid in locating the cause of an event. Some of these features need dedicated algorithms for analyzing and extracting them from waveform data.

Based on the above two requirements, designing the specific feature extractor $T_{Specific}$ is equivalent to designing a set of classifiers $C_{Signature}$ that detect the presence of a characteristic signature and a set of specific information extractors $P_{Specific}$

as summarized below:

$$T_{Specific}\left(x\right) = \left(\left(x_{Specific}(1), \cdots, x_{Specific}(S)\right), \bar{x}_{Specific}\right),\ S > 0 \tag{3.21}$$

$$x_{Specific}(s) = \left(\left(c_{Signature1}\left(s\right), \cdots, c_{SignatureNs}\left(s\right)\right), \left(p_{specific1}\left(s\right), \cdots, p_{specificNp}\left(s\right)\right)\right) \tag{3.22}$$

$$\bar{x}_{Specific} = \left(\left(\overline{c}_{Signature_1}, \cdots, \overline{c}_{Signature_{\overline{Ns}}}\right), \left(\overline{p}_{specific_1}, \cdots, \overline{p}_{specific_{\overline{Np}}}\right)\right) \tag{3.23}$$

$$c_{Signature_n}\left(s\right) = C_{Signature_n}\left(x_s\right),\ c_{Signature_n}\left(s\right) \in \{0,1\},\ 1 \leq n \leq Ns \tag{3.24}$$

$$\overline{c}_{Signature_{\overline{n}}} = \overline{C}_{Signature_{\overline{n}}}\left(x\right),\ \overline{c}_{Signature_{\overline{n}}} \in \{0,1\},\ 1 \leq \overline{n} \leq \overline{Ns} \tag{3.25}$$

$$p_{Specific_n}\left(s\right) = P_{Specific_n}\left(x_s\right),\ 1 \leq n \leq Np \tag{3.26}$$

$$\overline{p}_{Specific_{\overline{n}}} = \overline{P}_{Specific_{\overline{n}}}\left(x\right),\ 1 \leq \overline{n} \leq \overline{Np} \tag{3.27}$$

where:

$T_{Specific}\left(x\right)$ is the event specific feature extractor that needs to be designed.

$S$ is the number of sub-events in waveform data .

$x_{Specific}(s)$ represents event specific features extracted from sub-event data $x_s$.

$\bar{x}_{Specific}$ represents a sub-event independent component of specific features extracted from the whole of waveform data $x$.

$c_{Signature_n}\left(s\right)$ represents weather or not a $Signature_n$ was detected within the sub-event $s$.

$C_{Signature_n}\left(x_s\right)$ represents a detector for $Signature_n$.

$\overline{c}_{Signature_{\overline{n}}(s)}$ represents weather a sub-event independent signature $Signature_{\overline{n}}$ was detected within the waveform data.

$\overline{C}_{Signature_{\overline{n}}}(x_s)$ represents a detector for a sub-event independent signature $Signature_{\overline{n}}$.

$p_{Specific_n}(s)$ represents $n^{th}$ specific feature vector extracted using specific feature extractor $P_{Specific_n}(x_s)$ from sub-event data $x_s$.

$\overline{p}_{Specific_{\overline{n}}}$ represents $\overline{n}^{th}$ sub-event independent specific feature vector extracted using specific feature extractor $\overline{P}_{Specific_{\overline{n}}}(x)$ from waveform data $x$.

Now, the feature extraction problem can be stated as: Design feature extractors $T_{Shape}(x) = x_{Shape}$, $T_{Generic}(x) = x_{Generic}$ and $T_{Specific}(x) = x_{Specific}$ as part a of the algorithm that minimizes total perceived cost (Equation 3.8).

### 3.4.4 *Event Classification and Feature Analysis - Problem of Classifying Under Uncertainties*

Power distribution systems are diverse in terms of the different equipment installed on the feeder, nature of the loads, utility operating policies etc. Hence the nature of waveform data obtained from monitoring distribution feeders also varies considerably. A classifier that uses features extracted power system waveform data should, therefore, be able to provide accurate results under these diverse conditions. This is especially challenging because of uncertainties introduced in waveform data by various factors as elaborated below.

**Uncertainties in describing event signatures:**

One of the steps of designing the classifier is associating certain signatures or features with an event category. Event signatures for the same event category can vary across distribution systems or even within a distribution system. It is often not possible to describe event signatures or their relation to an event category, in a precise manner. This introduces uncertainty in partitioning input feature space.

**Uncertainties due to ambient data:**

A Waveform data instance provides a snapshot of electrical signals for the feeder being monitored when the waveform data was recorded. This snapshot represents the net electrical activity of all components that make up the power system as seen from the point of measurement during the period of measurement. Hence waveform data may contain electrical signatures from component(s) of interest superimposed over ambient signal representing the rest of the system. Let the electrical signatures from component(s) of interest be called event data. Power system event classification can be made much simpler if ambient signal can be removed from the waveform data. This would leave only the electrical signatures from components(s) of interest to be further analyzed and classified. This is equivalent to measuring event data at the source of the disturbance instead of the substation. In practice, it is not possible to remove ambient signal from event data entirely. This is because of the time varying nature of power system signals and the dynamic nature of power distribution system. Often, the component(s) causing the power system even may effect changes in the behavior of the rest of the system. Inability to fully discern event data from ambient data introduces uncertainties in classifying waveform data.

**Uncertainties in measurements:**

Data channels contained in waveform data may sometime be missing. This is because, the utility may opt not to measure some channels, or the sensor used for measurement may be faulty. One or more data channels contained in waveforms may be noisy because of faulty sensors. Missing or noisy measurements introduce uncertainty in the input data. For it to be used in a wide spread fashion, the power system event classifier should be able to handle uncertainties in input data due to noisy or missing data.

**Uncertainties introduced during data processing:**

Feature extraction process may be introduced uncertainties. Some features used for power system event classification (as detailed in Section 3.4.3) require complex algorithms or classifiers. Sometimes these algorithms may output erroneous estimates or wrong class labels. A power system event classifier using these features should be able to handle situations where a subset of the features may be erroneous or unreliable.

#### *3.4.4.1 Tools for handling uncertainties*

There are different interpretations for uncertainties. For example, in probability theory, uncertainty refers to lack of certainty in determining the outcome of an event when more than one outcome is possible. In this research context, uncertainty refers to imprecision. Specifically, uncertainty refers to two types of imprecision [79]: 1. Quantitative imprecision associated with measurable data and 2. Linguistic imprecision associated with capturing expert knowledge. Both fuzzy logic and probability theory has been used to handle the above manifestations of uncertainty. Fuzzy logic uses membership degrees and possibility distributions to handle uncertainties [80], while probability theory uses probability distributions and subjective probability to handle uncertainties [81, 82].

Then the following question arises: which of the two methods, fuzzy logic or probability theory is better suited to for handling uncertainties? There are no clear answers; this has long been a subject of debate among the probability theory and fuzzy logic community. For example, Lindley [83] says:

*"Our thesis is simply stated: the only satisfactory description of uncertainty is prob-*

> *ability. By this is meant that every uncertainty statement must be in the form of a probability; that several uncertainties must be combined using the rules of probability; and that the calculus of probabilities is adequate to handle all situations involving uncertainty. In particular, alternative descriptions of uncertainty are unnecessary."*

Zadeh [84] formulated fuzzy logic and the theory of approximate reasoning. He proposed the idea of imprecise probabilities. In Zadeh's opinion, probability theory is not adequate to represent imprecision associated with human perception [85]. Zadeh contradicts Lindley by saying:

> *"It is a fundamental limitation to base probability theory on bivalent logic. Probability theory should be based on fuzzy logic."*

It is not the intention of this research to prove either fuzzy logic or probability theory is better that the alternatives at handling imprecision. Existing literature indicates both fuzzy logic and probability theory provide tools for handling uncertainty associated with imprecision. Hence, the following assumptions are made:

1. Both fuzzy logic and probability based methods can handle imprecision in capturing expert knowledge.
2. When implemented correctly, the results produced by the two methods will not differ significantly.

Then, the choice between using fuzzy logic or probability based methods can be based on other application specific factors, such as, ease of implementation, maintainability and scalability.

#### 3.4.4.2 *Fuzzy expert system classifier vs. probabilistic expert system*

The reason for choosing an expert system based classifier for this problem was explained in Section 3.4.1. The expert system based classifier should be able to deal with uncertainty, for reasons explained in the previous section. This leaves two options for designing a classifier; fuzzy expert system based classifier or a Bayesian network [78] based classifier. Both fuzzy logic and Bayesian network are capable of representing complex logical reasoning involved in human decision making. However, from an implementation point of view, Bayesian networks have some disadvantages.

Bayesian networks require conditional probabilities to be defined at every node in Bayesian network. An expert may form rules combining multiple variables using logical operators. In a Bayesian network, each of the logical operation will become a node. Hence the number of parameters that need to be defined will increase with the number of nodes. In comparison, fuzzy logic can capture expert knowledge directly in the form of rules. Logical operators used by an expert can be replaced by fuzzy operations. Fuzzy operators can be defined globally, and they will not require extra parameters for every operation (node). This is a significant advantage for fuzzy logic based exert system over Bayesian network. For example, the current rule base uses a few hundred logical operations. When using a Bayesian network, logical operations will translate to a few hundred nodes, each requiring conditional probabilities to be defined. This would become clear with an example. Consider the following sample ‘capacitor event rule’, that an expert may use to determine if an event was caused by a capacitor:

> “If a *step change* in reactive power (Q) is observed **and** the *change is large* **and** *no step change* in real power (P) is observed **or** a *step change* in real power (P) is

observed **and** the *change is very small* then the *event was caused by a capacitor*"

The above statement may be represented using a fuzzy rule as follows:

> "If (Q_SHAPE is *STEP* **AND** Q_CHANGE is *LARGE*) **AND** (P_SHAPE *NOT STEP* **OR** (P_SHAPE IS *STEP* **AND** P_CHANGE IS *VERY SMALL* )) then EVENT_CAUSE is CAPACITOR"

Figure 3.5 shows the same 'capacitor event rule' represented in the form of a causal graph that can be used for Bayesian inference. For comparison purposes, Figure 3.6 represents the 'capacitor event rule' as a tree using fuzzy operators. From the 'capacitor event rule', it is possible to infer that the expert needs the following input features to make a decision: shape observed in reactive power waveform (Q_SHAPE), change in reactive power (Q_CHANGE), shape observed in real power waveform (P_SHAPE), change in real power P_CHANGE. These features are inputs to both fuzzy rule and Bayesian network. Figures 3.5 and 3.6 show input nodes as ellipses with solid lines. For the fuzzy rule and the Bayesian network to be useful, all their required parameters need to be defined.

Parameters needed for sample fuzzy rule: In figure 3.6, no parameters are needed for the input nodes 'STEP P_SHAPE', 'NOT STEP P_SHAPE' and 'STEP Q_SHAPE'. These are crisp conditions. They return a 0 or 1 value depending on the input shape features P_SHAPE and Q_SHAPE having a shape label that is 'STEP'. The nodes that are highlighted (dashes) correspond to the logical connectives fuzzy conjunction (AND) and fuzzy disjunction (OR) operators. These operators can be defined once for the whole expert system and do not require any extra parameters at the node level. This leaves only two input more nodes to be defined, 'VERY SMALL P_CHANGE' and 'LARGE Q_CHANGE'. These two nodes test for an imprecise or 'Fuzzy' condition. These fuzzy nodes impose an elastic condi-

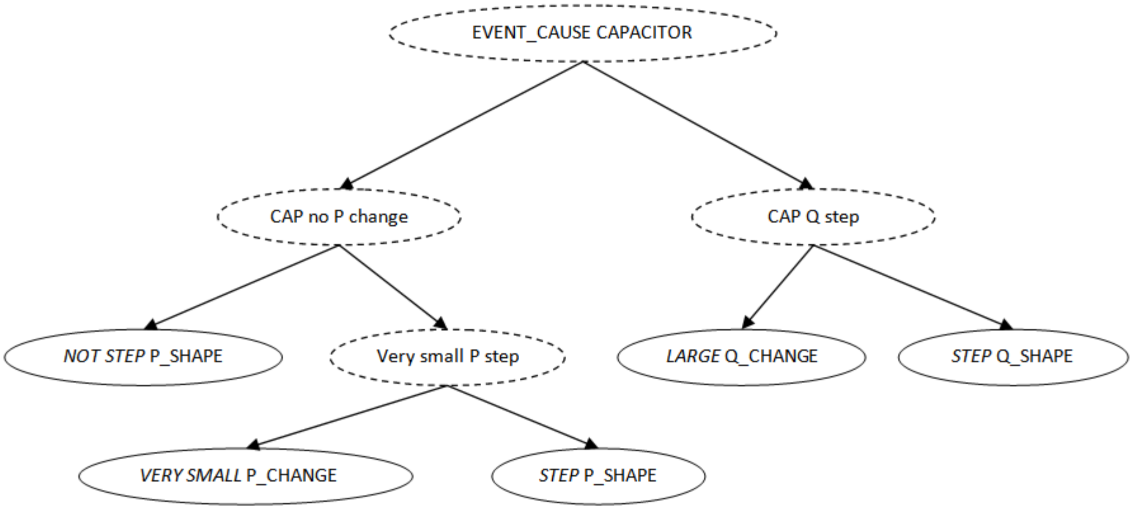


Figure 3.5: Representing expert knowledge as a Bayesian network

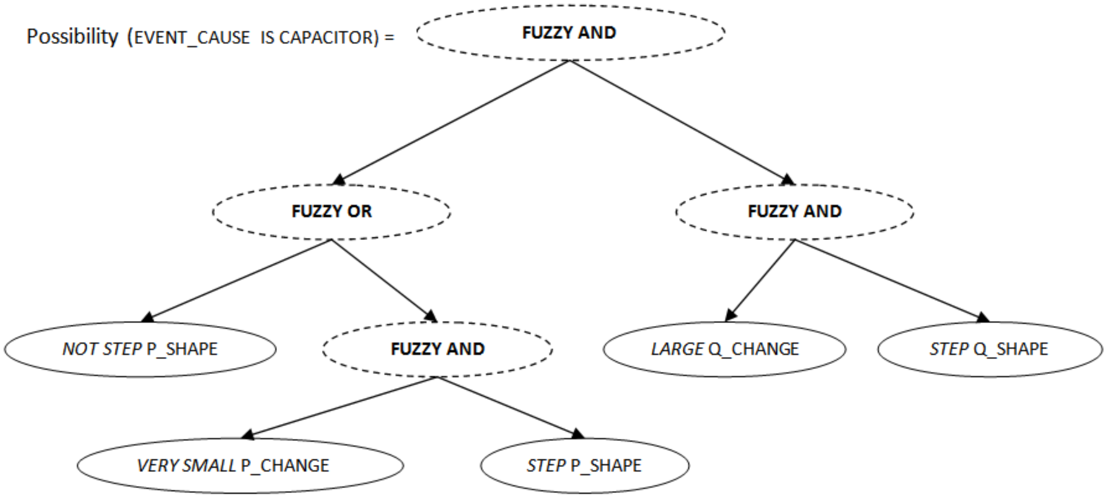


Figure 3.6: Representing fuzzy rule as tree

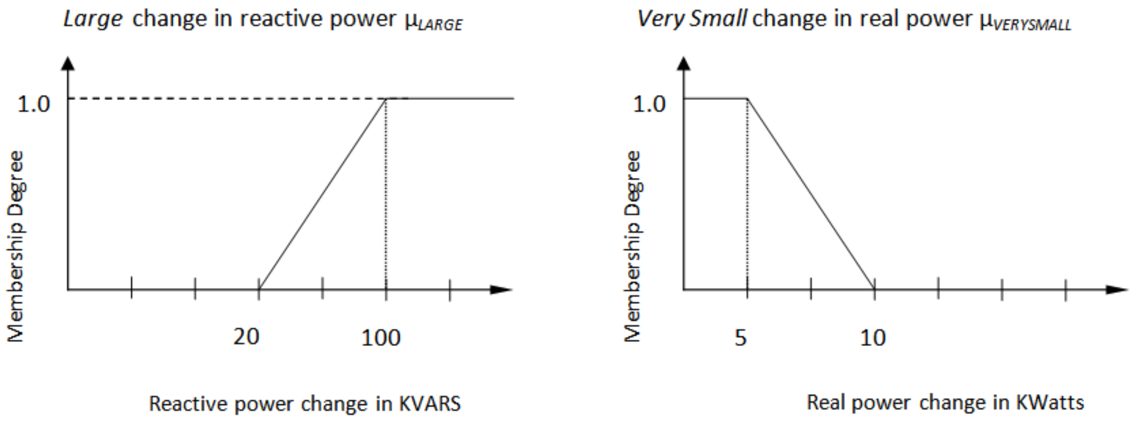


Figure 3.7: Membership functions $\mu_{LARGE}$ and $\mu_{VERYSMALL}$

tion and return a value in the continuous interval $[0, 1]$ representing the degree to which the condition was satisfied. For example 'LARGE Q_CHANGE' will return the degree to which change in Q belongs to the fuzzy set LARGE as defined by the membership function $\mu_{LARGE}$ shown in Figure 3.7 . Similarly, 'VERY SMALL P_CHANGE' will return the degree to which change in Q belongs to the fuzzy set LARGE as defined by the membership function $\mu_{VERYSMALL}$ shown in Figure 3.7. Both the membership functions $\mu_{LARGE}$ and $\mu_{VERYSMALL}$ need only two parameters to be defined. Hence, a total of four parameters needs to be defined for the fuzzy rule implementation.

Parameters needed for sample Bayesian network: In figure 3.5, every node that has a parent node needs conditional probability to be defined. To keep the analysis simple, all nodes used in the example are discrete, and take either *yes* or *no* value. Then each node except the node 'EVENT_CAUSE CAPACITOR' will need four conditional probability values probability values to be defined (number of parent values times number of child values). The 'EVENT_CAUSE CAPACITOR' node will need just two prior probabilities. This results in a total of $(2 + 8 * 4) = 34$ probability values. The number of parameters will increase further if the nodes with imprecise conditions 'VERY SMALL P_CHANGE' and 'LARGE Q_CHANGE' were made to have a continuous distribution.

The above example clearly shows that assigning conditional probability values to Bayesian network nodes is a major drawback of using Bayesian networks. Using Bayesian network would become more difficult when the rules become more complex. There are other disadvantages of using Bayesian networks for power system event classification. Many of the input features to the power system event classifier are continuous. Bayesian networks typically use discrete input variables. When using continuous variables in Bayesian networks, defining conditional probabilities and inferring output probabilities can be challenging [86]. Fuzzy logic on the other hand, can easily deal with both continuous and discrete variables without additional implementation issues.

From a maintainability and scalability point of view, Bayesian networks are at a disadvantage. Fuzzy rule base can be modified easily by either adding new rules or changing existing rules. New parameters will be needed only if new input or output variables are defined. However, in Bayesian networks, new rules or modifying existing rules may require changes to conditional probabilities at all nodes that are affected by the changes. In a large system, manually adding or modifying conditional probabilities at every node can be error prone. Bayesian networks have been used to solve a number of real world problems [87]. However, for the power system event classification problem, a fuzzy logic based expert system seems to be a good and better choice.

#### *3.4.4.3 Overview of fuzzy logic and possibility theory*

Humans reason effectively with fuzzy definitions. In order to capture imprecise and vague information, the theory of approximate reasoning: fuzzy logic was conceived by Zadeh [84]. Fuzzy logic allows Imprecise definitions to be formulated mathematically and processed by computers using fuzzy sets. This was an attempt to apply a more human like way of thinking, while programming computers.

Classical set theory uses crisp sets. An element can either belong to a set or not and there cannot be anything in between. Similarly, binary logic uses either true or false (zero or one) there cannot be values in between. Let $\mu_A\left(x\right)$ represent a function that tests the condition 'x is in A' or 'is x an element of A'. For a crisp $\mu_A\left(x\right) = 0$ if $x \notin A$ and $\mu_A\left(x\right) = 1$ if $x \in A$. It can be seen that crisp sets use binary two-valued logic. The truth in the statement 'x is an element of A' can either be one (true) or zero (false).

Fuzzy set theory is in strong contrast with classical set theory and binary logic. Fuzzy sets can allow partial membership or 'degree of membership'. A test for membership can return a value in the continuous interval $[0, 1]$, while crisp sets can return binary values $\{0, 1\}$ only. For example, if $B$ is a fuzzy set, then a test 'is x an element of B' may return a value in the continuous interval $[0, 1]$ defined by the fuzzy membership function $\mu_B\left(x\right)$.

The theory of approximate reasoning, as formulated by Zadeh, can be viewed as an application of possibility theory [88]. It is essentially a methodology for representing some available information (especially when it is vague and incomplete) in terms of possibility distributions and for deducing from this information what can be said about the values of variables of interest [23]. Dubois defines possibility distribution as:

> "Let $U$ be the set that represents the range of a variable $x$. Usually $x$ stands for the unknown value taken by some single-valued attribute applied to an

> object under consideration. For instance $x$ refers to the age of a man named Peter. A possibility distribution $\pi_x$ on $U$ is a mapping from $U$ to the unit interval$[0, 1]$ attached to the single-valued variable $x$. The function $\pi_x$ represents a flexible restriction which constraints the possible values of $x$ according to the available information, with the following conventions: $\pi_x(u) = 0$ means that $x = u$ is definitely not possible, $\pi_x(u) = 1$ means that absolutely nothing prevents that $x = u$"

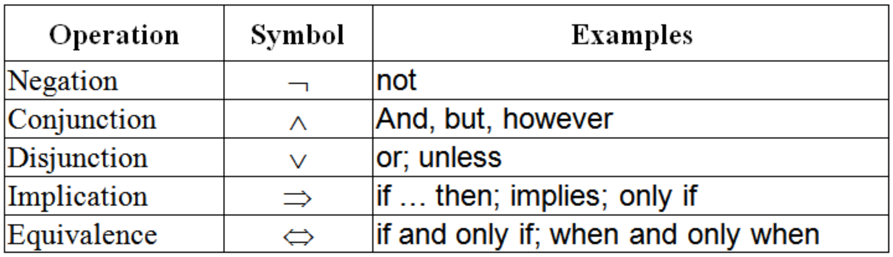
Thus, fuzzy logic generalizes the binary distinction between possible vs. impossible to a matter of degree called the possibility [89]. For instance, there would be some uncertainty in determining if a step change in reactive power Q wereq caused by a capacitor switching. Expert knowledge requires the change in Q ($\Delta Q$) to be 'Large' for it to be caused by a capacitor. Let the fuzzy set *LARGE* describe the 'largeness' of $\Delta Q$ in the context of capacitor switching being the cause. Let $\mu_{LARGE}\left(x\right)$ (Figure 3.7) represent the membership function associated with the fuzzy set *LARGE*. Then, by assigning the fuzzy set *LARGE* to $\Delta Q$ (Equation 3.28), the possibility distribution defining the possibility degree of $\Delta Q$ is obtained.

$$\pi_{\Delta Q}(x) = \mu_{LARGE}(x), \ x \in \Delta Q \tag{3.28}$$

In the assignment $\pi_{\Delta Q}(x) = \mu_{LARGE}(x)$, $\pi_{\Delta Q}\left(x\right)$ denotes the possibility distribution of $\Delta Q$ values for 'largeness'. $x$ is a variable that represents the magnitude of change in reactive power. For example, for a large $\Delta Q$ caused by capacitor switching ($\mu_{LARGE}$ in the Figure 3.7), it is 50% possible for $\Delta Q$ value to be 60 kVARS, it is impossible for $\Delta Q$ values to be less than 20 kVARS, and it is definitely possible for $\Delta Q$ values to greater than 100 kVARS.

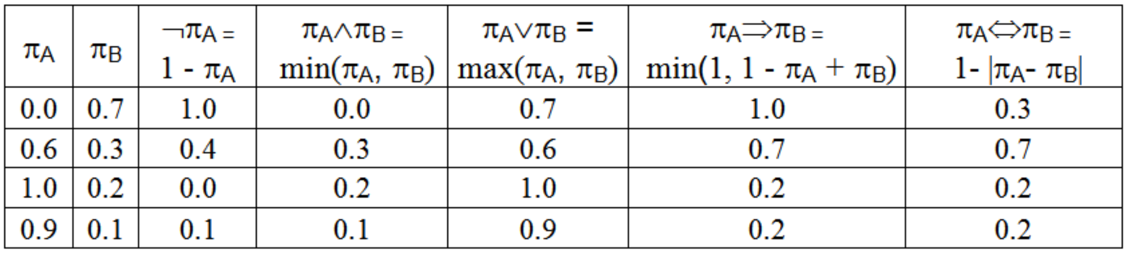
The use of fuzzy membership functions to describe possibility distributions

Table 3.2: Fuzzy Set Operators



| Operation | Symbol | Examples |
|---|---|---|
| Negation | $\neg$ | not |
| Conjunction | $\wedge$ | And, but, however |
| Disjunction | $\vee$ | or; unless |
| Implication | $\Rightarrow$ | if … then; implies; only if |
| Equivalence | $\Leftrightarrow$ | if and only if; when and only when |

Table 3.3: Numerical examples of fuzzy set operations on possibility distributions



| $\pi_A$ | $\pi_B$ | $\neg\pi_A =$ <br> $1 - \pi_A$ | $\pi_A \wedge \pi_B =$ <br> $\min(\pi_A, \pi_B)$ | $\pi_A \vee \pi_B =$ <br> $\max(\pi_A, \pi_B)$ | $\pi_A \Rightarrow \pi_B =$ <br> $\min(1, 1 - \pi_A + \pi_B)$ | $\pi_A \Leftrightarrow \pi_B =$ <br> $1 - \lvert\pi_A - \pi_B\rvert$ |
|---|---|---|---|---|---|---|
| 0.0 | 0.7 | 1.0 | 0.0 | 0.7 | 1.0 | 0.3 |
| 0.6 | 0.3 | 0.4 | 0.3 | 0.6 | 0.7 | 0.7 |
| 1.0 | 0.2 | 0.0 | 0.2 | 1.0 | 0.2 | 0.2 |
| 0.9 | 0.1 | 0.1 | 0.1 | 0.9 | 0.2 | 0.2 |

allows carrying out fuzzy set operations on the possibility distributions. This is helpful in representing complex propositions as a combination of fuzzy operators and possibility distribution. Table 3.2 describes a list of commonly used fuzzy set operations, and Table 3.3 gives numerical examples of fuzzy set operations on possibility distributions.

In the context of power distribution system event classification, possibility distributions can be used to describe uncertainties due to imprecision associated with capturing expert knowledge and imprecision associated with extracted features. A fuzzy expert system based classifier can process features extracted from waveform data using fuzzy rules; compute possibility values for different event causes; and assign class labels based on event cause possibility values. Designing such a classifier can be considered as one of the sub-problems involved in designing a power system event classification algorithm. Now, the problem can be stated as: Design a fuzzy logic based feature classifier $h_{Fuzzy}$ as a part of the algorithm that minimizes total perceived cost of misclassification (Equation 3.8), where:

$$h_{Fuzzy}(x_{Fea}) = \hat{y} \tag{3.29}$$

In the above equation, $x_{Fea}$ represent extracted features (Equation 3.12) and $\hat{y}$ represents output labels.

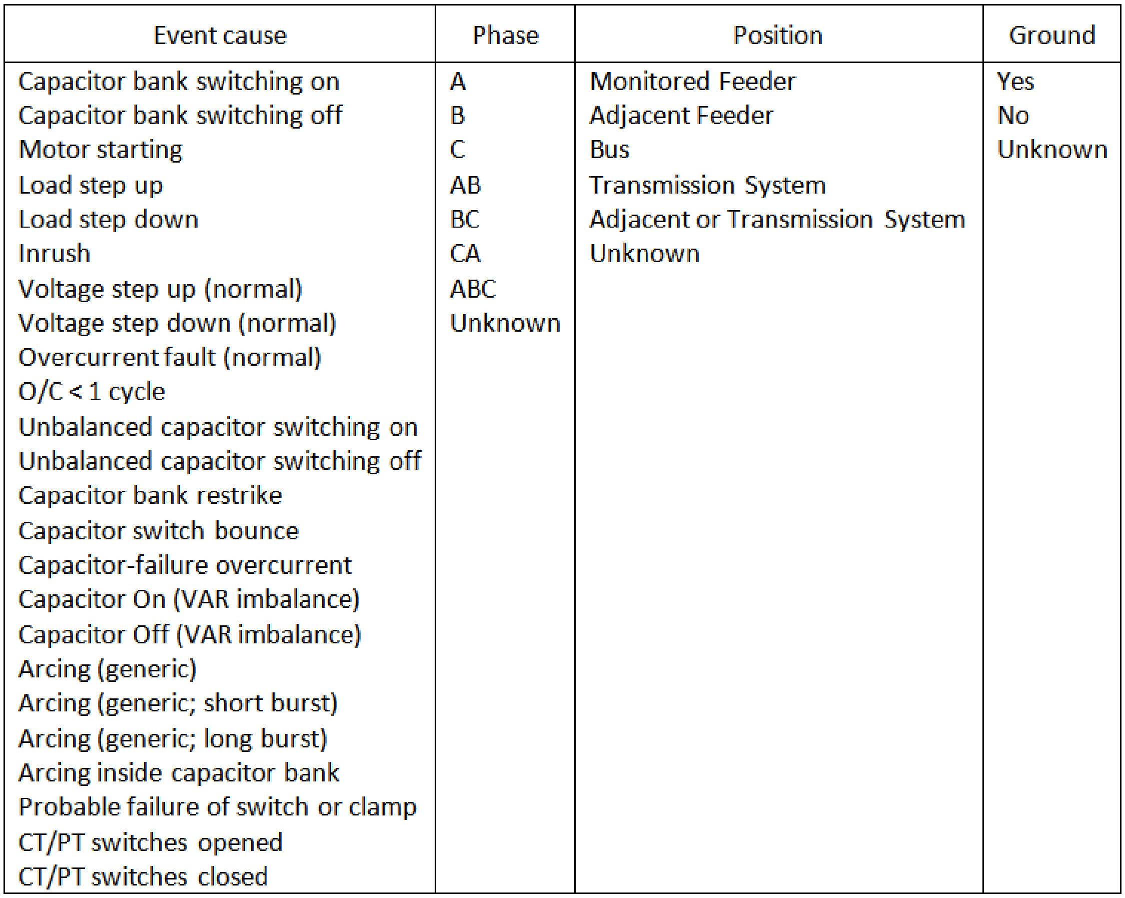
### 3.4.5 Power System Event Classification - A Large Scale Classification Problem

One of the reasons for choosing expert system based classifier for power system event classification is because of its large scale. The dimensions of both input feature space and output space are large. Feature extractors process a total of 36 signals (Equations 3.2, 3.3) and extract about 1000. To better aid in locating problematic components, the output classifications have four dimensions. 1) Event type, 2) Event phase, 3) Event Position, and 4) Ground involvement (Table 3.4).

Each of these dimensions can take say $N_e$, $N_{ph}$, $N_{pos}$ and $N_g$ discrete values. Then the total number of possible classifications resulting from the combination of these four dimensions is $N_e \times N_{ph} \times N_{pos} \times N_g$. Event types $N_e$ is currently 24, and increasing. As new event causes are identified, this number also increases. Event phase $N_{ph} = 8$, represent the different combinations of phases that could be involved during an event. Event position $N_{pos} = 6$ represents the location of the event relative to the feeder being monitored. Ground involvement $N_{ph} = 4$ represents if the event involved the neutral. Now $N_e \times N_{ph} \times N_{pos} \times N_g = 24 \times 8 \times 6 \times 3 = 3456$, is the total number of possible distinct classifications. This number will grow as $N_e$ increases. Designing a classifier that uses 1000 features to assign a possible combination of 3456 output labels is clearly a large scale classification task.

If a supervised learning technique were to be used instead of an expert sys-

Table 3.4: Output classification attributes and values



| Event cause | Phase | Position | Ground |
|---|---|---|---|
| Capacitor bank switching on | A | Monitored Feeder | Yes |
| Capacitor bank switching off | B | Adjacent Feeder | No |
| Motor starting | C | Bus | Unknown |
| Load step up | AB | Transmission System | |
| Load step down | BC | Adjacent or Transmission System | |
| Inrush | CA | Unknown | |
| Voltage step up (normal) | ABC | | |
| Voltage step down (normal) | Unknown | | |
| Overcurrent fault (normal) | | | |
| O/C < 1 cycle | | | |
| Unbalanced capacitor switching on | | | |
| Unbalanced capacitor switching off | | | |
| Capacitor bank restrike | | | |
| Capacitor switch bounce | | | |
| Capacitor-failure overcurrent | | | |
| Capacitor On (VAR imbalance) | | | |
| Capacitor Off (VAR imbalance) | | | |
| Arcing (generic) | | | |
| Arcing (generic; short burst) | | | |
| Arcing (generic; long burst) | | | |
| Arcing inside capacitor bank | | | |
| Probable failure of switch or clamp | | | |
| CT/PT switches opened | | | |
| CT/PT switches closed | | | |

tem based classifier, this should be modeled as a large scale optimization problem. Then there are several algorithms and optimization toolboxes available for solving large scale optimization problems. Some of the optimization methods available for linear programming (LP) are Simplex, LIPSOL (Matlab), CPLEX (C based tool box that uses Simplex and interior point methods). Some popular methods available for large scale non linear programming (NLP) problems are LOQO (based on interior point method), KINTRO (trust region algorithm), SNOPT (quasi Newton algorithm) [90] . Most of these techniques involve minimizing a linear or non-linear cost function subject to linear or nonlinear equality or inequality constraints using some variant of gradient descent techniques. A hierarchical rule based fuzzy expert system classifier is preferable over using large scale parametric optimization (black-box approach) for the following reasons:

#### *3.4.5.1 Ease of problem formulation:*

When using the LP or NLP approaches, the multi-class, multi-variable classification problem has to be formulated as a linear or non-linear constrained optimization problem. In the case of LP problem, this will reduce to a 'feasibility' problem that checks for the existence of hyper-planes that partitions the input feature space, but this approach will work only if the features are linearly separable. The non-linear nature of power system event classification problem preempts the usage of LP techniques. There is a different issue when trying to use NLP techniques for the classification problem. NLP techniques try to find optimal non-linear hyper-surfaces that can partition the input feature space. NLP techniques need a parametric model of the hyper-surface. However, no prior information regarding the nature of this hyper-surface is known except for its dimensionality. For example if a polynomial function is used to approximate this hyper-surface, a priori knowledge of the order of this polynomial is not available. If the order is too small, then it may not have a feasible solution. If an arbitrarily large order is chosen, then the number of parameters to be optimized will also increase, and may lead to over fitting. This situation is analogues to the problem of choosing the right number of hidden layer neurons when designing a multi-layered-perceptron. Such NLP approaches are better suited for optimizing the parameters of a system for which a model already exists, and are less suited for this classification problem, where the objective is to discover the mapping. In a rule based approach, there is a clear understanding of the relationship between the features and classes. Hence, formulating this relationship in the form of rules or decision trees (if using tree based classifiers) is much easier, when compared to the approach of trying to discover a model automatically.

#### *3.4.5.2 Ability to use features of enumerated type*

Most LP or NLP algorithms work well when trying to optimize functions that are continuous and the feature space is also continuous. When some of these features take discrete values, the classification problem becomes an integer programming problems and gradient descent techniques have o be combined with enumeration approach to tackle the discontinuities in the search space. Integer programming problem is far less tractable and convergence times may become far too large to be practical. The power system event classification problem has a number of discrete features. For example shapes features like ‘Dip’, ‘Step’ take discrete enumerated values. Rule based approach can easily handle both discrete and continuous features.

#### *3.4.5.3 Scalability*

A number of performance parameters related to a LP or NLP problem such as convergence time, processing power and memory requirements are directly related to the number of variables involved in the problem. The number of variables involved will be proportional (for LP) or a function of (dimension of feature space + number of classes + size of the training set). If one or more of these quantities increase, it may have an adverse effect on the performance parameters of the LP or NLP algorithm. For example, the Simplex (LP) algorithm has a worst case exponential time convergence, i.e., the time taken to converge to the optimum solution may increase exponentially with the number of variables. A number of NLP algorithms promise only a polynomial convergence, and no promises on the polynomial coefficients are made. The rule based classifier does not suffer from such issues. Addition of new features and classes can be made by adding more rules. Training sets and convergence time are not applicable to the rule based approach.

#### *3.4.5.4 Insight and maintenance*

In LP and NLP 'black box' approach, it is not easy to gain insight into the nature of decision boundaries and relationship between the features and classes. Some indirect methods such as, the branch and bound procedure, can be used for selecting important features. In a rule based approach, this relationship is evident. Further insight may not be necessary. From the point of view of maintenance, if faced with a problem of false positives or false negatives when the 'trained' algorithm is used in the field, there are not many options available in the case of LP or NLP approaches to fix the issue. A re-training maybe required with the training set that includes the problematic cases or, penalty parameters may be introduced to discourage misclassification. In the case of a rule based system, it is more transparent. The cause of misclassification may be tracked down to one or more rules. These rules may be debugged and modified as needed.

#### *3.4.5.5 Reducing dimensionality and incorporating structure*

Expert knowledge available about the relationship between the features and individual event types (classes) can easily be used in a rule based system to partition the feature space intuitively. I.e., not all the features are required for making a decision on any given event type. Unlike a 'black box' approach where the relationship between the features and event types needs to be learned through training sets, this relationship can directly be coded in the form of rules in a rule based system.

The proposed hierarchical classification system has multiple classification modules, each responsible for classifying certain categories of events. For example, classification modules for classifying capacitor related events, overcurrent related events, arcing related events, motor related events, load switching related events, current transformer and potential transformer relate events. Each of these classifiers is responsible for classifying a subset of event types. For example, the capacitor event classifier will be responsible for classifying only normal capacitor operations (switch on/off) and problems related to capacitor switching. Problems related to capacitor switching include, unbalanced capacitor switching (when one phase in a three phase capacitor bank fails to switch), capacitor switch bounce, etc. The capacitor classification module will not need all the 1000 features for its classification. It will only use a subset of these features that are needed for capacitor related event identification. The different classification modules may share some features, but the number of features each of the classification modules use is reduced from the total 1000 to a subset of may be a 100 features. These numbers are provided for illustrative purposes only. The actual number of features used may be different. Since each classification module is using only a subset of the total features, the dimensionality of the feature space used by each classification module would be reduced. This will help reduce (but not eliminate) the possibility of overlap of the event subspaces.

Structural knowledge about three phase nature of the power system also helps to simplify partitioning of the feature space related to some event types. For example, consider the case of classifying an overcurrent event. An overcurrent can involve one or more phases (phases A, B and C). For classification purposes, the three phases may be processed independent of each other. Within an overcurrent classification module, the overcurrent classifier may process features of each phase separately. A decision on whether or not that phase was involved in an overcurrent can be made independent of other phases. This lower level classifier (phase level classifier) uses approximately one third of the 100 features used by the higher level overcurrent classification module. The dimensionality of the feature space used by the lower level classifier has further reduced and hence also further reduced the possibility of an overlap of event space. A similar argument may also be made for capacitors, motors and arcing classification modules.

From the above discussion, it is evident that the hierarchical design of the classification system (which is explained in Chapter 6) and expert knowledge about the nature of the events contributes to simplifying the analysis and tackling problems associated with the high dimensionality of the input feature space. The problem of feature selection is also made easier because it is a rule based classifier, and the expert's prior experience in using features for manual classification is used to select the right features for the classifier.

Thus, in a hierarchical rule based classifier, 'sub' classifiers may analyze a subset of the total number of features. High dimensional feature space may be broken in to multiple lesser dimensional feature sub-spaces. This will simplify decision boundaries. In the LP or NLP approaches, it may not be possible to incorporate such structure to simplify the problem.

One of the sub-problems in designing a power system event classification algorithm, is to design a hierarchical fuzzy logic based feature classifier $h_{Fuzzy}$ as part a of the algorithm that minimizes total perceived cost of misclassification (Equation 3.8), where:

$$h_{Fuzzy}(x_{Fea}) = \hat{y} \tag{3.30}$$

In the above equation, $x_{Fea}$ represent extracted features (Equation 3.12) and $\hat{y}$ represents output labels. Next section presents the proposed overall classification scheme for power system event classification. The hierarchical fuzzy logic based classifier is a part of this scheme.

### *3.4.6 Classification System Overall Scheme*

Block diagram shown in Figure 3.8 illustrates various components of the proposed fuzzy logic based power distribution system event classification algorithm. For simplicity, this algorithm will here on be referred to as fuzzy logic based classification algorithm (FLCA). Data flow through the classification algorithm can be broken down into three stages: 1. Peprocessing stage, 2. Feature extraction stage and 3. Classification stage. These three stages are typical of most classification algorithms. What differentiates FLCA, and makes this research work novel, is how the three stages were designed to solve the power distribution system event classification problem.

DFA field units insert new waveform records into the database whenever they

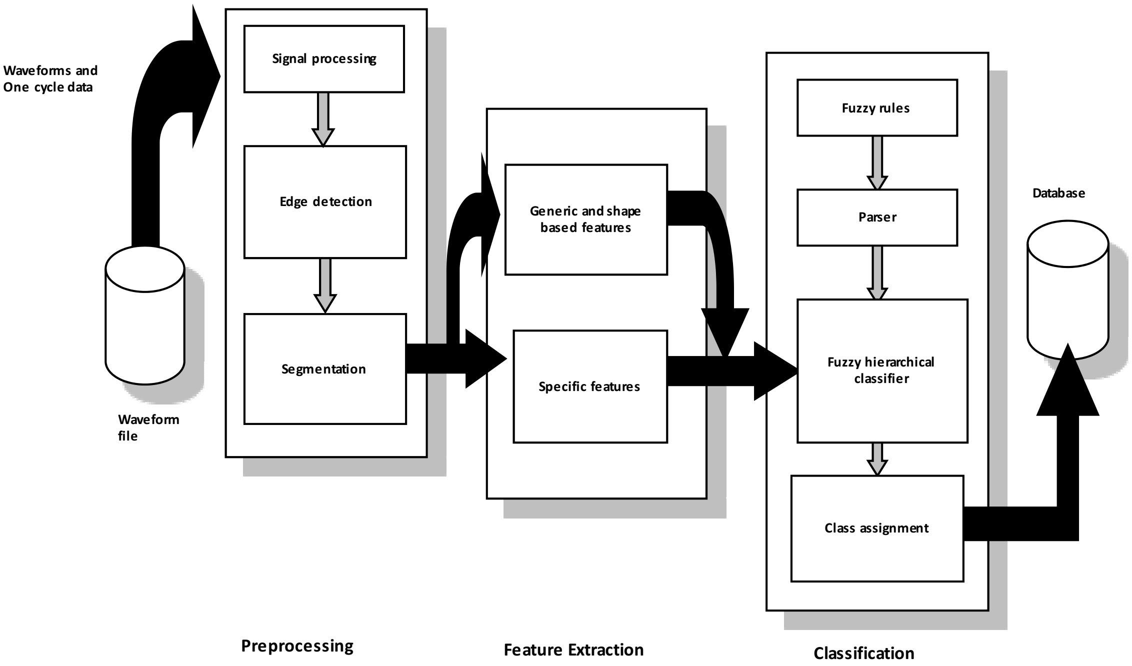


Figure 3.8: Block diagram of classification system

trigger new waveform files corresponding to power stem events. The FLCA is an online classification system. It frequently polls waveform records database for new waveform records. When a new waveform file is available, FLCA starts the classification process. First it deserializes the waveform files and caches all waveform data for further processing.

Figure 3.9 shows some of the waveforms recorded during a normal motor start. Going from left to right and top to bottom, Figure 3.9 shows three RMS currents, Sampled currents, RMS voltage, sampled voltage, Real power and Reactive power. Figure 3.9 also shows the multi-phase nature of a three phase motor start event. The plots show observable changes and distinct shapes on multiple phases. A human expert will be able to look at these plots and infer that the waveforms correspond to a three phase motor start. The expert infers based on studying relative behavior of three phase signals and shapes observed in these signals. FLCA

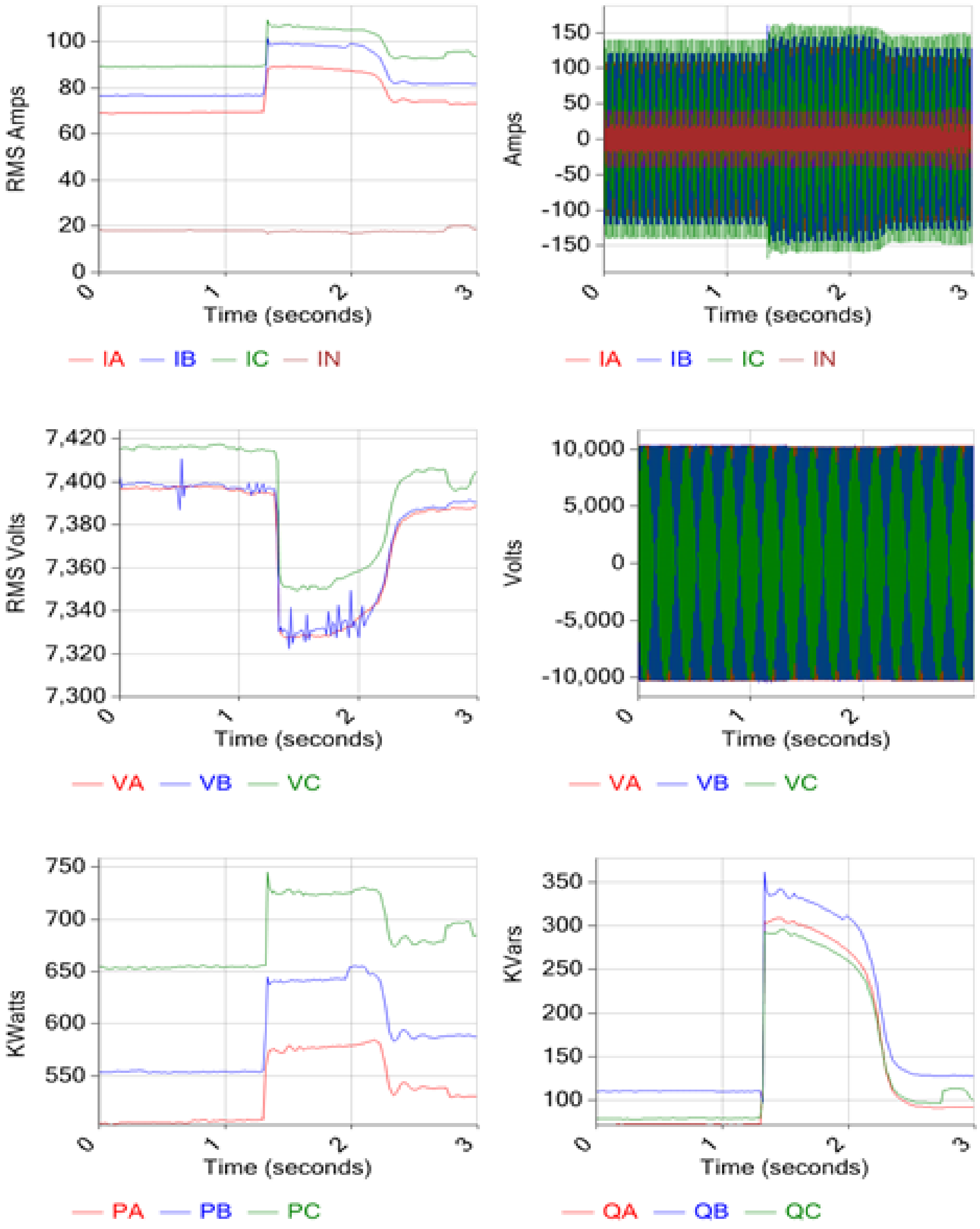


Figure 3.9: Waveforms recorded during a motor start event

tries to emulate the decision making process of a human expert. It handles three phase data at every stage of the classification process.

#### *3.4.6.1 Preprocessing stage*

In the preprocessing stage, FLCA converts the waveform data in a form suitable for further processing in feature extraction and classification stages. FLCA first does signal processing operations on waveform data. These signal processing operations are:

1. Detecting and substituting missing signals with estimated signals if possible.

2. Computing phasor differenced signals by removing steady state load components from high speed current and voltage signals.

3. Normalizing to remove the effect of sensor configurations.

After the above signal processing step, FLCA optionally does edge detection and segmentation. FLCA runs in two passes. In the first pass, FLCA assumes all the data in a waveform file is from a single power stem event. If FLCA is not able to assign classification in the first pass, FLCA uses segmentation break the event data into possible sub-events in the second pass.

To segment event data, a context based segmentation approach is used. Edges are detected using first order and second order difference on RMS waveforms. Then, RMS shape detection is used to check for valid shapes between edges. If no shapes are detected, then the edges are rejected. If valid shapes are detected, then the edge is used to create a segment.

FLCA extracts features for each segment and the inference engine processes each segment. If the segments are related, the inference engine pieces together

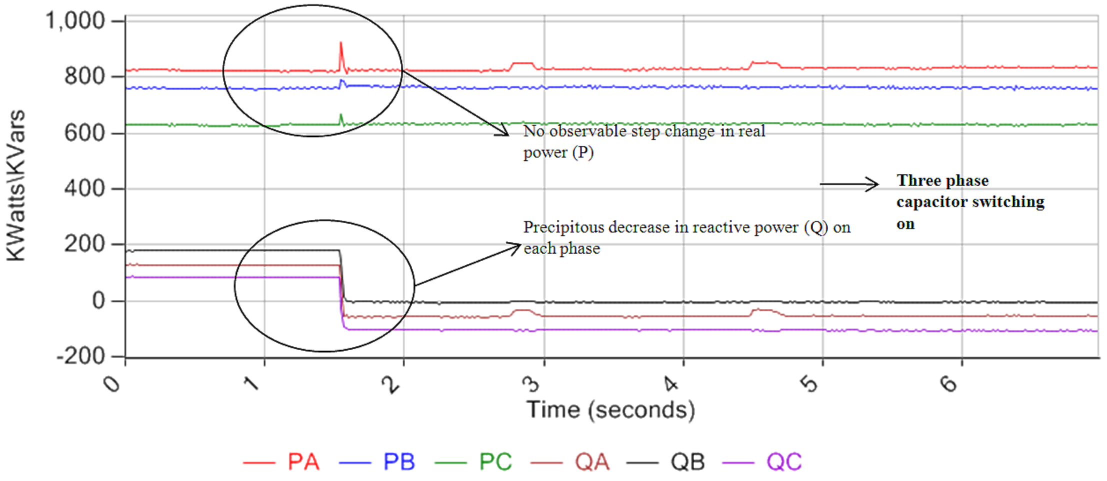


Figure 3.10: Real and reactive power waveforms during a capacitor switching event

information from each segment to assign a final classification. If the inference engine determines that the segments were from unrelated events, then it treats each segment independently and employs a conflict resolution strategy to assign a single classification to the whole waveform file. This two step process is needed because, a waveform file may contain data from one or more related or unrelated sub-events. However, a majority of waveform files contain data from a single event only. This is because, field units are typically configured to record waveform files only a few seconds long. The possibility of recording more than one unrelated event within a few seconds is low.

#### *3.4.6.2 Feature extraction stage*

A human expert visually scans waveform data for characteristic signatures (features) associated with different power stem events. Based on observed features, the expert associates the event with an underlying cause and then assigns a classification label based on the underlying cause.

For example, Figure 3.10 shows three phase real and reactive power waveforms observed during a capacitor switching on event. A human expert would observe that the reactive power decreased by about 200 kVARS on all phases while the real power did not show any step changes. Capacitors are typically used for VAR support and a capacitor switching on causes a decrease in the reactive power. Based on this knowledge, the expert would conclude that the event was caused by a capacitor switching on. For an automatic classification system such as FLCA, using the correct features is critical for classifier performance. Feature selection is an art. Choice of features for FLCA is the result of the knowledge gained by experts analyzing thousands of waveform files.

Features used by FLCA fall into three categories:

1. Generic features: These are features FLCA extracts from signals in a segment, irrespective of the type of signal. These features may provide some statistical information about the signal such as maximum, minimum, mean value. These features correspond to signal levels a human expert would look for while analyzing event waveforms. For example, an expert would look for maximum values of three phase RMS current waveforms in comparison to load current to determine if a waveform corresponds to an overcurrent event (Figure 3.11).

2. Shape based features: These are features FLCA extracts to represent shapes observed in RMS waveforms. In a majority of cases, a human expert would be able to determine the cause of a power system event based on the shapes observed in RMS waveforms. For example, consider the three phase RMS current plot (top left) in Figure 3.9. This is a characteristic shape associated with a motor start which an expert would be able to recognize immediately.

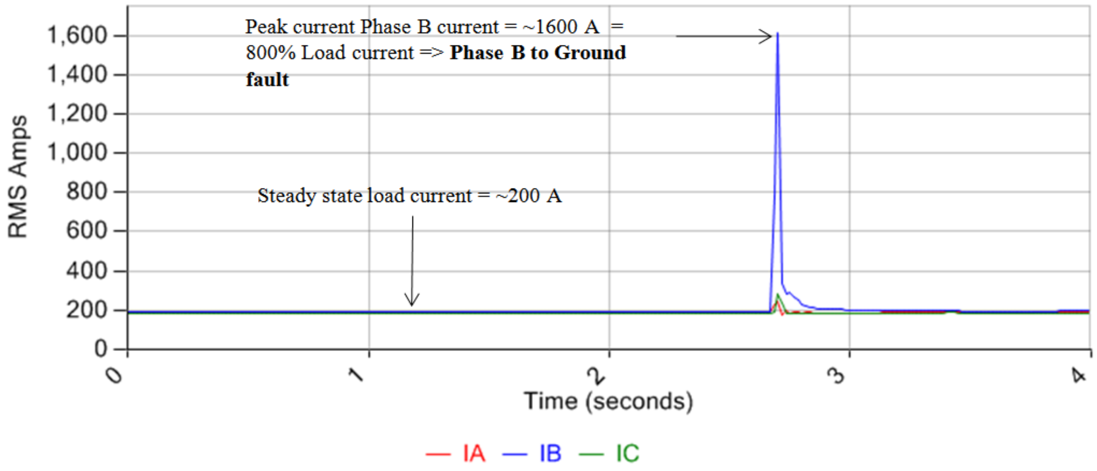


Figure 3.11: Using signal levels to determine event type

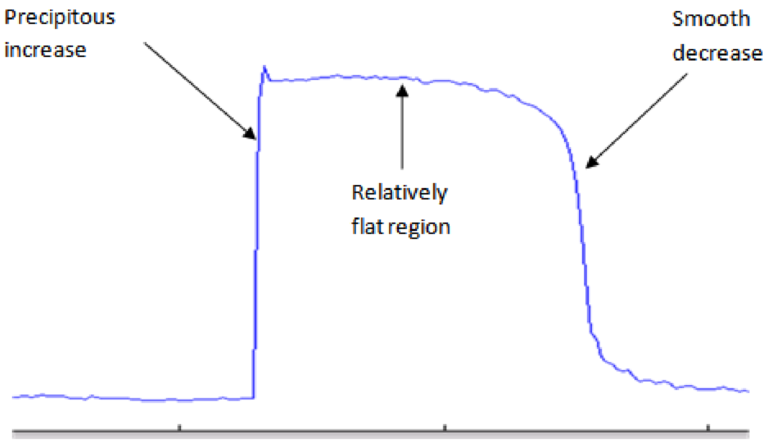


Figure 3.12: Motor start characteristic shape

This shape can be defined by a precipitous increase in values followed by a relatively flat region followed by a smooth decrease in values (Figure 3.12). FLCA uses a modified version of Dynamic Time Warping algorithm [91], specifically designed for recognizing shapes in RMS signals. Chapter 4 explains generic feature extraction and shape based feature extraction in more detail.

3. Event specific feature: These are features required to ascertain if an event was caused by a specific power system activity, such as, arcing or capacitor switching. There are some cases where a human expert may not be able to recognize the cause of an event just by looking at RMS waveform levels and shapes. The expert may have to analyze high speed waveforms further, to classify the event. For example, the top most plot in Figure 3.13 shows RMS voltage waveforms caused by an arcing capacitor switch. An expert analyzing the RMS voltage waveforms alone would not be able to tell, conclusively, they were caused by an arcing capacitor switch. This is because local variations, as seen in the RMS voltage plots, can be caused by a number of factors such as noise or other forms of arcing (not necessarily an arcing capacitor switch). An expert may look at phasor differenced voltage waveforms to clearly see what is causing the activity. Phasor differenced voltage waveforms are calculated by subtracting an estimated ambient (steady state) voltage phasor from the voltage waveform. This would show transient activity in the signal. The second plot in Figure 3.13 shows phasor differenced high speed voltage waveforms corresponding to the same event. Phasor differenced voltage signal clearly shows large voltage transients with peaks up to 6000V. Further zooming in (the third plot in Figure 3.13) the ex-

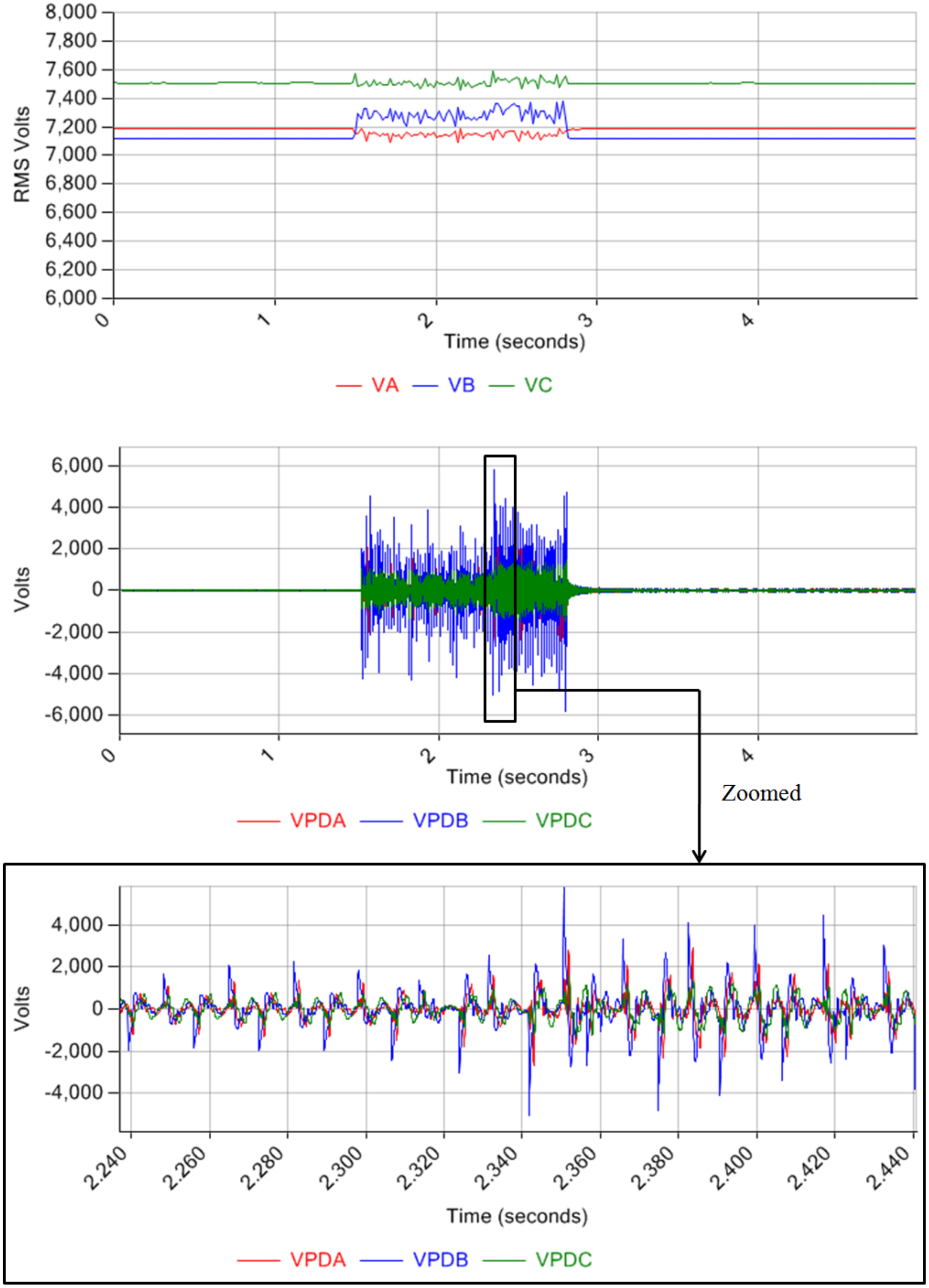


Figure 3.13: RMS and high speed voltage waveforms from an capacitor switch arcing event

pert would be able to verify high frequency voltage transients. The expert would also have to look at other features before diagnosing the cause of the event as possibly due to capacitor switch arcing. These features include, high frequency activity in current waveforms and the relative magnitudes of voltage transients on each phase This example illustrates that some power system event types need specialized features to determine the cause of the event. There are other situations where an expert may easily recognize an event, but would have to analyze further, to know more about the nature of an event. For example, Figure 3.14 shows RMS current waveforms from a multi-shot overcurrent event. An expert looking at these waveforms would be able to recognize that this is a Phase B to ground fault with a fault magnitude of about 450A. However, if the expert analyzes the waveform further, the expert would be able to provide more information that could help locating the fault quickly. On further analysis, an expert would be able to tell that the fault was likely beyond a single phase recloser. The expert could also tell that the recloser likely operated three times; twice on a fast curve and once on a slow curve, before going to a lockout causing an outage (service interruption). If the expert provides utility personnel with this information, then they could narrow down the search to locations down stream of Phase A single phase recloser that is set to operate three times. FLCA uses the following list of algorithms to extract event specific features:

(a) Transient detection algorithm for identifying high frequency transients associated with capacitor switching, capacitor restrike and capacitor arcing.

(b) Zero crossing based waveform analysis algorithm for detecting arcing.

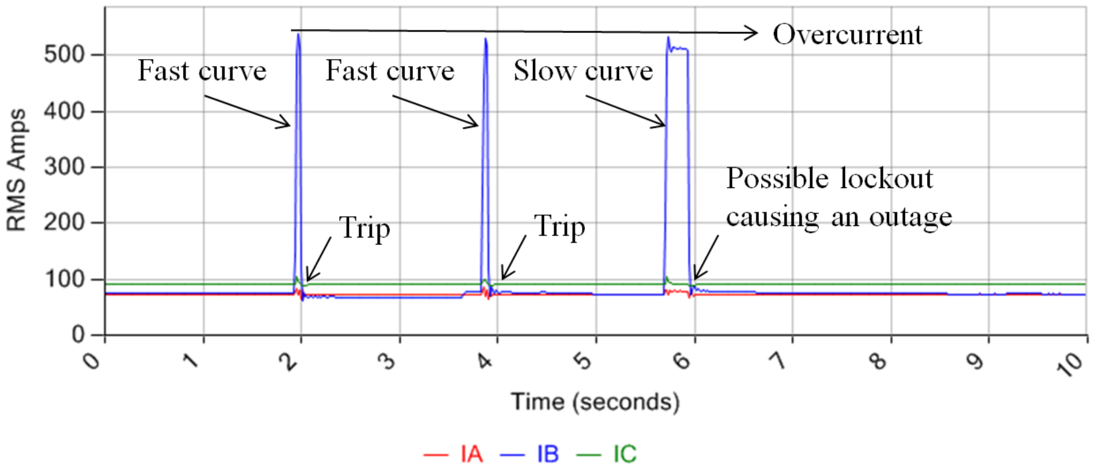


Figure 3.14: RMS current waveforms from multi-shot overcurrent event

(c) Multi-shot overcurrent analysis algorithm.

Chapter 5 explains each of the above algorithms in detail.

#### *3.4.6.3 Classification stage*

Features extracted from waveform data serve as input to a fuzzy classifier. The fuzzy rule based classifier is the final stage of FLCA. Features are the 'evidence' based on which the classifier assigns a category to the waveform data being processed. This is analogous to an expert making diagnosis based on evidence present in the signals and the expert's prior knowledge. Fuzzy logic allows mimicking human approximate reasoning through the use of linguistic rules. FLCA uses fuzzy rules and fuzzy membership functions to capture expert's knowledge. FLCA uses a fuzzy hierarchical classifier to assign possibility values for various power system event categories. The fuzzy hierarchical classifier was specifically designed for handling power system event features. The hierarchical classifier processes features and possibility values on per phase basis then at a segment level (when waveform data can be split into multiple segments) and finally at a global level considering the whole waveform file. The hierarchical classifier computes possibility values for each power system event category after processing the features. Finally, FLCA assigns a class label based on these possibility values. Chapter 6 explains the fuzzy hierarchical classifier in detail. FLCA writes the output classifications and related attributes to database. Reporting algorithms process these classifications and attributes and present them to utility personnel.

#### *3.4.6.4 Intelligent reporting to prevent information overload*

Automatic classification scheme such as the one outlined in the previous section is essential for intelligent monitoring and diagnosing problems on feeders being monitored. However, for it to be useful, the classification system cannot reports every event it classified. A majority of events would be normal system events such as, but not limited to, capacitor switching and motor starting. If all these events are reported, utility personnel may be overloaded with data, and they may miss abnormal events that were buried among normal system activity. It is desirable that a higher level algorithm go through the classification results and decide what information needs to be presented to utility personnel. Such intelligent reporting algorithms were also developed by the author as a part of DFA project. Sample results produced by reporting algorithms are also presented in Chapter 7.

## 3.5 Chapter Summary

In this chapter, the DFA platform used for acquiring waveform was introduced. Then, the power system event classification problem was formulated as a number of sub-problems. Each of these sub-problems including segmentation, feature extraction and event classification, were described in detail. The rationale for choosing a fuzzy expert system based classifier over a probabilistic expert system was explained. The power system event classification problem was shown to be a large scale problem, and the need for fuzzy hierarchical classifier was explained. Finally, a classification scheme was proposed as a solution to the power system event classification problem.

# 4. RMS SHAPE ANALYSIS

## 4.1 Introduction

Majority of power system events (90-95%) can be classified by a human expert based on the shapes observed in cycle-by-cycle waveforms. Cycle-by-cycle waveforms are computed from high-speed waveforms, using a non overlapping window of length equivalent to one cycle of the power system frequency. Cycle-by-cycle waveforms include, three-phase and neutral RMS (Root Mean Square) currents, three-phase RMS voltages, three-phase and neutral RMS of phasor differenced[1] currents, three-phase RMS of phasor differenced voltages, three-phase real and reactive phasor power. For data used in this research, high-speed waveforms were acquired at a rate of 15,360 samples/second. For a 60Hz power system frequency, each cycle corresponds to 256 samples. When a non overlapping window of length 256 is used, a total of $15,360/256 = 60$ cycle-by-cycle values are computed per second. For simplicity, from here on, all cycle-by-cycle waveforms will be referred to as 'RMS waveforms'.

Shapes that are commonly observed in RMS waveforms include, but are not limited to, step changes, surges, dips, and exponential decay (Figure 4.1). On most occasions, an expert associates the event with its root-cause based on shapes, and some simple features associated with these shapes. Hence, RMS waveform shape detection and shape parameter estimation are key factors in solving power system event classification problem. The following sections discuss challenges associated with RMS waveform shape detection and methods for solving this problem.

[1] Phasor differencing is a technique developed at Texas A&M's Power System Automation Laboratory; it is used for removing an estimated steady-state load component from a sampled signal in the presence of phase drifts introduced either due to sampling or due to changes in power system frequency [77].

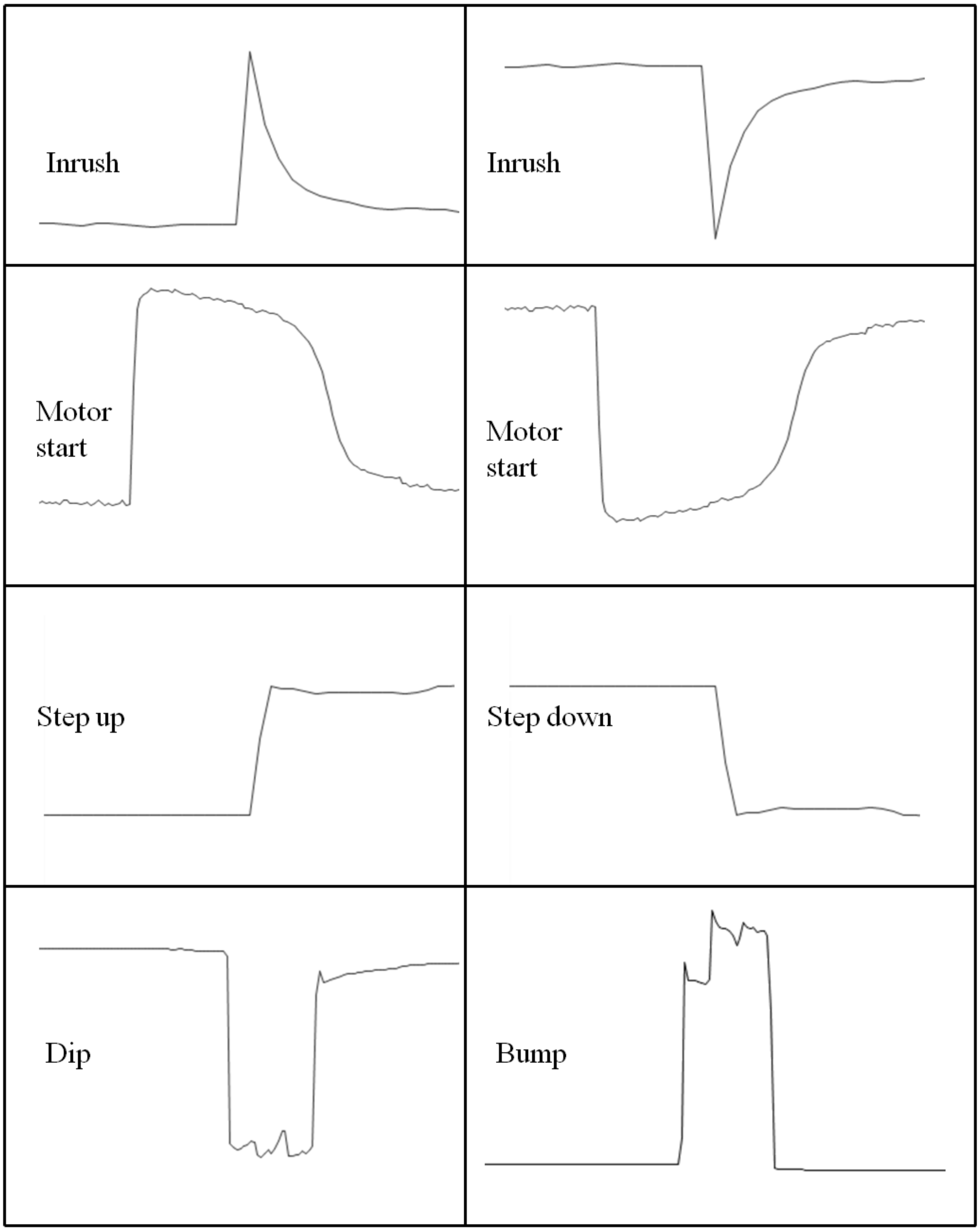


Figure 4.1: RMS waveform shape examples

## 4.2 RMS Waveform Shape Detection Problem

Power system event shape detection is challenging because, power system events are time-variant. Event data belonging to the same event category may have different magnitudes, durations and frequency characteristics. These differences can be due to one or more of the following reasons:

1. Changes in the state of the power system: Events with the same root-cause may have different characteristics depending on the state of the feeder when the events were recorded.

2. Multiple components of the same type: Events belonging to the same event category will have different characteristics if they were not caused by the same component. For example, a distribution feeder can have more than one capacitor bank. Each of these banks will have different characteristic based on their rating and location. Hence, waveforms recorded during capacitor switching events may differ in electrical characteristics as measured at the substation if they were caused by different capacitor banks.

3. Changes in a component's characteristic: Operating characteristics of a component may change over time due factors such as a manual change in the configuration of the component, weather and aging. As a result, events caused by the component may differ in their characteristics from similar or "identical" components, overtime.

Figure 4.2 shows real power (P) and reactive power (Q) waveform plots form three events caused by motor starts. Plots a, b are from motor start events recorded on the same feeder while plot c is a motor start event from another feeder. It can be seen that the plots have similar shapes but exhibit the following differences:

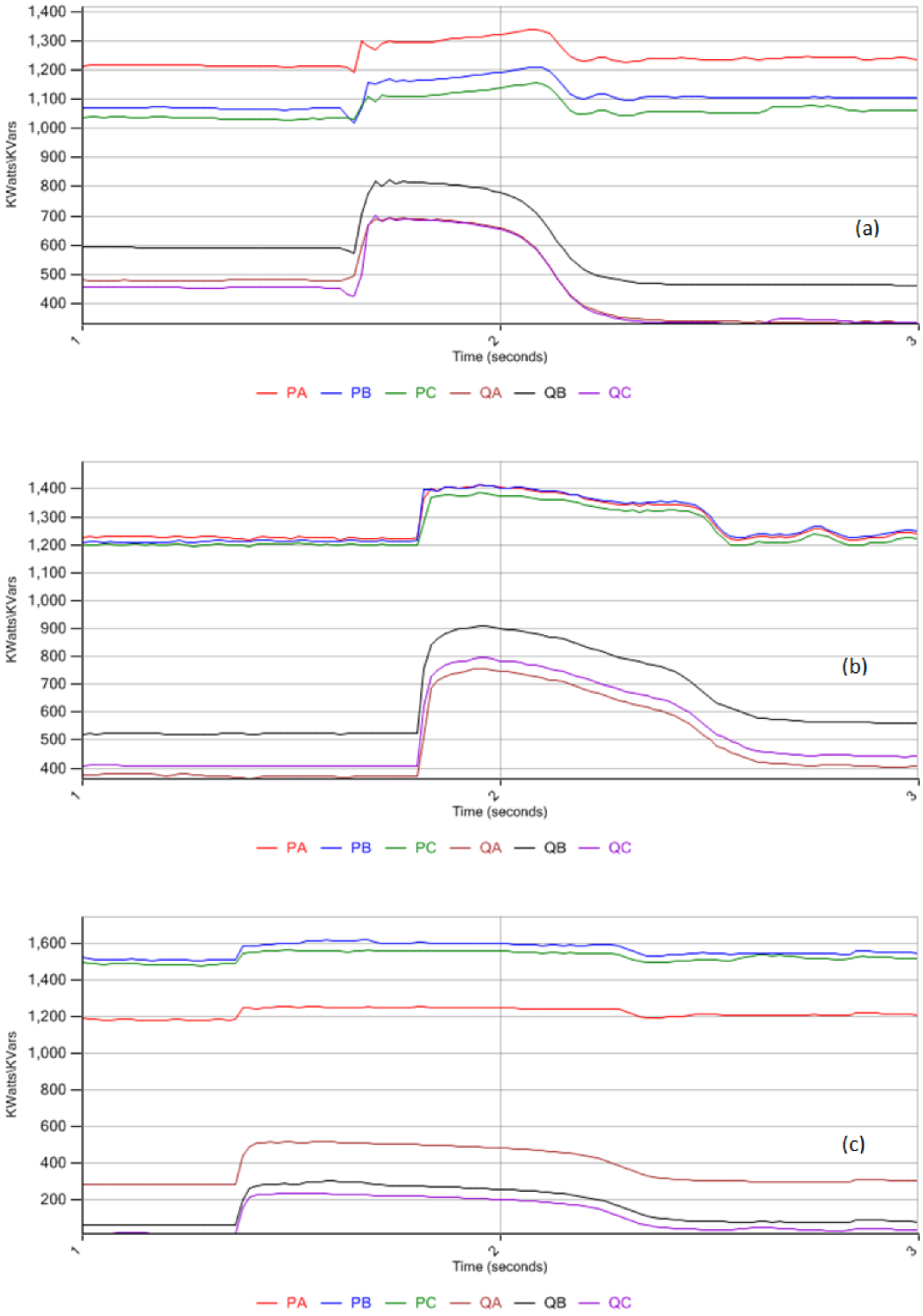


Figure 4.2: Real and reactive power plots from motor start events

1. Translation: Within the plot, events do not start at the same time.
2. Scaling: Duration and magnitude of the motor start events are different.
3. Warping: Event shape appears stretched or shrunk in time.

The effect of translation may be removed by detecting event start time and ignoring pre-event data. Scaling may be compensated by normalizing the data. However, warping is harder to remove or compensate for. Simple template matching techniques that uses Euclidean distance (i.e. trying to match a template with observed waveform on a point by point basis) will not work. Dynamic Time Warping (DTW) is a tool that could be used to solve the above problem.

## 4.3 Shape Template Matching Using DTW

DTW is a technique that finds the optimal alignment between two time series where one of the time series is warped (stretched or shrunk non-linearly) in time. DTW has been used to determine similarity between spoken sounds in the field of speech recognition [91]. In speech recognition, duration of sounds and pause between sounds are allowed to vary, but the overall sound needs to be similar. DTW and variations of DTW have become a widely used tools for time series pattern analysis [92]. DTW has previously been used for classification of power quality disturbances [93]. However, in the previous approach, DTW was not used for RMS shape recognition.

Figure 4.3 shows an example shape template for warping a motor start shape to a sample RMS waveform. Vertical lines connect each point in the shape template to one or more similar points on the RMS waveform. If no warping is needed, then each point in the shape template will be connected to a single point in the RMS waveform. DTW uses warp path distance instead of Euclidean distance as a

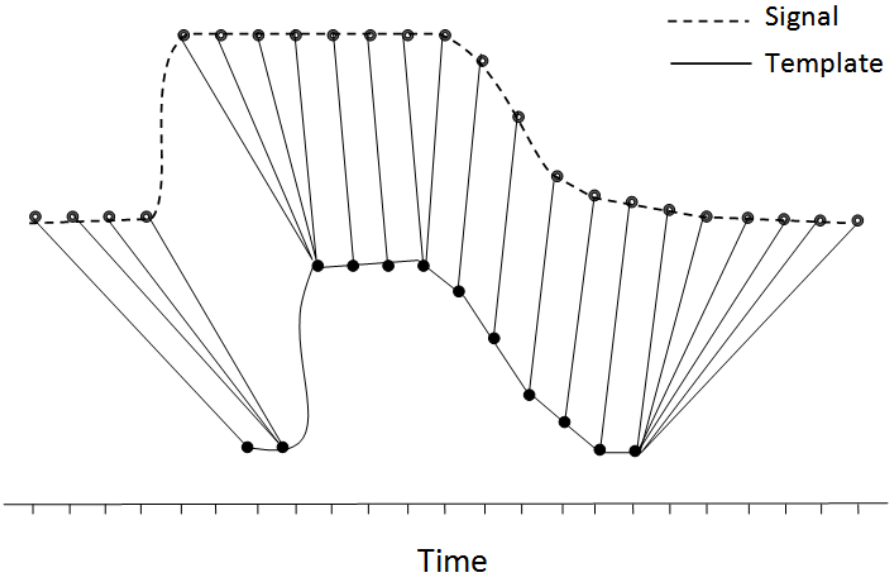


Figure 4.3: Warping a shape template to a RMS waveform

measure of similarity between the template and the signal of interest. The warp path distance is the sum of the distances between each point in the template to one or more corresponding points in the waveform. If the RMS waveform is identical to the template except for localized warping (i.e if the template can be stretched or shrunk to fit the RMS signal exactly) then the warp path distance will be zero.

### *4.3.1 Finding Optimal Warp Path*

Before the warp path distance between a shape template and a RMS waveform can be computed, an optimal warping path needs to be found. The general form of optimal time warping problem [94] was modified for use with RMS waveform shape recognition. The problem of finding optimal time warp path for RMS waveform shape matching can be stated as follows: Given a shape template $T = \{ t_1, t_2, \cdots, t_i, \cdots, t_{|T|} \}$ of lengths $|T|$ and a RMS waveform $X = \{ x_1, x_2, \cdots, x_j, \cdots, x_{|X|} \}$ of length $|X|$, find the optimal path $W^\star$ among all possible paths.

Consider a warp path $W$.

$$W = w_1, w_2, \cdots, w_k, \cdots, w_{|W|} \tag{4.1}$$

where:

$|W|$ is length of the warping path, where $|W| = max(|T|,\ |X|)$.

