Fall 1-31-2004

Adaptive learning for event modeling and pattern classification

Shuangshuang Dai
New Jersey Institute of Technology

ABSTRACT

ADAPTIVE LEARNING FOR EVENT MODELING AND PATTERN CLASSIFICATION

by
Shuangshuang Dai

It is crucial to detect, characterize and model events of interest in a new propulsion system. As technology advances, the amount of data being generated increases significantly with respect to time. This increase substantially strains our ability to interpret the data at an equivalent rate. It demands efficient methodologies and algorithms in the development of automated event modeling and pattern recognition to detect and characterize events of interest and correlate them to the system performance. The fact that the information required to properly evaluate system performance and health is seldom known in advance further exacerbates this issue.

Event modeling and detection is essentially a discovery problem and involves the use of techniques in the pattern classification domain, specifically the use of cluster analysis if a prior information is unknown. In this dissertation, a framework of Adaptive Learning for Event Modeling and Characterization (ALEC) system is proposed to deal with this problem. Within this framework, a wavelet-based hierarchical fuzzy clustering approach which integrates several advanced technologies and overcomes the disadvantages of traditional clustering algorithms is developed to make the implementation of the system effective and computationally efficient.

In another separate but related research, a generalized multi-dimensional Gaussian membership function is constructed and formulated to make the fuzzy classification of blade engine damage modes among a group of engines containing historical flight data

after Principal Component Analysis (PCA) is applied to reduce the excessive dimensionality. This approach can be effectively used to deal with classification of patterns with overlapping structures in which some patterns fall into more than one classes or categories.

ADAPTIVE LEARNING FOR
EVENT MODELING AND PATTERN CLASSIFICATION

by
Shuangshuang Dai

A Dissertation
Submitted to the Faculty of
New Jersey Institute of Technology
In Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy in Electrical Engineering

Department of Electrical and Computer Engineering

January 2004

APPROVAL PAGE

ADAPTIVE LEARNING FOR EVENT MODELING AND PATTERN CLASSIFICATION

Shuangshuang Dai

Dr. Atam P. Dhawan, Dissertation Advisor — Date
Professor of Electrical and Computer Engineering, NJIT

Dr. Edwin Hou, Committee Member — Date
Associate Professor of Electrical and Computer Engineering, NJIT

Dr. Chengjun Liu, Committee Member — Date
Assistant Professor of Computer Science, NJIT

Dr. Constantine Manikopoulos, Committee Member — Date
Associate Professor of Electrical and Computer Engineering, NJIT

Dr. Yun-Qing Shi, Committee Member — Date
Professor of Electrical and Computer Engineering, NJIT

BIOGRAPHICAL SKETCH

Author: Shuangshuang Dai

Degree: Doctor of Philosophy

Date: January 2004

Undergraduate and Graduate Education:

- Doctor of Philosophy in Electrical Engineering
  New Jersey Institute of Technology, Newark, NJ, 2004

- Master of Science in Biomedical Engineering
  Southeast University, Nanjing, P. R. China, 1997

- Bachelor of Science in Biomedical Engineering
  Southeast University, Nanjing, P. R. China, 1994

Major: Electrical Engineering

Presentations and Publications:

Shuangshuang Dai and Atam P. Dhawan,
"PCA-based Fuzzy Classification of Turbine Blade Damage Modes,"
The 40th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit,
Fort Lauderdale, Florida, July 2004, submitted and accepted.

Shuangshuang Dai and Atam P. Dhawan,
"Event Detection and Modeling of Engine Pressure Data: An Integrated Approach,"
AIAA Modeling and Simulation Technologies Conference and Exhibit,
Providence, Rhode Island, August 2004, submitted.

Shuangshuang Dai and Atam P. Dhawan,
"PCA-based Fuzzy Classification For Analysis of Turbine Blade Damage in
Aircraft Engines," Journal of Pattern Recognition, 2003, submitted.

Sachin V. Patwardhan, Shuangshuang Dai and Atam P. Dhawan, "Multi-spectral Image Analysis and Classification of Melanoma Using Crisp and Fuzzy Partitions," Computerized Medical Imaging and Graphics, 2003, submitted.

Shuangshuang Dai and Atam P. Dhawan
"Wavelet-based Adaptive Learning for Event Modeling and Characterization," IEEE Trans. Systems, Man and Cybernetics, 2003, submitted.

To my beloved wife, Fang Shi and dear son, Kevin Dai.

ACKNOWLEDGMENT

I would like to take this opportunity to thank my advisor Dr. Atam P. Dhawan for offering me a chance to work under his guidance and supervision, believing in me and encouraging my research efforts. I would also like to express my gratitude for his patience on both the personal and academic level. Special thanks are given to Dr. Edwin Hou, Dr. Chengjun Liu, Dr. Constantine Manikopoulous and Dr. Yun-Qing Shi for actively participating in my committee, reviewing this work and providing valuable criticisms and suggestions.

I want to thank the NASA Glenn Research Center to provide the research funding and to the reviewers, Mr. William Mall and Ms. Claudia Meyer for their insightful and critical comments and suggestions of the reports.

I also want to thank Mr. Sachin Patwardhan in our lab for numerous useful and constructive discussions and for his proofreading of my manuscript. His friendship is greatly appreciated.

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

LIST OF TABLES

Table **Page**

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Problem Statement and Motivation

With development of a new propulsion system, the information required to properly evaluate system performance and health is seldom known in advance. Capturing key elements within the growing volumes of data can become quite a monumental task. In addition, most developing systems are seldom static in their design and the task of developing and refining a health management system concurrently with the evolving client system design can be very difficult. Therefore, a need naturally arises for efficient methodologies and algorithms in the development of automated event modeling to detect and characterize events of interest and correlate them to the system performance. Event modeling and detection is essentially a discovery problem [1] and involves the use of techniques in the pattern classification domain, specifically the use of cluster analysis if a prior information is unknown.

Clustering is a division of data into groups of similar objects. From a machine learning perspective clusters correspond to hidden patterns, the search for clusters is unsupervised learning, and the resulting system represents a data concept. Traditional clustering algorithms can be classified into two main categories [2]: hierarchical and partitional. In hierarchical clustering, the number of clusters need not be specified a priori, and problems due to initialization and local minimum do not arise. However, since hierarchical methods consider only local neighbors in each step, they cannot incorporate a prior knowledge about the global shape or size of clusters. As a result, they cannot always separate overlapping clusters. Moreover, hierarchical clustering is static, and points committed to a given cluster in the early stages cannot move to a different cluster. Partitional clustering obtains a single partition of the data instead of a clustering structure by optimizing a criterion function defined either locally (on a subset of the patterns) or globally (defined over all of the patterns). Partitional clustering can be further divided into two classes: crisp clustering and fuzzy clustering. In crisp clustering, every data point belong to only one cluster, while in fuzzy clustering every data point belongs to every cluster to a certain degree as determined by the membership function [3].

Partitional algorithms are dynamic, and points can move from one cluster to another. They can incorporate knowledge about the shape or size of clusters by using appropriate prototypes and distance measures. The most often used partitional algorithm is fuzzy *k*-means method. However, there are three major difficulties encountered during fuzzy clustering of real data:

(1) The number of clusters can not always be defined a prior, and one has to find a cluster validity criterion [23] in order to determine the optimal number of clusters present in the data;

(2) The characteristics and location of cluster centroids are not necessarily known a prior, and initial guesses have to be made;

(3) The presence of large variability in cluster shapes, variations in cluster densities, and variability in the number of data points in each cluster.

Since the advantages and disadvantages of clustering algorithms of each major category have been demonstrated, an attempt can be made to find an approach which will combine the advantages of hierarchical and partitional clustering techniques. Within each category of algorithms, enhancement can be made to improve performance.

1.2 Objectives of this Study

As mentioned above, in order to build a robust framework to effectively detect events of interest for an operational propulsion system, several advanced technologies should be integrated in a unique manner to provide an adaptive self-learning system that detects and characterizes key features within the data as it is being collected and develops a knowledge base with key signatures and models of events of interest. As the system gains knowledge through interaction with the user, of what each characteristic feature means in relation to engine performance and health, this knowledge base becomes the foundation for routine health assessment. The knowledge base could provide information about variables correlated with near-term and long-term health conditions. Therefore, the objective of the dissertation can be summarized as follows:

- To compare various clustering algorithms about their advantages and disadvantages and decide how to modify within each standard algorithm and combine them together in order to better detect event of interest.
- To study the wavelet transform and choose the appropriate wavelet to aid feature extraction in the time-localized frequency.
- To study the fuzzy logic and generalize the membership function into multi-dimensional feature space.
- To complete system integration of the proposed framework including feature extraction, adaptive clustering and defining the signature models for prospective events.

1.3 State of the Art

Within each clustering algorithms, numerous variants have appeared. Hierarchical clustering is inflexible due to its greedy approach: after a merge or a split is selected it is not refined. Fisher [4] studied iterative hierarchical cluster redistribution to improve once constructed dendrograms. Karypis et al. [5] also researched refinements for hierarchical clustering. The problem with partitional algorithms is the initial guess of the number of clusters. A simple way to mitigate the effects of clusters initialization was suggested by Bradley and Fayyad [6]. First, *k*-means is performed on several small samples of data with a random initial guess . Each of these constructed systems is then used as a potential initialization for a union of all the samples. Centroids of the best system constructed this way are suggested as an intelligent initial guesses to ignite the *k*-means algorithm on the full data. Zhang [7] suggested another way to rectify the optimization process by soft assignment of points to different clusters with appropriate weights, rather than by moving them decisively from one cluster to another. Nowadays, probabilistic models have been proposed as a basis for cluster analysis. In this approach, the data are viewed as coming from a mixture of probability distributions, each representing a different cluster. Methods of this type have shown promise in a number of practical applications [8-10]. An approach which combines hierarchical clustering and EM (expectation-maximization) algorithm [11] for maximum likelihood has been proposed by Dasgupta and Raftery [12]. This approach can give much better performance than existing methods.

1.4 The Proposed Approach

In this dissertation, a self-learning adaptive system has been proposed to detect, characterize and model events of interest through spatio-temporal analysis based signatures of engine parameters for performance evaluation. A detailed description of this system is presented in Chapter 2. The clusters of spatio-temporal information of specific measured signals from an engine are obtained during windowed periods of the desired operation. Three window periods are utilized, namely, the Data Acquisition window, the Processing for Event Detection window and the Model based Characterization & Update window.

To better improve the detection results and capture the time information of the signal, a wavelet transform will be performed on the raw data to do the feature extraction for later stage processing.

Since a prior knowledge about the events to be modeled is not available, an unsupervised learning is performed using a combined hierarchical fuzzy clustering algorithm to form the clusters. In addition, the time-frequency localization based features obtained during the first preprocessing stage are multi-dimensional, fuzzy membership functions need to be extended to deal with multi-dimensional feature space. The clusters are then translated into prospective events.

After learning the events, the system performs classification analysis using a nearest neighborhood method for further analysis and characterization. The detected events are displayed to the user for assessment. In case of inconsistencies or new prospective events, the user is provided with the time-history based information of the past and current occurrences of the prospective events. With interaction from the user, the sensitivity thresholds of the clusters can be modified accordingly. This leads to the property of adaptive learning of our system.

1.5 Organization of the Dissertation

The whole dissertation is divided into five chapters:

(1) Introduction;

(2) Background and Literature Review;

(3) Methodology;

(4) Results and Discussion and

(5) Conclusions and Future Work.

Chapter 2 presents a brief background and literature review. In Section 2.1 the various clustering algorithms are presented and their advantages and disadvantages are compared and contrasted. Section 2.2 describes the proposed Adaptive Learning for Event Modeling and Characterization (ALEC) system from a high-level view. The implementations of the ALEC system are detailed in Chapter 3. Each individual component is thoroughly examined and corresponding algorithm is proposed and evaluated. Chapter 4 shows the experimental results to prove the feasibility of implementation of the system and good performance of each algorithm applied. In Chapter 5, conclusions drawn from the studies done so far are given and the future work to be done is also presented.

CHAPTER 2

BACKGROUND AND LITERATURE REVIEW

2.1 Background

In this section, a brief summary of various techniques for event detection and data clustering is presented. The advantages and disadvantages of these techniques are mentioned. The summary presented is based on an extensive review of the literature. This section formulates the proposed approaches used in this dissertation.

2.1.1 Event and Event Detection

Without a set of rules, different individuals will have different notions of what constitutes an event. An event can be defined as something that happens at some specific time and place [13]. From this definition it can be seen that an event is associated with two important characteristics: time and place. Event-related philosophy concludes that two events are the same if they have the same spatio-temporal history, and that events are identical if they have the same causes and effects. Lombard [1;15] discusses why these properties are not sufficient conditions for event identity. He presents a model for events that includes the aspect of change, which he defines as " the addition or loss of properties." Event detection is heavily studied in the world of media in which the objective is to identify stories in several continuous news streams that pertain to new or previously unidentified events [5]. While in this dissertation the counterpart of the news streams is the time series signals (e.g., pressure measurement of the rocket engine). Event detection consists of two tasks: retrospective detection and on-line detection. The former entails the discovery of previously unidentified events in an accumulated collection, and the latter strives to identify the onset of new events from signal feeds in real-time. Detection of events is in fact a knowledge discovery problem, i.e., mining the data stream for new patterns in signals. Bottom-up signal clustering appears to be a natural solution for the discovery of natural clusters without introducing any assumptions about the domain. Moreover, bottom-up clustering can result in a cluster hierarchy, thus allowing observation at any level of abstraction in the information space. Higher levels of clusters give progressively coarse grain overviews of the content of signal groups, while lower levels provide tighter clusters corresponding to specific events, or temporal phases of events.

2.1.2 Pattern Classification and Cluster Analysis

Before the discussion of pattern classification and cluster analysis can be delved into, some definitions are given first to make the discussion easy.

(1) Definitions and Notation

Pattern -- A pattern (or feature vector, observation, or datum) χ is a single data item used by the clustering algorithm. It typically consists of a vector of d measurements: $\chi = (x_1, \ldots x_d)$.

Feature -- The individual scalar components x_i of a pattern χ are called features (or attributes).

Dimensionality -- d is the dimensionality of the pattern or of the pattern space.

Pattern set -- A pattern set is denoted $\aleph = \{\chi_1, \ldots \chi_n\}$. The ith pattern in $\aleph$ is denoted $\chi_i = (x_{i,1}, \ldots x_{i,d})$. In many cases, a pattern set to be clustered can be viewed as an n x d pattern matrix.

Class -- A class, in the abstract, refers to a state of nature that governs the pattern generation process. More concretely, a class can be viewed as a source of patterns whose distribution in feature space is governed by a probability density specific to the class. Clustering techniques attempt to group patterns so that the classes thereby obtained reflect the different pattern generation processes represented in the pattern set.

Hard clustering -- Hard clustering techniques assign a class label l_i to each pattern χ_i, identifying its class.

Fuzzy clustering -- Fuzzy clustering procedures assign to each input pattern χ_i a fractional degree of membership f_{ij} in each output cluster j.

Distance measure -- A distance measure is a metric on the feature space used to quantify the similarity of patterns.

(2) Pattern Classification

A traditional pattern classification system can be viewed as a mapping from input variables representing the raw data set to an output variable representing one of the categories or classes. Because of the curse of dimensionality [44], it is nearly always advantageous to apply pre-processing transformations to the raw data before it is fed into the classification system. Pre-processing usually involves feature selection and/or feature extraction. Feature selection is the process of identifying the most effective subsets of the original features to be used in the clustering while feature extraction is the use of one or more transformations of the input features to generate new salient features.

After the preprocessing and pattern representation are established, interpattern similarity should be defined on pairs of patterns and it is often measured by a distance function. Finally, the output of the grouping step is a collection of different clusters and it can be hard (a partition of the data into groups) or fuzzy where each pattern has a variable degree of membership in each of the output clusters. The following figure shows the schematic diagram of a typical classification system which underlies the proposed approaches presented in this dissertation.

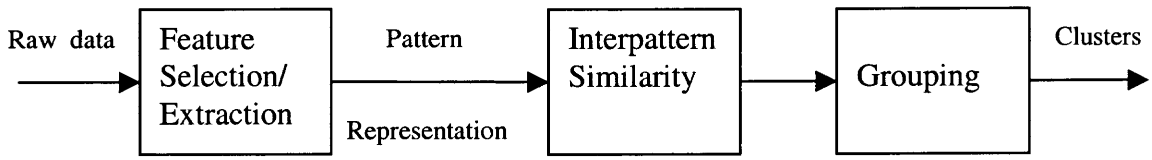


Figure 2.1 A typical classification system.

(3) Cluster Analysis

Cluster analysis is the organization of a collection of patterns (usually represented as a vector of measurements, or a point in a multidimensional space) into clusters based on similarity. Intuitively, patterns within a valid cluster are more similar to each other than they are to a pattern belonging to a different cluster.