$w_k$ represents the $k^{th}$ element of warp path, where $w_k = (i,\ j)$, $i$ represents the $i^{th}$element in the shape template $T$ and $j$ represents corresponding $j^{th}$element in the RMS waveform $X$.

$T$ , $X$ are both normalized and mapped to the interval $[0, 1]$.

There are some constraints imposed on the warp path for all shape templates and RMS waveforms. The warp path needs to start at the beginning of the shape template, and finish at the end of shape template. The beginning and end of the warp path should also coincide with the beginning and end of the RMS waveform. This constraint can be stated as:

$$w_1 = (1,\ 1)\ ,\ w_{|W|} = (|T|,\ |X|) \tag{4.2}$$

The above constraint ensures that whole of the template and RMS waveforms are traversed and matched. Another constraint requires the indices $i$ and $j$ to increase monotonically in the warp path:

$$w_k = (i,\ j)\ ,\ w_{k+1} = \left(i',\ j'\right) \quad i \leqslant i' < i + n_{T(i)},\ j' = j + 1 \tag{4.3}$$

where:

$n_{T(i)}$ is a template position dependent parameter that determines the number of template points that may be skipped when the warping path is at index $i$ of template $T$.

The above constraint ensures that the warping path always moves forward in time. It also ensures that a single point in the RMS waveform is not matched to more than one point in the shape template. However, a single point on the shape template may match more than one consecutive points on the RMS waveform. This will allow stretching of the template waveform. Equation 4.3 also imposes a template position dependent upper bound on $i'$. This is a local constraint that allows warping paths to skip some points on the shape template. Skipping points on shape template is equivalent to shrinking shape template to fit the RMS waveform. Two other local constraints are also introduced:

$$w_k = (i,\ j)\,,\ w_{k+m} = \left(i',\ j'\right),\quad i = i',\ 0 \leqslant m \leqslant Lb_{T(i)},\ j' = j + m \tag{4.4}$$

$$w_k = (i,\ j)\,,\ w_{k+m} = \left(i',\ j'\right),\quad i < i',\ Ub_{T(i)} < m,\ j' = j + m \tag{4.5}$$

Equation 4.4 imposes a bound on the minimum duration $Lb_{T(i)}$ for which the warping path $W$ has to stay at a template index $i$ for the template $T$. This forces the warping path to stay at certain key positions on the shape template for a minimum specified duration. Equation 4.5 imposes a bound on the maximum duration $Ub_{T(i)}$ for which the warping path $W$ can stay at a template index $i$ for the template $T$. This prevents the warping path from staying indefinitely at some positions on the shape template. This local upper bound is needed to prevent shape distortion (Figure 4.4).

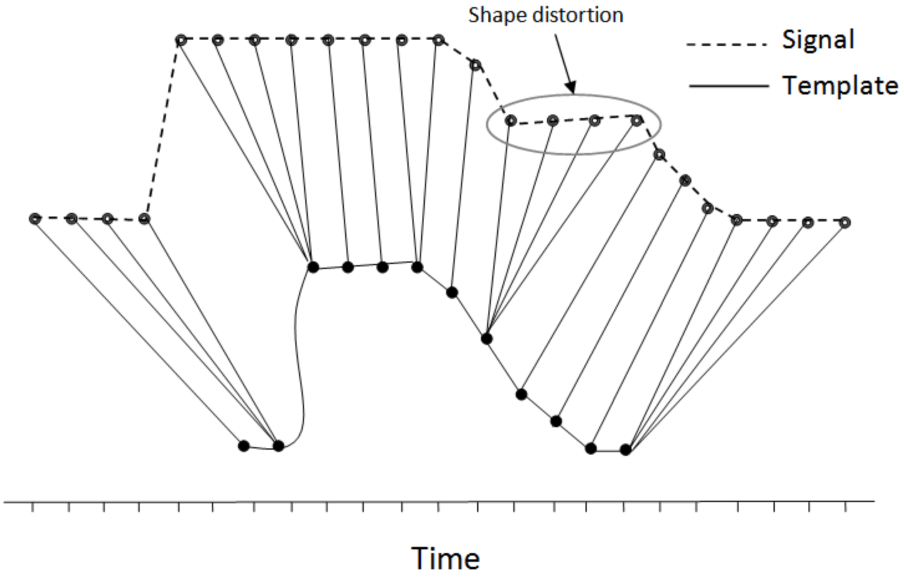


Figure 4.4: Shape distortion when warp path stays at some template points for too long

Optimal warp path $W^{\star}$ is the best template match warp path that minimizes the distance between shape template $T$ and RMS waveform $X$ over all possible warp paths $W$:

$$W^{\star} = arg \min_{W} \{Dist(W)\} \tag{4.6}$$

$$Dist(W) = \sum_{k=1}^{k=|W|} Dist\left(T\left(i_{w_k}\right),\ X\left(j_{w_k}\right)\right) \tag{4.7}$$

$Dist(W)$ is the total distance computed over the warp path $W$ that satisfies the constraints imposed by Equations (4.2-4.5). $Dist\left(T\left(i_{w_k}\right),\ X\left(j_{w_k}\right)\right)$ represents the distance between shape template $T$ and RMS waveform $X$ at the $k^{th}$ position in the warp path $W$. $i_{w_k}$ and $j_{w_k}$ represent shape template and RMS waveform indices at the $k^{th}$ position in the warp path $W$. Since the warping path always traverses RMS waveform at one sample increments (Equation 4.3), $w_k = (i,\ k)$ and $j_{w_k} = k$.

Equation 4.7 can be simplified to:

$$Dist(W) = \sum_{k=1}^{k=|W|} Dist\left(T\left(i_{w_k}\right),\ X\left(k\right)\right) \tag{4.8}$$

$$Dist\left(T\left(i_{w_k}\right),\ X\left(k\right)\right) = \left|T\left(i_{w_k}\right) - X\left(k\right)\right| \tag{4.9}$$

Dynamic programming [95] can be used to solve this optimization problem. This is because, if all optimal warping paths till $k^{th}$ point in RMS waveform $X$ are known, then it is possible to compute all the optimal paths for a one point increment $(k+1)$ on RMS waveform. In other words, if $W_{opt}^{k}$ is a set of all optimal warping paths till the $k^{th}$ point in RMS waveform $X$ for a shape template $T$, it is possible to find all optimal warping paths till $(k+1)^{th}$ point by solving a simple optimization problem for moving from $k^{th}$ point to $(k+1)$ as:

$$W_{k+1}^{\star} = f\left(W_{k}^{\star},\ X(k+1),\ T\right) \tag{4.10}$$

$$W_{k}^{\star} \in \left\{W_{k,T(i)}^{\star}\right\},\ i \in I,\ I \subset \{1, 2, \cdots, |T|\} \tag{4.11}$$

$$W_{k+1}^{\star} \in \left\{W_{k+1,T(i')}^{\star}\right\},\ i' \in I',\ I' \subset \{1, 2, \cdots, |T|\} \tag{4.12}$$

where:

$W_{k+1}^{\star}$ is a set of all possible optimal warp paths at $(k+1)^{th}$ point in RMS waveform. There can at most be $|T|$ optimal warp paths that end at $(k+1)^{th}$ point corresponding to each point $i$ on shape template $T$. However, due to constraints imposed by equations (4.2-4.5), in reality only a subset of optimal warp paths exists (Equation 4.12). $W_{k+1,T(i')}^{\star}$ represents one such optimal warp path that ends at $(i',\ k+1)$ and $i'$ belongs to the set of allowable template indices $I'$.

$W_k^\star$ is a set of all possible optimal warp paths at $k^{th}$ point in RMS waveform. There can at most be $|T|$ optimal warp paths that end at $k^{th}$. However, due to constraints imposed by equations (4.2-4.5), in reality only a subset of optimal warp paths exists (Equation 4.11). $W_{k,T(i)}^\star$ represents one such optimal warp path that ends at $(i,\ k)$ and $i$ belongs to the set of allowable template indices $I$.

The function $f$ in Equation 4.10 can be used to find all optimal warp paths $W_{k+1}^\star$ at $(k+1)^{th}$ point $X(k+1)$ of RMS waveform using a shape template $T$, when all optimal paths at $k^{th}$point $W_k^\star$ are known. The function $f$ uses the following relations:

$$W_{k+1,T(i')}^\star = \left\{W_{k,T(n)}^\star,\left(i',\ k+1\right)\right\},\ n \in I \tag{4.13}$$

$$n = arg\ \underset{i}{min} \left\{Dist(W_{k,T(i)}^\star) + Dist\left(T\left(i'\right),\ X\left(k+1\right)\right)\right\},\ i \in I \tag{4.14}$$

The above equations can be used to find the optimal warp path incrementally. This is because, $(k+1)^{th}$ element of any optimal warp path $W_{k+1,T(i')}^\star$ that ends at $(i',\ k+1)$ is $w_{k+1}^\star = (i',\ k+1)$. The point $(i',\ k+1)$ can only be reached from any one of $|I|$ points $w_k^\star = (i,\ k)\ ,\ i \in I$ from the previous step. If the optimal warp path $W_{k,T(i)}^\star$ to each of the previous points $(i,\ k)$ is known, and the cost $Dist(W_{k,T(i)}^\star)$ associated with each of these paths are also known; then the problem reduces to determining the single optimal point $(n,\ k)$ from which the point $(i',\ k+1)$ can be reached such that the total incremental cost (Equation 4.14) is minimized. Equation 4.10 shows recursive nature of the optimal warp path problem. Equation 4.14 shows how the problem can be broken into sub problem of minimizing warp path cost of moving from $k^{th}$ point to $(k+1)^{th}$ point. The ability to break the problem into sub problems that are smaller but similar is a requirement for applying dynamic programming technique [96].

### 4.3.2 DTW Based RMS Shape Classifier

Different shapes such as step changes, surges, dips etc., need to be detected in RMS waveforms. Hence, multiple shape templates $T_s$, each to detect a specific shape $Shape_s$ are needed. A classifier may be designed to classify the RMS waveform based on the shape observed in the waveform. To do this, for each $T_s$, optimal warp paths $W^\star_{T_s}$ and minimum cost $D_{T_s} = Dist\left(W^\star_{T_s}\right)$ are first computed using the method outlined in Section 4.3.1. Then a simple DTW based classifier $C_{Shape}$ assigns the shape label $Shape_s$ to the RMS waveform $X$ such that the $T_s$ has the least total warp path cost:

$$C_{Shape}\left(X\right) = arg\ \underset{s}{min}\left\{D_{T_s}\right\},\ 0 < s \leqslant N_s \tag{4.15}$$

It is possible that none of the shape templates match the shape observed in RMS waveform. However, equation 4.15 will assign a shape label even when none of the shape templates match. To avoid this scenario, a shape label '*Unknow*' is assigned if $\underset{s}{min}\left\{D_{T_s}\right\} > Th_U$, where $Th_U$ is a threshold for warp path cost.

### 4.3.3 Time and Memory Complexity of Using DTW

The advantage of using dynamic programming technique is that it is guaranteed to construct the optimal warping path for the given constraints. It does not impose any additional requirements on the nature of the cost function *Dist* or the constraints that can be used. However, its' main disadvantage is time and space complexity due to the enumerative nature of search. For example, if no local constraints were imposed, optimal incremental warping paths need to be computed for each point $k$ on RMS waveform $|T|$. This results in a total of $|X| * |T|$ cost computations, which is $O\left(N^2\right)$ if $N = |X| = |T|$. In terms of memory requirements, a matrix of size $|X| * |T|$ will be needed to store optimal warp paths at each point $k$. Such quadratic time and memory complexity, will be prohibitive for large values of $N$.

Time and memory complexity do not pose a problem for RMS waveform shape matching proposed in this research. This is because, the size of shape templates $|T|$ used for RMS shape matching is very small in comparison to length of RMS waveforms $|X|$ . Also, using local constraints reduces the number of template indices that need to be searched at each point. Hence, $|T| \ll |X|$ and $|X| * |T|$ becomes $|X| * |T| = c * |X|$ where $c$ is a small number, and $c = |T|$. This changes the time and space complexity from being quadratic to being linear $O\left(N\right)$. For example, waveform data used for this research are typically few seconds long (about five seconds), and for waveform data of length five seconds, $|X| = 5 * 60 = 300$. Shape templates used for this research have an average length of 12 points. However, due to local constraints that are imposed, at most 4 points on the shape template are considered at each point in the warping path. Hence, only a total of about $4 * |X| = 1200$ distance computations are required. Even when assuming a worst case scenario of waveform data being 10 minutes long, this translates to RMS waveforms of length $|X| = 10 * 60 * 60 = 36,000$ samples. Hence a total of about $4 * |X| = 144,000$ distance computations may be required which is still manageable by present day processors.

## 4.4 Fuzzy Shape Template Matching Using DTW

DTW based template matching outlined in Section 4.3 was initially used in this research for recognizing shapes. However, there were some drawbacks of using this approach. Shape templates do not account for the presence of local

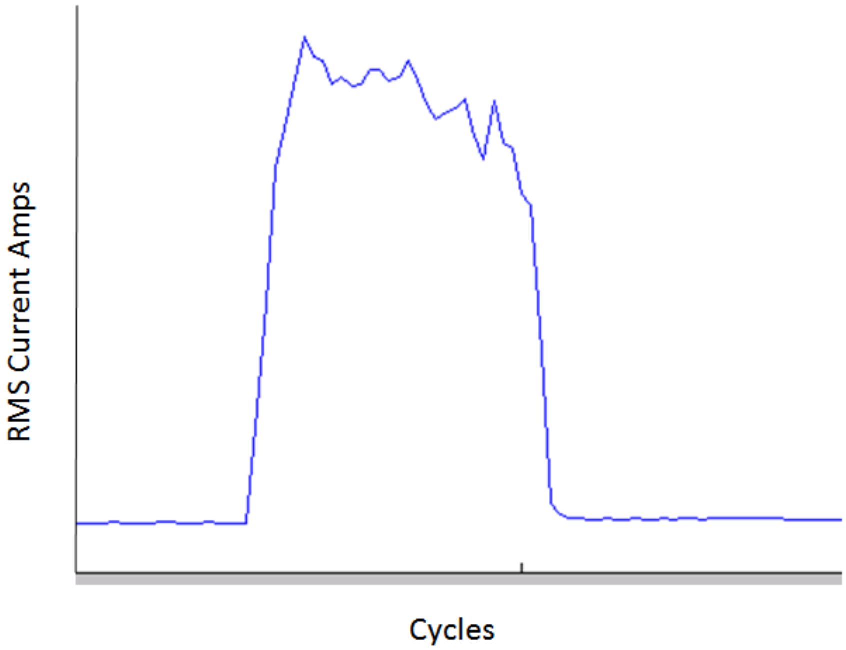


Figure 4.5: Motor start with local variations

variations or noise in RMS waveforms. Figure 4.5 shows one such example. Local variations will add to the cost function as they will be viewed as deviations from the reference template. When such deviations are aggregated over the warping path, it may result in high overall cost. Such high warp path cost may cause a shape template to be rejected even though the general shape of the event matched the reference template.

Some ways of improving the classification performance are:

1. Increasing the number of sample points on the shape templates.
2. learning shape templates by regressing over sample shapes.
3. Using vector quantization technique to identify optimal sample points on shape templates.

These approaches increase the possibility of 'over-fitting' the shape templates to

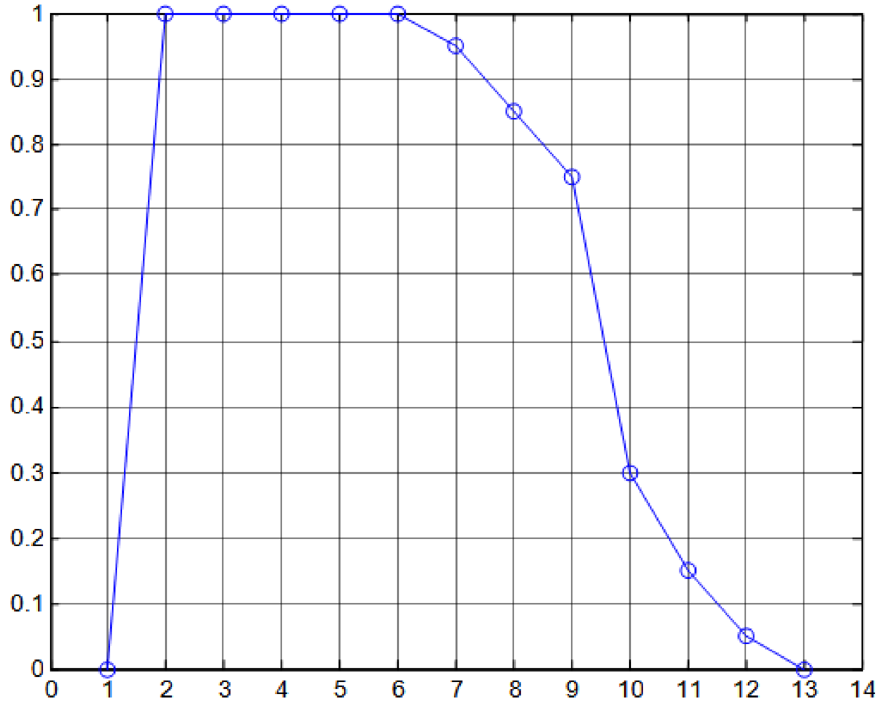


Figure 4.6: Sample motor start shape template

the test data an increase in computational time.

### *4.4.1 Fuzzy Shape Templates*

To overcome the uncertainty introduced due to local variations, shape templates can be fuzzified. This allows for 'elastic' matching in the presence of local variations or noise. Based on this idea, a new DTW approach based on fuzzy shape templates is proposed. This method will here on be referred to as Fuzzy Dynamic Time Warping (FDTW).

Figure 4.6 shows a sample motor start shape template. The horizontal axis corresponds to template index and the vertical axis corresponds to template values. A shape template is represented as a series of index, value pairs $T$, where $T = \left\{(1, v_1), (2, v_2), \cdots, (|T|, v_{|T|})\right\}$. When trying to match $k^{th}$ point on RMS waveform $X$ with $i^{th}$ point on template $T$, DTW technique uses the absolute distance $|X(k) - T(i)|$. When the RMS waveform has a shape that is very close to the template but has some local variations, points on the RMS waveform may be represented as $X(k) = T(i) + \Delta L,\ \Delta L < T(i)$. The absolute distance $|X(k) - T(i)| = \Delta L$ will contribute to the cost computed over the warp path. When visually analyzing RMS waveforms, an expert will ignore such local variations and will be able to match the general shape. In an attempt to mimic this behavior, template $T$ is fuzzified by assigning a fuzzy membership function to each point in the template:

$$T_{Fuzzy} = \left\{ (1, \mu_1), (2, \mu_2), \cdots, (i, \mu_i), \cdots, \left( \left| T_{fuzzy} \right|, \mu_{\left| T_{Fuzzy} \right|} \right) \right\},\ 1 \leqslant i \leqslant \left| T_{Fuzzy} \right| \tag{4.16}$$

where $T_{Fuzzy}$ is a fuzzy shape template represented as a sequence of (index, membership function) pairs. $\mu_i$ represents fuzzy membership function corresponding to fuzzy sets such as 'High', 'Medium' and 'Low'. Figure 4.7 shows how the 'crisp' motor start shape template in Figure 4.6 can be converted to a fuzzy shape template using three fuzzy membership functions. Here, three fuzzy membership functions $\mu_H$, $\mu_L$ and $\mu_M$ corresponding to fuzzy sets *High*, *Medium* and *Low* are used (shown in the far left side). Each template index $1, \cdots, 14$ (horizontal axis) is assigned either of the three fuzzy membership functions $\mu_H$, $\mu_L$ or $\mu_M$ instead of numeric values (shown directly below template indices). To keep the illustration simple, only three fuzzy sets are used for this example template. In reality, more than three fuzzy sets may be used for increased granularity. The equation below shows the indices and values of crisp motor start template $T_{Crisp-Motor}$ and fuzzy motor start template $T_{Fuzzy-Motor}$. Here the templates are shown in a vector form

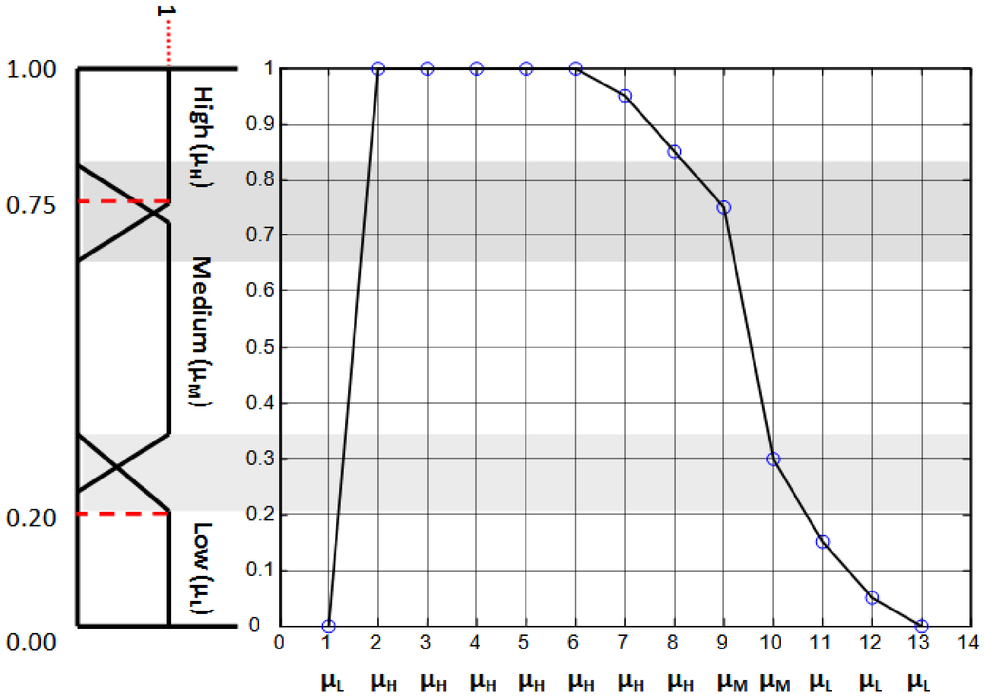


Figure 4.7: Sample fuzzy motor start shape template

without indices:

$$T_{Crisp-Motor} = \{0.0, 1.0, 1.0, 1.0, 1.0, 1.0, .95, .85, .75, 0.3, .15, .05, 0.0\} \tag{4.17}$$

$$T_{Fuzzy-Motor} = \{\mu_L, \mu_H, \mu_H, \mu_H, \mu_H, \mu_H, \mu_H, \mu_H, \mu_M, \mu_M, \mu_L, \mu_L, \mu_L\} \tag{4.18}$$

Fuzzy motor start shape template (Equation 4.18) can be interpreted as follows: "Motor start shape starts with *Low* values, followed by at least seven *High* values, followed by at least two *Medium* values and ends with at least three *Low* values". For matching $k^{th}$ point on RMS waveform $X$ with $i^{th}$ point on template $T_{Fuzzy}$, the fuzzy membership function $\mu_i$ assigned to the $i^{th}$ index of the fuzzy template can be used:

$$Match\left(T_{Fuzzy}\left(i\right),\ X\left(k\right)\right) = \mu_i\left(X\left(k\right)\right) \tag{4.19}$$

$$Dist\left(T_{Fuzzy}\left(i\right),\ X\left(k\right)\right) = \neg\mu_i\left(X\left(k\right)\right) \tag{4.20}$$

$$\neg\mu_i\left(X\left(k\right)\right) = 1 - \mu_i\left(X\left(k\right)\right) \tag{4.21}$$

When $\mu_i\left(X\left(k\right)\right)$ is used directly (Equation 4.19), it returns a value in the interval $[0, 1]$ that represents the degree to which $X\left(k\right)$ is in the fuzzy set assigned to the template position $i$. For example, when using the fuzzy template $T_{Fuzzy-Motor}$, for template index $i = 1$, $\mu_i = \mu_L$. Then, $\mu_L\left(X\left(k\right)\right)$ returns a value that represents the degree to which $X\left(k\right)$ is *Low*. The degree of match (Equation 4.19) can be converted to a cost or distance measure by determining the degree of 'not match' using the fuzzy *Not* operator (4.20).

The main reason for using a fuzzy shape template is to make the shape template less sensitive to local variations and noise. If the membership functions $\mu_i$ are designed carefully, the distance measure (Equation 4.20) can be made less sensitive to local variations. It is easier to demonstrate this with an example. Figure 4.8 shows RMS waveform of a noisy motor start (shown in blue) superimposed on a fuzzy motor shape template. Points on the crisp motor shape template (shown in black) are also shown for comparison. The RMS waveform (blue) shows a lot of local variation at the top corresponding to the values in the range $[0.7, 1.0]$. When the fuzzy membership function $\mu_H$ corresponding to the fuzzy set *High* is used to match these RMS values corresponding to template indices 2-8, all RMS values in the interval $[.75, 1.0]$ return a degree of match equal to 1.0 and the distance measure of 0.0. However, when the crisp template is used, none of points corresponding to template indices 2-8 will return a non zero distance. Hence, the crisp template will result in a higher warp path cost when compared to the fuzzy template.

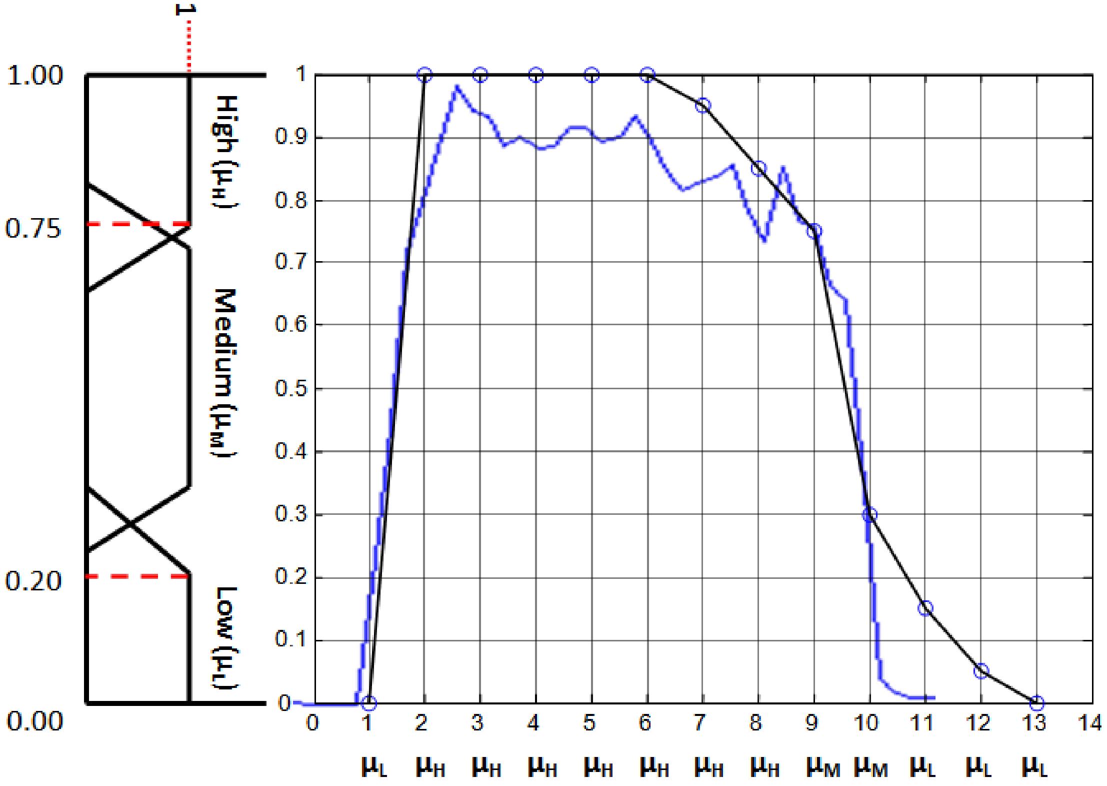


Figure 4.8: Fuzzy template matching of a noisy motor start

### 4.4.2 Computing Optimal Warp Path for Fuzzy Shape Templates

Similar to the method outlined in Section 4.3.1, before the warp path distance or the degree of match between a fuzzy shape template and a RMS waveform can be computed, an optimal warping path needs to be found. However, dynamic programming approach outlined in Section 4.3.1 cannot be used directly for computing the optimal warp path for fuzzy templates. This is because of the fuzzy nature of template matching. The problem of finding the optimal warping path for a fuzzy shape template can be formulated as a multistage fuzzy decision making problem under Bellman and Zadeh's framework [97, 98].

#### 4.4.2.1 Review of multistage decision making under fuzziness

Bellman and Zadeh [97] proposed that any conventional decision making process can be looked at as exploring a set of alternatives; taking into consideration constraints imposed on the alternatives, and evaluate the alternatives based on a performance function that evaluates the goodness of each alternative. Bellman and Zadeh went on to generalize the conventional decision making process by proposing a fuzzy decision making framework as follows:

If $X$ is a set of all alternatives (choices) and if $G$ is a fuzzy goal in $X$ and $C$ is a fuzzy constraint in $X$, then a fuzzy decision $D$ in $X$ is the result of attaining the fuzzy goal $G$ while satisfying the fuzzy constraints $C$. Since $G,\ C$ and $D$ are all fuzzy sets in $X$, there exists corresponding fuzzy membership functions $\mu_G,\ \mu_C$ and $\mu_D$. Then for any alternative $x \in X$, the following relationship may be defined [98] for a fuzzy decision involving $x$:

$$\mu_D(x) = \mu_C(x) \wedge \mu_G(x) \tag{4.22}$$

where $\wedge$ is the fuzzy conjunction operator. In the above equation, $\mu_D(x)$ provides a measure of goodness of the alternative $x$ based on the degree to which it satisfies the fuzzy constraints ($\mu_C(x)$), and the degree to which it attains the fuzzy goal ($\mu_G(x)$). The objective is to find the best alternative $x_{opt},\ x_{opt} \in X$ that meets the fuzzy constraints $C$ and attains the fuzzy goal $G$. Hence, this can be represented as an optimization problem that maximizes $\mu_D(x)$ as follows:

$$x_{opt} = arg \underset{x \in X}{max} \left\{ \mu_D\left(x\right) \right\} \tag{4.23}$$

$$x_{opt} = arg \underset{x \in X}{max} \left\{ \mu_C(x) \wedge \mu_G(x) \right\} \tag{4.24}$$

Bellman and Zadeh extended the above approach to the multistage decision making under fuzziness. Consider a memory less fuzzy system whose next state $s_{k+1}$ depends only on the system's current state $s_k$ and new input $x_k$. Then the system can be described by the following state transition equation:

$$s_{k+1} = f\left(s_k,\ x_k\right),\ k = 0, 1, 2, \cdots, N-1 \tag{4.25}$$

In the context of multistage decision making, in the above equation, $k$ can be considered as a specific stage. $x_k$ can be considered as a specific choice made at the stage $k$ where $x_k \in X$ and $X$ is the set of all alternatives. The next state of the system $s_{k+1}$ can be considered as the output as a result of the choice $x_k$, where $s_{k+1} \in S$ and $S$ is the set of all valid states. Then a fuzzy goal $G_{k+1}$ can be imposed on $s_{k+1}$to evaluate the goodness of the output. Additionally, a fuzzy constraint $C_k$ may be imposed on the choice $x_k$. Then, similar to Equation 4.22, the goodness of fuzzy decision $D_k$ at stage $k$ may be evaluated by:

$$\mu_{D_k}(x_k) = \mu_{C_k}(x_k) \wedge \mu_{G_{K+1}}(s_{k+1}) \tag{4.26}$$

The above equation evaluates the performance of a choice $x_k$ for a single stage. For a N stage decision making problem, assuming the initial state $s_0$ is known, the performance of a sequence of choices $x_0, x_1, \cdots x_{N-1} \in X \times X \cdots \times X$ can be viewed as a fuzzy decision $D$ in $X \times X \cdots \times X$, and can be evaluated as follows [98]:

$$\begin{aligned} &\mu_D(x_0, x_1, \cdots x_{N-1}|s_0) \\ = \mu_{C_0}(x_0) \wedge \mu_{G_1}(s_1) \wedge \cdots \wedge \mu_{C_k}(x_k) \wedge \mu_{G_{K+1}}(s_{k+1}) &\wedge \cdots \wedge \mu_{C_{N-1}}(x_{N-1}) \wedge \mu_{G_N}(s_N) \\ &= \bigwedge_{k=0}^{N-1} \left(\mu_{C_k}(x_k) \wedge \mu_{G_{K+1}}(s_{k+1})\right) \end{aligned} \tag{4.27}$$

In a multistage decision making problem, the objective is to find the optimal sequence of choices $x_0^\star, x_1^\star, \cdots x_{N-1}^\star \in X \times X \cdots \times X$ that maximizes the goodness of decision over all the stages. Then, the multistage decision optimization problem is equivalent to finding $x_0^\star, x_1^\star, \cdots x_{N-1}^\star$ such that:

$$x_0^\star, x_1^\star, \cdots x_{N-1}^\star = arg \max_{x_0, x_1, \cdots x_{N-1}} \{\mu_D(x_0, x_1, \cdots x_{N-1}|s_0)\} \tag{4.28}$$

$$x_0^\star, x_1^\star, \cdots x_{N-1}^\star = arg \max_{x_0, x_1, \cdots x_{N-1}} \left\{ \bigwedge_{k=0}^{N-1} \left(\mu_{C_k}(x_k) \wedge \mu_{G_{K+1}}(s_{k+1})\right) \right\} \tag{4.29}$$

or

$$\mu_D\left(x_0^\star, x_1^\star, \cdots x_{N-1}^\star|s_0\right) = \max_{x_0, x_1, \cdots x_{N-1}} \left\{ \bigwedge_{k=0}^{N-1} \left(\mu_{C_k}(x_k) \wedge \mu_{G_{K+1}}(s_{k+1})\right) \right\} \tag{4.30}$$

#### 4.4.2.2 *Fuzzy template matching - a special case of multistage decision making under fuzziness*

Finding the optimal warp path for matching a fuzzy template to an RMS waveform can be considered as a special case of multistage decision making problem outlined in the previous section. Let $T_{Fuzzy} = \{\mu_0, \cdots, \mu_{M-1}\}$ be a fuzzy shape template of length $M$. Let $R = \{r_0, \cdots, r_k, \cdots r_N\}$ be an RMS waveform containing $N+1$ samples. Then the objective is to find the optimal path $W_N^{\star}$ among all possible paths. Consider a warp path $W_N \in U \times U, \cdots, U$.

$$W_N = w_0, \cdots, w_k, \cdots, w_N \tag{4.31}$$

where:

$w_k$ represents the $k^{th}$ element of warp path, where $w_k = (i_k,\ k) \equiv \left(\mu_{i_k},\ R\left(k\right)\right)$, $i_k$ represents the $i_k^{th}$element in the shape template $T_{Fuzzy}$ and $k$ represents corresponding $k^{th}$element in the RMS waveform $R$.

$R$ is assumed to be normalized and mapped to the interval $[0, 1]$.

The general constraints described for the DTW technique regarding the beginning, end states and time monotonicity (Equations 4.2,4.3) are applicable here too:

$$w_0 = (0,\ 0)\,,\ w_{|W|-1} = w_N = (M,\ N) \tag{4.32}$$

$$w_k = (i_k,\ k)\,,\ w_{k+1} = (i_{k+1},\ k+1) \quad i_k \leqslant i_{k+1} < i_k + n_{T(i)} \tag{4.33}$$

where:

$n_{T(i)}$ is a template position dependent parameter that determines the number of template points that may be skipped when the warping path is at index $i_k$ of template $T_{fuzzy}$.

Local constraints such as the ones described in Equations 4.4 and 4.5 are also applicable:

$$w_k = (i_k,\ k)\,,\ w_{k+m} = (i_{k+1},\ k+m)\,,\quad i_k = i_{k+1},\ 0 \leqslant m \leqslant Lb_{T(i_k)} \tag{4.34}$$

$$w_k = (i_k,\ k)\,,\ w_{k+m} = (i_{k+1},\ k+m)\,,\quad i_k < i_{k+1},\ Ub_{T(i_k)} < m \tag{4.35}$$

Equation 4.34 imposes a bound on the minimum duration $Lb_{T(i_k)}$ for which the warping path $W$ has to stay at a template index $i_k$ for the template $T_{Fuzzy}$. Equation 4.35 imposes a bound on the maximum duration $Ub_{T(i)}$ for which the warping path $W$ can stay at a template index $i_k$ for the template $T_{Fuzzy}$. Then, the optimal warp path $W^\star = w_0^\star, \cdots, w_k^\star, \cdots, w_N^\star,\ W^\star \in U \times U, \cdots, U$ is the fuzzy template warp path that maximizes the degree of match of the fuzzy template $T_{Fuzzy}$ and RMS waveform $R$ over the whole warp path:

$$W_N^\star = arg\ \underset{W_N}{max}\left\{\mu_{Match_N}(W_N)\right\} \tag{4.36}$$

$$w_0^\star, \cdots, w_k^\star, \cdots, w_N^\star = arg\ \underset{w_0,\cdots,w_k,\cdots,w_N}{max}\left\{\mu_{Match_N}(w_0, \cdots, w_k, \cdots, w_N)\right\} \tag{4.37}$$

where $\mu_{Match_N}$ is a fuzzy membership function that represents the degree of match of fuzzy template warping path $W_N$ and $Match_N$ is a fuzzy set in $W_N \in U \times U, \cdots, U$. Since matching the fuzzy shape template over the whole warp path is equivalent to matching the fuzzy template at every point in the RMS waveform, using (4.19):

$$\mu_{Match_N}(w_0, \cdots, w_k, \cdots, w_N) = \left(\bigwedge_{k=0}^{N} \mu_{i_k}\left(R\left(k\right)\right)\right) \tag{4.38}$$

and

$$w_0^\star, \cdots, w_k^\star, \cdots, w_N^\star = arg \max_{w_0,\cdots,w_k,\cdots,w_N} \left\{ \bigwedge_{k=0}^{N} \mu_{i_k}\left(R\left(k\right)\right) \right\} \tag{4.39}$$

where $\wedge$ is the fuzzy conjunction operator. Since the initial condition $w_0 = (0,\ 0)$ is known, $w_0$may be dropped from the above equation and be rewritten as:

$$w_1^\star, \cdots, w_k^\star, \cdots, w_N^\star = arg \max_{w_1,\cdots,w_k,\cdots,w_N} \left\{ \bigwedge_{k=1}^{N} \mu_{i_k}\left(R\left(k\right)\right) \right\} \tag{4.40}$$

Since $k^{th}$ point in warp path is $w_k = (i_k,\ k)$ and $k$ increments by exactly 1 at every stage, finding the optimal $w_k^\star$ for $k^{th}$ point is the same as finding the optimal fuzzy shape template index $i_k^\star$. Without loss of generality, $\mu_{i_k}\left(R\left(k\right)\right)$ can be represented as $\mu_{Match}\left(w_k\right)$. This is because, for a given RMS waveform $R$, the $k^{th}$ point in the warp path $w_k$ represents the pair $\left(\mu_{i_k},\ R\left(k\right)\right)$. Hence, the problem of finding the optimal warping path may be redefined as the problem of finding the optimal sequence of fuzzy template indices $I_N^\star = i_1^\star, \cdots, i, \cdots, i_N^\star$ such that:

$$i_1^\star, \cdots, i_k^\star, \cdots, i_N^\star = arg \max_{i_1,\cdots,i_k,\cdots,i_N} \left\{ \bigwedge_{k=1}^{N} \mu_{Match}\left(w_k\right) \right\} \tag{4.41}$$

The problem finding the optimal sequence of fuzzy template indices $I^\star = i_1^\star, \cdots, i_k^\star, \cdots, i_N^\star$ may also be presented as a multistage decision making problem. Then, at each stage $k$ template index $i_k$ need to be incremented by a value $x_k \geqslant 0$ such that $i_{k+1} = i_k + x_k$; subject to the constraints defined by 4.33-4.35. Then, for any point $w_k = (i_k, k)$ in the warp path $W$, the next point $w_{k+1} = (i_{k+1},\ k+1)$ can be expressed as:

$$w_{k+1} = (i_{k+1},\ k+1) = (i_k + x_k,\ k+1) = f\left(w_k,\ x_k\right) \tag{4.42}$$

The above equation is very similar to the state transition equation (4.25). Now, the problem finding the optimal sequence of fuzzy template indices $I^\star = i_1^\star, \cdots, i_k^\star, \cdots, i_N^\star$ may be redefined as follows: Finding the optimal sequence of choices or values $X_{N-1}^\star = x_0^\star, \cdots, x_k^\star, \cdots, x_{N-1}^\star$ that will result in the optimal template indices $I^\star = i_1^\star, \cdots, i_k^\star, \cdots, i_N^\star$ given $i_0 = i_0^\star = 0$ and subject to the constraints defined by 4.33-4.35. It is convenient to make the constraints, part of the cost or match function. For this, at each stage $k$, a crisp constraint $c_k\left(x_k\right)$ can be introduced. $c_k\left(x_k\right)$ evaluates to 0 if any constraint is violated for a choice $x_k$and $c_k\left(x_k\right)$ evaluates to 1 if no constraints are violated. Also, at each stage $k$, a choice $x_k$ will induce a template match $\mu_{Match}\left(w_{k+1}\right)$ at the next point in the warp path. Now, Equation (4.41) can be rewritten as:

$$\begin{aligned} & X_{N-1}^\star \\ & = arg \max_{x_0,\cdots,x_k,\cdots,x_{N-1}} \left\{\left(c_0\left(x_0\right) \wedge \mu_{Match}(w_1)\right) \wedge \cdots \wedge \left(c_{N-1}\left(x_{N-1}\right) \wedge \mu_{Match}(w_N)\right)\right\} \\ & \qquad = arg \max_{x_0,\cdots,x_k,\cdots,x_{N-1}} \left\{ \bigwedge_{k=0}^{N} \left(\left(c_k(x_k)\right) \wedge \mu_{Match}(w_{k+1})\right) \right\} \end{aligned} \tag{4.43}$$

or, expressing in terms of goodness of decision $\mu_D$,

$$\begin{aligned} & \mu_D\left(x_0^\star, \cdots, x_k^\star, \cdots, x_{N-1}^\star | w_0 = (0,0), w_N = (M,N)\right) \\ & \qquad = \max_{x_0,\cdots,x_k,\cdots,x_{N-1}} \left\{ \bigwedge_{k=0}^{N} \left(c_k(x_k) \wedge \mu_{Match}(w_{k+1})\right) \right\} \end{aligned} \tag{4.44}$$

Comparing Equation (4.44) to Equation (4.30), it can be seen that the problem of finding optimal warp path for a fuzzy template is a special case of multistage decision making under Bellman and Zadeh framework. Here, fuzzy constraints $\left(\mu_{C_k}(x_k)\right)$ are replaced by crisp constraints $\left(c_k(x_k)\right)$ and a fuzzy goal is the same for each stage (i.e. $\mu_{Match}(w_{k+1}) \equiv \mu_{G_{K+1}}(s_{k+1})$).

Making use of the constraint on the final point on warp path $w_N = (M, N)$, equation (4.44) can be written as:

$$
\begin{aligned}
&\mu_D \left(x_0^\star, \cdots, x_k^\star, \cdots, x_{N-1}^\star | w_0 = (0,0), w_N = (M,N)\right) \\
&\qquad = \max_{x_0,\cdots,x_k,\cdots,x_{N-1}} \left\{ \bigwedge_{k=0}^{N-1} \left(c_k(x_k) \wedge \mu_{Match}(w_{k+1})\right) \right\} \\
&\qquad = \max_{x_0,\cdots,x_k,\cdots,x_{N-1}} \left\{ \left( \bigwedge_{k=0}^{N-2} \left(c_k(x_k) \wedge \mu_{Match}(w_{k+1})\right) \right) \right. \\
&\qquad\quad \left. \wedge \left(c_{N-1}\left(x_{N-1}\right) \wedge \mu_{Match}\left(w_N = \left(i_{N-1} + x_{N-1},\ N\right)\right)\right) \right\} \\
&\qquad = \max_{x_0,\cdots,x_k,\cdots,x_{N-2}} \left\{ \left\{ \bigwedge_{k=0}^{N-2} \left(c_k(x_k) \wedge \mu_{Match}(w_{k+1})\right) \right\} \right. \\
&\qquad\quad \left. \wedge \max_{x_{N-1}} \left\{c_{N-1}\left(x_{N-1}\right) \wedge \mu_{Match}\left(w_N = f\left(w_{N-1}, x_{N-1}\right)\right)\right\} \right\} \\
&\qquad = \max_{x_{N-1}} \left\{ \max_{x_0,\cdots,x_k,\cdots,x_{N-2}} \left\{ \bigwedge_{k=0}^{N-2} \left(c_k(x_k) \wedge \mu_{Match}(w_{k+1})\right) \right\} \right. \\
&\qquad\quad \left. \wedge \left\{c_{N-1}\left(x_{N-1}\right) \wedge \mu_{Match}\left(f\left(w_{N-1}, x_{N-1}\right)\right)\right\} \right\} \\
&\quad = \max_{x_{N-1}} \left\{ \mu_D \left(x_0^\star, \cdots, x_k^\star, \cdots, x_{N-2}^\star | w_0 = (0,0), w_{N-1} = f\left(w_{N-2}, x_{N-2}\right)\right) \right. \\
&\qquad\quad \left. \wedge \left\{c_{N-1}\left(x_{N-1}\right) \wedge \mu_{Match}\left(f\left(w_{N-1}, x_{N-1}\right)\right)\right\} \right\} \qquad (4.45)
\end{aligned}
$$

where,

$$
\begin{aligned}
&\mu_D\left(x_0^\star, \cdots, x_k^\star, \cdots, x_{N-2}^\star | w_0 = (0,0), w_{N-1} = f\left(w_{N-2}, x_{N-2}\right)\right) \\
&\qquad = \max_{x_0,\cdots,x_k,\cdots,x_{N-2}} \left\{ \bigwedge_{k=0}^{N-2} \left(c_k(x_k) \wedge \mu_{Match}(w_{k+1})\right) \right\} \\
&\qquad = \max_{x_0,\cdots,x_k,\cdots,x_{N-2}} \left\{ \left( \bigwedge_{k=0}^{N-3} \left(c_k(x_k) \wedge \mu_{Match}(w_{k+1})\right) \right) \right. \\
&\qquad\qquad \left. \wedge \left(c_{N-2}\left(x_{N-2}\right) \wedge \mu_{Match}\left(w_{N-1} = f\left(w_{N-2}, x_{N-2}\right)\right)\right) \right\} \qquad (4.46)
\end{aligned}
$$

The above equation may be iterated backwards to derive the following recursive relationship for any stage $v$:

$$
\begin{aligned}
&\mu_D\left(x_0^\star, \cdots x_v^\star | w_0 = (0,0), w_{v+1} = f\left(w_v, x_v\right)\right) \\
&\qquad = \max_{x_v} \left\{ \mu_D\left(x_0^\star, \cdots, x_k^\star, \cdots, x_{v-1}^\star | w_0 = (0,0), w_v = f\left(w_{v-1}, x_{v-1}\right)\right) \right. \\
&\qquad\qquad \left. \wedge \left(c_v\left(x_v\right) \wedge \mu_{Match}\left(f\left(w_v, x_v\right)\right)\right) \right\} \qquad (4.47)
\end{aligned}
$$

Let $X_v^\star = x_0^\star, \cdots, x_k^\star, \cdots, x_v^\star$; since $w_{v+1} = (i_{v+1},\ v+1)$, Let

$$\mu_D\left(x_0^\star, \cdots, x_k^\star, \cdots, x_v^\star | w_0 = (0,0), w_{v+1} = (i_{v+1},\ v+1)\right) = \mu_{D_{v+1}}\left(X_v^\star | i_{v+1}\right)$$

Then Equation (4.47) can be rewritten as

$$
\begin{aligned}
&\mu_{D_{v+1}}\left(X_v^\star | i_{v+1}\right) \\
&\qquad = \max_{x_v} \left\{ \mu_{D_v}\left(X_{v-1}^\star | i_v\right) \wedge \left(c_v\left(x_v\right) \wedge \mu_{Match}\left(i_{v+1},\ v+1\right)\right) \right\} \\
&\qquad = \max_{x_v} \left\{ \mu_{D_v}\left(X_{v-1}^\star | i_v\right) \wedge \left(c_v\left(x_v\right) \wedge \mu_{i_{v+1}}\left(R\left(v+1\right)\right)\right) \right\} \qquad (4.48)
\end{aligned}
$$

Since $i_{v+1} = i_v + x_v \Rightarrow i_v = i_{v+1} - x_v$, $\mu_{D_v}\left(X^\star_{v-1}|i_v\right) = \mu_{D_v}\left(X^\star_{v-1}|(i_{v+1} - x_v)\right)$and Equation (4.48) becomes:

$$\mu_{D_{v+1}}\left(X^\star_v|i_{v+1}\right) = \underset{x_v}{max}\left\{\mu_{D_v}\left(X^\star_{v-1}|(i_{v+1} - x_v)\right) \wedge \left(c_v\left(x_v\right) \wedge \mu_{i_{v+1}}\left(R\left(v+1\right)\right)\right)\right\} \tag{4.49}$$

The above equations can now be used to find the optimal warp path incrementally by applying dynamic programming. This is because, at any stage $v$, the above equation finds the optimal warp path that maximizes the degree of match $\mu_{D_{v+1}}\left(X_v|w_{v+1}\right)$ for a warp path that ends at $w_{v+1} = \left(i_{v+1}, v+1\right)$. For this, it uses the degrees of match $\mu_{D_v}\left(X^\star_{v-1}|w_v\right)$ for optimal paths ending in $w_v = \left(i_v, v\right)$ computed during the previous stage.

When starting at $v = 0$, using the initial condition $w_0 = (0,0)$, the degree of match of warp paths ending at $w_0$ can be computed as $\mu_{D_0}(0) = \mu_{Match}(0,0) = \mu_0\left(R(0)\right)$. Then for $v = 0$, the optimal warping paths that end at the next stage is:

$$\mu_{D_1}\left(X^\star_0|i_1\right) = \underset{x_0}{max}\left\{\mu_0\left(R(0)\right) \wedge \left(c_0\left(x_0\right) \wedge \mu_{i_1}\left(R\left(1\right)\right)\right)\right\} \tag{4.50}$$

Then, using (4.49), $\mu_{D_1}\left(X^\star_1|i_1\right), \cdots, \mu_{D_N}\left(X^\star_1|i_N\right)$ can be found incrementally. Because of the end condition $w_N = (M, N)$, for $v = N - 1$, the only warp path of interest is the one that ends at $N$, i.e. only $\mu_{D_{N-1}}\left(X^\star_{N-1}|i_N = M\right)$ needs to be found. Once $x^\star_{N-1}$ that maximizes $\mu_{D_{N-1}}\left(X^\star_{N-1}|i_N = M\right)$ is found, using the back tracking relation $i_v = i_{v+1} - x_v$, optimal warp path and optimal state transition choices $x^\star_{N-1}, \cdots, x^\star_0$ can be computed. The degree of match for the optimal warp path $\mu_{Match_N}\left(W^\star\right) = \mu_{D_{N-1}}\left(X^\star_{N-1}|M\right)$ can be used as a measure of the degree to which a fuzzy shape template matches an RMS waveform. Hence, the degree of match for the optimal warp path can be used to assign shape labels to RMS waveforms.

The optimal warp path $W^{\star}$ can also be used to estimate shape parameters. This will be explained in Section 4.5.

### 4.4.3 Fuzzy DTW Based RMS Shape Classifier

Similar to the DTW based RMS shape classifier (Section 4.3.2), a fuzzy DTW based RMS shape classifier (FDTW classifier) can be designed. FDTW classifier uses fuzzy shape templates $T_{Fuzzy_s}$ to detect specific shapes $Shape_s$. For each fuzzy template $T_{Fuzzy_s}$, optimal warp paths $W^{\star}_{T_{Fuzzy_s}}$ and degree of match cost $Match_s = \mu_{Match_N}\left(W^{\star}_{T_{Fuzzys}}\right)$ are first computed using the method outlined in Section 4.4.2.2. Then, a simple FDTW based classifier $C_{Shape}$ assigns the shape label $Shape_s$ to the RMS waveform $X$ such that, fuzzy shape template $T_{Fuzzy_s}$ has the maximum degree of match over the warp path:

$$C_{Shape}(X) = arg\ \underset{s}{max}\left\{Match_s\right\},\ 0 < s \leqslant N_s \tag{4.51}$$

It is possible that none of the shape templates match the shape observed in RMS waveform. However, equation 4.51 will assign a shape label even when none of the shape templates match. To avoid this scenario, a shape label '*Unknow*' is assigned if $\underset{s}{max}\left\{Match_s\right\} < Th_L$, where $Th_L$ is a threshold for degree of match over warp path.

### 4.4.4 Time and Space Complexity of Using FDTW

Time and memory complexity of using FDTW technique is similar to that of DTW technique (Section 4.3.3). This is because, at each stage, the DTW technique tries to compute a warp path with least incremental cost (Equation 4.14), while FDTW tries to compute the warp path with the best incremental match (Equation 4.49). The number of choices are dependent on the size of shape templates $|T|$ in

Table 4.1: Generic features

| Generic feature name | Expression |
|---|---|
| Maximum | $p_{max} = max\left(X\right)$ |
| Minimum | $p_{min} = min\left(X\right)$ |
| Mean | $p_{mean} = \frac{1}{N}\sum_{i=0}^{N-1} x_i$ |
| Start value | $p_{start} = x_0$ |
| End value | $p_{end} = x_N$ |
| Range | $p_{rng} = p_{max} - p_{min}$ |
| Percent change | $p_{pcnt} = \frac{p_{rng}}{p_{min}}$ |

the case of DTW, and $\left|T_{Fuzzy}\right|$in the case of FDTW technique. Similar to the DTW based shape template matching, Since $\left|T_{fuzzy}\right| \ll |X|$, $|X| * \left|T_{Fuzzy}\right| \approx c * |X|$, this makes the time and memory complexity linear $O\left(N\right)$.

## 4.5 RMS Waveform Feature Estimation

Shape labels assigned using DTW or FDTW techniques just by themselves are not enough to identify the cause of power system event. Information such as magnitude and duration of the event is needed to determine the cause of an event. For example, a 'dip' shape in RMS voltage waveform can be caused by overcurrents, arcing and loads switching on. Knowing that the size of the dip was large will help to narrow down the possible cause (e.g. overcurrent event).

Two types of features are extracted from RMS waveforms: 1. Generic features that do not depend on the shape of RMS waveform and 2. Shape features that are specific to the shape detected in RMS waveform. Generic features are extracted using basic operations on RMS waveforms. Table 4.1 lists all the generic features extracted from an RMS waveform $X = \{x_0, \cdots, x_{N-1}\}$. Shape feature extraction is more involved, and is a two step process. First, the optimal warp path that

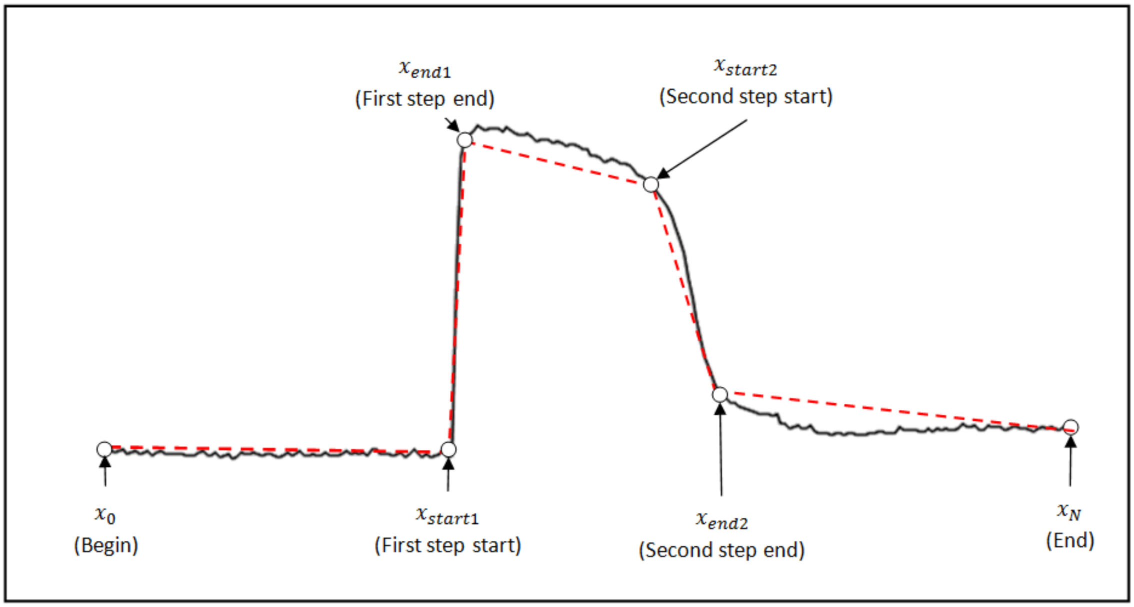


Figure 4.9: Pivot points for sample motor start shape

was computed during shape detection is used to find 'pivot' points. Pivot points correspond to points on RMS waveform where the warp path transitions from key points on shape template to another. Then, shape features are computed using these pivot points.

### 4.5.1 *Identifying Pivot Points on RMS Waveforms*

Pivot points are points on RMS waveform that show considerable changes in the slope of RMS waveform. These help define the shape observed in the waveform. For example, Figures 4.9, 4.10 show a possible set of pivot points (represented as small circles) for a motor start shape and step-up shape respectively. The number of pivot points needed to represent a shape, and the the location of these pivot points depend on the shape observed in RMS waveform. The shape of an RMS waveform is first determined using either DTW or FDTW technique. Then, the shape template that best matched the shape observed in RMS waveform and the corresponding optimal warp path are used to determine pivot points.

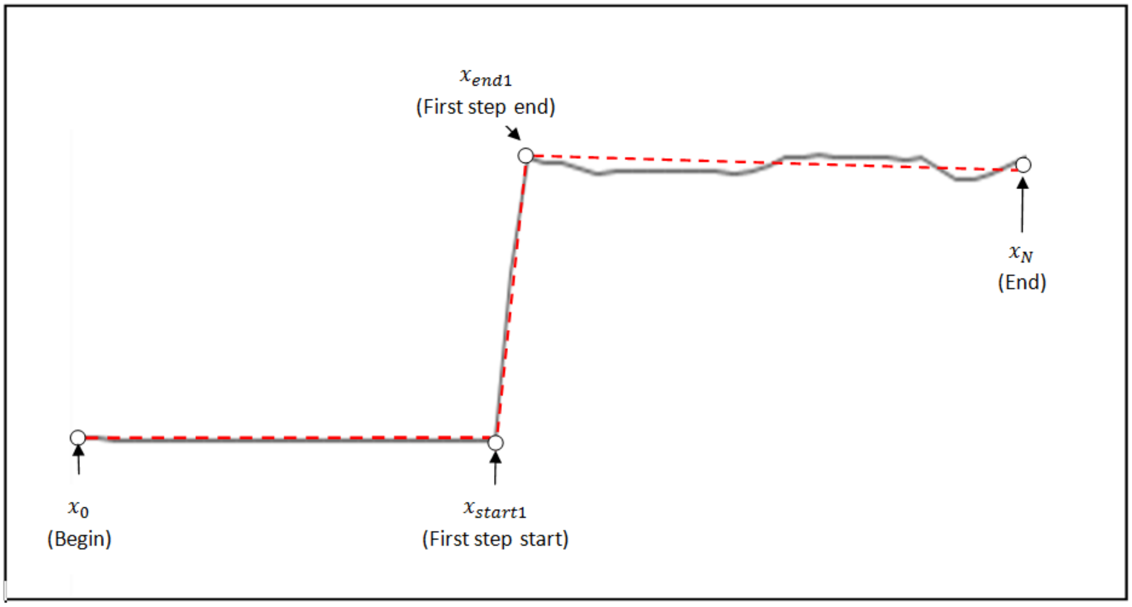


Figure 4.10: Pivot points for sample step-up shape

For the power system event classification problem, a majority of the shapes observed on RMS waveforms can be represented using either four or six pivot points. For example, let $X = \{x_0, \cdots, x_N\}$ represent an RMS current waveform. A sample motor start shape (Figures 4.9) can be described by the following sequence: *"Flat* region between $x_0$ and $x_{start1}$, *precipitous* upward step change between $x_{start1}$ and $x_{end1}$, *flat* region between $x_{end1}$ and $x_{start2}$, *slow* downward step change between $x_{start2}$ and $x_{end2}$, finally a flat region between $x_{end2}$ and $x_{N-1}$". Similarly, a sample step-up change caused by load switching (Figures 4.10) can be described by the following sequence: *"Flat* region between $x_0$ and $x_{start1}$, *precipitous* upward step change between $x_{start1}$ and $x_{end1}$, finally a flat region between $x_{end1}$ and $x_{N-1}$". Similar sequences can be be constructed for other shapes such as dips, step down and inrushes. These sequences illustrate how pivot points can be used to define the shape observed in RMS waveform.

For a RMS waveform $X = \{x_0, \cdots, x_N\}$, the start and end points $x_0$, $x_N$ are already known. Then, it suffices to determine the pivot points of the first step ($x_{start1}$, $x_{end1}$) for step up and step down shapes. For motor starts, inrushes, dips and bumps, the pivot points of the first step ($x_{start1}$, $x_{end1}$) and the pivot points of the second step ($x_{start2}$, $x_{end2}$) need to be determined. For a shape template $T = \left\{t_1, \cdots, t_{|T|}\right\}$, a template pivot point $p$ can be defined in terms of state transition $p = (\mathbf{t_s} \rightarrow t_e)$ or $p' = (t_s \rightarrow \mathbf{t_e})$. $p = (\mathbf{t_s} \rightarrow t_e)$ represents a point $w_p$ in the optimal warp path $W^\star = \{w_0, \cdots, w_N\}$ where the warp path starts to transition from a template point $t_s$ to $t_e$ such that, $w_p = \left(t_s,\ x_p\right)$ and $w_{p+1} = \left(t_e,\ x_{p+1}\right)$. Similarly, $p' = (t_s \rightarrow \mathbf{t_e})$ represents a point $w_{p'}$ in the optimal warp path $W^\star = \{w_0, \cdots, w_N\}$ where the warp path finishes to transition from a template point $t_s$ to $t_e$ such that, $w_{p'} = \left(t_e,\ x_{p'}\right)$ and $w_{p'-1} = \left(t_e,\ x_{p'-1}\right)$. Here, the template $T = \left\{t_1, \cdots, t_{|T|}\right\}$ is used to represent both crisp and fuzzy shape templates. Without loss of generality, $T = \left\{t_1, \cdots, t_{|T|}\right\}$ can be substituted by $T = \left\{\mu_1, \cdots, \mu_{|T|}\right\}$.

Figure 4.11 shows an optimal warp path computed using a sample motor start RMS waveform $X = \{x_0, \cdots, x_{14}\}$ for a fuzzy shape template:

$$T_{motor} = \{\mu_1 = \mu_L,\ \mu_2 = \mu_H, \mu_3 = \mu_M, \mu_4 = \mu_L\}$$

Pivot points $P$ for a shape template $T$ are determined heuristically and do not change. Let the motor start shape be described by the following pivot points:

$$P = \{p_0, p_{start1}, p_{end1}, p_{start2}, p_{end2}, p_N\}$$

where $p_0$is the starting point, $p_N = p_{14}$ is the ending point, $p_{start1} = (\boldsymbol{\mu_L} \rightarrow \mu_H)$, $p_{end1} = (\mu_L \rightarrow \boldsymbol{\mu_H})$, $p_{start2} = (\boldsymbol{\mu_H} \rightarrow \mu_M)$ and $p_{end3} = (\mu_M \rightarrow \boldsymbol{\mu_L})$. Based on this state transition information, and an optimal warp path $W^\star = \{w_0, \cdots, w_{14}\}$ ( Table

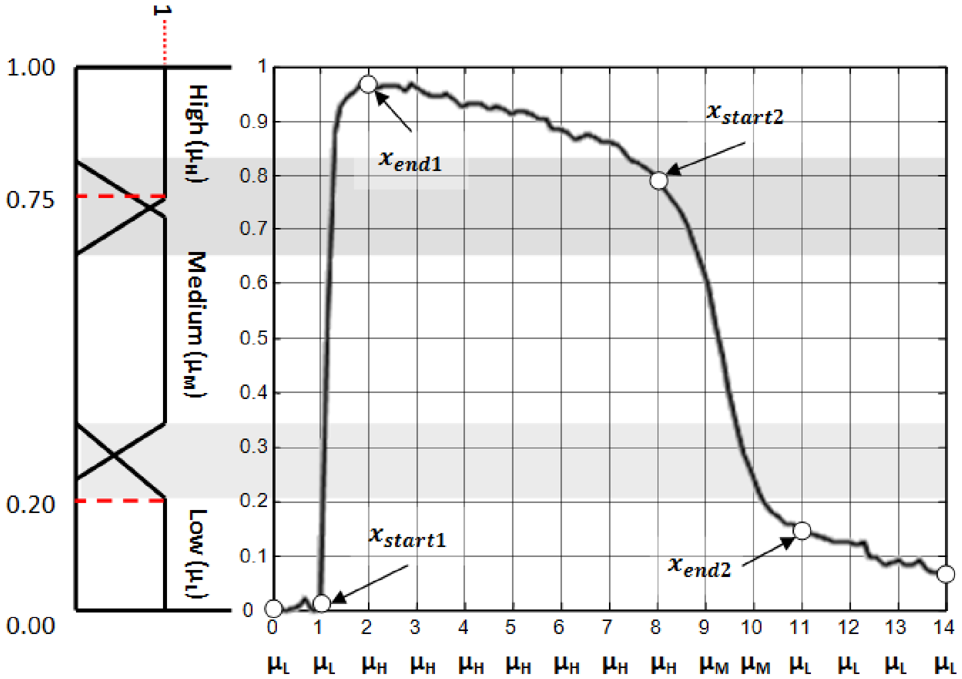


Figure 4.11: Computing pivot points using fuzzy motor start shape template

Table 4.2: Optimal warp path for sample RMS waveform in Figure 4.11

| Warp path index | Warp path point | Computed pivot point | Comments |
|---|---|---|---|
| $w_0$ | $(\mu_L, x_0)$ | $x_0$ | Start point |
| $w_1$ | $(\mu_L, x_1) \rightarrow$ | $x_{start1}$ | First step start |
| $w_2$ | $\rightarrow (\mu_H, x_2)$ | $x_{end1}$ | First step end |
| $w_3$ | $(\mu_H, x_3)$ | - | |
| $w_4$ | $(\mu_H, x_4)$ | - | |
| $w_5$ | $(\mu_H, x_5)$ | - | |
| $w_6$ | $(\mu_H, x_6)$ | - | |
| $w_7$ | $(\mu_H, x_7)$ | - | |
| $w_8$ | $(\mu_H, x_8) \rightarrow$ | $x_{start2}$ | Second step start |
| $w_9$ | $(\mu_M, x_9)$ | - | |
| $w_{10}$ | $(\mu_M, x_{10})$ | - | |
| $w_{11}$ | $\rightarrow (\mu_L, x_{11})$ | $x_{end2}$ | Second step end |
| $w_{12}$ | $(\mu_L, x_{12})$ | - | |
| $w_{13}$ | $(\mu_L, x_{13})$ | - | |
| $w_{14}$ | $(\mu_L, x_{14})$ | $x_{14}$ | End point |

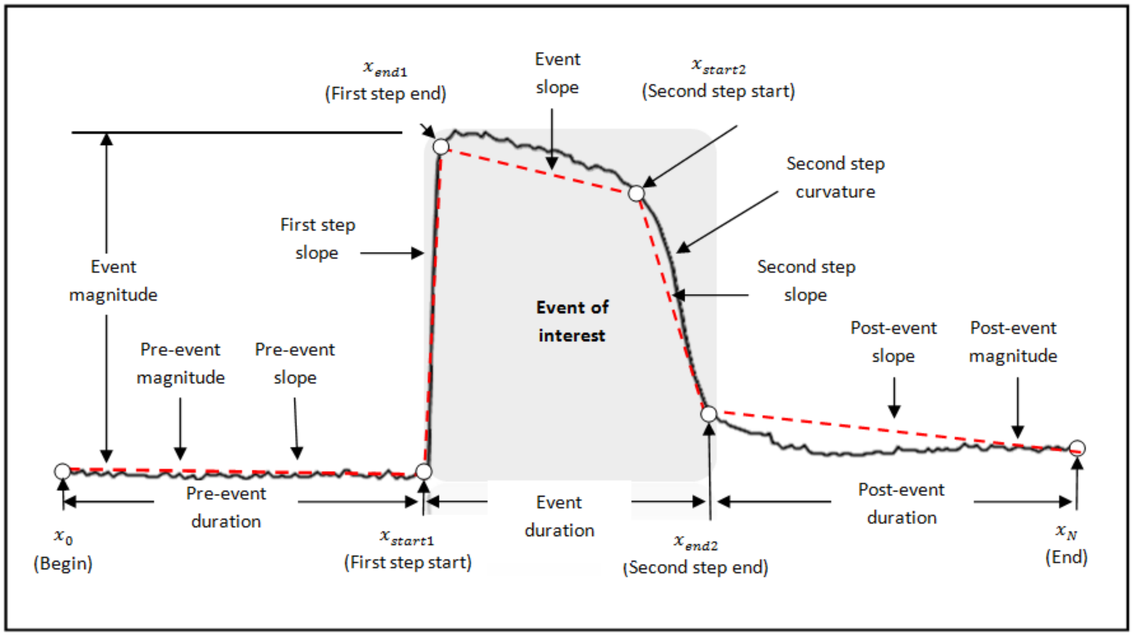


Figure 4.12: Subset of shape features for a sample motor start shape

4.2), pivot points $X_p$ on the RMS waveform $X$ can be easily computed as:

$$X_p = \{x_0, x_{start1} = x_1, x_{end1} = x_2, x_{start2} = x_8, x_{end2} = x_{11}, x_{14}\}$$

Once pivot points on RMS waveform are identified, shape features can be computed.

### 4.5.2 *Computing Shape Features for RMS Waveforms*

Shape features quantify important characteristics of shapes observed in RMS waveform. These features help an expert classify the RMS waveform. Figure 4.12 shows how shape features can be used to describe a motor start shape. Table 4.3 shows how pivot points can be used to calculate shape features for different shapes. These shape features were chosen based on experience gained by visually analyzing RMS waveforms. RMS shape features, RMS shape labels and generic features extracted from RMS waveforms serve as inputs to fuzzy expert system

Table 4.3: Shape features ($x_{Shape}$)

| Applicable shapes | Shape feature name | Expression | Comment |
|---|---|---|---|
| All shapes | Shape label | $c_{shape}$ | Shape label assigned to RMS waveform |
| All shapes | Pre-event duration | $p_{dur-pre} = start1$ | - |
| | Pre-event magnitude | $p_{pre} = \frac{1}{start1} \sum_{i=0}^{start1-1} x_i$ | Average pre-event magnitude |
| | Pre-event slope | $p_{sl-pre} = \frac{x_0 - x_{start1}}{start1}$ | - |
| | First step slope | $p_{sl-step1} = \frac{x_{start1} - x_{end1}}{end1 - start1}$ | - |
| Shapes with two steps: motor start, inrush, bump and dip | Event duration | $p_{dur-event} = end2 - end1$ | - |
| | Event magnitude | $p_{event} = \frac{1}{start2 - end1} \sum_{i=end1}^{start2-1} x_i$ | Average event magnitude |
| | Event slope | $p_{sl-event} = \frac{x_{end1} - x_{start2}}{start2 - end1}$ | - |
| | Second step slope | $p_{sl-step1} = \frac{x_{start2} - x_{end2}}{end2 - start2}$ | - |
| | Post-event duration | $p_{dur-post} = N - end2$ | - |
| | Post-event magnitude | $p_{post} = \frac{1}{N - end2} \sum_{i=end2}^{N-1} x_i$ | Average post-event magnitude |
| | Post-event slope | $p_{sl-post} = \frac{x_{end2} - x_N}{N - end2}$ | - |
| Shapes with one step: step-up and step-down | Post-event duration | $p_{dur-pre} = N - end1$ | - |
| | Post-event magnitude | $p_{post} = \frac{1}{N - end1} \sum_{i=end1}^{N-1} x_i$ | Average post-event magnitude |
| | Post-event slope | $p_{sl-post} = \frac{x_{end1} - x_N}{N - end1}$ | - |

classifier.

Extracting RMS waveform shape features ($x_{Shape}$) and generic features ($x_{Geenric}$) is one of the sub-problems of power system event classification problem (explained in Section 4.2 of Chapter 2). Waveform data instances may contain one or more sub-events. Hence RMS waveform feature extraction is applied to RMS waveforms corresponding to each sub-event of a cycle-by-cycle waveform.

## 4.6 Chapter Summary

In this chapter, RMS waveform shape detection problem was introduced and its relevance to power system event detection was explained. Then two methods, DTW based RMS waveform shape detection method and FDTW based RMS waveform shape detection methods were derived. Finally, an RMS waveform feature extraction method was outlined using pivot points.

# 5. EVENT SPECIFIC FEATURE EXTRACTION

## 5.1 Introduction

RMS waveform based shape features and generic features are sufficient for identifying a majority of power system events. This is especially true for normal power system operations. However, there are a number of abnormal power system events where RMS waveform analysis alone is not sufficient:

1. Some event characteristics may not be observable on RMS waveforms and may require analysis of high speed waveforms. It is possible to design algorithms to detect the presence or absence these features. For example, Figure 5.1 shows RMS current waveform for phase C (plot (a)) and high speed differenced current waveform [77] for phase C (plot (c)) captured during an event caused by arcing. Just looking at the RMS waveform in plot (a), it would be hard to determine the cause of the event. Only a small 'bump' is observed, and it could be caused by small loads. However, zooming in on the differenced current waveform (plot (d)) clearly reveals arcing signatures characterized by 'flat' regions near zero crossing of current waveforms (dotted circle in plot (d)).

2. Certain event characteristics may be observable in RMS waveforms but, may not fit into the general category of generic features or shape features. These features may help improve the classification accuracy for some event categories. For example, Figure 5.2 shows waveforms from a three-phase capacitor switching on. Plots (c) and (d) show classic capacitor switching signatures on phases B and C, characterized by large step decrease in reactive

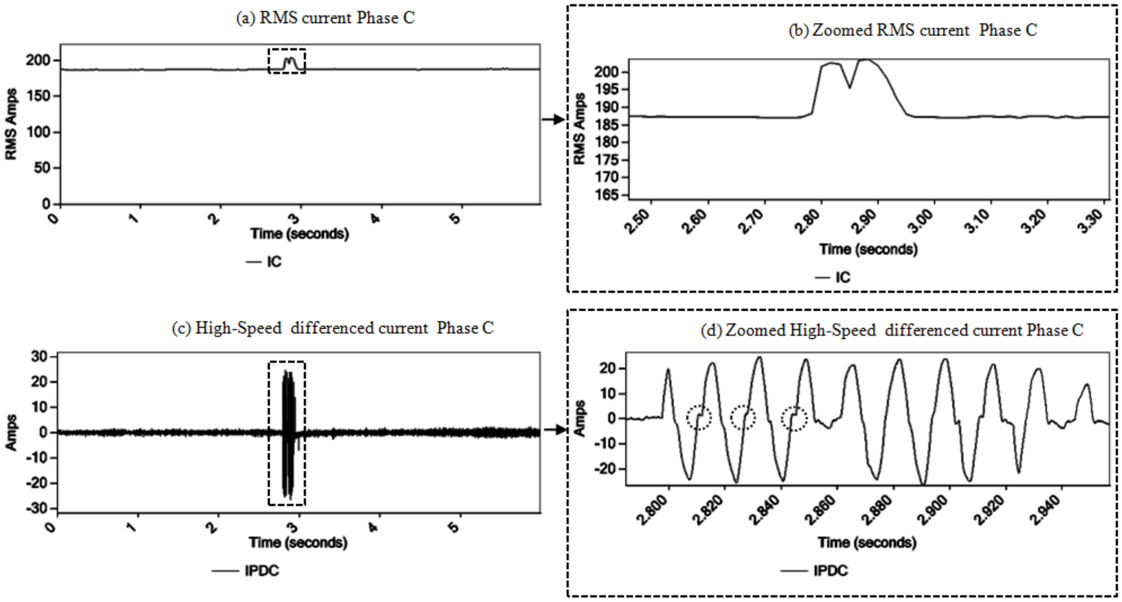


Figure 5.1: Example waveforms caused by arcing

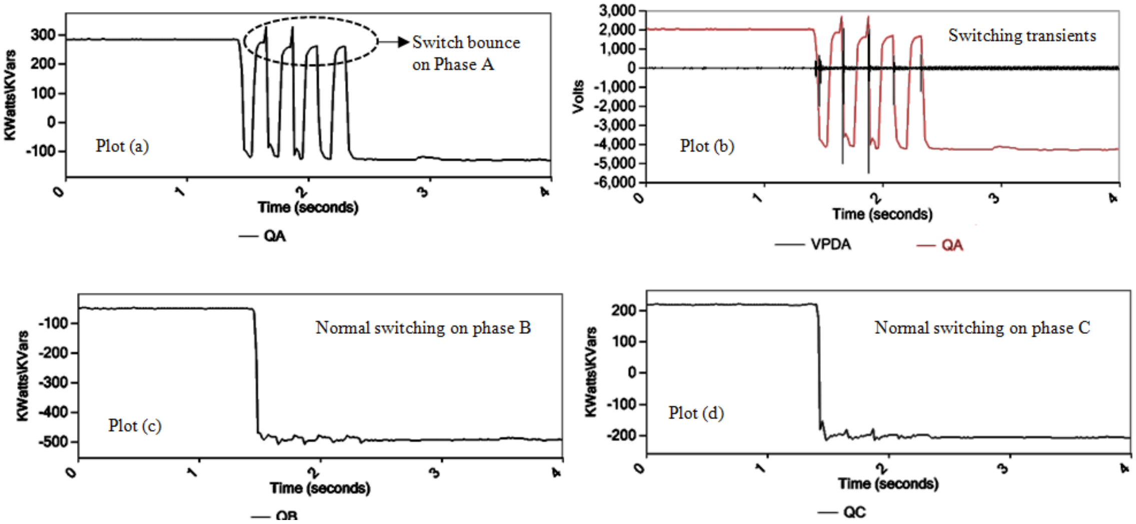


Figure 5.2: Example waveforms caused by capacitor switch bounce

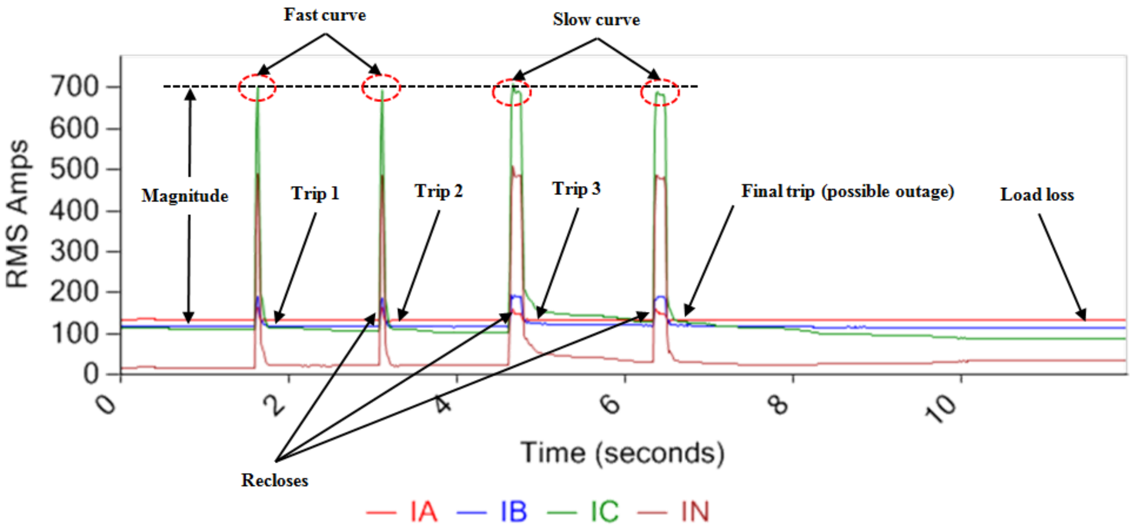


Figure 5.3: Waveforms from an overcurrent fault that caused a single-phase recloser to trip

power QB and QC. However, Plot (a) shows switch bounce on phase A, characterized by one or more stepping up and down in reactive power QA. The evidence of switch bounce is further supported by voltage transients seen on phase A (plot (b)) as phase A switch makes and breaks contact during switch bounce. Capacitor switch bounce is an abnormal condition and results in voltage transients that degrade power quality. Switch bounce may also result in deterioration of switch contacts. This example shows that shape templates that look for step up or step down shapes may not be enough to detect the switch bounce shape observed in plot (a). An algorithm specific to detecting switch bounce pattern is needed to extract feature indicative of switch bounce. A classifier can use switch bounce feature, and other evidence such as the RMS shape features and voltage transients to identify the cause of the event.

3. One of the requirements for the power system event classification algorithm

is the ability to extract event specific features that aid in locating the cause of an event. These features may or may not influence classification results. These features may require dedicated algorithms for analyzing and extracting them from high-speed or RMS waveform data. For example, Figure 5.3 shows RMS current waveforms corresponding to a Phase C to ground overcurrent fault. A classification algorithm can label this event as an overcurrent fault on monitored feeder involving phase C and ground. However, this classification information will not be of much help in locating the fault. This is because, a utility personal will have to look for all feeder sections where phase C is available. In some cases, this could be equivalent to searching the entire feeder. Visual analysis of RMS waveform in Figure 5.3 indicates that a single-phase recloser tripped and reclosed three times before finally tripping and causing an outage (since the part of the feeder down stream of the device would have been taken out of service). Providing utility personnel with extra information such as the device that operated, fault magnitude, fault duration and load loss would help reduce the search area for fault location [99]. This is because, utility personnel can limit their search to locations downstream of a protective device. Further, phase and fault current information can be used to further reduce the search area if a short circuit model of the feeder is available. Dedicated algorithms are needed to extract such event specific information.

The rest of this chapter focuses on extracting features specific to arcing events, abnormal capacitor operations and overcurrent events.

## 5.2 Features Specific to Arcing Events

Arcing events on power distribution system may have multiple causes, including but not limited to, downed power lines, trees contacting the power lines, and failing hardware on the power system. Arcing events are more difficult to detect since they involve relatively low current magnitudes when compared to current magnitudes measured during overcurrent faults. Arcing events may have current magnitudes anywhere between a few tens of amps to a few hundred amps. Most conventional overcurrent protection devices, such as fuses, reclosers, relays have time delays that prevent a temporary fault from causing an outage. Such a protection device de-energizes the power line, only if the overcurrent fault persists. Arcing faults seldom cause protective device operation, due to their low current magnitudes. Such overcurrent protection devices cannot distinguish a fault current from the levels of current ordinarily drawn by customers. Therefore, the line may remain energized even though dangerous arcing exists on the power line.

Arcing events can be broadly classified into three categories: line-to-line arcing, line-to-neutral arcing and series arcing. Some examples of line-to-line and line-to-neutral arcing include those caused by trees contacting power lines and broken power lines. Series arcing is caused by intermittent conductivity within a line. Series arcing may be caused by, but are not limited to, a faulty series switch, clamp or cutout. Electrical characteristics of line-to-line, line-to-ground and series arcing are distinct. Hence, it is possible to extract features that help to identify series and parallel arcing. Only line-to-line, line-to-ground arcing characteristics will be discussed in this dissertation.

### *5.2.1 Feature Extraction for Arcing Events*

Line-to-line or line-to-ground arcing are caused by an electrical breakdown of an otherwise non conducting medium such as air. The electrical break down may be caused by a number of factors such as degradation of insulating material, environmental factors such as ice, tree contact or animal contact. Arcing is typically characterized by sporadic, high energy, electrical discharges occurring near voltage peaks when the voltage is maximum. The arcs tend to extinguish themselves near near zero crossing of voltage waveforms because the voltage is not high enough to sustain an arc. It is possible to recognize parallel arcing signature by visual analysis of event waveforms. Hence, it is also possible to extract features that represent arcing signatures. Classification algorithms can use these features to detect line-to-ground and line-to-line arcing [67].

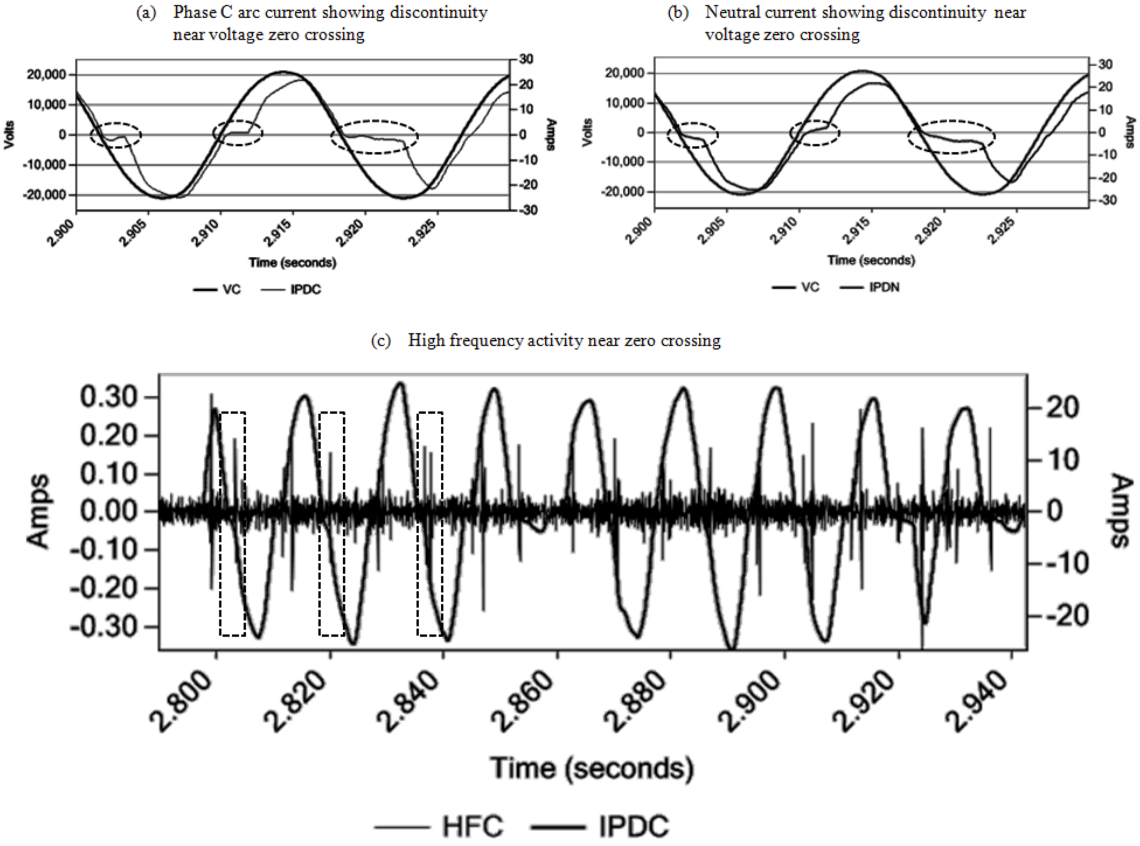
#### *5.2.1.1 Single line-to-ground arcing features*

Figure 5.4 shows example waveforms recorded during a phase C to ground arcing event. The following are the characteristic signatures of a single line-to-ground arcing:

1. Peaks of arcing current are close to voltage peaks. Plot (a) shows phase C differenced current (IPDC). IPDC is an estimate of the arc current on phase C. The plot clearly shows the voltage peaks (VC) aligning with arc current peaks (IPD).

2. Arc currents are close to zero near voltage zero crossing. Plot (a) shows flat areas of discontinuity (dotted circles) near voltage zero crossing.

3. Neutral current closely follows phase current. Plot (b) shows differenced neutral current (IPDN). Plots (a) and plot (b) are almost identical.

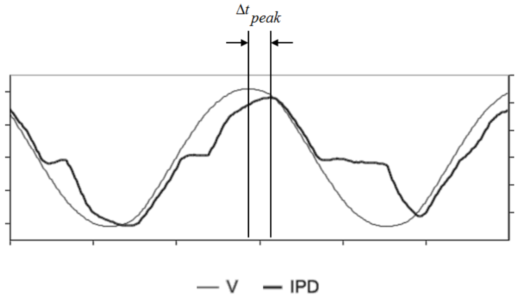
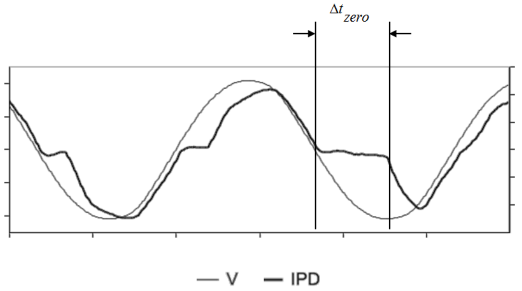

(a) Phase C arc current showing discontinuity near voltage zero crossing

(b) Neutral current showing discontinuity near voltage zero crossing

(c) High frequency activity near zero crossing

Figure 5.4: Signature of a single line-to-ground arcing event

Table 5.1: Event specific features $p_{Specific_{SLG-Arc}}$ for single line-to-ground arcing

| Feature name | Expression | Comment |
|---|---|---|
| Mean current peak offset | $\overline{\Delta t_{peak_{ph}}},\ ph \in \{A, B, C\}$ | Calculates the average time offset between voltage and current peaks<br><br> |
| Mean period of discontinuity | $\overline{\Delta t_{zero_{ph}}},\ ph \in \{A, B, C\}$ | Calculates the mean duration of the flat regions where arc current is near zero<br><br> |
| Peak arcing current | $MaxAbsIPD_{Ph} =$<br>$Max\left(\lvert IPD_{Ph}\rvert\right),$<br>$ph \in \{A, B, C\}$ | Peak values of differenced currents |
| Percentage phase to neutral difference | $PcntDiff_{ph} =$<br>$\frac{Max(\lvert IPD_{Ph}+IPD_N\rvert)}{MaxAbsIPD_{Ph}},$<br>$ph \in \{A, B, C\}$ | Maximum of absolute sample by sample difference between phase and neutral currents over maximum absolute phase current |
| High frequency activity | $BoolIsHFPresent_{ph},$<br>$ph \in \{A, B, C\}$ | Detects if high frequency activity was observed |

4. Arc current magnitudes tend to be low and involve high impedance. In the example shown in Figure 5.4, arc current peaks are near 25 Amps, while the voltage peaks are near 20kV, which is approximately equivalent to very high impedance of 800 ohms.

5. High frequency (greater than 2kHz) activity is observed during an arcing event, and such high frequency activity are predominantly observed near areas of discontinuity. Plot (c) shows high frequency current (HFC) obtained by high pass filtering phase C current. High frequency activity (dotted rectangles) can be seen during the arcing event.

Based on the above characteristics, an algorithm was designed to extract event specific features for single line-to-ground arcing (Table 5.1). The pseudo code for the algorithm is shown in Algorithm 1 (Figure 5.5). The algorithm extracts arcing features are only if the waveforms pass an initial screening for arcing in order to improve efficiency. First, the algorithm uses RMS shape features to eliminate waveforms that do not contain arcing related shapes. Then, it uses peak arcing current feature ($MaxAbsIPD_{Ph}$) to eliminate waveforms with peak current magnitudes that are either too high or too low. The algorithm uses percentage phase to neutral difference feature $PcntDiff_{ph}$ to eliminate waveforms further when the phase current and neutral current do not track each other closely. Once a waveform passes these tests, the algorithm analyzes each cycle of current waveforms to identify cycles that exhibited arcing characteristics. The algorithm computes the following for each cycle:

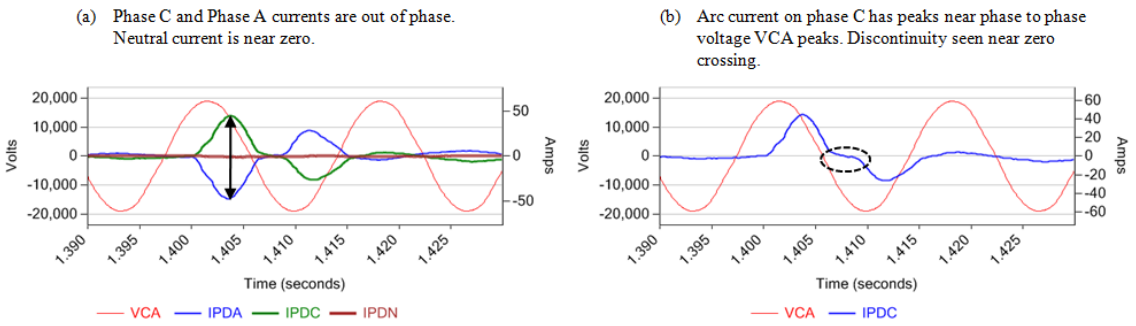
1. Peak value of high frequency current ($MaxAbsHFICyc$).

2. Peak differenced current magnitude ($MaxAbsIPDCyc$).

**Algorithm 1** Extracting single line-to-ground arcing features

```
Using input features RMS_Shape_IPD, RMS_Shape_I
Initialize output features Max_Abs_IPD,Pcnt_Diff
Initialize output features Delta_t_peak,delta_t_zero
Initialize output features Bool_Is_HF_Present
For each phase ph {
  if(RMS_Shape_IPD[ph] = ('Bump' or 'UnKnown')
      || RMS_Shape_I[ph] = ('Bump' or 'UnKnown')){
    Compute Max_Abs_IPD[ph], Pcnt_Diff[ph]
    // For SLG arcing, Pcnt_Diff should be small and
    // Max_Abs_IPD should not be too high
    If(Pcnt_Diff[ph] < DiffThreshold &&
        Max_Abs_IPD[ph] < OverCurrentThreshold &&
        Max_Abs_IPD[ph] < ArcSensitvityThreshold) {
      For each cycle cyc {
        Compute MaxAbsIPDCyc[cyc]
        // High frequency current peak
        Compute MaxAbsHFICyc[cyc]
        // Check if the cycle could have been involved in arcing
        If(MaxAbsIPDCyc[cyc]/Max_Abs_IPD[ph] > RelativeThreshold) {
          // Add it to candidate arcing cycles
          ArcCycles.Add(cyc)
          // Offset between current and voltage peak
          Compute DeltaTPeak[cyc]
          // duration of current zero crossing
          Compute DeltaTZero[cyc]
        }
        else
          NonArcCycles.Add(cyc)
      }
    }
    Compute Delta_t_peak[ph] = Mean(DeltaTPeak[ArcCycles])
    Compute delta_t_zero[ph] = Mean(DeltaTZero[ArcCycles])
    Compute MaxArcHFI = Max(MaxAbsHFICyc[ArcCycles])
    Compute MeanNonArcHFI = Mean(MaxAbsHFI[NonArcCycles])
    // If high frequency content increased during candidate arcing cycles
    If(MeanNonArcHFI/MaxArcHFI < HFIRelativeThresh)
      Bool_Is_HF_Present[ph] = true
  }
}
```

Figure 5.5: Extracting single line-to-ground event


(a) Phase C and Phase A currents are out of phase. Neutral current is near zero.

(b) Arc current on phase C has peaks near phase to phase voltage VCA peaks. Discontinuity seen near zero crossing.

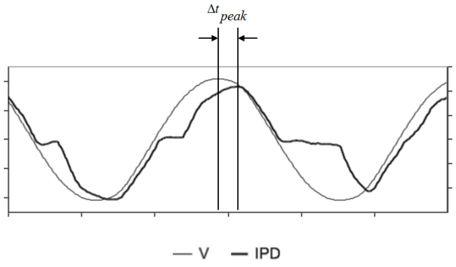
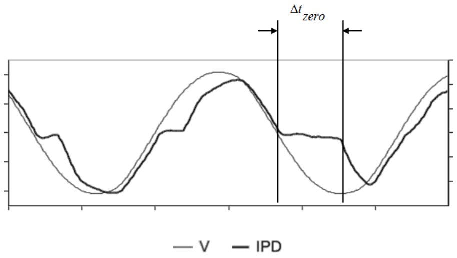
Figure 5.6: Signature of a line-to-line arcing event

The algorithm then chooses cycles with peak differenced current magnitudes close to the maximum current magnitude computed for the entire waveform. For these candidate arcing cycles, the algorithm computes the following:

1. The distance between the current peaks and voltage peaks ($\Delta t_{peak_{ph}}$).
2. The duration of the flat region near current zero crossings ($\Delta t_{zero_{ph}}$).

Using $\Delta t_{peak_{ph}}$ and $\Delta t_{zero_{ph}}$ values computed above; the algorithm computes the mean time offset between voltage and current peaks $\overline{\Delta t_{peak_{ph}}}$ and the mean duration of flat regions near current zero crossings ($\overline{\Delta t_{zero_{ph}}}$) for all candidate arcing cycles. Then the algorithm computes the ratio of the mean peak high frequency current during non-arcing cycles ($MeanNonArcHFI$) and the peak high frequency current during candidate arcing cycle ($MaxArcHFI$). The algorithm compares this ratio against a threshold, to verify if there were a significant increase in high frequency current during candidate arcing cycles. If the algorithm determines that there was an increase in high frequency current during arcing cycles, it sets the value of high frequency activity feature $BoolIsHFPresent_{ph}$ to true.

Table 5.2: Event specific features $p_{Specific_{L2L-Arc}}$ for line-to-line arcing

| Feature name | Expression | Comment |
|---|---|---|
| Mean current peak offset | $\overline{\Delta t_{peak_{ph1}}},\ ph1 \in \{A, B, C\}$ | Calculates the average time offset between phase-to-phase voltage and current peaks<br><br> |
| Mean period of discontinuity | $\overline{\Delta t_{zero_{ph1}}},\ ph1 \in \{A, B, C\}$ | Calculates the mean duration of the flat regions where arc current is near zero<br><br> |
| Peak arcing current | $MaxAbsIPD_{Ph1} = Max\left(\lvert IPD_{Ph1}\rvert\right),$<br>$ph1 \in \{A, B, C\}$ | Peak values of differenced currents |
| Percentage phase-to-phase difference | $PcntDiff_{(ph1,ph2)} = \frac{Max\left(\lvert IPD_{ph1}-IPD_{ph2}\rvert\right)}{Max(MaxAbsIPD_{Ph1},MaxAbsIPD_{Ph2})}$<br>$(ph1, ph2) \in \{(A, B), (B, C), (C, A)\}$ | Maximum of absolute sample by sample difference between phase-to-phase currents over maximum absolute phase currents |
| High frequency activity | $BoolIsHFPresent_{ph1},$<br>$ph1 \in \{A, B, C\}$ | Detects if high frequency activity was observed |

### 5.2.1.2 *Line-to-line arcing features*

Figure 5.6 shows example waveforms recorded during a phase C to A arcing event. The following are the characteristic signatures of a single line-to-line arcing:

1. Peaks of arcing current are close to phase-to-phase voltage peaks. Plot (b) shows phase C differenced current (IPDC) which is an estimate of the arc current on phase C. The plot clearly shows the phase-to-phase voltage peaks

(VCA) aligning with arc current peaks (IPD). This is similar to a single line-to-ground arcing event, except that the arc is sustained by the voltage across two-phases. This causes the arc current to peak near phase-to-phase voltage peaks.

2. Arc currents are close to zero near voltage zero crossing. Plot (b) shows flat areas of discontinuity (dotted circles) near voltage zero crossing. Again, this behavior is similar to a single line-to-ground arcing event.

3. Neutral current is near zero as one of the arcing phases provides the return path for the arc current. Plot (a) shows differenced neutral current (IPDN), and it is near zero.

4. Arc current magnitudes tend to be low and involve high impedance. In the example shown in Figure 5.6, arc current peaks are near 50 Amps. However, the phase-to-phase voltage peaks are near 20kV. This is approximately equivalent to a high impedance of $20kV/50 = 400\Omega$.

5. Arc current is seen on two-phases and they are out of phase with one another. This is because, one of the phases provides a return path for arc current. This behavior is different from single line-to-ground arc, but is expected.

Based on the above characteristics, an algorithm was designed to extract event specific features for line-to-line arcing (Table 5.2). The pseudo code for the algorithm (Algorithm 2, Figure 5.7) is similar to the algorithm used for single line-to-ground arcing. This is because, single line-to-ground and line-to-line arcing features are very similar. Calculations and the algorithm used to identify single line-to-ground arcing can be reused for identifying line-to-line arcing. Consider

**Algorithm 2** Extracting single line-to-line arcing features

```
Using input signals IPD, V, HFI
Using input features RMS_Shape_IPD, RMS_Shape_I
Initialize output features Max_Abs_IPD,Pcnt_Diff
Initialize output features Delta_t_peak,delta_t_zero
Initialize output features Bool_Is_HF_Present

For each phase (ph1,ph2) {
  if((RMS_Shape_IPD[ph1] = ('Bump' or 'UnKnown')
      || RMS_Shape_I[ph2] = ('Bump' or 'UnKnown'))
     && RMS_Shape_IPD[ph1]=RMS_Shape_IPD[ph2]){
    Compute Max_Abs_IPD[ph1], Pcnt_Diff[ph1, ph2]
    // For SLG arcing, Pcnt_Diff should be small and
    // Max_Abs_IPD should not be too high
    If(Pcnt_Diff[ph1,ph2] < DiffThreshold &&
        Max_Abs_IPD[ph1] < OverCurrentThreshold &&
        Max_Abs_IPD[ph1] < ArcSensitvityThreshold &&
        Max_Abs_IPD[Neutral]/Max_Abs_IPD[ph1] < NeutralThreshold) {
      For each cycle cyc {
        Compute MaxAbsIPDCyc[cyc]
        // High frequency current peak
        Compute MaxAbsHFICyc[cyc]
        // Check if the cycle could have been involved in arcing
        If(MaxAbsIPDCyc[cyc]/Max_Abs_IPD[ph1] > RelativeThreshold) {
          //compute phase to phase voltage for this cycle
          Compute V[ph1,ph2] = V[ph1]   V[ph2]
          // Add it to candidate arcing cycles
          ArcCycles.Add(cyc)
          // Offset between current and voltage peak
          Compute DeltaTPeak[cyc]
          // duration of current zero crossing
          Compute DeltaTZero[cyc]
        }
        else
          NonArcCycles.Add(cyc)
      }
    }
    Compute Delta_t_peak[ph1] = Mean(DeltaTPeak[ArcCycles])
    Compute delta_t_zero[ph1] = Mean(DeltaTZero[ArcCycles])
    Compute MaxArcHFI = Max(MaxAbsHFICyc[ArcCycles])
    Compute MeanNonArcHFI = Mean(MaxAbsHFI[NonArcCycles])
    // If high frequency content increased during candidate arcing cycles
    If(MeanNonArcHFI/MaxArcHFI < HFIRelativeThresh)
      Bool_Is_HF_Present[ph1] = true
  }
}
```

Figure 5.7: Extracting single line-to-line event

a single line-to-ground arcing event involving a phase $ph1$ and neutral $N$. Also, consider a line-to-line arcing event involving the phases $ph1$ and $ph2$. Algorithm 1 can be reused for extracting line-to-line arcing features after the following modifications:

1. Replacing the phase voltage $V_{ph1}$ in the single line-to-ground arcing algorithm with phase-to-phase voltage $V_{(ph1,ph2)}$ (Algorithm 2, line 24).
2. Replacing the neutral current $IPD_N$ with the negative of another phase current $-IPD_{ph2}$ when computing percentage phase-to-phase difference feature $PcntDiff_{(ph1,ph2)}$ (Table 5.2).
3. Verifying that the neutral current remained relatively low (Algorithm 2, line 16).

Except for the above difference, the algorithm used for extracting line-to-line arcing features $p_{Specific_{L2L-Arc}}$, is very similar to the algorithm used for extracting single line-to-ground arcing features.

## 5.3 Features Specific to Abnormal Capacitor Operations

Capacitors banks are used to provide voltage and VAR support on distribution system feeders. Capacitor bank terminals are connected to distribution feeders through mechanical switches. These mechanical switches are operated by controllers that connect or disconnect the capacitor banks based on factors such as time, voltage levels and temperature [100]. Most utilities employ three-phase switched capacitor banks with switches that connect or disconnect each phase. A distribution feeder can have one or more capacitor banks. Capacitor bank switching operations are one of the most common, normal power system events observed on the distribution feeder. Since capacitor banks are ubiquitous on power system feeders and they are also switched on and off on a regular basis; capacitor banks and capacitor bank switches are prone to failures. Capacitor bank related problems were also one of the most common abnormal power system events observed in the data used for this research (collected using the DFA platform).

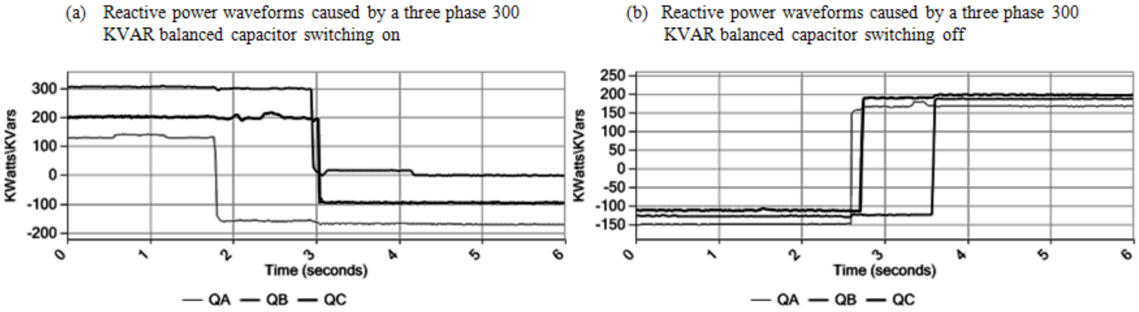
Manual analysis of data collected through the DFA platform indicated a variety of problems or failure in capacitor banks and components associated with switching of capacitor banks. Waveform data recorded during these problems showed distinct electrical characteristics that can be recognized. These electrical characteristics can be used to classify event data based on the underlying capacitor problem or failure mode [101]. Capacitor related problems that have been observed and have identifiable signatures include, unbalanced capacitor, capacitor VAR imbalance, capacitor switch restrike, capacitor switch bounce and arcing inside capacitor, switch or connection. Each of these abnormal capacitor problems will be discussed separately in the following subsections. For the purpose of feature extraction, abnormal capacitor operations are grouped in two categories: 1. capacitor problems that cause reactive power imbalance and 2. capacitor problems that cause voltage transients.

### *5.3.1 Capacitor Problems that Cause Reactive Power Imbalance*

During normal switching of three-phase capacitor banks, all three-phases switch on or off within a short interval of time. This results in either a decrease in reactive power (Q) on all three-phases after a capacitor switches on (Figure 5.8, plot (a)) or an increase in reactive power after a capacitor switches off (Figure 5.8, plot (b)).

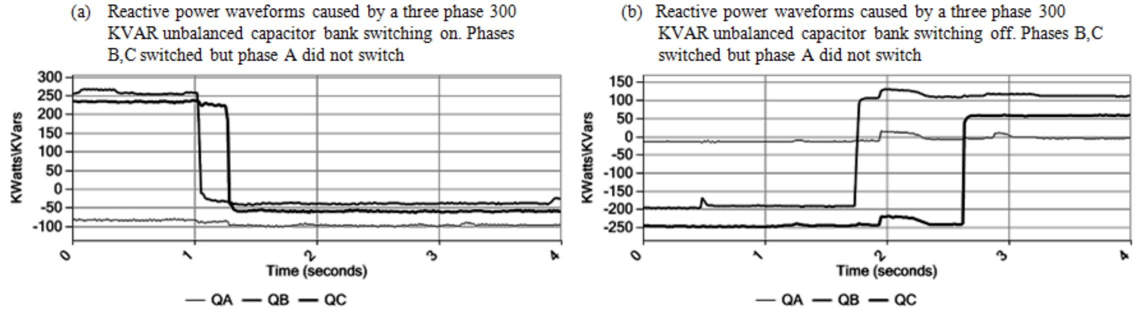
#### *5.3.1.1 Unbalanced capacitor operation*

Sometimes, due to factors such as faulty capacitor bank, faulty switch or a blown fuse on one of the phases, not all of the three-phases of a capacitor bank


(a) Reactive power waveforms caused by a three phase 300 KVAR balanced capacitor switching on

(b) Reactive power waveforms caused by a three phase 300 KVAR balanced capacitor switching off

Figure 5.8: Balanced capacitor bank switching


(a) Reactive power waveforms caused by a three phase 300 KVAR unbalanced capacitor bank switching on. Phases B,C switched but phase A did not switch

(b) Reactive power waveforms caused by a three phase 300 KVAR unbalanced capacitor bank switching off. Phases B,C switched but phase A did not switch

Figure 5.9: Unbalanced capacitor switching

may switch. This is an abnormal condition, and reactive power or voltage changes are seen on only one or two-phases instead of all three-phases. The failed phase(s) will not be able to provide the desired voltage or VAR support. This will lead to voltage or reactive power imbalance and will cause the feeder to operate under non optimal conditions. Figure 5.9, plots (a) and (b) show example reactive power waveforms caused by unbalanced 300kVAR bank switching on and off respectively. Step changes can be observed on phase B and C (QA and QC), but phase A (QA) remains relatively unchanged. RMS shape features can be easily used to identify the step changes or absence of step changes in reactive power waveform. The classifier can then use RMS shape features to detect unbalanced capacitor operations. However, there are scenarios where a small step change may be observed even when a phase did not switch. It is not possible to identify

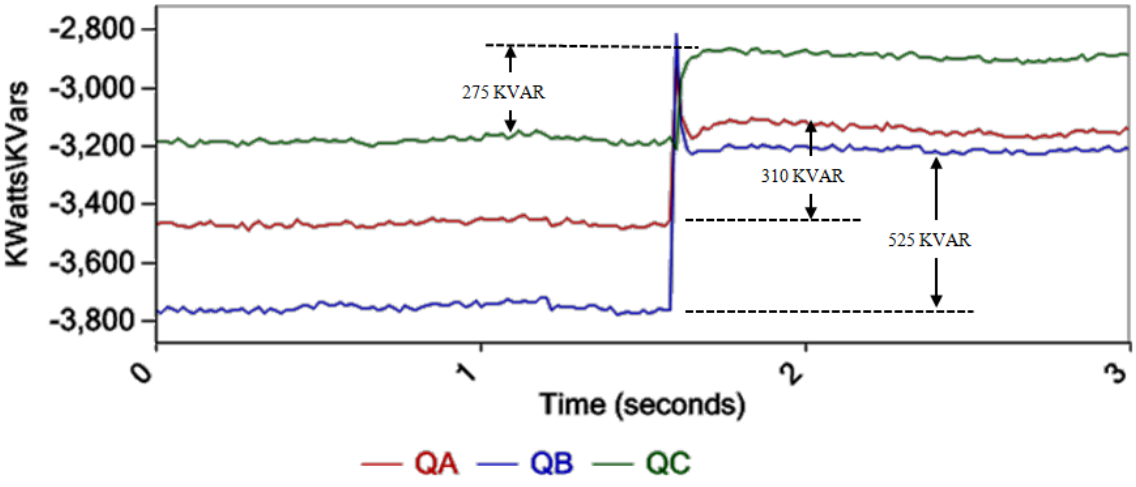


Figure 5.10: Capacitor VAR imbalance

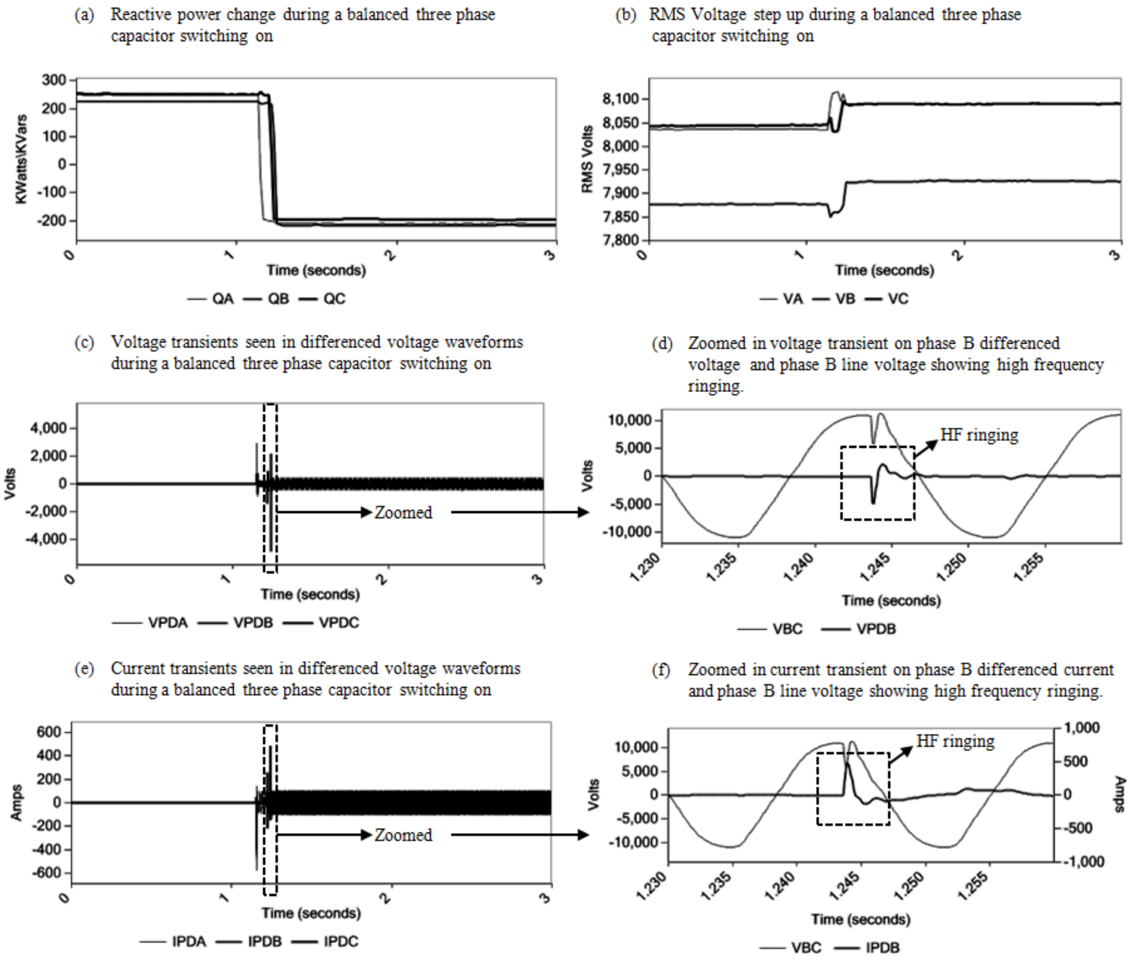
capacitor unbalanced operations based on RMS shape features alone. In order to quantify the amount of unbalance between the three-phases, a percentage unbalance feature $p_{Specific_{pcnt-unb}}$is extracted using a simple calculation:

$$p_{Specific_{pcnt-unb}} = \frac{Max\left(\Delta Q_{ph}\right) - Min\left(\Delta Q_{ph}\right)}{Max\left(\Delta Q_{ph}\right)},\ ph \in \{A, B, C\} \tag{5.1}$$

where $\Delta Q_{ph}$ is the change in reactive power observed on phase $ph$.

#### *5.3.1.2 Capacitor VAR imbalance*

Capacitor VAR imbalance is caused when all phases of a capacitor bank switch, but one or more phases do not exhibit the rated change in reactive power. This may be caused by the failure of on or more capacitor cans within a phase. Under normal operating conditions, capacitors are allowed 10-15% deviation [100] from rated values. However, large deviations in VAR changes between phases may not be acceptable. Figure 5.10 shows example waveforms recorded during a capacitor switching off event,. It can be seen that all the three-phases switched and showed significant change in VARS. However, the VAR change on phases A


(a) Reactive power change during a balanced three phase capacitor switching on

(b) RMS Voltage step up during a balanced three phase capacitor switching on

(c) Voltage transients seen in differenced voltage waveforms during a balanced three phase capacitor switching on

(d) Zoomed in voltage transient on phase B differenced voltage and phase B line voltage showing high frequency ringing.

(e) Current transients seen in differenced voltage waveforms during a balanced three phase capacitor switching on

(f) Zoomed in current transient on phase B differenced current and phase B line voltage showing high frequency ringing.

Figure 5.11: Transients caused by normal capacitor operation

($\Delta Q_A \approx 310kVARS$) and C ($\Delta Q_C \approx 275kVARS$) are much smaller compared to the change in phase B ($\Delta Q_B \approx 525kVARS$). Similar to unbalanced capacitor identification, RMS shape features and percentage unbalance feature $p_{Specific_{pcnt-unb}}$ (Equation 5.1) can be used to identify capacitor VAR imbalance.

### 5.3.2 *Capacitor Problems that Cause Voltage Transients*

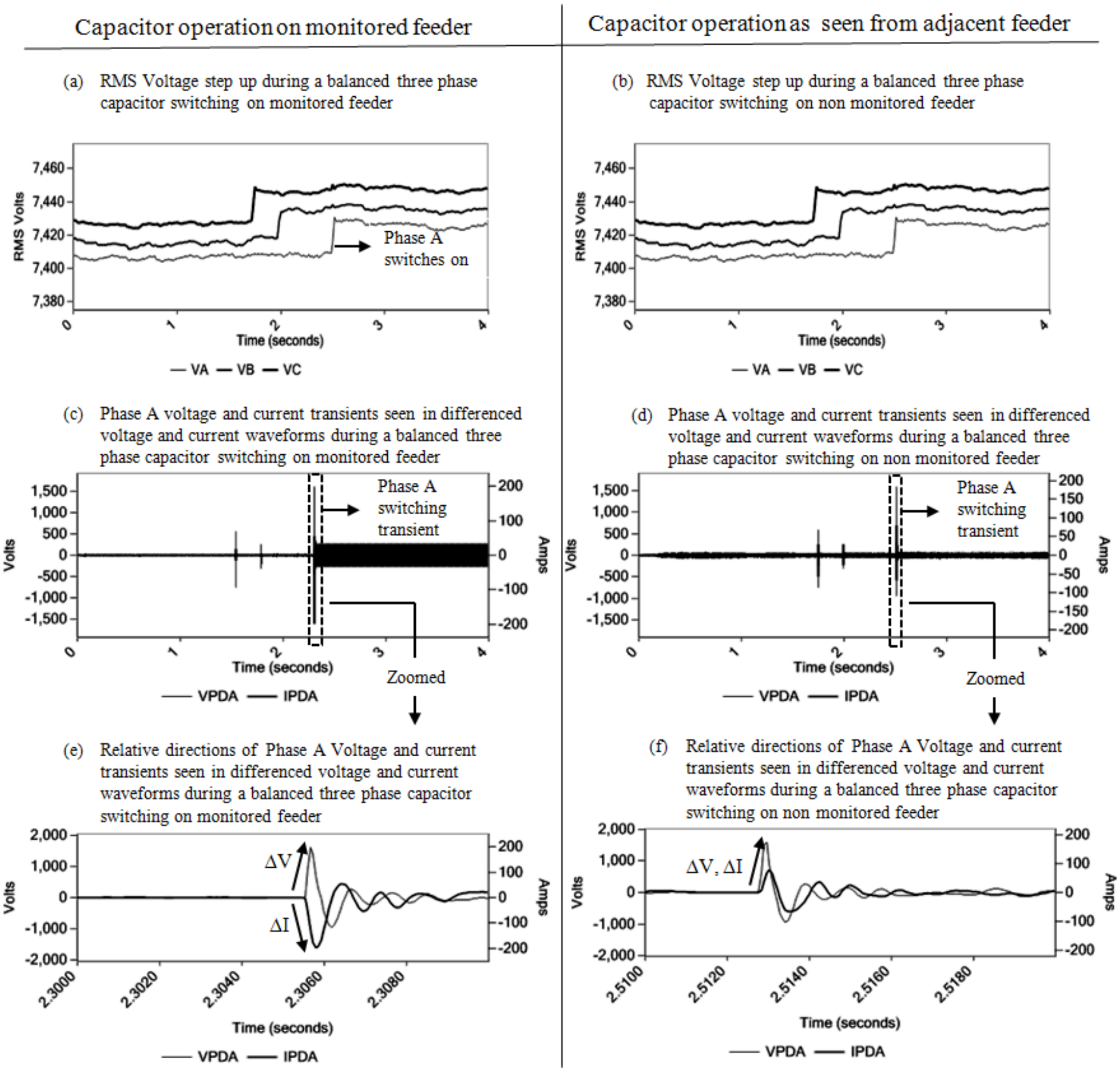
When a capacitor is connected to a distribution feeder or is disconnected from distribution feeder, it causes the capacitor to energize or de-energize. Voltage and current transients are produced during capacitor switching operations, due to the physics involved in charging and discharging a capacitor banks (i.e. voltage on a capacitor cannot change instantaneously when switched). These are high frequency transients since the capacitor tries to charge itself rapidly to bring the capacitor voltage closer to system voltage. Then, the transient quickly decays and reaches system voltage. As a result, capacitor switching related transients look like high frequency 'ringing'. These high frequency transients can be seen on both voltage and current waveforms. Depending on when the capacitor was switched (relative to the voltage peak), and the initial charge in the capacitor bank, voltage transients can have a peak value as high as twice the system voltage [100]. Figure 5.11 shows waveforms from a normal three-phase balanced capacitor switching on. Plots (c) and (e) show high frequency transients on current and voltage waveforms. Plots (d) and (f) show zoomed in phase B voltage and current transients along with phase B voltage for reference. High frequency ringing transients (dotted box) can be seen on both voltage and current waveform.

Both capacitor switching on and switching off can cause voltage transients. Voltage transients seen during capacitor switching off is called a restrike, and is considered abnormal. Capacitor restrike will be described in the next section. Voltage transients seen during capacitor switching on operations is considered normal when such transients are caused by normal switching operations. Normal capacitor switching operations cause at most one transient per phase. However, if the switch contacts are faulty, multiple switching transients may be seen due to switch bounce or due to arcing switch contacts. There are other situations where the switches are healthy, but capacitor cans or connections may be faulty and may cause voltage transients because of arcing inside the capacitor.

Capacitor switching transients are detrimental to power quality since they can cause several power quality issues including, but not limited to over voltage, harmonics and ferroresonance [100]. Capacitor transients can also adversely affect sensitive loads connected to the feeder or other power system components connected to the feeder [102]. Hence, power utility companies try to minimize the number of transients and the size of voltage transients caused by capacitor switching operations. Capacitor switching transients caused by normal or abnormal capacitor operations on the feeder being monitored show unique characteristics. These unique characteristics can be identified in the waveform data recorded at the substation. These characteristics are:

1. Capacitor switching transient frequencies is a function of system frequency and the voltage raise caused by the capacitor. This typically corresponds to a frequency range of 300 to 1000 Hz [100]. Hence it is possible to use a high pass or band pass filter to extract possible high frequency voltage transients from recorded voltage waveforms. Then a simple peak or outlier detection algorithm can be used to detect voltage transients. However, other system events such as inrushes and overcurrents can also cause high frequency transients. RMS waveform shape analysis can be used to eliminate these other sources of high frequency transients and ascertain if the transients were caused' by capacitor switching.

2. Relative behavior of voltage and current transients during capacitor switching can be used to further ascertain the location of the capacitor relative to the monitoring device. Capacitor switching on the feeder being monitored (downstream of the monitoring device) causes the voltage and current transients to initially change in opposite directions. This is because, when a capacitor is switched on, the capacitor acts as a short circuit trying to charge as fast as it can and as a result, causes the system voltage to decrease. How-



(a) RMS Voltage step up during a balanced three phase capacitor switching on monitored feeder

(b) RMS Voltage step up during a balanced three phase capacitor switching on non monitored feeder

(c) Phase A voltage and current transients seen in differenced voltage and current waveforms during a balanced three phase capacitor switching on monitored feeder

(d) Phase A voltage and current transients seen in differenced voltage and current waveforms during a balanced three phase capacitor switching on non monitored feeder

(e) Relative directions of Phase A Voltage and current transients seen in differenced voltage and current waveforms during a balanced three phase capacitor switching on monitored feeder

(f) Relative directions of Phase A Voltage and current transients seen in differenced voltage and current waveforms during a balanced three phase capacitor switching on non monitored feeder

Figure 5.12: Comparison of waveforms from capacitor switching as seen from monitored feeder and adjacent feeder

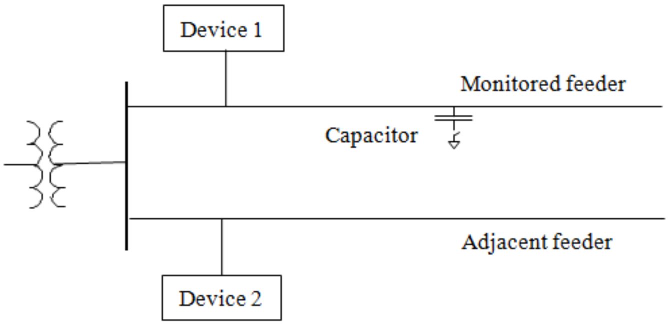


Figure 5.13: Simplified one-line diagram of monitored circuits

ever, when a capacitor operates on a parallel feeder or an adjacent feeder (upstream of the monitoring device), the direction of the current transient will be reversed as the current will flow towards the capacitor on adjacent feeder and away from the feeder being monitored. This will cause the voltage and current transients to have initial directions that are identical. Providing the position of a faulty capacitor bank relative to the motoring device that observed the event will help utility personnel narrow down the location of the problematic device. Figure 5.12 shows the same capacitor switching on event recorded by two monitoring devices (Device 1, Device 2 in Figure 5.13). Plots (a), (c) and (e) are waveforms recorded by Device 1 monitoring the feeder on which the capacitor switched. Plots (b), (d) and (f) are waveforms recorded by Device 2 monitoring an adjacent feeder tied to the same bus. Plots (a), (b) show RMS step change in voltage waveforms. Both these plots look almost identical even though they were recorded by two different devices. This is expected as both the devices (Device 1 and Device 2) are monitoring parallel feeders that are tied to the same bus. The voltage transients (VPDA) seen in plots (c) and (d) are also almost identical for the same reason. Plots (e) and (f) show the current transients (IPDA) seen by

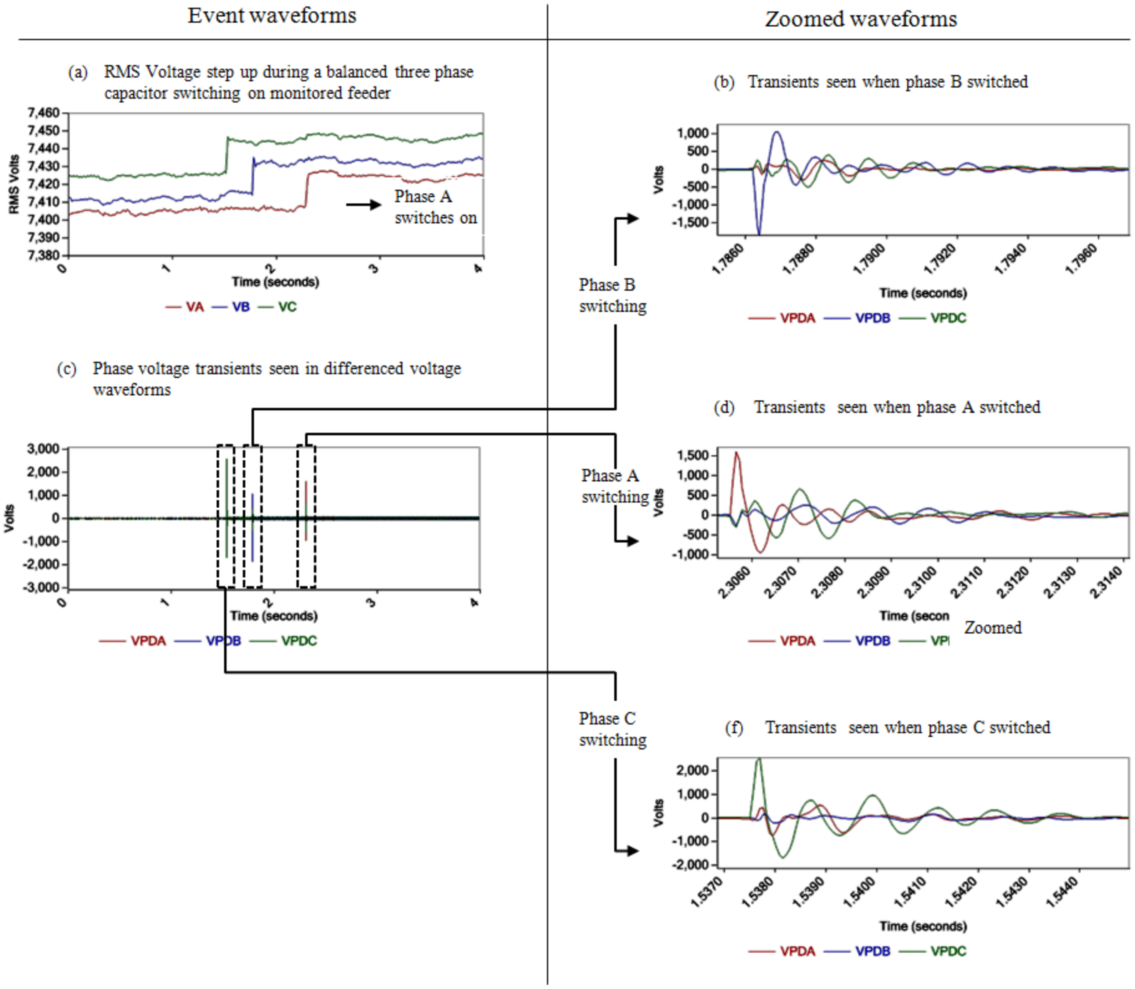


Figure 5.14: Relative magnitudes of voltage transients on three-phases during capacitor switching

Device 1 (monitored feeder) and Device 2 (adjacent feeder). It is clear that the current transient seen on the monitored feeder has a larger magnitude, and more importantly an opposite polarity when compared to the current transient seen on adjacent feeder.

3. Relative magnitudes of voltage transients seen on the three-phases helps to identify the phase that switched. Irrespective of the phase that switched, voltage transients are generally seen on all three-phases. However, the phase

Table 5.3: Event specific features $p_{Specific_{v-trans}}$ for capacitor related voltage transients

| Feature name | Expression | Comment |
|---|---|---|
| Peak voltage transient magnitude | $MaxAbsVTrans_{-ph}$<br>$ph \in \{A, B, C\}$ | Uses high pass filtered voltage waveforms to estimate the peak magnitude of voltage transient. |
| Number of voltage transients | $N_{v-trans_{ph}}$<br>$ph \in \{A, B, C\}$ | Uses high pass filtered current and voltage waveforms to detect voltage transients<br>possibly caused by capacitor operation on monitored feeder. |
| Number of voltage transients non-step | $N_{v-trans-non-step_{ph}}$<br>$ph \in \{A, B, C\}$ | Uses high pass filtered current and voltage waveforms to detect voltage transients possibly caused by capacitor operation on monitored feeder. Excludes voltage transients that were accompanied by a corresponding step change in RMS voltage, current or power waveforms. |
| Time of first voltage transients non-step | $T_{v-trans-non-step_{ph}}$<br>$ph \in \{A, B, C\}$ | The time relative to the beginning of event data when the first transient caused by capacitor operation on monitored feeder was seen. Excludes voltage transients that were accompanied by a corresponding step change in RMS voltage, current or power waveforms. |

that switched tends to have the highest peak magnitude for the voltage transient when compared to the phases that did not switch. This is expected because the voltage transients seen on others phases were caused by mutual coupling among the three-phases. Identifying the phase that switched also helps to identify the problematic phase if the transients were caused by a abnormal capacitor operation. Figure 5.14 shows relative magnitudes of voltage transients on the three-phases during a normal capacitor switching operation. Plot (a) shows RMS voltage waveforms. It is clear from the plot that the capacitor first switched on phase C followed by phase B and finally phase A. Plot (b) shows the voltage transients observed on differenced high-speed voltage waveforms coincident with the switching on phases C, B and A. Plots (b), (d) and (f) show zoomed in versions of the plot corresponding to the switching of phases C, B and A. It can be seen that, when phase B switched (plot (b)), phase B had the highest absolute peak transient magnitude. Same was true when phase A switched (plot (d)), and when phase C switched (plot (f)).

Based on the above characteristics, an algorithm was designed to extract features related to transients caused by capacitor operation (Table 5.3). The pseudo code for the algorithm is shown in Algorithm 3 (Figure 5.15). The voltage transient feature extraction algorithm works processes waveforms in three stages:

1. Initial screening: Analysis of high-speed waveforms for voltage transient detection is an expensive operation. Hence, the algorithm does an initial screening to eliminate waveforms that were caused by other events such as inrushes or overcurrents.

2. Signal pre-processing: High-speed voltage and current waveforms need to be pre-processed before voltage transient related features can be extracted. The algorithm pre-processes current and voltage waveforms corresponding to each phase.

   (a) First, the algorithm filters high-speed voltage (V) and current (I) waveforms to remove frequency components below 300Hz, and obtains high-pass filtered voltage (HPFV) and high-pass filtered current (HPFI) signals. The reason for high pass filtering will be clearer with an example. Figure 5.16 shows voltage and current waveforms from a phase C capacitor switch arcing event. Plots (a) and (c) show high-speed current and voltage waveforms from phase C. Plots (b) and (d) show corresponding waveforms after applying a high-pass filter. Voltage and current transients are clearly visible in plots (b) and (d).

**Algorithm 3** Extracting capacitor voltage transient features

```
Using input features RMS_Shape_I, RMS_Shape_V
Using input features  RMS_Pcnt_change_I,RMS_Pcnt_change_V
Initialize output features N_v_trans, N_v_Trans_non_step
Initialize output features Max_Abs_V_Trans, T_v_Trans_non_step
For each phase ph { // Initial screening
  //Ignore possible inrush and overcurrents
  if(RMS_Shape_I[ph] != ('Inrush') && RMS_Shape_v[ph] != ('Inrush') &&
    RMS_Pcnt_change_I[ph] < OCPcntThrehold &&
     RMS_Pcnt_change_V[ph] < OCVPcntThreshold ){
      return //exit algorithm
  }
}
For each phase ph { // Estimate cycle by cycle high frequency peak magniudes
  // High pass filter V and I to remove frequencies below 300 Hz
  Compute HPFI[ph], HPFV[ph]
  // For each cycle, compute the peak value and index of HPFV[ph]
  Compute HPFV_peak_mag[ph], HPFV_peak_indx[ph]
  // Remove outliers and estimate average peak HPFV value for the whole event
  // this is an estimate of ambient High frequency content in voltage signal
  Compute Mean_peak_mag[ph]
  // Compute differenced peak HPFV by removing ambient high frequency content
  Compute Diff_peak_mag[ph] = HPFV_peak_mag[ph]    Mean_peak_mag[ph]
  // Reset Diff_peak_mag[ph] values less than zero
  Assign (Diff_peak_mag[ph] = 0, Where Diff_peak_mag[ph] < 0 )
  Compute Max_Abs_V_Trans[ph] = Max(Diff_peak_mag)
}
Fore each phase ph { // Transient detection
  For each cycle cyc {
    // Check if the peak value HF voltage is considerably larger  than ambient HF
        voltage and is not too small in comparison to line voltage
    If(Diff_peak_mag[ph,cyc] > VTransAbsThreshold &&
      Diff_peak_mag[ph,cyc]/Mean(VRMS[ph]) > VTransPcntThreshold){
      // check phase ph had the largest transient magnitude for cycle cyc
      if(Diff_peak_mag[ph,cyc]/Max(Diff_peak_mag[:,cyc]) >= 1.0){
        //Find relative direction of current and voltage transients
        Compute HPFI_Direction[ph,cyc], HPFV_Direction[ph,cyc]
        //check for monitored feeder
        if(HPFI_Direction[ph,cyc] = HPFV_Direction[ph,cyc]){
          // Increment transients count
          N_v_trans[ph] = N_v_trans[ph] + 1;
          // Check for step change in signals that could have been caused by
              capacitor switching on or load switching
          If(!(Step(P,cyc)  ||  Step_Down(Q,cyc)||  Step_Up(V,cyc))){
            // Increment transients count that did not coincide with step change
            N_v_Trans_non_step[ph] = N_v_Trans_non_step[ph] + 1;
            // Store the first cycle with transients not caused by step changes
            if(N_v_Trans_non_step[ph] == 1)
              T_v_Trans_non_step[ph] = cyc
          }
        }
      }
    }
  }
}
```

Figure 5.15: Extracting capacitor voltage transient features

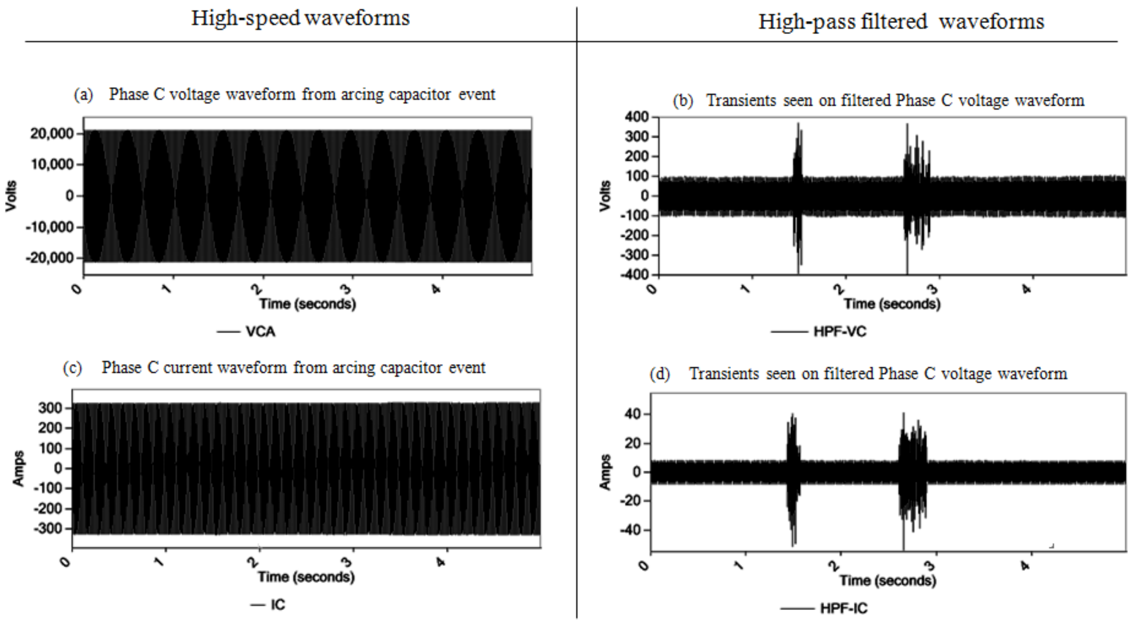


Figure 5.16: High-pass filtered voltage and current signals from sample capacitor arcing event

(b) The algorithm uses only cycle-by-cycle peak values of high-pass filtered voltage, to improve computational efficiency. The algorithm calculates cycle-by-cycle peak magnitudes (HPFV_peak_mag) from HPFV signals. It uses a non-overlapping moving window of one cycle length, then it calculates absolute maximum value of voltage samples within each window. For example, in the Figure 5.17, plots (a), (c) and (e) show cycle-by-cycle peak magnitudes values computed from high pass voltage signals obtained from a capacitor arcing event (same as the event shown in figure 5.16). Voltage transients are clearly visible on phase C (plot (e)) while transients are not seen on phases A and B (plots (c) and (e)).

(c) The algorithm estimates an ambient high frequency level (Mean_peak_mag) to avoid false positives due to noisy voltage signals. It computes

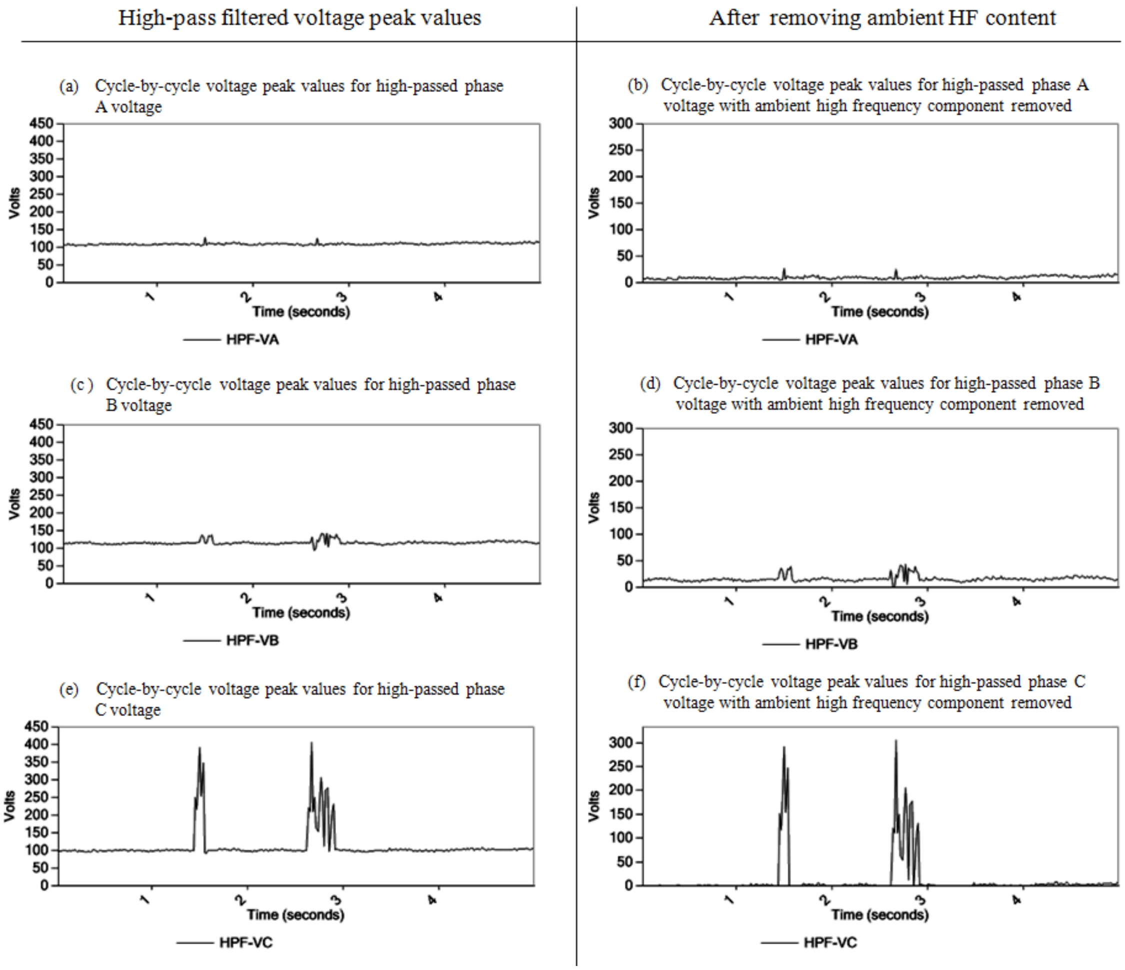


Figure 5.17: Cycle-by-cycle peak values of high-pass filtered voltage waveforms caused by arcing capacitor

the ambient high frequency level by sorting HPFV_peak_mag values, removing outliers, and then computing an average on the rest of values. The algorithm computes a differenced signal (Diff_peak_mag) by removing the ambient high frequency level from HPFV_peak_mag values computed in the previous step (b). In the Figure 5.17, plots (b), (d) and (f) show HPFV_peak_mag values after removing the ambient high frequency level.

3. Transient detection: For each phase, the algorithm chooses candidate cycles for further analysis. The algorithm uses the differenced high frequency peak magnitude values (Diff_peak_mag) computed in the previous stage, to accomplish this.

    (a) First, the algorithm chooses cycles that have peak differenced magnitudes greater than an absolute threshold (VTransAbsThreshold) and greater than relative threshold (VTransPcntThreshold) with respect to the line voltage.

    (b) Then, it compares the peak magnitude for a given cycle and phase to the peak magnitudes on other phases for the same cycle. It does this comparison to ensure that the voltage transient was caused by an operation on the phase being analyzed.

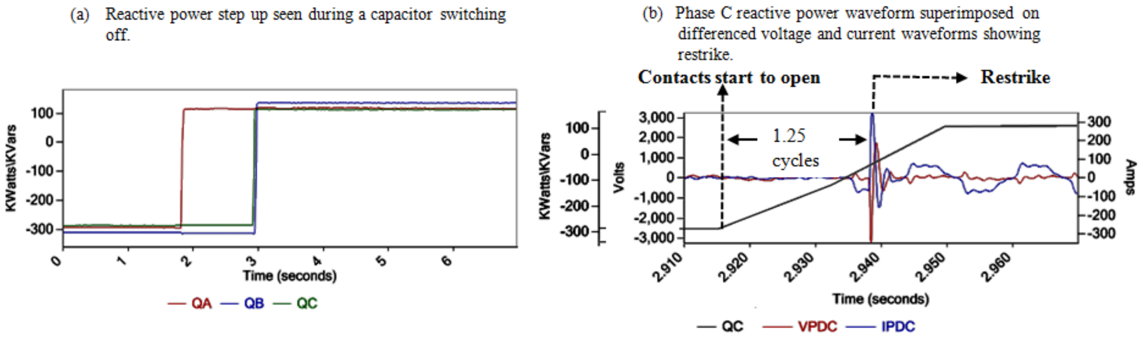
    (c) The algorithm determines the relative directions of the current and voltage transients for each cycle. It does this to confirm if the transient were caused by a normal or abnormal operation on the monitored feeder. If the relative directions are opposite, it increments the transient count feature N_v_trans ($N_{v-trans_{ph}}$).

(d) The algorithm further checks and ignores transients caused by normal capacitor switching operations on either the monitored feeder or on an adjacent feeder. It ignores transients that coincide with step changes in RMS voltage and power waveforms. If it does not detect any such transients, then it increments a non-capacitor-switching-on related transient count N_v_Trans_non_step ($N_{v-trans-non-step_{ph}}$). If the transient happens to be first such transient, it stores the index of the corresponding cycle as a feature T_v_Trans_non_step ( $T_{v-trans-non-step_{ph}}$). This feature can be used to detect capacitor restrike and will be discussed in the next section.

Transient count related features extracted using the above algorithm and RMS waveform shape features together, can be used to detect abnormal capacitor switching operations on monitored feeder.

#### 5.3.2.1 *Capacitor switch restrike*

Restrikes are caused by a dielectric breakdown between switch contacts when a capacitor is being de-energized. Restrikes are more likely to happen after at least half a cycle has passed since the capacitor switch contacts started to open. At this point, the capacitor current is zero, the capacitor would be fully charged, and the instantaneous voltage across the switch contacts may be twice as high as the system voltage [100]. Restrikes not only cause high frequency transients, they may also damage the equipment or switch contacts. This is because, unlike transients caused by normal capacitor switching on; restrikes are preceded by a high energy discharge across capacitor switch contact. It is possible to detect restrike by looking for high frequency voltage transients within a few cycles of a capacitor switch opening. RMS shape features can be used to determine the time


(a) Reactive power step up seen during a capacitor switching off.

(b) Phase C reactive power waveform superimposed on differenced voltage and current waveforms showing restrike.

Figure 5.18: Waveforms from capacitor restrike event

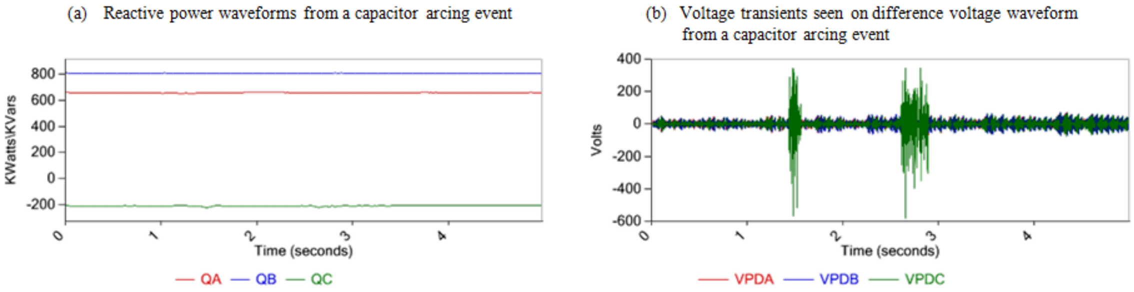
of switch contact opening (based on step change in reactive power). The time of the first high frequency transient ($T_{v-trans-non-step_{ph}}$) could then be used to determine if the transient was seen just after the switches started to open. Using this evidence, an expert system based classifier can label event waveforms as possibly being caused by a capacitor restrike. For example, Figure 5.18 shows waveforms from a capacitor restrike event. Plot (a) shows reactive power step change caused by a 400 KVAR three-phase capacitor switching off. However, analysis of high-speed differenced voltage and current waveforms clearly show a transient caused by restrike, about 1.25 cycles after capacitor switch contacts start to open (plot (b)). If the time at which reactive power (Q) starts to step up and the time at which the voltage transient is detected are known, then it is possible to associate the transient with a restrike.

#### 5.3.2.2 *Capacitor switch bounce*

Switch bounce is a common phenomenon observed with any mechanical switch. Mechanical switches do not make contact cleanly and often make and break contact multiple times before closing. This is due to factors such as inertia, mechanical design and aging. Capacitor switches are no exception; hence, they are also subject to varying degrees of switch bounce. Switch bouncing is generally not a problem if the bounce lasts a fraction of a cycle. However, due to mechanical aging and/or damaged contacts, switch bounce can also last several cycles. In these situations, the switch contacts making and breaking contact cause several voltage transients further damaging the contacts. This not only causes the equipment to degrade but also causes power quality issues. Capacitor switch bounce on monitored feeder is readily recognizable by looking at reactive power waveforms. The repeated step up and down seen in reactive power is a good indicator of capacitor switch bounce (Figure 5.2, plot (a)). Voltage transients can also be used to confirm this hypothesis (Figure 5.2, plot (b)). RMS shape analysis can be used to detect step up and down seen on reactive power waveforms. Then, the transient count feature $N_{v-trans_{ph}}$ can be used as another evidence to confirm capacitor switch bounce.

#### 5.3.2.3 *Arcing inside capacitor can, switch or connection*

Internal failure of a capacitor and faulty switch contacts or connections can cause continuous or sporadic high frequency voltage and current transients [102]. This is because, faulty connections lead to intermittent connectivity of the capacitor bank and will have characteristics similar to the transients caused by switching capacitor banks. It is possible to detect arcing capacitor banks by looking for multiple voltage transients within the waveform data. RMS shape analysis can be used to eliminate transients that were caused by normal capacitor banks switching on either monitored feeder or adjacent feeders. Non-capacitor-switching-on related transient count feature ($N_{v-trans-non-step_{ph}}$) is a good indicator of capacitor arcing. It may be used as an input feature to detect events caused by arcing capacitors. For example, Figure 5.19 shows reactive power waveforms (plot (a)) and differenced voltage waveforms (plot (b)). Plot (a) shows no noticeable step



(a) Reactive power waveforms from a capacitor arcing event

(b) Voltage transients seen on difference voltage waveform from a capacitor arcing event

Figure 5.19: Reactive power and voltage waveforms from a capacitor arcing event

changes in reactive power Q on any of the three-phases. However, voltage transients can clearly be seen in plot (b).

#### 5.3.2.4 *Overactive capacitor*

Voltage transients are caused by normal capacitor switching operations. It is normal for capacitors to switch on and off automatically, few times a day. However, due to a faulty controller or a wrongly configured capacitor switch controller, there have been documented instances [102] where a capacitor bank may switch tens or hundreds of times a day. Such operations will not only quickly wear out switch contacts but also affect other devices on the system. Unlike arcing capacitors, individual events waveforms will not show any abnormal activity as they will look like normal capacitor switching. Classification algorithms developed for this research typically look at few seconds of data only. When a capacitor switches frequently, but successive operations are several seconds or minutes apart, they will not show up in a single waveform file. Hence classification algorithm as described in this research, just by itself will not be able to detect abnormal conditions spanning large time intervals. This is limitation of the classification algorithm described in research. However, the classification algorithm is capable of identifying and labeling each of the switching operations. A higher level clustering algorithm could analyze classification results over time and group the numerous capacitor switching events as belonging to an overactive capacitor.

## 5.4 Features Specific to Overcurrents

It is possible to analyze waveforms acquired during an overcurrent fault, and obtain information such as the protective device that operated, protective device configuration, fault magnitude, fault duration and load loss. This information would help narrow down the possible location of fault to a smaller search area and thus expedite fault location. Fault location in electric power distribution systems has always been a major concern for utility companies. The expansiveness of power distribution systems make them vulnerable to various factors like weather, disturbances caused by animals or human activity, overloading of the system and the aging of or defect in, the components of the power system. These hinder the reliable supply of power to customers. Thus, accurate fault location is necessary to expedite restoration of power to outaged areas. This gains more significance in light of the cost and quality conscious utility business environment that exists now.

### *5.4.1 Overcurrent Faults*

An overcurrent fault is a random disturbance (which in most cases is a short circuit to ground or between phases, often through some amount of impedance in the fault path) that occurs on a power system and results in abnormal system conditions. When overcurrents occur on any part of a power system, the appropriately located protective devices react rapidly to clear the fault and isolate or limit damage to system components and minimize the effect on the remainder of the system. As a result of the operation of a protective device, a portion of the load on the power system may be disconnected. In order to restore supply to the affected parts of the system, the faulted section must be located and then maintenance and repair must be carried out as quickly as possible to keep the duration of the interruption of the service to the customers a minimum.

Circuit breakers configured for overcurrent tripping are used to connect the primary feeders to the bus of the associated substation. When a short circuit fault or an abnormal event occurs on a feeder, the faulted power system component must be quickly isolated from the system. The designated circuit breakers or other protective devices that operate to isolate the faulty component may also interrupt service to some consumers supplied by the feeder. Sectionalizing switches are often installed at the junction of the subfeeders and the main feeders. When trouble on a subfeeder has been located, opening the appropriate switch can isolate the faulty section, and services can be restored to the rest of the feeder before repairs are made. The subfeeders and laterals are sometimes fused to prevent the tripping of the feeder at the substation and thus reduce the extent of outage when fault occurs on one of them. The protective devices are properly coordinated so that the circuit will be open at the proper point to keep the outage area to a minimum. Experience with faults on open wired circuits has shown that the deenergizing these circuits causes the temporary faults to clear themselves in most cases. For this reason, automatic reclosers are installed at a number of points on the circuit to reduce the amount of exposure of the substation equipments to faults. Automatic reclosers trip and reclose a number of times to clear transient faults or to isolate permanent faults

#### *5.4.1.1 Overview of protective device characteristics*

The most widely used protective devices in distribution systems are fuses, reclosers and circuit breakers. The time current characteristics(TCCs) and time settings of each device is based on their rating and manufacturer.

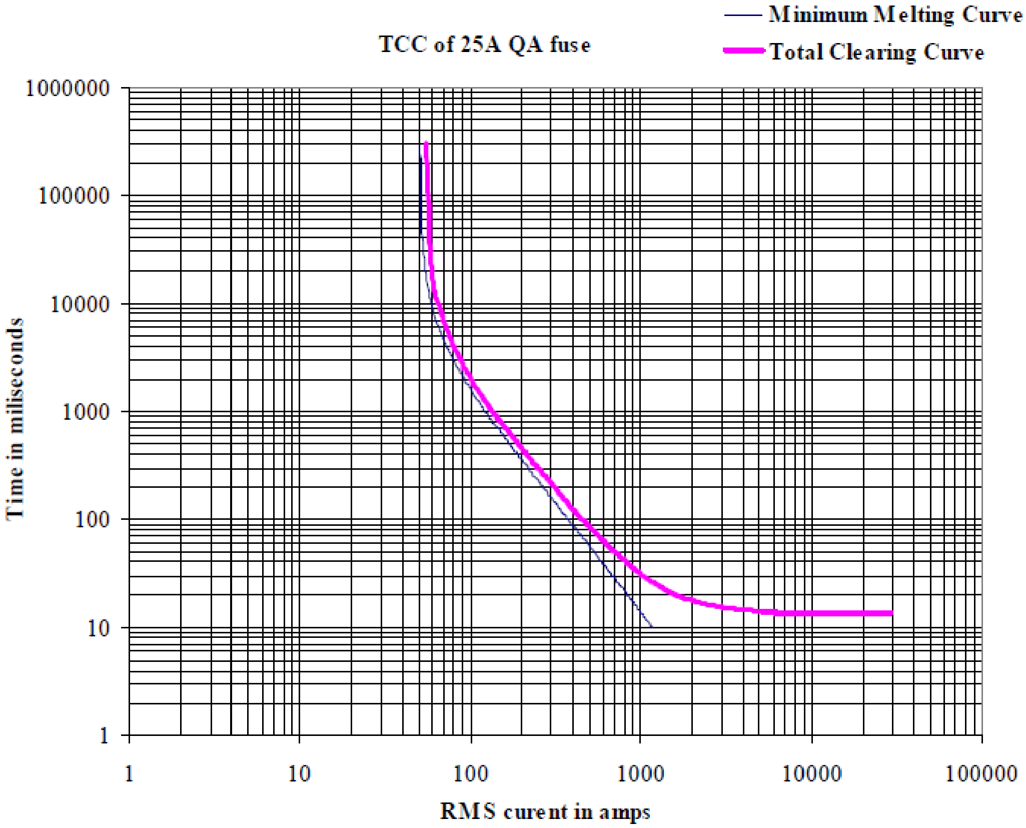


Figure 5.20: TCCs of a 25A QA type fuse

- Fuses: Fuses are overcurrent devices with a circuit-opening fusible member (fuse link) that is destroyed by the passage of overcurrent during a fault or an overload. Fuses are designed to blow within a specified time for a given value of fault current. When a fuse blows, it isolates the section it is protecting until replaced. The time-current characteristics of fuses are represented by two curves, the minimum melting curve and the total clearing curve. The minimum melting curve is a plot of the minimum time vs. current required to melt a fuse link. The total clearing curve is a plot of the maximum time vs. the current required to melt a fuse link and extinguish the arc. Figure 5.20 shows the TCCs of a 25A QA type fuse. The total clearing curve is shown as a bold line, and the minimum melting curve is shown as a thin line. It can be seen that the two curves together define the region of operation for a 25A QA type fuse, i.e., a 25A QA fuse has a high possibility of operating if the sub-event (RMS current, time values) lies within this region.

- Automatic circuit reclosers: An automatic circuit recloser is an overcurrent protective device that automatically trips and recloses a preset number of times to clear temporary faults and isolate permanent faults. Reclosers can be set for various different operations such as, two instantaneous trips and reclose operations followed by two time delay operations before a lockout, one instantaneous trip plus three time-delay operations, or etc. The instantaneous and time-delay characteristics are functions of their ratings and are set by the utility when placed in service. Figure 5.21 shows the TCCs of a 70A '4H' type hydraulic recloser. The time-delay curve is shown in bold. It can be observed that, for any given current level, the recloser operates faster on the instantaneous curve when compared to the time-delay curve.

- Automatic circuit breakers: Circuit breakers are automatic interrupting devices, which are capable of breaking and reclosing a circuit under all conditions, faulted or normal. The operating characteristics of circuit breakers are quite similar to that of reclosers except, unlike the reclosers, circuit breakers are not self-contained units. Also they are controlled by reclosing relays located outside the circuit breaker.