Since similarity is fundamental to the definition of a cluster, a measure of the similarity between two patterns drawn from the same feature space is essential to most clustering procedures. Because of the variety of feature types and scales, the proper choice of distance measure is of great importance. It is most common to calculate the dissimilarity between two patterns using a distance measure defined on the feature space. Euclidean distance is the most popular metric and it is defined as:

$$d_2(x_i,x_j)=\left(\sum_{k=1}^{d}\left(x_{i,k}-x_{j,k}\right)^2\right)^{1/2}$$

$$=\ \left\|x_i-x_j\right\|_2 \qquad (2.1)$$

It is noted that Euclidean distance is actually a special case (p=2) of the Minkowski metric which is :

$$d_p(x_i,x_j)=\left(\sum_{k=1}^{d}\left(x_{i,k}-x_{j,k}\right)^p\right)^{1/p}$$

$$=\ \left\|x_i-x_j\right\|_p \qquad (2.2)$$

The Euclidean distance has an intuitive appeal as it is commonly used to evaluate the proximity of objects in two or three-dimensional space. It works well when a data set has "compact" or "isolated" clusters [16]. The drawback to the direct use of the Minkowski metrics is the tendency of the largest-scaled feature to dominate all others. Solutions to this problem include normalization of the continuous features or other weighting schemes. Linear correlation among features can also distort distance measures. This distortion can be alleviated by applying a whitening transformation to the data or by using the squared Mahalanobis distance:

$$d_M(x_i,x_j)=(x_i-x_j)A^{-1}(x_i-x_j)^T \qquad (2.3)$$

Where the patterns x_i and x_j are assumed to be row vectors and A is the sample covariance matrix of the patterns or the known covariance matrix of the pattern generation process. $d_M(x_i,x_j)$ assigns different weights to different features based on their variances and pairwise linear correlations. It is implicitly assumed here that class conditional densities are unimodal and characterized by multidimensional spread, i.e., that the densities are multivariate Gaussian. The regularized Mahalanobis distance was used in [16] to extract hyperellipsoidal clusters.

Traditional clustering algorithms can be classified into two main categories [2]: hierarchical and partitional. In hierarchical clustering, the number of clusters need not be specified in advance, therefore no problems due to initialization and local minima occur. It builds a cluster hierarchy or, in other words, a tree of clusters. Every cluster node contains child clusters; sibling clusters partition the points covered by their common parent. Such an approach allows exploring data on different level of granularity. Hierarchical clustering methods are divided into agglomerative and divisive [2;17]. An agglomerative clustering starts with one-point (singleton) clusters and recursively merges two or more appropriate clusters. A divisive clustering starts with one cluster of all data points and recursively splits the most appropriate cluster. The process continues until a stopping criterion, e.g., the desired number k of clusters, is achieved.

To merge or split subsets of points rather than individual points, the distance between individual points has to be generalized to the distance between subsets. Such derived proximity measure is called a linkage metric. The type of the linkage metric used significantly affects hierarchical algorithms, since it reflects the particular concept of closeness and connectivity. Major inter-cluster linkage metrics include single link, average link and complete link [18-19]. The underlying dissimilarity measure (usually distance) is computed for every pair of points with one point in the first set and another point in the second set. A specific operation such as minimum (single link), average (average link), or maximum (complete link) is applied to pair-wise dissimilarity measures:

$$d(C_1, C_2) = operation\{\ d(x, y) \mid x \in C_1,\ y \in C_2\ \} \qquad (2.4)$$

Early examples include the algorithm SLINK [20] which implements single link, Voorhees' method [21] which implement average link, and the algorithm CLINK [22] which implements complete link. Of the three prominent algorithms, SLINK is referenced the most which is related to finding the Euclidean minimal spanning tree and has $O(N^2)$ complexity.

All of the above linkage metrics can be derived as instances of the Lance-Williams updating formula[23].

$$d(C_i, C_j, C_k) = a(i)d(C_i, C_k) + a(k)d(C_j, C_k)$$
$$+ b\ d(C_i, C_j) + c\ |d(C_i, C_k) - d(C_j, C_k)| \qquad (2.5)$$

Here a, b, c are coefficients corresponding to a particular linkage. This formula expresses a linkage metric between the union of two clusters, C_i and $C_j,$ and a third cluster, C_k in terms of underlying components. The Lance-Williams formula has an utmost importance since it makes manipulation with dissimilarity computationally feasible. Jain and Dubes [2] introduced the original average link agglomeration algorithm – Group-Average Method.

Linkage metrics-based hierarchical clustering suffers from time complexity. Under reasonable assumptions, such as reducibility condition, linkage metrics methods have $O(N^2)$ complexity [19]. Ward [23] implements an agglomerative clustering based not on a linkage metric, but on an objective function used in k-means. The merger decision is made in terms of its effect on the objective function. Chiu et al. [24] proposed another hierarchical clustering algorithm using a model-based approach in which maximum likelihood estimates were introduced.

Traditional hierarchical clustering is inflexible due to its greedy approach: after a merge or a split is selected, it is not refined. In addition, since they consider only local neighbors in each step, they cannot incorporate a prior knowledge about the global shape or size of clusters. As mentioned in Chapter 1, they cannot always separate overlapping clusters. Moreover, hierarchical clustering is static, and points committed to a given cluster in the early stages cannot move to a different cluster.

A partitional clustering algorithm obtains a single partition of the data instead of a clustering structure, such as the dendrogram produced by a hierarchical technique. Partitional methods have advantages in applications involving large data sets for which the construction of a dendrogram is computationally prohibitive. The partitional techniques usually produce clusters by optimizing defined either locally (on a subset of the patterns) or globally (over all of the patterns). K-means [25] is the simplest and most commonly used algorithm employing a squared error criterion which is defined as:

$$e^2(\aleph, \ell) = \sum_{j=1}^{K} \sum_{i=1}^{n_j} \left\| x_i^{(j)} - c_j \right\|^2 \tag{2.6}$$

It starts with a random initial partition and keeps reassigning the patterns to clusters based on the similarity between the pattern and the cluster centers until a convergence criterion is met, e.g., there is no reassignment of any pattern from one cluster to another, or the squared error ceases to decrease significantly after some number of iterations. The k-means algorithm is popular because it is easy to implement and its time complexity is $O(N)$, where N is the number of patterns. A major problem with this algorithm is that it is sensitive to the selection of the initial partition and may converge to a local minimum of the criterion function value if the initial partition is not properly chosen. Bradley and Fayyad [6] suggested a way to mitigate the effects of cluster initialization.

One variation to the *k*-means algorithm is to permit the splitting and merging of the resulting clusters. Typically, a cluster is split when its variance is above a pre-specified threshold and two clusters are merged when the distance between their centroids is below another pre-specified threshold. Under such a scheme, it is possible to obtain the optimal partition starting from any arbitrary initial partition, provided proper threshold values are specified.

Another variation of the *k*-means algorithm involves selecting a different criterion function altogether. Diday [26] and Symon [27] described a dynamic clustering approach obtained by formulating the clustering problem in the framework of maximum-likelihood estimation. The regularized Mahakanobis distance was used in Mao and Jain [16] to obtain hyperellipsoidal clusters.

On the other hand, partitional clustering algorithms can be divided into two classes: crisp (or hard) clustering and fuzzy clustering. Hard clustering is the traditional approach in which each pattern belongs to one and only one cluster. Hence, the clusters in a hard clustering are disjoint. Fuzzy clustering extends this notion to associate each pattern with every cluster using a membership function [2]. Fuzzy set theory was initially applied to clustering in Ruspini [28]. The most popular fuzzy clustering algorithm is the fuzzy *k*-means (FCM) algorithm. A generalization of the FCM algorithm was proposed by Bezdek [29] through a family of objective functions. A fuzzy c-shell algorithm and an adaptive variant for detecting circular and elliptical boundaries was presented in Dave [30]. It was also extended in medical image analysis to segment magnetic resonance images [31]. Even though it is better than the hard k-means algorithm at avoiding local minima, FCM can still converge to local minima of the squared error criterion. The design of the membership function is the most important problem in fuzzy clustering; different choices include those based on similarity decomposition and centroids of clusters.

2.1.2 Wavelet Transform

The wavelet transform is a synthesis of ideas that emerged over many years from different fields, such as mathematics and signal processing. In order to better appreciate the role wavelet plays in the analysis, an introduction is briefly given here to the history of wavelets. In the early days of signal analysis, the Fourier transform proved to be an extremely useful tool, which broke down a signal into constituent sinusoids of different frequencies of interest. However, Fourier analysis has a serious drawback in that time information is lost when transforming from the time domain to the frequency domain. When the frequency representation of a signal is looked into, it is impossible to tell when a particular event took place. If the signal properties do not change much over time, this drawback may be ignored. However, most interesting signals like the ones used in this project contain numerous abruptly changing data points, and these changes are often the most important part of the signals . Therefore, Fourier analysis is not suited for detecting characteristic changes in the time-series signals.

The next step forward in correcting the Fourier's deficiency comes with the Short-Time Fourier Transform (STFT) [32]. This technique adapted the Fourier transform to analyze only a small section of the signal at a time. As a matter of fact, STFT maps a signal into a two-dimension function of time and frequency. The STFT represents a sort of compromise between the time- and frequency-based views of a signal. It tells some information about both when and at what frequencies a signal event occurs. This information is, nevertheless, obtained with limited precision, and that precision is determined by the size of the window. A major shortcoming with STFT is that the window size is fixed for all frequencies, once a particular size for the time window is chosen. In real applications, many signals require a variable window size in order to determine more accurately either time or frequency.

Wavelet transform allows the use of long time intervals where more precise low-frequency information is wanted, and shorter regions where high-frequency information is needed. Thus, one major advantage afforded by wavelets is the ability to perform local analysis – that is, to analyze a localized area of a larger signal. Thus, wavelet analysis is capable of revealing aspects of data that other signal analysis techniques miss, such as trends, breakdown points, and discontinuities in higher derivatives. Wavelet theory has been under intensive study during the last decade [32-35]. Scaling and shifting are two of the most important concepts in wavelet analysis. Scaling a wavelet simply means stretching (or compressing) it while shifting a wavelet simply means delaying (or hastening) its onset. Daubechies wavelets [32] are compactly orthonormal wavelets which make discrete wavelet analysis practicable. Wavelet analysis has seen numerous applications in statistics [36], time series analysis [37] and image processing [38]. Coifman et al. [39] have generalized the wavelet basis function to include a library of modulated waveform orthonormal bases called wavelet packets. Furthermore, Wavelet transform has been extensively used in data mining field [40] because of its many favorable properties, such as vanishing moments, hierarchical and multi-resolution decomposition structure, linear time and space complexity of the transformations, decorrelated coefficients and a wide variety of basis functions.

2.2 Adaptive Learning for Event Modeling and Characterization (ALEC)

By combining the techniques introduced above, an integrated approach can be developed to effectively monitor and characterize the event of interest of the propulsion system. The following figure gives a schematic representation of the operating windows of the proposed ALEC system using time-history based adaptive clustering of the selected measurements as well as wavelet based temporal frequency components.

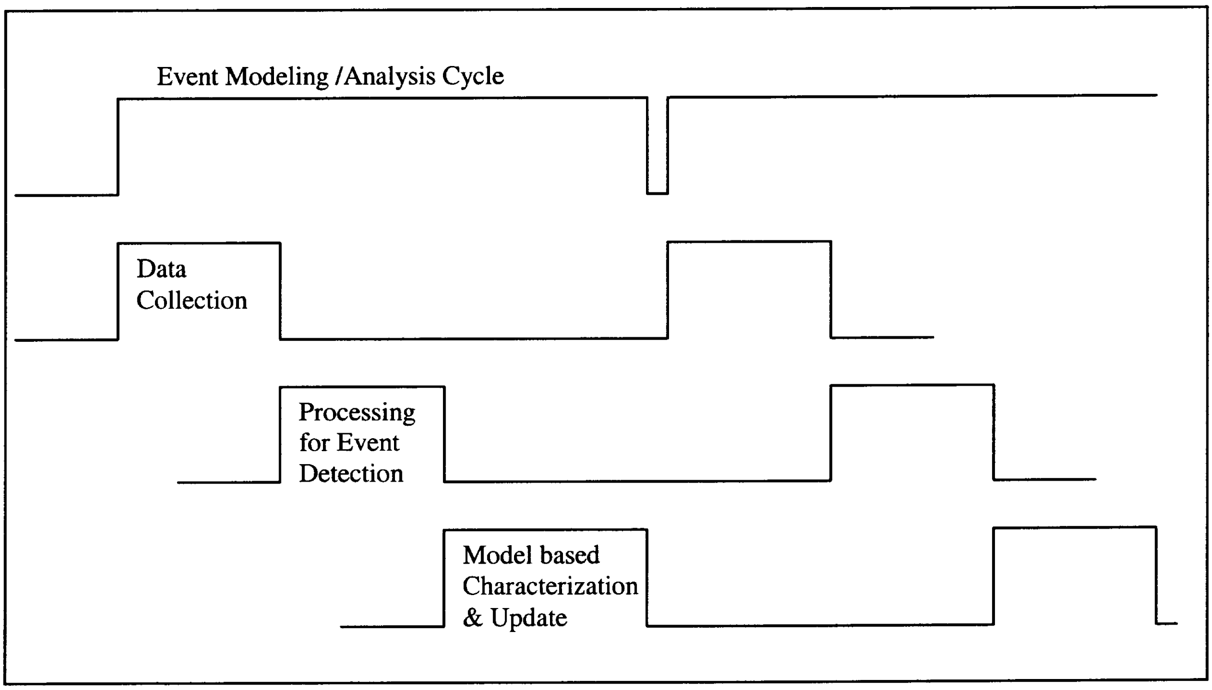


Figure 2.2 The time windows for the operations of the proposed ALEC system.

The figure above shows that one cycle consists of three major time windows.

Below is a brief explanation of the functional operation of each time window.

(a) Data Collection window: In this time window, the input signal is obtained. The signal should be collected over a time period that is large enough for accumulating sufficient number of samples to implement wavelet transform.

(b) Processing for Event Modeling time window: In this window, the time-frequency features are computed using the wavelet transform.

(c) Model-based Characterization and Update time window: In this time window, the time-frequency features are analyzed and clustered for characterization and the event signature models are updated accordingly.

The cycle of steps (a)-(c) is repeated for another sequential time window as the data acquisition is continued.

2.3 Summary

In this chapter, an extensive literature review is presented to demonstrate the state of the art of the approaches. With focus in algorithms directly applicable to the implementation and realization of the ALEC system, a clear understanding can be obtained as to how to combine the advantages of each major category of clustering algorithm to produce an effective and feasible framework to characterize and model the event of interests.

CHAPTER 3

METHODOLOGY

3.1 ALEC Method

In Section 2.2, the time windows are given and labeled to show how measurement and processing occurs at each stage. Next a schematic diagram of the proposed ALEC system during the training phase is shown as follows:

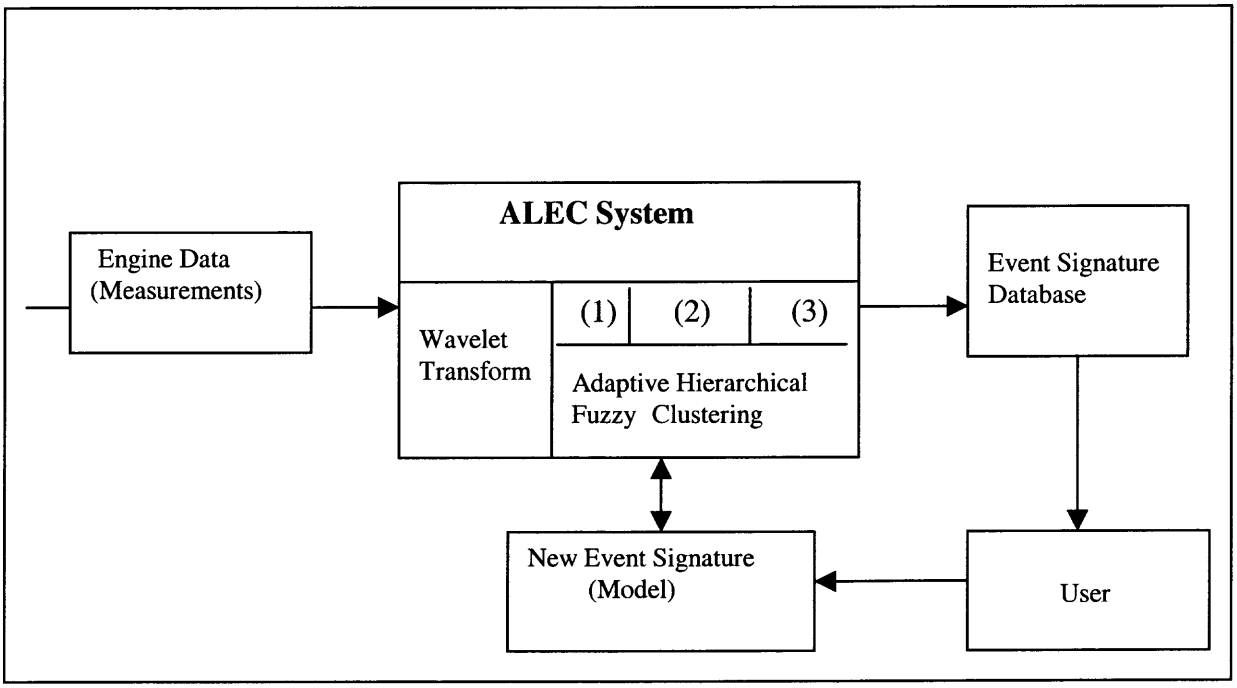


Figure 3.1 System diagram during the training phase of ALEC. (1) Absolute measurement (2) Wavelet decomposition features (3) Statistical features

As can be seen from the above, after the raw data is fed, a series of operations are performed to obtain the necessary measurements in order to be used in later processing. These measurements include (1) Absolute measurement, the raw value acquired at that particular instant; (2) Wavelet decomposition features which contains approximation and (3) Statistical features.