### *5.4.2 Overcurrent Fault Categories*

For the purpose of fault location, overcurrent faults may be categorized based on different criteria such as phase and ground involvement, protective device that operated, whether or not the fault resulted in an outage, fault duration and cause

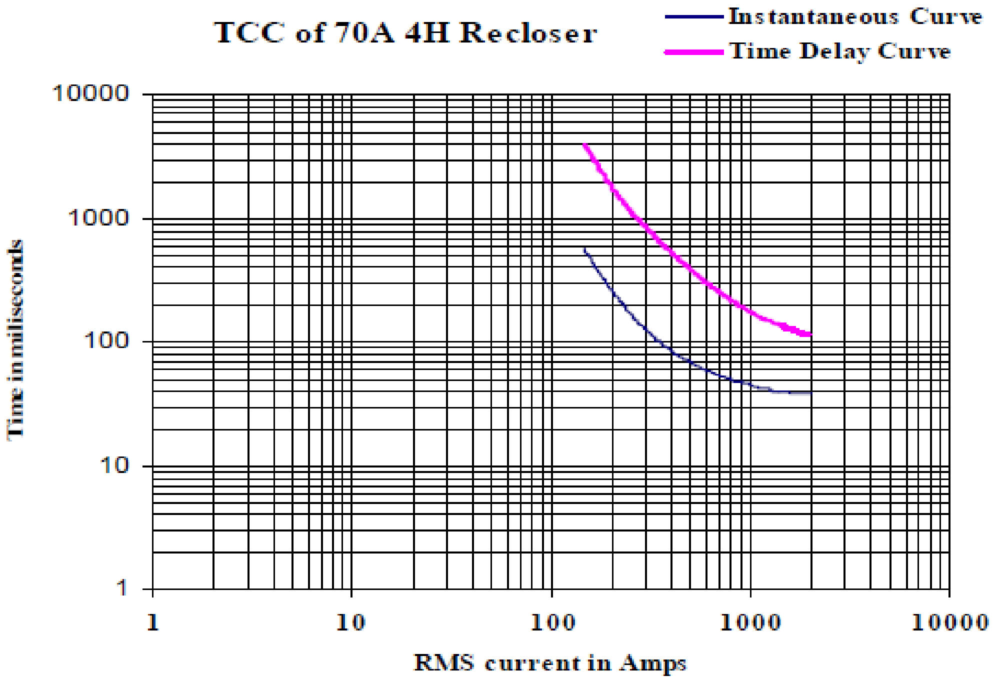
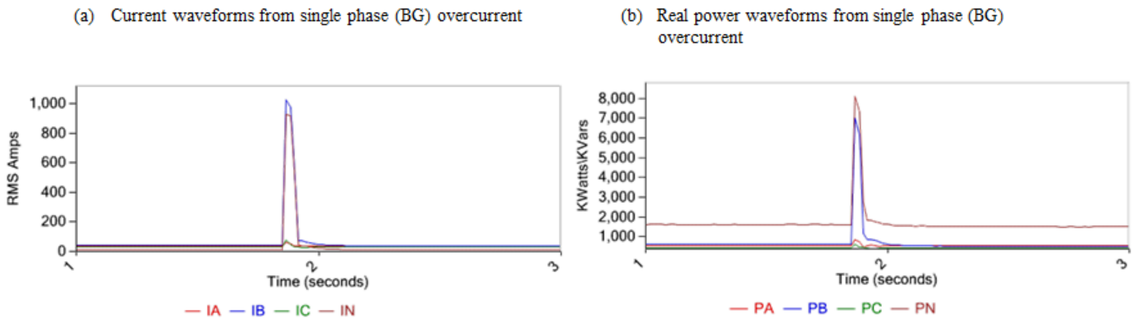


Figure 5.21: TCCs of a 70A ‘4H’ type hydraulic recloser


(a) Current waveforms from single phase (BG) overcurrent

(b) Real power waveforms from single phase (BG) overcurrent

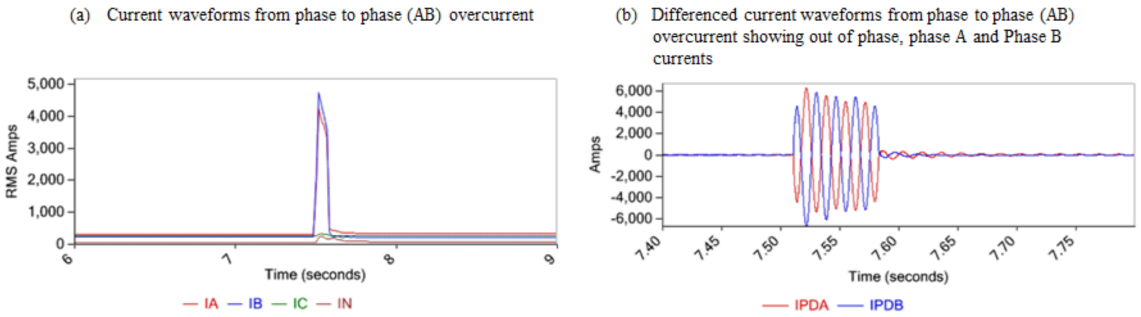
Figure 5.22: Example waveforms caused by a single-phase B to ground overcurrent

of the fault. The objective of this research is not to develop a full fledged automatic fault location algorithm. However, each of these categories can be considered as different output features that aid a human operator in fault location. Algorithms were designed to extract these overcurrent specific features:

1. Phase and ground involvement: One of the most common criteria used to classify overcurrent faults is phase(s) and ground involvement during overcurrent fault. The following are fault categories grouped based on phase and ground involvement:

   (a) Single-phase to ground faults are most common faults observed on a distribution system. These include phase A to ground (AG), phase B to ground (BG) and phase C to ground (CG) faults. Single-phase to ground overcurrents are identified by comparing phase current and neutral current waveforms during an overcurrent. The neutral current will closely track the phase current that was involved in the overcurrent. Figure 5.22 shows RMS current and real power waveforms caused by single a single-phase to ground overcurrent fault.

   (b) phase-to-phase faults that do not involve ground. Phase-to-phase faults


(a) Current waveforms from phase to phase (AB) overcurrent

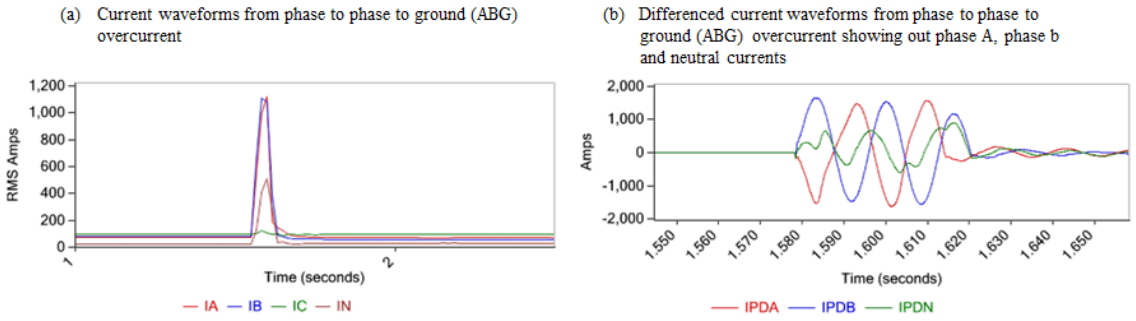
(b) Differenced current waveforms from phase to phase (AB) overcurrent showing out of phase, phase A and Phase B currents

Figure 5.23: Example waveforms caused by phase A to phase B overcurrent fault

include Phase A to B (AB), phase B to C (BC) and phase C to A (CA) faults. Phase-to-phase overcurrents are identified by comparing the phase currents of the phases that were involved in the overcurrent. During a phase-to-phase overcurrent, the involved phases will cause the current waveforms samples to be equal in magnitude but out of phase with each other. Figure 5.23 shows RMS current and differenced current waveforms from a phase B to C overcurrent fault.

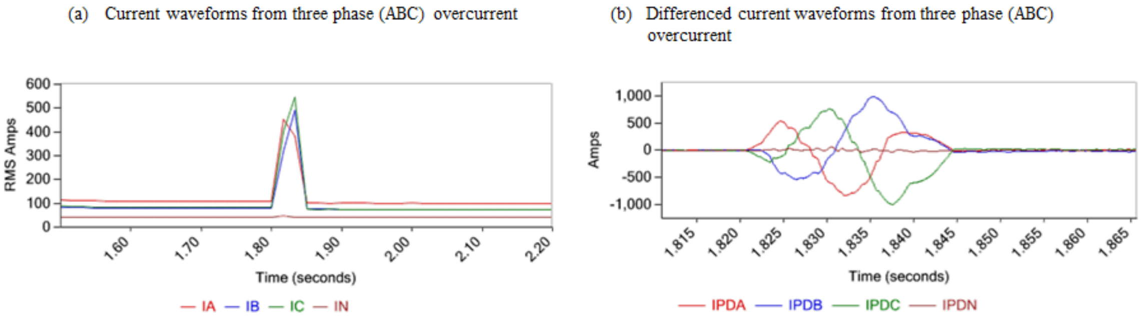
(c) Two-phase faults that involve ground. Two-phase faults that involve ground include Phase A to B to ground (ABG), phase B to C to ground (BCG) and phase C to A to ground (CAG) faults. Unlike phase-to-phase faults, two-phase faults that involve ground do not result in equal fault current magnitudes on the involved phases. Two-phase to ground faults cause relatively high magnitudes of current to flow in two of the phases and the neutral. Figure 5.24 shows RMS current and differenced current waveforms from a phase A to B to ground fault.

(d) Three-phase faults that do not involve ground (ABC fault). These faults are identified by looking for relatively high current magnitudes on all three-phases without noticeable change in neutral current. Figure 5.25 shows RMS current and differenced current waveforms from a three-


(a) Current waveforms from phase to phase to ground (ABG) overcurrent

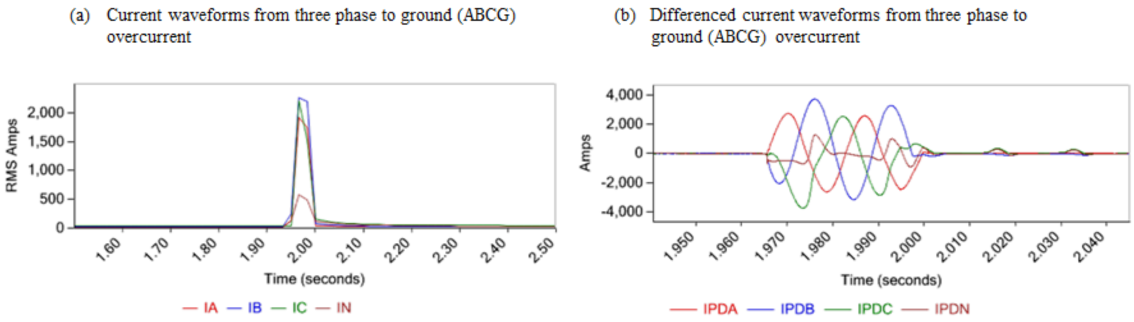
(b) Differenced current waveforms from phase to phase to ground (ABG) overcurrent showing out phase A, phase b and neutral currents

Figure 5.24: Example waveforms caused by a phase A to phase B to ground overcurrent


(a) Current waveforms from three phase (ABC) overcurrent

(b) Differenced current waveforms from three phase (ABC) overcurrent

Figure 5.25: Example waveforms caused by a three-phase overcurrent fault that did not involve ground


(a) Current waveforms from three phase to ground (ABCG) overcurrent

(b) Differenced current waveforms from three phase to ground (ABCG) overcurrent

Figure 5.26: Example waveforms caused by a three-phase to ground overcurrent

phase overcurrent fault that did not involve ground.

(e) Three-phase faults that involve ground (ABCG fault). These faults are identified by looking for relatively high current magnitudes on all three-phases and neutral current. Figure 5.26 shows RMS current and differenced current waveforms from a three-phase overcurrent fault that involved ground.

(f) Evolving faults whose phase and ground involvement changes as the fault progresses. While most overcurrent faults involve the same phases and ground over the duration of the faults, there are some instances where the phase and ground involvement change over the duration of fault. This may be due to environmental factors or due to the changes induced by the overcurrent fault. Evolving faults are mentioned here for completeness. Identifying evolving faults are left as a future objective of this research. Figure 5.27 shows RMS current waveforms from an overcurrent fault that starts as a BC fault then evolves into ABC fault after about 30 cycles.

2. Protective device that operated: RMS current and real power waveforms can be analyzed to determine if an overcurrent were interrupted by a protective

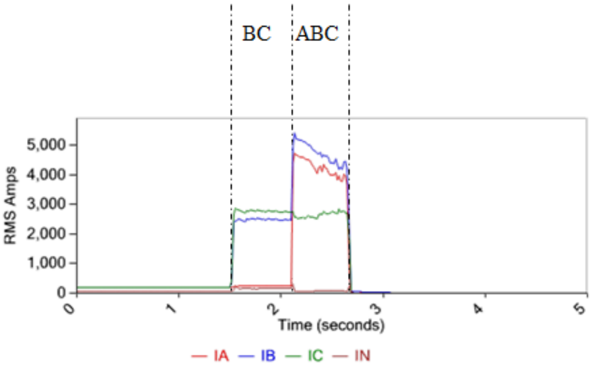


Figure 5.27: Example waveforms from an evolving fault

device. It is possible to determine the type of protective device that operated, and extract information about protective device settings (Figure 5.3, Section 5.1). Fault magnitude information and information about the type of protective device that operated, can be used by utility personal to narrow down the location of the fault. The following are fault categories based on protective device operations:

(a) Faults interrupted by single-phase automatic reclosers: These faults can be recognized either by detecting a single-phase reclose transient after a single-phase to ground fault that caused a loss in a single-phase (single-phase trip). In the case where the fault did not clear and caused an outage, more than one or more trips and recloses may be observed on the same phase. Figure 5.28 shows a phase B to ground overcurrent fault interrupted and cleared by a single-phase automatic recloser. A reclose transient can clearly be seen 2.5 seconds after the fault.

(b) Faults interrupted by three-phase automatic reclosers: These faults can be recognized either by detecting simultaneous reclose transients on three-phases after a fault that caused a loss on all three-phases (three-

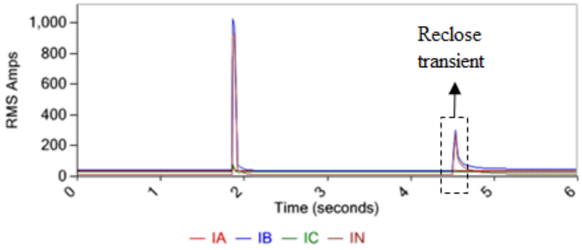


Figure 5.28: Example waveforms from a single-phase overcurrent fault interrupted by a single-phase automatic recloser

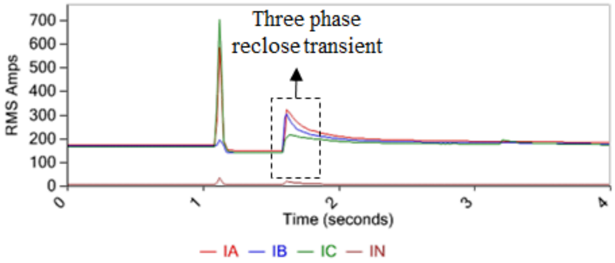


Figure 5.29: Example waveforms from a single-phase overcurrent fault interrupted by a three-phase automatic recloser

phase trip). In the case where the fault did not clear and caused an outage, one or more trips and recloses may be observed on all phases. Figure 5.29 shows a phase C to ground overcurrent fault interrupted and cleared by a three-phase automatic recloser. A three-phase reclose transient can clearly be seen about 0.5 seconds after the fault.

(c) Faults interrupted by automatic circuit breakers: These faults can be recognized by detecting 100% load loss on all phases after a substation circuit breaker tripped. When the substation circuit breaker trips, the whole feeder is taken out of service. Thus, resulting in the feeder current measurements made at the substation go to zero. In reality, due to noise in measurement and DC offset, the current measurement does not truly go to zero. It is possible to remove the DC component from

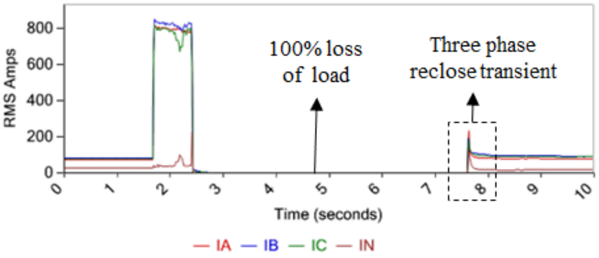


Figure 5.30: Example waveforms from a three-phase overcurrent fault interrupted by a circuit breaker

current measurements and detect if the current measurements are close to the noise floor. Figure 5.30 shows RMS current waveforms from a three-phase overcurrent fault interrupted by a substation automatic circuit breaker. Circuit breaker operation is evident from 100% loss of load (all currents go to zero) just after the three-phase fault.

(d) Faults interrupted by fuses: These faults can be recognized by a single-phase faults that resulted in a load loss on a single-phase. Unlike faults interrupted by automatic reclosers, faults interrupted by fuses are not capable of reclosing. Only a single trip that results in a load loss can be observed. It is not possible to distinguish an overcurrent interrupted by a fuse from one interrupted by a single-phase recloser that tripped once based on load loss. It may be possible to identify fuse operations based on analysis of waveforms. Identification of overcurrents interrupted by a fuse is left as a future objective.

3. Load loss: Real power and RMS current waveforms can be used to determine if a fault resulted in a load loss or outage. the following are fault categories based on load loss:

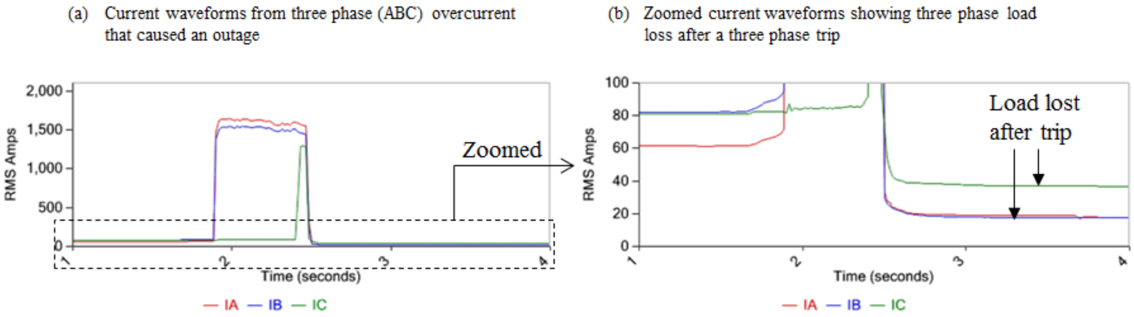
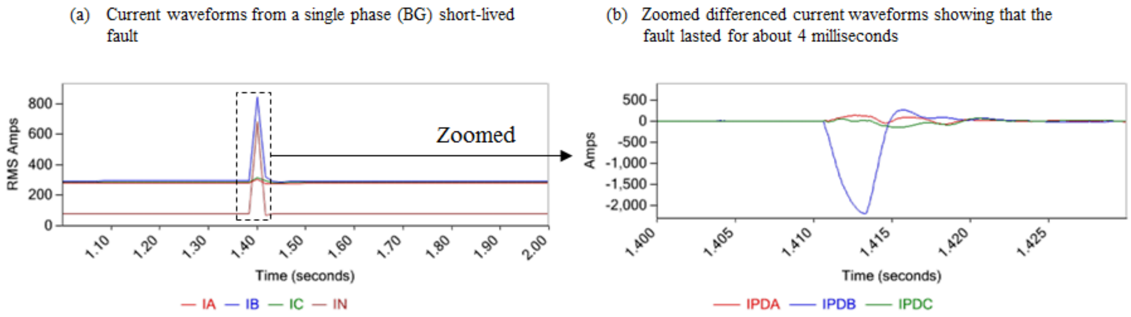


Figure 5.31: Example waveforms from a three-phase overcurrent that was interrupted by a three-phase recloser and resulted in an outage

(a) Faults that resulted in an outage: These are faults that caused a protective device to operate and resulted in sustained service interruption. These faults need immediate attention as power needs to be restored to the affected section of the feeder as soon as possible. Sustained service interruptions affect the reliability indices of utility companies. Hence, they make all efforts to restore power as soon as possible. Figure 5.31 shows RMS current waveforms from a three-phase fault interrupted by a three-phase automatic recloser that tripped and caused an outage.

(b) Faults that resulted in momentary service interruption: These are faults that caused an automatic recloser or a substation circuit breaker to trip and reclose one or more times but did not result in permanent load loss as the temporary fault cleared. Figure 5.29 shows a phase C to ground overcurrent fault interrupted and cleared by a three-phase automatic recloser. A three-phase reclose transient can clearly be seen about 0.5 seconds after the fault, causing a momentary loss of load for about half a second.

(c) Faults that self clear: These are faults that are temporary in nature that did not cause a protective device to operate, as they self cleared.


(a) Current waveforms from a single phase (BG) short-lived fault

(b) Zoomed differenced current waveforms showing that the fault lasted for about 4 milliseconds

Figure 5.32: Example waveforms from a single-phase (BG) short-lived fault

4. Fault duration: High-speed current waveforms can be analyzed, and the duration of overcurrent faults can be estimated accurately. Fault magnitude and fault duration together can be used to match a specific protective device's TCC. This information may help utility engineers to detect problems with protective device coordination. Short lived faults that last less than or equal to half a cycle are of special interest as they may be cable failure precursors. Figure 5.32 shows RMS current and differenced current waveforms from a short-lived fault that lasted only 4 milliseconds (quarter cycle) before self clearing.

Other than the above categories, it is possible to analyze RMS and high speed waveforms to determine the cause of an overcurrent such as capacitor failure, cable failure, lightning arrester failure, vegetation and animal contact. Classification of overcurrent faults based on cause is currently being researched and is left as a future objective.

### 5.4.3 Overcurrent Feature Extraction Algorithm

Figure 5.33 shows the block diagram of overcurrent feature extraction algorithm. The overcurrent feature extraction algorithm extracts overcurrent specific features. These features serve as inputs to fuzzy power system event classifica-

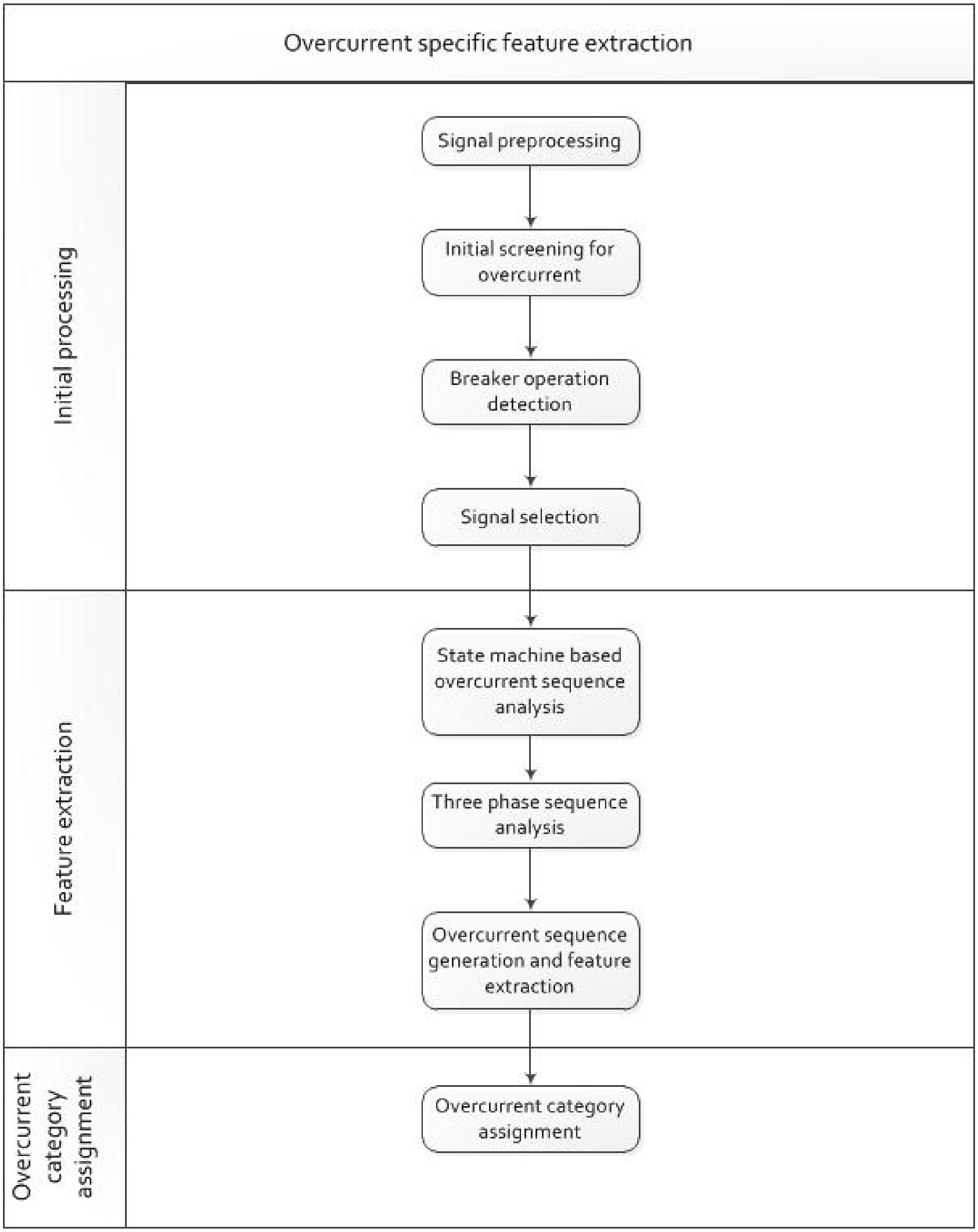

Overcurrent specific feature extraction
Initial processing
Signal preprocessing
Initial screening for overcurrent
Breaker operation detection
Signal selection
Feature extraction
State machine based overcurrent sequence analysis
Three phase sequence analysis
Overcurrent sequence generation and feature extraction
Overcurrent category assignment
Overcurrent category assignment

Figure 5.33: Overcurrent feature extraction algorithm

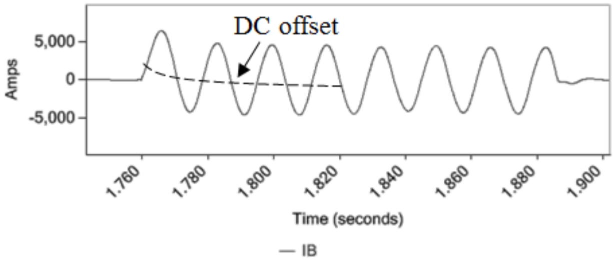


Figure 5.34: High-speed current waveform from an overcurrent event showing DC offset

tion algorithm. Overcurrent feature extraction algorithm also extracts overcurrent specific parameters that serve as output features that aid in fault location. Overcurrent feature extraction can be split into three stages: 1. Initial processing, 2. Feature extraction and 3. Category assignment. Each of these stages will be discussed in following sections.

#### *5.4.3.1 Initial processing*

The initial processing stage prepares event data to aid estimating overcurrent features in an efficient and accurate way. During this stage, the event data is processed in a sequence of four steps:

1. Signal pre-processing: This is the very first step in overcurrent feature extraction algorithm. Current waveforms measured during an overcurrent event often contain a significant decaying DC component during the initial cycles (Figure 5.34). DC component can introduce error in estimating of overcurrent magnitudes. Hence, during the signal pre-processing stage, the DC component is estimated and removed from current waveforms. Another source of error in estimating overcurrent magnitudes is computing RMS current values. RMS quantities are typically calculated starting at the first sample of the event waveforms and using a non overlapping window of length

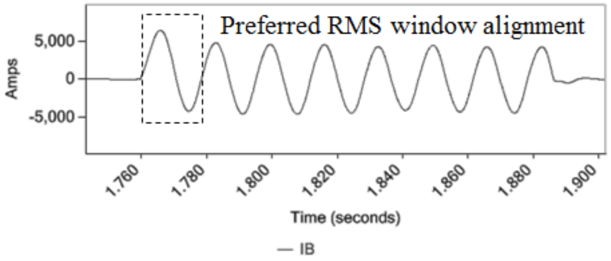


Figure 5.35: Preferred alignment of RMS computation window

equivalent to once cycle of the power system frequency (60Hz). However, the first sample may correspond to any point on the current waveform relative to the current zero crossing. If the RMS computation window is aligned with current zero crossing prior to the inception of overcurrent, RMS overcurrent magnitude estimates will be more accurate (Figure 5.35).

2. Initial screening for overcurrent: In order to improve computational efficiency; only event waveforms that are likely to have been caused by overcurrents are processed further for extracting overcurrent specific features. Event waveforms with current magnitudes that are below a certain threshold are screened and are not processed any further.

3. Breaker operation detection: Overcurrent that were interrupted by substation circuit breaker are flagged and are processed differently. Circuit breaker operations result in a total loss of load. The loss of load makes it easier to detect trips and recloses when analyzing an overcurrent sequence involving breaker. For an event waveform to be flagged as breaker operation, the following condition should be met: the event waveform should contain high current magnitudes and there should be periods in the measurement when the current magnitudes on all phases are near the measurement noise floor.

4. Signal selection: One of the goals of overcurrent feature extraction algorithm is to determine the operating sequence of a protective device that may have interrupted the overcurrent. The overcurrent feature extraction algorithm uses a state machine to track the sequence of operations over time. The operating sequence may be tracked by analyzing real power or RMS currents. Real power or RMS current waveforms can be used to detect load loss or faulted conditions. As a part of analyzing operating sequence, reclose transients are also detected. Reclose transients are generally more visible in reactive power or differenced current waveforms than on real power or current waveforms. The choice of the signals depends on the following conditions:

   (a) If power measurements are not available: Availability of real and reactive power measurements are not guaranteed. A utility company may not instrument voltages and only current measurements may be available. Without voltage measurements, it is not possible to compute real and reactive power. Under this scenario, RMS current waveforms are used for overcurrent sequence analysis. Only differenced current is used for detecting reclose transients.

   (b) If the overcurrent involved a breaker: It is easier to detect if there was a trip or reclose by looking for 100% in the case of breaker operations. RMS current values can easily be compared to a threshold that represents a measurement noise floor. Values below the threshold represent a tripped state, values that are above the noise floor represent a reclosed state. Hence RMS currents are a good choice for tracking breaker operations.

   (c) All other scenarios: For all other scenarios (excluding the previous two),

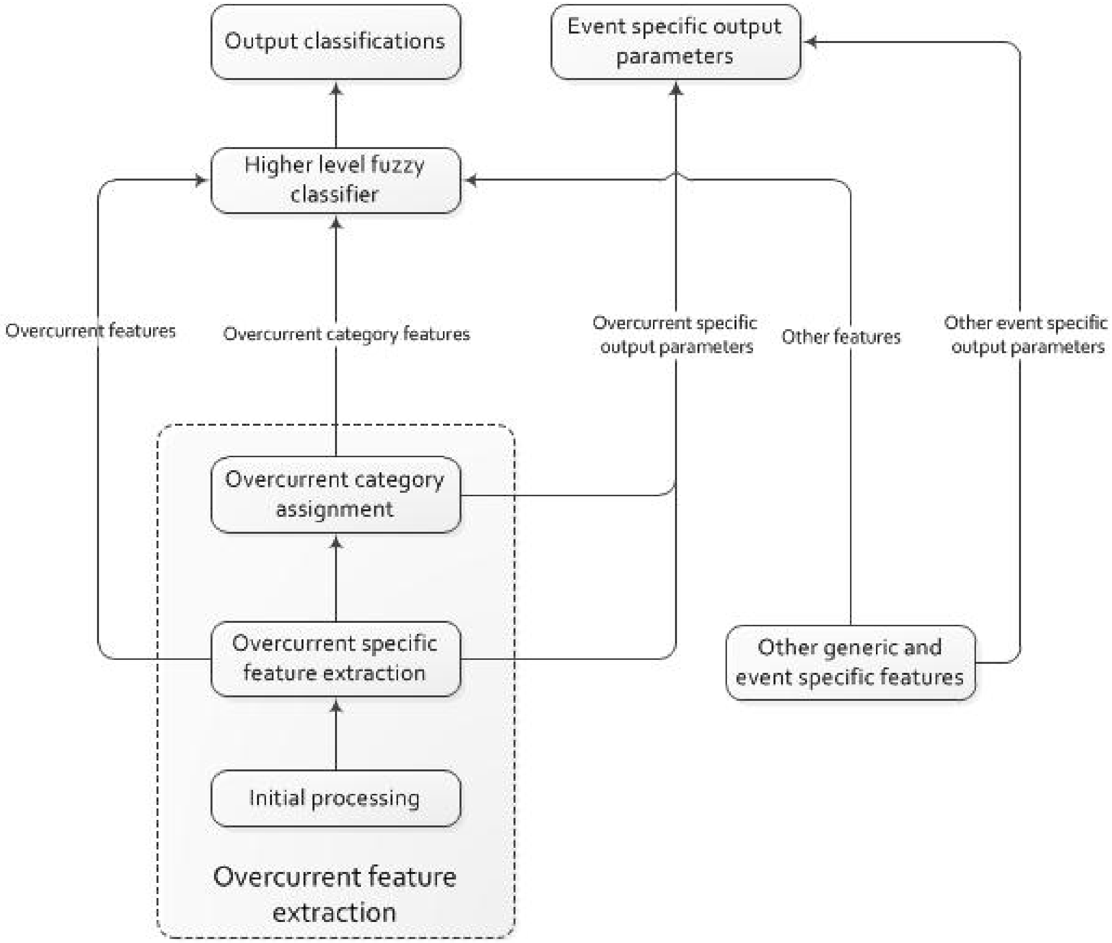

Output classifications
Event specific output parameters
Higher level fuzzy classifier
Overcurrent features
Overcurrent category features
Overcurrent specific output parameters
Other features
Other event specific output parameters
Overcurrent category assignment
Overcurrent specific feature extraction
Other generic and event specific features
Initial processing
Overcurrent feature extraction

Figure 5.36: Relation between overcurrent feature extraction algorithm and higher level fuzzy classifier

real power waveforms are used for analyzing overcurrent sequence. Both reactive power and differenced current are used for detecting reclose transient.

#### *5.4.3.2 Feature extraction*

The overcurrent feature extraction algorithm can be considered as a lower level classifier. This lower level classifier analyzes waveforms and assigns overcurrent categories that are used as both inputs to a higher level fuzzy classifier. Figure 5.36 shows the overcurrent feature extraction algorithm (dotted box) in relation to

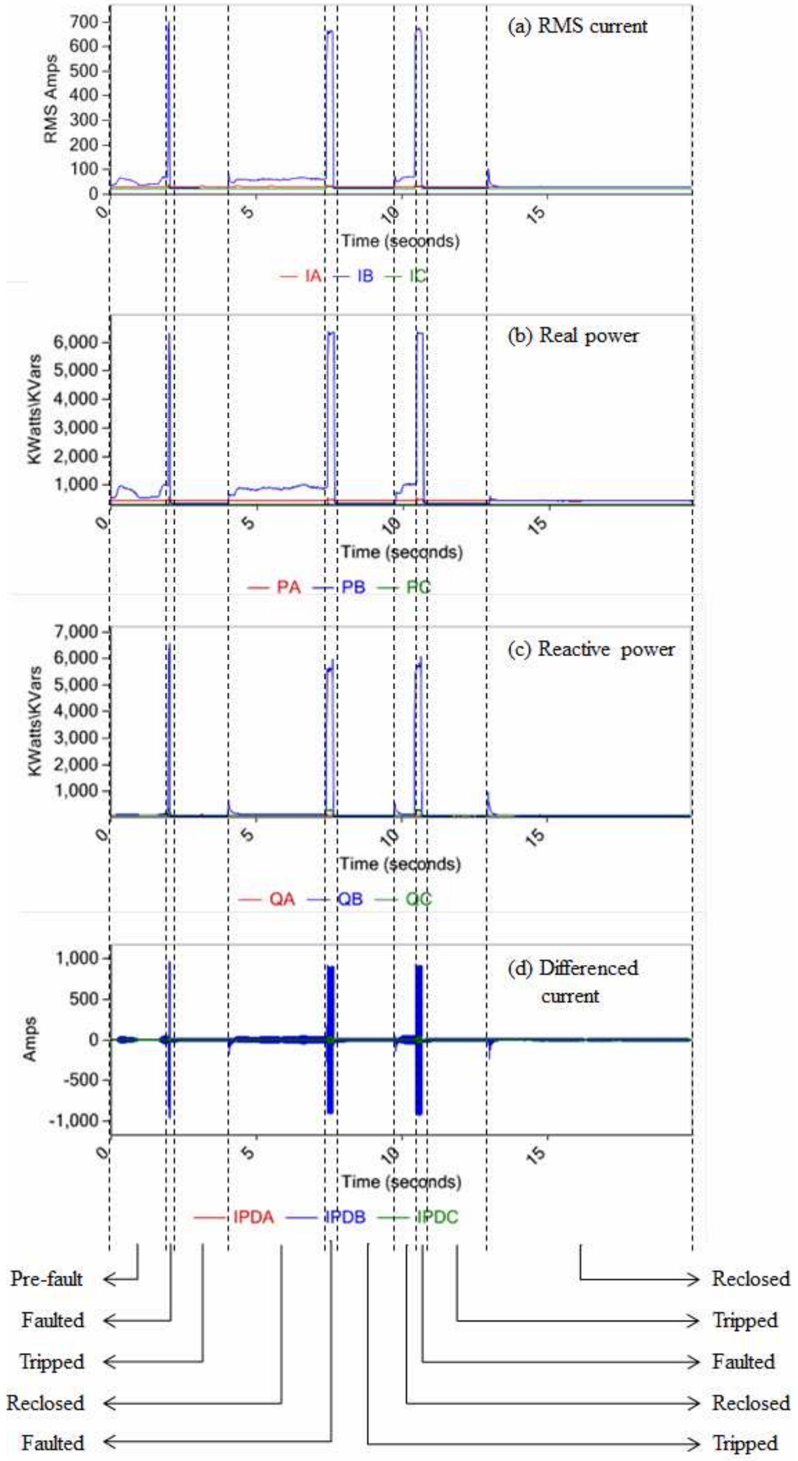


Figure 5.37: Waveforms from an overcurrent fault that caused a single-phase recloser to trip and reclose

the higher level fuzzy classifier. Overcurrent feature extractor, being a lower level classifier; extracts features that are used as input features for overcurrent category assignment, higher level fuzzy classifier and as output parameters to aid in fault location. During this feature extraction stage, data is processed in three steps:

State machine based overcurrent sequence analysis: Overcurrents that are interrupted by line reclosers or substation breakers exhibit complex sequences comprised of one or more faults (high-current), trips and recloses. Figure 5.37 shows one such example. For the purpose of analyzing an overcurrent sequence, each fault, trip and reclose can be considered a state (i.e. 'pre-fault', 'faulted state', 'tripped state' and 'reclosed state'). Once these states are identified in event waveforms, attributes specific to each of these states can be extracted. A state machine (Figure 5.38) was designed to analyze waveforms and store a sequence of states along with the duration (in cycles) of each state. The state machine analyzes waveforms corresponding to each phase. The state machine moves through the RMS current or real power waveforms on a cycle by cycle basis (equivalent sample by sample basis for RMS waveforms). State transitions are triggered by two events, a movement to the next sample (when the sample index is incremented) and when end of samples is reached. The following are the allowable states and state transitions:

- Samples available: This is a generic state that encompasses all other states and state transitions. This state is entered as soon as the sequence analysis algorithm starts. The state machine remains in this state as long as there is at least one more sample that needs to be processed. When end of samples is reached, this state is exited. On existing, the final state for the overcurrent sequence is registered before exiting the sequence analysis algorithm.

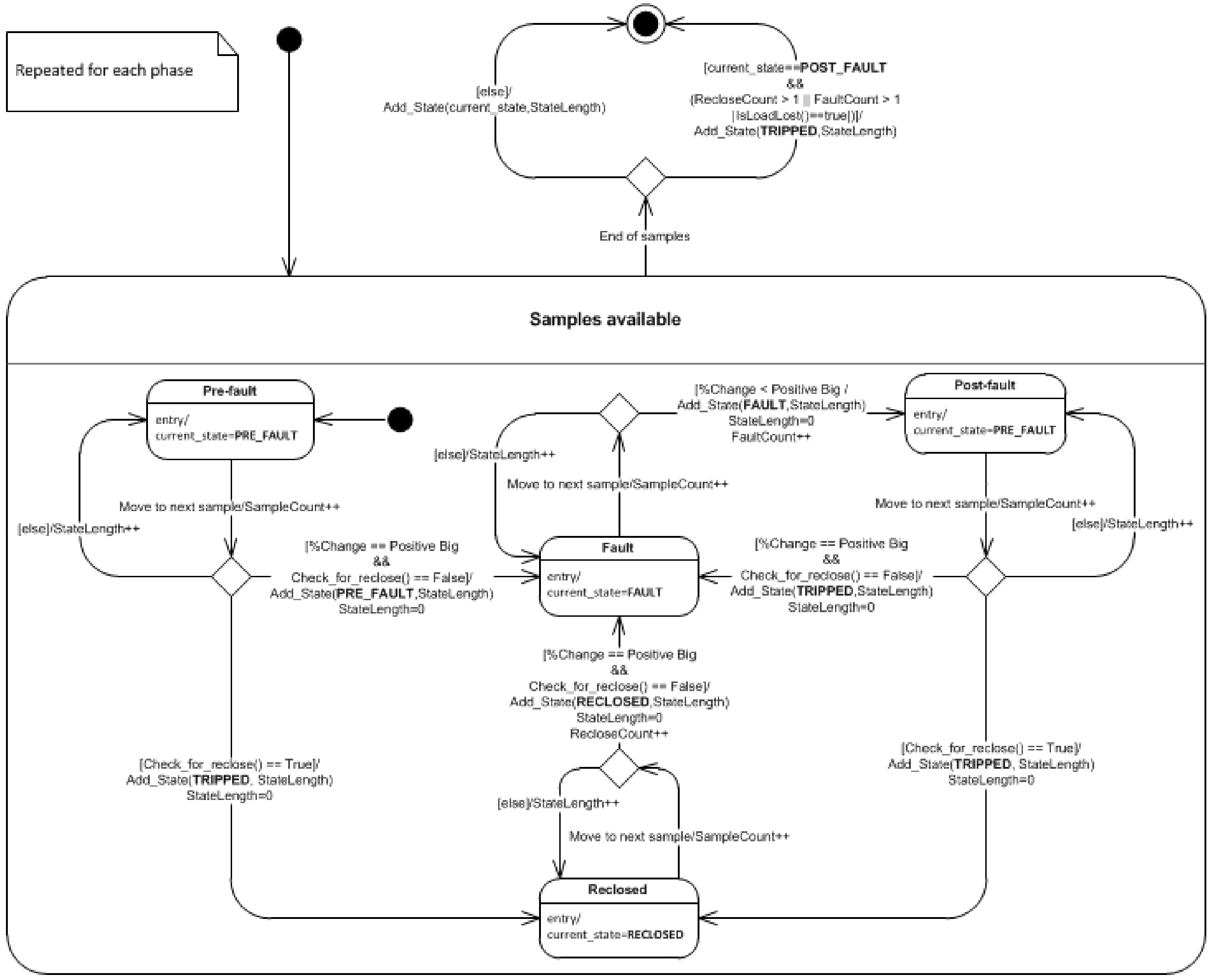

Repeated for each phase
[else]/
Add_State(current_state,StateLength)
[current_state==POST_FAULT
&&
(RecloseCount > 1 || FaultCount > 1
|IsLoadLost()==true)]/
Add_State(TRIPPED,StateLength)
End of samples
Samples available
Pre-fault
entry/
current_state=PRE_FAULT
[else]/StateLength++
Move to next sample/SampleCount++
[%Change == Positive Big
&&
Check_for_reclose() == False]/
Add_State(PRE_FAULT,StateLength)
StateLength=0
[Check_for_reclose() == True]/
Add_State(TRIPPED, StateLength)
StateLength=0
Fault
entry/
current_state=FAULT
[else]/StateLength++
Move to next sample/SampleCount++
[%Change < Positive Big /
Add_State(FAULT,StateLength)
StateLength=0
FaultCount++
[%Change == Positive Big
&&
Check_for_reclose() == False]/
Add_State(RECLOSED,StateLength)
StateLength=0
RecloseCount++
[else]/StateLength++
Move to next sample/SampleCount++
Reclosed
entry/
current_state=RECLOSED
Post-fault
entry/
current_state=PRE_FAULT
Move to next sample/SampleCount++
[else]/StateLength++
[%Change == Positive Big
&&
Check_for_reclose() == False]/
Add_State(TRIPPED,StateLength)
StateLength=0
[Check_for_reclose() == True]/
Add_State(TRIPPED, StateLength)
StateLength=0

Figure 5.38: State machine based overcurrent sequence analysis

- Pre-fault: This is a sub-state of the 'samples available' state. This is the first state the sequence analysis algorithm enters, and it represents the initial state of the overcurrent sequence. The conditions for state transition are as follows:

  - If there was a large positive percentage change in RMS current or Real power waveform, and if no reclose transient was detected, the algorithm transitions to a 'fault' state. Before the state transition, the current state is registered as 'pre-fault' along with the duration for which the algorithm had stayed in the pre-fault state.
  - If a reclose transient was detected, then the algorithm transitions into a 'reclosed' state. The current state is registered as 'tripped' along with the duration for which the algorithm stayed in this state.
  - If neither of the above condition was met, and there is at least one sample left for processing, the algorithm stays in this state, and the duration of the state is increased by one.
  - If the algorithm exits from this state for lack of new samples, the final state is registered as 'pre-fault' along with the duration for which the algorithm stayed in this state. This generally means that the waveforms did not contain even a single overcurrent.

- Fault: This is also a sub-state of 'samples available' state. This state is entered whenever there was a large positive percentage change in RMS current or real power relative to pre-fault values, and if no reclose transient was detected. The conditions for state transition are as follows:

  - If RMS current or real power does not continue to be large relative to pre-fault values, then the algorithm transitions into a 'post-fault' state. Before the state transition, the current sate is registered as 'fault' along with the duration for which the algorithm had stayed in this state.

  - If the above conditions were not met, and there was at least one sample left for processing, the algorithm stays in this state, and the duration of the state is increased by one.
  - If the algorithm exits from this state for lack of new samples, the final state is ignored and not registered as it means the waveform is incomplete.

- Post-fault: This is also a sub-state of 'samples available' state. This state is entered whenever the algorithm exits a 'fault' state. This is a temporary state, and this state is never registered. The algorithm stays in this state till it can be determined if the fault that preceded this state caused a protective device to trip. The conditions for state transition are as follows:

  - If there was a large positive percentage change in RMS current or Real power waveform, and if no reclose transient was detected, the algorithm transitions to a 'fault' state. Before the state transition, the current state is registered as 'tripped' along with the duration for which the algorithm had stayed in the post-fault state.
  - If a reclose transient was detected, then the algorithm transitions into a 'reclosed' state. The current state is registered as 'tripped' along with the duration for which the algorithm stayed in this state.
  - If the above conditions were not met and there was at least one sample left for processing, the algorithm stays in this state and the duration of

the state is increased by one.

  - If the algorithm exits from this state for lack of new samples, the final state is registered as 'tripped' if there was at least one reclosed state or more than one fault state was registered; else the final state is registered as 'pre-fault'. A final state of pre-fault implies that the overcurrent event did not cause a protective device to operate. The duration of the final state is also registered.

- Reclosed: This is also a sub-state of 'samples available' state. This state is entered whenever there a reclose transient detected. The conditions for state transition are as follows:

  - If there was a large positive percentage change in RMS current or Real power waveform and if no reclose transient was detected; the algorithm transitions to a 'fault' state. Before the state transition, the current state is registered as 'reclosed' along with the duration for which the algorithm had stayed in the post-fault state.
  - If the above conditions were not met, and there was at least one sample left for processing, the algorithm stays in this state and the duration of the state is increased by one.
  - If the algorithm exits from this state for lack of new samples, the final state is registered as 'reclosed'. The duration of the final state is also registered.

Table 5.4 shows the result of processing the waveforms from overcurrent event shown in Figure 5.37 using state machine based overcurrent sequence analysis.

Table 5.4: State sequence generated from overcurrent example in Figure 5.37

| | |
|---|---|
| Phase A | {(**PREFAULT**,1200)} |
| Phase B | {(**PREFAULT**,120),(**FAULT**,3),(**TRIPPED**,120),(**RECLOSED**,198),(**FAULT**,14),}<br>(**TRIPPED**,126),(**RECLOSED**,38),(**FAULT**,14),(**TRIPPED**,138),(**RECLOSED**,429) |
| Phase C | {(**PREFAULT**,1200)} |
| Merged | {(**PREFAULT**,120),(**FAULT**,3),(**TRIPPED**,120),(**RECLOSED**,198),(**FAULT**,14),}<br>(**TRIPPED**,126),(**RECLOSED**,38),(**FAULT**,14),(**TRIPPED**,138),(**RECLOSED**,429) |

The sequence corresponding to each phase is represented as comma separated set of (***state***, *duration in cycles*) pair.

Three-phase sequence analysis: Overcurrent sequence analysis algorithm outlined in the previous section analyzes and generates state sequences on a per phase basis. However, sequence information from all three-phases needs to be considered together to provide accurate estimates of overcurrent fault parameters and the nature of interrupting device. The per phase state sequences are merged to form a single state sequence (Table 5.4, last column). The merging of per phase sequences is done based on three conditions:

1. Multi phase faults: When an overcurrent fault involves more than one phase, the duration of fault state needs to be changed so that it spans the faults duration in each of the invoiced phases.

2. Multi phase load loss: When a fault is interrupted by three-phase protective device, the trip will cause load loss on all three-phases. Information from all the three-phases needs to be considered to determine if the interrupting device was a three-phase device. Information from all the three-phases is also used to refine the estimates of duration of tripped state.

3. Multi phase reclose transients: When a fault is interrupted by a three-phase device and the device recloses, the reclose transient may sometimes not be

Table 5.5: Overcurrent state specific features

| State | Extracted features | Sequence string format | Example sequence string |
|---|---|---|---|
| Pre-fault | Mean pre-fault load, duration | None | - |
| Fault | Fault duration, fault magnitude faulted phase(s) | F-(duration cycles,magnitude amps,phase and ground) | F-(3.0c,653A,BG) |
| Tripped | Load loss percentage, duration | T-(load loss A,load loss B,load loss C)%-duration seconds or cycles | T-(0,37,0)%-2.1s |
| Reclosed | Duration | C-duration seconds or cycles | C-3.3s |

prominent on the faulted phase(s). If reclose transients are detected on non faulted phases, they can be used to confirm that the device reclosed. In this scenario, the merged state sequence is altered to include a reclosed state. Presence of reclose transient on a non faulted phase or more than one phase, can be used to confirm that the overcurrent was interrupted by a three-phase automatic recloser.

Sequence generation and feature extraction: The merged state sequence generated during the previous stage is used to extract overcurrent features. Overcurrent specific features are used by the fuzzy classification algorithm, and for cluster analysis to detect recurrent overcurrents. The merged state sequence is also used to generate an overcurrent sequence string. The overcurrent sequence string is a compact representation of the overcurrent event.

Each state in the merged state sequence represents a segment of the event waveforms. From each of these segments, features that are relevant to the corresponding state are extracted. Table 5.5 summarizes the features that are extracted for each state along with a string representation for that state. The sequence string can be used by utility personal for fault location. The sequence string can also be used by utility personal to verify if protective devices coordinated and op-

Table 5.6: Sequence string generated for overcurrent example in Figure 5.37

| Sample overcurrent sequence string |
|---|
| F-(3.0c,675A,BG)-T-(0,37,0)%-2.1s-C-3.3s-<br>F-(13.5c,637A,BG)-T-(0,35,0)%-2.2s-C-38c-<br>F-(13.5c,645A,BG)-T-(0,35,0)%-2.3s-C |

erated correctly. Table 5.6 shows a sequence string generated for the single-phase multi-shot overcurrent fault example in Figure 5.37. The sequence string can be interpreted as follows:

> An initial phase B to ground fault of magnitude 675A was interrupted by a recloser after 3 cycles. The recloser trip caused a 37% loss of load on phase B. When the recloser closed after fault 2.1 seconds, the phase B to ground fault reappeared with a magnitude of 6378A causing the recloser to trip after 13.5 cycles. The second trip caused a 35% loss of load on phase B. The recloser reclosed for a second time after 2.2 seconds. The phase B to ground fault reappeared after 38 cycles and lasted 13.5 cycles before the recloser tripped for a third time and cleared the fault. The final trip caused a 35% loss of load. The recloser closed back for a third time after 2.3 seconds.

#### *5.4.3.3 Overcurrent category assignment*

The final step in overcurrent specific feature extraction is overcurrent category assignment. The category assignment is done based on the based on conditions outlined in Section 5.4.2:

1. Phase and ground involvement: The overcurrent sequence string has phase and ground information for individual faults in an over current sequence. This information is extracted by analyzing current waveforms corresponding to faulted state in the merged state sequence generated after three-phase

**Algorithm 4** Protective device type identification

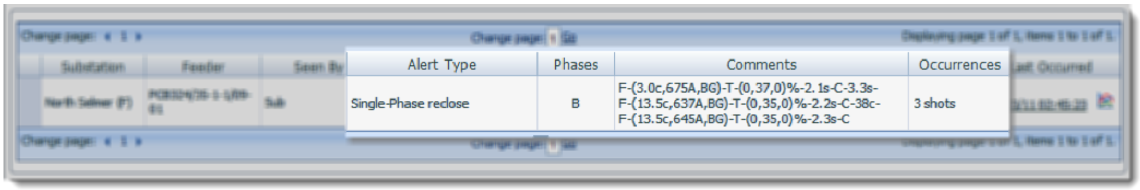
```
// Conditions for assigning protective device category
if(Min(RMS_current) < NOISE_FLOOR))
  Device_Type = BREAKER
else if(Trip_Count > 1 || Reclose_Count > 0){
  //Automatic line recloser
  If(Load_Lost_Phase_Count > 1 || Faulted_Phases > 1)
    Device_Type = THREE_PHASE_RECLOSER
  else
    Device_Type = SINGLE_PHASE_RECLOSER
}
else if(Trip_Count == 1 ){
  if(Faulted_Phases > 1)
    Device_Type = THREE_PHASE_RECLOSER
  else
    Device_Type = SINGLE_PHASE_RECLOSER_OR_FUSE
}
else {
  // Possibly self cleared
  Device_Type = UNKNOWN
}
```

Figure 5.39: Protective device type identification

sequence analysis. A phase and ground involvement category is assigned to the entire overcurrent and is the same as the phase and ground involvement for the first fault within a sequence.

2. Protective device that operated: It is possible to infer the nature of the protective device that interrupted the overcurrent based on estimated load loss, phases that were involved in the fault, reclose transients and number of trips. Algorithm 4 (Figure 5.39) shows the conditions used to assign protective device category.

3. Load loss: If the final state of the merged state sequence is ‘TRIPPED’ then the overcurrent is assigned a tripped category, indicating a possible outage. If the final state of the merged state sequence is ‘RECLOSED’, then the overcurrent is assigned a reclosed category. For all other final states, a self-cleared category is assigned.



| Substation | Feeder | Seen By | Alert Type | Phases | Comments | Occurrences | Last Occurred |
|---|---|---|---|---|---|---|---|
| [illegible] | [illegible] | [illegible] | Single-Phase reclose | B | F-(3.0c,675A,BG)-T-(0,37,0)%-2. 1s-C-3.3s-<br>F-(13.5c,637A,BG)-T-(0,35,0)%-2.2s-C-38c-<br>F-(13.5c,645A,BG)-T-(0,35,0)%-2.3s-C | 3 shots | [illegible] |

Figure 5.40: Screen capture of overcurrent specific output information presented to utility user

4. Fault duration: If the maximum duration of all the individual faults within an overcurrent sequence is less than or equal to half cycle (˜0.8 ms), then a short-lived fault category is assigned to the overcurrent event.

Figure 5.40 shows a screen capture of the overcurrent specific information presented to a utility user. The screen capture shows the information generated by the algorithm after processing the overcurrent example in Figure 5.37. Location and time information has been blurred for confidentiality purposes.

Based on information in Figure 5.40, utility personnel can narrow down the location of fault to sections of the feeder that is protected by a single-phase recloser on phase B. Knowing that the recloser servers about 35% of the load will help to narrow down the the location further. Finally, the fault magnitude estimate of 675A when used in conjunction with available short circuit model of the feeder can help reduce the search area further.

## 5.5 Chapter Summary

In this chapter, the need for event specific feature extraction was explained with examples. Features specific to line-to-ground arcing ad line-to-line arcing were identified, and algorithms to extract these features were described. Then, features specific to abnormal capacitor operations were identified, and the algorithms needed to extract these features were described. Finally, overcurrent specific feature extraction methodology was described in detail.

# 6. FUZZY CLASSIFIER

## 6.1 Introduction

Classification of power system events is a challenging task even for experts. This is due to the diverse nature of power distribution systems, and due to uncertainties introduced by these diverse conditions. It was shown in Chapter III that a fuzzy expert system based classifier is well suited for classifying power distribution system event data. The power distribution event classification problem has a number of requirements that influence the architecture of the fuzzy classifier. The major factors that influenced design of the fuzzy classifier are:

***Hierarchical classifier:*** It was previously shown (Chapter III, Section 5) that the power distribution system event classification problem is a large scale classification problem due to the large dimensionality of input feature space and output space. A hierarchical classifier was proposed to split the large input feature space into more manageable sub-spaces. A hierarchical classifier design also helps to incorporate structure into the classifier that is better suited for handling three phase features and multi-segment events [103].

***Modularity:*** Over time, new types of failure signatures and incipient failure signatures may be discovered and characterized. These new failure signatures need to be added to the classification system on a regular basis. These changes should not require changing the classifier architecture or significantly affect classification performance of existing categories. A classifier architecture that allows the easy addition of new event signature recognition modules without adversely affecting classification performance of existing categories is desirable.

***Inference engine:*** One of the major challenges of designing a hierarchical fuzzy expert system classifier for large scale classification problem is the dearth of fuzzy inference engines that can handle large scale fuzzy inference [104]. Hierarchical structure of the proposed fuzzy expert system requires rule chaining. In a hierarchical expert system, higher level rules that dependent on a particular linguistic variable should not fire until all the lower level rules with the same linguistic variable on the consequent side have been evaluated. Most of the existing fuzzy expert system shells with the exception of FuzzyShell [105, 106] do not handle this condition. A choice had to be made between designing an inference engine tailored to the power system event classification problem or using an existing inference engine.

***Efficiency:*** The fuzzy classifier is part of the classification algorithm that is required to run on remote field hardware with limited memory and processing power. The classification algorithm as a whole needs to be able efficient in terms of CPU usage and memory usage. The large scale nature of the classification problem requires a large and a complex fuzzy rule base. The fuzzy inference engine needs to perform fuzzy inference and assign class labels in an efficient manner.

***Portability:*** For the power system event classification system to be practical, and to allow wide spread deployment, all components of the classification system should be easily portable. This requires that the fuzzy classifier implementation to be platform independent to allow deployment on different hardware and operating system platforms.

Another important aspect of power distribution system event classification is presenting classification information to utility personnel in an intelligent manner. For the classification system to be usable, it cannot report every event that was classified. This is because, a majority of power system events are normal system events that are of little interest to utility personnel. If all these events are reported, utility personnel will be overloaded with data. Worse still, they may also miss the relatively few but more important abnormal events that get buried among normal system activity. It is desirable that only events of interest are presented to utility personnel. Hence, an intelligent reporting framework was developed to process the raw classification information generated by the fuzzy classifier. The reporting framework is responsible for presenting abnormal system activity as action items to utility personnel. These action items contain information about the nature of the event. They also contain event specific parameters that may help to locate the cause of the abnormal event.

The rest of this chapter describes the structure of fuzzy hierarchical classifier, followed by the design of a custom fuzzy inference engine, and finally provides a brief description of the intelligent reporting framework.

## 6.2 Hierarchical Classifier Architecture

Generic features, RMS shape based features (Chapter 3) and specific features (Chapter 4) extracted from event data, are used as inputs to the fuzzy classifier. Features are the 'evidence' based on which the classifier assigns a category to the capture being processed. Fuzzy classifier is the final phase of FLCA algorithm (Chapter 3, Section 6). Figure 6.1 shows detailed schematic of fuzzy hierarchical classifier.

### *6.2.1 Modular Structure*

A modular and hierarchical classifier was designed to handle the large dimensionality of input feature space. Modular nature takes advantage of the fact that only a sub-set of features are needed to classify an event into a specific category. Each classifier module (dotted box, Figure 6.1) is responsible for assigning possibility values for a subset of event categories. Event categories that share a lot

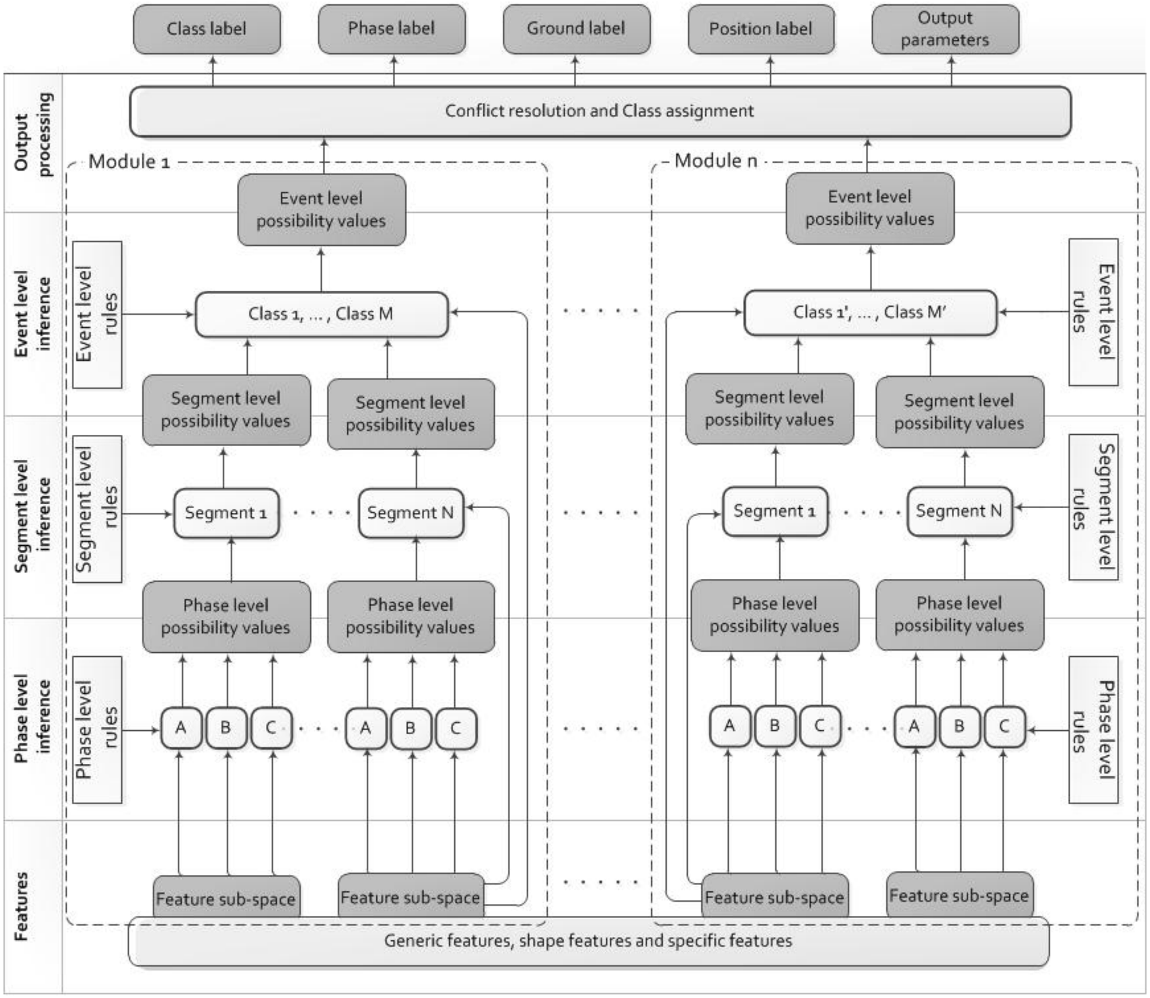


Figure 6.1: Fuzzy hierarchical classifier

Table 6.1: Fuzzy classifier modules

| **Module name** | **Event cause categories** | **Other attributes** |
|---|---|---|
| Capacitor reactive power related | Capacitor bank switching on<br>Capacitor bank switching off<br>Unbalanced capacitor switching on<br>Unbalanced capacitor switching off<br>Capacitor On (VAR imbalance)<br>Capacitor Off (VAR imbalance)<br>. | Phase: Single phase or three phase.<br>Ground: Grounded or ungrounded.<br>Position: Monitored feeder or non-monitored feeder<br>. |
| Capacitor transient related | Capacitor bank restrike<br>Capacitor switch bounce<br>Arcing inside capacitor bank<br>. | Position: Monitored feeder or non-monitored feeder<br>. |
| Arcing related | Arcing (generic)<br>Arcing (generic; short burst)<br>Arcing (generic; long burst)<br>Probable failure of switch or clamp<br>. | Phase: Single phase, two phase or three phase.<br>Position: Monitored feeder only<br>. |
| Overcurrent related | Overcurrent fault (normal)<br>O/C ¡ 1 cycle<br>Capacitor-failure overcurrent<br>Inrush<br>. | Phase: Single phase, two phase or three phase.<br>Ground: Yes or no.<br>Position: Monitored feeder, non-monitored feeder or transmission system<br>. |
| Step change related | Load step up<br>Load step down<br>Voltage step up (normal)<br>Voltage step down (normal)<br>CT/PT switches opened<br>CT/PT switches closed<br>. | Position: Monitored feeder<br>. |
| Motor related | Motor start | Phase: Single phase, two phase or three phase.<br>Ground: Yes or no.<br>Position: Monitored feeder or non-monitored feeder<br>. |

of features, are grouped into the same module. For example, a capacitor module is responsible for assigning possibility values for normal capacitor switching and unbalanced capacitor operation categories. It was shown in Chapter 5, Section 3 that normal capacitor operations and reactive power imbalance can be identified based on features extracted from real and reactive power. For this reason, both normal and unbalanced capacitor operations were grouped into the same module. Similarly, modules were designed for classifying overcurrent related events, arcing related events, motor starts and load switching related events, etc. Table 6.1 shows the complete list of modules and the event categories assigned to each module. Each of these modules uses only a subset of features. Hence, each module uses a lower dimensional sub-space compared to the larger dimensional input feature space. This design does allow some overlap in the feature sub-space used by modules. This because some features may be shared by more than one module. The overlap of input feature sub-space opens up the possibility of overlapping event sub-space. A conflict resolution strategy is used to resolve scenarios where an event data generates high possibility values for more than one event category.

Each module uses a subset of input features and processes features and possibility values at three levels; phase level, segment level and event level. At each of these levels, fuzzy inference is performed by lower level classifiers (solid white blocks) using features and possibility values. These lower level classifiers are called Basic Fuzzy Processing Modules (BFPM).

#### *6.2.1.1 Basic Fuzzy Processing Module (BFPM), the building block*

The hierarchical classifier was designed using BFPMs as building blocks at different levels within each classification module. Figure 6.2 shows the block diagram of BFPM. BFPM uses expert's knowledge represented in the forms of fuzzy

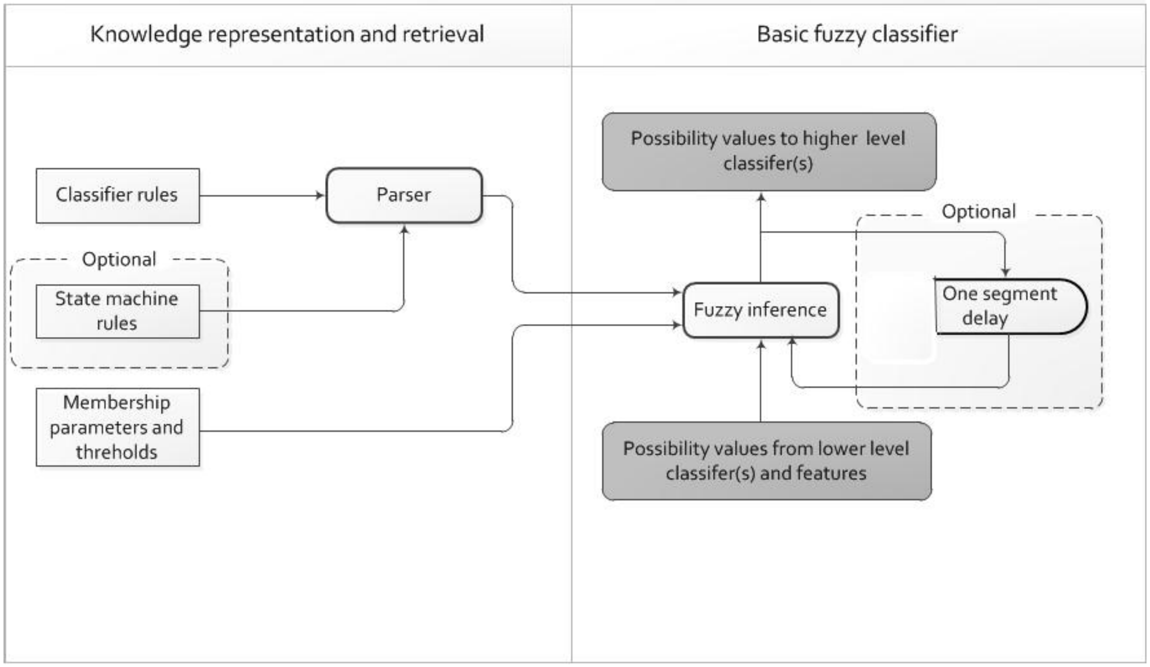

Knowledge representation and retrieval
Basic fuzzy classifier
Classifier rules
Parser
Optional
State machine rules
Membership parameters and threholds
Possibility values to higher level classifer(s)
Optional
One segment delay
Fuzzy inference
Possibility values from lower level classifer(s) and features

Figure 6.2: Basic Fuzzy Processing Module

rules, fuzzy membership functions to process its inputs. The inputs are either features or possibility values from other BFPMs. The outputs of the BFPMs are possibility values. BFPM can receive as inputs, possibility values from other BFPMs at the same level in the hierarchy or a lower level in the hierarchy. When BFPM receives possibility values from a lower level BFPM, it is equivalent to accumulating or gathering the evidence collected from multiple sources (signals) in support or against an argument. When the BFPM receives as inputs possibility values from BFPMs from the same level in the hierarchy, it is equivalent to reusing the evidence already collected by other BFPMs. This is helpful from an efficiency point of view and allows the sharing of information among BFPMs.

BFPMs can also receive possibility values from itself with a delay of one processing step (iteration). This would allow the BFPMs to act as a simple Fuzzy

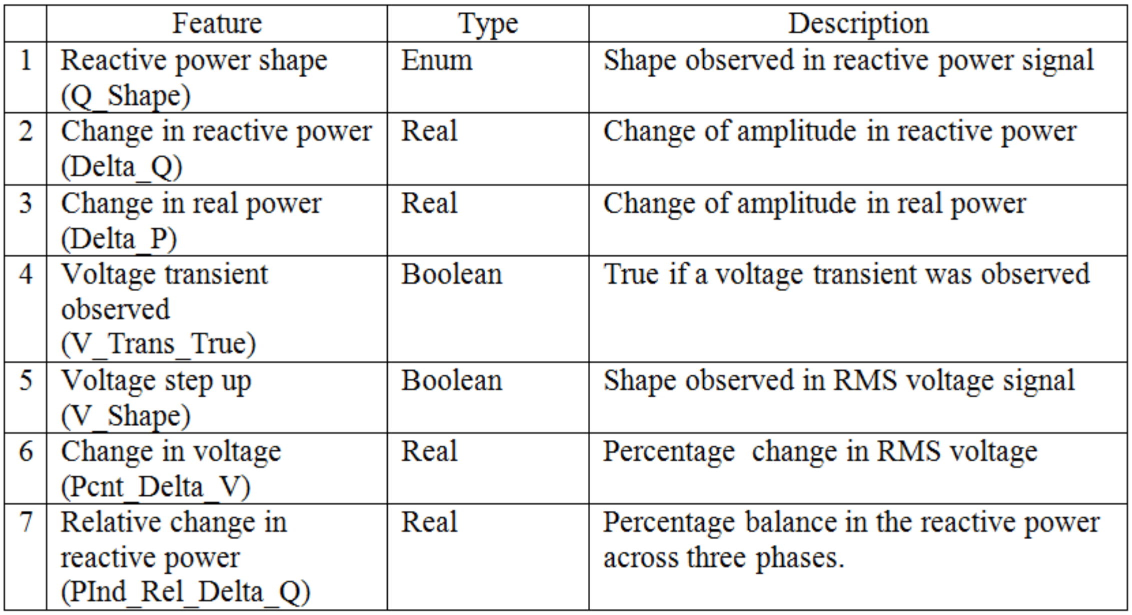
State Machine (FSM). The possibility values calculated in the previous processing step would serve as the fuzzy states of the fuzzy state machine. The ability of the BFPMs to act as FSMs is essential when the classifier has to go through multiple iterations while processing data that have multiple segments. The BFPMs need to remember the results of the previous segment(s) before making a decision on the current segment. This would allow the classifier to make a classification by combining the information obtained from multiple segments. The delay loop of BFPM is an optional component, and is used at the segment level inference in some modules.

The next three subsections will discuss how BFPMs are used for fuzzy inference at phase level, segment level and event level. To illustrate how fuzzy inference is used for classifying power distribution system events; an example application that uses fuzzy inference at each level to detect three phase capacitor switching on event will be shown. The rules used in the example application have been simplified for illustrative purposes. In the example application, it is assumed that waveform data was generated by a three phase balanced capacitor switching on operation. The example application will show fuzzy inference using a subset of features used by capacitor reactive power module. Table 6.2 shows lists the features used in the example application. Table 6.3 shows sample feature values for a capacitor switching on operation. The inference engine used by the hierarchical classifier allows the use of both fuzzy and crisp rules in tandem. In the following sub-sections, the term 'rule' is used to represent both fuzzy and crisp rules.

#### *6.2.1.2 Phase level inference*

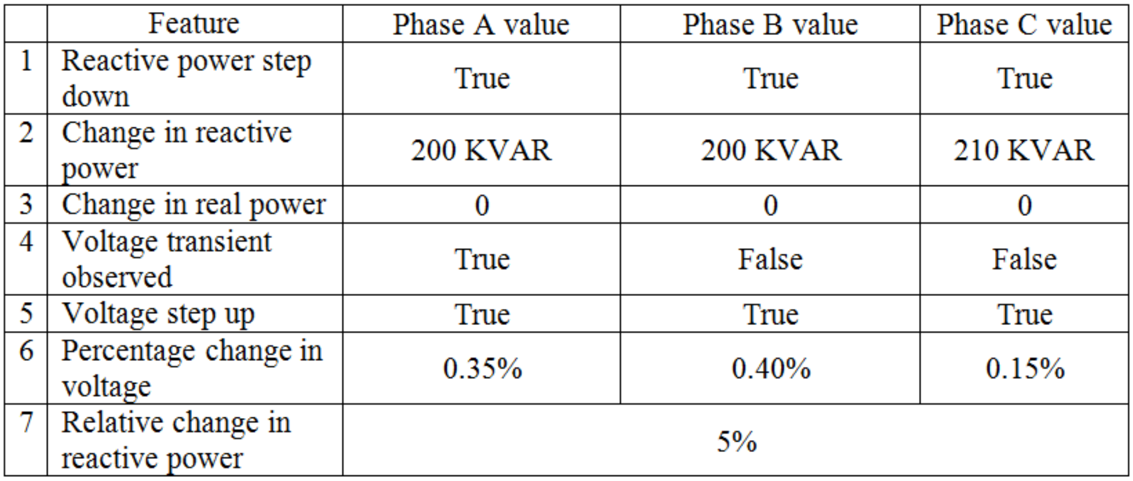
Phase level inference is the first level of feature processing within a module. At the phase level, input features corresponding to each phase A, B and C are

Table 6.2: Description of sample input features



| | Feature | Type | Description |
|---|---|---|---|
| 1 | Reactive power shape (Q_Shape) | Enum | Shape observed in reactive power signal |
| 2 | Change in reactive power (Delta_Q) | Real | Change of amplitude in reactive power |
| 3 | Change in real power (Delta_P) | Real | Change of amplitude in real power |
| 4 | Voltage transient observed (V_Trans_True) | Boolean | True if a voltage transient was observed |
| 5 | Voltage step up (V_Shape) | Boolean | Shape observed in RMS voltage signal |
| 6 | Change in voltage (Pcnt_Delta_V) | Real | Percentage change in RMS voltage |
| 7 | Relative change in reactive power (PInd_Rel_Delta_Q) | Real | Percentage balance in the reactive power across three phases. |

Table 6.3: Sample feature values

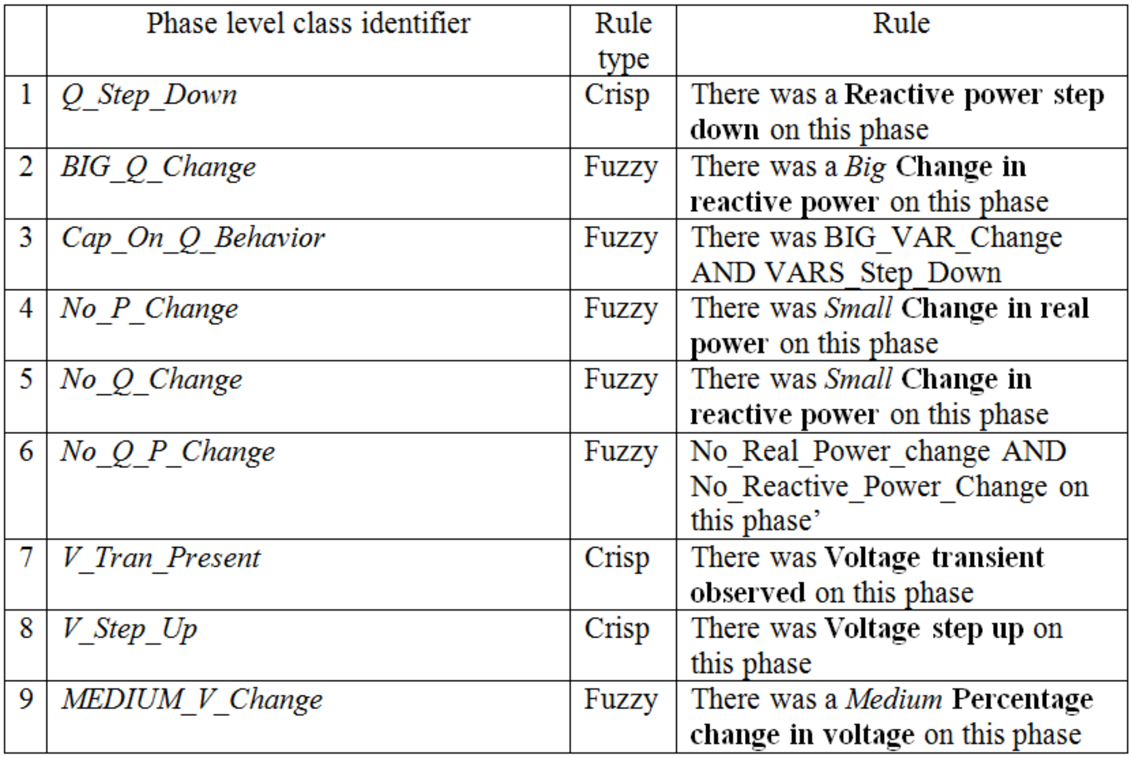


| | Feature | Phase A value | Phase B value | Phase C value |
|---|---|---|---|---|
| 1 | Reactive power step down | True | True | True |
| 2 | Change in reactive power | 200 KVAR | 200 KVAR | 210 KVAR |
| 3 | Change in real power | 0 | 0 | 0 |
| 4 | Voltage transient observed | True | False | False |
| 5 | Voltage step up | True | True | True |
| 6 | Percentage change in voltage | 0.35% | 0.40% | 0.15% |
| 7 | Relative change in reactive power | 5% | | |

processed independently by a phase level classifier (represented by BFPMs A, B and C in Figure 6.1). Phase level classifiers perform fuzzy inference using fuzzy phase level rules corresponding to each module. For example, in a capacitor reactive power module, a phase level classifier may detect whether or not a large step change in reactive power was detected on phase A. The phase level classifiers do not perform defuzzification. Instead, they output phase level possibility values for each phase. Phase level classifiers use the same set of rules to process features from each phase. This is possible because event signatures are not dependent on phase labels. For example, a capacitor switching on phase A will have similar electrical characteristics to a capacitor switching on phase B or C. One of the benefits of using phase level classifiers is that, they effectively partition the feature dimensions used by a module into three non overlapping sub-spaces. Hence, each phase level classifier uses approximately one third of the features needed by each module. It should be noted that each module may use some features that are not phase dependent. These features are not processed by phase level classifiers, but they are used by higher level classifiers. If multiple segments are detected in an event, then features corresponding to each of these segments are processed separately, and phase level possibility values are generated for each segment. An example of phase level inference for three phase capacitor switching on event will be shown next.