Here is the description of the processing stages during the training phase which roughly correspond to the blocks shown in Figure 3.1:

(a) The features in the wavelet domain are clustered using the hierarchical fuzzy clustering method described later in this chapter.

(b) Using a specific threshold on the clustered features, events are flagged and presented to the user for comments and categorization.

(c) Signatures of categorized events are stored in the model signature database.

(d) Other signature of the prospective events that have not been acknowledged by the user are stored in the event database for future references and correlation analysis.

3.2 Feature Extraction

Feature extraction is an essential part in any pattern recognition system and belongs to the preprocessing stage of a classification system. The choice of preprocessing will be one of the most significant factors in determining the performance of the final system. The features to be used in the ALEC system include direct measurement features and processed features.

3.2.1 Direct Measurement Features

In the modeling of event of interests of the propulsion system, the direct measurement is the pressure value taken at the pneumatic valve. Its unit is psia and sampled at certain frequency. Therefore, each measurement carries two pieces of relevant information: the pressure values which can be used as the elements of features in a later classification stage, and the timestamp. This timestamp will allow us to look into the time history of the signal when inconsistencies occur.

3.2.2 Processed Features - Wavelet Transform

Wavelet transform is the decomposition of a signal with a family of real orthonormal bases $\psi_{m,n}(x)$ obtained through translation and dilation of a kernel function $\psi(x)$, known as the mother wavelet, i.e.,

$$\psi_{m,n}(x) = 2^{-m/2}\psi(2^{-m}x - n) \tag{3.1}$$

where m and n are integers. Due to the orthonormal property, the wavelet coefficients of a signal $f(x)$ can be easily computed via

$$c_{m,n} = \int_{-\infty}^{+\infty} f(x)\psi_{m,n}(x)dx$$

and the synthesis formula

$$f(x) = \sum_{m,n} c_{m,n}\psi_{m,n}(x)$$

can be used to recover $f(x)$ from its wavelet coefficients.

To construct the mother wavelet $\psi(x)$, a scaling function may be first determined, which satisfies the two-scale difference equation [33 ; 41]

$$\phi(x) = \sqrt{2}\sum_{k} h(k)\phi(2x - k) \tag{3.2}$$

Then, the wavelet kernel $\psi(x)$ is related to the scaling function via

$$\psi(x) = \sqrt{2}\sum_{k} g(k)\phi(2x - k) \tag{3.3}$$

where

$$g(k) = (-1)^k h(1-k) \tag{3.4}$$

The coefficients $h(k)$ in (3.2) have to meet several conditions for the set of basis wavelet functions in (3.1) to be unique, orthonormal, and have a certain degree of regularity [41]. Several different sets of coefficients $h(k)$ satisfying the above conditions can be found in [32-33; 35; 42].

The coefficients $h(k)$ and $g(k)$ play a very crucial role in a given discrete wavelet transform. To perform the wavelet transform does not require the explicit forms of $\phi(x)$ and $\psi(x)$ but only depends on $h(k)$ and $g(k)$. Consider a J-level wavelet decomposition which can be written as

$$f_0(x) = \sum_k c_{0,k}\phi_{0,k}(x)$$

$$= \sum_k \left(c_{J+1,k}\phi_{J+1,k}(x) + \sum_{j=0}^{J} d_{j+1,k}\psi_{j+1,k}(x) \right) \tag{3.5}$$

where coefficients $c_{0,k}$ are given and coefficients $c_{j+1,n}$ and $d_{j+1,n}$ at scale $j+1$ are related to coefficients $c_{j,k}$ at scale j via

$$c_{j+1,n} = \sum_k c_{j,k} h(k-2n)$$

$$d_{j+1,n} = \sum_k c_{j,k} g(k-2n) \tag{3.6}$$

where $0 \le j \le J$. Thus, (3.6) provides a recursive algorithm for wavelet decomposition through $h(k)$ and $g(k)$, and the final outputs include a set of J-level wavelet coefficients

$d_{j,n}$, $1 \le j \le J$, and the coefficient $c_{J,n}$ for a low-resolution component $\phi_{J,k}(x)$. By using a similar approach, a recursive algorithm can be derived for function synthesis based on its wavelet coefficients $d_{j,n}$, $1 \le j \le J$, and $c_{J,n}$

$$c_{j,k} = \sum_n c_{j+1,n} h(k-2n) + \sum_n d_{j+1,n} g(k-2n) \qquad (3.7)$$

It is convenient to view the decomposition (3.6) as passing a signal $c_{j+1,n}$ through a pair of filters H and G with impulse response $\tilde{h}(x)$ and $\tilde{g}(x)$ and downsampling the filetered signals by two (dropping every other sample), where $\tilde{h}(x)$ and $\tilde{g}(x)$ are defined as

$$\tilde{h}(x) = h(-n), \quad \tilde{g}(x) = g(-n)$$

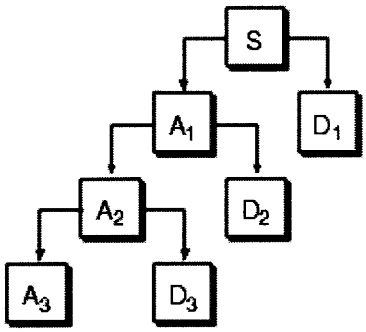
The pair of filters H and G correspond to the halfband lowpass and highpass filters, respectively, and are called the quadrature mirror filters in the signal processing literature. The reconstruction procedure is implemented by upsampling the subsignals c_{j+1} and d_{j+1} (inserting a zero between neighboring samples) and filtering with $h(n)$ and $g(n)$, respectively, and adding these two filtered signals together. Usually the signal decomposition scheme is performed recursively to the output of the lowpass filter $\tilde{h}$. It leads to the conventional wavelet transform or the so-called pyramid-structure wavelet decomposition.

In summary the wavelet transform decomposes the signal as a linear combination of weighted basis functions to provide frequency localization with respect to the sampling parameter such as time or space. The multi-resolution approach of the wavelet transform establishes a basic framework of the localization and representation of different frequencies at different scales.

Wavelet transforms may also used to smooth data. This is accomplished by applying a wavelet transformation to the noisy data, thresholding the resulting coefficients which are below some value in magnitude, and then inverse transforming to obtain a smoother version of the original data. This process has been coined Wavelet Shrinkage by Donoho and Johnstone [48-51].

Standard wavelet transforms have traditionally been implemented using Quadrature Mirror Filters. Since filters are used, this requires that the samples be uniformly spaced. There is an alternative wavelet transform based upon the idea of interpolating subdivision [53-54] which was created by Wim Sweldens [55-56]. Sweldens coined these wavelets as Second Generation Wavelets. These biorthogonal wavelets are constructed through a technique he called Lifting. Lifting is a method of increasing the number of vanishing moments of the decomposition function.

Daubechies wavelets [32;52] are compactly orthonormal wavelets which make discrete wavelet analysis practicable. The names of the Daubechies family wavelets are written *dbN*, where *N* is the order. In the dissertation *db*2 has been chosen for its good derivative property. On the other hand, wavelet analysis can be performed at multiple levels depending on the requirements of the applications. The following figure depicts a three-level decomposition of a signal:



$$
\begin{aligned}
S &= A_1 + D_1 \\
&= A_2 + D_2 + D_1 \\
&= A_3 + D_3 + D_2 + D_1
\end{aligned}
$$

Figure 3.2: A three-level wavelet decomposition tree, *A* means approximation and *D* means detail.

3.3 Fuzzy Hierarchical Clustering

Cluster analysis can be generally defined as decomposing or partitioning a data set into groups so that the points within one group are similar to each other and are as different as possible from the points in other groups. Currently there are three categories of solutions to this problem: those based on an attempt to find the optimal partition into a specified number of clusters, those based on a hierarchical attempt to discover cluster structure and those based on a probabilistic model for the underlying cluster [43]. In some cases, some prior knowledge is available to guide the process of clustering. Therefore, a general hierarchical clustering combined with supervised knowledge can be depicted in Figure 3.3.

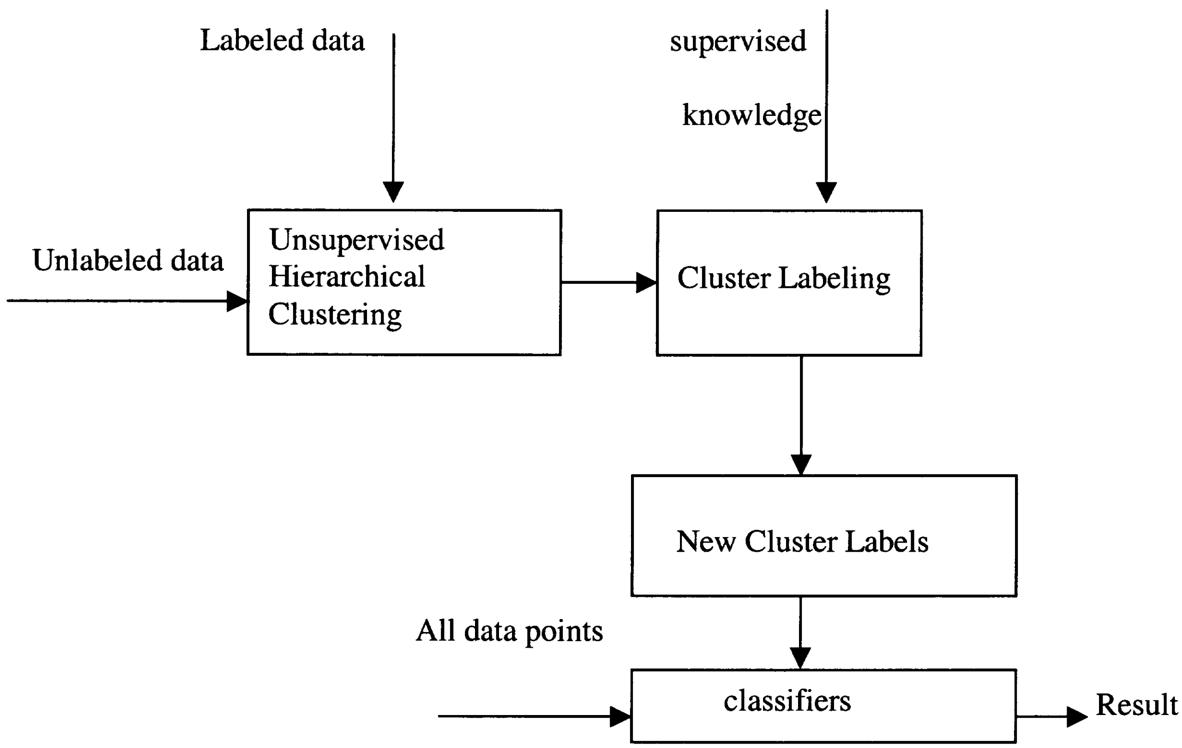


Figure 3.3 Hierarchical clustering with prior knowledge.

3.3.1 Hierarchical Clustering with the Agglomerative Method

There are generally two ways to perform hierarchical clustering, one is the agglomerative (which merges) and the other is the divisive (which divides). Hierarchical methods of cluster analysis permit a convenient graphical display in which the entire sequence of merging (or splitting) is shown. Because of its tree-like nature, the display has the name of dendrogram. The agglomerative method is usually chosen because it is more important and more widely used. One reason for the popularity of agglomerative method is that during the merging process the choice of threshold is not a big concern which will be illustrated in the details of the algorithm shown below. In contrast, divisive methods are more computationally intensive and the difficulty of choosing potential allocations to clusters during the splitting stages.

Agglomerative methods are based on measures of distance between clusters. Essentially, starting with an initial clustering, they merge those two clusters that are nearest, to form a reduced number of clusters. This is repeated until just one cluster is obtained. Usually the starting point for the process is the initial clustering in which each cluster consists of a single data point. Suppose that *n* sample points are to be clustered, the initial number of clusters will be *n* as well. Therefore, an agglomerative algorithm for clustering can be described as follows:

Assume there are n data points $D=\{x(1),\ldots,x(n)\}$ and a function $D(C_i, C_j)$ is defined as measuring the distance between two clusters C_i and C_j.

Algorithm 1. (Agglomerative Hierarchical Clustering)

Step 1: for $i=1,\ldots, n$ let $C_i = \{ X(i) \}$

Loop: While there is more than one cluster left do

Minimizing the distance $D(C_k, C_h)$ between any two clusters

Let C_i and C_j be the clusters with minimum distance

$C_i = C_i \cup C_j$;

Remove cluster C_j;

End

In the above algorithm, some consideration should be taken about the distance measure. Normally Euclidean distance is employed which assume some degree of commensurability between the different variables. It makes less sense if the variables are non-commensurate, that is, variables are measured in different units. This is the case in the study of the fuzzy classification of turbine blade fatigue modes. A common strategy is to standardize the data by dividing the sample value of each of the variables by its sample standard deviation, so that they are equally important [44]. This strategy is utilized in this dissertation to classify the turbine blade fatigue modes. Figure 3.4 shows a sample dendrogram.

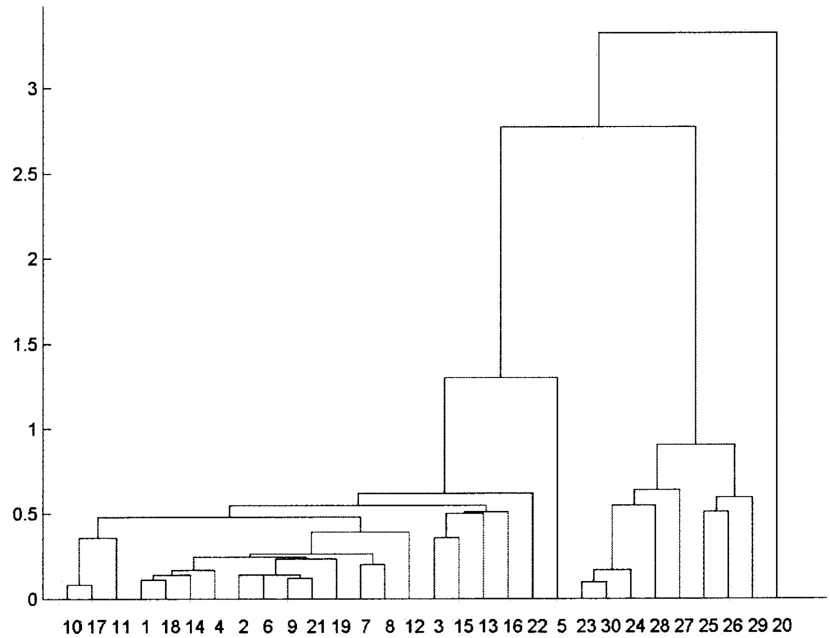


Figure 3.4 A sample dendrogram.

3.3.2 Fuzzy Classification

Conventional classification approaches always assign a new unidentified object into exactly one category by means of classifier constructed from the training data set. Even though they are suitable for various applications and have proven to be an important tool, they do not reflect the nature of human concepts and thoughts, which tend to be abstract and imprecise. In real world, to set a crisp boundary often makes the result intuitively unreasonable. It can be observed from Figure 4.1 that the engine mode categories are actually overlapping with each other, which implies that a particular engine may fall into two or more different categories. Thus the introduction of fuzzy logic into the realm of classification becomes necessary. Another better justification for employing fuzzy logic is to represent via membership functions the extent to which various fatigue modes might be present in an engine blade.

(1) Fuzzy Set and Membership Function

In contrast to a classical set, a fuzzy set is a set without a crisp boundary. If X is a collection of objects denoted generically by x, then a fuzzy set A in X is defined as a set of ordered pairs:

$$A = \{(x, \mu_A(x)) \mid x \in X\} \qquad (3.8)$$

where $\mu_A(x)$ is called the membership function (MF) for the fuzzy set A and its value ranges from 0 to 1. In short, a membership function can be viewed as a curve that defines how each point in the input space is mapped to a membership value (or degree of membership) between 0 and 1. The input space, or X in the definition, is sometimes referred to as universe of discourse which may consist of discrete objects or continuous space. Next one example is shown to illustrate the concept of membership function. Suppose a researcher is given a task of defining a person as "middle-aged". Always in practice, a range of ages, say, between 40 and 50, is considered as "middle-aged". This statement can be expressed in a mathematical way. Here the generalized Cauchy distribution is used to specify the MF :

$$\mu_A(x) = bell(x; a, b, c) = \frac{1}{1 + \left|\frac{x-c}{a}\right|^{2b}} \qquad (3.9)$$

where c is the median value of the range, in this example it is 45, a and b are parameters to adjust the width and sharpness of the curve. Now the curve of membership function is drawn as $\mu_A(x) = bell(x;15,3,45)$ in Figure 3.5.