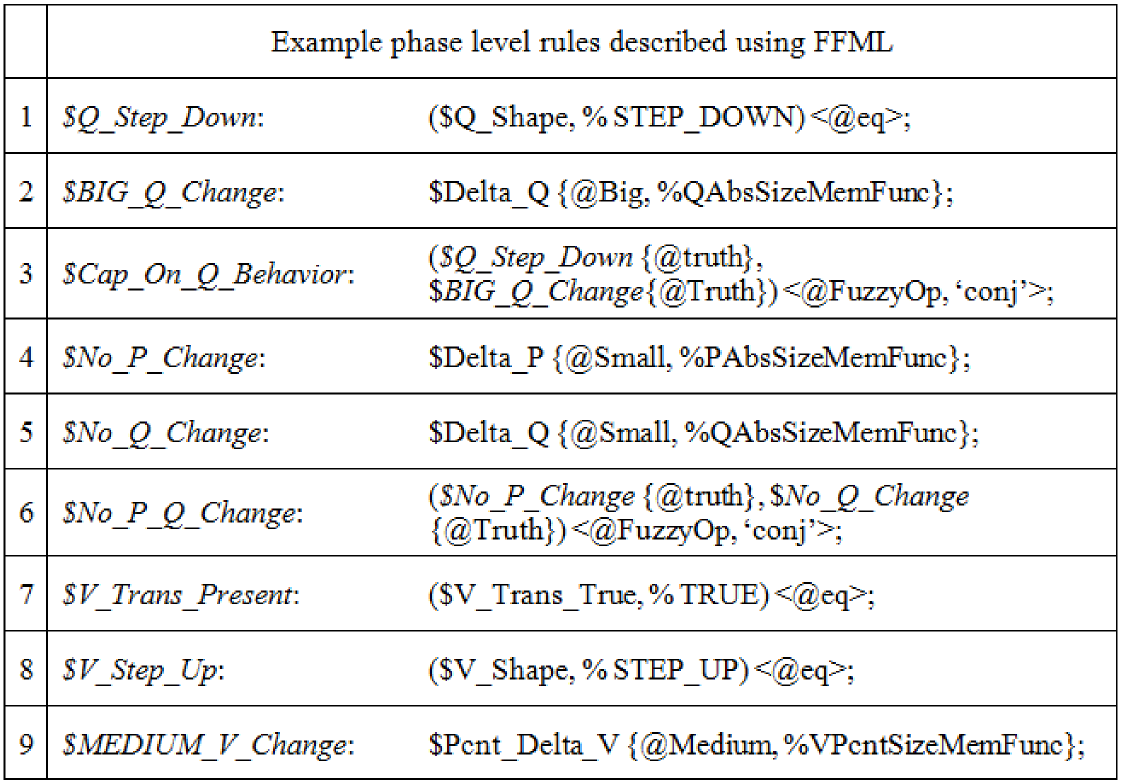
Phase level classifiers use inputs features extracted from signals of each phase (phases A, B and C), and assign phase level possibilities for phase level classes. The Table 6.4 shows a descriptive version of example rules used by phase level classifiers. these rules are used to assign output possibility values for phase level classes. Rules in Table 6.4 lists some observations that help to identify a capacitor switching on event. A capacitor switching on event on the monitored feeder

Table 6.4: Example phase level rules



| | Phase level class identifier | Rule type | Rule |
|---|---|---|---|
| 1 | *Q_Step_Down* | Crisp | There was a **Reactive power step down** on this phase |
| 2 | *BIG_Q_Change* | Fuzzy | There was a *Big* **Change in reactive power** on this phase |
| 3 | *Cap_On_Q_Behavior* | Fuzzy | There was BIG_VAR_Change AND VARS_Step_Down |
| 4 | *No_P_Change* | Fuzzy | There was *Small* **Change in real power** on this phase |
| 5 | *No_Q_Change* | Fuzzy | There was *Small* **Change in reactive power** on this phase |
| 6 | *No_Q_P_Change* | Fuzzy | No_Real_Power_change AND No_Reactive_Power_Change on this phase’ |
| 7 | *V_Tran_Present* | Crisp | There was **Voltage transient observed** on this phase |
| 8 | *V_Step_Up* | Crisp | There was **Voltage step up** on this phase |
| 9 | *MEDIUM_V_Change* | Fuzzy | There was a *Medium* **Percentage change in voltage** on this phase |

Table 6.5: Example phase level rules in FFML



| | Example phase level rules described using FFML | |
|---|---|---|
| 1 | *$Q_Step_Down*: | ($Q_Shape, % STEP_DOWN) <@eq>; |
| 2 | *$BIG_Q_Change*: | $Delta_Q {@Big, %QAbsSizeMemFunc}; |
| 3 | *$Cap_On_Q_Behavior*: | (*$Q_Step_Down* {@truth},<br>*$BIG_Q_Change*{@Truth}) <@FuzzyOp, ‘conj’>; |
| 4 | *$No_P_Change*: | $Delta_P {@Small, %PAbsSizeMemFunc}; |
| 5 | *$No_Q_Change*: | $Delta_Q {@Small, %QAbsSizeMemFunc}; |
| 6 | *$No_P_Q_Change*: | (*$No_P_Change* {@truth}, *$No_Q_Change*<br>{@Truth}) <@FuzzyOp, ‘conj’>; |
| 7 | *$V_Trans_Present*: | ($V_Trans_True, % TRUE) <@eq>; |
| 8 | *$V_Step_Up*: | ($V_Shape, % STEP_UP) <@eq>; |
| 9 | *$MEDIUM_V_Change*: | $Pcnt_Delta_V {@Medium, %VPcntSizeMemFunc}; |

is typically recognized by VARS (reactive power) stepping down, big change in VARS, and small or no change in real power. The capacitor switching on event on an adjacent feeder can be recognized by observing if a voltage transient was observed, if there was a medium percentage change in voltage, if the voltage stepped up and if there was not much change in VARS or real power. Table 6.5 shows the same rules expressed using FFML. FFML (Fuzzy Feature Manipulation Language) is a custom language that was developed for representing rules used by the fuzzy hierarchical classifier. FFML will be described in Section 6.3. The inference engine automatically evaluates all phase level rules for each phase and assigns phase level possibilities for phases A, B and C. Rule 1 in Table 6.4 is expanded as follows:

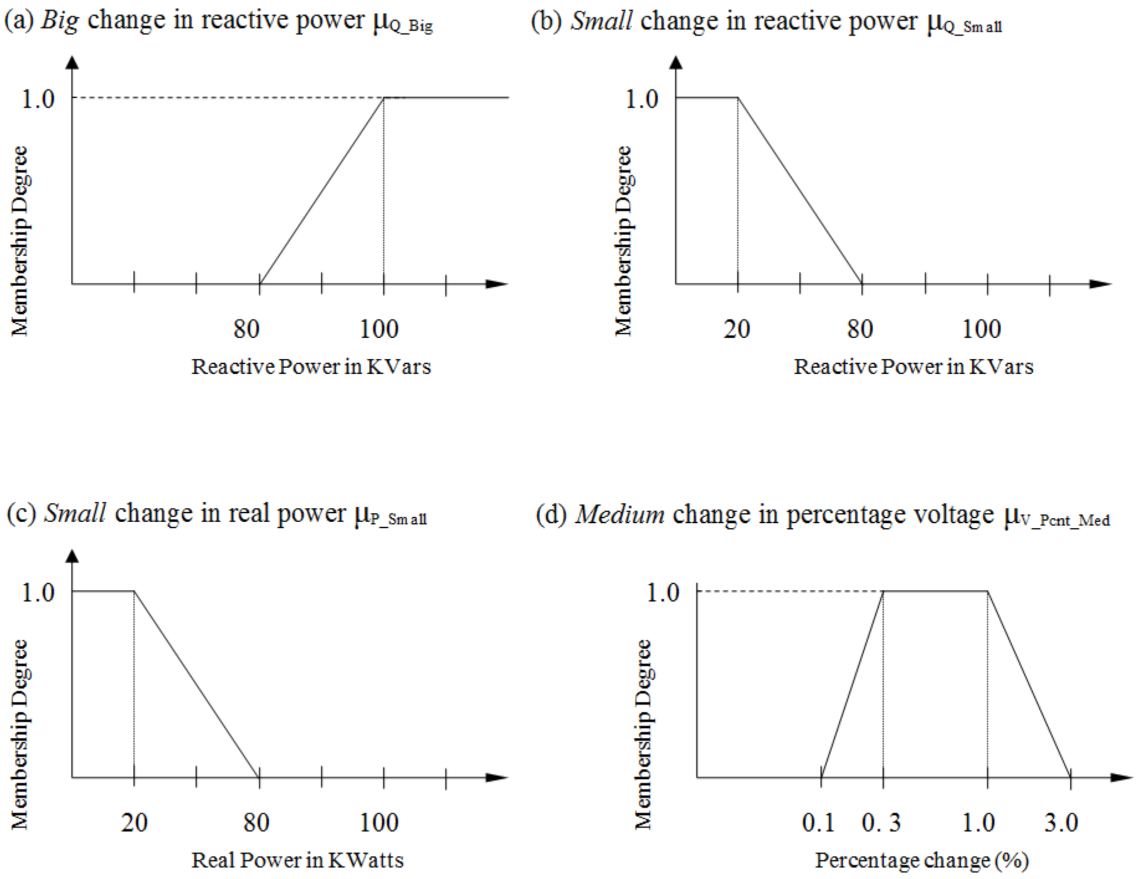


Figure 6.3: Example membership functions used by phase level rules

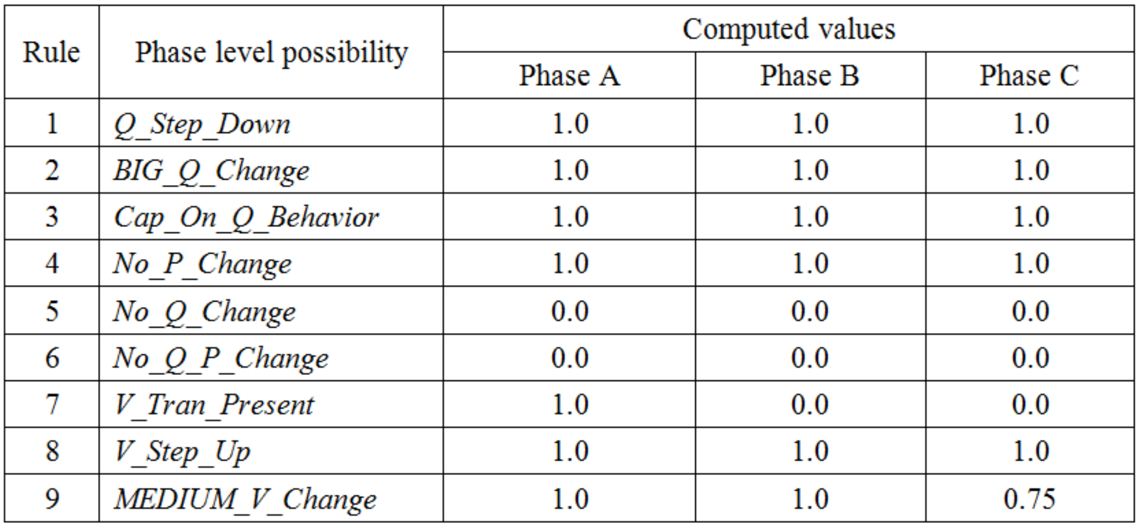
*Rule* $1_{PhaseA}$: IF there was a **Reactive power step down** observed on Phase A THEN Class IS *Q_Step_Down_Phase_A*

*Rule* $1_{PhaseB}$: IF there was a **Reactive power step down** observed on Phase B THEN Class IS *Q_Step_Down_Phase_A*

*Rule* $1_{PhaseB}$: IF there was a **Reactive power step down** observed on Phase C THEN Class IS *Q_Step_Down_Phase_A*

In all examples used in this section, the possibility value for a class *Cl* is represented by $\pi_{Cl}$. Rule 1 computes phase level possibility values for classes *Q_Step_Down_Phase_A, Q_Step_Down_Phase_B and Q_Step_Down_Phase_C* by evaluating the truth in the antecedent portion of the rule as follows:

Table 6.6: Sample phase level membership functions



| Rule | Phase level possibility | Computed values | | |
|---|---|---|---|---|
| | | Phase A | Phase B | Phase C |
| 1 | *Q_Step_Down* | 1.0 | 1.0 | 1.0 |
| 2 | *BIG_Q_Change* | 1.0 | 1.0 | 1.0 |
| 3 | *Cap_On_Q_Behavior* | 1.0 | 1.0 | 1.0 |
| 4 | *No_P_Change* | 1.0 | 1.0 | 1.0 |
| 5 | *No_Q_Change* | 0.0 | 0.0 | 0.0 |
| 6 | *No_Q_P_Change* | 0.0 | 0.0 | 0.0 |
| 7 | *V_Tran_Present* | 1.0 | 0.0 | 0.0 |
| 8 | *V_Step_Up* | 1.0 | 1.0 | 1.0 |
| 9 | *MEDIUM_V_Change* | 1.0 | 1.0 | 0.75 |

$$\pi_{Q\_Step\_down\_Phase\_A} = \mu_{Step\_down}\left(Q\_Shape[Phase\_A]\right)$$

$$\pi_{Q\_Step\_down\_Phase\_B} = \mu_{Step\_down}\left(Q\_Shape[Phase\_B]\right)$$

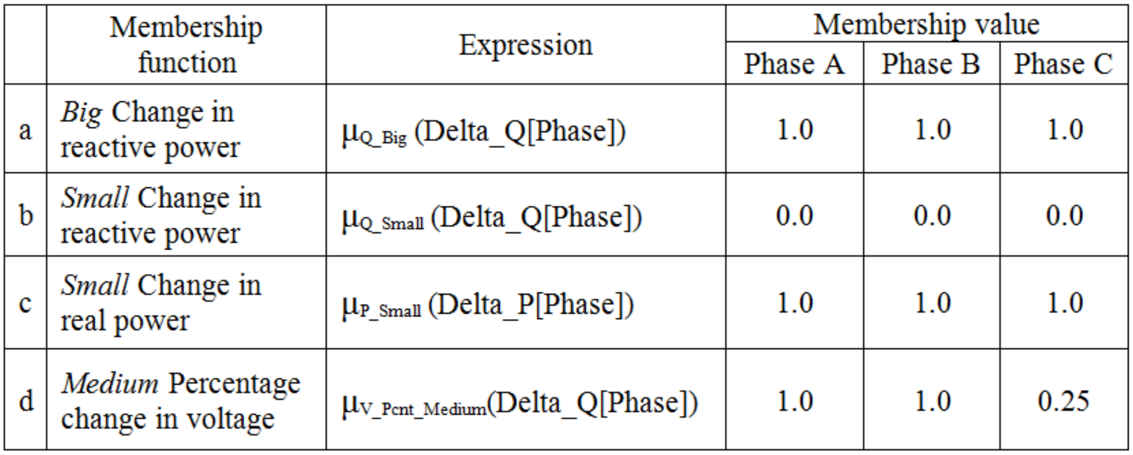
$$\pi_{Q\_Step\_down\_Phase\_C} = \mu_{Step\_down}\left(Q\_Shape[Phase\_C]\right)$$

where:

$$\mu_{Step\_down}\left(Shape\right) = \begin{cases} 1 & Shape = Step\_down \\ 0 & \text{Otherwise} \end{cases}$$

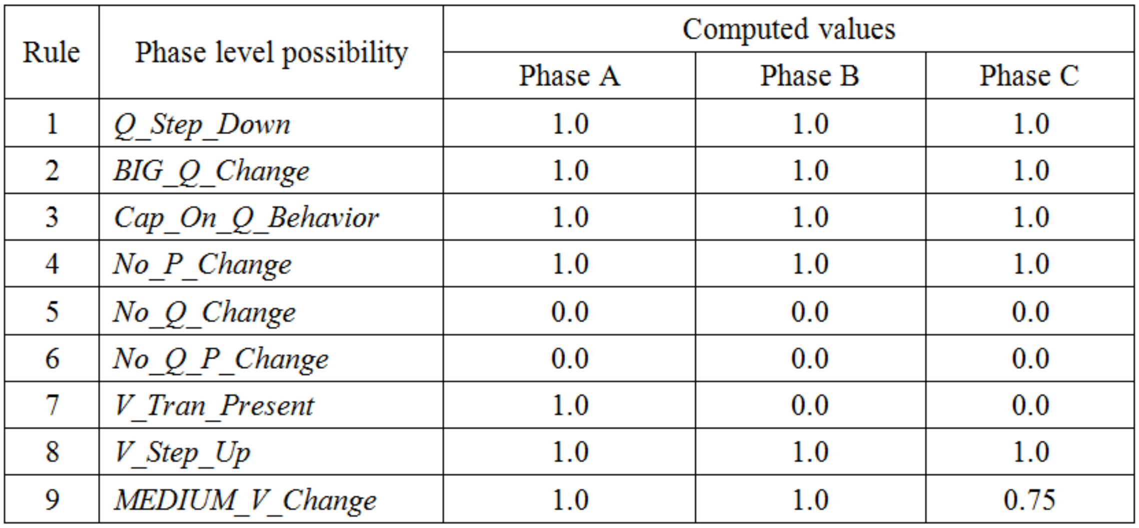
Rule 1 is a crisp rule that checks whether or not a downward step change in reactive power (Q) was observed on a given phase using shape feature for Q corresponding to phases A, B and C. The possibility values computed by Rule 1 is either 1 or 0. The rules 2, 4, 5 and 9 use linguistic descriptors like Big, Small and Medium. Membership functions (Figure 6.3) are required to evaluate these rules. Using feature values in Table 6.3 and the membership functions in Table 6.6, the input phase level features are fuzzified for use in rules 2, 4, 5 and 9. The calculated membership values are shown in Table 6.7.

Table 6.7: Sample computed membership values



| | Membership function | Expression | Membership value | | |
|---|---|---|---|---|---|
| | | | Phase A | Phase B | Phase C |
| a | *Big* Change in reactive power | $\mu_{Q\_Big}$ (Delta_Q[Phase]) | 1.0 | 1.0 | 1.0 |
| b | *Small* Change in reactive power | $\mu_{Q\_Small}$ (Delta_Q[Phase]) | 0.0 | 0.0 | 0.0 |
| c | *Small* Change in real power | $\mu_{P\_Small}$ (Delta_P[Phase]) | 1.0 | 1.0 | 1.0 |
| d | *Medium* Percentage change in voltage | $\mu_{V\_Pcnt\_Medium}$(Delta_Q[Phase]) | 1.0 | 1.0 | 0.25 |

Table 6.8: Sample phase level possibility values



| Rule | Phase level possibility | Computed values | | |
|---|---|---|---|---|
| | | Phase A | Phase B | Phase C |
| 1 | *Q_Step_Down* | 1.0 | 1.0 | 1.0 |
| 2 | *BIG_Q_Change* | 1.0 | 1.0 | 1.0 |
| 3 | *Cap_On_Q_Behavior* | 1.0 | 1.0 | 1.0 |
| 4 | *No_P_Change* | 1.0 | 1.0 | 1.0 |
| 5 | *No_Q_Change* | 0.0 | 0.0 | 0.0 |
| 6 | *No_Q_P_Change* | 0.0 | 0.0 | 0.0 |
| 7 | *V_Tran_Present* | 1.0 | 0.0 | 0.0 |
| 8 | *V_Step_Up* | 1.0 | 1.0 | 1.0 |
| 9 | *MEDIUM_V_Change* | 1.0 | 1.0 | 0.75 |

Rule 2 in Table 6.4 computes phase level possibility values for classes *BIG_Q_Change_Phase_A, BIG_Q_Change_Phase_B and BIG_Q_Change_Phase_C* by evaluating the truth in the antecedent portion of the rule as follows:

$\pi_{Big\_Q\_Change\_Phase\_A} = \mu_{Q\_Big}\left(Delta\_Q[Phase\_A]\right)$

$\pi_{Big\_Q\_Change\_Phase\_B} = \mu_{Q\_Big}\left(Delta\_Q[Phase\_B]\right)$

$\pi_{Big\_Q\_Change\_Phase\_C} = \mu_{Q\_Big}\left(Delta\_Q[Phase\_C]\right)$

where:

$\mu_{Q\_Big}$ is a membership function (Figure 6.3(a)) used to describe the degree to which change in Q is big

Rule 3 in Table 6.4 computes phase level possibility values for classes *Cap_On_Q_Behavior_Phase_A, Cap_On_Q_Behavior_Phase_B and Cap_On_Q_Behavior _Phase_C* by replacing 'AND' in Rule 3, Table 6.4 by fuzzy conjunction operator $\wedge$ as follows:

$$\pi_{Cap\_On\_Q\_Behavior\_Phase\_A} = \pi_{Q\_Step\_down\_Phase\_A} \wedge \pi_{Big\_Q\_Change\_Phase\_A} \tag{6.1}$$

$$\pi_{Cap\_On\_Q\_Behavior\_Phase\_B} = \pi_{Q\_Step\_down\_Phase\_B} \wedge \pi_{Big\_Q\_Change\_Phase\_B} \tag{6.2}$$

$$\pi_{Cap\_On\_Q\_Behavior\_Phase\_C} = \pi_{Q\_Step\_down\_Phase\_C} \wedge \pi_{Big\_Q\_Change\_Phase\_C} \tag{6.3}$$

Table 6.8 lists phase level possibility values computed using rules 1-9 and membership values computed in Table 6.7. The fuzzy conjunction operator 'Min' is used to evaluate the 'AND' condition in the rules. Phase level possibility values are used as inputs to segment level inference.

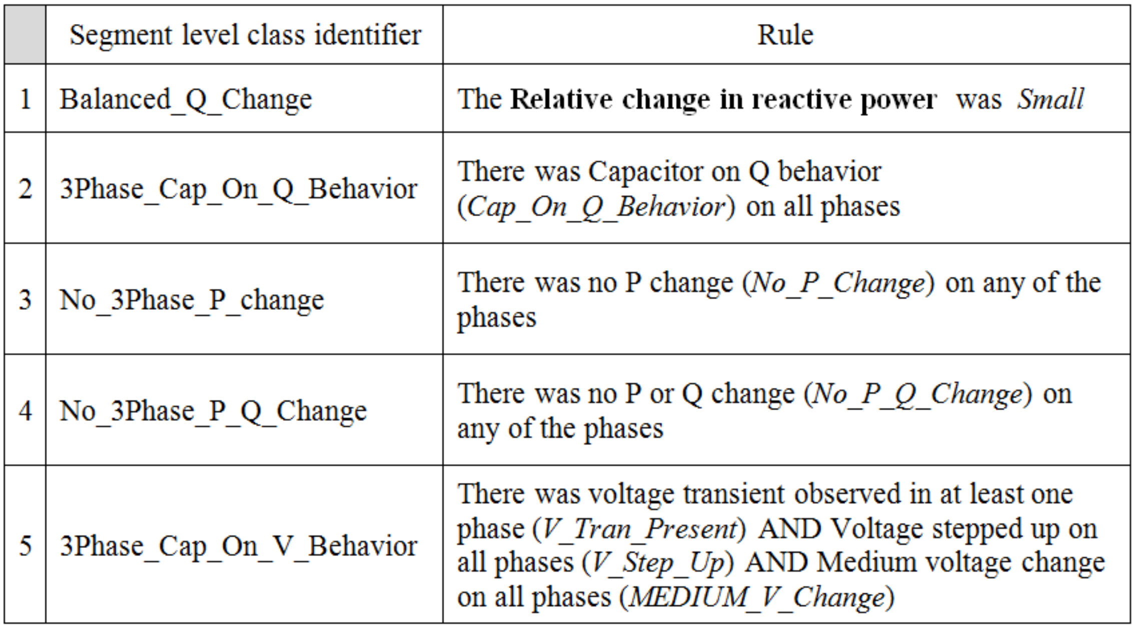
#### *6.2.1.3 Segment level inference*

This is the second level of processing within a module. Segment level classifiers use phase independent features, phase level possibility values and a subset of segment level possibility values from a previous segment as inputs. Unlike phase level classifiers that process each phase independently, segment level classifiers (represented by BFPMs Segment 1,. . . , Segment N) in Figure 6.1) do not process each segment independently. This is because, some events such as overcurrents can span multiple segments, and it is useful to track the event across multiple segments before a class label can be assigned. For each module, the number of output possibility values generated by phase level classifiers is much less when compared to the number of input features used by phase level classifiers. As a result, the dimension of the input space for segment level classifiers (comprised mostly of phase level possibility values) is also small. Segment level classifiers perform fuzzy inference using fuzzy segment level rules corresponding to each module. For example, in a capacitor reactive power module, a segment level classifier may detect whether or not a large step change in reactive power was detected on all phases within a segment. Segment level classifiers do not perform defuzzification. Instead, they output segment level possibility values for each segment. Segment level classifiers within a module use the same set of rules to process each segment. Each module may use some features that are independent of the segment or phase. These features are not processed by segment level classifiers. Instead, they are used by event level classifiers.

Example segment level inference for three phase capacitor switching on event

Each power system event may involve one or more phases. It is important to analyze the relative behavior of different phases within a segment. For the sake of simplicity, this example assumes that the data contains only a single segment. This example is a continuation of the example discussed for phase level inference in the previous subsection.

Table 6.9: Example segment level rules



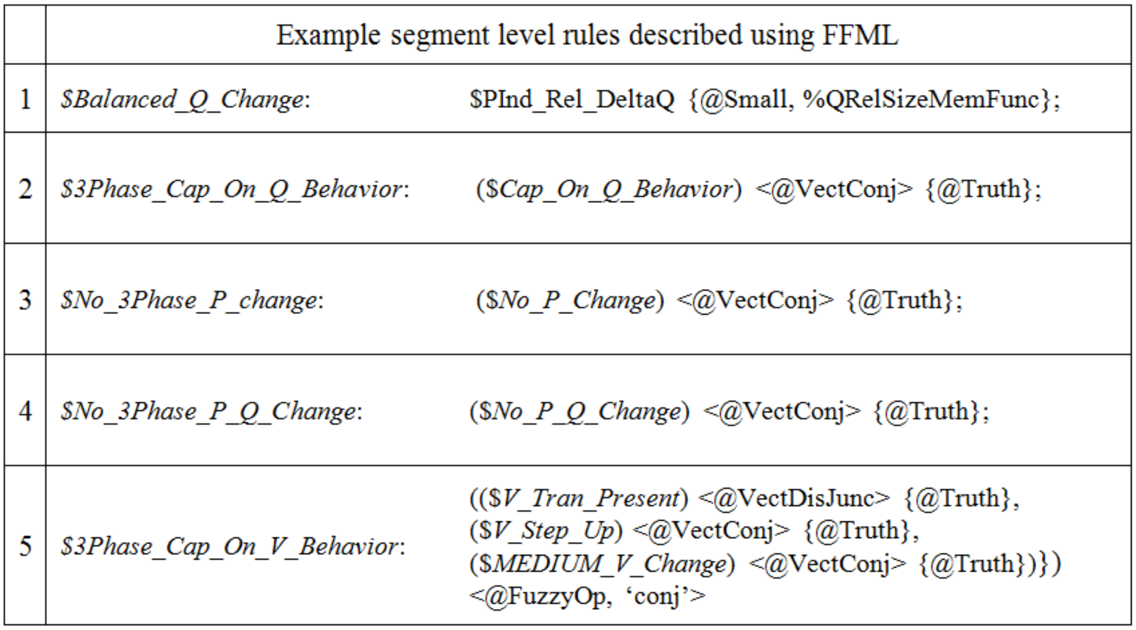
| | Segment level class identifier | Rule |
|---|---|---|
| 1 | Balanced_Q_Change | The **Relative change in reactive power** was *Small* |
| 2 | 3Phase_Cap_On_Q_Behavior | There was Capacitor on Q behavior (*Cap_On_Q_Behavior*) on all phases |
| 3 | No_3Phase_P_change | There was no P change (*No_P_Change*) on any of the phases |
| 4 | No_3Phase_P_Q_Change | There was no P or Q change (*No_P_Q_Change*) on any of the phases |
| 5 | 3Phase_Cap_On_V_Behavior | There was voltage transient observed in at least one phase (*V_Tran_Present*) AND Voltage stepped up on all phases (*V_Step_Up*) AND Medium voltage change on all phases (*MEDIUM_V_Change*) |

In the previous step, phase level possibility values were calculated for each phase, independent of one another (Table 6.8). The segment level rules combine phase level possibility values that were computed from different phases. They may also use as inputs, features that are not associated with any single phase (phase-independent features).

A descriptive version of sample segment level rules for the detecting three phase capacitor switching on event is listed in Table 6.9. Table 6.10 shows segment level rules represented using FFML.

Rule 1 in Table 6.9 requires computing membership degree of $PInd\_Rel\_DeltaQ$ (relative change in reactive power) using the membership function $\mu_{QRel\_Small}$ (*Small* relative change in reactive power) shown in Figure 6.4 . The value for the phase independent feature $PInd\_Rel\_DeltaQ$ is 0.05 for this example (feature 7, Table 6.3). $PInd\_Rel\_DeltaQ$ is a measure of reactive power imbalance between all

Table 6.10: Example segment level rules in FFML



| | Example segment level rules described using FFML | |
|---|---|---|
| 1 | *$Balanced_Q_Change*: | $PInd_Rel_DeltaQ {@Small, %QRelSizeMemFunc}; |
| 2 | *$3Phase_Cap_On_Q_Behavior*: | ($*Cap_On_Q_Behavior*) <@VectConj> {@Truth}; |
| 3 | *$No_3Phase_P_change*: | ($*No_P_Change*) <@VectConj> {@Truth}; |
| 4 | *$No_3Phase_P_Q_Change*: | ($*No_P_Q_Change*) <@VectConj> {@Truth}; |
| 5 | *$3Phase_Cap_On_V_Behavior*: | (($*V_Tran_Present*) <@VectDisJunc> {@Truth},<br>($*V_Step_Up*) <@VectConj> {@Truth},<br>($*MEDIUM_V_Change*) <@VectConj> {@Truth})})<br><@FuzzyOp, 'conj'> |

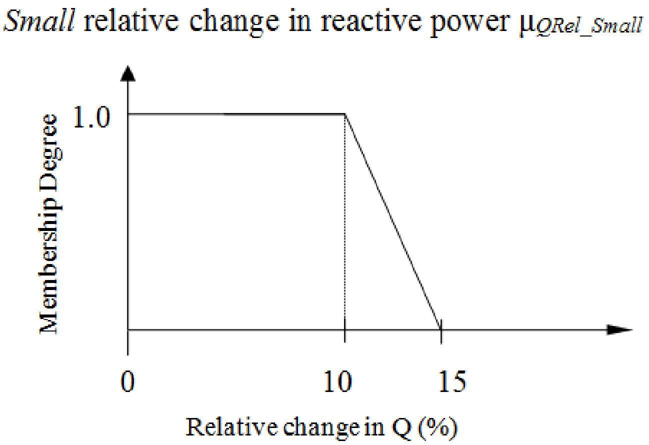


Figure 6.4: Example segment level membership function

phases and is computed using the following expression:

$$PInd\_Rel\_DeltaQ = \frac{\underset{ph}{Max}(Delta\_Q[ph]) - \underset{ph}{Min}(Delta\_Q[ph])}{\underset{ph}{Min}(Delta\_Q[ph])},$$

$$ph \in \{Phase\_A, Phase\_B, Phase\_C\}$$

Rule 1 computes segment level possibility value for class *Balanced_Q_Change* as follows:

$\pi_{Balanced\_Q\_Change} = \mu_{QRel\_Small}\left(PInd\_Rel\_DeltaQ\right)$

Rules 2, 3 and 4 in Table 6.9 use the phase level possibility values *Capacitor_On_Q_Behavior*, *No_P_Change* and *No_Q_Change* that have already been calculated for each phase in the previous level (Table 6.8). However, they also involve a condition 'on all phases'. This is equivalent to using 'AND' condition on individual phases. For example, Rule 2 in Table 6.9 should be interpreted as:

*Rule* 2: IF there was Capacitor on Q behavior on phase A AND
there was Capacitor on Q behavior on phase B AND
there was Capacitor on Q behavior on phase C
THEN Class is *3Phase_Cap_On_Q_Behavior*

The Boolean operator 'AND' is replaced by fuzzy conjunction operator 'Min', and Rule 2 is evaluated as follows:

$$\pi_{3Phase\_Cap\_On\_Q\_Behavior} = \pi_{Cap\_On\_Q\_Behavior\_Phase\_A} \wedge \pi_{Cap\_On\_Q\_Behavior\_Phase\_B} \wedge \pi_{Cap\_On\_Q\_Behavior\_Phase\_C} \quad (6.4)$$

$$\begin{aligned}\pi_{3Phase\_Cap\_On\_Q\_Behavior} = Min\,\big(&\pi_{Cap\_On\_Q\_Behavior\_Phase\_A},\\ &Min\,\big(\pi_{Cap\_On\_Q\_Behavior\_Phase\_B}\\ &\pi_{Cap\_On\_Q\_Behavior\_Phase\_C}\big)\,\big)\end{aligned} \tag{6.5}$$

Rule 2 in Table 6.10 shows a more compact representation of the rule using a vectorized form of fuzzy conjunction operator 'VectConj'. The fuzzy inference engine used by BFPMs supports vectorized form of operators such as fuzzy conjunction and disjunction that automatically combines possibility values from three phases (Equation 6.7). This avoids verbosity in the rule base, and also reduce user errors. Another advantage is that, vectorized operators help in reducing CPU and memory usage and hence improve efficiency. This is because a vectorized operator can be represented as a single node in a parse tree composed of a unary operator taking a vector as input argument. This is in contrast to using two binary operators to combine possibility values from three phases explicitly (Equation 6.5)

$$\begin{aligned}\pi_{3Phase\_Cap\_On\_Q\_Behavior} = Min\,\big(&\pi_{Cap\_On\_Q\_Behavior\_Phase\_A},\\ &\pi_{Cap\_On\_Q\_Behavior\_Phase\_B},\\ &\pi_{Cap\_On\_Q\_Behavior\_Phase\_C}\big)\end{aligned} \tag{6.6}$$

$$\pi_{3Phase\_Cap\_On\_Q\_Behavior} = VectMin\,\big(\pi_{Cap\_On\_Q\_Behavior}\big) \tag{6.7}$$

where:

$$VectMin\,(X) = \underset{ph}{Min}(x_{ph}),\ ph \in \{Phase\_A, Phase\_B, Phase\_C\}\,,$$
$$X = \{x_{Phase\_A}, x_{Phase\_B}, x_{Phase\_C}\}$$

and,

$$\pi_{Cap\_On\_Q\_Behavior} = \begin{bmatrix} \pi_{Cap\_On\_Q\_Behavior\_Phase\_A} \\ \pi_{Cap\_On\_Q\_Behavior\_Phase\_B} \\ \pi_{Cap\_On\_Q\_Behavior\_Phase\_C} \end{bmatrix} \tag{6.8}$$

Phase level possibility values for *Capacitor_On_Q_Behavior* were calculated using Equations 6.1-6.3, as shown in in row 3 of Table 6.8. Using these example phase level possibilities and equation 6.5, segment level possibility value for *3Phase_Cap_On_Q_Behavior* can be computed as follows:

$$\pi_{3Phase\_Cap\_On\_Q\_Behavior} = Min\left(1.0, 1.0, 1.0\right) = 1.0$$

In a similar fashion, *No_P_Change* and *No_P_Q_Change* may also be calculated using Rules 3 and 4. These values are shown in Table 6.8. Segment level rule, Rule 5 in Table 6.9 is a more complex rule. Rule 5 combines three conditions using the operator 'AND'. Within each of these conditions, phase level possibility values are used. The first condition 'There was voltage transient observed in at least one phase', uses the phase level possibility values *V_Trans_Present* that has already been computed for each phase in the previous step (feature 7, Table 6.8). However, the condition 'in at least one phase' should be interpreted as follows:

> "Voltage transient was observed on phase A OR voltage transient was observed on phase B OR voltage transient was observed on phase C"

Here, the individual phase level possibility values using the Boolean 'OR' operator.The fuzzy disjunction operator 'Max' can be used to replace the OR operator. Further, the vectorized version of Max operator 'VectMax' is used for computational efficiency. where:

$$VectMax\left(X\right) = \underset{ph}{Max}(x_{ph}),\ ph \in \left\{Phase\_A, Phase\_B, Phase\_C\right\},$$
$$X = \left\{x_{Phase\_A}, x_{Phase\_B}, x_{Phase\_C}\right\}$$

Rule 5 can be evaluated using vectorized fuzzy conjunction (‘*VectConj*’) and fuzzy disjunction operators (‘*VectDisJunc*’) as shown in row 5 of Table 6.10. *VectConj* operator is implemented using vectorized *Min* operator *VectMin,* and *VectDisjunc* operator is implemented using vectorized *Max* operator *VectMax* as follows:

$$\begin{aligned}\pi_{3Phase\_Cap\_On\_V\_Behavior} = VectMax\left(\pi_{V\_Trans\_Present}\right) \wedge \\ VectMin\left(\pi_{V\_Step\_Up}\right) \wedge \\ VectMin\left(\pi_{MEDIUM\_V\_Change}\right)\end{aligned} \tag{6.9}$$

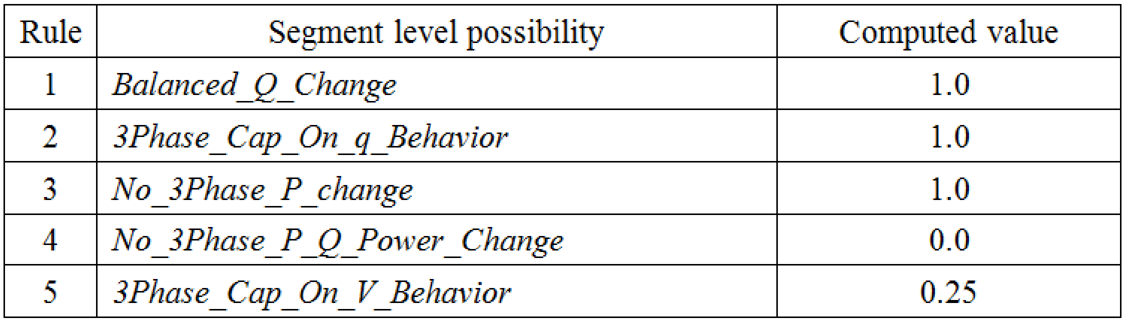
$$\begin{aligned}\pi_{3Phase\_Cap\_On\_V\_Behavior} = Min\left(VectMax\left(\pi_{V\_Trans\_Present}\right),\right. \\ VectMin\left(\pi_{V\_Step\_Up}\right) \\ \left.VectMin\left(\pi_{MEDIUM\_V\_Change}\right)\right)\end{aligned} \tag{6.10}$$

where:

$$\pi_{V\_Trans\_Present} = \begin{bmatrix} \pi_{V\_Trans\_Present\_Phase\_A} \\ \pi_{V\_Trans\_Present\_Phase\_B} \\ \pi_{V\_Trans\_Present\_Phase\_C} \end{bmatrix} \tag{6.11}$$

$$\pi_{V\_Step\_Up} = \begin{bmatrix} \pi_{V\_Step\_Up\_Phase\_A} \\ \pi_{V\_Step\_Up\_Phase\_B} \\ \pi_{V\_Step\_Up\_Phase\_C} \end{bmatrix} \tag{6.12}$$

Table 6.11: Sample segment level possibility values



| Rule | Segment level possibility | Computed value |
|---|---|---|
| 1 | *Balanced_Q_Change* | 1.0 |
| 2 | *3Phase_Cap_On_q_Behavior* | 1.0 |
| 3 | *No_3Phase_P_change* | 1.0 |
| 4 | *No_3Phase_P_Q_Power_Change* | 0.0 |
| 5 | *3Phase_Cap_On_V_Behavior* | 0.25 |

$$\pi_{MEDIUM\_V\_Change} = \begin{bmatrix} \pi_{MEDIUM\_V\_Chang\_Phase\_A} \\ \pi_{MEDIUM\_V\_Chang\_Phase\_b} \\ \pi_{MEDIUM\_V\_Chang\_Phase\_C} \end{bmatrix} \tag{6.13}$$

Using phase level possibility values from Table 6.8 and substituting these values in Equation 6.10, the possibility value for *3Phase_Cap_On_V_behavior* can be computed as:

$$\begin{aligned} \pi_{3Phase\_Cap\_On\_V\_Behavior} = Min\,( &VectMax\,([1.0, 0.0, 0.0])\,, \\ &VectMin\,([1.0, 1.0, 1.0]) \\ &VectMin\,([1.0, 1.0, 0.25])) \\ &= 0.25 \end{aligned}$$

The above example demonstrated how complex rules can be represented in compact form, and be evaluated efficiently using vectorized operators. The inference engine designed for the fuzzy hierarchical classifier automatically pareses and evaluates rules. Segment level possibility values computed in Table 6.11 are used as inputs to the event level of the inference engine.

#### *6.2.1.4 Event level inference*

This is the final level of processing within a module. Event level classifiers use segment independent features and segment level possibility values to compute class possibility values for the whole event. The dimension of the input space for event level classifiers is smaller in comparison to that of segment level classifiers. Event level classifiers (represented by BFPM Class 1,. . . , Class 1′ in Figure 6.1) use event level rules to perform fuzzy inference. Event level rules have fuzzy confidence degrees such as *Medium* and *High* associated with them. There may be multiple rules with the same class name on the consequent portion of the rule but with different confidence degrees associated with them. These confident degrees make the consequent portion of event level rules fuzzy. As a result, evaluating the antecedent portion of event level rules results in an output possibility distribution, instead of a single possibility value. Since event level rules have fuzzy consequents, defuzzification [107] is needed to compute single event level possibility value for each class. In order to calculate event level possibility values, the following operations are done as a four step inference process:

1. Possibility value for antecedent portion of the rule is computed based on input possibility values and fuzzified input features.

2. Possibility values computed using the antecedent portion of event level rules are used to clip the membership function associated with the consequent portion of the rule. This results in a possibility distribution for each event level rule.

3. Multiple rules may have the same event level class at the consequent portion of the rule. Possibility distributions corresponding to the same event level

class are combined to create a single possibility distribution for each event level class. Possibility distributions are combined using fuzzy disjunction operator *Max* over the output variable which is the possibility of an event level class.

4. Center of area defuzzification is then used to calculate a single possibility value for each event level class.

Typically, numerical confidence values have been used with fuzzy if then rules to convey an experts confidence in a rule [108]. In the second step of the inference process, rules that have singleton consequent portion (i.e. discrete output variable such as event category), result in single possibility value for each rule. Numerical confidence values are used to scale these possibility values. Using confidence degrees instead of numerical confidence values result in clipped possibility distributions, in the first step of inference. Combining the possibility distributions, and applying center of area defuzzification on the combined possibility distribution, provides a mechanism that allows all the event level rules to contribute to output possibility value. This inference process will be illustrated with an example in the next section.

Event level classifiers compute class possibilities for each segment, and then combine these class possibilities using the fuzzy disjunction operator 'Max', to output a single class possibility value for the whole event. The fuzzy disjunction operator is equivalent to using an 'OR' to combine the possibility values generated for each segment. This is equivalent to using the following rule to combine possibility values:

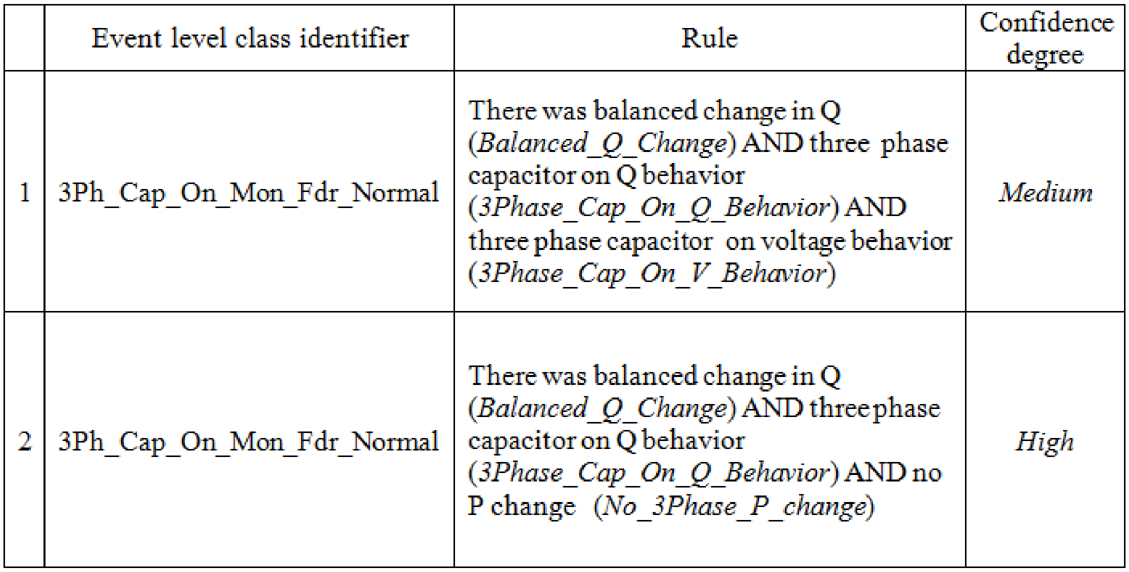
"If *segment 1 shows evidence of the cause being Class A* OR s*egment 2 shows evidence of the cause being Class A* OR ... *segment N shows evidence of the cause* *being Class A* Then *cause is Class A"*

One of the requirements placed on the classifier is the ability to assign phase, ground and position labels, in addition to event category labels. This requirement can potentially lead to rule explosion if every combination is enumerated. The event level classifier does not evaluate possibility values on a per phase basis. Depending on event category, possibility values are computed for single phase, two phase or three phase. Rules are not enumerated for each phase combination. This was done to reduce the number of rules. Similarly, not all ground and position categories are enumerated. The phase, ground and position attributes that are enumerated are listed under 'other attributes' in Table 6.1. For example, for 'Capacitor bank switching on' category (abbreviated as CAP-On), the following combinations are enumerated by using rules:

(CAP-On, Monitored feeder, Three phase, Grounded)
(CAP-On, Monitored feeder, Three phase, Ungrounded)
(CAP-On, Monitored feeder, single phase, Grounded)
(CAP-On, Non-monitored feeder, Three phase, Ground unknown)
(CAP-On, Non-monitored feeder, Single phase, Grounded)

Not all combinations of phase ground and position are enumerated for two reasons: 1. Certain combinations are not feasible, for example, single phase capacitors cannot operate in an ungrounded configuration and 2. Lack of knowledge to represent some combinations, for example, current implementation does not have the features and rules required to detect an ungrounded capacitor bank switching on a non-monitored feeder. Event possibility values for valid subset of class, phase, position and ground combinations are output for each module. These possibility values are used as inputs to the conflict resolution and class as-

Table 6.12: Example event level rules



| | Event level class identifier | Rule | Confidence degree |
|---|---|---|---|
| 1 | 3Ph_Cap_On_Mon_Fdr_Normal | There was balanced change in Q (*Balanced_Q_Change*) AND three phase capacitor on Q behavior (*3Phase_Cap_On_Q_Behavior*) AND three phase capacitor on voltage behavior (*3Phase_Cap_On_V_Behavior*) | *Medium* |
| 2 | 3Ph_Cap_On_Mon_Fdr_Normal | There was balanced change in Q (*Balanced_Q_Change*) AND three phase capacitor on Q behavior (*3Phase_Cap_On_Q_Behavior*) AND no P change (*No_3Phase_P_change*) | *High* |

signment stage.

Example event level inference for three phase capacitor switching on event : The truth values that were obtained from the segment level rules and features that are independent of the segment (i.e., common to all segments), are used as inputs to the event level rules. This example is a continuation of the three phase capacitor switching on example discussed in the previous subsection.

In the previous step, segment level possibility values were calculated (Table 6.11). A descriptive version of sample segment level rules for the detecting three phase capacitor switching on event are listed in Table 6.12 . Table 6.13

shows segment level rules represented using FFML. Event level rules may have confidence degrees associated with them. Confidence degrees reflect expert's confidence in the antecedent portion of a rule as being indicative of the class in the consequent portion of the rule. For example, Rule 1 in Table 6.12 is interpreted as follows:

Table 6.13: Example event level rules in FFML

| | Example event level rules described using FFML | |
|---|---|---|
| 1 | *$3Ph_Cap_On_Mon_Fdr_Normal* {@*confMedium, %eventPossib*}: | (*$Balanced_Q_Change* {@truth},<br>*$3Phase_Cap_On_Q_Behavior* {@Truth},<br>*$3Phase_Cap_On_V_Behavior* {@Truth})<br><@FuzzyOp, 'conj'>; |
| 2 | *$3Ph_Cap_On_Mon_Fdr_Normal* {@*confHigh, %eventPossib*} : | (*$Balanced_Q_Change* {@truth},<br>*$3Phase_Cap_On_Q_Behavior* {@Truth},<br>*$No_3Phase_P_change* {@Truth})<br><@FuzzyOp, 'conj'>; |

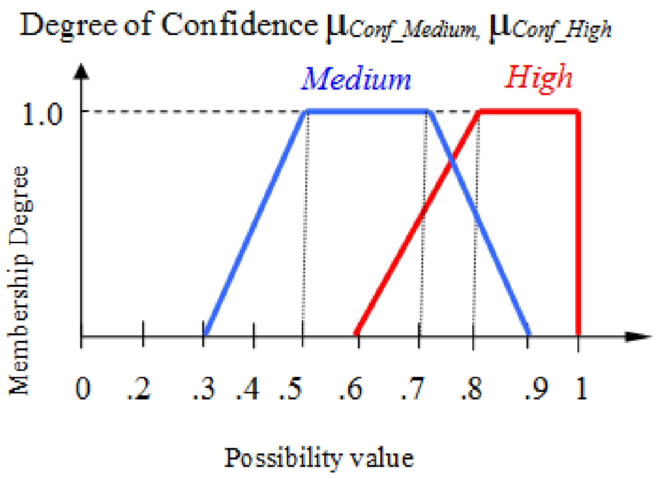


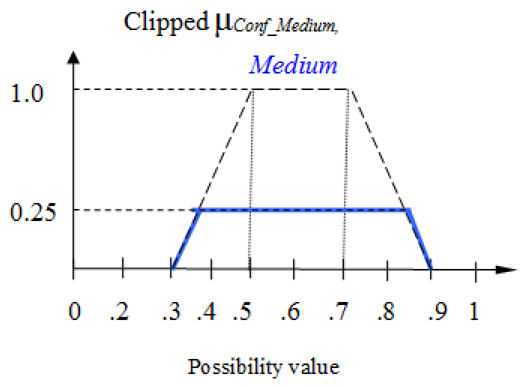
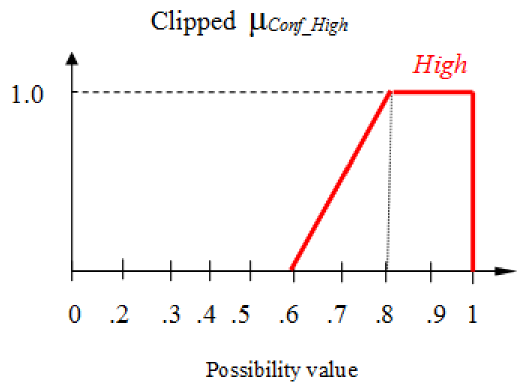
Figure 6.5: Confidence degree membership functions

*Rule* 1: IF there was balanced change in Q AND
there was three phase capacitor on Q behavior AND
there was three phase capacitor on voltage behavior
THEN Class is 'Three phase monitored feeder normal capacitor
switching on (*3Ph_Cap_On_Mon_Fdr_Normal*)' with
a *Medium* degree of confidence

FFML allows assigning confidence degrees in the form of fuzzy membership functions to the consequent portion of the rule. This is done by specifying an output variable 'Event possibility' (*eventPossib*) and a confidence degree membership function (*confMedium*) as shown in table 6.13. In Table 6.12, the class identifier is the same while the antecedent portion (evidence) used by the two rules differ. Rule 1 uses '*3Phase_Capacitor_On_V_Behavior*', while rule 2 uses *'No_3Phase_P _change*'. This is because of the expert's belief that the evidence used by Rule 2 are more indicative of three phase capacitor switching on than evidences used by Rule 1. Consequently, Rule 2 was assigned higher confidence degree than Rule 1. The confidence degrees are static and are part of the rule. Confidence degrees are not assigned during inference and are not dependent on the truth in the antecedent portion of the rule. Figure 6.5 shows *Medium* and *High* fuzzy membership functions for degree of confidence.

In order to evaluate event level rules, first antecedent possibility value is computed using segment level possibility values (Table 6.11).Then, the antecedent possibility value is used to clip degree of confidence membership function associated with that rule. This is shown in Table 6.14. Clipped degree of confidence membership functions represent a possibility distribution. Before a possibility value can be assigned to an event level class, consequent possibility distributions

Table 6.14: Example clipping using computed antecedent possibility value

| Rule | Computation of antecedent possibility value | Example antecedent possibility values | Clipped degree of confidence membership functions |
|---|---|---|---|
| 1 | $\pi_{Balanced\_Q\_Change} \wedge$<br>$\pi_{3Phase\_Cap\_On\_Q\_Behavior} \wedge$<br>$\pi_{3Phase\_Cap\_On\_V\_Behavior}$ | Min (1.0, 1.0,.25) = 0.25 |  |
| 2 | $\pi_{Balanced\_Q\_Change} \wedge$<br>$\pi_{3Phase\_Cap\_On\_Q\_Behavior} \wedge$<br>$\pi_{No\_3Phase\_P\_Change}$ | Min (1.0, 1.0, 1.0) = 1.00 |  |

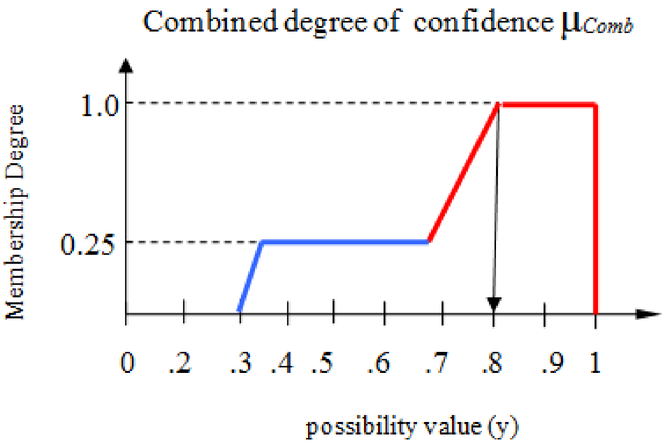


Figure 6.6: Combined consequent possibility distribution

of all the rules referring to that event level class need to be combined. This is done by superimposing the clipped degree of confidence membership functions using a *Max* operator. The resulting combined possibility distribution $\mu_{Comb}$ is shown in Figure 6.6. However, a single possibility value is needed for every event level class. Then, these event level class possibility values can easily be compared, a single class label for the event data can be assigned. In this example, a single confidence value needs to be calculated for 'Three phase monitored feeder normal capacitor switching on' category. Center of Area (COA) defuzzification is used to obtain final possibility value from combined membership function $\mu_{Comb}$ as shown below:

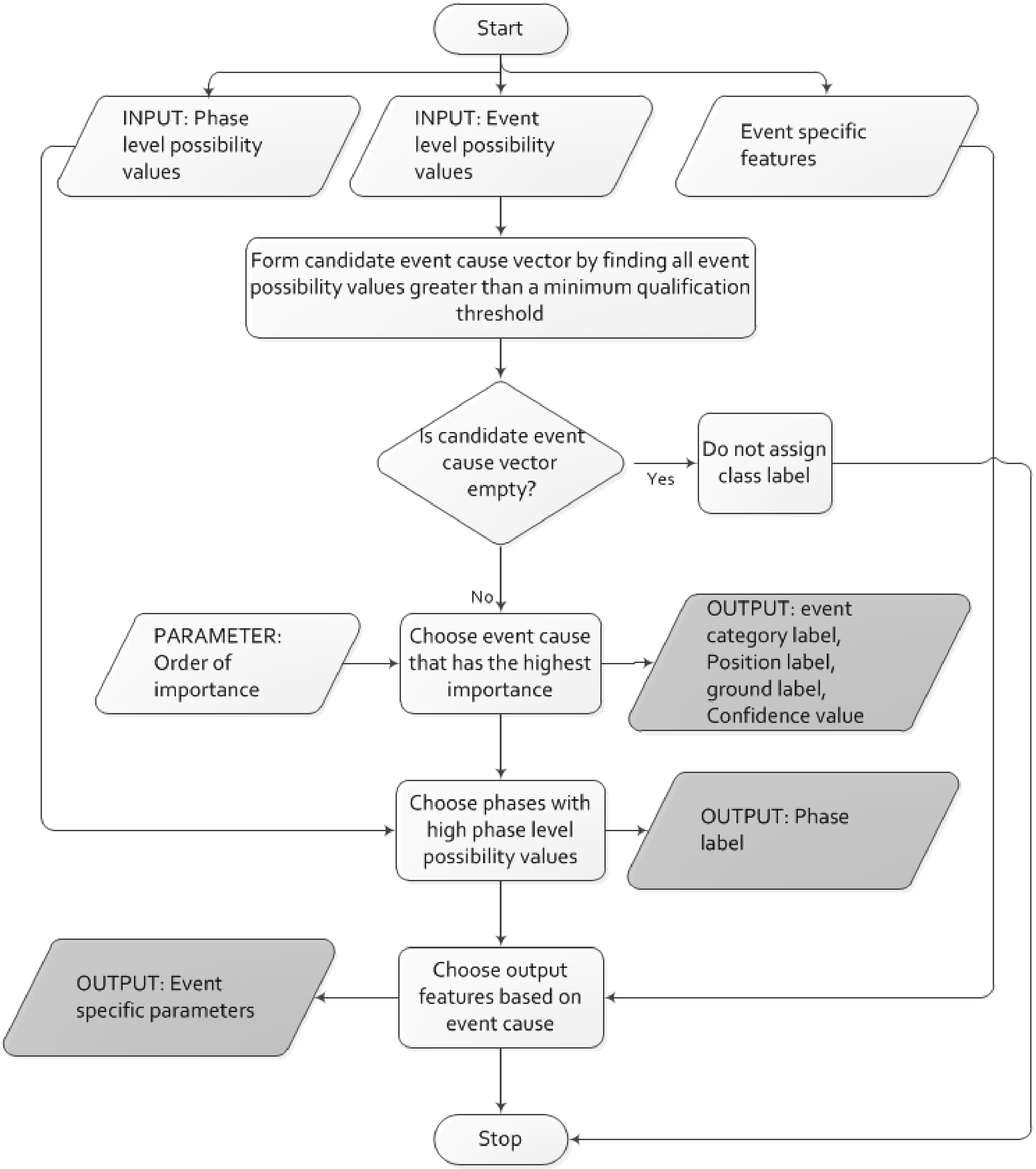
$$\pi_{3Ph\_Cap\_On\_Mon\_Fdr\_Normal} = \frac{\sum\limits_{N_{inf}}^{i=0} \mu_{Comb}\left(y_i\right) \times y_i}{\sum\limits_{N_{inf-1}}^{i=0} \mu_{Comb}\left(y_i\right)} \tag{6.14}$$

In the above equation, the combined possibility distribution $\mu_{Comb}$ is sampled at $N_{inf} + 1$ discrete points $y_0, \ldots, y_i, \ldots, y_{N_{inf}}$, where $y_i = i/N_{inf}$. Then COA is computed using these sample points $y_i$. For the example rules used in Table 6.14, the event level possibility value $\pi_{3Ph\_Cap\_On\_Mon\_Fdr\_Normal}$ computed using COA method of defuzzification was 0.8. Event level possibility values are computed for all other classes (Table 6.1) in a similar fashion. These event level possibility values are then used to assign output class labels and to choose appropriate output parameters.

### 6.2.2 *Conflict Resolution and Class Assignment*

Conflict resolution and class assignment are the final stage of fuzzy hierarchical classifier. This stage uses event level possibility values computed by all modules. It uses these possibility values to assign a single combination of event category, phase, position and ground label. The output label can be considered as a point in four dimensional space spanned by the combination of event category, phase, position and ground labels. Assigning a single label to the whole event is challenging as power system event data can represent an arbitrary duration in time. During this period, the data may contain waveforms caused by one or more related or non related power system phenomenon. For example, event data may contain waveforms that were caused by capacitor switching followed by a three phase motor start. This will result in high possibility values from both the capacitor module and motor module. This causes a conflict. One of the objectives of the final stage is to assign a label that will be perceived as most important by an expert when event data contains waveforms corresponding to more than one power system phenomenon.

Figure 6.7 shows the flow chart for conflict resolution and class assignment stage. Candidate event cause vector is formed by screening event level possibility values and choosing values greater than a minimum qualification threshold. Then, the chosen event causes are ranked based on their order of importance. The order of importance is predefined and determined based on expert knowledge.



Start
INPUT: Phase level possibility values
INPUT: Event level possibility values
Event specific features
Form candidate event cause vector by finding all event possibility values greater than a minimum qualification threshold
Is candidate event cause vector empty?
Yes
Do not assign class label
No
PARAMETER: Order of importance
Choose event cause that has the highest importance
OUTPUT: event category label, Position label, ground label, Confidence value
Choose phases with high phase level possibility values
OUTPUT: Phase label
OUTPUT: Event specific parameters
Choose output features based on event cause
Stop

Figure 6.7: Conflict resolution and class assignment

Table 6.15: Hypothetical event level possibility values

| Event cause category | Event level possibility |
|---|---|
| (CAP-On, Monitored feeder, Three phase, Grounded) | 0.7 |
| (Motor start, Monitored feeder, Three phase) | 0.8 |
| (Load step up, Monitored feeder, Three phase) | 0.7 |

Table 6.16: Sample order of importance

| event category | Position | Phase | Ground | Rank |
|---|---|---|---|---|
| CAP-On | Monitored feeder | X | X | 10 |
| CAP-On | Non-Monitored feeder | X | X | 25 |
| Motor start | Monitored feeder | X | X | 23 |
| Motor start | Non-Monitored feeder | X | X | 25 |
| Load step up | Monitored feeder | X | X | 23 |

The candidate event cause with the highest importance is chosen and is used to assign event category label, position label and ground label. It is possible for more then one event cause to have the same level of importance. In this case, the event with higher possibility value is chosen.

Consider the hypothetical scenario where event data contained waveforms caused by the following sequence of events: three phase capacitor bank switching on, followed by a motor start, followed by a large three phase load switching on. Table 6.15 lists hypothetical possibility values generated by corresponding event level classifiers. The possibility values have been ordered by importance first and then by the possibility value. Table 6.16 shows a sample rank assignment for a subset of event categories. The smaller the value of rank, higher the importance. A 'X' indicates that the attribute is not used for determining the rank. For the hypothetical set of possibility values listed in Table 6.15, the combination (CAP-On, Monitored feeder, Three phase, Grounded) will be chosen as the event cause. These also form the output attributes: (event category label = CAP-On, position label = Monitored feeder, phase label = ABC, ground label = Yes, Confidence =

Table 6.17: Mapping between possibility values and confidence label

| Possibility value range | Confidence label |
|---|---|
| [0.0, 0.15) | Low |
| [0.15, 0.30) | Medium Low |
| [0.30, 0.45) | Medium |
| [0.45, 0.60) | Medium High |
| [0.60, 0.75) | High |
| [0.75, 1.0] | Very High |

Medium high). Table 6.17 shows the mapping between possibility values and confidence label. Confidence labels provide a simple mechanism to inform users of the confidence the algorithm has on it's decision based on evidence found in the data. .

In the above example, choice of phase label was simplified because the phase attribute was three phase. Event level classifiers with phase attribute of single phase or two phase may also output high possibility values. When the phase attribute is single phase or two phase, then the specific phase combination should be determined and used as output phase label. This is done by using phase level possibility values that have already been computed by phase level classifiers. Each event category type has a predefined phase level classifier assigned to it for the purpose of determining phase label. When the phase attribute is single phase, the phase of the highest phase level possibility value is used to determine phase label. When the phase attribute is two phase, the phase of the top two phase level possibility values are used to determine phase label. For example, consider the scenario where the highest event level possibility was generated by the following combination of attributes: (CAP-On, Monitored feeder, *Single phase*, Grounded). Single phase implies, the phase label can be 'Phase A', 'Phase B' or 'Phase C'. Clearly, more information is needed to compute a phase label. Phase level possibilities that best represent capacitor switching on characteristics are used to accomplish this. In this example, the phase level possibility values *Cap_On_Q_Behavior* (Tables 6.4 and 6.8) computed for each phase can be used. Then, the phase with highest possibility value can be used to choose the phase label.

## 6.3 Design of a Large Scale Fuzzy Inference Engine

Fuzzy hierarchical classifier, as described in the previous section, uses few hundred rules distributed across different classifier modules and inference levels within a module. The number of rules are likely to increase as new modules are added. Further, rules at a higher level within the hierarchical classifier cannot be evaluated until output possibilities have been computed for the lower level classifiers. This is because, the antecedent portion of rules in higher level classifiers depend on the consequent portion of rules in lower level classifiers. This introduces dependency between rules used by the hierarchical classifier. A dependency relationship needs to be first established, and the rules need to be evaluated in the correct order. For example, consider the following set of rules:

Rule $H$: IF $expression_H(A_H^1, ..., A_H^n)$ THEN $C^1$

Rule $L1$: IF $expression_{L1}(A_{L1}^1, ..., A_{L1}^n)$ THEN $A_H^1$

.

.

.

Rule $Ln$: IF $expression_{Ln}(A_{Ln}^1, ..., A_{Ln}^n)$ THEN $A_H^n$

where, for a rule $H$, $expression_H()$ on the antecedent portion of the rule, is a fuzzy expression involving possibility values $A_H^1, ..., A_H^n$ and fuzzy operators. $C^1$is the possibility value computed as a result of evaluating the expression on the antecedent portion of the rule $H$. Before Rule $H$ can be evaluated, all possibility values $A_H^1, ..., A_H^n$ need to be known. This implies, all rules $(L1, ..., Ln)$ that have a consequent portion matching any of $A_H^1, ..., A_H^n$ have to be evaluated first. This form of multi-level inferences requires inference engines that support rule chaining. However, with the exception of FuzzyShell [105], there were no readily available, general purpose fuzzy inference engines that could handle large number of rules and rule chaining.

### *6.3.1 FuzzyShell vs. Custom Inference Engine*

FuzzyShell extended CLIPS [109], a widely used expert system shell to allow fuzzy inference. Similar to CLIPS, FuzzyShell uses production systems [110] and Rete networks [111] to support rule chaining in a large scale expert system. FuzzyShell met the requirements of an inference engine to be used with fuzzy hierarchical classifier. However, it was not chosen for the following reasons:

1. While FuzzyShell is very effective as a stand alone expert system shell. However, integrating the shell with an existing application can be challenging. Prior experience gained by working with a similar expert system shell, CLIPS [52] showed that passing large amount of data from and to expert system shells can be both tedious and inefficient. Using a custom inference engine designed specifically for the fuzzy hierarchical classifier will help to overcome this issue by providing greater flexibility for data transfer.

2. The fuzzy hierarchical classifier needs to work on embedded environments where computational resources are limited. Computational efficiency and memory usage are a primary concern. Parsing and evaluating a large rule base during run-time is an expensive operation. General purpose expert system shells are designed to be very flexible. They do not make any assumptions about when and in what order inputs are available. Rules are

fired when all the inputs to a rule are available. Hence, there is a significant overhead in tracking input variables and order in which the rules need to be executed. In contrast, the requirements for the inference engine used by the fuzzy hierarchical classifier are far less stringent. For a given rule base, the rules and the input features (not the feature values) used by the hierarchical classifier are fixed and fully defined. This implies that the execution plan for the rules is deterministic and fixed for the rule base. I.e., the order in which the rules are executed, and the order in which input variables are consumed can be defined prior to execution of the rules. If a custom inference engine is used, it is then possible to parse the rules, and create an execution plan as a sequence of operation on input variables. This execution plan is analogous to using compiled code. The inference engine can then be replaced by a virtual machine that runs the execution plan. This would result in improved efficiency both in terms of memory and processor usage.

3. A custom fuzzy inference engine tailored for fuzzy hierarchical classifier provides more flexibility in incorporating the hierarchical structure. Another advantage of a custom inference engine is that it can better exploit the symmetry introduced by the three phase nature of power system feature analysis. This would help to reduce redundancy in the rule base.

Based on the above considerations, a custom fuzzy inference engine, Fuzzy Feature Analysis Engine (FFAE) and a custom language for fuzzy rule base, Fuzzy Feature Manipulation Language (FFML) were developed. With the exception of conflict resolution and class assignment, all the other stages (phase level, segment level and event level inference) of fuzzy hierarchical classifier described in the Section 6.2 were functionally replaced by FFAE.

## 6.4 Fuzzy Feature Analysis Engine (FFAE)

Figure 6.8 shows the schematic of FFAE. FFAE consists two modules; an offline rule optimizer and compiler, and an online delayed fuzzy inference module. The offline rule optimizer and compiler parses the FFML rule base and produces a highly optimized compiled FFML code block. FFML code block contains a series of instructions to be performed on input features and computed possibility values. As the rule optimizer and compiler is an offline module, it does not do fuzzy inference using input features. Inference is delayed till the compiled FFML code block is executed by the online delayed fuzzy inference module. The delayed fuzzy inference module uses a virtual machine that executes the compiled FFML code block by plugging in input feature values. These two modules will be described in the following subsections.

### *6.4.1 Offline Rule Optimizer and Compiler*

Offline rule optimizer and compiler consists of three modules as seen in Figure 6.8. All the FFML rules used by fuzzy hierarchical classifier are first parsed using Spirit parser [112]. A production system creates an execution plan using the information generated by the parser. The execution plan is an ordering of rules based on their dependency. For example, if the antecedent portion a rule A is dependent on the consequent part of another rule B, then the rule A is defined to be dependent on rule B. Hence, rule B will be added to the execution plan before rule A. Finally, a FFML compiler compiles the rules in the same order as they appear on the execution plan and generates a FFML code block. The code block contains the sequence of operations to be performed on input features in order to compute possibility values. Each of the three modules used by rule optimizer and compiler will be explained further in the following paragraphs.

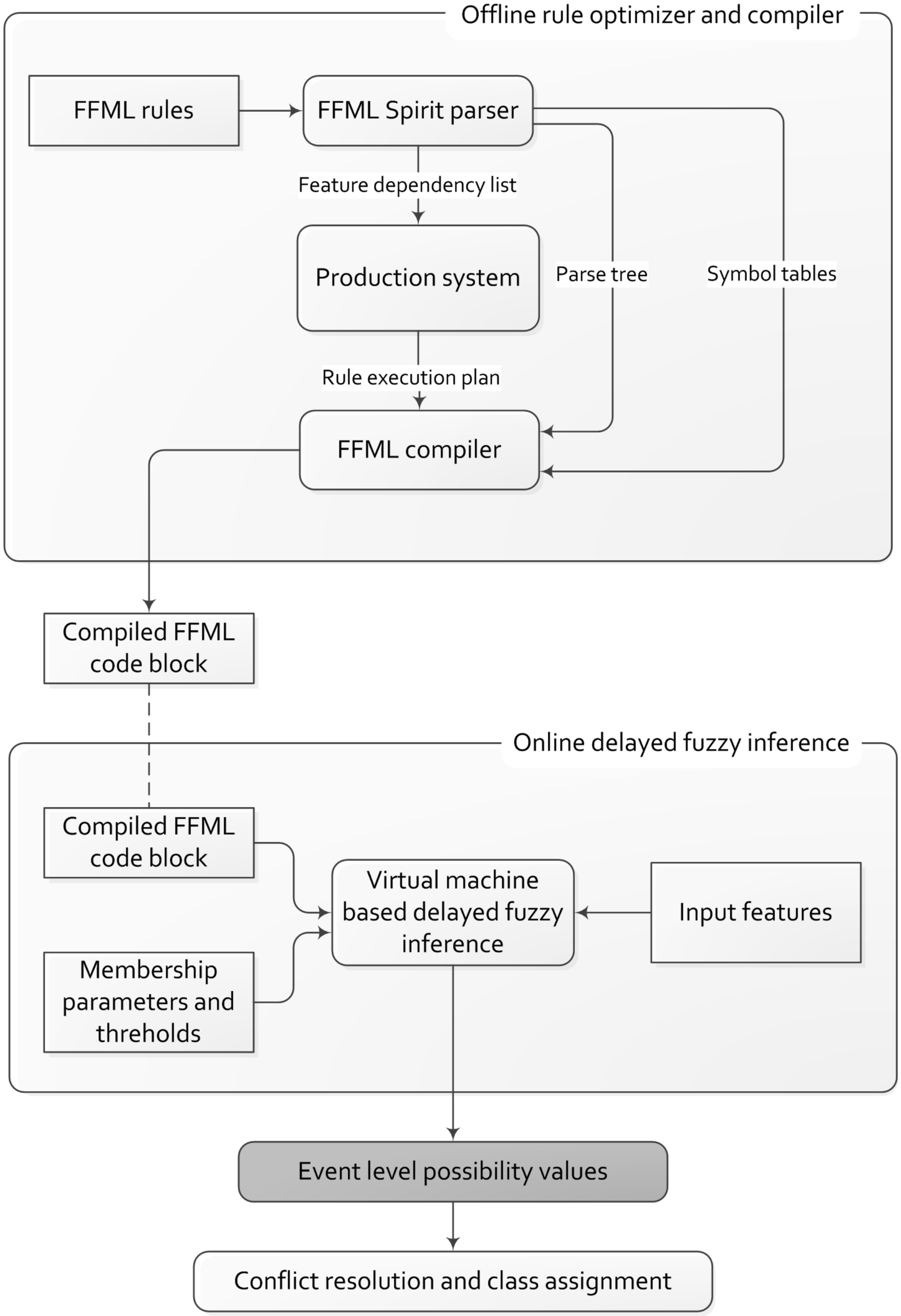

Offline rule optimizer and compiler
FFML rules
FFML Spirit parser
Feature dependency list
Production system
Parse tree
Symbol tables
Rule execution plan
FFML compiler
Compiled FFML code block
Online delayed fuzzy inference
Compiled FFML code block
Membership parameters and threholds
Virtual machine based delayed fuzzy inference
Input features
Event level possibility values
Conflict resolution and class assignment

Figure 6.8: Fuzzy Feature Analysis Engine (FFAE) overview

#### *6.4.1.1 FFML Spirit parser*

FFAE is a domain specific expert system shell designed to process features extracted from power system event data and to classify power system events. The domain knowledge needed by FFAE is stored in the form of rules composed using FFML.

FFML Language : FFML is a custom language that was developed to capture expert knowledge required for classifying power system event data. FFML supports some unique features that greatly simplify constructing rules for the power system domain and computing with possibility values:

1. FFML supports computing with possibility values. I.e, the consequent portion of a rule can be a variable that simply holds the possibility value computed using the antecedent portion of the rule. When the rule is executed, a variable with the computed possibility value is asserted.

2. FFML supports both fuzzy and crisp consequents. However, FFML allows only a single variable in the consequent portion.

3. FFML supports mixing both fuzzy and crisp variables to construct complex expressions in the antecedent portion of the rules. These expressions support both fuzzy and boolean operators. For example, 'MaxCurrent is *High*' is a fuzzy expression that returns a value between zero and one depending on the degree to which the value of MaxCurrent is *High*. The expression 'bReclosed == true' is a crisp comparison. It returns zero or one depending on whether or not the variable bReclosed is set to true. FFML supports combining these two expressions using a fuzzy disjunction or conjunction operator.

4. FFML supports user defined functions.

5. FFML supports vectorized rules. When one or more features used by the antecedent portion of the rule is three phase in nature, all operations are vectorized. The rules are automatically evaluated for each phase, without the explicitly enumerating rules for each phase. For example, consider the following phase level FFML rule (Table 6.5, Rule 1):

   (FFML) $VARS_Step_Down: ($Q_Shape, % STEP_DOWN) ¡@eq¿;

   (Text)VARS_Step_Down=(Q_Shape is STEP_DOWN)

   Q_Shape is a three phase feature vector that represents the shape observed in reactive power signals corresponding to each phase. When FFAE evaluates this rule, Q_Shape value corresponding to each phase A, B and C is compared to the value STEP_DOWN. The results of the comparisons are then stored as a vector possibility values into the output variable VARS_Step_Down. This is equivalent to the following:

   VARS_Step_Down[Phase_A]=(Q_Shape[Phase_A] is STEP_DOWN)
   VARS_Step_Down[Phase_B]=(Q_Shape[Phase_B] is STEP_DOWN)
   VARS_Step_Down[Phase_B]=(Q_Shape[Phase_C] is STEP_DOWN)

   Thus, a single FFML rule can be used instead of enumerating the same rule for each phase. This greatly simplifies the rule base. It also reduces the possibility of errors introduced due to replicating the same rule for each phase. Vectorized rules also help improve computational efficiency. FFML allows the use of more than one three phase features in the antecedent portion of the rules. Phase independent features may also be combined with three phase features.

6. FFML supports a special operator, fuzzy combination operator. This operator allows enumerating possibility values for different phase combinations. For example, consider the following rule that computes the possibility of an unbalanced capacitor operation:

   (FFML) $Capacitor_Two_Phase_Unbalanced_On:

   ($Cap_On_Q_Behavior, $Cap_On_Q_Behavior,

   $No_P_Q_Behavior)

   `<@FuzzyComb, ’disjunc’, ’conj’>`

   (Text) Capacitor_Two_Phase_Unbalanced_On =

   Capacitor on Q behavior on two phases AND

   No P or Q change on third phase

   The above rule can be expanded further to explicitly enumerate all phase combinations as follows:

   (Text) Capacitor_Two_Phase_Unbalanced_On =

   (Capacitor on Q behavior on phase A AND

   Capacitor on Q behavior on phase B AND

   No P or Q change on third phase C)

   OR

   (No P or Q change on third phase A AND

   Capacitor on Q behavior on phase B AND

   Capacitor on Q behavior on phase C)

   OR

   (Capacitor on Q behavior on phase A AND

No P or Q change on third phase B AND

Capacitor on Q behavior on phase C)

The fuzzy combination operator ‘`<@FuzzyComb, ’disjunc’, ’conj’>`’ used in the example FFML rule, automatically enumerates the combinations across phases and computes a single possibility. The combination operator accepts three arguments for which combinations are to be generated. The combination operator also accepts two operators as arguments. The first operator is applied across different combinations and the second operator is used within each combination. For example, the fuzzy combination operator of the general form (Arg1, Arg2, Arg3)`<@FuzzyComb, op1, op2>` is evaluated as follows:

(*Arg1*, *Arg2*, *Arg3*)`<@FuzzyComb`, **Op1**, **Op2**> =

(*Arg1*[Phase_A] **Op2** *Arg2*[Phase_B] **Op2** *Arg3*[Phase_C] )

**Op1**

(*Arg1*[Phase_A] **Op2** *Arg2*[Phase_C] **Op2** *Arg3*[Phase_B] )

**Op1**

(*Arg1*[Phase_B] **Op2** *Arg2*[Phase_A] **Op2** *Arg3*[Phase_C] )

**Op1**

(*Arg1*[Phase_B] **Op2** *Arg2*[Phase_C] **Op2** *Arg3*[Phase_A] )

**Op1**

(*Arg1*[Phase_C] **Op2** *Arg2*[Phase_B] **Op2** *Arg3*[Phase_A] )

**Op1**

(*Arg1*[Phase_C] **Op2** *Arg2*[Phase_A] **Op2** *Arg3*[Phase_B] )

> Thus, FFML provides a mechanism to compactly represent combinations of three phase features and possibility values using fuzzy combination operator.