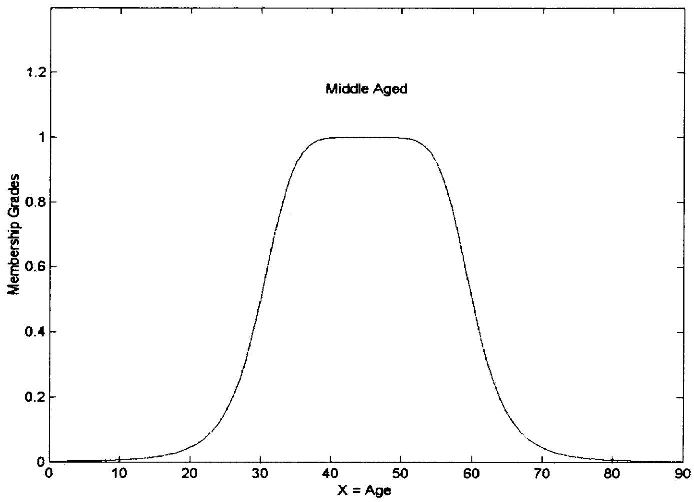


Figure 3.5 A plot of the Bell membership function *bell(x; 15, 3, 45).*

As shown above, it can be seen that the definition of "middle aged" is very natural without a crisp boundary. If your age is between 40 and 50 you MF value is 1 which is considered middle-aged. If you are 35 years old, you are more likely considered to be middle-aged than to be considered as young because your MF value is around 0.8. This model can be extended into a more meaningful one in which X, the universe of discourse is partitioned into several fuzzy sets whose MFs cover X in a more or less uniform manner. The following figure shows three MFs that define a person as "young", "middle aged" or "old".

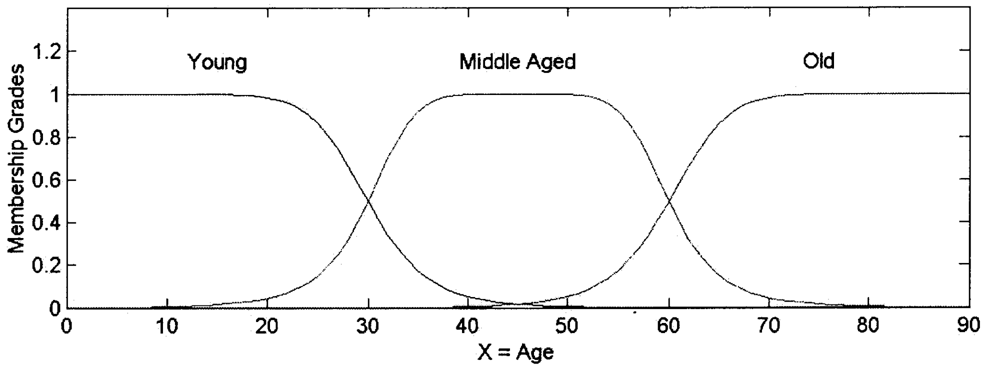


Figure 3.6 A plot of three Bell MFs for "young", "middle aged" and "old".

Therefore, a particular age has three corresponding MF values in different categories. As mentioned before, the three MFs totally cover the value range of X and the transition from one MF to another is smooth and gradual.

(2) Membership Function Formulation

The classes of parameterized functions used to define functions include the following: Triangular MFs, Trapezoidal MFs, Gaussian MFs and Bell MFs, as used above. In the project of classifying blade fatigue modes, the Gaussian MF is used with the assumption that the sample engines are selected from a population with Gaussian distribution. In addition, the one-dimensional Gaussian MF needs to be extended to multi-dimensional Gaussian MF since in the analysis each engine has many parameters which are correlated with each other. This means that the multi-dimensional Gaussian MF is not simply a multiplication of MF of each individual variable. Therefore, a need arises to derive the following formula to calculate the multi-dimensional Gaussian MF.

$$\mu_A(X) = gaussian(X;M,K) = \exp\{-\frac{1}{2}(X-M)^T K^{-1}(X-M)\} \qquad (3.10)$$

Where X and M are column vectors defined by:

$X = [x_1, x_{2,} ... x_n]^T$ and $M = [m_1, m_{2,} ... m_n]^T = [E(x_1), E(x_2), ... E(x_n)]^T$

m_i is the mean value of variable x_i.

K is covariance matrix of variables x_i, which is defined as:

$$K = \begin{bmatrix} \mathrm{var}(x_1) & \mathrm{cov}(x_1, x_2).... & \mathrm{cov}(x_1, x_n) \\ \mathrm{cov}(x_2, x_{1)} & \mathrm{var}(x_2).... & \mathrm{cov}(x_2, x_n) \\ \mathrm{cov}(x_n, x_1) & \mathrm{cov}(x_n, x_2).... & \mathrm{var}(x_n) \end{bmatrix} \qquad (3.11)$$

(3) Fuzzy *k*-means Clustering

The fuzzy *k*-means algorithm [45] is based on minimization of the following objective function, with respect to *U*, a fuzzy *K*-partition of the dataset, and to *V*, a set of *K* prototypes:

$$J_q(U,V) = \sum_{j=1}^{N} \sum_{i=1}^{K} (u_{ij})^q d^2(X_j, V_i); K \le N \quad (3.12)$$

where q is any real number greater than 1, X_j is the j-th m-dimensional feature vector, V_j is the centroid of the ith cluster, u_{ij} is the degree of membership of X_j in the ith cluster. $d^2(X_j, V_i)$ is any inner product metric (distance between X_j and V_j), N is the number of data points. K is the number of clusters. The parameter q is the weighting exponent for u_{ij} and controls the "fuzziness" of the resulting clusters [29].

Fuzzy partition is carried out through an iterative optimization of (3.12) according to [45]:

1) Choose primary centroid V_i (prototypes);

2) Compute the degree of membership of all feature vectors in all the clusters:

$$u_{ij} = \frac{(1/d^2(X_j, V_i))^{1/(q-1)}}{\sum_{k=1}^{K}(1/d^2(X_j, V_i))^{1/(q-1)}} \qquad (3.13)$$

3) Compute new centroids $\hat{V}_i$:

$$\hat{V}_i = \frac{\sum_{j=1}^{N}(u_{ij})^q X_j}{\sum_{j=1}^{N}(u_{ij})^q} \qquad (3.14)$$

and update the degree of membership, u_{ij} to $\hat{u}_{ij}$, according to (3.13).

4) if $\max[\left|u_{ij} - \hat{u}_{ij}\right|] < \varepsilon$ stop, otherwise goto Step 3

where ε is a termination criterion between 0 and 1.

Computation of the degree of membership u_{ij} depends on the definition of the distance measure, $d^2(X_j,V_i)$ [46].

$$d^2(X_j,V_i)=(X_j-V_i)^T A(X_j-V_i) \qquad (3.15)$$

The inclusion of A (an $m \times m$ positive-definite matrix) in the distance measure results in weighting according to the statistical properties [29]. In the dissertation, Euclidean distance which is the default measurement for fuzzy k-means algorithm was used. Therefore, A equals the identity matrix.

3.4 Competitive Learning and Classification of Events

Clustering algorithms such as *k*-means and hierarchical (introduced above) clustering typically have all data present before clustering begins. However, our case is not applicable to this assumption because the data or the raw signal arrives continuously. Under this condition, two problems may be run into: (1) there is a short supply of memory to store all the patterns themselves (this problem would not occur if the machine had large amounts of computer memory) or (2) the clusters need to be used even before the full data are present which is exactly the case studied here [46].

In order to evaluate the performance of the clustering algorithm, some criteria must be established in guiding the clustering process and comparing different schemes. Here the simple yet useful sum-of-squared-error criterion is used , which is briefly stated as follows. Let D be the set of n samples, that is, $D = \{ x_1 , x_2 \ldots x_n \}$. Suppose that these samples needs to be partitioned into exactly c distinct subsets $D_1, D_2 \dots D_c$. Each subset is to represent a cluster. On the other hand, let n_i be the number of samples in D_{i1} and let m_i be the mean of those samples.

$$m_i = \frac{1}{n_i} \sum_{x \in D_i} x \qquad (3.16)$$

Then the sum-of-squared errors is defined by

$$J_e = \sum_{i=1}^{c} \sum_{x \in D_i} \| x - m_i \|^2 \qquad (3.17)$$

Our goal is to try to minimize J_e. This type of clustering is often called *mimimum variance partitions.*

The other important question related to on-line clustering is the stability or convergence problem. Sometimes a cluster structure may be present which is unstable and continually wander and drift. Under such condition, the most recently acquired piece of information can cause major reorganization. The reason for this problem is that the clustering is guided by the global criterion one of which is discussed above. Every sample can have an influence on the location of a cluster center, regardless of how remote it might be. Some scheme must be come up with to overcome this influence and confine the learning adjustments to the cluster that is most similar to the pattern currently being presented. As a result, the characterizations of previously discovered clusters that are not related to the current pattern are not disrupted. The scheme is referred to *competitive learning*. In the ALEC project, a leader-follower clustering algorithm belonging to the category of competitive learning has been implemented. The algorithm is described as follows which alter only the cluster center most similar to a new pattern being presented. Let w_{i1} denote the current center for cluster i, let η denote a learning rate and let θ denote a threshold.

Algorithm 2. (Leader-Follower Clustering)

Begin initialize η and θ

$X \rightarrow W_1$

Do accept new X

Calculate the fuzzy membership function and find nearest cluster j

if $\| X - W_j \| < \theta$

then $W_j + \eta X \rightarrow W_j$

else add new X-> W

normalize weight

until no more patterns

return $W_1, W_2, \ldots$

End

In the project of event modeling and characterization of engine pressure the following events of interest are defined and intended to be characterized and detected. Here a brief introduction to each of them is given so that a better understanding of how the engine works can be gained.

(1) PV07 Commanded Open

(2) PV02 Commanded Open/PV02 Open Command Discontinued

(3) SV02 Commanded Open/SV02 Open Command Discontinued

The symbol PV07 is a pneumatic valve immediately downstream of the RP-1 tank that is opened to allow the propellant to begin "bleeding" into the engine feedline; the symbol PV02 is a second pneumatic valve downstream of the RP-1 tanks that must be opened to allow propellant into the engine.

The symbol SV02 represents the pressurization valve that is opened to pressurize the tanks from approximate 23psia to 53psia.The system goes into a controller process to maintain the tank pressure from 53psia to 47psia. The overshoot is due to the rate of pressurization and the very small amount of ullage initially in the tank.

3.5 Principal Component Analysis

In one project conducted in this dissertation, a population of 17 engines were provided and the dimension of variables was as much as 28. The detail for this project will be detailed in Chapter 4. Because of the curse of dimensionality, it is nearly always advantageous to apply preprocessing transformation to the raw data before it is fed into the classification system. This comes to the use of Principal Component Analysis (PCA). The central idea of PCA is to reduce the dimensionality of a data set which consists of a large number of interrelated variables while retaining as much as possible of the variation present in the data set [47].

The goal here is to map vectors X^d in a d-dimensional space $(x_1, x_2 ... x_d)$ onto vectors Z^M in an M-dimensional space $(z_1, z_2 ... z_M)$ where M<d. Without loss of generality we express vector X as a linear combination of a set of d orthonormal vectors u_i

$$X = \sum_{i=1}^{d} x_i u_i \qquad (3.18)$$

Where the vectors u_i satisfy the orthonormality relation

$$u_i^T u_j = \delta_{ij} \qquad (3.19)$$

Therefore the coefficient in (3.18) can be expressed as

$$x_i = u_i^T X \qquad (3.20)$$

Now suppose that only a subset of M<d of the basis vectors u_i are to be retained, so that only M coefficients x_i are used. In general, PCA does not retain a subset of the original set of basis vectors. It finds a new set of basis vectors that spans the original d-dimensional space such that the data can be well represented by a subset of these new basis vectors. Here v_i is used to denote the new basis vectors which meet the orthonormality requirment . As above, only M coefficients x_i are used and the remaining coefficients will be replaced by constants b_i. Now each vector x is approximated by an expression of the form

$$\widetilde{X} = \sum_{i=1}^{M} x_i v_i + \sum_{i=M+1}^{d} b_i v_i \tag{3.21}$$

$$x_i = v_i^T X \tag{3.22}$$

An attempt to choose the basis vectors v_i and the coefficients b_i is to be made such that the approximation given by (3.21), with the values of x_i determined by (3.22), gives the best approximation to the original vector X on average for the whole set data set. Then next step is to minimizes the sum of squares of errors over the whole data set. The sum-of-square error can be written as follows:

$$E_M = \frac{1}{2} \sum_{i=M+1}^{d} v_i^T A v_i \tag{3.23}$$

Where A is the covariance matrix of the set of vectors X^n, which is defined as follows:

$$A = \sum (x^n - \overline{x})(x^n - \overline{x})^T \tag{3.24}$$

Now the problem is converted to minimizing E_M with respect to the choice of basis vectors v_i. A minimum value is obtained when the basis vectors satisfy the following condition.

$$Av_i = \beta_i v_i \tag{3.25}$$

Thus v_i (i=M+1...d) are the eigenvectors if the covariance matrix. Note that, since the covariance matrix is real and symmetric, its eigenvectors can indeed be chosen to be orthonormal. Finally the minimum of error is in the form:

$$E_M = \frac{1}{2} \sum_{i=M+1}^{d} \beta_i \tag{3.26}$$

Therefore, the minimum error is achieved by rejecting the (d-M) smallest eigenvalues and their corresponding eigenvectors. The first M largest eigenvalues are then retained. Each of the associated eigenvectors v_i is called a *Principal Component.*

In the theory of matrix, an algorithm called *Singular Value Decomposition* (SVD) can be employed to calculate the eigenvalues and its corresponding eigenvectors. The use of SVD has two important implications. First it is computationally efficient and second it provides additional insight into what a PCA actually does. It also provides a way to represent the results of a PCA both graphically and analytically.

After the exploration of the concept of PCA and how it can be obtained, another question occurs naturally: how the number of PCAs or a subset of the original variables will be selected, that is, how to determine the dimension of M in the Z^n. There are several approaches to this problem [47]. In the project of classifying engine blade fatigue modes, the approach employed is to choose a subset of the original variables after discarding the variables which contribute below the threshold of variance (0.02 in the case) in the data set .

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Fuzzy Classification of Engine Blade Fatigue Modes

In this section, a task has been carried out to classify engine blade fatigue modes using a generalized multi-dimension Gaussian membership function. First the problem description is given, then Principal Component Analysis is applied for feature extraction. Finally results and discussion are presented.

4.1.1 Problem Description

The purpose of this work is to investigate the combined use of statistical analysis and fuzzy logic techniques in developing a model for turbine blade fatigue to the higher pressure turbine of an aircraft engine. To investigate this task, a data set was provided that contained flight history information for a set of engines whose turbines were warranty-repaired by Pratt and Whitney for a reason other than that of blade damage, which corresponds to the normal incremental wear and tear of engine parts. The predetermined categories of blade fatigue are oxidation/erosion (OE) related, thermo-mechanical (TMF) related and other reason related as shown in the following figure. Figure 4.1 shows that some engine blade falls into more than one category which results in an overlapping structure. In order to address this problem, generalized multi-dimensional Gaussian membership function is then formulated so that its validity can be verified.

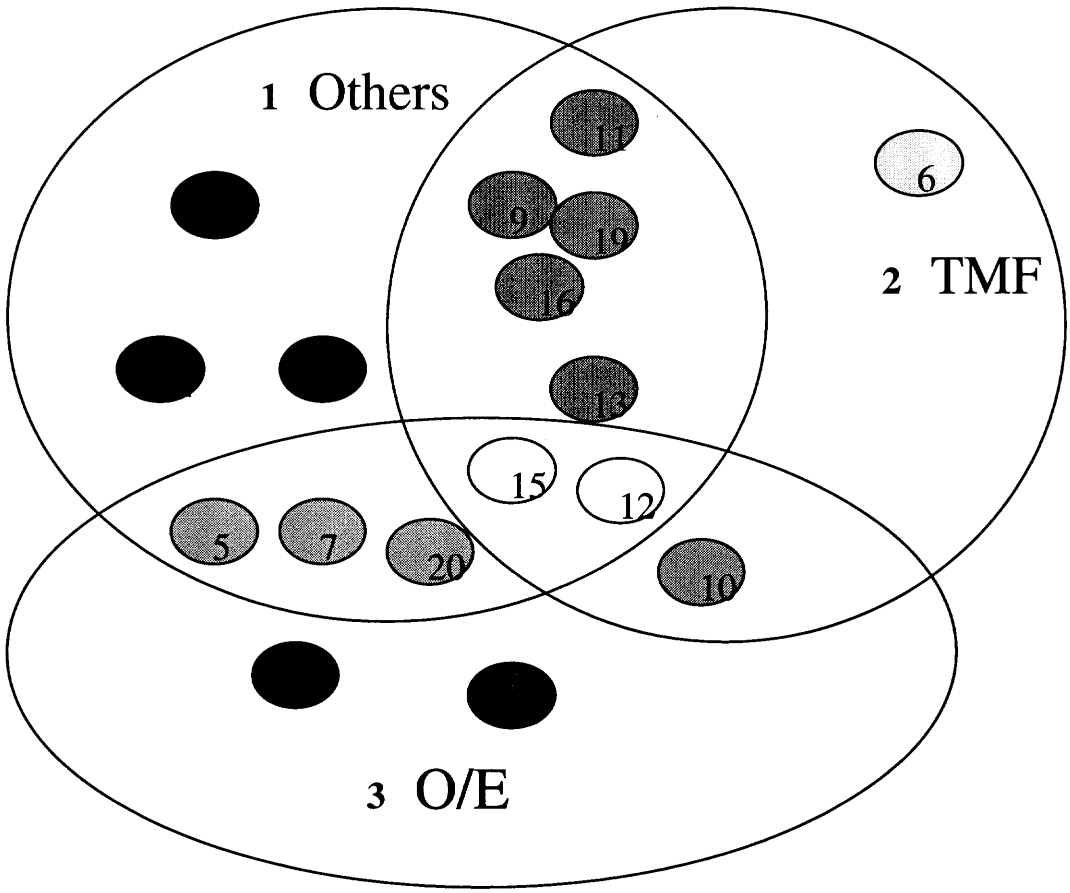


Figure 4.1 Predefined engine blade fatigue modes.

As can be seen from Figure 4.1, the fatigue modes for the first 17 engines have been predetermined by the domain expert in a manner which contains overlapping structures. For each engine, the flight history data was provided in a text file which contains 28 parameters shown in the following table:

Table 4.1 A List of 28 Engine Variables and Their Description

Parameter name[1]	Parameter description	Engineering units	Notes	Data recorded in Take-off files	Data recorded in Cruise files
ACID1	XXAircraft Id	Code	Lowest 4 Bytes	x	x
ACID2	XXAircraft Id	Code	Highest 4 Bytes	x	x
DATE	Date	MM/DD/YY		x	x
GMT	Greenwich Mean time	(hhmm)		x	x
MACH	MACH Number	mach		x	x
TAT	Total Air Temperature	deg.C		x	x
IAS	Indicated Air Speed	knots		x	x
ALT	Altitude (Pressure)	ft.		x	x
SAT	Static Air Temperature	deg.C			x
DEPART	XXDeparture	City-Code		x	x
ARRIVE	XXDestination	City-Code		x	x
N1	Low Pressure Rotor Speed	xx.x %	100%=2990rpm	x	x
N2	High Pressure Rotor Speed	xx.x %	100% = 10800rpm		x
WF	Fuel Flow	PPH			x
TLA	Throttle Lever Angle	deg.	(-45.0 to 90.0)		x
T2	Engine Inlet Temperature	deg.C			x
T25	LPC Exit Temperature	deg.C			x
T3	Compressor Exit Temp	deg.C			x
EGT	Exhaust Gas Temperature	deg.C		x	x
P25	LPC Exit Pressure	psia			x
PB	Burner Pressure	psia			x
P5	Exhaust Gas Pressure	psia		x	x
P2	Engine Inlet Pressure	psia		x	x
PAMB	Ambient Pressure	psia			x
TCAPOS	Turbine Cooling Air Position	% open	(-5.0 to 105.0)	x	x
CYCINS	Cycles since installation	nnnnn		x	x
HRSINS	Hours since installation	nnnnn		x	x
TCATMP	TCA Temperature	deg.C			

1: the appearance of –999 in data field indicates that the data was not recorded.

The data from these engines is contained in two files: cruise data and take off data designated by engine_*xx*cr.txt for cruise data and engine_*xx*to.txt. The '*xx*' in the file name refers to the engine number. Here is a general description of each of the variables. ACID1 and ACID2 are aircraft identification values coded in hexadecimal for the highest (ACID2) and lowest (ACID1) 4 bytes of the identification code. Actually, this was an arbitrary code employed by United Technologies Research Center (UTRC) team to de-identify flight data per the agreement with the airline that owns the data. These values only changed if the engine was placed on a different aircraft. DATE is the date when the data point was collected and recorded. GMT is the Greenwich Mean Time corresponding to the data point. MACH is the mach number that is an indication of speed. TAT is the total air temperature. IAS is the indicated air speed. ALT is the altitude measured in feet above sea level. SAT is the static air temperature at a specified point in flight. DEPART and ARRIVE are the letter codes for the city of departure and arrival respectively. N1 is the measure of the low pressure rotor speed given as percentage of full speed where 100% is 2990rpm. N2 is a measure of the high pressure rotor speed given as a percentage of full speed where 100% is 10800rpm. WF is the fuel flow to the engine measured in pounds per hour. TLA is the throttle lever angle measured in degrees ranging from –45.0 to 90.0. T2 is the engine inlet temperature, T25 is the LPC exit temperature, T3 is the compressor exit temperature, and EGT is the exhaust gas temperature, all measured in degrees Celsius. P25 is a measure of the LPC exit pressure in units of psia. PB is the burner pressure measured in units of psia. P5 is a measure of the exhaust gas pressure in units of psia. P2 is a measure of the engine inlet pressure in units of psia. PAMB is a measure of ambient pressure in units of psia. TCAPOS is a measure of the cooling air position given as a percentage of full open ranging from –5% to 105.0%. CYCINS is a calculation of cycles since installation measured in cycle counts. TCATMP is a measure of the turbine cooling air temperature measured in units of degrees Celsius.

With respect to the number of parameters which is 28 in this case, the total number of sample engines is only 122 including the pre-classified 17 engines. In addition, some parameters, such as ACID1, ACID2, DEPART, ARRIVE, don't contribute anything to the final analysis of the data set. Thus, without the reduction of dimensionality by eliminating some variables, the result of classification would be extremely poor and error prone.

The PCA algorithm was implemented in the Matlab environment and the original variables ALT, N1, P2 , TCAPOS and TAT were finally chosen.

4.1.2 Preprocessing and Results

(1) Some engines have one or two abnormal (isolated) data point(s) for some parameters. We plot the ALT parameter for engine 13 and 14:

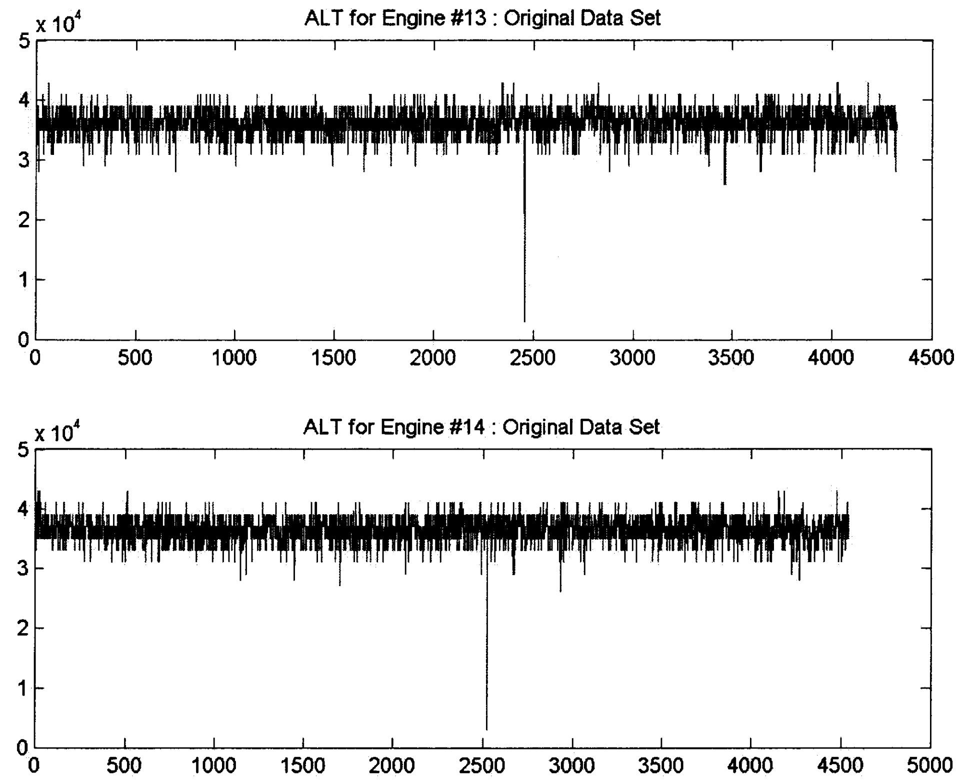


Figure 4.2 Some engine blades have isolated data point(s).

Figure 4.2 shows that there are abrupt drops in the ALT (altitude) parameter. By looking into the data files themselves, a fact was identified that the ALTs at the 2454th data point for engine 13 and at the 2523rd for engine 14 both have a value of 3000 while on average ALT is at least 35,000 ft. Therefore these isolated data points needs to be discarded.

(2) Filling missing data values

Some parameters for all engines have a few missing values which must be taken care of in order to have correct result. The following plot illustrates this problem.

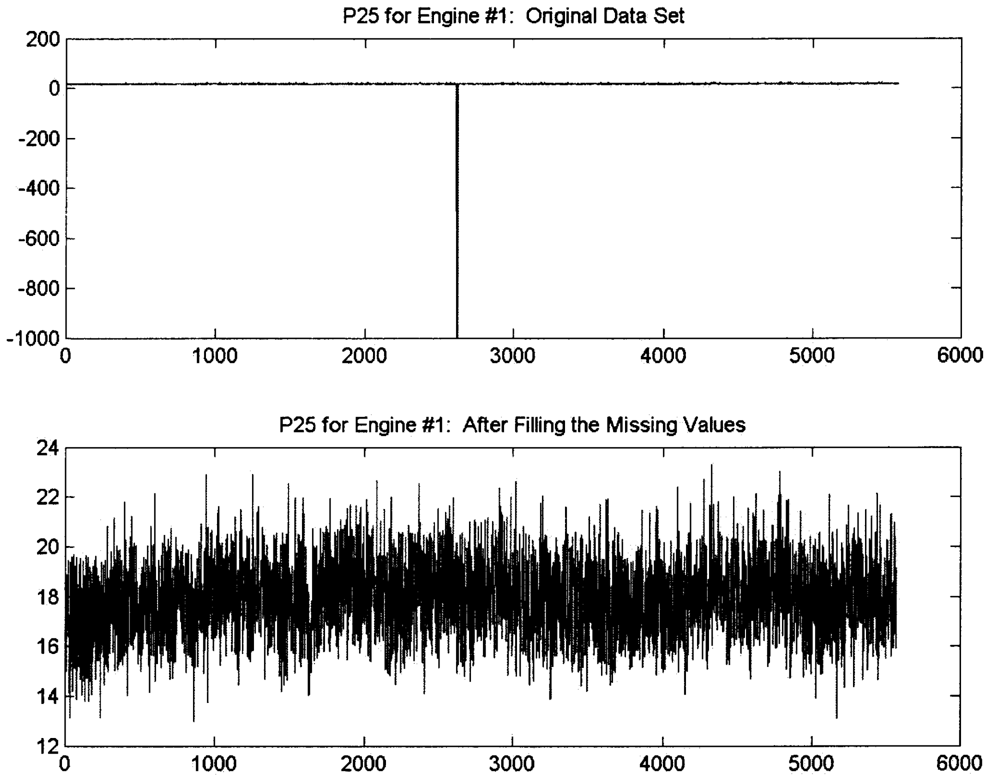


Figure 4.3 Some engine blades have missing data values.

At the first glance of the plot for this original data set, people might think the P25 (LPC Exit Pressure) parameter for engine #1 may be safely ignored because almost every data point has a value of zero so that it would have zero mean and zero variation. Again, the data file was looked into and it was found out that this is not the case. Almost every data point for P25 parameter contains a value of around 20. Therefore, there must be a few isolated points that prevent the proper plot of the data points. A careful examination of data file confirmed that only a few data points (2620th, 2621st, 2622nd and 2626th) contain missing values as indicated by a value of –999, which means that data is unavailable. The action taken to remedy this situation is to fill the missing value with the average value of other valid data points. After fixing this problem, a plot of the P25 parameter for engine #1 was drawn again in Figure 4.3 as a comparison , which was the true representation of the data set.

(3) Too few data points available for some engines.

By examining the data files, it was also found that there are quite a few engines containing too few data points which will to some extent affect the interpretation of the final categorization. The following plot shows this problem:

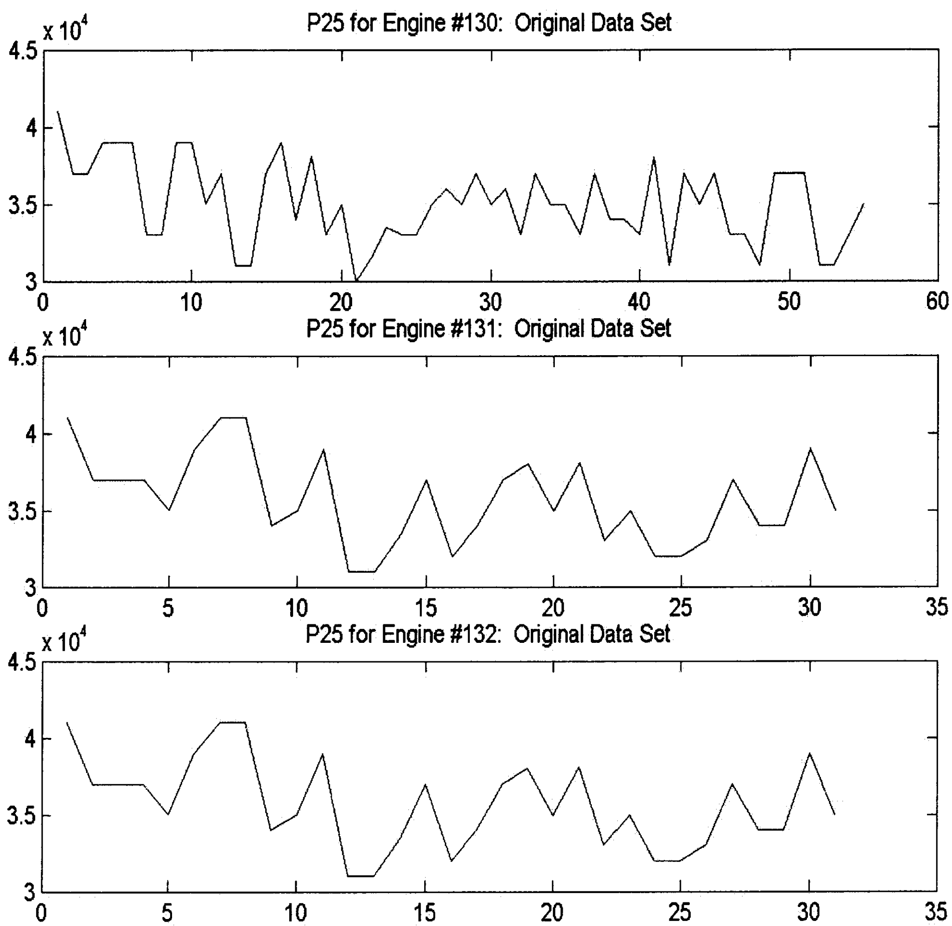


Figure 4.4 Some engine blades have too few data points.

After the inspection of data file for these engines indicated above, it was found out that engine # 130 contains 55 data points while engine #131 and # 132 only contains 31 data points. It was noted that there were seven engines which contained fewer than 500 data points while 80% of the engines contained at least 2,000 data points. For such engines, no special action is taken. These engines are still used in the principal component analysis to find the variables which contributed most to the variation of the data.

(4) Fuzzy Classification

After the preprocessing outlined above was carried out, the following procedures were used to formulate the MFs for each category of fatigue modes.

(a) Given Figure 4.1, three categories of fatigue modes were defined as follows:

TMF = {6, 9, 10, 11, 12, 13, 15, 16, 19};
OE = {4, 5, 7, 10, 12, 15, 17, 20};
OTHER = {1, 5, 7, 8, 9, 11, 12, 13, 14, 15, 16, 19, 20};

(b) For each category (which serves as fuzzy set) the mean value of each of feature variables chosen (the ratio of its mean to standard deviation of each variable) were calculated and their corresponding covariance matrix was obtained using Equation (3.11);

(c) For each category the MF using Equation (3.10) was formulated;

(d) For each unidentified engine, its feature value vector was first calculated and substitute for X in Equation (3.10) for each category's MF obtained in Step C. Thus for each engine three MF values corresponding to respective category would be obtained.

The following table shows the result obtained from the above procedure.

Table 4.2 Fuzzy Classification Result of Engine Blade Fatigue Modes

Engine#	MF_TMF	MF_OE	MF_OTHER
1	0.8936	0.4114	0.8480
4	0.1827	0.6966	0.6878
5	0.0008	0.2991	0.1019
6	0.2368	0.0007	0.4810
7	0.0001	0.1603	0.0513
8	0.0626	0.0000	0.2341
9	0.2536	0.0000	0.1257
10	0.3430	0.0985	0.2161
11	0.1864	0.0000	0.2609
12	0.1675	0.1313	0.5966
13	0.2723	0.0396	0.4494
14	0.0026	0.0000	0.0680
15	0.2272	0.2823	0.4883
16	0.7409	0.0166	0.7315
17	0.0019	0.4389	0.1358
19	0.2085	0.0014	0.0926
20	0.2512	0.5140	0.7413
21	0.0178	0.0002	0.2217
22	0.6335	0.4073	0.6360
23	0.1996	0.0000	0.2046
24	0.0327	0.0000	0.0340
25	0.0011	0.1990	0.1119
26	0.7240	0.0189	0.7341
27	0.3481	0.0002	0.5253

28	0.0575	0.5401	0.4607
29	0.7801	0.0385	0.7438
30	0.0000	0.0000	0.0000
31	0.0000	0.0000	0.0005
32	0.0476	0.0492	0.3053
33	0.0129	0.0000	0.0489
34	0.4487	0.9738	0.9183
35	0.2830	0.1621	0.2625
36	0.0155	0.0000	0.1052
37	0.5773	0.1643	0.5724
38	0.2144	0.7175	0.6366
39	0.5604	0.0001	0.3360
40	0.1682	0.0000	0.1781
41	0.0004	0.0511	0.0549
42-132	0.0000	0.0000	0.0000

4.1.3 Discussion

If the MF values for engine #1,#4-17,#19 and #20 were compared with the pre-classification as shown in Figure 4.1, it can be shown that the calculated MF values basically reflect the categories it belong to. For example, Engine #12 has a value of 0.1675 for TMF mode, of 0.1313 for OE mode and of 0.5966 for OTHER mode. Also it has been shown that it is more meaningful to compare the relative MF values to determine its category(s) than to use a fixed threshold. For example, engine #7 it has MF values of 0.0001, 0.1603 and 0.0513 for TMF, OE and OTHER respectively. By looking at Figure 4.1, it showed that Engine #7 falls into categories of OE and OTHER.