Before FFML rules can be used by the inference engine (FFAE), the rules need to be parsed, and represented in memory, in a form that is usable by FFAE. Spirit [112] library based parser was designed for this purpose.

Spirit parser : Spirit library was used to define the formal grammar for FFML in EBNF (Extended Backus-Naur Form). Spirit library was also used to generate a parser that reads the rules and create a parse tree. The parse tree is an in memory representation of the rules in the form of a directed graph. A number of operations on the rules (including inference) can then be performed by traversing the parse tree. The Spirit library based parser together with the functions that operate on the parse tree will here on be referred to as 'Spirit parser'.

Figure 6.9 shows the detailed view of the Spirit parser. FFML rule base is only input to the FFML parser. The modularity and hierarchy in the fuzzy hierarchical classifier is enforced through how rules are arranged in FFML rule base and their dependency. The FFML base has multiple modules, each handling a subset of event categories. Within each module, rules are arranged in three groups. Phase level rules, segment level rules and event level rules. Phase level rules only use input features as inputs. Segment level rules are dependent the output of phase level rules and phase independent input features. Event level rules are dependent on outputs from segment level rules and segment independent input features. This dependency was explained in detail in Section 6.2. It is possible to process rules in each module independently. However, doing so will not be able to take advantage of any inherent similarities in rules across modules. I.e, cer-

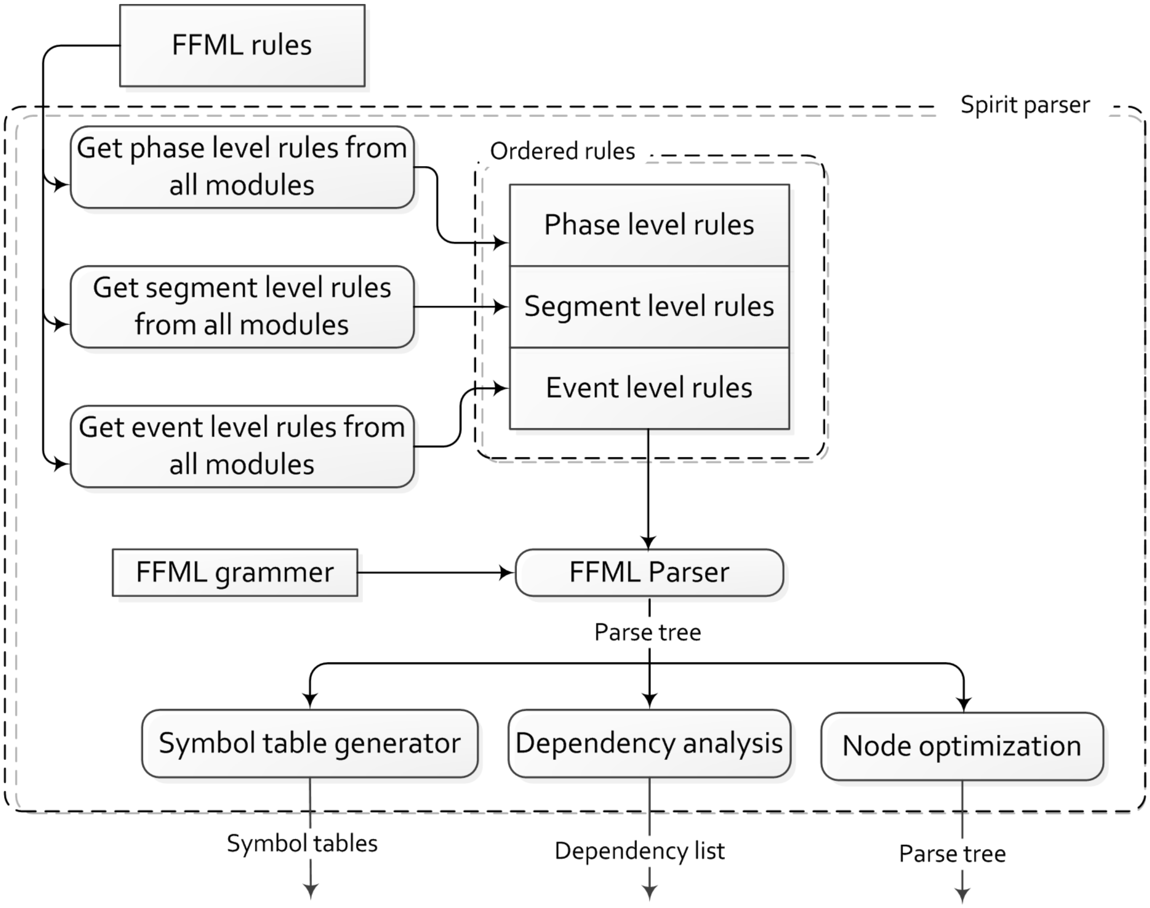


Figure 6.9: Detailed view of Spirit parser

tain conditions in antecedent portion of rules could appear in multiple modules. If these repetitive 'patterns' can be identified, then they can be evaluated once and reused across modules. This would help improve computational efficiency when rues are evaluated. For this reason, the rule base is preprocessed before it is parsed. All phase level rules, segment level rules and event level rules are merged across modules. The rule hierarchy is preserved by adding phase level rules from all modules first, followed by segment level rules and event level rules. This is shown as the block labeled 'Ordered rules' in Figure 6.9. It should be noted that ordering rules according to the hierarchy is not a requirement. The offline rule optimizer and compiler is capable of automatically rearranging rules based on their dependency. The reason the ordering was done is because of two reasons:

1. Dependency between the three levels in the hierarchy is explicit in the rule base, and ordering the rules would help reduce processing time required by the offline rule optimizer and compiler. This is because, the dependency need not be inferred again.

2. Grouping phase level rules, segment level rules and event level rules allow these three levels of rules to be independently optimized for recurring patterns. This would help reduce the memory and computational complexity of the rule optimizer. For example, antecedent portions of phase level rules from different modules or within a module are more likely to show similarities. It is unlikely that phase level rules have antecedent portions that are similar to event level or segment level rules.

Ordered rules are then parsed, and a parse tree is generated. The parse tree is an in memory representation of the rule base in the form of a graph. Figure 6.10 shows example rules 1 (R1), 2 (R2) and 3 (R3) from Table 6.5 as directed graphs

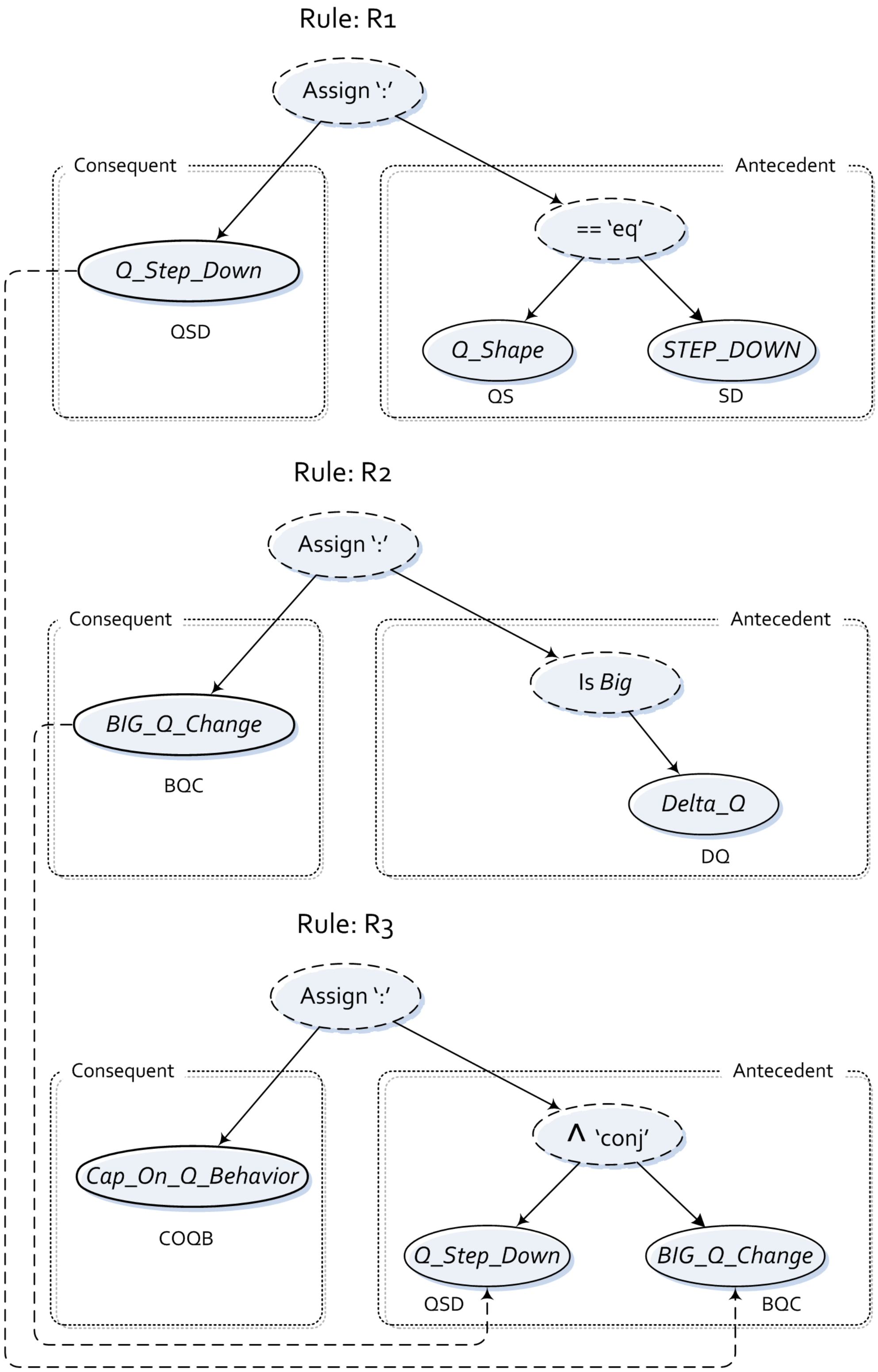


Figure 6.10: Example rules represented as graphs

generated by Spirit parser. Nodes in the graph fall into three broad categories; operators, output variables and inputs. Operators are shown inside dotted ellipses in Figure 6.10. Operators include Boolean operators, fuzzy operators, comparison operators, functions, assignment operator and fuzzy membership functions. For example, the upper most node in R1, R2 and R3 is the assignment operator ':'. The portion of the graph to the right side of the assignment operator is the antecedent, and the portion of the graph to the left side is the consequent. The graphs imply that the quantity computed on the antecedent portion of the graph is assigned to the output variables (*Q_Step_Down*, *Big_Q_Change* and *Cap_On_Q_Behavior*) on the consequent portion. The single leaf node (Terminal node) on the left side of the assignment operator (consequent portion) is the output variable. The graph on the right side of the assignment operator (antecedent portion) can however, have multiple leaf nodes. Each of leaf on the antecedent portion of the graph represents an input. Inputs can be of three types:

1. Input features: Input features are features extracted from processing event data. In Figure 6.10, *Delta_Q* and *Q_Shape* represent input features.

2. Input constants: Input constants are values such as thresholds and other parameters that are independent of event data. In Figure 6.10, *STEP_DOWN* represents an input constant which is a numerical representation (enum) for RMS shape.

3. Computed values: These are values that were computed as a result of firing other rules. For example, Rule R3 in Figure 6.10 uses computed values *Q_Step_Down* and *Big_Q_Change* as inputs. *Q_Step_Down* and *Big_Q_Change* are output variables for rules R1 and R2 respectively. This dependency relationship is shown throw dotted arrows.

In the case of a rule with a fuzzy consequent, the consequent portion cannot be fully evaluated until the antecedent portion of all rules with the same variable on the consequent portion are evaluated. A simple assignment operator will not be sufficient to represent these rules. Instead, temporary variables are created to hold the possibility values that are calculated from the antecedent portion of rules with fuzzy consequent. A separate defuzzification node is then created that is dependent on the antecedent portion of all rules with the same fuzzy consequent variable. This is easily illustrated with an example. Consider the event level rules 1 and 2 from Table 6.10. Figure 6.11 shows the resultant graphs after being processed by the parser. As a result of parsing rules 1 and 2, three assignment nodes FR1, FR2 and DFR1 are generated. The right side of the nodes FR1 and FR2 represent the antecedent portion of rules 1 and 2 respectively. However, the fuzzy consequents of these rules do not appear on the consequent portion (left side) of the nodes FR1 and FR2. Instead, temporary variables are generated to hold the possibility values generated by the antecedent portion of the rules 1 and 2. A separate defuzzification node DF1 is generated to compute the value of the output variable *3Ph_Cap_On_Mon_Fdr_Normal.* The left side of the defuzzification node represents the defuzzification operation needed to compute the output variable *3Ph_Cap_On_Mon_Fdr_Normal.* The leaf nodes on the antecedent portion of the node DF2 are the computed temporary variables *3Ph_Cap_On_Mon_Fdr_Normal_1 and 3Ph_Cap_On_Mon_Fdr_Normal_2.* These hold the possibility values from evaluating antecedent portion of rules 1 and 2 respectively. These possibility values are used to clip the consequent membership functions *confMedium* and *confHigh* of the rules 1 and 2 respectively. These clipping functions appear as the immediate parent nodes. The outputs of the clipped membership functions are then superimposed and defuzzified using Center of Area (COA) defuzzification. Hence 'COA

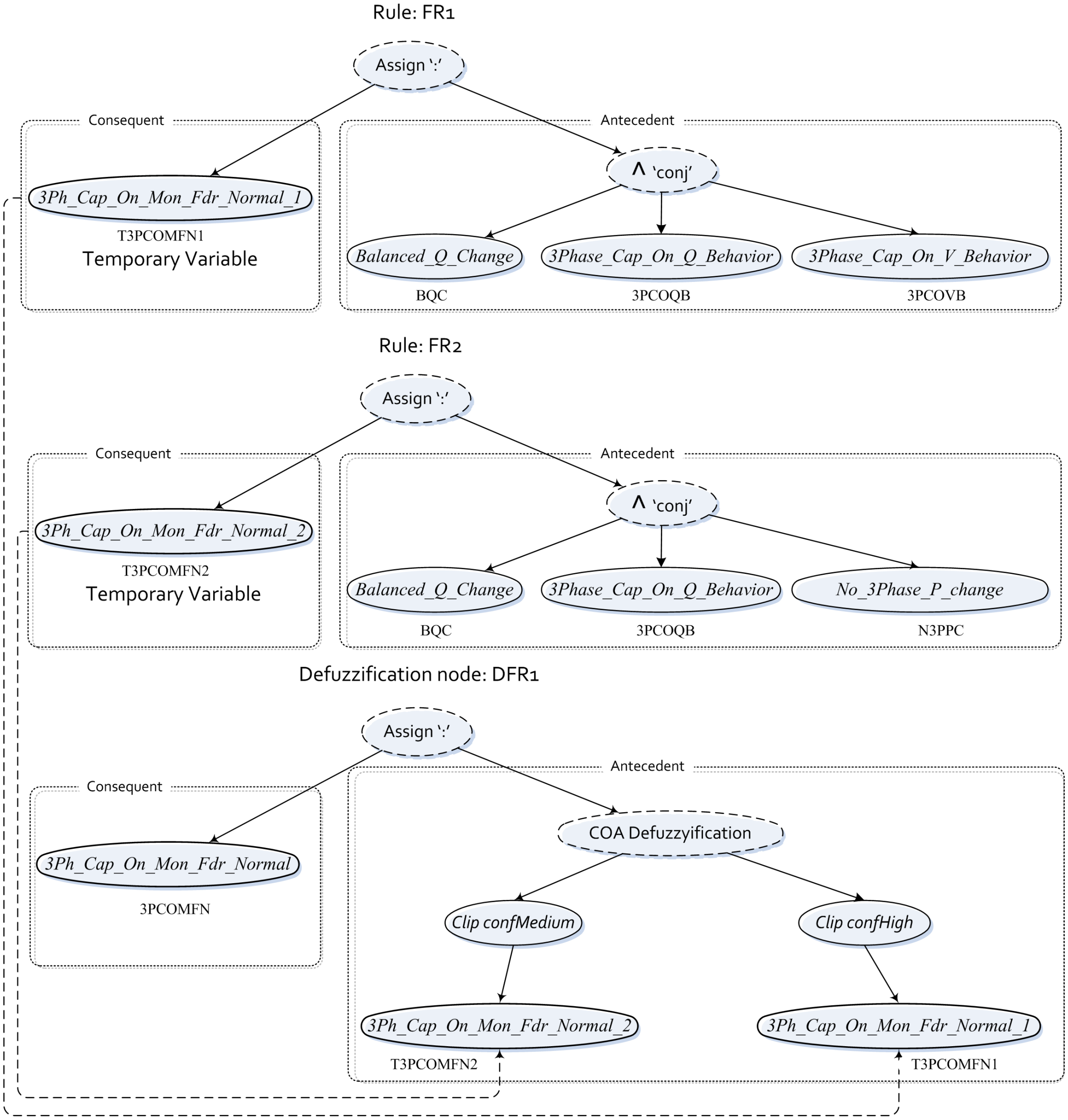


Figure 6.11: Example rules with fuzzy consequent represented as graphs

Defuzzification′ node appears as the immediate parent of clipped membership functions. Finally, the output of the COA defuzzification is assigned to the output variable *3Ph_Cap_On_Mon_Fdr_Normal* that appears on the right side of the DF1 node. The defuzzification node DFR1 serves a dual purpose:

1. It acts as an evidence aggregation node that accumulates possibility values from antecedent portion of all rules that have the same output variable on their consequent portion.

2. It acts as an extra rule that introduces a dependency between a fuzzy output variable and all rules that have a consequent portion pointing to the same variable. This dependency makes sure that a fuzzy output variable is not computed until the inputs for all the dependent rules are available.

The parse tree is then traversed to generate other data structures that aid in rule inference. These operations are shown in Figure 6.9:

1. Symbol table generation: Symbol tables represent a mapping between variable names and the actual memory location in which the values for these variables are stored. When rules are compiled, input and output variables (including constants) are represented as memory offsets instead of using string literals. This helps to improve computational efficiency.

2. Dependency analysis: This operation establishes the dependency between the leaf nodes and a rule. A rule cannot be evaluated until all its leaf nodes have been assigned a value. Dependency analysis creates a dependency list. A dependency list consists of two components. A dependency table and a dependency count tracker. The dependency table is a data structure that subscribes a rule to all its input variables (identified through leaf nodes). It

establishes a one-to-many relationship between an input and all the rules that use that input. When a value is assigned to an input variable, it notifies all the dependent rules that subscribe to it. When a rule is informed by one of its input nodes, the dependency count for that rule is decreased by one in the dependency count tracker. A rule becomes active when its dependency count becomes zero. This is analogous to the pattern matching and join network used in RETE networks [111]. However, this implementation is much simpler.

3. Node optimization: This operation uses string comparison to detect repeating patterns in rules within a level (phase, segment or event). Then, nodes corresponding to repeating patterns are marked so that they are computed only once. Patterns are similarities observed in antecedent portion of rules in the way input variables are combined to form expressions. For example, consider two rules of the form ‘Rule1: If A1 AND ***(A2 OR A3)*** THEN C1’ and ‘Rule2: If A4 AND ***(A2 OR A3)*** THEN C2’. Clearly, the pattern ‘***(A2 OR A3)***’ on the antecedent portion is common to both rules Rule1 and Rule2. This is referred to as the repeating pattern. The pattern ‘***(A2 OR A3)***’ can be computed once and then used in all locations (nodes) where this pattern appears. This would help improve efficiency when evaluating rules. During the parse tree creation, text identifiers are generated for each node. For leaf nodes that are not dependent on other output nodes, these are automatically generated by the parser. Figure 6.10 shows example text identifiers under each input node (QSD, QS, etc.). The text identifiers shown in Figure 6.10 are provided for illustrative purposes only, and do not correspond to actual text labels that are generated by the parser. For higher level nodes (nodes

with one or more child nodes), text identifiers are computed as a combination of the text identifiers of the child nodes and the operation done by that node. For example, in the Figure 6.10, the identifier '(QS_eq_SD)' is generated for the node representing the equality comparison (== 'eq') in graph R1. If the operator is commutative, then the child nodes appear in alphabetical order within the text identifier. This means, if the nodes QS and SD are swapped, since equality comparison is commutative, the same identifier '(QS_eq_SD)' will be generated. If the operator is not commutative, then the child nodes appear in the order in which they are connected (from left to right). During the node optimization stage, rules are traversed starting at the assignment operator in a breath first fashion. Text identifiers for each node that is traversed are added to a sorted list. Before the text identifier is added, the sorted list is first searched for the existence of that text identifier. If the text identifier exists, then the node is marked as being a repetitive pattern in the parse tree. FFML compiler uses this modified parse tree to generate optimized code. The optimized code block is then used by the online component of FFAE for inference during run-time. However, before the FFML compiler can compile the rules, the rules need to be ordered based on their dependency. This is done by the production system.

#### *6.4.1.2 Production System*

The offline rule optimizer and compiler uses a production system to order rules based on their dependency. This production system shares many of the features of a production rule system used in traditional expert systems[113]. A production system provides an automated mechanism to execute rules based on matching antecedent portion of rules with available inputs (facts). Facts are analogous to input features in the case of an expert system based classifier. When rules are executed, consequent portion of the rules may result in actions that create new inputs (asserting new facts). These new facts in turn will cause other rules to execute. This process of using rules to infer new facts from existing facts is continued till no new facts are discovered. Figure 6.12 shows the essential components of a production system. A production system consists of a representation of parsed rules in memory, a representation of data (facts) in memory and an inference engine that consumes facts and uses rules to infer new facts. The inference engine of a production system works by repeating the following sequence of steps:

1. When new facts are introduced into memory, a pattern matching mechanism is used to match the facts needed by antecedent portion of rules. Brute force pattern matching of rule antecedents to available facts can be a very expensive process. It is not practical for use with large scale expert systems. RETE networks [111] are commonly used as an efficient implementation of the pattern matching mechanism. When all the facts required by the antecedent portion of a rule are matched, the rule is placed on an agenda. This is because, all the inputs required by the rule are available. Hence, the rule can be executed. Multiple such rules can be placed on the agenda.

2. When more than one rule is placed on the agenda, the production system needs to decide which rule to execute first. For this reason, the agenda is also called a conflict set. This is because, a conflict resolution strategy may be used to choose which rule needs to be fired first. The choice of the conflict resolution strategy may be dependent on the application. In the second step, a conflict resolution strategy used to select a single rule from the agenda for execution.

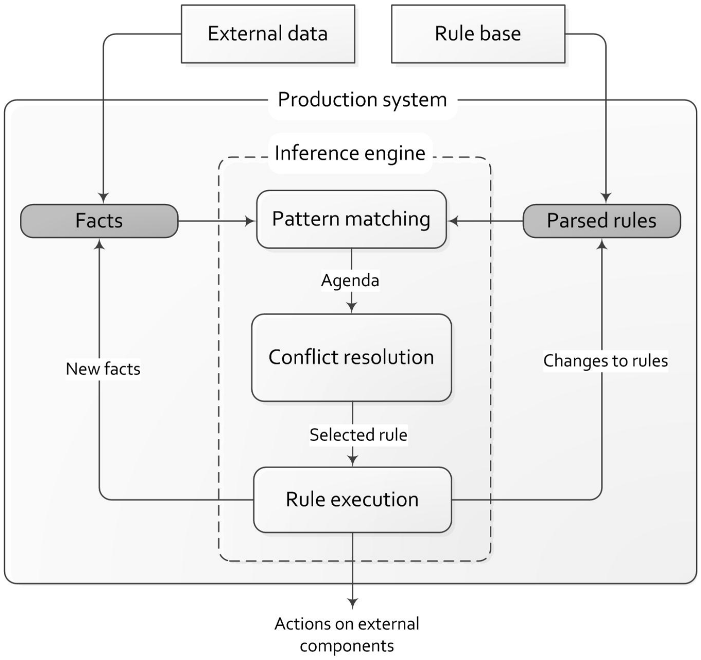


Figure 6.12: Schematic of a typical production systems

3. In the final step, inference engine executes the selected rule and performs the action specified by the consequent portion of the rule. As a result of the action, new facts may be introduced into the memory of the product. Actions specified by the consequent portion of the rule may also affect components that are external to the production system.

The production system used by the offline component of FFAE is much simpler when compared to a traditional production system. This because of for two reasons. The production system does not do any inference, its only function is to order the rules based on their dependency. The second reason being, no conflict resolution is needed, and all rules in the agenda are selected for execution. Figure 6.13 shows the schematic of the production system used by FFAE. Since the production system operates in an offline fashion, all input features that are derived from event data and all input parameters such as thresholds are assumed to be

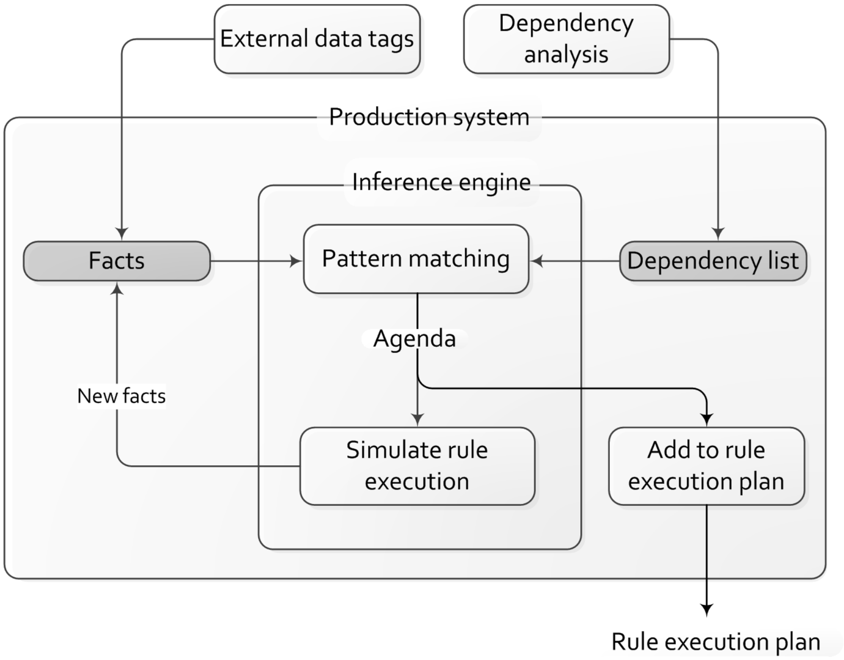


Figure 6.13: Production system used by FFAE

available. During the initialization, the identifiers corresponding to these input features and parameters are introduced into memory as facts. works by repeating the following sequence of steps:

1. When new facts are introduced into memory, these facts use the dependency list to inform all the rules that subscribe to the fact (input variable). The dependency counter for the dependent rules are then decremented. All rules with a dependency count of zero are then placed on the agenda.

2. In the second step, rule execution is simulated for all rules placed on the agenda. This is done by simply introducing the text identifier for the output variable of all rules in the agenda into the facts memory. All rules on the agenda are also appended to the rule execution plan. The rule execution plan is n list that contains all the rules ordered according to their dependency.

The rule execution plan, symbol tables and the parse tree are input to the FFML compiler.

#### 6.4.1.3 FFML Compiler

FFML compiler generates optimized code that represents rule inference as a sequence of operations on memory locations. The code generated by the FFML compiler is referred to as FFML code block. The FFML code block has two sections. A header containing symbol tables and the executable section that contain a sequence of instructions to be executed by the virtual machine. FFML compiler embeds the symbol tables generated by the Spirit parser as the header. The compiler then traverses the parse tree corresponding to each rule in the order in which they appear in the execution list. Based on the type of node, instructions are generated as a sequence of operations on memory locations. These memory locations hold values computed for the child nodes of a given node. This code is then appended to the execution section of FFML code block. Based on input features, and using the code generated by the FFML compiler, event possibility values are computed by the online delayed inference component of FFAE. The online delayed inference uses a virtual machine to execute the FFML compiler generated code.

### 6.4.2 Online Delayed Fuzzy Inference

Online component of FFAE is responsible for processing input features extracted from event data and computing output event possibility values. Computed event possibility values are then used to assign class labels for the event data being processed. The online component of FFAE is called 'online delayed fuzzy inference'. This is because, the production system, which is traditionally responsible for inference, is part of the offline component of FFAE. The production system simulates the firing of rules without actually inferring. Inference is

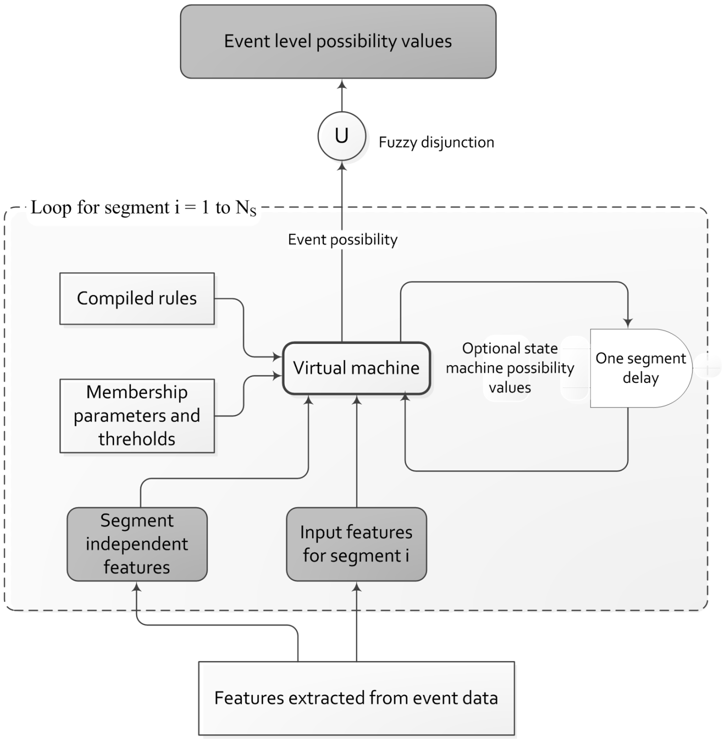

Event level possibility values
U
Fuzzy disjunction
Loop for segment i = 1 to N_S
Event possibility
Compiled rules
Virtual machine
Optional state machine possibility values
One segment delay
Membership parameters and threholds
Segment independent features
Input features for segment i
Features extracted from event data

Figure 6.14: Schematic of online delayed fuzzy inference

delayed, and is done in the online component of FFAE using a virtual machine.

Figure 6.14 shows the detailed schematic of online delayed fuzzy inference engine. Central to the online fuzzy inference engine is the virtual machine. The virtual machine is initialized with compiled FFML rules that were produced by the rule optimizer and compiler and input. Compiled FFML rules are analogous to executable code, and the virtual machine is analogous to a processor that executes instructions present in compiled FFML rules. The virtual machine uses symbol tables to load input parameters in to working memory when the inference engine is initialized. These input parameters include thresholds and fuzzy membership parameters. Every time the inference engine is requested to process event data, it does the following:

1. The virtual machine uses the symbol tables to request segment independent

features. It then loads the features into working memory. This is done only once for a given event.

2. For each segment $i$, $i = 1, \ldots, N_s$, where $N_s > 0$ is the in the number of segments detected in event data:

   (a) The virtual machine requests for input features corresponding to segment $i$.

   (b) If the rule base supports operation in a state machine mode, the virtual machine also requests for a subset of possibility values computed for the previous segment $i - 1$ when $i > 1$.

   (c) The virtual machine executes the compiled FFML rules using the input features. It computes event level possibility values using these features. This is equivalent to evaluating phase level, segment level and event level rules and inferring event level possibility values.

   (d) Computed event level possibility values are stored for further processing, and are not output.

3. For each segment, event level possibility values are combined using the fuzzy disjunction operator *Max.* Then, a single event level possibility value is output for each event category. The fuzzy disjunction operator is equivalent to using an 'OR' to combine the possibility values generated for each segment. For example, for a hypothetical class '*Class A*', the fuzzy disjunction operator is equivalent to using the following rule to combine possibility values:

"If s*egment 1 shows evidence of the cause being Class A* OR s*egment 2 shows evidence of the cause being Class A* OR ...*segment N shows evidence of the cause* *being Class A* Then *cause is Class A"*

Event level possibility values computed by the online fuzzy inference engine are then used as inputs to the conflict resolution and class label assignment module. This was as described in Section 6.2. Class label assignment and conflict resolution module assigns classification attributes such as event type, event phase, ground information and position information. Class labels are generated for most of the events processed by FFAE. However, a majority of these events are normal system events. It would be both difficult and error prone for a human operator to mine through the classification information and find events of interest that need action. A reporting framework is needed for processing the raw classification information, and presenting events of interest in a user friendly manner

## 6.5 Intelligent Reporting Framework

Depending on the sensitivity of thresholds used to trigger capture of event data, a DFA monitoring unit could record a few hundred event data files per day. This would correspond to few tens to few hundred classifications generated by the fuzzy hierarchical classifier for each feeder that is being monitored. Even for a small sized utility, where a few ten feeders may require monitoring, the classification information can quickly become unmanageable for utility personnel to handle . Hence an intelligent reporting framework was developed. The intelligent reporting framework processes the raw classification information generated by the fuzzy classifier. It then presents the information as actionable real-time alerts and reports to utility personal. Only information that will be perceived as being important is presented to the utility user. Since only a subset of abnormal events is presented to the user, the number of events that require user's attention is greatly reduced. The reporting framework is composed of two layers:

1. Reporting algorithms: Reporting algorithms process classification information generated by the fuzzy hierarchical classifier in a soft real-time fashion. Reporting algorithms use the classification information and output features as inputs to look for a subset of reportable events. Reportable events fall into three categories:

   (a) Single occurrence abnormal events: These correspond to a single occurrence of an abnormal event that a utility user may perceive to be important and hence needs to be reported. An example of a single occurrence abnormal event is a substation automatic circuit breaker lockout. A breaker lockout would take the whole feeder out of service, and hence utility personnel need to be notified of the condition as soon as possible. These events can be marked for reporting based on the classification information and output features from that event.

   (b) Recurrent abnormal events: Sometimes a single occurrence of an abnormal event may not be of much interest to utility users. However, repetitive occurrence of the same event within a relatively short duration may be perceived as being important. Some examples of recurrent abnormal events are recurrent faults that have similar magnitudes and durations, failure signatures of clamps or line switches, failure signatures of underground cable. Detecting recurrent abnormal events requires clustering algorithms. Clustering algorithms were designed to process classification information and output parameters from multiple events.

   (c) Recurrent normal events: Normal system events are the least important events and are mostly ignored. However, repetitive occurrence of a large number of normal system event within a short duration may indicate a system abnormality. One such example is excessive capacitor switching operations. It is normal for capacitors to automatically switch on and off, a few times a day. However, due to a faulty controller or a wrongly configured capacitor switch controller, there have been documented instances [102] where a capacitor bank may switch tens or hundreds of times a day. Such operations will not only quickly wear out switch contacts but also affect other devices on the system. Clustering algorithms were designed to processes classification information from multiple events, and to detect these abnormalities.

2. Presentation layer: Information generated by reporting algorithms are written into a database in the form of reportable events, cluster related information and reportable parameters or features. This information is not readily usable by utility user. The presentation layer is responsible for delivering the relevant information to utility users in a user friendly and timely fashion. It also allows some level of customization so that users can decide how what events are presented to them. Currently two forms of presentation are used. A web based interface that presents reportable items and an email based interface that emails users reportable events. Both the web based interface and email interface provide two types of reportable items:

   (a) Real time alerts: These correspond to event information that is presented to the users as soon as the corresponding event data is processed. Real time alerts are generated for events that may require users' immediate attention.

   (b) Periodic reports: Periodic reports contain a report of all occurrences of a subset of event types during a period. A single report is generated for all feeders within a utility. Periodic reports are generated on a regular bases (once a day, once a week, etc.) based on user preference. Periodic reports server the dual purpose. Periodic reports inform utility personnel about the general health of all the feeders. Periodic reports also help draw the attention of utility personal to less urgent conditions such as unbalanced capacitor operations.

Example results produced by the reporting algorithms will be explored further in Chapter 7.

## 6.6 Chapter Summary

In this chapter, the challenges and factors that influenced the design of the fuzzy power distribution system event classifier were outlined. The design of the fuzzy hierarchical classifier was then described in detail. Examples were provided to show how input features were processed at the phase level, the segment level end the event level. The reasoning behind using a custom fuzzy inference engine as a part of the fuzzy hierarchical was explained. FFML, a custom language for representing expert knowledge, the offline and online component of the fuzzy inference engine that uses FFML rules were described in detail. Finally, an intelligent reporting framework that presents the classification information generated by the fuzzy classifier was introduced.

# 7. RESULTS AND CASE STUDIES

## 7.1 Introduction

This chapter presents a fuzzy logic based Intelligent Power System Event Reporting System (IPSERS) was implemented based on the methodologies proposed in previous chapters. It then evaluates the effectiveness of IPSERS in accurately classifying and reporting power system events. This chapter also presents case studies of utility companies benefiting from IPSERS. IPSERS was developed as a part of Distribution Fault Anticipation (DFA) project. IPSERS is currently online and reporting events in a soft real-time fashion from nearly 100 feeders belonging to 12 participating utilities. Throughout the rest of this chapter, the terms IPSERS and DFA-IPSERS will be used interchangeably.

## 7.2 Overview of IPSERS

IPSERS is a new, on-line, non-intrusive, classification system for identifying and reporting normal and abnormal power system events occurring on a distribution feeder based on their underlying cause. IPSERS uses signals acquired at the distribution substation. The following key steps were involved in the design and implementation of IPSERS:

- A fuzzy logic based classification algorithm (FLCA) was designed and implemented. FLCA uses a fuzzy hierarchical classifier that is both modular and scalable.

- Feature extractors capable of extracting features corresponding to power system components that are either in the process of failing or have already failed were developed. FLCA uses these feature extractors to analyze current and voltage waveforms.

- A fuzzy dynamic time warping (FDTW) technique was developed and implemented for extracting shape based features from RMS waveforms. FLCA uses FDTW and shape based features for RMS waveform analysis.

- Feature extractors for estimating appropriate parameters corresponding to events of interest were developed. These parameters are then reported to users.

- Reporting algorithms and a presentation framework were developed for identifying and reporting events of interest to utility users in a timely and user friendly fashion.

- FLCA and reporting algorithms were optimized for efficient use of memory and CPU. These two components of IPSERS run on remote DFA monitoring units that have limited processing power.

Figure 7.1 presents the overall schematic of IPSERS. IPSERS is composed of four layers:

1. Remote data acquisition: DFA monitoring units record current and voltage waveforms from distribution feeders using as described in Chapter 3, Section 3. Each monitoring unit then stores the recorded data in databases and as waveform files on flash memory based storage devices. DFA monitoring units run Windows CE operating system on an Intel Celeron based single board computers with 512MB on board RAM.

2. Remote algorithmic processing: Fuzzy Logic Based Classification algorithm (FLCA) process data collected by remote DFA monitoring units. Chapter

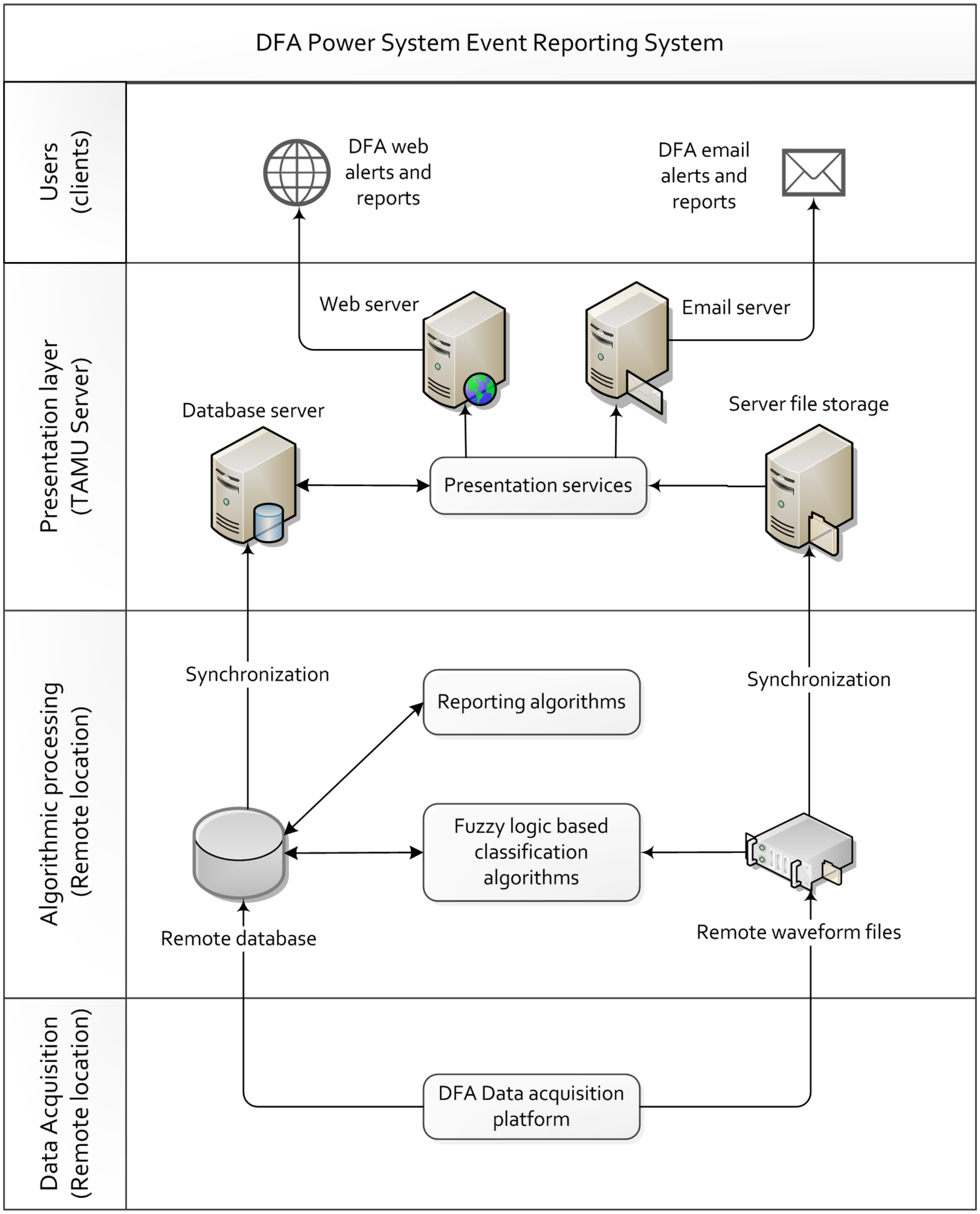

DFA Power System Event Reporting System
Users (clients)
DFA web alerts and reports
DFA email alerts and reports
Presentation layer (TAMU Server)
Web server
Email server
Database server
Server file storage
Presentation services
Algorithmic processing (Remote location)
Synchronization
Synchronization
Reporting algorithms
Fuzzy logic based classification algorithms
Remote database
Remote waveform files
Data Acquisition (Remote location)
DFA Data acquisition platform

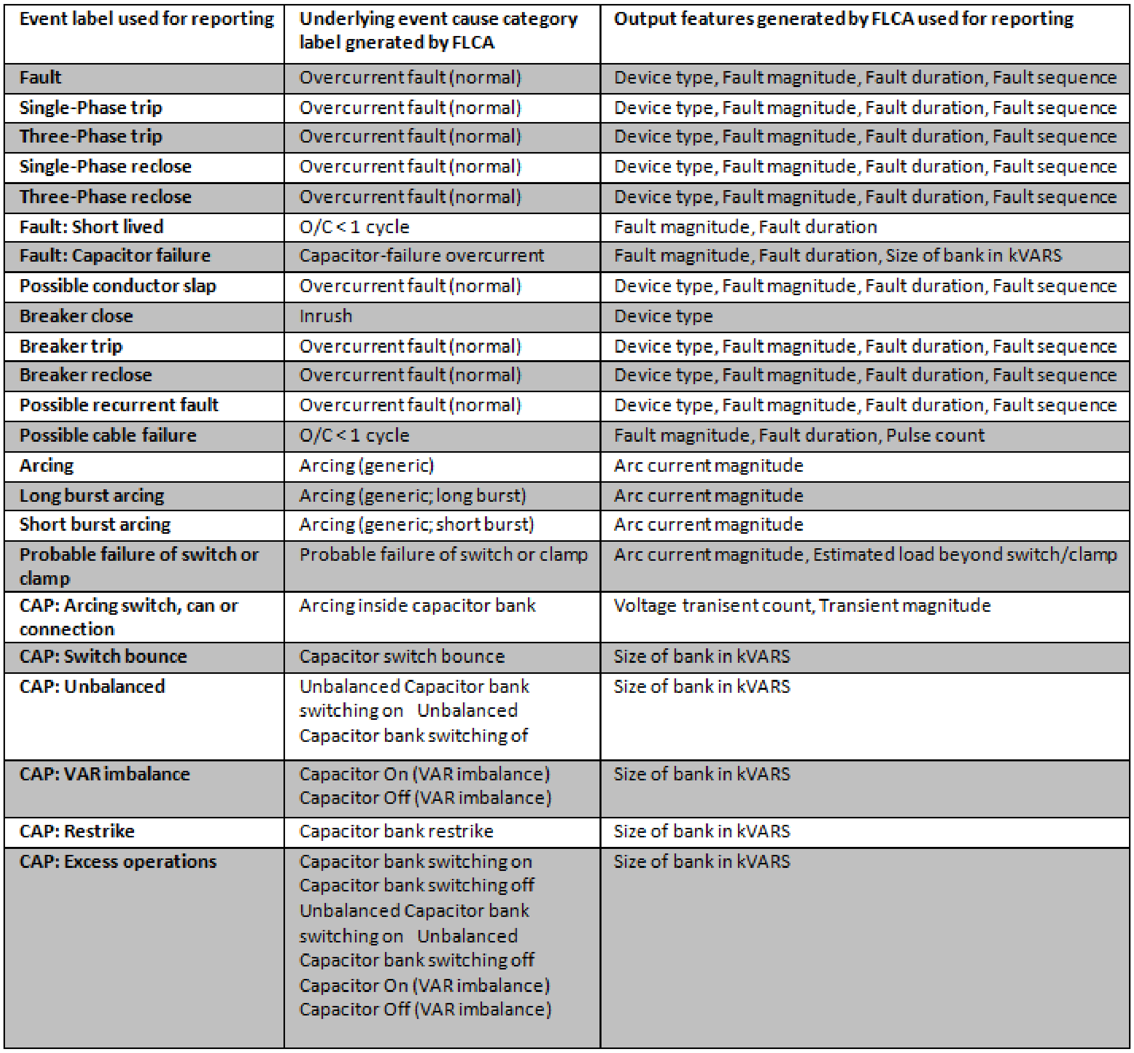
Figure 7.1: Schematic of IPSERS

3, Section 4 provided an overview of FLCA algorithms. Chapters 4, 5 and 6, described the various components of FLCA algorithm in detail. FLCA is responsible for generating class labels and extracting reportable parameters from waveform data. FLCA implementation uses optimized C++ code. FLCA runs on remote monitoring units and writes the results into the remote database. In order to avoid overloading users with too much classification data, reporting algorithms further process the classification information generated by FLCA and find reportable events. Chapter 6, Section 5 discussed reporting algorithms. Reporting algorithms are database intensive algorithms implemented in C++. They run on remote units and write the result of processing into the remote database.

3. Presentation: Central storage located in TAMU synchronizes data with DFA monitoring units using as synchronization software. The synchronization software currently requires broad band internet connection between remote monitoring units and server. Presentation services running on the central server located at TAMU use the information generated by reporting algorithms and create web-baed and email alerts and reports. These alerts and reports present classified power system events in a user friendly manner. Since the server collects data from all feeders belonging to a utility, data could be aggregated at a utility level and then be presented to the appropriate utility personnel.

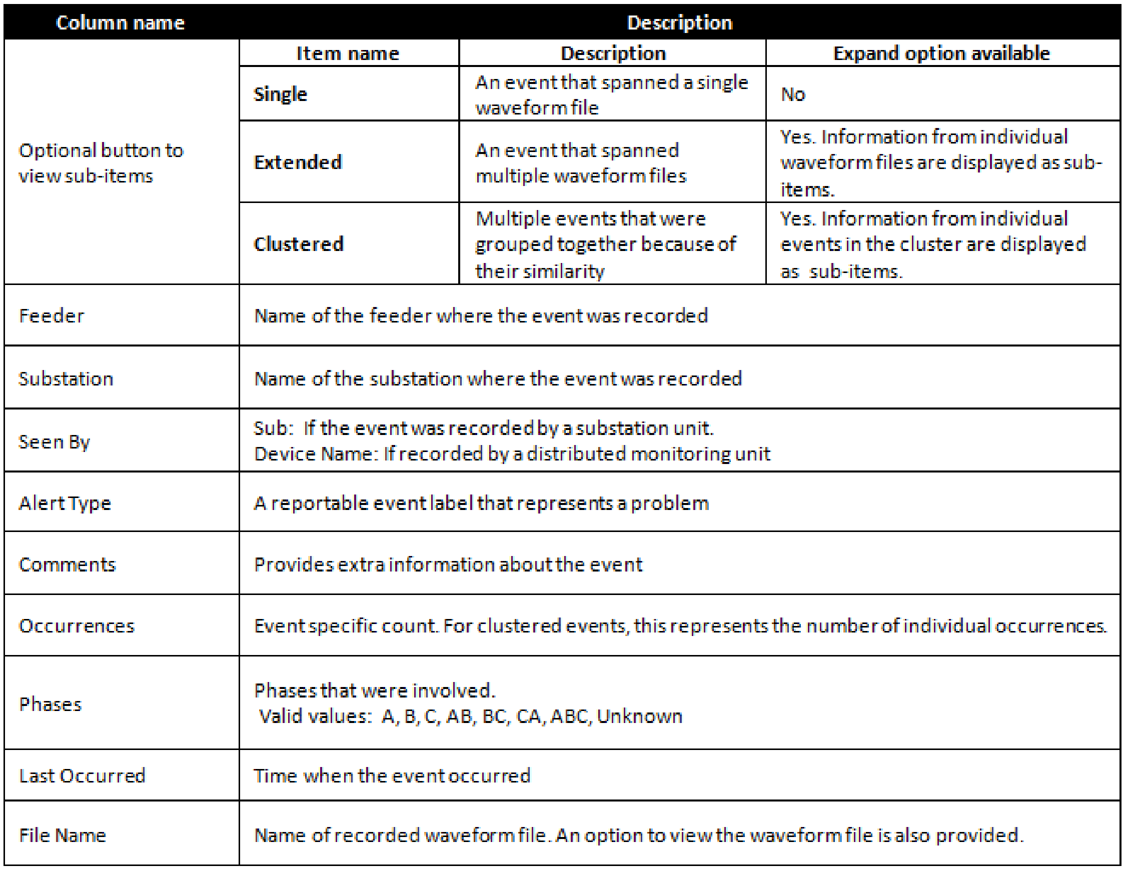
4. Client: Users can log into DFA website and see the web-based alerts and reports corresponding to their utility. Utility users can also subscribe to DFA emails and choose to receive alerts and reports via email. These alerts and reports provide utility users with increased situational awareness. They

Table 7.1: Relationship between event labels used for reporting and information generated by FLCA



| Event label used for reporting | Underlying event cause category label gnerated by FLCA | Output features generated by FLCA used for reporting |
|---|---|---|
| Fault | Overcurrent fault (normal) | Device type, Fault magnitude, Fault duration, Fault sequence |
| Single-Phase trip | Overcurrent fault (normal) | Device type, Fault magnitude, Fault duration, Fault sequence |
| Three-Phase trip | Overcurrent fault (normal) | Device type, Fault magnitude, Fault duration, Fault sequence |
| Single-Phase reclose | Overcurrent fault (normal) | Device type, Fault magnitude, Fault duration, Fault sequence |
| Three-Phase reclose | Overcurrent fault (normal) | Device type, Fault magnitude, Fault duration, Fault sequence |
| Fault: Short lived | O/C < 1 cycle | Fault magnitude, Fault duration |
| Fault: Capacitor failure | Capacitor-failure overcurrent | Fault magnitude, Fault duration, Size of bank in kVARS |
| Possible conductor slap | Overcurrent fault (normal) | Device type, Fault magnitude, Fault duration, Fault sequence |
| Breaker close | Inrush | Device type |
| Breaker trip | Overcurrent fault (normal) | Device type, Fault magnitude, Fault duration, Fault sequence |
| Breaker reclose | Overcurrent fault (normal) | Device type, Fault magnitude, Fault duration, Fault sequence |
| Possible recurrent fault | Overcurrent fault (normal) | Device type, Fault magnitude, Fault duration, Fault sequence |
| Possible cable failure | O/C < 1 cycle | Fault magnitude, Fault duration, Pulse count |
| Arcing | Arcing (generic) | Arc current magnitude |
| Long burst arcing | Arcing (generic; long burst) | Arc current magnitude |
| Short burst arcing | Arcing (generic; short burst) | Arc current magnitude |
| Probable failure of switch or clamp | Probable failure of switch or clamp | Arc current magnitude, Estimated load beyond switch/clamp |
| CAP: Arcing switch, can or connection | Arcing inside capacitor bank | Voltage tranisent count, Transient magnitude |
| CAP: Switch bounce | Capacitor switch bounce | Size of bank in kVARS |
| CAP: Unbalanced | Unbalanced Capacitor bank switching on Unbalanced Capacitor bank switching of | Size of bank in kVARS |
| CAP: VAR imbalance | Capacitor On (VAR imbalance)<br>Capacitor Off (VAR imbalance) | Size of bank in kVARS |
| CAP: Restrike | Capacitor bank restrike | Size of bank in kVARS |
| CAP: Excess operations | Capacitor bank switching on<br>Capacitor bank switching off<br>Unbalanced Capacitor bank switching on Unbalanced Capacitor bank switching off<br>Capacitor On (VAR imbalance)<br>Capacitor Off (VAR imbalance) | Size of bank in kVARS |

Table 7.2: Information contained in reported items



| Column name | Description | | |
|---|---|---|---|
| Optional button to view sub-items | Item name | Description | Expand option available |
| | **Single** | An event that spanned a single waveform file | No |
| | **Extended** | An event that spanned multiple waveform files | Yes. Information from individual waveform files are displayed as sub-items. |
| | **Clustered** | Multiple events that were grouped together because of their similarity | Yes. Information from individual events in the cluster are displayed as sub-items. |
| Feeder | Name of the feeder where the event was recorded | | |
| Substation | Name of the substation where the event was recorded | | |
| Seen By | Sub: If the event was recorded by a substation unit.<br>Device Name: If recorded by a distributed monitoring unit | | |
| Alert Type | A reportable event label that represents a problem | | |
| Comments | Provides extra information about the event | | |
| Occurrences | Event specific count. For clustered events, this represents the number of individual occurrences. | | |
| Phases | Phases that were involved.<br>Valid values: A, B, C, AB, BC, CA, ABC, Unknown | | |
| Last Occurred | Time when the event occurred | | |
| File Name | Name of recorded waveform file. An option to view the waveform file is also provided. | | |

also serve as diagnostic tools that help users identify existing problems on their feeders. Since information presented through the web based interface and emails are similar, this chapter shows examples from DFA web interface only.

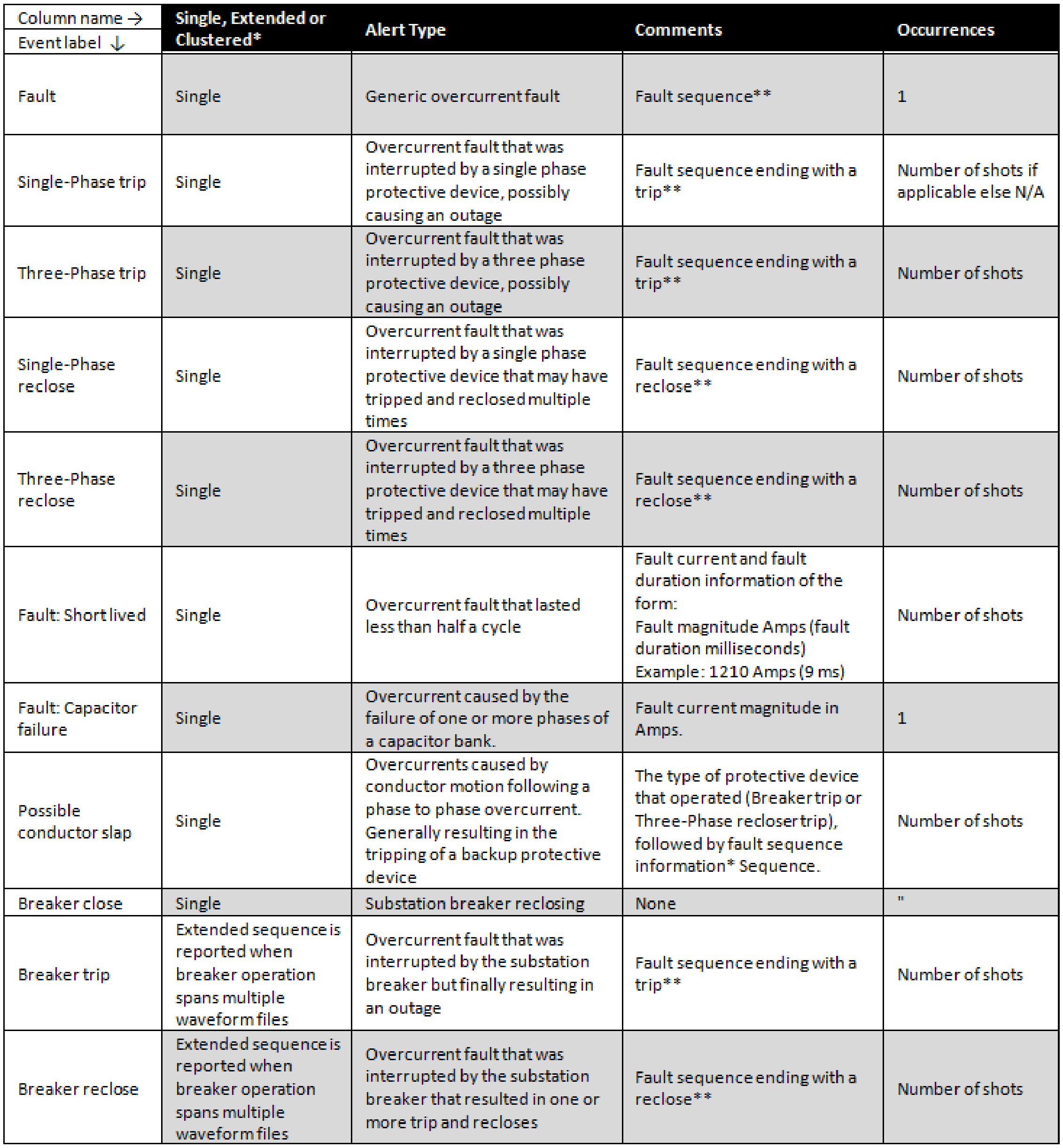
### *7.2.1 Details of Information Presented to Users*

Table 7.1 lists all reportable event categories presented to users. The table also shows the underlying class labels and features (output by FLCA) used by reporting algorithms to generate these reportable event items. Table 7.2 shows typical information presented to the user for each reportable item. Information presented to the user include the location of the device that recorded the event (substation name, feeder name etc.), an event label that indicates possible cause of the event, phases involved, comments that provide information specific to the event, the event time and a link to the waveform file that contains the event data. The web interface provides an expand button to allow users access the individual events within the cluster of events. Most of the information presented to the user depends on the reported event. Tables 7.3, 7.4 and 7.5 provide details of event specific information. In Tables 7.3, 7.4, 7.5 the 'Alert Type' column explains what the event label represents. The next section shows examples of reported events belonging to each of the categories listed in Tables 7.3, 7.4 and 7.5.

### *7.2.2 Real World Examples of Information Presented to Users*

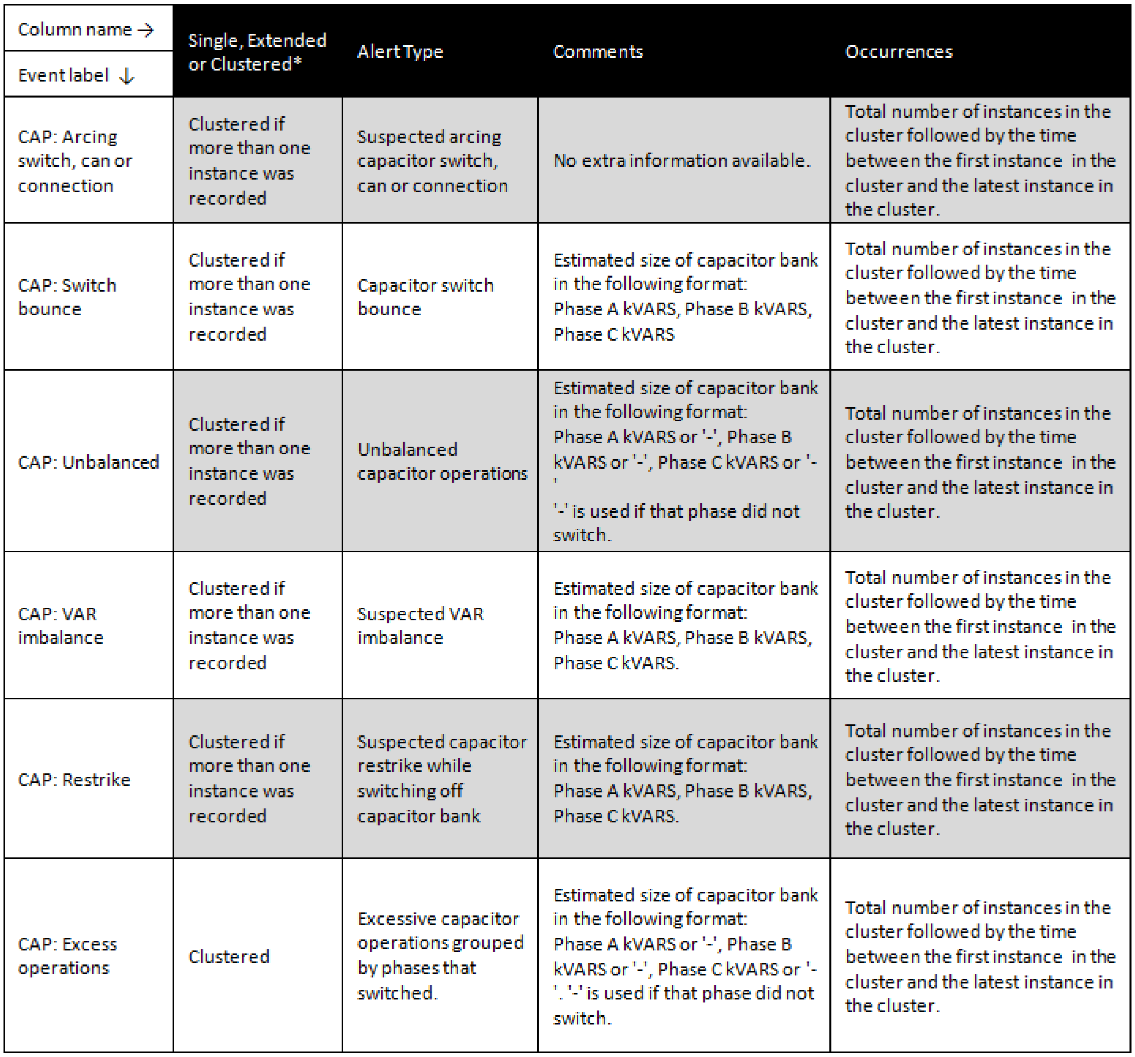
Figure 7.2 shows a screen capture of real world examples of non clustered events reported through a web interface. Figure 7.2 shows one example each for the event labels described in Table [tab:Event-specific-info-single]. The examples use obfuscated substation names and feeder names for confidentiality security reasons. The breaker close event example (third item in the screen capture) has

Table 7.3: Event specific information for non clustered events. *Definition for single, extended and clustered is provided in Table 7.2. ** Fault sequence information was explained in detail in Chapter 4, Section D.



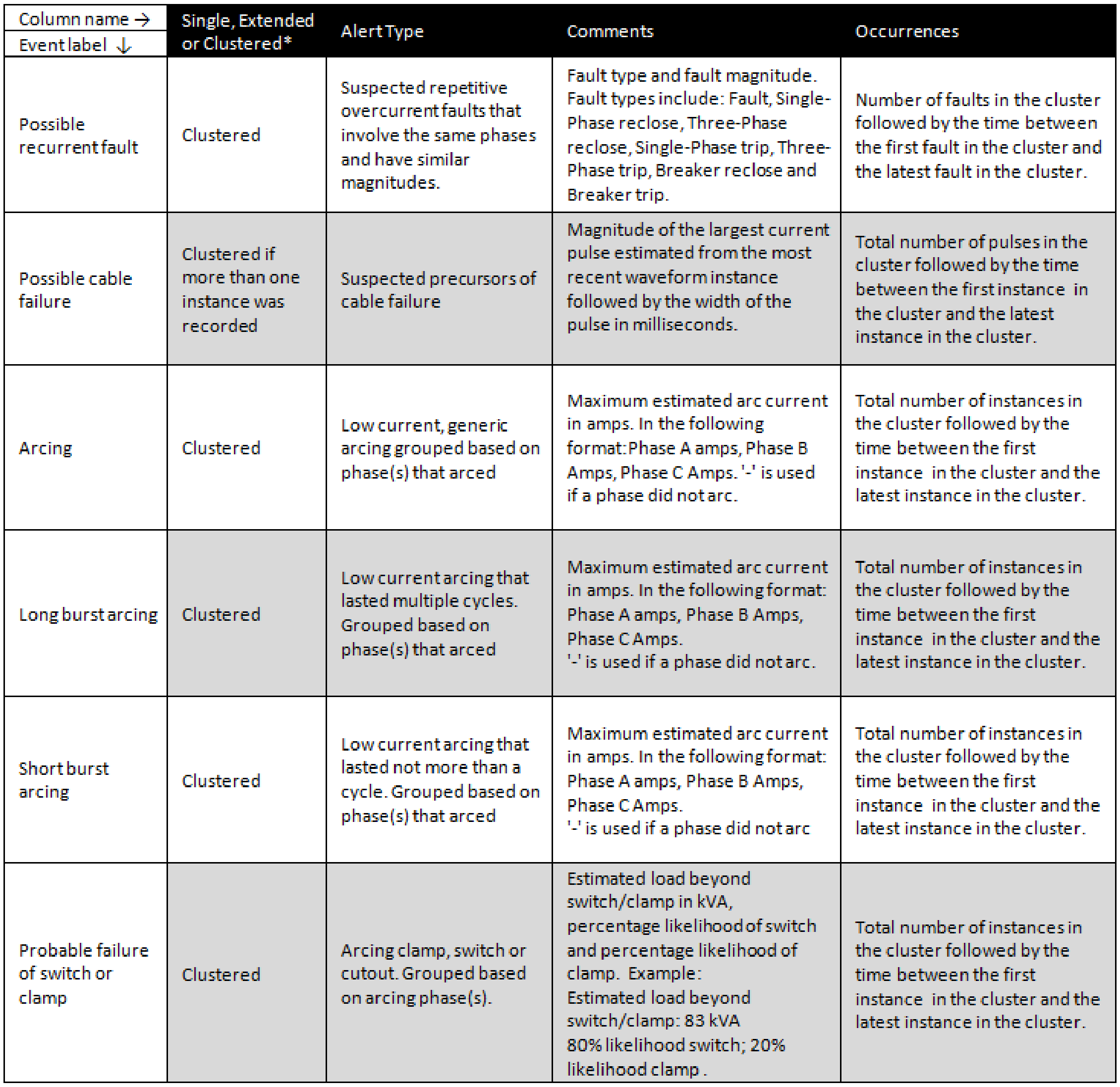
| Column name →<br>Event label ↓ | Single, Extended or Clustered* | Alert Type | Comments | Occurrences |
|---|---|---|---|---|
| Fault | Single | Generic overcurrent fault | Fault sequence** | 1 |
| Single-Phase trip | Single | Overcurrent fault that was interrupted by a single phase protective device, possibly causing an outage | Fault sequence ending with a trip** | Number of shots if applicable else N/A |
| Three-Phase trip | Single | Overcurrent fault that was interrupted by a three phase protective device, possibly causing an outage | Fault sequence ending with a trip** | Number of shots |
| Single-Phase reclose | Single | Overcurrent fault that was interrupted by a single phase protective device that may have tripped and reclosed multiple times | Fault sequence ending with a reclose** | Number of shots |
| Three-Phase reclose | Single | Overcurrent fault that was interrupted by a three phase protective device that may have tripped and reclosed multiple times | Fault sequence ending with a reclose** | Number of shots |
| Fault: Short lived | Single | Overcurrent fault that lasted less than half a cycle | Fault current and fault duration information of the form:<br>Fault magnitude Amps (fault duration milliseconds)<br>Example: 1210 Amps (9 ms) | Number of shots |
| Fault: Capacitor failure | Single | Overcurrent caused by the failure of one or more phases of a capacitor bank. | Fault current magnitude in Amps. | 1 |
| Possible conductor slap | Single | Overcurrents caused by conductor motion following a phase to phase overcurrent. Generally resulting in the tripping of a backup protective device | The type of protective device that operated (Breaker trip or Three-Phase recloser trip), followed by fault sequence information* Sequence. | Number of shots |
| Breaker close | Single | Substation breaker reclosing | None | " |
| Breaker trip | Extended sequence is reported when breaker operation spans multiple waveform files | Overcurrent fault that was interrupted by the substation breaker but finally resulting in an outage | Fault sequence ending with a trip** | Number of shots |
| Breaker reclose | Extended sequence is reported when breaker operation spans multiple waveform files | Overcurrent fault that was interrupted by the substation breaker that resulted in one or more trip and recloses | Fault sequence ending with a reclose** | Number of shots |

Table 7.4: Event specific information for capacitor problem related events

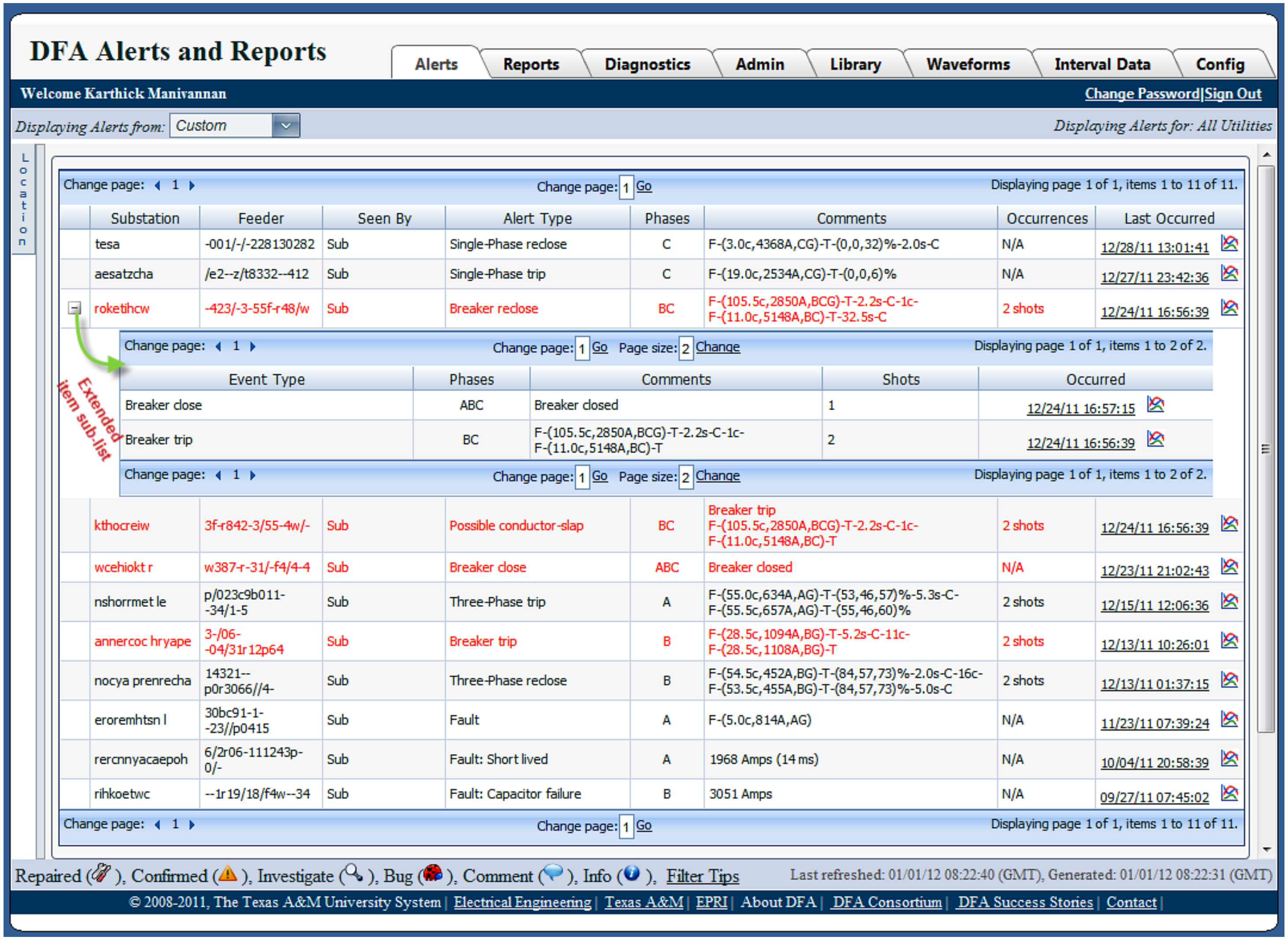


| Column name →<br>Event label ↓ | Single, Extended or Clustered* | Alert Type | Comments | Occurrences |
|---|---|---|---|---|
| CAP: Arcing switch, can or connection | Clustered if more than one instance was recorded | Suspected arcing capacitor switch, can or connection | No extra information available. | Total number of instances in the cluster followed by the time between the first instance in the cluster and the latest instance in the cluster. |
| CAP: Switch bounce | Clustered if more than one instance was recorded | Capacitor switch bounce | Estimated size of capacitor bank in the following format:<br>Phase A kVARS, Phase B kVARS, Phase C kVARS | Total number of instances in the cluster followed by the time between the first instance in the cluster and the latest instance in the cluster. |
| CAP: Unbalanced | Clustered if more than one instance was recorded | Unbalanced capacitor operations | Estimated size of capacitor bank in the following format:<br>Phase A kVARS or '-', Phase B kVARS or '-', Phase C kVARS or '-'<br>'-' is used if that phase did not switch. | Total number of instances in the cluster followed by the time between the first instance in the cluster and the latest instance in the cluster. |
| CAP: VAR imbalance | Clustered if more than one instance was recorded | Suspected VAR imbalance | Estimated size of capacitor bank in the following format:<br>Phase A kVARS, Phase B kVARS, Phase C kVARS. | Total number of instances in the cluster followed by the time between the first instance in the cluster and the latest instance in the cluster. |
| CAP: Restrike | Clustered if more than one instance was recorded | Suspected capacitor restrike while switching off capacitor bank | Estimated size of capacitor bank in the following format:<br>Phase A kVARS, Phase B kVARS, Phase C kVARS. | Total number of instances in the cluster followed by the time between the first instance in the cluster and the latest instance in the cluster. |
| CAP: Excess operations | Clustered | Excessive capacitor operations grouped by phases that switched. | Estimated size of capacitor bank in the following format:<br>Phase A kVARS or '-', Phase B kVARS or '-', Phase C kVARS or '-'. '-' is used if that phase did not switch. | Total number of instances in the cluster followed by the time between the first instance in the cluster and the latest instance in the cluster. |

Table 7.5: Event specific information for non capacitor related clustered events



| Column name → <br> Event label ↓ | Single, Extended or Clustered* | Alert Type | Comments | Occurrences |
|---|---|---|---|---|
| Possible recurrent fault | Clustered | Suspected repetitive overcurrent faults that involve the same phases and have similar magnitudes. | Fault type and fault magnitude. Fault types include: Fault, Single-Phase reclose, Three-Phase reclose, Single-Phase trip, Three-Phase trip, Breaker reclose and Breaker trip. | Number of faults in the cluster followed by the time between the first fault in the cluster and the latest fault in the cluster. |
| Possible cable failure | Clustered if more than one instance was recorded | Suspected precursors of cable failure | Magnitude of the largest current pulse estimated from the most recent waveform instance followed by the width of the pulse in milliseconds. | Total number of pulses in the cluster followed by the time between the first instance in the cluster and the latest instance in the cluster. |
| Arcing | Clustered | Low current, generic arcing grouped based on phase(s) that arced | Maximum estimated arc current in amps. In the following format:Phase A amps, Phase B Amps, Phase C Amps. '-' is used if a phase did not arc. | Total number of instances in the cluster followed by the time between the first instance in the cluster and the latest instance in the cluster. |
| Long burst arcing | Clustered | Low current arcing that lasted multiple cycles. Grouped based on phase(s) that arced | Maximum estimated arc current in amps. In the following format: Phase A amps, Phase B Amps, Phase C Amps. '-' is used if a phase did not arc. | Total number of instances in the cluster followed by the time between the first instance in the cluster and the latest instance in the cluster. |
| Short burst arcing | Clustered | Low current arcing that lasted not more than a cycle. Grouped based on phase(s) that arced | Maximum estimated arc current in amps. In the following format: Phase A amps, Phase B Amps, Phase C Amps. '-' is used if a phase did not arc | Total number of instances in the cluster followed by the time between the first instance in the cluster and the latest instance in the cluster. |
| Probable failure of switch or clamp | Clustered | Arcing clamp, switch or cutout. Grouped based on arcing phase(s). | Estimated load beyond switch/clamp in kVA, percentage likelihood of switch and percentage likelihood of clamp. Example: Estimated load beyond switch/clamp: 83 kVA 80% likelihood switch; 20% likelihood clamp . | Total number of instances in the cluster followed by the time between the first instance in the cluster and the latest instance in the cluster. |



Change page: ◂ 1 ▸ Change page: 1 Go Displaying page 1 of 1, items 1 to 11 of 11.

| Substation | Feeder | Seen By | Alert Type | Phases | Comments | Occurrences | Last Occurred |
|---|---|---|---|---|---|---|---|
| tesa | -001/-/-228130282 | Sub | Single-Phase reclose | C | F-(3.0c,4368A,CG)-T-(0,0,32)%-2.0s-C | N/A | 12/28/11 13:01:41 |
| aesatzcha | /e2--z/t8332--412 | Sub | Single-Phase trip | C | F-(19.0c,2534A,CG)-T-(0,0,6)% | N/A | 12/27/11 23:42:36 |
| roketihcw | -423/-3-55f-r48/w | Sub | Breaker reclose | BC | F-(105.5c,2850A,BCG)-T-2.2s-C-1c-<br>F-(11.0c,5148A,BC)-T-32.5s-C | 2 shots | 12/24/11 16:56:39 |

Change page: ◂ 1 ▸ Change page: 1 Go Page size: 2 Change Displaying page 1 of 1, items 1 to 2 of 2.

| Event Type | Phases | Comments | Shots | Occurred |
|---|---|---|---|---|
| Breaker close | ABC | Breaker closed | 1 | 12/24/11 16:57:15 |
| Breaker trip | BC | F-(105.5c,2850A,BCG)-T-2.2s-C-1c-<br>F-(11.0c,5148A,BC)-T | 2 | 12/24/11 16:56:39 |

Change page: ◂ 1 ▸ Change page: 1 Go Page size: 2 Change Displaying page 1 of 1, items 1 to 2 of 2.