Therefore the calculated MF values agree with the classifications even though none of these MF values is even larger than 0.5 which seems a good threshold. Only two engines are exception to this rule. Engine #1 falls exclusively into the category of OTHER as shown in Figure 4.1, while calculated MF values are *0.8936* for TMF, 0.4114 for OE and *0.8480* for OTHER. Similarly, Engine #6 falls exclusively into the category of TMF while calculated MF values are *0.2368* for TMF, *0.4810* for OTHER and 0.0007 for OE.

All the engines (#21- #132) cab be classified. It seems that most of engines have MF values of 0.0000 giving an impression that they are not classified. As a matter of fact, they have extremely smaller MF values. In the code of our program, a format was imposed that the precision was restricted to the fourth digit after the decimal point. In the command window of Matlab environment, the precise MF values can be displayed before they are formatted and written to the text file for later reading. For example, the MF values for engine #132 are 3.3900e-294 for TMF, 5.0918e-319 for OE and 2.3753e-135. In this case, if none of MF values is significant, this engine will be classified as normal.

4.2 Event Detection and Modeling of Engine Pressure Data

4.2.1 Problem Description

Simulated and real data sets are both provided by the NASA Glenn Research Center. The event list is also included with the data set. The purpose of the ALEC system is to try to detect those events with raw data as the input. In the first place, the ALEC system doesn't have any priori knowledge about the signatures of the events to be characterized.

As described in Chapter 3, a combination of wavelet analysis, unsupervised learning and adaptive fuzzy clustering is to be used to achieve this goal. For the purpose of performance comparison between the proposed approach and the traditional hierarchical clustering one, results from each of these two approaches are to be presented for both simulated and real data sets.

4.2.2 Results on Simulated and Real Data

(1) Results on Simulated Data

Two sets of raw signals are provided for the purpose of analysis and clustering. The signals are sampled at 40ms for a certain time period starting at 0ms and ending at 97,799,660ms. Therefore, there is a total of 244,5000 samples in each data set. Due to the volume of data and limited computing capacity of the computer system available, partition of data has to be performed first in order for the signal to be feasibly analyzed, The signal was segmented every 5,000 data points.

Along with the raw signals, a list of events shown below are predefined and used in the refinement of the parameters of the algorithms.

Table 4.3 Event List from the Last Segment of Simulated Signal (5000 data points)

Time (Relative)	Time (Absolute)
T-00:04:20:900	9519100
T-00:04:20:740	9519260
T-00:04:50:430	9549570
T-00:04:50:250	9549750
T-00:03:19:190	9580810
T-00:03:19:000	9581000
T-00:03:48:170	9611830
T-00:03:47:970	9612030
T-00:02:17:330	9642670
T-00:02:17:120	9642880
T-00:02:46:550	9673450
T-00:02:46:330	9673670
T-00:01:15:760	9704240
T-00:01:15:530	9704470
T-00:01:44:940	9735060
T-00:01:44:700	9735300
T-00:00:13:940	9766060
T-00:00:13:680	9766320

Matlab's Toolboxes such as Wavelet, Statistics and Fuzzy Logic were combined to implement our algorithms outlined below. For the purpose of illustration, the focus was given to the last segment which contained the most events as observed from the event list. This segment started at a time of 9,580,000ms and ended at 9,779,960ms. It was sampled at 25Hz and had a total of 5,000 data points.

First a plot of the original signal (the last segment) was drawn and then a three-level wavelet decomposition using db2 wavelet was performed. After that, a detail reconstruction at level 1, level 2 and level 3 was carried out. Figure 4.6 shows the reconstruction of approximation A3 and details at three levels.

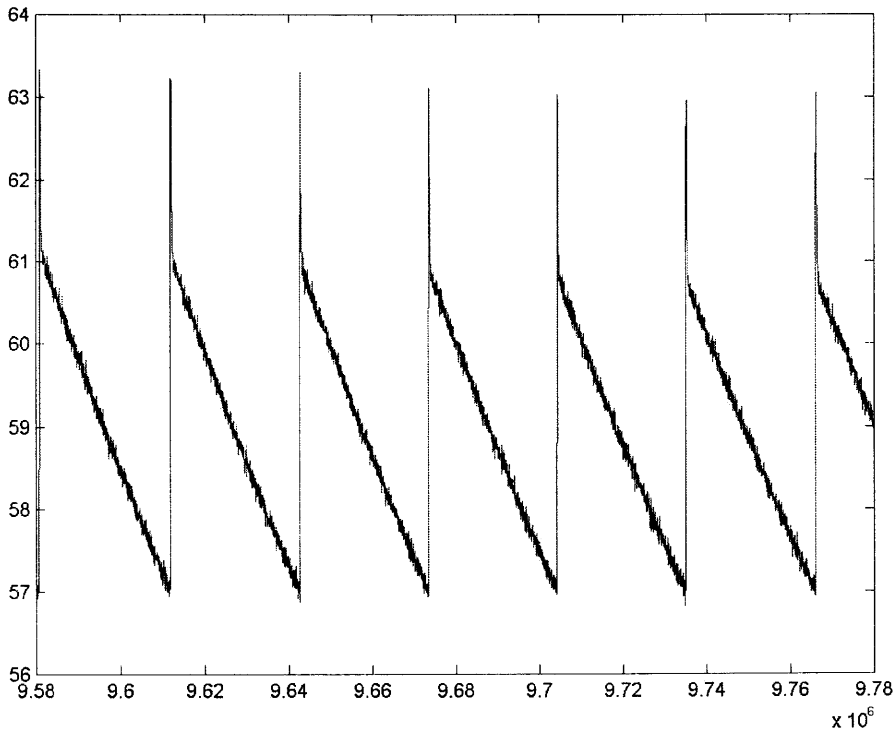


Figure 4.5 The last segment of the simulated signal 2.

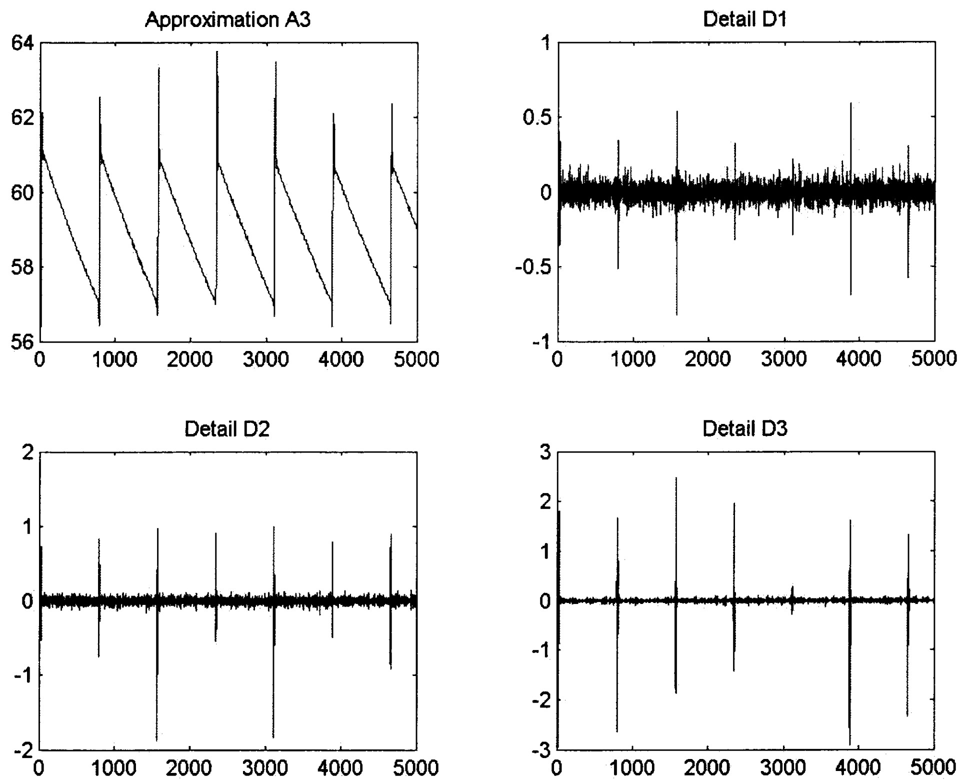


Figure 4.6 A plot of approximation A3 and details D1, D2 and D3 after three-level *db2* wavelet decomposition is performed on simulated signal 2.

The following procedures were performed step by step to obtain the final result of event modeling and characterization.

(1) Get the first leg of the segment, say, 1000 samples.

(2) Perform three-level wavelet decomposition and form the feature vectors after normalization.

(3) Do hierarchical clustering to find initial clusters and initialize database using the predefined event list.

(4) For incoming signals do fuzzy k-means clustering, calculate the membership function with previous clusters and add new clusters accordingly based on leader-follower algorithm and flag the event if it is not in the database.

(5) Repeat Step 4 until all the signal has been processed.

The initial clustering results for the last segments of the two sets of simulated signals are shown on the next few pages. The time values and their corresponding data values are listed only for clusters which contains no more than 10 data points. Otherwise, a symbol of "*X*" is placed. For the purpose of comparison, for each raw signal, two different clustering results are shown, one with the hierarchical clustering of the whole segment when all the signals are present, the other with the adaptive hierarchical fuzzy clustering after wavelet transform is performed.

Table 4.4 Clustering Result for Last Segment of Simulated Signal 1 (10 Clusters)
(Number of data points=5000, *X* means omitted
Algorithm: Hierarchical Clustering)

Cluster	# of data points in this cluster	Time(sec.)	Data Value	Average	Max	Min
#1	4988	*X*	*X*	58.994944	63.327534	56.828144
#2	3	9611800	57.048904	58.606861	61.822487	56.949192
		9673800	61.822487			
		9766040	56.949192			
#3	1	9704240	56.987427	56.987427	56.987427	56.987427
#4	1	9742640	57.008797	57.008797	57.008797	57.008797
#5	1	9642840	62.800018	62.800018	62.800018	62.800018
#6	1	9642880	63.113914	63.113914	63.113914	63.113914
#7	1	9642800	62.333115	62.333115	62.333115	62.333115
#8	1	9642920	63.307735	63.307735	63.307735	63.307735
#9	1	9642720	58.128113	58.128113	58.128113	58.128113
#10	2	9642680	56.870113	57.319447	57.768780	56.870113
		9704280	57.768780			

Table 4.5 Clustering Result for Last Segment of Simulated Signal 1 (15 Clusters)
(Number of data points=5000, *X* means omitted
Algorithm: Adaptive Hierarchical Fuzzy Clustering)

Cluster	# of data points in this cluster	Time(sec.)	Data Value	Average	Max	Min
#1	4972	*X*	*X*	58.995594	63.327534	56.828144
#2	2	9673720	62.930683	62.44018	62.930683	61.949677
		9704360	61.949677			
#3	1	9704520	62.907265	62.907265	62.907265	62.907265
#4	10	*X*	*X*	57.788401	61.559910	56.955296
#5	2	9611840	57.044876	57.087919	57.130962	57.044876
		9766080	57.130962			
#6	1	9642760	60.840183	60.840183	60.840183	60.840183
#7	2	9673800	61.822487	58.606861	61.822487	56.949192
		9766040	56.949192			
#8	1	9704240	56.987427	56.987427	56.987427	56.987427
#9	1	9642640	57.008797	57.008797	57.008797	57.008797
#10	1	9642840	62.800018	62.800018	62.800018	62.800018
#11	1	9642880	63.113914	63.113914	63.113914	63.113914
#12	1	9642800	62.333115	62.333115	62.333115	62.333115
#13	1	9642920	63.307735	63.307735	63.307735	63.307735
#14	1	9642720	58.128113	58.128113	58.128113	58.128113
#15	2	9642680	56.870113	57.319447	57.768780	56.870113
		9704280	57.768780			

Table 4.6 Clustering Result for Last Segment of Simulated Signal 2 (10 Clusters)
(Number of data points=5000, *X* means omitted
Algorithm: Hierarchical Clustering)

Cluster	# of data points in this cluster	Time(sec.)	Data Value	Average	Max	Min
#1	4983	*X*	*X*	58.994944	63.327534	56.828144
#2	5	9643080	60.728146	57.913474	60.728146	56.947231
		9673520	57.261398			
		9704240	56.947231			
		9735120	57.192970			
		9766160	57.437626			
#3	2	9581000	59.248848	59.323622	59.398396	59.248848
		9612040	59.398396			
#4	2	9581080	60.340618	60.347149	60.353680	60.340618
		9612120	60.353680			
#5	1	9673640	58.776394	58.776394	58.776394	58.776394
#6	2	9581160	60.869476	60.8111095	60.869476	60.752743
		9612200	60.752743			
#7	2	9580840	57.054195	57.118973	57.183750	57.054195
		9611880	57.183750			
#8	1	9704360	57.719246	57.719246	57.719246	57.719246
#9	1	9704280	57.076664	57.076664	57.076664	57.076664
#10	1	9673560	57.665771	57.665771	57.665771	57.665771

Table 4.7 Clustering Result for Last Segment of Simulated Signal 2 (15 Clusters)
(Number of data points=5000, *X* means omitted
Algorithm: Adaptive Hierarchical Fuzzy Clustering)

Cluster	# of data points in this cluster	Time(sec.)	Data Value	Average	Max	Min
#1	4972	*X*	*X*	58.956684	61.180935	56.942448
#2	1	9673800	60.397797	60.397797	60.397797	60.397797
#3	3	9581120	60.717632	60.350489	60.717632	59.757690
		9612160	60.576145			
		9704520	59.757690			
#4	2	9581040	59.973118	59.973904	59.974689	59.973118
		9612080	59.974689			
#5	1	9642760	57.541943	57.541943	57.541943	57.541943
#6	4	9580800	56.989128	57.120558	57.388290	56.957870
		9611840	56.957870			
		9704320	57.388290			
		9766120	57.146942			
#7	5	9643080	60.728146	57.9134742	60.728146	56.947231
		9673520	57.261389			
		9704240	56.947231			
		9735120	57.192970			
		9766160	57.437626			
#8	2	9581000	59.248848	59.323622	59.398396	56.248848
		9612040	59.398396			
#9	2	9581080	60.340618	60.347149	60.353680	60.340618
		9612120	60.353680			
#10	1	9673640	58.776394	58.776394	58.776394	58.776394
#11	2	9581160	60.869476	60.811110	60.869476	60.752743
		9612200	60.752743			
#12	2	9580840	57.054195	57.118973	57.183750	57.054195
		9611880	57.183750			
#13	1	9704360	57.719246	57.719246	57.719246	57.719246
#14	1	9704280	57.076664	57.076664	57.076664	57.076664
#15	1	9673560	57.665771	57.665771	57.665771	57.665771

To better view and interpret the result, the event detection outcome can be tabulated in a more readable format from the result of adaptive hierarchical fuzzy clustering algorithm on simulated signal 2.

Table 4.8 A Combination of Event List and Detected Events for Simulated Signal 2

Event #	Event Time	Detected Time	Cluster #
1	9580810	9580800	6
		9580840	12
2	9581000	9581000	8
		9581040	4
		9581080	9
		9581120	3
		9581160	11
3	9611830	9611840	6
		9611880	12
4	9612030	9612040	8
		9612080	4
		9612120	9
		9612160	3
		9612200	11
5	9642760	9642760	5
6	9642880		
		9643080	7
7	9673450	9673520	7
		9673560	15
8	9673670	9673640	10
		9673800	2
9	9704240	9704240	7
10	9704470	9704280	14
		9704320	6
		9704360	13
		9704520	3
11	9735060	9735120	7
12	9735300		
13	9766060	9766120	6
		9766160	7
14	9766320		

(2) Result on Real Data

Two sets of measurements (signals) are provided. They differ only in end time and sample rate. One set starts at 9,229,000ms and ends at 9410040ms, which is sampled at 25Hz and contains a total of 4527 samples. The other one starts at 9,229,000ms and ends at 9410020ms, which is sampled at 50Hz and contains a total of 9052 samples. Within each set, there are three measurement labeled as MPRE301P, MPRE302P and MPRE105P. In this report, the focus is given to the one with 50Hz sample rate and a label of MPRE301P. The following is the plot of the corresponding signal.

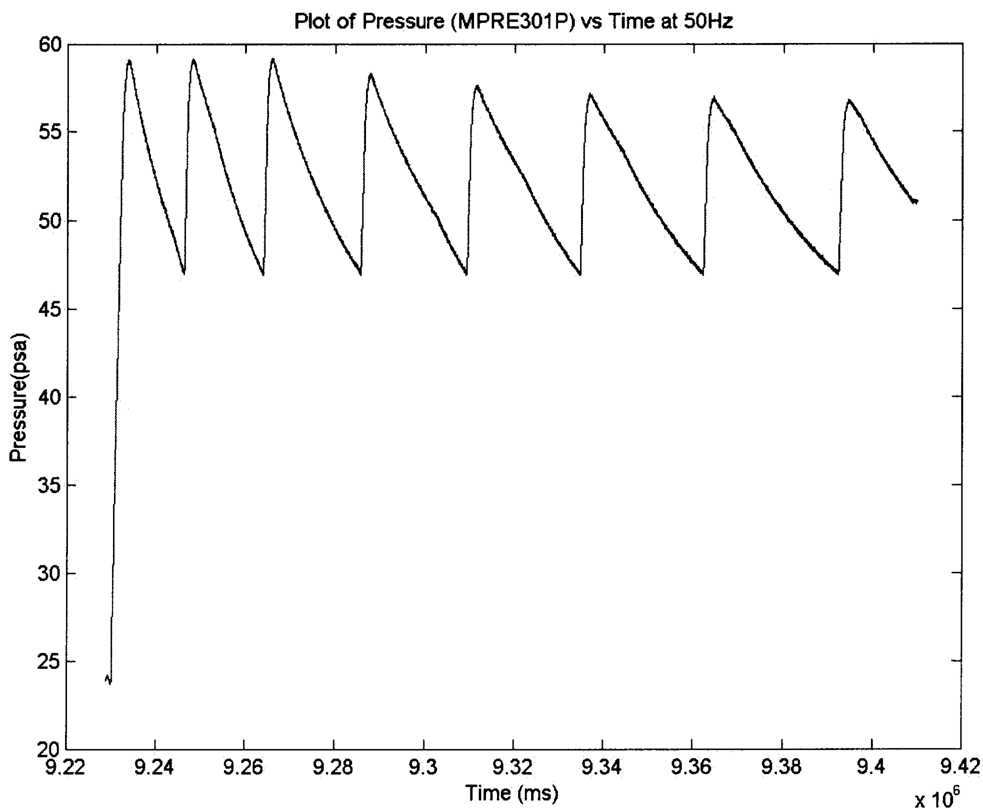


Figure 4.7 The last segment of the real signal (MPRE301P at 50Hz).

As can be seen in Figure 4.7, there is no overshooting issue as observed in the simulated data set shown in Figure 4.5. In the previous data sets, the signals were taken from a simulation of LOX tank ullage pressure which produced an artificial overshoot that quickly resolved, but caused the development system to key in on that feature. For the purpose of illustration, only the last data segment will be reported here in detail, although each segment has similar results. This data segment starts at a time of 9,310,040 ms and ends at 9,410,020ms. It is sampled at 50Hz and has a total of 5,000 data points.

A three-level wavelet decomposition using db2 wavelet was first applied to the data segment. After the decomposition, a detail reconstruction at level 1, level 2 and level 3 was performed to verify the decomposition results and this step is not absolutely required except for the purpose of verification. Figure 4.8 shows the reconstruction of approximation A3 and details at three levels.

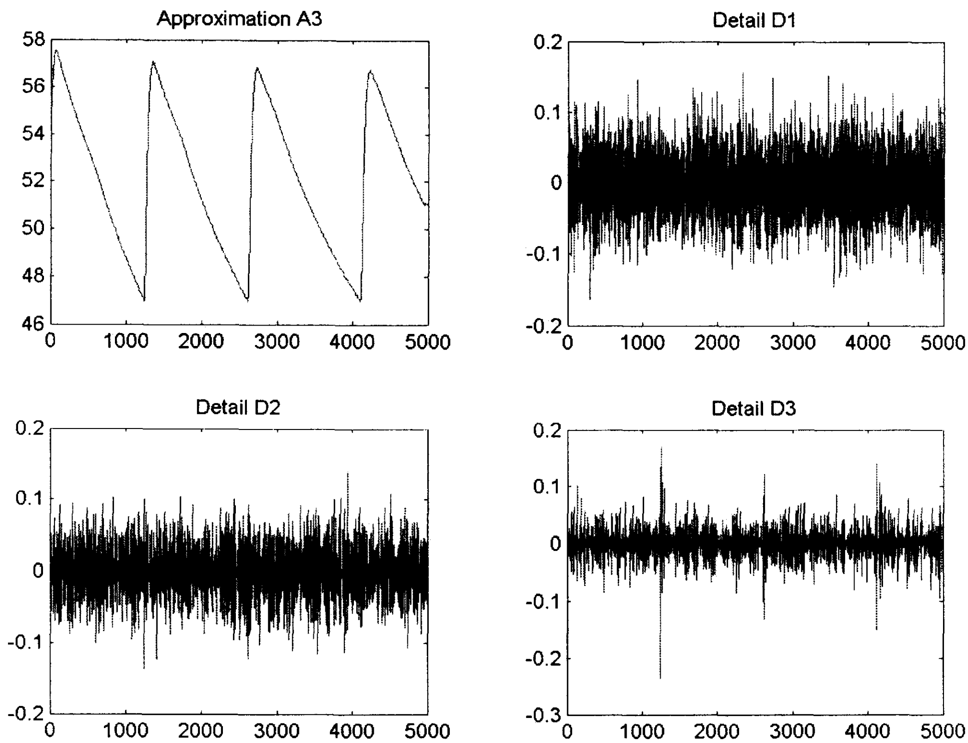


Figure 4.8 A plot of approximation A3 and details D1, D2 and D3 after the three-level db2 wavelet decomposition is performed on real signal.

Initial clustering results for the last segment with the label of MPRE301P and sample rate of 50Hz are presented here. Table 4.9 shows the expected events from the system being monitored corresponding to the last segment. In Table 4.10 and 4.11, the initial clustering results are reported for this last data segment. In the table, corresponding time and data values are listed only for clusters which contain no more than 10 data points. Otherwise, a symbol of "X" is placed in the "Time" and "Data Value" fields. For the purpose of comparison, for each raw signal, different clustering results are shown, one with the traditional hierarchical clustering of the whole segment when all the signals are present, and the other with the adaptive hierarchical fuzzy clustering.

Table 4.9 Event List from the Last Segment of Real Signal (5,000 data points)

Flight Time	Time (msec)	Major Event	Minor Event
T-00:07:25:190	9334810		SV02 Commanded Open
	9334860		MMSW102X indicates SV02 Open
T-00:07:24:500	9335500		SV02 Open Command Discontinued
	9335510		MMSW102X loses SV02 Open Indication
T-00:07:57:730	9362270		SV02 Commanded Open
	9362320		MMSW102X indicates SV02 Open
T-00:07:56:980	9363020		SV02 Open Command Discontinued
	9363030		MMSW102X loses SV02 Open Indication
T-00:06:27:820	9392180		SV02 Commanded Open
	9392230		MMSW102X indicates SV02 Open
T-00:06:26:980	9393020		SV02 Open Command Discontinued
	9393030		MMSW102X loses SV02 Open Indication

Table 4.10 Clustering Result for Last Segment of Real Signal at 50Hz (7 Clusters)
(Number of data points = 5000, *X* means omitted
Algorithm: Traditional Hierarchical Clustering)

Cluster	# of data points in this cluster	Time(sec.)	Data Value	Average	Max	Min
#1	4992	X	X	49.268860		
#2	2	9335500	52.982689	53.077349	53.172009	52.982689
		9363020	53.172009			
#3	1	9311640	57.372505	57.372505	57.372505	57.372505
#4	1	9388660	48.039326	48.039326	48.039326	48.039326
#5	2	9362340	47.249926	47.093082	47.406769	47.093082
		9362380	47.406769			
#6	1	9334860	47.211018	47.211018	47.211018	47.211018
#7	1	9334920	47.434807	47.434807	47.434807	47.434807

Table 4.11 Clustering Result for Last Segment of Real Signal at 50Hz (13 Clusters)
(Number of data points=5000, *X* means omitted
Algorithm: Adaptive Hierarchical Fuzzy Clustering)

Cluster	# of data points in this cluster	Time(sec.)	Data Value	Average	Max	Min
#1	4985	X	X	47.268860	57.638893	
#2	3	9335500	52.982689	53.10098	53.172009	52.982689
		9363020	53.172009			
		9393020	53.145596			
#3	1	9311640	57.372505	57.372505	57.372505	57.372505
#4	1	9341660	55.035538	55.035538	55.035538	55.035538
#5	1	9388660	48.039326	48.039326	48.039326	48.039326
#6	1	9362340	47.093082	47.093082	47.093082	47.093082
#7	1	9392220	47.045036	47.045036	47.045036	47.045036
#8	1	9334860	47.211018	47.211018	47.211018	47.211018
#9	1	9334920	47.434807	47.434807	47.434807	47.434807
#10	1	9362380	47.406769	47.406769	47.406769	47.406769
#11	1	9392300	47.41877	47.41877	47.41877	47.41877
#12	2	9335520	53.209351	53.283785	53.358219	53.209351
		9363040	53.358219			
#13	1	9334900	47.209438	47.209438	47.209438	47.209438

To better view and interpret the result, the event detection outcome can be tabulated in a more readable format from the result of adaptive hierarchical fuzzy clustering algorithm on real signal.

Table 4.12 A Combination of Event List and Detected Events for Real Signal

Event #	Event Time	Detected Time	Cluster #
1	9334810	X	X
	9334860	9334860	8
		9334900	13
		9334920	9
2	9335500	9335500	2
		9335520	12
3	9362270	X	X
	9362320	9362340	6
		9362380	10
4	9363020	9363020	2
		9363040	12
5	9392180	X	X
	9392230	9392220	7
			11
6	9393020	9393020	2
	9393030	X	X

Additional Events (not shown in the event list but detected in our clustering analysis)

9311640	3
9341660	4

A more intuitive presentation of the event detection result can be graphically drawn in Figure 4.9.

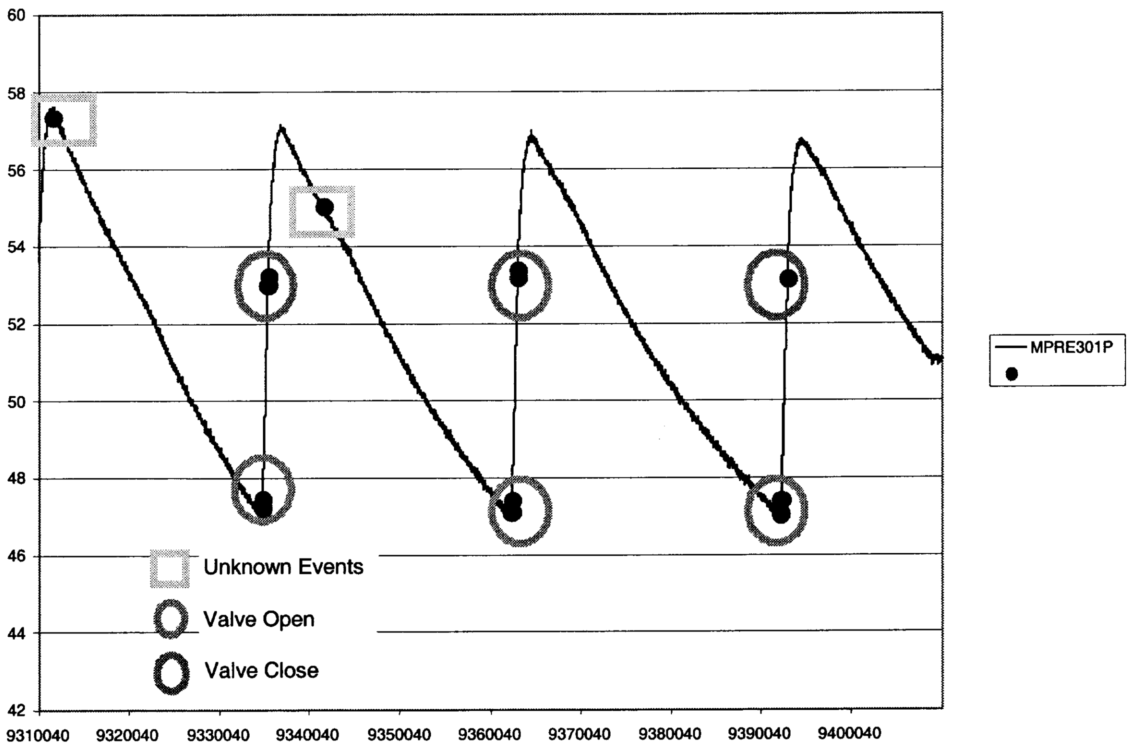


Figure 4.9 Graphical display of event detection result for real signal.

As shown above, most of the events can be successfully detected except for the last one in which only event of "*SV02 Open Command Discontinued*" was detected while event "MMSW102X loses SV02 Open Indication" went unnoticed. The reason can be due to the fact that the two events occur within 10 msecs of each other. After careful inspection of Table 4.12, it can be seen that clusters follow similar patterns for each alternate event: "open" and "close". For example, for "open" events such as #1, #3, #5 (they all contain the same sequence of sub-events of *SV02 Commanded Open* and *MMSW102X indicates SV02 Open* as shown in Table 4.9), the absolute data values are very close to each other (values among Cluster 8, 6 and 7 and values among Cluster 13, 10 and 11). At the same time, for "close" event, the sequence of clusters show the same patterns for "close" event, e.g., Event #2 and #4 both have Cluster 2 and Cluster 3. In addition, two more additional events not on the event list were detected which was shown immediately under the Table 4.12. The adaptive learning property of this system can also be demonstrated as follows. Suppose that after the verification with the domain expert that one of the two additional detected events is true indication of new event, say, at the time of 9311640, a update of the new event list is performed. With this new knowledge in place, the whole processing of time series signal was redone to reflect such event signature database update. The following three more events have been identified and they occurred at the time of 9339320ms, 9366620ms and 9396480ms respectively. A plot of detected events was again shown in Figure 4.10.

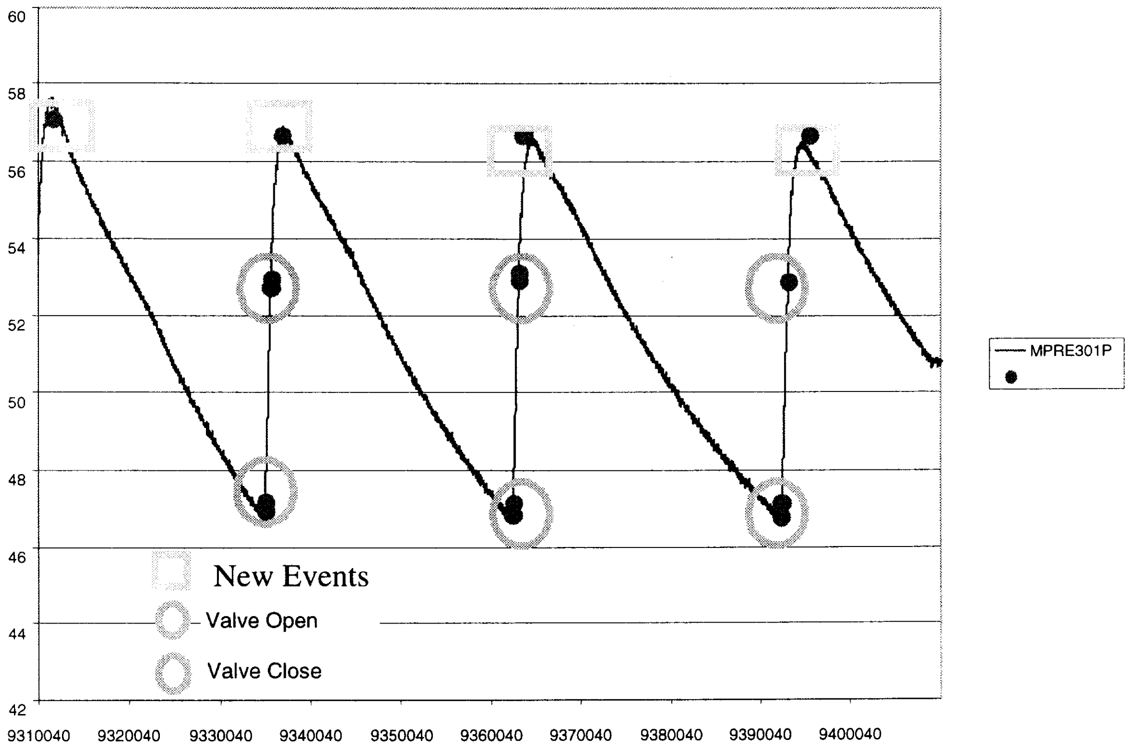


Figure 4.10 Adaptive learning of new events.

4.2.3 GUI-based System Integration

It is the ultimate goal of the ALEC system to let the user interact with it so that both user and system simultaneously learn about new events and their characteristics. Figure 3.1 shows that this system is an integration of several advanced technologies ranging from wavelet transform, hierarchical clustering to fuzzy clustering and database system. In order to reduce the user's learning curve and present an integrated environment, a Graphical User Interface (GUI) has been designed and prototyped. The GUI has been developed with the following in mind [57-58]:

(1) Present commands, options, or data to the user on the appropriate application display.

(2) Display information appropriate to completing a task on the screen, so the user can be selective in attending to information relevant to his or her needs.

(3) Organize information in a meaningful way to help the user focus on essential task information. This makes the decision-making process easier as well as reducing the potential for errors.

As introduced in Chapter 3, the ALEC system is an adaptive system which consists of two phases: training phase and learning phase. The training phase has been described in detail in Chapter 3. After learning the events and storing the cluster statistics-based model signatures, the system performs classification analysis using a nearest neighorhood method to detect events. Then the user are presented with the events for evaluation. In some cases, either inconsistencies occur or new prospective events have been identified, user is then provided with the time-history based information of the past and current occurrences of the prospective events. From the feedback of the user, the sensitivity thresholds of the clusters can be modified accordingly and the event model signature database will then be updated for further analysis. Figure 4.11 illustrates the concept of learning phase with the user interaction.