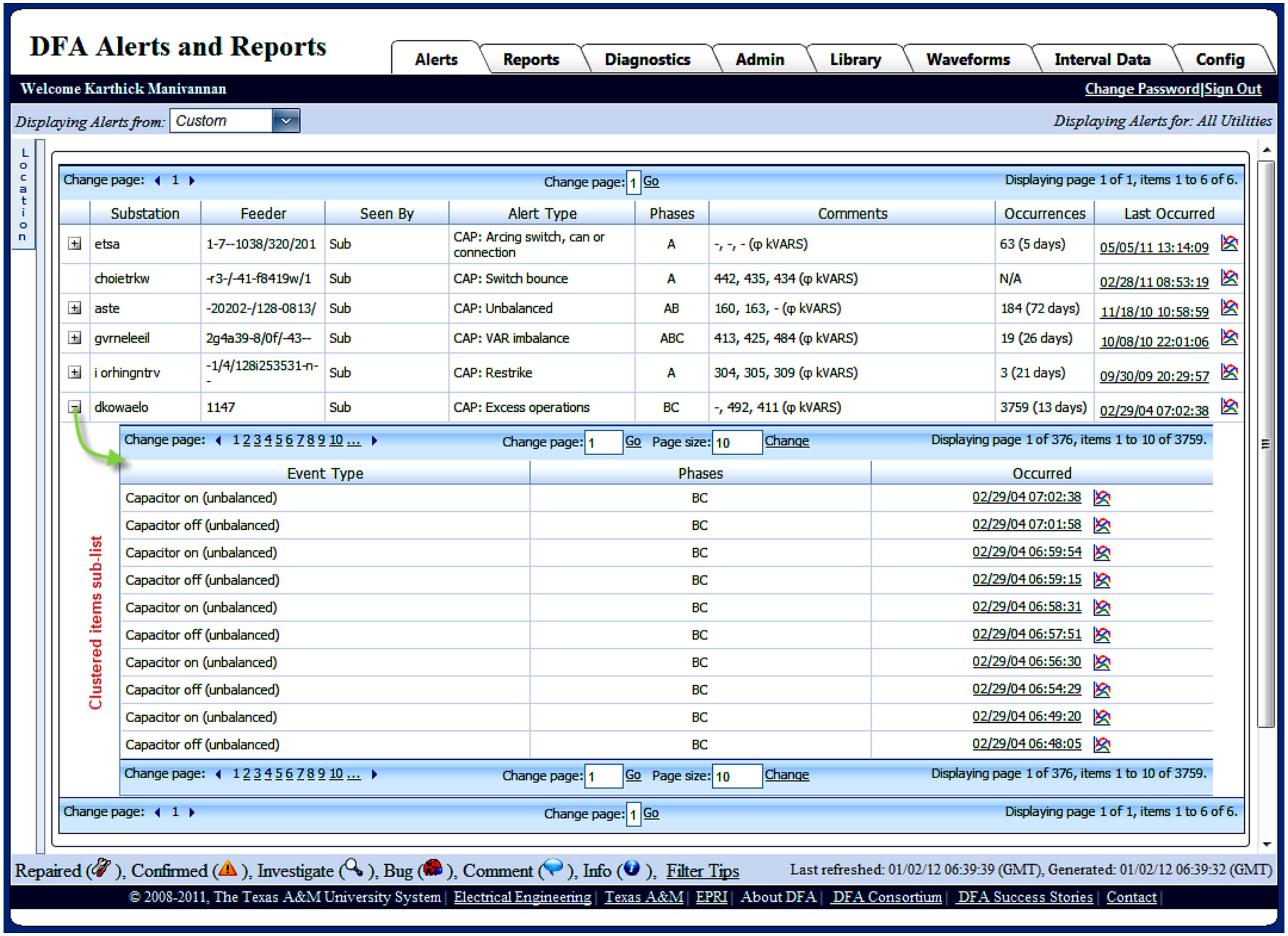
| Substation | Feeder | Seen By | Alert Type | Phases | Comments | Occurrences | Last Occurred |
|---|---|---|---|---|---|---|---|
| kthocreiw | 3f-r842-3/55-4w/- | Sub | Possible conductor-slap | BC | Breaker trip<br>F-(105.5c,2850A,BCG)-T-2.2s-C-1c-<br>F-(11.0c,5148A,BC)-T | 2 shots | 12/24/11 16:56:39 |
| wcehiokt r | w387-r-31/-f4/4-4 | Sub | Breaker close | ABC | Breaker closed | N/A | 12/23/11 21:02:43 |
| nshormet le | p/023c9b011--34/1-5 | Sub | Three-Phase trip | A | F-(55.0c,634A,AG)-T-(53,46,57)%-5.3s-C-<br>F-(55.5c,657A,AG)-T-(55,46,60)% | 2 shots | 12/15/11 12:06:36 |
| annercoc hryape | 3-/06--04/31r12p64 | Sub | Breaker trip | B | F-(28.5c,1094A,BG)-T-5.2s-C-11c-<br>F-(28.5c,1108A,BG)-T | 2 shots | 12/13/11 10:26:01 |
| nocya prenrecha | 14321--p0r3066//4- | Sub | Three-Phase reclose | B | F-(54.5c,452A,BG)-T-(84,57,73)%-2.0s-C-16c-<br>F-(53.5c,455A,BG)-T-(84,57,73)%-5.0s-C | 2 shots | 12/13/11 01:37:15 |
| eroremhtsn l | 30bc91-1--23//p0415 | Sub | Fault | A | F-(5.0c,814A,AG) | N/A | 11/23/11 07:39:24 |
| rercnnyacaepoh | 6/2r06-111243p-0/- | Sub | Fault: Short lived | A | 1968 Amps (14 ms) | N/A | 10/04/11 20:58:39 |
| rihkoetwc | --1r19/18/f4w--34 | Sub | Fault: Capacitor failure | B | 3051 Amps | N/A | 09/27/11 07:45:02 |

Change page: ◂ 1 ▸ Change page: 1 Go Displaying page 1 of 1, items 1 to 11 of 11.

Repaired (), Confirmed (), Investigate (), Bug (), Comment (), Info (), Filter Tips Last refreshed: 01/01/12 08:22:40 (GMT), Generated: 01/01/12 08:22:31 (GMT)

© 2008-2011, The Texas A&M University System | Electrical Engineering | Texas A&M | EPRI | About DFA | DFA Consortium | DFA Success Stories | Contact |

Figure 7.2: Screen capture of non clustered events reported through the web interface of IPSERS



| Substation | Feeder | Seen By | Alert Type | Phases | Comments | Occurrences | Last Occurred |
|---|---|---|---|---|---|---|---|
| etsa | 1-7--1038/320/201 | Sub | CAP: Arcing switch, can or connection | A | -, -, - (φ kVARS) | 63 (5 days) | 05/05/11 13:14:09 |
| choietrkw | -r3-/-41-f8419w/1 | Sub | CAP: Switch bounce | A | 442, 435, 434 (φ kVARS) | N/A | 02/28/11 08:53:19 |
| aste | -20202-/128-0813/ | Sub | CAP: Unbalanced | AB | 160, 163, - (φ kVARS) | 184 (72 days) | 11/18/10 10:58:59 |
| gvrneleeil | 2g4a39-8/0f/-43-- | Sub | CAP: VAR imbalance | ABC | 413, 425, 484 (φ kVARS) | 19 (26 days) | 10/08/10 22:01:06 |
| i orhingntrv | -1/4/128i253531-n-- | Sub | CAP: Restrike | A | 304, 305, 309 (φ kVARS) | 3 (21 days) | 09/30/09 20:29:57 |
| dkowaelo | 1147 | Sub | CAP: Excess operations | BC | -, 492, 411 (φ kVARS) | 3759 (13 days) | 02/29/04 07:02:38 |

Change page: 1 2 3 4 5 6 7 8 9 10 ...
Change page: 1 Go Page size: 10 Change
Displaying page 1 of 376, items 1 to 10 of 3759.
Clustered items sub-list

| Event Type | Phases | Occurred |
|---|---|---|
| Capacitor on (unbalanced) | BC | 02/29/04 07:02:38 |
| Capacitor off (unbalanced) | BC | 02/29/04 07:01:58 |
| Capacitor on (unbalanced) | BC | 02/29/04 06:59:54 |
| Capacitor off (unbalanced) | BC | 02/29/04 06:59:15 |
| Capacitor on (unbalanced) | BC | 02/29/04 06:58:31 |
| Capacitor off (unbalanced) | BC | 02/29/04 06:57:51 |
| Capacitor on (unbalanced) | BC | 02/29/04 06:56:30 |
| Capacitor off (unbalanced) | BC | 02/29/04 06:54:29 |
| Capacitor on (unbalanced) | BC | 02/29/04 06:49:20 |
| Capacitor off (unbalanced) | BC | 02/29/04 06:48:05 |

Change page: 1 2 3 4 5 6 7 8 9 10 ...
Change page: 1 Go Page size: 10 Change
Displaying page 1 of 376, items 1 to 10 of 3759.
Change page: 1
Change page: 1 Go
Displaying page 1 of 1, items 1 to 6 of 6.
Repaired ( ), Confirmed ( ), Investigate ( ), Bug ( ), Comment ( ), Info ( ), Filter Tips
Last refreshed: 01/02/12 06:39:39 (GMT), Generated: 01/02/12 06:39:32 (GMT)
© 2008-2011, The Texas A&M University System | Electrical Engineering | Texas A&M | EPRI | About DFA | DFA Consortium | DFA Success Stories | Contact |

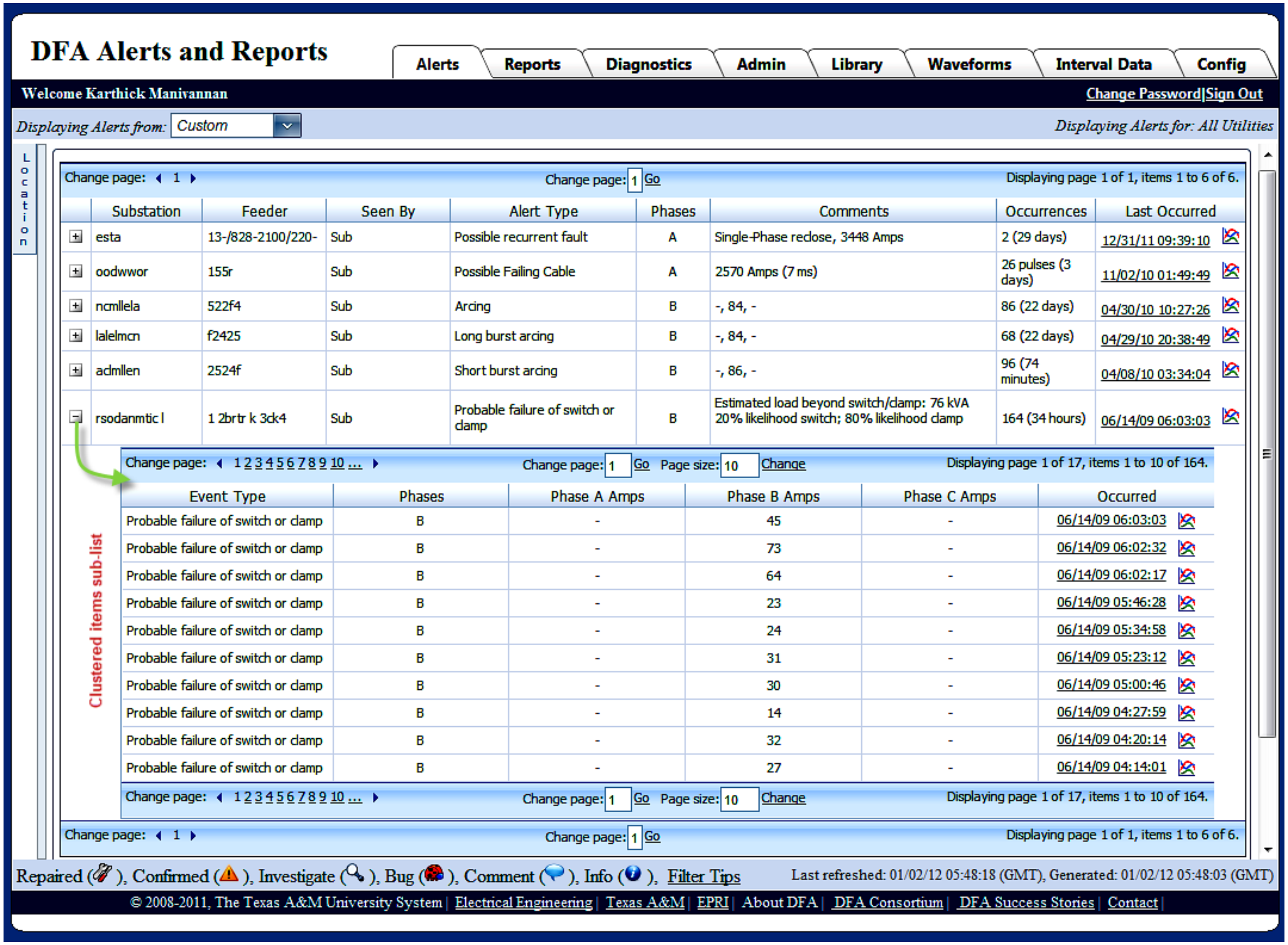
Figure 7.3: Screen capture of capacitor related events reported through the web interface of IPSERS

an extended list that spanned two waveform files, a breaker trip and a breaker reclose. Breaker trip was recorded in one waveform file. Then the breaker reclosed after 32.5 seconds, which was recorded in another waveform file. The reporting algorithms are capable of piecing the breaker trip and breaker close events together and reporting them as a single breaker reclose operation

Figure 7.3 shows a screen capture of real world examples of capacitor related events reported through a web interface. Figure 7.3 shows one example each corresponding to the event labels described in Table 7.4. The last item in the screen capture corresponds to an excessive capacitor operation cluster. The expanded cluster shows individual occurrences. The individual items in the cluster show excessive operation of an unbalanced capacitor bank that operated several thousand times within a span of two weeks.

Figure 7.4 shows a screen capture of real world examples of clustered events reported through a web interface excluding those related to capacitors. Figure 7.4 shows one example each corresponding to the event labels described in Table 7.5. The last item in the screen capture corresponds to a switch or clamp failure cluster. The expanded cluster shows individual occurrences.

Through event reports such as those shown in Figures 7.2, 7.3 and 7.4, IPSERS provides utility users easy access to information regarding the health of their feeders. This information can then be used by utility users to do a condition based maintenance if the reported event corresponds to a failing component. In other scenarios, event information can be used to diagnose an existing problem. However, the usefulness of a system such as IPSERS is dependent on the accuracy of the reported events. The accuracy of IPSERS depends on the accuracy of the underlying classification algorithm (FLCA) and the accuracy of reporting algorithms. The following section will analyze classification performance of FLCA



*Displaying Alerts from:* Custom

*Displaying Alerts for: All Utilities*

Change page: ◂ 1 ▸ Change page: 1 Go Displaying page 1 of 1, items 1 to 6 of 6.

| | Substation | Feeder | Seen By | Alert Type | Phases | Comments | Occurrences | Last Occurred |
|---|---|---|---|---|---|---|---|---|
| ⊞ | esta | 13-/828-2100/220- | Sub | Possible recurrent fault | A | Single-Phase reclose, 3448 Amps | 2 (29 days) | 12/31/11 09:39:10 |
| ⊞ | oodwwor | 155r | Sub | Possible Failing Cable | A | 2570 Amps (7 ms) | 26 pulses (3 days) | 11/02/10 01:49:49 |
| ⊞ | ncmllela | 522f4 | Sub | Arcing | B | -, 84, - | 86 (22 days) | 04/30/10 10:27:26 |
| ⊞ | lalelmcn | f2425 | Sub | Long burst arcing | B | -, 84, - | 68 (22 days) | 04/29/10 20:38:49 |
| ⊞ | admllen | 2524f | Sub | Short burst arcing | B | -, 86, - | 96 (74 minutes) | 04/08/10 03:34:04 |
| ⊟ | rsodanmtic l | 1 2brtr k 3ck4 | Sub | Probable failure of switch or clamp | B | Estimated load beyond switch/clamp: 76 kVA 20% likelihood switch; 80% likelihood clamp | 164 (34 hours) | 06/14/09 06:03:03 |

Clustered items sub-list

Change page: ◂ 1 2 3 4 5 6 7 8 9 10 ... ▸ Change page: 1 Go Page size: 10 Change Displaying page 1 of 17, items 1 to 10 of 164.

| Event Type | Phases | Phase A Amps | Phase B Amps | Phase C Amps | Occurred |
|---|---|---|---|---|---|
| Probable failure of switch or clamp | B | - | 45 | - | 06/14/09 06:03:03 |
| Probable failure of switch or clamp | B | - | 73 | - | 06/14/09 06:02:32 |
| Probable failure of switch or clamp | B | - | 64 | - | 06/14/09 06:02:17 |
| Probable failure of switch or clamp | B | - | 23 | - | 06/14/09 05:46:28 |
| Probable failure of switch or clamp | B | - | 24 | - | 06/14/09 05:34:58 |
| Probable failure of switch or clamp | B | - | 31 | - | 06/14/09 05:23:12 |
| Probable failure of switch or clamp | B | - | 30 | - | 06/14/09 05:00:46 |
| Probable failure of switch or clamp | B | - | 14 | - | 06/14/09 04:27:59 |
| Probable failure of switch or clamp | B | - | 32 | - | 06/14/09 04:20:14 |
| Probable failure of switch or clamp | B | - | 27 | - | 06/14/09 04:14:01 |

Change page: ◂ 1 2 3 4 5 6 7 8 9 10 ... ▸ Change page: 1 Go Page size: 10 Change Displaying page 1 of 17, items 1 to 10 of 164.

Change page: ◂ 1 ▸ Change page: 1 Go Displaying page 1 of 1, items 1 to 6 of 6.

Repaired ( ), Confirmed ( ), Investigate ( ), Bug ( ), Comment ( ), Info ( ), Filter Tips Last refreshed: 01/02/12 05:48:18 (GMT), Generated: 01/02/12 05:48:03 (GMT)

© 2008-2011, The Texas A&M University System | Electrical Engineering | Texas A&M | EPRI | About DFA | DFA Consortium | DFA Success Stories | Contact |

Figure 7.4: Screen capture of non capacitor related clustered events reported through the web interface of IPSERS

and reporting algorithms.

## 7.3 Classification Performance on Field Data

Fuzzy logic based classification algorithm (FLCA) and reporting algorithm that are part of IPSERS process a large volume of data on a daily basis. IPSERS has been online for a few years and has processed over a million records. Currently IPSERS is processing data from about 100 feeders and underground networks belonging to twelve utilities. In the year 2011 alone, IPSERS processed about 373610 waveform files. To simplify the performance analysis, only results corresponding to data acquired from substation based measurements of radial distribution feeders will be considers. IPSERS currently analyzes data from other locations such as non substation based locations on distribution feeders, transmission system and low voltage networks. However, analysis data from these locations fall outside the scope of this dissertation.

### *7.3.1 Classification Performance of FLCA*

FLCA has been online and classifying waveform files for the past several years. FLCA has also been evolving to accommodate new event signatures. Both the rule base and feature extraction algorithms of FLCA are also continuously refined to improve classification accuracy. FLCA has been exhibiting good classification accuracy in spite of the complex nature of the classification problem. Because FLCA is a rule based system, it does not require a training set. The rules, membership parameters and feature extraction algorithms together embed expert knowledge. Other than during the initial development stage, membership parameters and thresholds used by the rule base rarely required changes. Using methods such as cross-validation and bootstrapping [114] for evaluating performance of machine learning techniques are not applicable here, due to the lack of a training set. Further, it is not possible to compare the classification performance of FLCA with any existing algorithm. This is because, the power system event classification problem as stated in this dissertation, has not been solved before.

FLCA classifies large volume of data on a continuous basis and generates a large number of classifications. Verifying the accuracy of the large number of classifications was one of the challenges in trying to evaluate the performance of FLCA. For example, when considering waveform files acquired through substation based monitoring of non-network radial distribution feeders; the number of waveform files processed by FLCA in the year 2011 alone was 199295. This is too large a data set to manually verify classification accuracy and present results. Hence, this classification accuracy evaluation uses only a subset of the data corresponding to a continuous window of one week. The classification accuracy evaluation and results presented here use a data window of one week from December 2011. This data window was chosen because, the results would then correspond to the most recent version of FLCA. During this period, FLCA processed a total of 1609 waveform files acquired from substation based measurements of radial distribution feeders. FLCA assigned class labels for 952 files. FLCA did not assign classification to the rest of the 657 files. It should be noted that FLCA does not always assign a class label to a waveform file. When FLCA does not generate high enough possibility for any of the event categories, it does not assign a classification. The most common reasons for not assigning a classification are:

1. The waveform data correspond to normal or abnormal system events, but the magnitude of these events were too small and hard to distinguish from ambient noise. The FLCA uses sensitivity thresholds to avoid false positives on abnormal events. If the event magnitudes fall bellow the sensitivity

threshold, FLCA does not assign a class label.

2. The waveform data contained normal system variations captured because of sensitive triggering. In this case, the waveform data does not contain any event signatures that can be recognized and labeled by FLCA.

3. The waveform data contained data from normal or abnormal system events that have not been fully characterized and understood. Hence, FLCA lacks the required knowledge to classify the data.

The classification results uses a color coded confusion matrix that weighs different regions based on user perspective.

#### *7.3.1.1 Color coded confusion matrix*

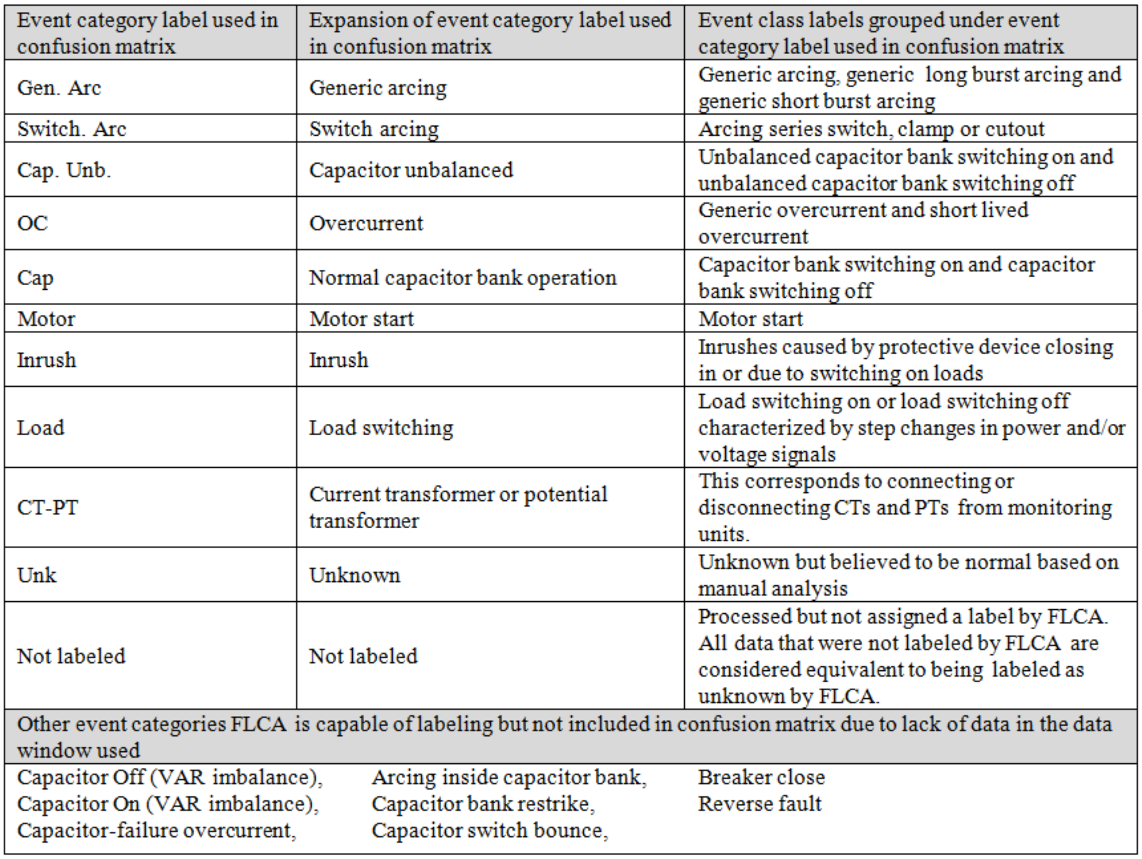
Table 7.6 shows the result of classifying 1609 waveform files by FLCA during the one week window in the form of a confusion matrix. Each column in the confusion matrix represents classes labeled by FLCA while each row in the confusion matrix represents the actual class (i.e., the class assigned manually based on visual analysis of waveform data). Values along the diagonal (shown in shades of green) represent the number of instances where manual analysis and FLCA concur. Values on off diagonal elements represent the number of instances where FLCA misclassified. The rightmost column with the column name 'Not labeled' represents data processed but not labeled by FLCA for reasons mentioned previously. The bottom most row manually labeled as labeled 'Unk', stands for unknown but normal events. This represents event data that the author is not able to label as but believes to be normal system events. Multiple event labels were grouped together into a single category to simplify their analysis. For example, the category 'Cap' in the confusion matrix represents both capacitor bank switching on and capacitor

Table 7.6: Confusion matrix of data labeled by FLCA using a one week window

| | | | | Labled by FLCA | | | | | | | | | | | | | | Not labled |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | Abnormal | | | | | Normal | | | | | | | | | |
| | | | | Monitored | | | | Non-Mon. | Monitored | | | | | Non-Mon. | | | | |
| | | | | Gen. Arc | Switch. Arc | Cap. Unb. | OC | OC | Cap | Motor | Inrush | Load | CT-PT | Cap | Motor | Inrush | Loa d | |
| Actual (manually classified) | Abnormal | Monitored | Gen. Arc | 39 | | | | | | 1 | | | | | | | | 5 |
| | | | Switch. Arc | | 2 | | | | | | | | | | | | | |
| | | | Cap. Unb. | | | 1 | | | | | | | | | | | | |
| | | | OC | | | | 7 | | | | | | | | | | | |
| | | Non-Mon. | OC | | | | | 174 | | 3 | | | | | | | | 33 |
| | Normal | Monitored | Cap | | | | | | 134 | | | | | | | | | 15 |
| | | | Motor | 5 | | | | | 1 | 202 | 1 | | | | | | | 42 |
| | | | Inrush | | | | | | | 3 | 84 | | | | | | | 45 |
| | | | Load | | | | | | | | | 19 | | | | | | 110 |
| | | | CT-PT | | | | | | | | | | 2 | | | | | |
| | | Non-Mon. | Cap | | | | | 6 | | | | | | 180 | | | | 21 |
| | | | Motor | | | | | | | | | | | | 45 | | | |
| | | | Inrush | | | | | 13 | | | | | | | | 17 | | 32 |
| | | | Load | | | | | | | | | | | 5 | | | | 132 |
| | | Unk | | | | | | 2 | | 3 | 2 | | | | | | | 222 |

| Legend | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Correct classifications | | | | | Misclassifications | | | | | | | | | |
| Usefulness | High | Medium | Low | Overall | Cost of misclassification | High | | Medium | | Low | | Very Low | | |
| Color code | 🟩 | 🟩 | 🟩 | | Color code | 🟥 | 🟥 | 🟧 | 🟨 | 🟦 | 🟦 | ⬜ | ⬜ | |

Table 7.7: Description of event category labels



| Event category label used in confusion matrix | Expansion of event category label used in confusion matrix | Event class labels grouped under event category label used in confusion matrix |
|---|---|---|
| Gen. Arc | Generic arcing | Generic arcing, generic long burst arcing and generic short burst arcing |
| Switch. Arc | Switch arcing | Arcing series switch, clamp or cutout |
| Cap. Unb. | Capacitor unbalanced | Unbalanced capacitor bank switching on and unbalanced capacitor bank switching off |
| OC | Overcurrent | Generic overcurrent and short lived overcurrent |
| Cap | Normal capacitor bank operation | Capacitor bank switching on and capacitor bank switching off |
| Motor | Motor start | Motor start |
| Inrush | Inrush | Inrushes caused by protective device closing in or due to switching on loads |
| Load | Load switching | Load switching on or load switching off characterized by step changes in power and/or voltage signals |
| CT-PT | Current transformer or potential transformer | This corresponds to connecting or disconnecting CTs and PTs from monitoring units. |
| Unk | Unknown | Unknown but believed to be normal based on manual analysis |
| Not labeled | Not labeled | Processed but not assigned a label by FLCA. All data that were not labeled by FLCA are considered equivalent to being labeled as unknown by FLCA. |
| Other event categories FLCA is capable of labeling but not included in confusion matrix due to lack of data in the data window used | | |
| Capacitor Off (VAR imbalance),<br>Capacitor On (VAR imbalance),<br>Capacitor-failure overcurrent, | Arcing inside capacitor bank,<br>Capacitor bank restrike,<br>Capacitor switch bounce, | Breaker close<br>Reverse fault |

bank switching off events labels. Event categories have also been abbreviated in the confusion matrix. Table 7.7 describes each event category and the event labels that were grouped into that event category. The bottom of 7.7 shows event labels that FLCA is capable of assigning, but, did not appear in the confusion matrix. These event labels do not appear in the confusion matrix because, the data window used to generate the confusion matrix did not contain waveform data from these event classes.

In the confusion matrix, class labels have been grouped under two levels. First by normality and then by position with respect to the point of measurement:

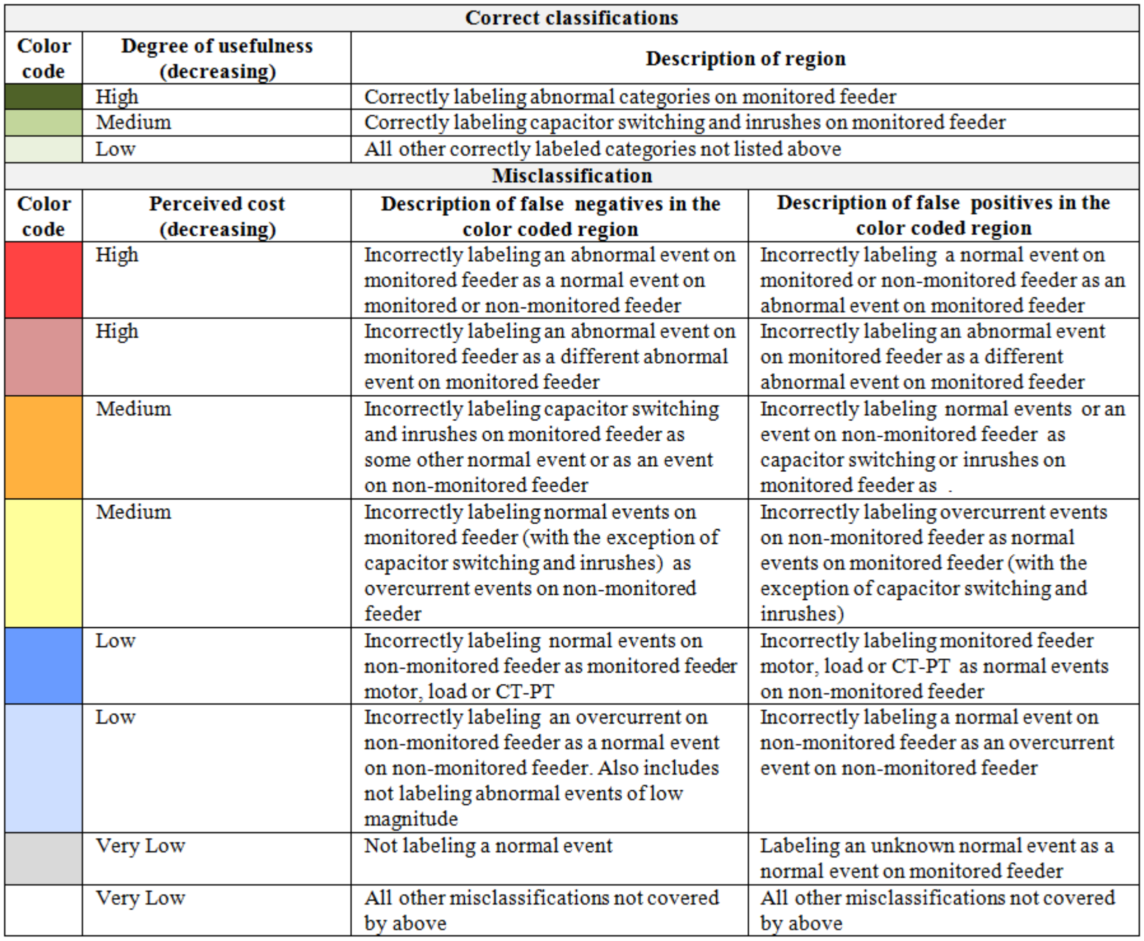
1. Normal vs. abnormal: Event data may represent either normal system events

or abnormal events. It is useful to group events by normality. This is because, abnormal system events are most likely to be reported to users. Misclassification between normal and abnormal events can be expensive from a user's perspective. False positives (i.e., labeling normal events as abnormal events) could cause the user to become desensitized and loose trust in the system. False negatives (i.e., labeling abnormal events as normal events) would cause the user to miss true abnormal events and could undermine the value of the system. Misclassification within abnormal events could be expensive too. This is because the user may be provided with a wrong cause of the event and hence could cause delay in locating the source of the problem. Misclassification within normal event categories is much more tolerable from the perspective of the user as IPSERS does not report normal normal events to the user. However, certain normal events such as those caused by capacitor switching and inrushes may be of interest to some utilities.

2. Position with respect to the monitoring device: Events on the feeder being monitored (down stream of the monitoring device) are of most interest to users. However, monitoring devices can also record data corresponding to events that happened on non-monitored locations such as adjacent feeders, transmission system or bus. However, data corresponding to events on non-monitored location will only contain partial information needed to diagnose the event cause. These will have usable voltage measurements but not current measurements.

Clearly, it is hard to evaluate the performance of FLCA based on overall accuracy of labeling all event categories. This is because, from a user perspective, accuracy of correctly labeling some events may carry more weighting than that

Table 7.8: Legend of color codes used in confusion matrix (Table 7.6)



| Correct classifications | | | |
|---|---|---|---|
| **Color code** | **Degree of usefulness (decreasing)** | **Description of region** | |
| 🟩 | High | Correctly labeling abnormal categories on monitored feeder | |
| 🟩 | Medium | Correctly labeling capacitor switching and inrushes on monitored feeder | |
| 🟩 | Low | All other correctly labeled categories not listed above | |
| **Misclassification** | | | |
| **Color code** | **Perceived cost (decreasing)** | **Description of false negatives in the color coded region** | **Description of false positives in the color coded region** |
| 🟥 | High | Incorrectly labeling an abnormal event on monitored feeder as a normal event on monitored or non-monitored feeder | Incorrectly labeling a normal event on monitored or non-monitored feeder as an abnormal event on monitored feeder |
| 🟥 | High | Incorrectly labeling an abnormal event on monitored feeder as a different abnormal event on monitored feeder | Incorrectly labeling an abnormal event on monitored feeder as a different abnormal event on monitored feeder |
| 🟧 | Medium | Incorrectly labeling capacitor switching and inrushes on monitored feeder as some other normal event or as an event on non-monitored feeder | Incorrectly labeling normal events or an event on non-monitored feeder as capacitor switching or inrushes on monitored feeder as . |
| 🟨 | Medium | Incorrectly labeling normal events on monitored feeder (with the exception of capacitor switching and inrushes) as overcurrent events on non-monitored feeder | Incorrectly labeling overcurrent events on non-monitored feeder as normal events on monitored feeder (with the exception of capacitor switching and inrushes) |
| 🟦 | Low | Incorrectly labeling normal events on non-monitored feeder as monitored feeder motor, load or CT-PT | Incorrectly labeling monitored feeder motor, load or CT-PT as normal events on non-monitored feeder |
| 🟦 | Low | Incorrectly labeling an overcurrent on non-monitored feeder as a normal event on non-monitored feeder. Also includes not labeling abnormal events of low magnitude | Incorrectly labeling a normal event on non-monitored feeder as an overcurrent event on non-monitored feeder |
| ⬜ | Very Low | Not labeling a normal event | Labeling an unknown normal event as a normal event on monitored feeder |
| | Very Low | All other misclassifications not covered by above | All other misclassifications not covered by above |

other categories. For this reason, the confusion matrix is color coded. For this reason, classification accuracies and misclassification rates within each color coded region will be presented and analyzed separately. Overall accuracy will also be presented. Each color represents a degree of usefulness of correctly labeled data or the perceived cost of misclassified data. The degree of usefulness of correctly labeled data and the perceived cost of misclassified data are highly subjective and will depend on both the user and utility practices. The color codes presented in the confusion matrix is based on the author's perspective. Table 7.8 presents a legend that associates each color code with a degree of importance and also describes the region in the confusion matrix that the color code represents.

Percentage classification accuracy, percentage false positives and percentage false negatives, were calculated for color coded regions (7.8) within the confusion matrix. The following paragraphs, outline the method used for computing these values.

Percentage classification accuracy for a class $M$ can be calculated using the following equations:

$$PcntCA_M = \frac{TP_M}{TP_M + MC_M} \times 100 \tag{7.1}$$

$$MC_M = FN_M + FP_M \tag{7.2}$$

where, $PcntCA_M$ represents the percentage accuracy for a class $M$, $TP_M$ represents true positives for a class $M$, $MC_M$ represents misclassification involving class M, $FN_M$ represents false negatives for a class $M$ and $FP_M$ represent false positives for a class $M$. Consider any instance of event data $E$ for which FLCA assigned a class $A$, such that $A \in U$, where $U$ is the set of all class labels. The labeling of instance $E$ by FLCA is counted towards true positives ($TP_M$) for class $M$, $M \in U$ if $A = M$ and if $M$ is indeed the correct class label for $E$. The labeling of instance $E$ by FLCA is counted towards false negative ($FN_M$) for class $M$ if $A \neq M$ and if $M$ is indeed the correct class label for $E$. The labeling of instance $E$ by FLCA is counted towards false negative ($FP_M$) for class $M$ if $A = M$ but $M$ is not the correct class label for $E$.

Equation 7.1 can be further generalized to calculate classification accuracy for a region $R_{cm} \subset U$ in the confusion matrix, where $R_{cm}$ represents a subset of valid classes that can be assigned to an event instance $E$. The following equations were used for calculating percentage classification accuracy for a region $R_{cm}$:

$$PcntCA_{R_{cm}} = \frac{TP_{R_{cm}}}{TP_{R_{cm}} + MC_{R_{cm}}} \times 100 \tag{7.3}$$

$$TP_{R_{cm}} = \sum_{i \in R_{cm}} TP_i \tag{7.4}$$

$$MC_{R_{cm}} = FN_{R_{cm}} + FP_{R_{cm}} \tag{7.5}$$

where, $PcntCA_{R_{cm}}$ represents the percentage accuracy for a region $R_{cm}$, $MC_{R_{cm}}$ represents misclassification involving class labels in the region $R_{cm}$, $TP_{R_{cm}}$ represents true positives for a the region $R_{cm}$, $FN_{R_{cm}}$ represents false negatives for a the region $R_{cm}$ and $FP_{R_{cm}}$ represent false positives for a the region $R_{cm}$. Consider any instance of event data $E$ for which FLCA assigned a class $A$, such that $A \in U$ , where $U$ is the set of all class labels. The labeling of instance $E$ by FLCA is counted towards true positives ($TP_{R_{cm}}$) for region $R_{cm}$ if $A \in R_{cm}$ and if $A$ is indeed the correct class label for $E$. The labeling of instance $E$ by FLCA is counted towards false negative ($FN_{R_{cm}}$) for region $R_{cm}$ if the correct class label for $E$ is $M$, $M \in R_{cm}$ but $A \neq M$. The labeling of instance $E$ by FLCA is counted towards false positive ($FP_{R_{cm}}$) for region $R_{cm}$ if $A \in R_{cm}$ but $A$ is not the correct class label for $E$.

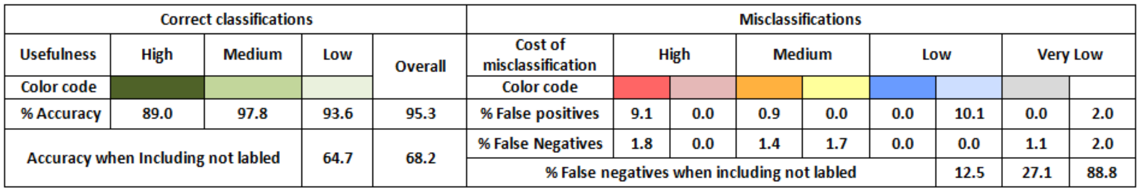
The following equation were used for calculating percentage false negatives $PcntFN_{R_{cm}}$ and percentage false positives $PcntFP_{R_{cm}}$ for a region $R_{cm}$:

$$PcntFN_{R_{cm}} = \frac{FN_{R_{cm}}}{TP_{R_{cm}} + MC_{R_{cm}}} \times 100 \tag{7.6}$$

$$PcntFP_{R_{cm}} = \frac{FP_{R_{cm}}}{TP_{R_{cm}} + MC_{R_{cm}}} \times 100 \tag{7.7}$$

In the confusion matrix7.6, all cells that do not have diagonal lines are true positives. Cells that correspond to true positives appear in shades of green. Shades of

Table 7.9: Analysis of classification accuracy



| Correct classifications | | | | | Misclassifications | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Usefulness | High | Medium | Low | Overall | Cost of misclassification | High | | Medium | | Low | | Very Low | |
| Color code | 🟩 | 🟩 | 🟩 | | Color code | 🟥 | 🟥 | 🟧 | 🟨 | 🟦 | 🟦 | ⬜ | |
| % Accuracy | 89.0 | 97.8 | 93.6 | 95.3 | % False positives | 9.1 | 0.0 | 0.9 | 0.0 | 0.0 | 10.1 | 0.0 | 2.0 |
| Accuracy when Including not labled | | | 64.7 | 68.2 | % False Negatives | 1.8 | 0.0 | 1.4 | 1.7 | 0.0 | 0.0 | 1.1 | 2.0 |
| | | | | | % False negatives when including not labled | | | | | | 12.5 | 27.1 | 88.8 |

green also represent the degree of usefulness from user's perspective. Darker the shade of green, more useful a true positive is. For example, correct classification of abnormal events on monitored feeder is most useful. Hence, these appear in the darkest shade of green. Table 7.8 lists the degree of usefulness associated with each shade of green. It also describes what true positives in the color coded region represent.

For a given region, cells with upward diagonals ╱ count toward false negative and cells with downward diagonals ╲ count towards false positives. Cells with both downward and upward diagonal count toward both false positive and false negatives. Cells belonging to the same region have the same color. Cells are color coded in different shades starting from red to white. Red represents the most expensive misclassification from a user's perspective, and white represents the least expensive misclassification. For example, labeling normal events as abnormal events on monitored feeder may be perceived as an expensive false alarm. Hence, this region of false positives is coded in red and has an upward diagonal. Table 7.8 lists the perceived cost of misclassification associated with each color coded region. It also describes what false positives or false negatives in the color coded region represent.

#### 7.3.1.2 *Analysis of classification accuracy and misclassification rates*

Table 7.9 presents percentage classification accuracy, percentage false positives and percentage false negatives. Equations 7.3-7.7 were used to calculate these values for each color coded regions within the confusion matrix.

Percentage classification accuracy is the proportion of all correct identification (true positives) for a data set expressed as a percentage. Percentage classification accuracy is traditionally used as a measure of classification accuracy for classification algorithm. Computing overall classification accuracy in this manner, does not provide insight into whether or not FLCA classified certain important but relatively infrequent events correctly.

High classification accuracy on normal system events would easily mask the classification accuracy achieved in events of interest that are relatively infrequent. For example, ignoring events that FLCA did not label, FLCA had a classification accuracy of 95.3% for the chosen data window of one week. This is a high classification accuracy given the complex nature of the classification problem. Classification accuracy was also computed separately for regions considered being of high, medium and low usefulness. It can be seen in the Table7.9 that the classification accuracy drops to 89% for the region of high usefulness. This region represents correct classification of abnormal events on monitored feeder. While 89% classification rate on the important events looks low, it should be noted that all of them involved the misclassification of low current, single phase, generic arcing events. Moreover, IPSERS did not report any of these false positives on generic arcing events to users. This is because, IPSERS does not report generic arcing events to users, unless, IPSERS detected multiple such events on the same feeder within a short period that involve the same phases. Higher level reporting algorithms help filter raw classification data and present only important and useful information to the user. They also help to avoid false alarms. Computing classification accuracy based on regions did help to uncover a possible area for improvement of FLCA.

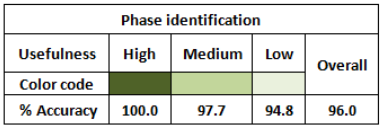
I.e., improving the identification of low current generic arcing events that FLCA confused with single phase motor starts. A 9.1% false positive percentage was computed for the region of high cost (coded in red and downward diagonal line). All these false positives were due to FLCA labeling motor starts as generic arcing events. A 1.8% false negative was computed for the region of high misclassification cost. This was due to FLCA labeling a single instance of generic arcing event as a motor start.

FLCA achieved a high classification accuracy of 97.8 for the region of medium usefulness. This region corresponds to the correct identification of normal capacitor bank switching on monitored feeder and inrush transients on monitored feeder. Correct identification of capacitor operations is helpful in determining the health of capacitor banks on a feeder. IPSERS does not report Individual capacitor operations to users, but it reports excessive operation of capacitor banks. However, in the case of excessive capacitor bank operations (several hundreds to thousands within a week), occasional misclassification does not significantly affect what IPSERS reports to the user. Hence normal capacitor bank switching was assigned to the region of medium importance where some error is tolerable. Inrush transients are not typically reported to users unless the reclosing of substation circuit breaker caused the inrush. Some users may also be interested in inrushes caused by the closing of automatic line reclosers. However, these correspond to a small percentage of the total number of inrush transients that occur on a feeder. Majority of the inrush transients are not of much importance as they caused by switching of loads. For this reason, inrushes were also assigned to the region of medium importance. False positives and false negatives for this region (coded in orange) also carry a medium cost. It can be seen that here too all the misclassification were due to FLCA misclassifying motor starts.

All events on non-monitored feeder and some events on the monitored feeder such as motor starts, and load switching carry the least importance as they are never reported to the user. They are classified and ignored. Hence these events were assigned to the region of low importance. The classification accuracy for this region was computed to be 93.6%. Even though this is is not a high classification accuracy, some amount of misclassification in this region is tolerable as it does not affect the results presented to users. This is also the reason why most false positives or false negatives in this region also carry a low cost (regions coded blue through white). The only exception are the false positives and false negatives corresponding to overcurrent events on non-monitored feeder. Overcurrents on non-monitored feeder cause temporary dips in line voltage and can have some value in terms of power quality. For this reason, misclassification related to overcurrents on non-monitored feeder were classified as normal events on monitored feeder were assigned a medium cost (region coded in yellow). A total of 3 false negatives corresponding to a relatively low 1.7 % false negative percentage was computed for this region.

The above analysis of classification accuracy did not include the 41% of total events that FLCA processed, but did not label. This could be considered equivalent to FLCA labeling them as being unknown but normal evens. If the analysis includes the events that FLCA did not label, the overall accuracy, drops drastically from 95.3% to 68.2%. However, the revised 68.2% classification accuracy can be misleading. This is because, almost all events that FLCA did not label corresponded to events of low importance. The only exception were a few generic arcing events that had low current magnitudes. Further, most of the events that FLCA did not label involved events of low magnitudes that were hard to distinguish from ambient noise even during manual analysis. FLCA was designed

Table 7.10: Phase identification performance of FLCA



| Phase identification | | | | |
|---|---|---|---|---|
| Usefulness | High | Medium | Low | Overall |
| Color code | ■ | ▨ | □ | |
| % Accuracy | 100.0 | 97.7 | 94.8 | 96.0 |

to ignore events that did not have large enough magnitudes that meet sensitivity thresholds. This was done to decrease false alarms caused by normal system variations. For this reason, the events that FLCA did not label were added to the region of low importance and the classification accuracy and misclassification rates were presented separately. When monitoring units have sensitive triggering, it could cause a large number of non labeled events. This could cause almost continuous stream of data containing minor system variations. It would, however, be beneficial if FLCA could label these as normal system variations; this is a deficiency of FLCA that needs to be addressed in the future.

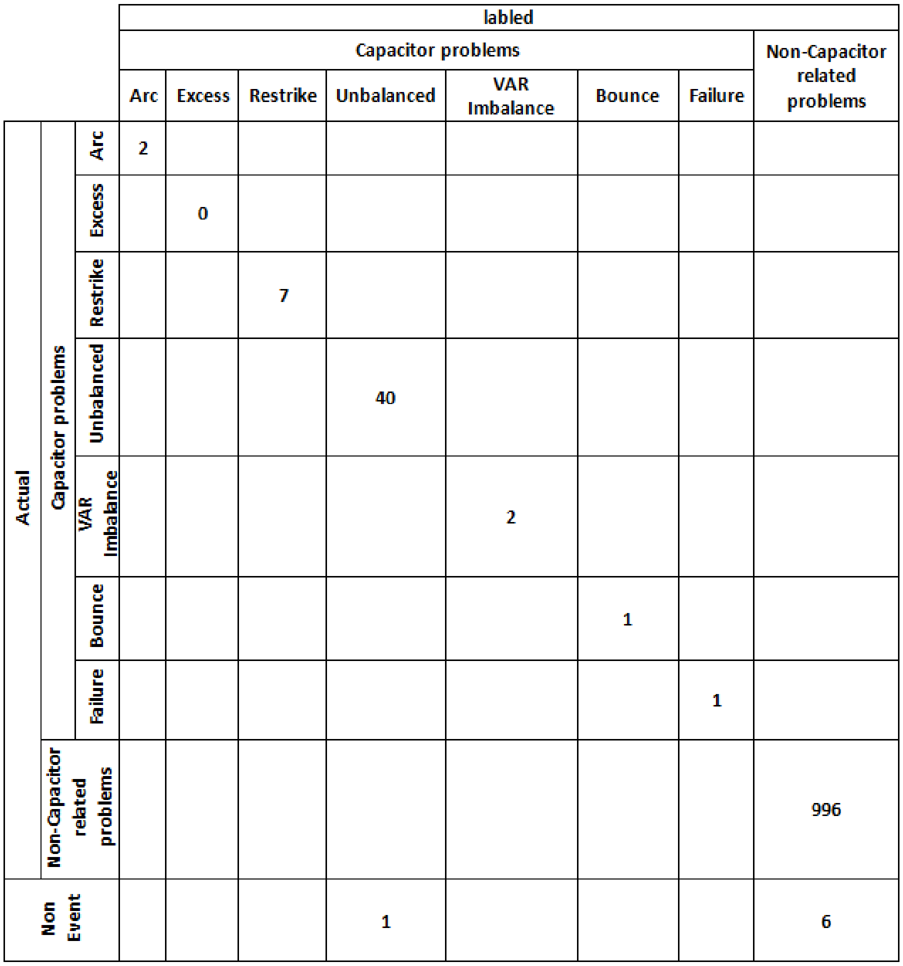
Another attribute of FLCA output label is phase information. The confusion matrix did not include phase information to simplify analysis. Table 7.10 shows classification accuracy for phase identification by FLCA. Only events with correctly labeled event categories were used to calculate the accuracy of phase identification. This is because, for wrongly labeled events, phase identification will not be beneficial. FLCA was able to achieve relatively high accuracy of correctly identifying phases involved during an event. FLCA achieved overall accuracy of 96% for the data window used. FLCA assigned all of the events of high importance (abnormal events on monitored feeder) correct phase labels. Most of the errors in phase identification were associated with events of low importance on non-monitored feeder.

Based on this performance analysis using one week data window, it can be seen that FLCA has good overall classification accuracy. Classification accuracy for most abnormal events was also very good. However, some areas of improvement, such as reducing the confusion between generic arcing events and motor starts were also identified. The percentage of non labeled events can also be reduced. The data window of 1609 events used for this analysis was about 0.81% of all events that FLCA processed in that year. It is difficult to judge the overall performance of the entire system based on the performance of FLCA on a one a relatively small data window. Instead of using FLCA's performance, it would be more appropriate to judge the performance of the system based on what IPSERS ultimately reported to the user. For this reason, classification performance of reporting algorithms will be analyzed next.

### *7.3.2 Classification Performance of Reporting Algorithms*

In contrast to FLCA, reporting algorithms label and report a much reduced subset of events. For example, in the year 2011 reporting algorithms processed 115055 events labeled by FLCA but reported only 1056 events, i.e., IPSERS reported only about 1% of the events that FLCA labeled. This is not surprising because, most of the event labels generated by FLCA correspond to normal system behavior. Reporting algorithms were designed to detect and report a very small subset comprising of abnormal events. As the number of events labeled and reported by reporting algorithms are relatively small, accuracy evaluation of reporting algorithms was done using data from to a one year window. From the user's perspective, classification accuracy of reporting algorithm directly reflect the overall accuracy of IPSERS. This is because; it is the data generated by reporting algorithms that IPSERS presented to the user in a user friendly format. IPSERS did not report individual classification results of FLCA. However, the system does provide a user interface through which users may access classification

Table 7.11: Confusion matrix of capacitor related problems reported to users based on one year of data



| | | | labled | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | Capacitor problems | | | | | | | Non-Capacitor related problems |
| | | | Arc | Excess | Restrike | Unbalanced | VAR Imbalance | Bounce | Failure | |
| Actual | Capacitor problems | Arc | 2 | | | | | | | |
| | | Excess | | 0 | | | | | | |
| | | Restrike | | | 7 | | | | | |
| | | Unbalanced | | | | 40 | | | | |
| | | VAR Imbalance | | | | | 2 | | | |
| | | Bounce | | | | | | 1 | | |
| | | Failure | | | | | | | 1 | |
| | Non-Capacitor related problems | | | | | | | | | 996 |
| Non Event | | | | | | 1 | | | | 6 |

information generated by FLCA for individual waveform files.

Tables 7.11 and 7.12 show classification results of reporting algorithms in the form of a confusion matrix. These results correspond to events that IPSERS reported to users during a one year data window starting January 1, 2011. For simplifying analysis, two confusion matrices are presented. The first one for showing classification performance on capacitor related problems (Table 7.11) and the second for showing classification performance for non-capacitor related problems (Table 7.12). Each column in the confusion matrix represents event type reported

Table 7.12: Confusion matrix of non-capacitor related problems reported to users based on one year of data. Legend: ‘Rec.’ implies reclose, ‘Mis.’ implies miscoordination or conductor slap.

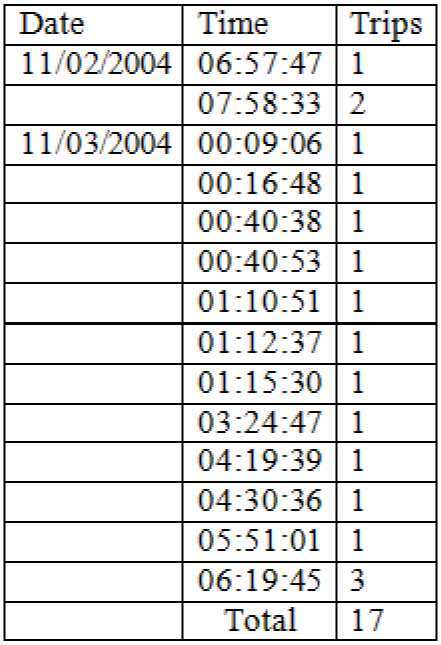
| | | | | labled | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | Overcurrent intrupted by protective device | | | | | | | | | Repetitive overcurrent | | | Other overcurrents | | Other problems | | | Capacitor related problems |
| | | | | Breaker | | | | Single-phase | | Three-phase | | | | | | | | Failing Switch | Failing Cable | Generic Arc | |
| | | | | Rec. | Trip | Close | Mis. | Rec. | Trip | Rec. | Trip | Mis. | 1-ph | 2-ph | 3-Ph | Gen. | Short | | | | |
| Actual | Overcurrent intrupted by protective device | Breaker | Rec. | 38 | | | | | | | | | | | | | | | | | |
| | | | Trip | | 29 | | | | | | | | | | | | | | | | |
| | | | Close | | | 18 | | | | | | | | | | | | | | | |
| | | | Mis. | | | | 11 | | | | | | | | | | | | | | |
| | | Single-phase | Rec. | | | | | 221 | 4 | | | | | | | 1 | | | | | |
| | | | Trip | | | | | 2 | 248 | | | | | | | | | | | | |
| | | Three-phase | Rec. | | | | | | | 69 | | | | | | | | | | | |
| | | | Trip | | | | | | | 1 | 57 | | | | | | | | | | |
| | | | Mis. | | | | | | | | | 3 | | | | | | | | | |
| | Repetitive overcurrent | | 1-ph. | | | | | | | | | | 108 | | | | | | | | |
| | | | 2-ph | | | | | | | | | | | 11 | | | | | | | |
| | | | 3-Ph | | | | | | | | | | | | 2 | | | | | | |
| | Other overcurrents | | Gen. | | | | | | | | | | | | | 64 | | | | | |
| | | | Short | | | | | | | | | | | | | | 95 | | | | |
| | Other problems | Failing Switch | | | | | | | | | | | | | | | | 6 | | | |
| | | Failing Cable | | | | | | | | | | | | | | | | | 7 | | |
| | | Generic Arc | | | | | | | | | | | | | | | | | | 3 | |
| | Capacitor related problems | | | | | | | | | | | | | | | | | | | | 53 |
| | Non Event | | | | | | | | 1 | 2 | 1 | | | | | 1 | | | | 1 | 1 |

IPSERS while each row in the confusion matrix represents the actual class (i.e., the class assigned manually based on visual analysis of waveform data). It should be noted that for many event categories such as but not limited to, repetitive overcurrents and cable failure precursors, it is not possible to determine the event category based on manual analysis alone. On most cases, feedback from utility personnel through field investigation is required. However, only a small fraction of the events that IPSERS reports get investigated. Manual event labels represent the author's best estimate of event cause; for most cases, these do not represent actual results of field investigation. Values along the diagonal represent the number of instances where manual analysis and reporting algorithms agree. Values on off diagonal elements represent the number of instances where reporting algorithms misclassified. The bottom most row manually labeled as 'Non Event' stands for normal events that IPSERS should not have reported. The confusion matrices used to show the overall performance of IPSERS was not color coded. This is because, all events that IPSERS reports to the users are considered important. An equal weight was assigned to all reported events.

The overall accuracy of IPSERS was found to be very good. Out of 1056 events that IPSERS reported, there were only 7 instances where IPSERS reported a normal event as an abnormal event. This is a false positive rate of 0.66%. There were 8 instances where there was some confusion between abnormal categories and all these instances were within sub-categories of overcurrent related events. Thus, there were a total of 15 misclassifications, resulting in an overall high classification accuracy of 98.6%. Table 7.11 shows that there was only a singe instance where IPSERS reported a capacitor related problem as a result of misclassification. This corresponds to classification accuracy of 98.2% for capacitor related problems. The detailed description and parameters reported for capacitor related problems can be found in in Table 7.4. Table 7.12 shows the performance of IPSERS in reporting non capacitor related problems. The detailed description and parameters reported for capacitor related problems can be found in Tables 7.3 and 7.5. There were a total of 6 false alarms where IPSERS reported a normal event as an abnormal non-capacitor related event. This corresponds to a false positive rate of 0.60% for non-capacitor related events. The majority of the false positives were a result of FLCA classifying reclose transients as overcurrents. There was one instance where IPSERS reported an unknown but normal event as generic arcing. There were a total of 8 instances where there was some confusion among non-capacitor related problems. Thus, there were a total of 14 misclassifications, resulting in a overall classification accuracy of 98.6% for non-capacitor related problems. Most of the misclassification were a result of IPSERS either failing to identify a reclose operation or falsely identifying normal variation as a reclose operation. Clearly, improving identification of inrush transients is one area, which would help making IPSERS even more accurate. Another area that would help improve classification accuracy of IPSERS is accurate identification of generic arcing events. The need to improve identification accuracy for generic arcing events was also inferred form the classification results of FLCA.

Good accuracy rate on reported events is necessary to gain the confidence of utility personnel. This would help DFA-IPSERS to be integrated into day to day operations. A few reported abnormal events such as excessive capacitor operations, failing switch or clamp, conductor slap and some repetitive overcurrents have triggered an investigation or corrective actions. These instances are proof to both the potential usefulness and the success of DFA-IPSERS. Next section provides real world examples where utility companies either benefited or could have benefited from IPSERS.

Table 7.13: List of individual interruptions



| Date | Time | Trips |
|---|---|---|
| 11/02/2004 | 06:57:47 | 1 |
|  | 07:58:33 | 2 |
| 11/03/2004 | 00:09:06 | 1 |
|  | 00:16:48 | 1 |
|  | 00:40:38 | 1 |
|  | 00:40:53 | 1 |
|  | 01:10:51 | 1 |
|  | 01:12:37 | 1 |
|  | 01:15:30 | 1 |
|  | 03:24:47 | 1 |
|  | 04:19:39 | 1 |
|  | 04:30:36 | 1 |
|  | 05:51:01 | 1 |
|  | 06:19:45 | 3 |
|  | Total | 17 |

## 7.4 DFA-IPSERS Case Studies

The following case studies (organized in a chronological order) provide examples of a variety of failures and failure precursors. Utilities involved in these cases had no other means to inform them that underlying failures were developing, or to help them fix problems after they occurred, other than the classification and reporting provided through the DFA-IPSERS. DFA-IPSERS used only signals acquired at the substation for reporting and classification.

### *7.4.1 Case Study 1: Tree Limb Burns Down Line, Causing Outage*

In November 2004, repetitive overcurrent faults caused a three-phase pole-top recloser on a feeder at one of Pickwick Electric Cooperative's substation to trip multiple times. Since the faults did not persist, none of them caused the recloser to lockout until the recloser had tripped 17 times over a period of two days before

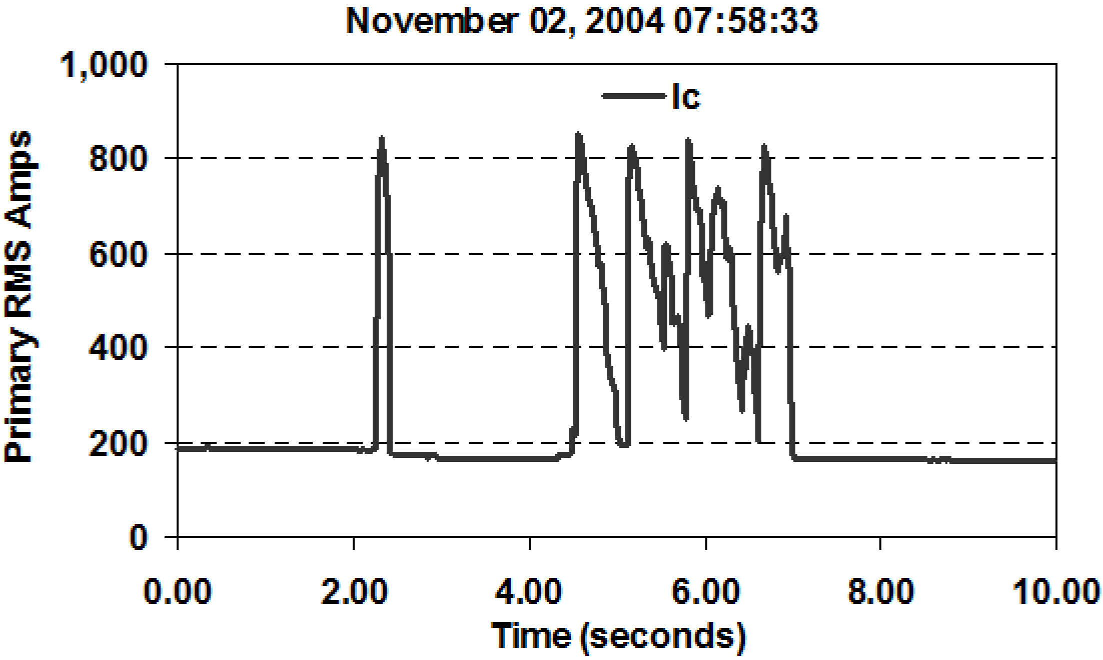


Figure 7.5: A second fault tripped recloser twice without causing a lock out

finally locking out. Table 7.13 lists the individual interruptions that began at 6:57 AM on November 2 and continued till 6:19 AM on November 3. Figure 7.5 shows the RMS phase current the DFA monitoring unit measured at the substation during one of the initial episodes that caused the recloser to trip twice. Figure 7.6 shows the final episode of overcurrent where the overcurrent became persistent causing the recloser to lock out. Investigation revealed a broken tree limb that had burned down a span of line putting 140 customers out of service for 62 minutes while the crew repaired the line. The tree limb also explained the intermittent nature of the fault. The crew found that a fork in the broken tree limb had hung on the phase conductor. The limb pulled the phase conductor down to within about two feet of the neutral conductor. The fork was in continuous contact with the phase conductor. Contact with the neutral occurred a few feet farther along the limb, causing the intermittent faults.

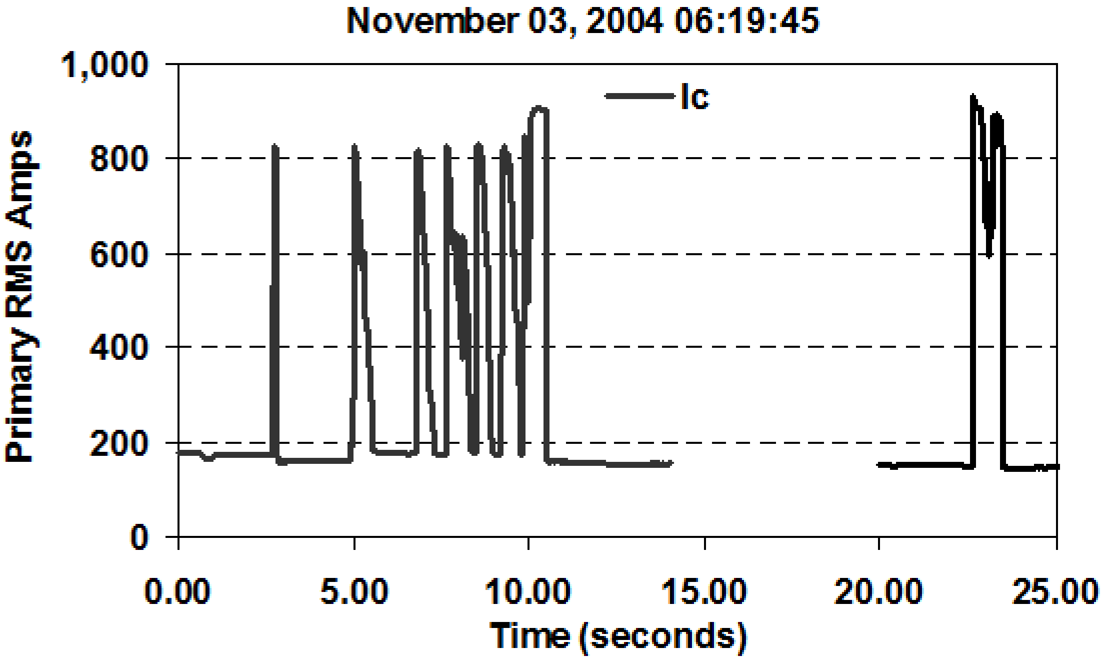
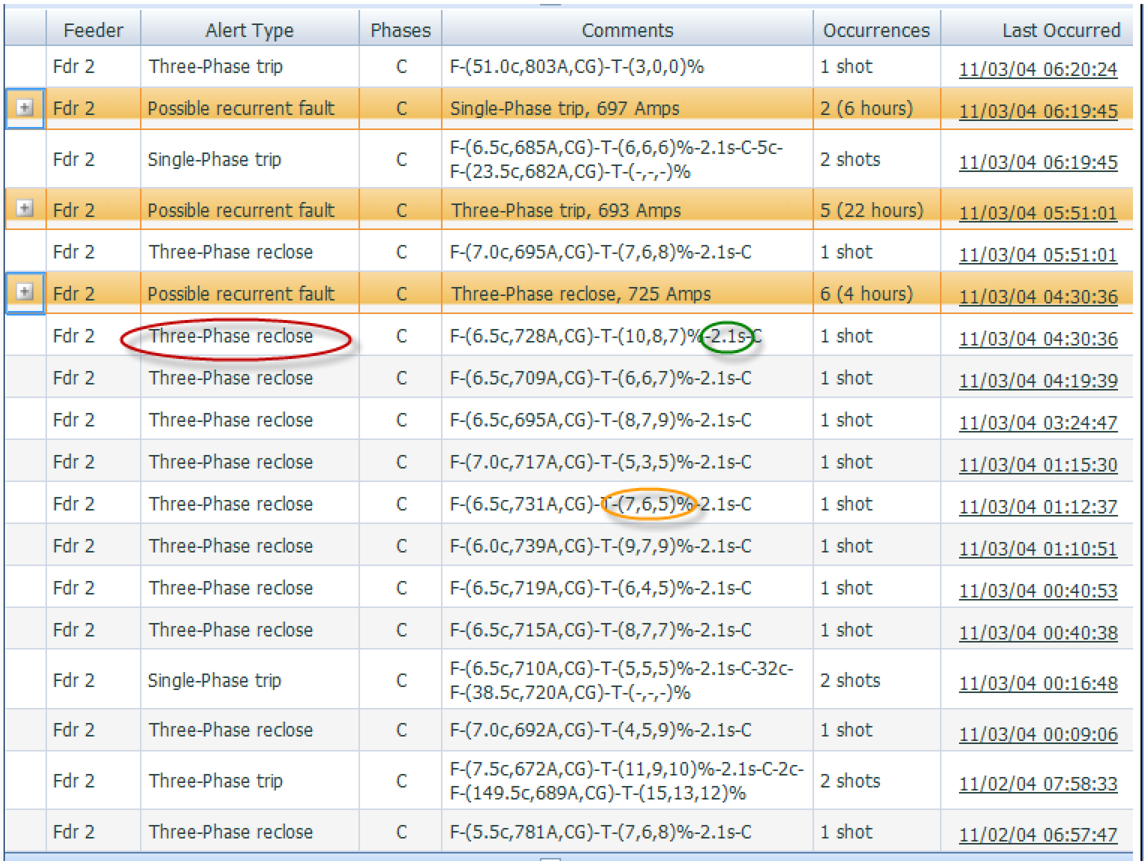


Figure 7.6: Current waveforms from final instance of fault that caused the recloser to lock out

Pickwick participates in the DFA project sponsored by EPRI. A DFA Prototype at the substation recorded each fault as they happened. Because a pole-top recloser operated, instead of the substation breaker, Pickwick had no indication of a problem until the lights-out calls from customers. Moreover, in 2004, automatic reporting through DFA-IPSERS was not available. Figure 7.7 shows alert items IPSERS generated after processing of waveform files recorded during the recurrent fault episodes. All the faults that were manually identified in Table 7.13 have also been identified automatically by DFA-IPSERS. From the automatic alerts generated by DFA-IPSERS the following observations can be made:

1. On a majority of the faults, the interrupting device was also correctly identified as being a three-phase automatic recloser. For each instance, IPSERS correctly computed a reclose interval of 2.1 seconds (circled in green) and



| | Feeder | Alert Type | Phases | Comments | Occurrences | Last Occurred |
|---|---|---|---|---|---|---|
| | Fdr 2 | Three-Phase trip | C | F-(51.0c,803A,CG)-T-(3,0,0)% | 1 shot | 11/03/04 06:20:24 |
| ⊞ | Fdr 2 | Possible recurrent fault | C | Single-Phase trip, 697 Amps | 2 (6 hours) | 11/03/04 06:19:45 |
| | Fdr 2 | Single-Phase trip | C | F-(6.5c,685A,CG)-T-(6,6,6)%-2.1s-C-5c-F-(23.5c,682A,CG)-T-(-,-,-)% | 2 shots | 11/03/04 06:19:45 |
| ⊞ | Fdr 2 | Possible recurrent fault | C | Three-Phase trip, 693 Amps | 5 (22 hours) | 11/03/04 05:51:01 |
| | Fdr 2 | Three-Phase reclose | C | F-(7.0c,695A,CG)-T-(7,6,8)%-2.1s-C | 1 shot | 11/03/04 05:51:01 |
| ⊞ | Fdr 2 | Possible recurrent fault | C | Three-Phase reclose, 725 Amps | 6 (4 hours) | 11/03/04 04:30:36 |
| | Fdr 2 | Three-Phase reclose | C | F-(6.5c,728A,CG)-T-(10,8,7)%-2.1s-C | 1 shot | 11/03/04 04:30:36 |
| | Fdr 2 | Three-Phase reclose | C | F-(6.5c,709A,CG)-T-(6,6,7)%-2.1s-C | 1 shot | 11/03/04 04:19:39 |
| | Fdr 2 | Three-Phase reclose | C | F-(6.5c,695A,CG)-T-(8,7,9)%-2.1s-C | 1 shot | 11/03/04 03:24:47 |
| | Fdr 2 | Three-Phase reclose | C | F-(7.0c,717A,CG)-T-(5,3,5)%-2.1s-C | 1 shot | 11/03/04 01:15:30 |
| | Fdr 2 | Three-Phase reclose | C | F-(6.5c,731A,CG)-T-(7,6,5)%-2.1s-C | 1 shot | 11/03/04 01:12:37 |
| | Fdr 2 | Three-Phase reclose | C | F-(6.0c,739A,CG)-T-(9,7,9)%-2.1s-C | 1 shot | 11/03/04 01:10:51 |
| | Fdr 2 | Three-Phase reclose | C | F-(6.5c,719A,CG)-T-(6,4,5)%-2.1s-C | 1 shot | 11/03/04 00:40:53 |
| | Fdr 2 | Three-Phase reclose | C | F-(6.5c,715A,CG)-T-(8,7,7)%-2.1s-C | 1 shot | 11/03/04 00:40:38 |
| | Fdr 2 | Single-Phase trip | C | F-(6.5c,710A,CG)-T-(5,5,5)%-2.1s-C-32c-F-(38.5c,720A,CG)-T-(-,-,-)% | 2 shots | 11/03/04 00:16:48 |
| | Fdr 2 | Three-Phase reclose | C | F-(7.0c,692A,CG)-T-(4,5,9)%-2.1s-C | 1 shot | 11/03/04 00:09:06 |
| | Fdr 2 | Three-Phase trip | C | F-(7.5c,672A,CG)-T-(11,9,10)%-2.1s-C-2c-F-(149.5c,689A,CG)-T-(15,13,12)% | 2 shots | 11/02/04 07:58:33 |
| | Fdr 2 | Three-Phase reclose | C | F-(5.5c,781A,CG)-T-(7,6,8)%-2.1s-C | 1 shot | 11/02/04 06:57:47 |

Figure 7.7: Screen capture of automatic alert items generated through offline processing of recorded waveform files corresponding to overcurrent episodes in case study 1

reported an estimated load loss (circled in orange).

2. On all the faults, IPSERS correctly identified the faulted phase as C. IPSERS also reported correct estimates for fault currents and fault durations.

3. The similarity of the individual fault episodes also caused the reporting algorithms to cluster and generate recurrent fault related alerts (highlighted in orange). A majority (13 out of 14) faults became a part of some recurrent fault cluster. Ideally, IPSERS should have clustered all the faults into one recurrent fault cluster. Because of the nonlinear and dynamic nature of the phenomenon that caused these faults, some variations in fault magnitude and duration were observed across individual episodes. This resulted in more than one cluster. However, a recurrent fault alert with five or six similar faults within a few hours would easily draw the attention utility personnel.

IPSERS would have presented the above information to utility personnel in a near real-time fashion had DFA-IPSERS been active in 2004. However, the first version of DFA-IPSERS became active only in 2008. If real-time reporting had been available to operations personnel, Pickwick confirmed that they would have dispatched a crew after the first few faults around 1:00 AM on November 3. Using information reported by DFA framework, Pickwick personnel believe that they would have located the source of the problem within a few hours. They would have also had time to take remedial action, and potentially avoided the line burn down and the outage [115].

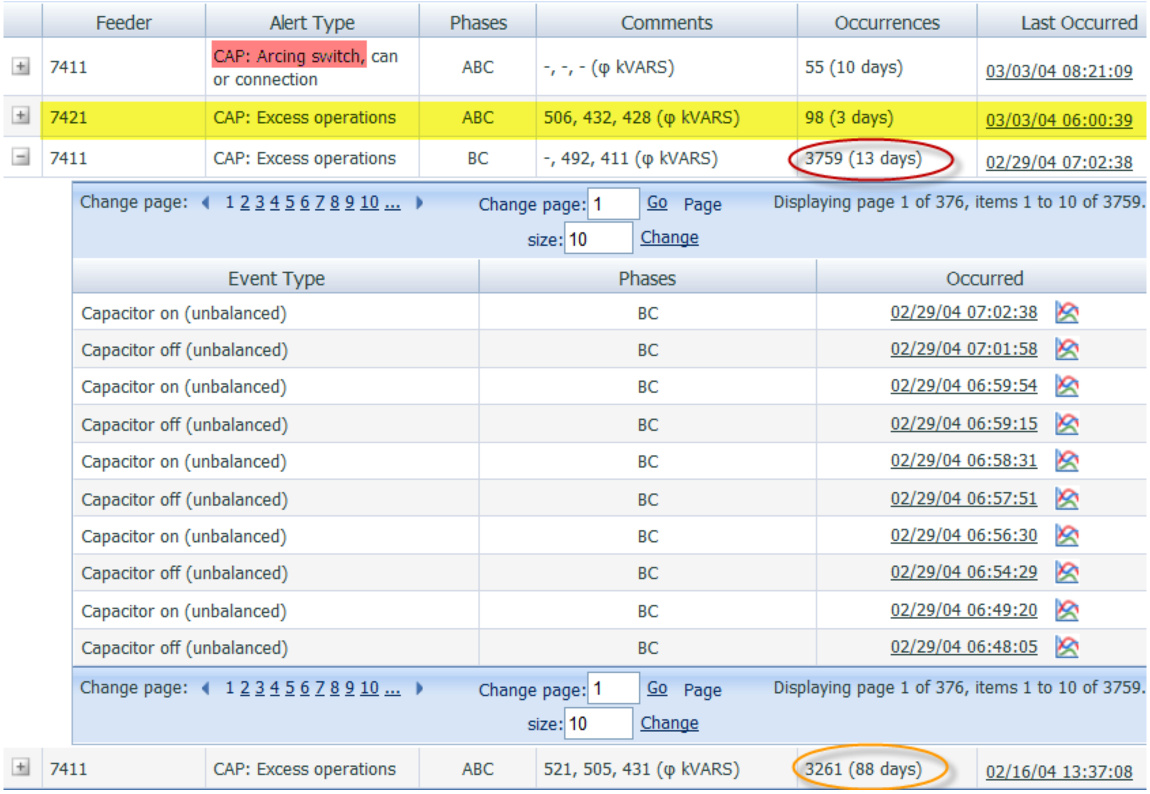
### *7.4.2 Case Study 2: Misoperating Capacitor Controllers*

There are instances when a problem can be introduced as a result of routine maintenance. One such example happened on August 9, 2004. Pickwick Electric Cooperative performed annual maintenance on some of its capacitor banks on feeders monitored by DFA prototype units. This included testing the banks and replacing their controllers. At 13:21 on the same day, one bank began switching on and off repetitively. By the end of the day, it had cycled 22 times. DFA monitoring unit was recording the capacitor switching operations. A prototype FLCA algorithm was classifying the waveforms automatically. However, automatic reporting through DFA-IPSERS was not available then. Shortly after the capacitor began cycling, TAMU personnel notified Pickwick and Pickwick dispatched a crew to fix the problematic bank. The crew made a simple change to the bank's controller settings and corrected the problem less than 24 hours after it began. Without the information provided through DFA framework, the bank would have continued to operate excessively for an indefinite time until the next maintenance cycle or until it caused a problem. It is difficult to predict the effects of prolonged misoperation if Pickwick had allowed the condition to persist.

Another utility company, TXU Electric Delivery also participates in the DFA project and has a DFA prototype monitoring unit installed at one of its substations. Unlike Pickwick, one of TXU's primary objectives in the DFA project was to document the consequences of faults and other anomalous behavior without taking remedial action based on information provided by DFA framework. This would enable them and others to quantify the benefits of anticipating and preventing these problems, and create a valuable body of information that did not exist previously. In January 2004, a capacitor bank on one of TXU's feeders monitored by

DFA prototype unit began to misoperate. During early January, it switched on and off an average of 28 times per day, similar to the 22 operations Pickwick experienced the day their problem began. Over the next month, the problem worsened. By the middle of February 2004, the bank had switched more than 3,000 times in a period of less than two months. On February 16, the bank's phase-A capacitor finally failed, and the capacitor bank started to operate under an unbalanced condition. The average number of daily operations increased further. On February 29 the problem further worsened. The internal contacts of the bank's phase-B oil switch failed to make a good connection. This was because of the excessive wear and tear they had accumulated during the past two months. This poor connection caused contact arcing and severe voltage transients. However, TXU received no reports of customer problems during this period. For the next four days, the controller continued to switch the bank on and off many times. Whenever the capacitor switch closed, the switch arced internally and caused repetitive transients on all feeders connected to the affected bus. On March 4, the phase B capacitor finally failed in an open-circuit condition, removing the problem from the system. The entire sequence and the field findings after the entire sequence of events was documented in detail [116].

DFA-IPSERS was not available during the period described above. However, to show the effectiveness of DFA-IPSERS, IPSERS was used to process the data recorded during the entire sequence offline. Figure7.8 shows the resulting automatic alerts that IPSERS generated . The entire sequence described above can also be seen in the alert items automatically generated by IPSERS. The bottom most line item reports the excessive operations till February 16 (circled in orange). The alert line item also shows that all the three phases switched (under phases column). IPSERS also estimated capacitor bank size information and displayed it



| | Feeder | Alert Type | Phases | Comments | Occurrences | Last Occurred |
|---|---|---|---|---|---|---|
| ⊞ | 7411 | CAP: Arcing switch, can or connection | ABC | -, -, - (φ kVARS) | 55 (10 days) | 03/03/04 08:21:09 |
| ⊞ | 7421 | CAP: Excess operations | ABC | 506, 432, 428 (φ kVARS) | 98 (3 days) | 03/03/04 06:00:39 |
| ⊟ | 7411 | CAP: Excess operations | BC | -, 492, 411 (φ kVARS) | 3759 (13 days) | 02/29/04 07:02:38 |

Change page: ◂ 1 2 3 4 5 6 7 8 9 10 ... ▸ Change page: 1 Go Page size: 10 Change Displaying page 1 of 376, items 1 to 10 of 3759.

| Event Type | Phases | Occurred |
|---|---|---|
| Capacitor on (unbalanced) | BC | 02/29/04 07:02:38 |
| Capacitor off (unbalanced) | BC | 02/29/04 07:01:58 |
| Capacitor on (unbalanced) | BC | 02/29/04 06:59:54 |
| Capacitor off (unbalanced) | BC | 02/29/04 06:59:15 |
| Capacitor on (unbalanced) | BC | 02/29/04 06:58:31 |
| Capacitor off (unbalanced) | BC | 02/29/04 06:57:51 |
| Capacitor on (unbalanced) | BC | 02/29/04 06:56:30 |
| Capacitor off (unbalanced) | BC | 02/29/04 06:54:29 |
| Capacitor on (unbalanced) | BC | 02/29/04 06:49:20 |
| Capacitor off (unbalanced) | BC | 02/29/04 06:48:05 |

Change page: ◂ 1 2 3 4 5 6 7 8 9 10 ... ▸ Change page: 1 Go Page size: 10 Change Displaying page 1 of 376, items 1 to 10 of 3759.

| | Feeder | Alert Type | Phases | Comments | Occurrences | Last Occurred |
|---|---|---|---|---|---|---|
| ⊞ | 7411 | CAP: Excess operations | ABC | 521, 505, 431 (φ kVARS) | 3261 (88 days) | 02/16/04 13:37:08 |

Figure 7.8: Screen capture of automatic alert items generated through offline processing of recorded waveform files corresponding to misoperating capacitor banks in case study 2

under the comments column. On February 16th, the phase A bank failed. Hence, IPSERS marked the individual operations that followed as unbalanced (shown as an expanded list) but still clustered them under excessive operations (circled in red). The excessive operations also possibly caused another capacitor bank on an adjacent feeder to misbehave (highlighted in yellow). The capacitor switch contact started to degrade rapidly due to excessive operations. Hence, they started to arc, and this caused the arcing switch related alert item (highlighted in red) to be generated. TXU Electric Delivery did not take remedial actions for the purpose of documenting the failure. Hence, it did not take advantage of the information provided by DFA framework. If DFA-IPSERS were integrated into daily operations of a utility, then DFA-IPSERS would have made the same information available to utility personnel in a near real-time fashion. IPSERS would have reported thousands of operations well before the first failure on February 16th. This would have most likely made utility personnel to take remedial actions. Such remedial actions would have avoided the sequence of events that ultimately lead to the failure of capacitor bank on March 4.

### *7.4.3 Case Study 3: Prevented Outage Caused by Failing External Transformer Bushing*

On December 11, 2005 07:39:58 a DFA monitoring unit recorded a fault of approximately 2,400 RMS amps (Figure 7.9) . The fault had lasted about two cycles, before a single-phase poletop recloser tripped then reclosed two seconds later. The fault did not persist after the reclose and hence there were no outages or customer complaints. The utility company was unaware that anything had happened, except for automatic classifications provided by FLCA through from the DFA framework.

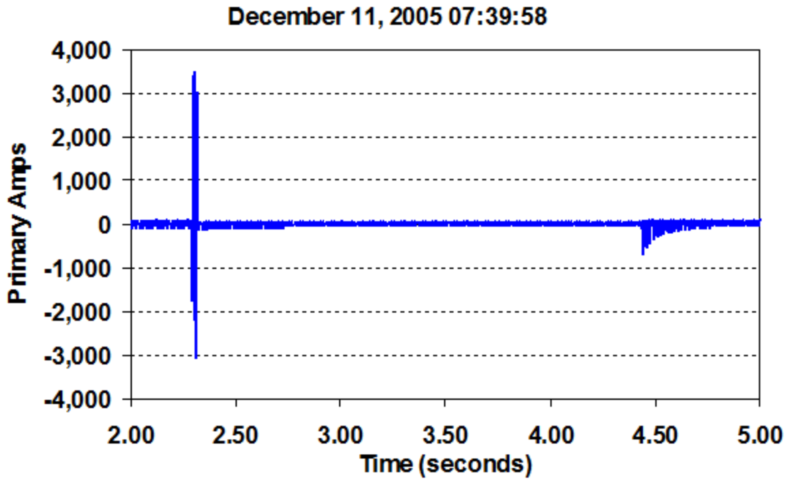


Figure 7.9: Single phase fault cleared by a single phase poletop recloser

Another fault occurred two days later on December 13, 08:21:05. It was also on the same phase and approximately, produced the same current level as the first fault. The same single-phase recloser operated after two cycles and reclosed two seconds later. Similar to the previous occurrence, there were no outages or customer complaints, and the utility had no indication of the fault, except for information from the DFA. Alerted to the two nearly identical faults classified by FLCA, utility personnel searched for the cause the next day. A two-man crew found the problem in less than one hour. They found a poletop service transformer with a damaged bushing possibly caused by a squirrel. The conclusion was that the animal caused the first short circuit (December 11). The poletop recloser cleared the fault properly. However, the short circuit arc across the transformer bushing caused permanent damage to the bushing and resulted in the later fault on December 13.

After finding the damaged transformer bushing, the utility put replacement of the transformer on its work list. However, before the repair could be made, the fault recurred on December 18, prompting the utility to expedite the repair. The utility knew of this third fault too through classification information provided by DFA. The fault did not recur after the utility made the replacement. This entire sequence and the remedial action taken by the utility were documented [76, 117]. This case illustrates several ways in which the utility benefited from both the DFA framework, and the classification results provided by FLCA in the following ways:

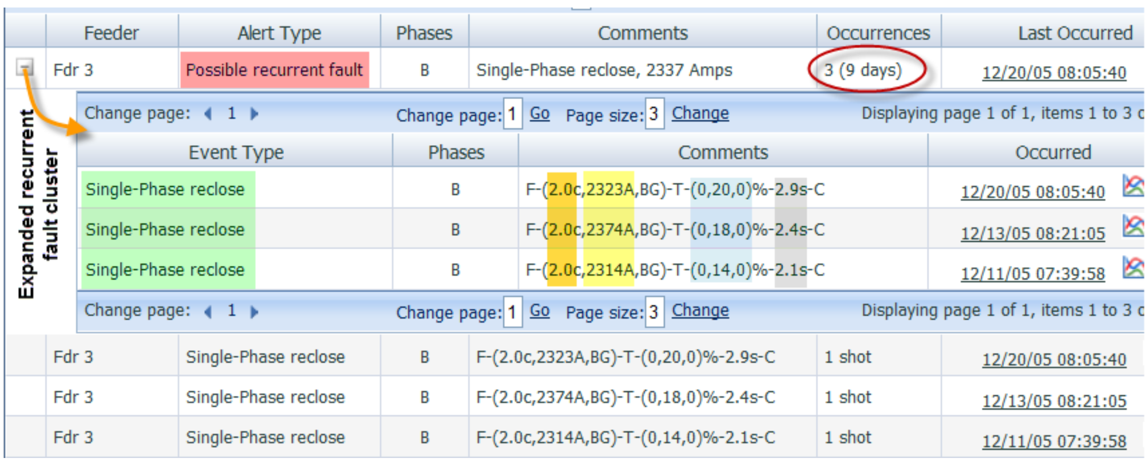
1. Notification that a problem was developing: Without the classification results provided by DFA framework, the utility had no other indication that a problem existed. No other device indicated a problem and there were no customer complaints.

2. Locating the problem: Locating such a fault is challenging, especially when it is intermittent fault that does not cause customer calls. The utility's feeder was long and geographically dispersed. This made the problem even more difficult. However, the utility was able to combine the fault current characteristics with the utility's fault current map and known recloser placements to locate the problem with a two-man crew in less than one hour.

3. Avoided customer complaints: The utility detected and repaired an incipient failure, avoiding events that could have caused customer complaints.

4. Prioritizing utility action: Notification of the third fault through the DFA framework caused the utility to take expedited action to replace the damaged transformer before any further damage could happen.

When this case was documented, automatic reporting through DFA-IPSERS was not available. Only automatic classification results of FLCA were available through a graphical user interface, which is part of DFA framework. The utility user still



Expanded recurrent fault cluster

| Feeder | Alert Type | Phases | Comments | Occurrences | Last Occurred |
|---|---|---|---|---|---|
| Fdr 3 | Possible recurrent fault | B | Single-Phase reclose, 2337 Amps | 3 (9 days) | 12/20/05 08:05:40 |

Change page: ◀ 1 ▶ Change page: 1 Go Page size: 3 Change Displaying page 1 of 1, items 1 to 3 c

| Event Type | Phases | Comments | Occurred |
|---|---|---|---|
| Single-Phase reclose | B | F-(2.0c,2323A,BG)-T-(0,20,0)%-2.9s-C | 12/20/05 08:05:40 |
| Single-Phase reclose | B | F-(2.0c,2374A,BG)-T-(0,18,0)%-2.4s-C | 12/13/05 08:21:05 |
| Single-Phase reclose | B | F-(2.0c,2314A,BG)-T-(0,14,0)%-2.1s-C | 12/11/05 07:39:58 |

Change page: ◀ 1 ▶ Change page: 1 Go Page size: 3 Change Displaying page 1 of 1, items 1 to 3 c

| Feeder | Alert Type | Phases | Comments | Occurrences | Last Occurred |
|---|---|---|---|---|---|
| Fdr 3 | Single-Phase reclose | B | F-(2.0c,2323A,BG)-T-(0,20,0)%-2.9s-C | 1 shot | 12/20/05 08:05:40 |
| Fdr 3 | Single-Phase reclose | B | F-(2.0c,2374A,BG)-T-(0,18,0)%-2.4s-C | 1 shot | 12/13/05 08:21:05 |
| Fdr 3 | Single-Phase reclose | B | F-(2.0c,2314A,BG)-T-(0,14,0)%-2.1s-C | 1 shot | 12/11/05 07:39:58 |

Figure 7.10: Screen capture of automatic alert items generated through offline processing of recorded waveform files corresponding to repetitive overcurrent episodes documented in case study 3

had to open and view individual waveform files that FLCA had marked as overcurrents, and check if they were recurrent overcurrents. Other information such as overcurrent magnitude, duration and protective device information had to be inferred through manual analysis of the waveforms. Automatic reporting through DFA-IPSERS automates this whole process and makes it both efficient and less error prone. Utility users can now quickly access all the required information either through a web interface or email. IPSERS was made to process waveform files recorded during the recurrent fault episodes. Figure 7.10 shows the resulting automatic alerts that IPSERS generated.. The first line item in Figure 7.10 corresponds to a recurrent fault cluster (highlighted in red). Reporting algorithms automatically clustered the three overcurrents that happened over a period of 9 days and presented them as an alert to draw the attention of utility users. The recurrent fault cluster can be expanded to view the individual faults. IPSERS clearly identified the operated device as a single phase recloser (highlighted in green). IPSERS also estimated and reported fault current magnitudes (highlighted in yellow) and fault durations (highlighted in orange). IPSERS reported other useful information such as load loss (highlighted in blue) and reclose interval (highlighted in gray) too. Equipped with this information, utility personnel can locate the fault in a timely fashion.

This is another example that shows how classification and reporting algorithms developed as a part of this research can be effective tools for utility personnel. The next three case studies presents examples where DFA-IPSERS was online and helped utility companies to prevent or diagnose problems on their feeders.

### *7.4.4 Case Study 4: Prevented Vegetation Related Outage*

Pickwick Electric Cooperative (PEC) used DFA-IPSERS to avoid an impending vegetation related outage caused by tree encroachment in an overhead line. DFA-IPSERS provided information that enabled PEC to find and remove the encroachment before it caused an outage. Other than the information provided by DFA-IPSERS, PEC had no other indication that a problem existed.

During a rainy period in July 2010, the DFA-IPSERS reported four faults that resulted in momentary operation of a single-phase line recloser. Figure 7.11 shows RMS waveforms of fault current recorded during the four faults. Reporting algorithms also clustered the four individual faults, due to their similar nature. DFA-IPSERS presented them as recurrent fault alert to utility users at PEC. Figure 7.12 shows a screen capture of the alert items reported to users at PEC. No customers experienced a sustained outage, due to momentary nature of the faults. Hence, no one reported the interruptions to PEC. PEC located the line recloser at a remote point on the feeder. The line recloser did not have the capability of reporting its operations either. As a result, PEC had no indication of the ongoing problem other than what DFA-IPSERS reported.

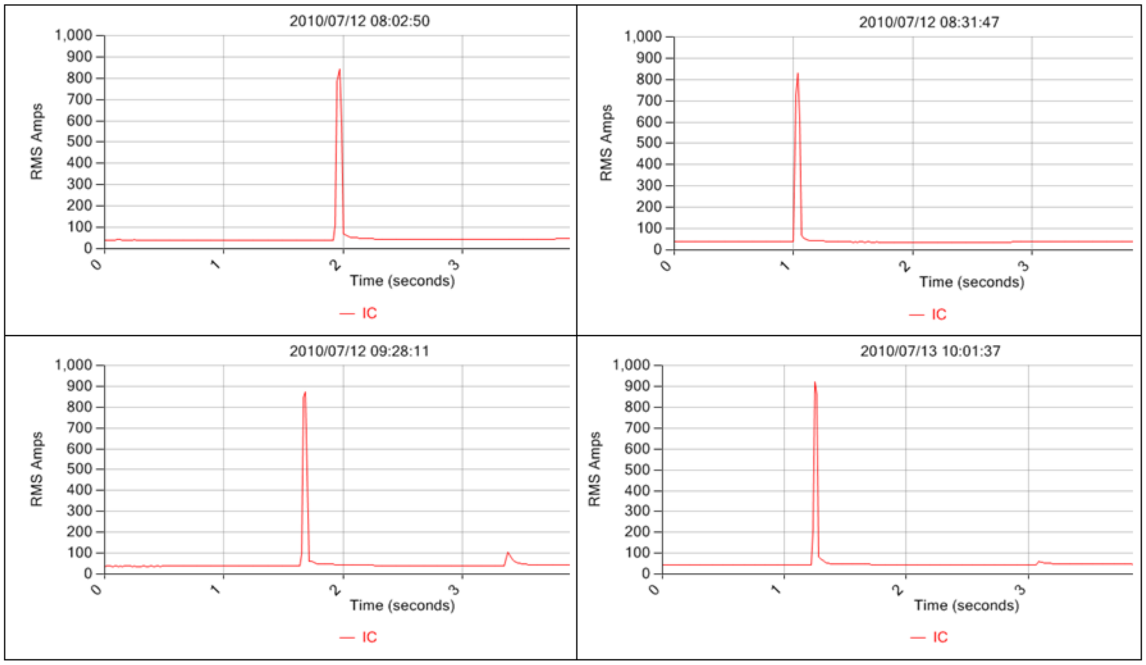
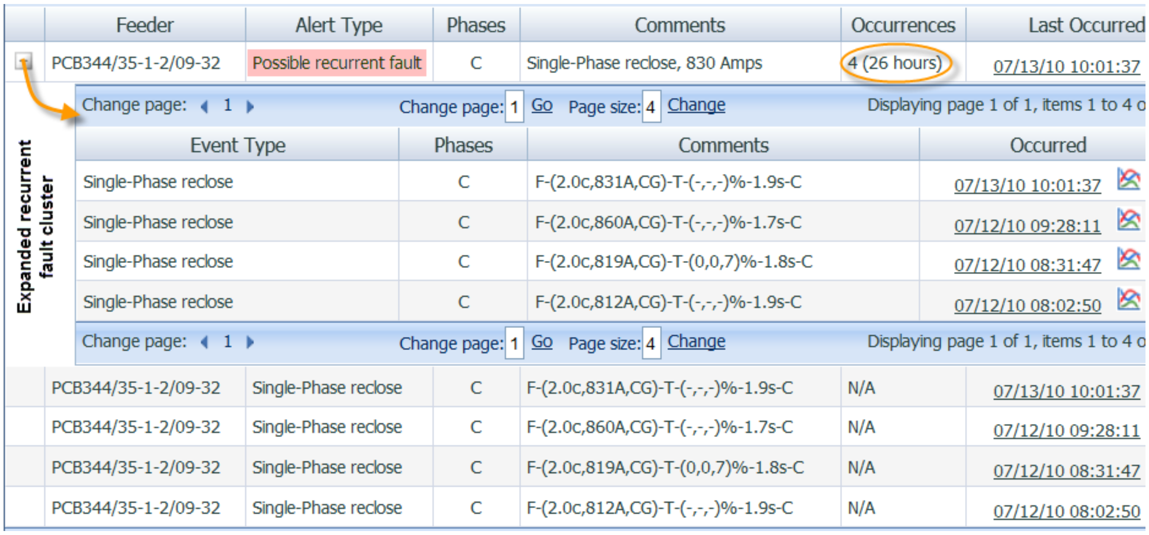


Figure 7.11: RMS waveforms of fault currents recorded during repetitive faults



| Feeder | Alert Type | Phases | Comments | Occurrences | Last Occurred |
|---|---|---|---|---|---|
| PCB344/35-1-2/09-32 | Possible recurrent fault | C | Single-Phase reclose, 830 Amps | 4 (26 hours) | 07/13/10 10:01:37 |

Expanded recurrent fault cluster

Change page: ◂ 1 ▸ Change page: 1 Go Page size: 4 Change Displaying page 1 of 1, items 1 to 4 o

| Event Type | Phases | Comments | Occurred |
|---|---|---|---|
| Single-Phase reclose | C | F-(2.0c,831A,CG)-T-(-,-,-)%-1.9s-C | 07/13/10 10:01:37 |
| Single-Phase reclose | C | F-(2.0c,860A,CG)-T-(-,-,-)%-1.7s-C | 07/12/10 09:28:11 |
| Single-Phase reclose | C | F-(2.0c,819A,CG)-T-(0,0,7)%-1.8s-C | 07/12/10 08:31:47 |
| Single-Phase reclose | C | F-(2.0c,812A,CG)-T-(-,-,-)%-1.9s-C | 07/12/10 08:02:50 |

Change page: ◂ 1 ▸ Change page: 1 Go Page size: 4 Change Displaying page 1 of 1, items 1 to 4 o

| Feeder | Alert Type | Phases | Comments | Occurrences | Last Occurred |
|---|---|---|---|---|---|
| PCB344/35-1-2/09-32 | Single-Phase reclose | C | F-(2.0c,831A,CG)-T-(-,-,-)%-1.9s-C | N/A | 07/13/10 10:01:37 |
| PCB344/35-1-2/09-32 | Single-Phase reclose | C | F-(2.0c,860A,CG)-T-(-,-,-)%-1.7s-C | N/A | 07/12/10 09:28:11 |
| PCB344/35-1-2/09-32 | Single-Phase reclose | C | F-(2.0c,819A,CG)-T-(0,0,7)%-1.8s-C | N/A | 07/12/10 08:31:47 |
| PCB344/35-1-2/09-32 | Single-Phase reclose | C | F-(2.0c,812A,CG)-T-(-,-,-)%-1.9s-C | N/A | 07/12/10 08:02:50 |

Figure 7.12: Screen capture of automatic alert items reported to users

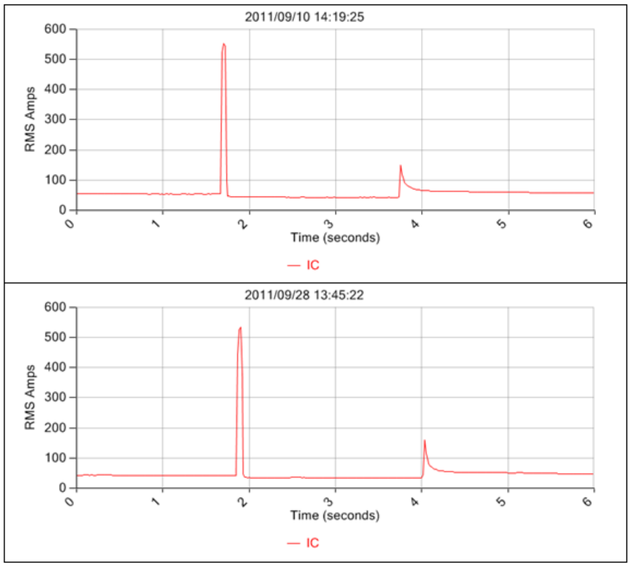


Figure 7.13: RMS waveforms of fault currents from the two identical faults

Alerts generated by DFA-IPSERS (Figure 7.12) provided the characteristics of individual fault such as fault magnitude and duration, plus valuable protective device information that enabled PEC to narrow the search area and find the encroachment a few hours after the fourth fault. Targeted tree trimming resolved the problem, without any customer complaints. This case was documented in [118].

PEC has been involved with DFA project for several years. This is one of many instances where PEC has benefited from using automatic reporting through the DFA framework.

### *7.4.5 Case Study 5: Avoided Outage by Detecting and Locating Incipient Failure*

On September 28, 2011, one one of the feeders monitored by DFA, DFA-IPSERS reported a recurrent fault that included two identical faults that happened 18 days apart. Figure 7.13 shows RMS fault currents recorded during these faults. Fault location on this circuit was challenging because, the circuit had 139 circuit miles. Figure 7.14 shows the circuit map of the affected feeder. The utility had no knowl-

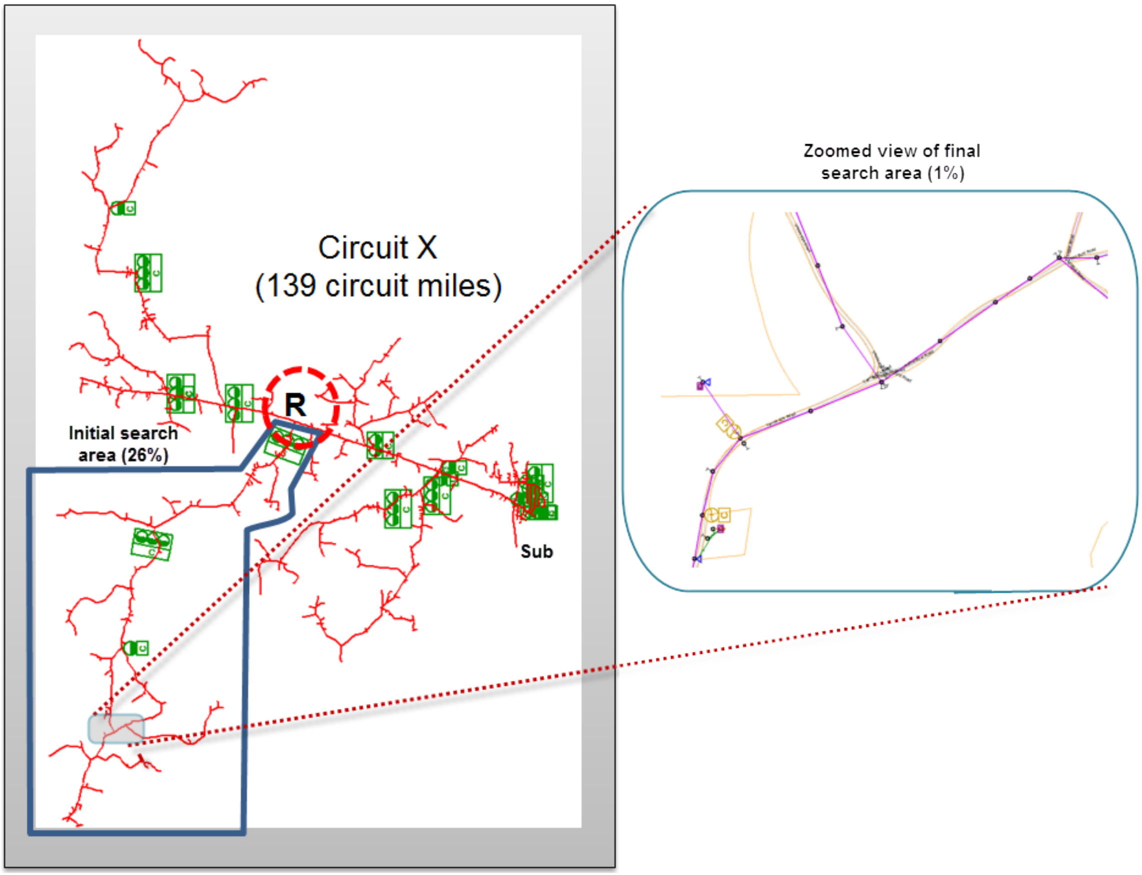
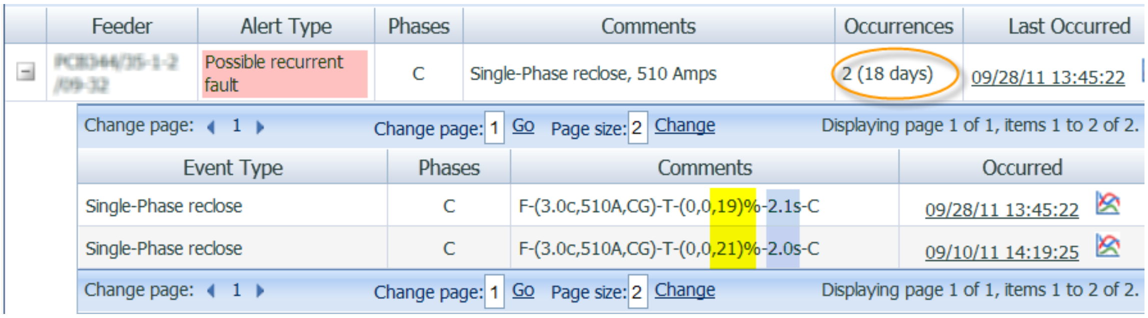


Figure 7.14: Circuit map showing the affected feeder that had 139 miles of exposure



|  | Feeder | Alert Type | Phases | Comments | Occurrences | Last Occurred |
|---|---|---|---|---|---|---|
| ⊟ | [redacted] | Possible recurrent fault | C | Single-Phase reclose, 510 Amps | 2 (18 days) | 09/28/11 13:45:22 |

Change page: ◀ 1 ▶ Change page: 1 Go Page size: 2 Change Displaying page 1 of 1, items 1 to 2 of 2.

| Event Type | Phases | Comments | Occurred |
|---|---|---|---|
| Single-Phase reclose | C | F-(3.0c,510A,CG)-T-(0,0,19)%-2.1s-C | 09/28/11 13:45:22 |
| Single-Phase reclose | C | F-(3.0c,510A,CG)-T-(0,0,21)%-2.0s-C | 09/10/11 14:19:25 |

Change page: ◀ 1 ▶ Change page: 1 Go Page size: 2 Change Displaying page 1 of 1, items 1 to 2 of 2.

Figure 7.15: Screen capture of recurrent fault reported to users

edge of these faults other than what DFA-IPSERS reported. Figure 7.15 shows a screen capture of the recurrent fault reported to utility users. The recurrent fault alert generated by DFA-IPSERS clearly showed the following information:

1. The faults were located downstream of a single phase recloser protecting Phase C.

2. There was an estimated load interruption of 19-20%.

3. The recloser had a 2 second open interval.

4. The fault current estimate was 510Amps.

Utility users used the above information and compared it to the system model. As a result, they narrowed down the search are to an area a bank of single phase reclosers (marked as **R** in Figure 7.14) were protecting. Utility personnel chose this recloser bank because, the system model indicated that the reclosers had a two second open interval, and a 23% load beyond them. They concluded that the location of the problem that caused the recurrent faults was downstream of the identified recloser bank (Figure 7.14, highlighted with blue border). This narrowed down the search area by 74%.

Utility personnel initially patrolled downstream of recloser bank **R** and found cracked dead-end bells close to the recloser bank, but their circuit model estimated

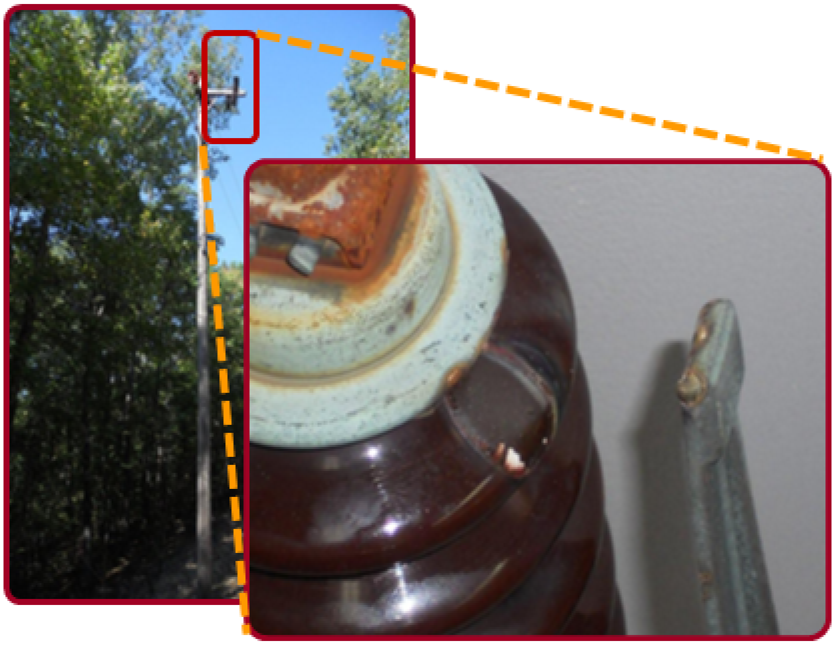


Figure 7.16: Picture of failing arrester

a fault current of 1086Amps at this location and hence concluded that the cracked dead-end bells were not the cause of recurrent faults. Then, the utility personnel used the fault current estimates reported by DFA-IPSERS and targeted an area that matched this area, based on their circuit model. This drastically reduced the search area to about 1% of total circuit miles (Figure 7.14, highlighted by an oval and zoomed). This search area corresponded to 4 spans on either side of a tee. After searching this area, utility personnel found a failing lightning arrester (Figure 7.16). They replaced the failing arrester and possibly avoided further interruptions and outage to 53 customers.

This case study showed an example where automatic alerts generated by DFA-IPSERS and the system model, together helped detect an incipient failure and avoided a possible outage.

### *7.4.6 Case Study 6: Detected and Helped Locate Fault-induced Conductor Slap*

Fault-induced conductor slap, or clash, occurs when electromagnetic forces resulting from fault currents cause sufficient movement of conductors to cause them to touch one another, or come close enough for flashover [119]. One typical scenario involves a phase-to-phase fault at some point on an overhead feeder. The fault may cause substantially large and equal fault currents to flow in opposite directions, in two parallel phase conductors. The fault currents will flow over the entire length between the substation and the fault. Such parallel conductors carrying currents in opposite directions experience electromagnetic forces that push them away from each other. Significant fault currents can cause the conductors to move a substantial distance away from each other. On the return swing, conductors may contact each other, or come close enough to cause flashover.

There are several scenarios in which a conductor slap may cause either a midpoint recloser located between the initial fault and the substation, or the circuit breaker, to operate[119, 76]. Conductor slaps that result in the breaker operation are especially serious because, they have the potential to lockout the circuit breaker. Breaker lockouts could cause a larger number of customers to experience an outage than the outage the initial fault would have caused. Figure 7.17 shows an one-line diagram of such typical scenario. Multiple utilities have used DFA-IPSERS to identify and locate conductor-slap events. If, left uncorrected, conductor-slaps may recur and continue to cause unnecessary breaker or recloser operations. This will not only degrade system components but also cause outages that affect utility reliability indices. Currently, other than DFA-IPSERS, no other system exist that is cable of automatically detecting and reporting conductor-slap related interruptions. Following paragraphs describe how a utility company de-

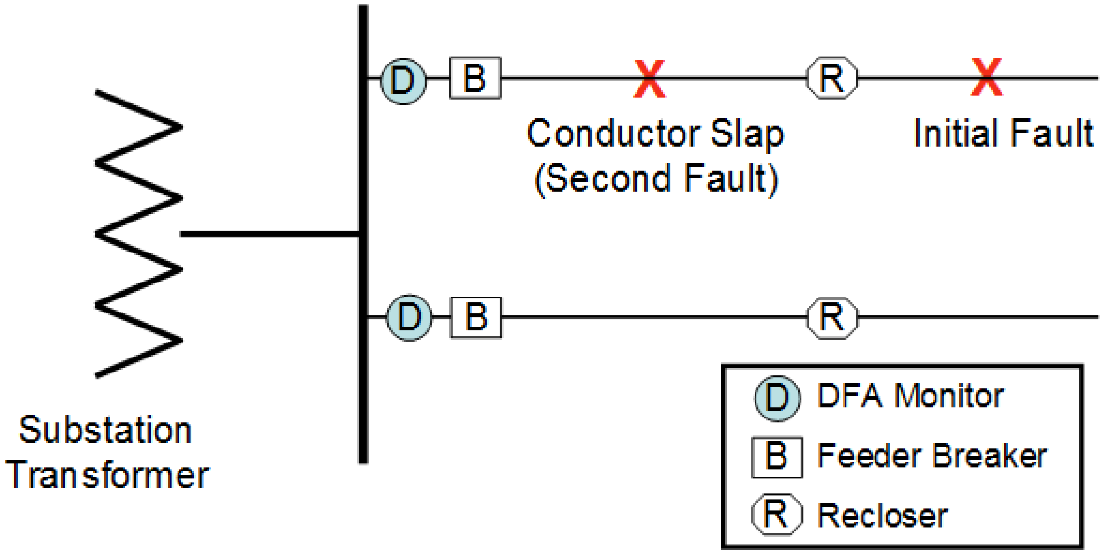


Figure 7.17: One-line diagram of a typical fault induced conductor slap scenario

tected and fixed conductor slap, using the information provided by DFA-IPSERS.

On November 6, 2011, a customer reported a pole fire on one of the feeders that DFA was monitoring. Utility crew responded and arrived at the location pictured in Figure 7.18 . However, they did not find the pole fire. About the same time when the customer reported a pole fire, a fault on the same circuit (but not near the pole fire) caused a mid-point line recloser to trip. Normally, the mid-point recloser should have tripped and isolated the fault. However, the substation breaker tripped too and locked out ht entire circuit. The circuit's self-healing system did not respond properly to avoid the outage either. Utility personnel were uncertain if the 'pole fire' was related to the fault elsewhere on the circuit. They could not explain why the breaker operated either; that was until DFA-IPSERS reported the initial fault and breaker lockout with information such as the protective device operation sequence and fault current magnitudes. DFA-IPSERS also automatically recognized the conductor slap event and sent an email along with RMS current waveforms. Figure 7.19 shows a screen capture of conductor slap

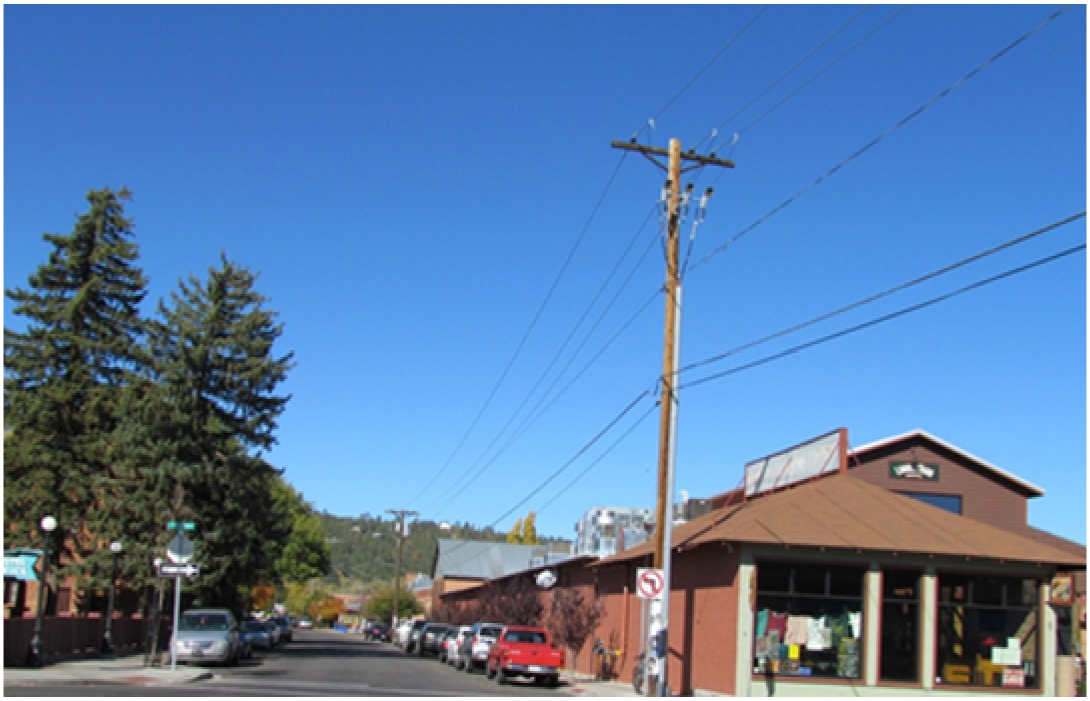
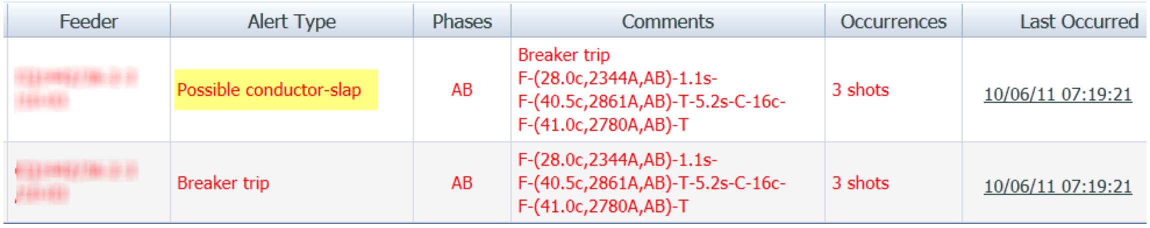


Figure 7.18: Picture of the pole where a pole fire was reported



| Feeder | Alert Type | Phases | Comments | Occurrences | Last Occurred |
|---|---|---|---|---|---|
| [redacted] | Possible conductor-slap | AB | Breaker trip<br>F-(28.0c,2344A,AB)-1.1s-<br>F-(40.5c,2861A,AB)-T-5.2s-C-16c-<br>F-(41.0c,2780A,AB)-T | 3 shots | 10/06/11 07:19:21 |
| [redacted] | Breaker trip | AB | F-(28.0c,2344A,AB)-1.1s-<br>F-(40.5c,2861A,AB)-T-5.2s-C-16c-<br>F-(41.0c,2780A,AB)-T | 3 shots | 10/06/11 07:19:21 |

Figure 7.19: Screen capture of conductor slap alert

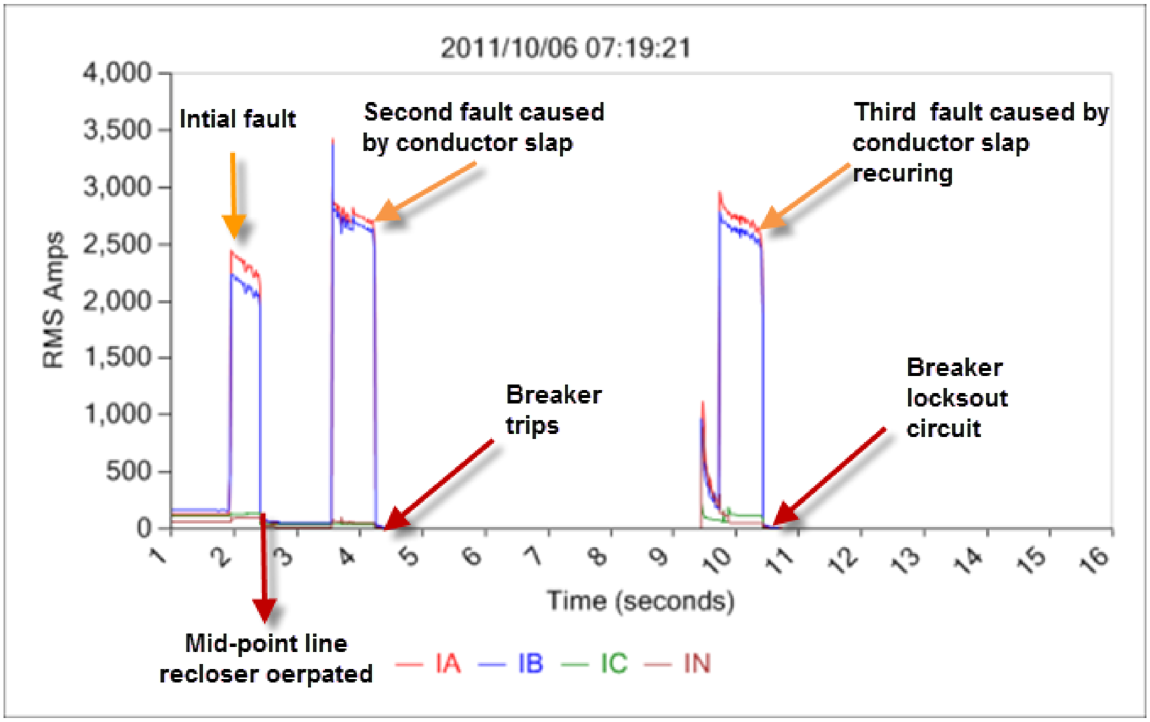


Figure 7.20: RMS current waveforms explaining the sequence of operations

and breaker lockout alerts that DFA-IPSERS reported. Figure 7.20 clearly shows the first fault that caused a mid-point recloser operation. It also shows the two subsequent faults that fault induced conductor slap caused.

Utility personnel used fault current estimates provided by DFA-IPSERS as inputs to their system model. The system model put the location of conductor slap within few pole spans of initially reported pole fire. With the knowledge, that conductor slap was the cause of breaker lockout, utility personnel conducted the search again near the reported pole fire. Their search identified arced wires, consistent with conductor slap (Figure 7.21 ). Conductor slap also explained the 'pole fire' reported by the customer.

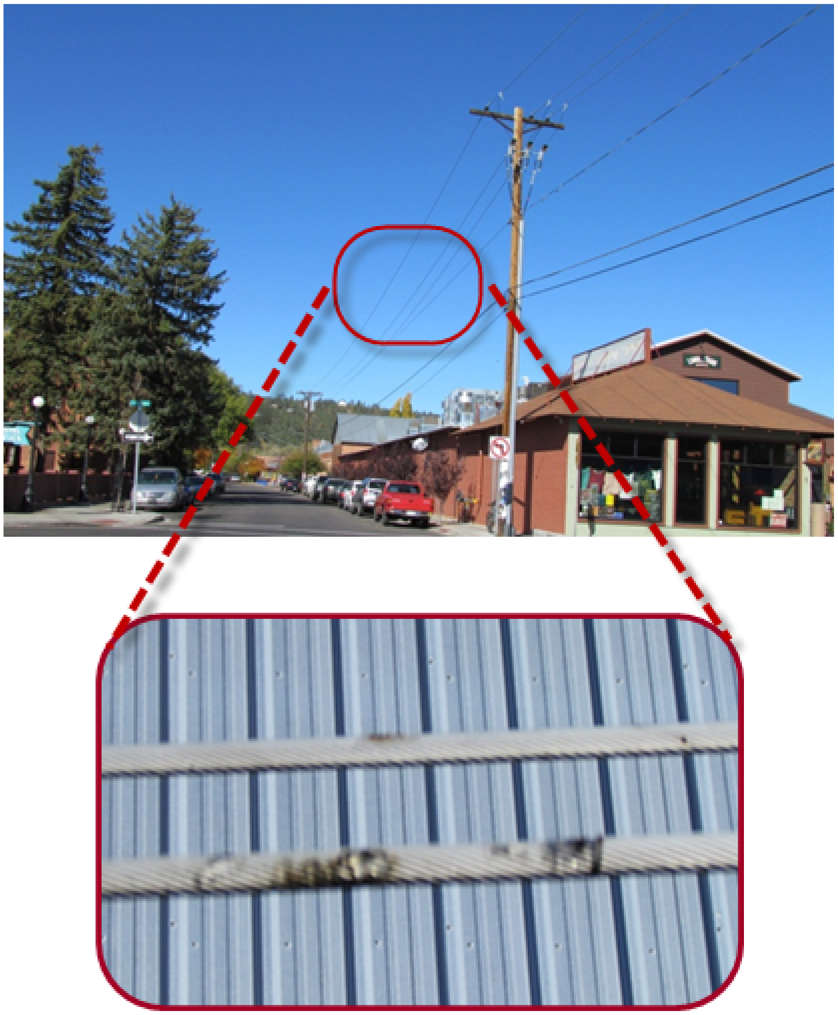


Figure 7.21: Pictures of the location of conductor slap and arced conductors

## 7.5 Chapter Summary

In this chapter, a fuzzy logic based, Intelligent Power System Event Reporting System (IPSERS) was presented. Classification performance of the Fuzzy Logic based Classification Algorithm (FLCA) and reporting algorithms were analyzed in detail. Finally, several case studies that documented the benefits of IPSERS were provided. These case studies showed real examples where IPSERS either helped, or could have helped utility companies to detect, locate and fix components that are failing or have already failed.

# 8. CONCLUSIONS

## 8.1 Overview

The solution presented in this dissertation changes how power utility companies monitor the health of distribution feeders, and how they respond to problems on distribution feeders. Prior to this research, utility companies lacked the necessary tools to notify them of apparatus on their feeders that have either failed or were in the process of failing. On most occasions, utility companies are unaware of a problem unless it is either reported by a customer, or it caused an outage. To remedy this situation, this research has produced a framework for intelligently monitoring and reporting problems on distribution feeders. By reporting problems in near real-time, and presenting relevant information to utility personnel, this research has provided utility personnel with much needed tools to locate and fix problems on distribution feeders, in a timely fashion.

## 8.2 Conclusions

With regard to the goals set forth in this dissertation, the results of this project can only be described as an unqualified success. A new, on-line, non-intrusive, classification system was developed for identifying, and reporting normal and abnormal power system events occurring on a distribution feeder based on their underlying cause, using signals acquired at the distribution substation. Analysis of the classification accuracy of the classification system, and real world examples of utility companies benefiting from this research, have provided incontrovertible evidence of success. As a part of developing the classification and reporting system, the following research goals were achieved:

1. ***Developing a fuzzy logic based expert system:*** A fuzzy logic based classifier (FLCA) was developed for classifying power system event data. FLCA is a fuzzy hierarchical classifier that is both modular and scalable. A custom fuzzy inference engine that supports a hierarchical structure, and is best suited for power system event classification was also developed for use with FLCA. FLCA and the custom inference engine were described in detail in Chapter 6. The classification accuracy of FLCA was analyzed, and the proof for the effectiveness of FLCA in classifying power system events was provided in Chapter 7.

2. ***Identifying characteristic event signatures:*** Characteristic features for power system events caused by low current arcing, overcurrents, misoperating or failing capacitor banks were identified. Algorithms were developed to extract these event specific features. Event specific features and algorithms to extract these features were presented in Chapter 5.

3. ***Developing RMS shape analysis algorithm:*** The Fuzzy Dynamic Time Warping (FDTW) based RMS shape analysis algorithm was developed to detect shapes and extract shape based features from root mean square (RMS) current, voltage and power waveforms. FDTW and shape based features were explained in Chapter 4.

4. ***Identifying and computing important parameters:*** Event-specific parameters that aid in locating the root-cause of an event were identified, and algorithms for extracting these parameters were developed. Parameters reported for each event category were summarized in Chapter 7. Case studies that document how these reported parameters helped utility companies find the physical location of failed or failing components, were also presented in

Chapter 7.

5. ***Intelligent reporting to avoid information overload:*** IPSERS, an intelligent reporting framework for processing raw classification information generated by the fuzzy classifier, and reporting events of interest in a timely and user friendly manner was developed. IPSERS was described in Chapter 7. The effectiveness of IPSERS was proved through the analysis of classification accuracies and case studies presented in Chapter 7.

6. ***Design of efficient classification algorithms:*** In order to report problems on distribution feeders to utility personnel in a timely fashion, all components of the expert system classifier were designed to be efficient and operate in soft real-time.

## 8.3 Future Research

There is a lot of scope for improving and extending this research. The following are suggested research directions:

1. This research uses data collected from distribution substation only. It does not take advantage of measurements made from multiple points on a distribution feeder, when such measurements are available. Distributed measurements can possibly be used to provide better location information for events recorded on distribution feeders. The relative value of using distributed data measurements has not been evaluated.

2. The scope of this research was limited to radial distribution feeders. Ideas presented in this research can be extended to low and medium voltage networks , looped feeders and transmission systems.

3. The intelligent reporting framework developed as apart of this research is a stand-alone system. For the system to be even more effective, considerable work remains to be done in seamlessly integrating the reporting framework into day-to-day utility operations.

4. New failure signatures for power system components are being identified and documented on a regular basis using data collected by DFA monitoring units. In order to detect these new failure signatures, classification algorithms need to be extended and updated on a regular basis.

5. There is still some room for improving the classification accuracy and accuracy of parameters reported by algorithms that were developed for this research.

In summary, this research provides a system framework to detect, locate, and possibly prevent failures on distribution feeders. It is hoped that this and future work will have a serious impact on detecting and mitigating failures on distribution feeders, reducing the safety hazards and economic impact they produce and thereby improving power quality and reliability.

# REFERENCES

[1] T. Gonen, *Electric power distribution system engineering*, ser. McGraw-Hill Series in Electrical Engineering. McGraw-Hill, 1986. [Online]. Available: http://books.google.com/books?id=eN9SAAAAMAAJ

[2] *Distribution Fault Anticipator, Phase I: Proof of Concept, Final Report Palo Alto, CA, EPRI Publ. 1001879*, Dec. 2001, prepared for the Electric Power Research Institute (EPRI).

[3] M. H. J. Bollen, "What is power quality?" *Electric Power Systems Research*, vol. 66, no. 1, pp. 5 –14, 2003, power Quality. [Online]. Available: http://www.sciencedirect.com/science/article/B6V30-48KVGYT-1/2/e0a2e24677d95f93bfb57c74a9ed68c7

[4] R. C. Dugan, S. Santoso, M. F. McGranaghan, and H. W. Beaty, *Electric Power Systems Quality*, 2nd ed., S. S. Chapman, Ed. McGraw-Hill Professional, 2002. [Online]. Available: http://books.google.com/books?id=Y4IvvSJq1bMC

[5] Z. Chen and P. Urwin, "Power quality detection and classification using digital filters," *Power Tech Proceedings, 2001 IEEE Porto*, vol. 1, pp. 1–6, 2001.

[6] Z. Lu, D. R. Turner, Q. H. Wu, J. Fitch, and S. Mann, "Morphological transform for detection of power quality disturbances," *Power System Technology, 2004. PowerCon 2004. 2004 International Conference on*, vol. 2, pp. 1644–1649 Vol.2, Nov. 2004.

[7] J. Serra and L. Vincent, "An overview of morphological filtering," *Circuits Syst. Signal Process.*, vol. 11, no. 1, pp. 47–108, 1992.

[8] E. Styvaktakis, M. H. J. Bollen, and I. Y. H. Gu, "Expert system for classification and analysis of power system events," *Power Delivery, IEEE Transactions on*, vol. 17, no. 2, pp. 423–428, Apr. 2002.

[9] J. Zhang, A. Swain, N. K. C. Nair, and J. J. Liu, "Estimation of power quality using an unscented kalman filter," *TENCON IEEE Region 10 Conference*, pp. 1–4, Nov. 2007.

[10] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification (2nd Edition)*. Wiley-Interscience, Nov. 2000. [Online]. Available: http://www.amazon.ca/exec/obidos/redirect?tag=citeulike09-20&amp;path=ASIN/0471056693

[11] S.-K. Chai, Sekar, and Rajan, "Power disturbance identification through pattern recognition system," *Southeastcon 2000. Proceedings of the IEEE*, pp. 154–157, 2000.

[12] P. K. Dash, S. Mishra, M. A. Salama, and A. C. Liew, "Classification of power system disturbances using a fuzzy expert system and a fourier linear combiner," *Power Delivery, IEEE Transactions on*, vol. 15, no. 2, pp. 472–477, Apr. 2000.

[13] P. K. Dash, D. P. Swain, A. C. Liew, and S. Rahman, "An adaptive linear combiner for on-line tracking of power system harmonics," *Power Systems, IEEE Transactions on*, vol. 11, no. 4, pp. 1730–1735, Nov. 1996.

[14] Y. Liao and J.-B. Lee, "A fuzzy-expert system for classifying power quality disturbances," *International Journal of Electrical Power & Energy Systems*, vol. 26, no. 3, pp. 199 – 205, 2004. [Online]. Available: http://www.sciencedirect.com/science/article/B6V2T-4B2CKMB-2/2/86ba86526efb860b829da8cbde584704

[15] J. C. Goswami and A. K. Chan, *Fundamentals of wavelets: theory, algorithms, and applications*, ser. Wiley series in microwave and optical engineering. John Wiley, 1999.

[16] R. G. Stockwell, L. Mansinha, and R. P. Lowe, "Localization of the complex spectrum: The s transform," *Signal Processing, IEEE Transactions on*, vol. 44, no. 4, pp. 998–1001, Apr. 1996.

[17] J. B. Reddy, D. K. Mohanta, and B. M. Karan, "Power system disturbance recognition using wavelet and s-transform techniques," *International Journal of Emerging Electric Power Systems*, vol. 1, 2004. [Online]. Available: http://www.bepress.com.ezproxy.tamu.edu:2048/cgi/viewcontent.cgi?article=1007&context=ijeeps

[18] M. V. Chilukuri and P. K. Dash, "Multiresolution s-transform-based fuzzy recognition system for power quality events," *Power Delivery, IEEE Transactions on*, vol. 19, no. 1, pp. 323–330, Jan. 2004.

[19] L. Shangwei and S. Yarning, "Study on the classification method of power disturbances based on the combination of s transform and svm multi-class classifier with binary tree," *Electric Utility Deregulation and Restructuring and Power Technologies, 2008. DRPT 2008. Third International Conference on*, pp. 2275–2280, Apr. 2008.

[20] J. C. Goswami and A. K. Chan, *Fundamentals of wavelets: theory, algorithms, and applications*, ser. Wiley series in microwave and optical engineering. John Wiley, 1999.

[21] S. Santoso, E. J. Powers, W. M. Grady, and P. Hofmann, “Power quality assessment via wavelet transform analysis,” *Power Delivery, IEEE Transactions on*, vol. 11, no. 2, pp. 924–930, Apr. 1996.

[22] S. Santoso, E. J. Powers, W. M. Grady, and A. C. Parsons, “Power quality disturbance waveform recognition using wavelet-based neural classifier. i. theoretical foundation,” *Power Delivery, IEEE Transactions on*, vol. 15, no. 1, pp. 222–228, Jan 2000.

[23] S. Santoso, E. J. Powers, W. M. Grady, and A. C. Parsons, “Power quality disturbance waveform recognition using wavelet-based neural classifier. ii. application,” *Power Delivery, IEEE Transactions on*, vol. 15, no. 1, pp. 229–235, Jan 2000.

[24] C. M. Bishop, *Neural Networks for Pattern Recognition*. Oxford University Press, Nov. 1995. [Online]. Available: http://www.amazon.ca/exec/obidos/redirect?tag=citeulike09-20&amp;path=ASIN/0198538642

[25] A. Elmitwally, S. Farghal, M. Kandil, S. Abdelkader, and M. Elkateb, “Proposed wavelet-neurofuzzy combined system for power quality violations detection and diagnosis,” *Generation, Transmission and Distribution, IEE Proceedings*, vol. 148, no. 1, pp. 15–20, Jan 2001.

[26] D. Nauck and R. Kruse, “A neuro-fuzzy method to learn fuzzy classification rules from data,” *Fuzzy Sets and Systems*, vol. 89, no. 3, pp.

277 – 288, 1997, application of Neuro-Fuzzy Systems. [Online]. Available: http://www.sciencedirect.com/science/article/B6V05-3SNVJ8N-J/2/a4c7ff0282d10e162fcb2c8f6ce27d7a

[27] J. Chung, E. J. Powers, W. M. Grady, and S. C. Bhatt, "Power disturbance classifier using a rule-based method and wavelet packet-based hidden markov model," *Power Delivery, IEEE Transactions on*, vol. 17, no. 1, pp. 233–241, Jan 2002.

[28] J. C. Goswami and A. K. Chan, *Fundamentals of wavelets: theory, algorithms, and applications*, ser. Wiley series in microwave and optical engineering. John Wiley, 1999.

[29] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, Feb 1989.

[30] T. K. Abdel-Galil, E. F. El-Saadany, A. M. Youssef, and M. M. A. Salama, "Disturbance classification using hidden markov models and vector quantization," *Power Delivery, IEEE Transactions on*, vol. 20, no. 3, pp. 2129–2135, Jul. 2005.

[31] H. Zang and X. Yu, "Disturbance classification utilizing wavelet and multiclass support vector machines," in *Proc. Fourth International Conference on Natural Computation ICNC '08*, vol. 3, 18–20 Oct. 2008, pp. 170–174.

[32] C. Kocaman, H. Usta, M. Ozdemir, and I. Eminoglu, "Classification of two common power quality disturbances using wavelet based svm," in *MELECON 2010 - 2010 15th IEEE Mediterranean Electrotechnical Conference*, 26-28 2010, pp. 587 –591.

[33] G. J. Klir and B. Yuan, *Fuzzy Sets and Fuzzy Logic: Theory and Applications*, illustrated ed. Upper Saddle River, NJ: Prentice Hall PTR, 1995.

[34] L. M. E., S. Arumugam, and S. Chandrasekar, “Fuzzy recognition system for power quality events using spline wavelet,” *Power Systems Conference and Exposition, 2004. IEEE PES*, pp. 1502–1505 vol.3, Oct. 2004.

[35] M. B. I. Reaz, F. Choong, M. S. Sulaiman, F. Mohd-Yasin, and M. Kamada, “Expert system for power quality disturbance classifier,” *Power Delivery, IEEE Transactions on*, vol. 22, no. 3, pp. 1979–1988, Jul. 2007.

[36] M. I. Chacon, J. L. Duran, and L. A. Santiesteban, “A wavelet-fuzzy logic based system to detect and identify electric power disturbances,” *Computational Intelligence in Image and Signal Processing, 2007. CIISP 2007. IEEE Symposium on*, pp. 52–57, Apr. 2007.

[37] T. K. Abdel-Galil, M. Kamel, A. M. Youssef, E. F. El-Saadany, and M. M. A. Salama, “Power quality disturbance classification using the inductive inference approach,” *Power Delivery, IEEE Transactions on*, vol. 19, no. 4, pp. 1812–1818, Oct. 2004.

[38] W. E. Kazibwe and H. M. Sendaula, “Expert system targets power quality issues,” *Computer Applications in Power, IEEE*, vol. 5, no. 2, pp. 29–33, Apr. 1992.

[39] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*. Pearson Education, 2003. [Online]. Available: http://portal.acm.org/citation.cfm?id=773294

[40] J. Wang and C. Wang, "Bayes method of power quality disturbance classification," *TENCON 2005 2005 IEEE Region 10*, pp. 1–4, Nov. 2005.

[41] C. J. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, pp. 121–167, 1998.

[42] P. Janik and T. Lobos, "Automated classification of power-quality disturbances using svm and rbf networks," *Power Delivery, IEEE Transactions on*, vol. 21, no. 3, pp. 1663–1669, Jul. 2006.

[43] J. Valyon and G. Horváth, "A weighted generalized ls-svm," 2003.

[44] A. Girgis, C. Fallon, and D. Lubkeman, "A fault location technique for rural distribution feeders," *Industry Applications, IEEE Transactions on*, vol. 29, no. 6, pp. 1170 –1175, Nov. 1993.

[45] J. Kim, M. Baran, and G. Lampley, "Estimation of fault location on distribution feeders using pq monitoring data," in *Power Engineering Society General Meeting, 2007. IEEE*, 24-28 2007, pp. 1 –4.

[46] D. Sabin, C. Dimitriu, D. Santiago, and G. Baroudi, "Overview of an automatic underground distribution fault location system," in *Power Energy Society General Meeting, 2009. PES '09. IEEE*, 26-30 2009, pp. 1 –5.

[47] D. Thomas, R. Carvalho, and E. Pereira, "Fault location in distribution systems based on traveling waves," in *Power Tech Conference Proceedings, 2003 IEEE Bologna*, vol. 2, 23-26 2003, p. 5 pp. Vol.2.

[48] E. Martinez and E. Richards, "An expert system to assist distribution dispatchers in the location of system outages," in *Rural Electric Power Confer-*

*ence, 1991. Papers Presented at the 35th Annual Conference*, Apr. 1991, pp. A2/1 –A2/5.

[49] C. Fukui and J. Kawakami, “An expert system for fault section estimation using information from protective relays and circuit breakers,” *Power Delivery, IEEE Transactions on*, vol. 1, no. 4, pp. 83 –90, Oct. 1986.

[50] Y. Liu and N. Schulz, “Knowledge-based system for distribution system outage locating using comprehensive information,” *Power Systems, IEEE Transactions on*, vol. 17, no. 2, pp. 451 –456, may 2002.

[51] H. Monsef, A. Ranjbar, and S. Jadid, “Fuzzy rule-based expert system for power system fault diagnosis,” *Generation, Transmission and Distribution, IEE Proceedings-*, vol. 144, no. 2, pp. 186 –192, mar 1997.

[52] H.-J. Lee, D.-Y. Park, B.-S. Ahn, Y.-M. Park, J.-K. Park, and S. Venkata, “A fuzzy expert system for the integrated fault diagnosis,” *Power Delivery, IEEE Transactions on*, vol. 15, no. 2, pp. 833 –838, Apr. 2000.

[53] K. Muthu-Manivannan, “Fuzzy logic based operated device identification in power distribution systems,” Master’s thesis, Texas A&M University, May 2002.

[54] B. Das, “Fuzzy logic-based fault-type identification in unbalanced radial power distribution system,” *Power Delivery, IEEE Transactions on*, vol. 21, no. 1, pp. 278 – 285, Jan. 2006.

[55] J. Zhu, D. Lubkeman, and A. Girgis, “Automated fault location and diagnosis on electric power distribution feeders,” *Power Delivery, IEEE Transactions on*, vol. 12, no. 2, pp. 801 –809, Apr. 1997.

[56] C.-F. Chien, S.-L. Chen, and Y.-S. Lin, “Using bayesian network for fault location on distribution feeder,” *Power Delivery, IEEE Transactions on*, vol. 17, no. 3, pp. 785 – 793, Jul. 2002.

[57] Z. Yongli, H. Limin, and L. Jinling, “Bayesian networks-based approach for power systems fault diagnosis,” *Power Delivery, IEEE Transactions on*, vol. 21, no. 2, pp. 634 – 639, Apr. 2006.

[58] H.-T. Yang, W.-Y. Chang, and C.-L. Huang, “A new neural networks approach to on-line fault section estimation using information of protective relays and circuit breakers,” *Power Delivery, IEEE Transactions on*, vol. 9, no. 1, pp. 220 –230, Jan. 1994.

[59] T. Bi, Y. Ni, C. Shen, and F. F. Wu, “An on-line distributed intelligent fault section estimation system for large-scale power networks,” *Electric Power Systems Research*, vol. 62, no. 3, pp. 173 – 182, 2002. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0378779602000421

[60] L. S. Martins, J. Martins, V. F. Pires, and C. Alegria, “A neural space vector fault location for parallel double-circuit distribution lines,” *International Journal of Electrical Power &amp; Energy Systems*, vol. 27, no. 3, pp. 225 – 231, 2005. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0142061504001449

[61] J. Mora-Florez, V. Barrera-Nuez, and G. Carrillo-Caicedo, “Fault location in power distribution systems using a learning algorithm for multivariable data analysis,” *Power Delivery, IEEE Transactions on*, vol. 22, no. 3, pp. 1715 –1721, Jul. 2007.

[62] *Detection of Arcing Faults on Distribution Feeders, Final Report Palo Alto, CA, EPRI Project 1285-3*, Dec. 1982, prepared for the Electric Power Research Institute (EPRI).

[63] C. Benner and B. Russell, "Practical high-impedance fault detection on distribution feeders," *Industry Applications, IEEE Transactions on*, vol. 33, no. 3, pp. 635 –640, May/Jun. 1997.

[64] B. Russell and C. Benner, "Arcing fault detection for distribution feeders: security assessment in long term field trials," *Power Delivery, IEEE Transactions on*, vol. 10, no. 2, pp. 676 –683, Apr. 1995.

[65] B. D. Russell and C. L. Benner, "Performance of high-impedance fault detection algorithms in long-term field trials," *Electric Power Systems Research*, vol. 31, no. 2, pp. 71 – 77, 1994. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0378779694900833

[66] C. Benner, P. Carswell, and B. D. Russell, "Improved algorithm for detecting arcing faults using random fault behavior," *Electric Power Systems Research*, vol. 17, no. 1, pp. 49 – 56, 1989. [Online]. Available: http://www.sciencedirect.com/science/article/pii/037877968990059X

[67] M. Sedighizadeh, A. Rezazadeh, and N. I. Elkalashy, "Approaches in High Impedance Fault Detection - A Chronological Review," *Advances in Electrical and Computer Engineering*, vol. 10, no. 3, pp. 114–128, 2010. [Online]. Available: http://dx.doi.org/10.4316/aece.2010.03019

[68] K. Muthu-Manivannan, C. L. Benner, P. Xu, and B. D. Russell, "Arcing event detection," US Patent Application 20110137590, 2010.

[69] A. Farag, C. Wang, T. Cheng, G. Zheng, Y. Du, L. Hu, B. Palk, and M. Moon, "Failure analysis of composite dielectric of power capacitors in distribution systems," *Dielectrics and Electrical Insulation, IEEE Transactions on*, vol. 5, no. 4, pp. 583 –588, Aug. 1998.

[70] W.-J. Lee, K. Narayanan, T. Maffetone, and P. Didsayabutra, "The design of a capacitor bank early warning system," *Industry Applications, IEEE Transactions on*, vol. 39, no. 2, pp. 306 – 312, Mar./Apr. 2003.

[71] S. Santoso, J. Lamoree, and M. McGranaghan, "Signature analysis to track capacitor switching performance," in *Transmission and Distribution Conference and Exposition, 2001 IEEE/PES*, vol. 1, 2001, pp. 259 –263 vol.1.

[72] D. Sochuliakova, D. Niebur, C. Nwankpa, R. Fischl, and D. Richardson, "Identification of capacitor position in a radial system," *Power Delivery, IEEE Transactions on*, vol. 14, no. 4, pp. 1368 –1373, Oct. 1999.

[73] H. Khani, M. Moallem, S. Sadri, and M. Dolatshahi, "A new method for online determination of the location of switched capacitor banks in distribution systems," *Power Delivery, IEEE Transactions on*, vol. 26, no. 1, pp. 341 –351, Jan. 2011.

[74] B. Russell and C. Benner, "Intelligent systems for improved reliability and failure diagnosis in distribution systems," *Smart Grid, IEEE Transactions on*, vol. 1, no. 1, pp. 48 –56, Jun. 2010.

[75] *Distribution Fault Anticipator, Phase II: Algorithm Development and Second-Year Data Collection, Final Report Palo Alto, CA, EPRI Publ. 101066*, Nov. 2005, prepared for the Electric Power Research Institute (EPRI).

[76] *Distribution Fault Anticipator, Phase III: System Integration, Technical Update Palo Alto, CA, EPRI Publ. 1012435*, Dec. 2006, prepared for the Electric Power Research Institute (EPRI).

[77] K. Muthu-Manivannan, C. L. Benner, P. Xu, and B. D. Russell, "Phase drift compensation for sampled signals," US Patent Application 20090116600, 2009. [Online]. Available: http://www.patentlens.net/patentlens/patent/US_2009_0116600_A1/en/

[78] R. G. Cowell, A. P. Dawid, S. L. Lauritzen, and D. J. Spiegelhalter, "Introduction," in *Probabilistic Networks and Expert Systems*, ser. Information Science and Statistics, M. Jordan, S. L. Lauritzen, J. F. Lawless, and V. Nair, Eds. Springer New York, 1999, pp. 1–4.

[79] T. Ross, *Fuzzy logic with engineering applications*. John Wiley, 2004. [Online]. Available: http://books.google.com/books?id=WVMTwrzGhosC

[80] L. Zadeh, "The role of fuzzy logic in the management of uncertainty in expert systems," *Fuzzy Sets and Systems*, vol. 11, no. 1-3, pp. 197 – 198, 1983. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0165011483800815

[81] G. Shafer, "Belief functions and parametric models," in *Classic Works of the Dempster-Shafer Theory of Belief Functions*, ser. Studies in Fuzziness and Soft Computing, R. Yager and L. Liu, Eds. Springer Berlin / Heidelberg, 2008, vol. 219, pp. 265–290.

[82] A. Gelman, *Bayesian data analysis*, ser. Texts in statistical science. Chapman & Hall/CRC, 2004. [Online]. Available: http://books.google.com/books?id=TNYhnkXQSjAC

[83] D. V. Lindley, "The probability approach to the treatment of uncertainty in artificial intelligence and expert systems," *Statistical Science*, vol. 2, no. 1, pp. pp. 17–24, 1987. [Online]. Available: http://www.jstor.org/stable/2245603

[84] L. Zadeh, "A theory of approximate reasoning," in *Machine intelligence*, J. Hayes, D. Michie, and L. Mikulich, Eds. Elsevier, Amsterdam, 1979, pp. 149–194.

[85] L. A. Zadeh, "Toward a perception-based theory of probabilistic reasoning with imprecise probabilities," *Journal of statistical planning and inference*, vol. 105, no. 1, pp. 233–264, 2002.

[86] B. Cobb, R. Ruma, and A. Salmeran, "Modeling conditional distributions of continuous variables in bayesian networks," in *Advances in Intelligent Data Analysis VI*, ser. Lecture Notes in Computer Science, A. Famili, J. Kok, J. Pea, A. Siebes, and A. Feelders, Eds. Springer Berlin / Heidelberg, 2005, vol. 3646, pp. 742–742.

[87] W. Wiegerinck, B. Kappen, and W. Burgers, "Bayesian networks for expert systems: Theory and practical applications," in *Interactive Collaborative Information Systems*, ser. Studies in Computational Intelligence, R. BabuA¡ka and F. Groen, Eds. Springer Berlin / Heidelberg, 2010, vol. 281, pp. 547–578.

[88] D. Dubois and H. Prade, "Fuzzy sets in approximate reasoning, part 1: Inference with possibility distributions," *Fuzzy Sets and Systems*, vol. 40, no. 1, pp. 143 – 202, 1991.

[89] J. Yen and R. Langari, *Fuzzy logic: intelligence, control, and information*. Prentice Hall, 1999.

[90] G. Pillo and A. Murli, *High performance algorithms and software for nonlinear optimization*, ser. Applied optimization. Kluwer Academic Publishers, 2003, vol. 82. [Online]. Available: http://books.google.com/books?id=tQtG11l0V44C

[91] C. Myers, L. Rabiner, and A. Rosenberg, "An investigation of the use of dynamic time warping for word spotting and connected speech recognition," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '80.*, vol. 5, Apr. 1980, pp. 173 – 177.

[92] A. W.-C. Fu, E. Keogh, L. Y. Lau, C. A. Ratanamahatana, and R. C.-W. Wong, "Scaling and time warping in time series querying," *The VLDB Journal*, vol. 17, pp. 899–921, Jul. 2008. [Online]. Available: http://dx.doi.org.lib-ezproxy.tamu.edu:2048/10.1007/s00778-006-0040-z

[93] A. Youssef, T. Abdel-Galil, E. El-Saadany, and M. Salama, "Disturbance classification utilizing dynamic time warping classifier," *Power Delivery, IEEE Transactions on*, vol. 19, no. 1, pp. 272 – 278, Jan. 2004.

[94] S. Salvador and P. Chan, "Toward accurate dynamic time warping in linear time and space," *Intell. Data Anal.*, vol. 11, pp. 561–580, Oct. 2007. [Online]. Available: http://dl.acm.org/citation.cfm?id=1367985.1367993

[95] R. Bellman, *Dynamic Programming*, ser. Dover Books on Mathematics. Dover Publications, 2003. [Online]. Available: http://books.google.com/books?id=fyVtp3EMxasC

[96] S. Dasgupta, C. Papadimitriou, and U. Vazirani, *Algorithms*. McGraw-Hill Higher Education, 2008. [Online]. Available: http://books.google.com/books?id=3sCxQgAACAAJ

[97] R. E. Bellman and L. A. Zadeh, “Decision-making in a fuzzy environment,” *Management Science*, vol. 17, no. 4, pp. pp. B141–B164, 1970. [Online]. Available: http://www.jstor.org/stable/2629367

[98] J. Kacprzyk and A. O. Esogbue, “Fuzzy dynamic programming: Main developments and applications,” *Fuzzy Sets and Systems*, vol. 81, no. 1, pp. 31 – 45, 1996, fuzzy Optimization. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0165011495002391

[99] J. Li, “Methodology for designing the fuzzy resolver for a radial distribution system fault locator,” Ph.D. dissertation, Texas A&M University, Dec. 2005.

[100] “IEEE guide for the application of shunt power capacitors,” *IEEE Std 1036-2010 (Revision of IEEE Std 1036-1992)*, pp. 1 –97, 17 2011.

[101] K. Muthu-Manivannan, C. L. Benner, P. Xu, and B. D. Russell, “Diagnosis and position identification for remote capacitor banks,” US Patent Application 20100169029, 2010.

[102] C. Benner and B. Russell, “Investigation of incipient conditions leading to the failure of distribution system apparatus,” in *Power Systems Conference and Exposition, 2004. IEEE PES*, Oct. 2004, pp. 703 – 708 vol.2.

[103] K. Muthu-Manivannan, C. L. Benner, P. Xu, and B. D. Russell, “Identification of power system events using fuzzy logic,” US Patent Application 20090327201, 2009.

[104] J. Pan and A. C. Kak, “Design of a large-scale expert system using fuzzy logic for uncertainty-reasoning,” Tech. Rep., 1994.

[105] J. Pan, G. Desouza, and A. Kak, “Fuzzyshell: A large-scale expert system shell using fuzzy logic for uncertainty reasoning,” *Fuzzy Systems, IEEE Transactions on*, vol. 6, no. 4, pp. 563 –581, Nov. 1998.

[106] Z. Sosnowski, G. DeSouza, and A. Kak, “Comments on ldquo;fuzzyshell: A large-scale expert system shell using fuzzy logic for uncertainty reasoning rdquo; [and reply],” *Fuzzy Systems, IEEE Transactions on*, vol. 8, no. 6, pp. 817 –821, dec 2000.

[107] J. Yen and R. Langari, *Fuzzy logic: intelligence, control, and information*. Prentice Hall, 1999.

[108] T. Nomura, “A proposal of interpretations on numerical degrees of confidence for fuzzy if-then rules and a mathematical verification of properties under various reasoning methods,” in *Knowledge-Based Intelligent Electronic Systems, 1997. KES ’97. Proceedings., 1997 First International Conference on*, vol. 2, may 1997, pp. 542 –549 vol.2.

[109] *CLIPS User’s Manual*, 5th ed., Software Technology Branch, Lyndon B, Johnson Space Center, Jan. 1992.

[110] C. L. Forgy, “On the efficient implementation of production systems,” Ph.D. dissertation, Department of Computer Sceince, CMU, Feb. 1979.

[111] C. L. and Forgy, “Rete: A fast algorithm for the many pattern/many object pattern match problem,” *Artificial Intelligence*, vol. 19, no. 1, pp. 17 – 37, 1982. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0004370282900200

[112] H. K. J. D. Guzman, "Spirit 2.5.1," http://www.boost.org/doc/libs/1_48_0/libs/spirit/doc/html/index.html, Nov. 2011.

[113] J. Giarratano and G. Riley, *Expert systems: principles and programming*. Thomson Course Technology, 2005, no. v. 1. [Online]. Available: http://books.google.com/books?id=SbgZRAAACAAJ

[114] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *International Joint Conference On Artificial Intellignece*, 1995, pp. 1137–1143.

[115] J. Bowers, C. L. Benner, B. D. Russell, and A. Sundaram, "Tree limb burns down line, causes outage. dfa technology could have prevented damage, outage," https://dfaweb.tamu.edu/DFAReports/DFASuccess/200501PickwickTreeLimbBurndown.pdf, Jan. 2005.

[116] J. Bowers, J. Fangue, B. Muston, C. L. Benner, B. D. Russell, and A. Sundaram, "Misbehaving capacitor controllers spell trouble dfa technology corrects problem, prevents damage," https://dfaweb.tamu.edu/DFAReports/DFASuccess/200505CapacitorControllerProblemsComparison.pdf, May. 2005.

[117] C. L. Benner and B. D. Russell, "Dfa technology alerts utilities to bad bushings, avoids outages," https://dfaweb.tamu.edu/DFAReports/DFASuccess/200512.AvoidedFault.BushingFailure.CaseStudy.pdf, Dec. 2005.

[118] C. L. Benner and B. D. Russell, "Pickwick uses dfa technology to predict, prevent vegetation outage," https://dfaweb.tamu.edu/DFAReports/

DFASuccess/201007.Pickwick.RecurrentFault.Tree.CaseStudy.pdf, Jul. 2010.

[119] D. Ward, "Overhead distribution conductor motion due to short-circuit forces," *Power Delivery, IEEE Transactions on*, vol. 18, no. 4, pp. 1534 – 1538, Oct. 2003.