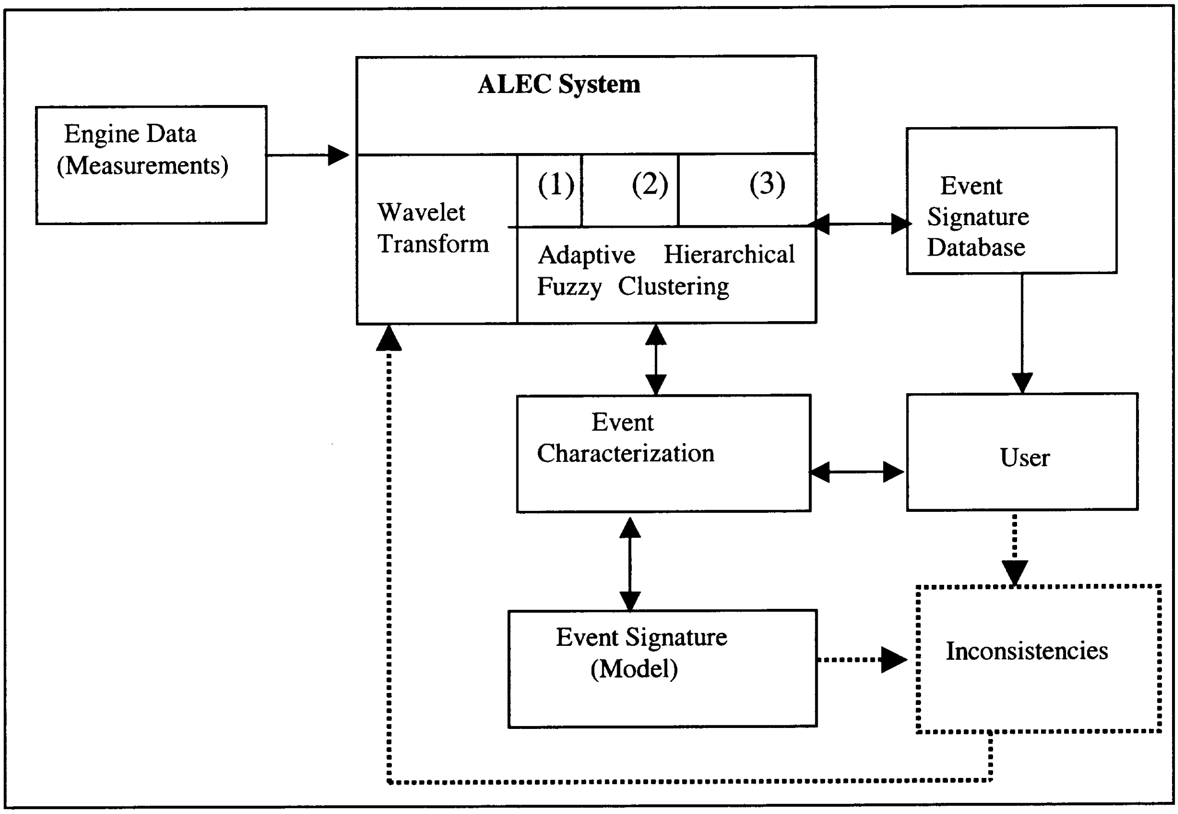


Figure 4.11 System diagram during the data analysis phase of ALEC. (1) Absolute measurement (2) Wavelet decomposition features (3) Statistical features

engines is transformed into the wavelet domain and the specified features are computed. Using the event signature model database, the features from incoming signals are compared with the model signatures of categorized events. In case of a match, the event is classified and reported to the user. In case there is no match to the event, the features are matched with the clustered data in the flagged event database. If it is matched to any cluster, the entire information with past history is reported to the user for additional comments and categorization. If a new categorization is obtained, the data is classified accordingly and the database are updated.

There are a lot of GUI development software under Windows, and a few under Unix and Linux, but sometimes people are asked to develop a GUI tool that can run both under window system and Unix or Linux based system, it might be a painful job. Matlab is a choice to address this problem. In this dissertation, GUIDE, the Matlab Graphical User Interface Development Environment [59], is used because it provides a set of tools for creating GUIs. GUIDE stores GUIs in two files, which are generated the first time you save or run the GUI: (1) FIG-file: a file with extension .fig that contains a complete description of the GUI figure layout and the components of the GUI: push buttons, menus, axes, and so on. When you make changes to the GUI layout in the Layout Editor, your changes are saved in the FIG-file. (2) M-file : a file with extension .m that contains the code that controls the GUI, including the callbacks for its components. This file is referred to as the *GUI M-file*. When you first run or save a GUI from the Layout Editor, GUIDE generates the GUI M-file with blank stubs for each of the callbacks. You can than program the callbacks using the M-file editor. A user manual for the ALEC system is also provided to facilitate the learning the individual module of the system. The following figure is the screen shot of the main menu of the ALEC system.

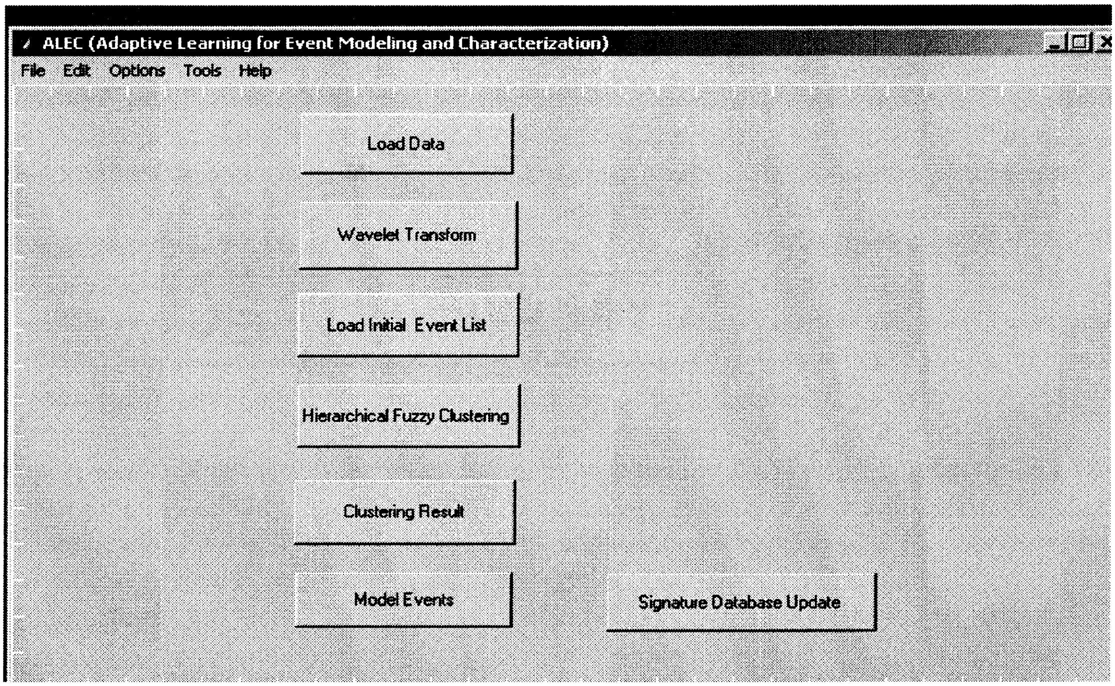


Figure 4.12 The main menu of graphical user interface of ALEC.

Figure 4.12 shows that the ALEC system consists of the following major modules integrated together to provide an automated analysis of raw data and result display.

(1) Load data: enable the user to load the raw data.

(2) Wavelet transform: analyze the data or signal, the default is db2 wavelet.

(3) Load initial event list: get the predefined event list to be used in the guidance of algorithm tuning.

(4) Hierarchical fuzzy clustering/clustering result: the core of the system to produce the clustering results and display to user for evaluation.

(5) Model events: let the user view the statistics of the cluster.

(6) Signature database update: user can update the model database to reflect the current state of the system.

4.2.4 Discussion

For both simulated and real signals, it can be observed that the wavelet-based hierarchical fuzzy clustering method can detect more events than traditional hierarchical clustering method. The reason for this improvement is that wavelet decomposition is performed first on the raw signal which can localize the characteristics of the signal, thus permitting a more detailed description at abrupt changes. Furthermore, with the wavelet-based hierarchical fuzzy clustering method, the processing time is greatly reduced because the new incoming signal only affect the most similar cluster as described in the leader-follower algorithm presented in Chapter 3.

The ability of adaptive learning of ALEC system has been demonstrated by updating the new event model signature database. The new signature would provide new statistics about the detected event, thus providing a guideline for detecting similar events.

It should be pointed out that wavelet-based approach can be easily extended to higher dimension to include additional features. In this experiment, after three-level decomposition is performed, a vector is formed for each signal(or measurement), including absolute measurement, A3, D3, D2, D1 after they have been normalized. Thus the dimension is 5. However, a tradeoff must be found between processing time and produced results. Our experiments have shown that higher dimension can produce more clusters but the processing time also increases greatly. In the analysis of real signal, the D1 level can be further decomposed and included in the feature vector. After the dimension reaches certain number (7 in this case), the clustering result almost remain the same even more dimension can be added. Put in another way, in the experiment, dimension of 7 and dimension of 9 produce the same results. In summary, for the practical use of the wavelet-based approach, a tradeoff should be found between the accuracy and response time, especially for adaptive real-time system.

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

5.1 Conclusions

Results of fuzzy classification of blade engine fatigue mode using generalized multi-dimensional Gaussian membership fuction are effective and simple for classification of patterns with overlapping structures in which some patterns fall into more than one classes. This method is more robust than the hierarchical clustering in which each pattern can only fall into one class. In addition, this approach is also successfully applied to classify skin lesion images into melanoma and dysplastic nevus. Furthermore, the experimental results also show the important role of Principle Component Analysis plays during the preprocessing stages of a classification system with excessive dimensionality feature space which is 28 in our project. PCA achieves its optimal representation of the original patterns using only a subset of principle component in the sense of minimum mean-square error.

The framework of ALEC system was proposed and demonstrated to be feasible in implementation. The wavelet-based adpative hierarchical fuzzy clustering approach uniquely and successfully integrates several advanced technologies to effectively detect and characterize the event of interest through the interaction with the user. By using the adaptive approach, the algorithm can be computationally efficient to satisfy the on-line requirement. Results of the integrated approach have shown that it can detect more events by 50% and 85% than traditional hierarchical clustering while the computation time is reduced between 20% and 25%.

The proposed approach can be extended to applications in image processing, control system and data mining where on-line new data is continuouly being generated and collected.

5.2 Future Work

The following tasks need to be investigated and completed before the final ALEC system is accomplished and put into practical use.

- Design some customized wavelet to extract useful features for better event characterization and detection.
- Complete system evaluation involving domain experts about the accuracy of flagging events.

REFERENCES

[1] Y. Yang, T. Pierce, and J. Carbonell, "A study on retrospective and on-line event detection," In *Proceedings of the 21th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR98)*, pp. 28-36, Melbourne, Australia, August, 1998.

[2] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*, Prentice Hall, Englewood Cliffs, NJ, 1998.

[3] L. A. Zadeh, "Fuzzy sets as a basis for a theory of possibility," *Fuzzy Sets and Systems*, vol. 1, pp. 3-28, 1978.

[4] D. Fisher, "Iterative optimization and simplification of hierarchical clustering," *Journal of Artificial Intelligence Research*, vol. 4, pp. 147-179, 1996.

[5] G. Karypis and E.-H. Han, "Multilevel refinement for hierarchical clustering," *Technical Report #99-020*, 1999.

[6] P. Bradley and U. Fayyad, "Refining initial points for k-means clustering," In *Proceedings of the 15th ICML*, pp. 91-99, Madison, WI.

[7] B. Zhang, "Generalized k-harmonic means – dynamic weighting of data in unsupervised learning," In Proceedings of the 1st SIAMICDM, Chicago, IL, 2001.

[8] J. G. Campbell, C. Frakey, F. Murtagh, and A. E. Raftery, "Linear flaw detection in woven textiles using model-based clustering," *Pattern Recognition Letters*, vol.18, pp.1539-1548, 1997.

[9] G. Celeux and G. Govaert, "Gaussian parsimonious clustering models," *Pattern Recognition*, vol. 28, pp. 781-793, 1995.

[10] S. Mukerjee, E. D. Feigelson, G. J. Babu, F. Murtagh, C. Fraley, and A. E. Raftery, "Three types of gramma ray bursts," *The Astrophysical Journal*, vol. 508, pp. 314-327, 1998.

[11] A. P. Dempster. N. M. Laird and D. B. Rubin, "Maximum likelihood for incomplete data via the EM algorithm,", *Journal of the Royal Statistical Society*, ., vol. 39, pp. 1-38, 1977.

[12] A. Dasgupta and A. E. Raftery, "Detecting features in spatial point processes with clutter via model-based clustering," *Journal of the American Statistical Association*, vol. 93, pp. 294-302, 1998.

[13] G. Doddington, Presentation Slides at The DARPA Broadcast News Workshop, VA, 1999.

[14] L.B. Lombard, *Events: A metephysical study*, Routledge & Kegan Paul, Boston, MA, 1986.

[15] R. Papka, "On-line New Event Detection, Clustering, and Tracking", Ph.D. dissertation, University of Massachusetts, Amherst, MA, 1999.

[16] J. Mao and A. K. Jain, "A self-organizing network for hyperellipsoidal clustering (HEC)," *IEEE Trans. Neural Network* vol. 7, pp. 16-29, 1996.

[18] F. Murtagh, "A survey of recent advances in hierarchical clustering algorithms," *Computer Journal*, 26, 4, 354-359, 1983.

[19] C. Olson, "Parallel algorithms for hierarchical clustering," *Parallel Computing*, 21, pp.1313-1325, 1995.

[20] R. Sibson, "SLINK: An optimally efficient algorithm for the single link cluster method," *Computer Journal*, 16, 30-34, 1973.

[21] E. M. Voorhees, "Implementing agglomerative hierarchical clustering algorithms for use in document retrieval," *Information Processing and Management*, 22, 6, pp. 465-476, 1986.

[22] D. Defays, "An efficient algorithm for a complete link method," *The Computer Journal*, 20, 364-366, 1988.

[23] J. H. Ward, "Hierarchical grouping to optimize an objective function," *Journal Amer. Stat. Assoc.*, 58, 301, 235-224, 1963.

[24] T. Chiu, D. Fang, J. Chen and Y. Wang, "A Robust and scalable clustering algorithm for mixed type attributes in large database environments," In *Proceedings of the 7th ACM SIGKDD*, 263-268, San Francisco, CA, 2001.

[25] J. McQueen, "Some methods for classification and analysis of multivariate observations," In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, 281-297, 1967.

[26] E. Diday, " The dynamic cluster method in non- hierarchical clustering,", *J. Comput. Inf. Sci.*, 2, 61-88, 1973.

[27] M. J. Symon, "Clustering criterion and multivariate normal mixture," *Biometrics* vol. 77, pp. 35-43, 1977.

[28] E. H. Ruspini, "A new approach to clustering," *Inf. Control* 15, 22-32, 1995.

[29] J. C. Bezdek, *Pattern Recognition With Fuzzy Objective Function Algorithms*, Plenum Press, New York, NY, 1981.

[30] R. N. Dave, "Generalized fuzzy C-shells clustering and detection of circular and elliptic boundaries," *Pattern Recogn.* 25, 713-722, 1992.

[31] D. L. Pham, J. L. Prince, "Adaptive fuzzy segmentation of magnetic resonance images," *IEEE Trans on Med. Imaging*, 18, 9, 737-752, 1999.

[32] I. Daubechies, "The wavelet transform, time-frequency localization and signal analysis," *IEEE Trans. Information Theory*, vol. 36, pp. 961-1005, Sept. 1990.

[33] I. Daubechies, "Orthonormal bases of compactly supported wavelets," *Communications on Pure and Applied Mathematics*, vol. 41, pp. 909-996, Nov. 1988.

[34] C. E. Heil and D. F. Walnut, "Continuous and discrete wavelet transforms," *SIAM Review*, vol. 31, pp. 628-666, Dec. 1989.

[35] S. G. Mallat, "Multiresolution approximation and wavelet orthonormal bases of L2," *Trans. Amer. Math. Soc.*, June 1989.

[36] B. Vidakovic, *Statistical Modeling by Wavelets*, John Wiley & Sons, New York, 1999.

[37] P. A. Morettin, "From fourier to wavelet analysis of time series," In *Proceedings in Computational Statistics*, pp. 111-122, 1996

[38] L. Prasad, S.S. Iyengar, and S. S. Ayengar, *Wavelet Analysis with Applications to Image Processing*, CRC Press, 1997.

[39] R. R. Coifman and M. V. Wickerhauser, "Entropy-based algorithms for best basis selection," *IEEE Trans. Information Theory*, vol. 38, pp. 713-718, Mar. 1992.

[40] T. Li, Q. Li, S. Zhu and M.Qgihara, "A survey on wavelet application in data mining," *SIGKDD Explorations*, 4, 2, 49-68, 2001.

[41] G. Strang, "Wavelets and dilation equations: a brief introduction," *SIAM Rev.*, vol. 31, pp. 614-627, Dec. 1989.

[42] S. G. Mallat, A theory for multiresolution signal decomposition: the wavelet represenatation," *IEEE Trans. Pattern Anal. & Mach. Intell.*, 11, 674-693, 1989.

[43] T. Mannila, H. Smyth and D. J. Hand, *Principles of Data Mining*, MIT Press, 2001.

[44] C. M. Bishop, *Neural Networks for Pattern Recognition*, Oxford Press, 1994.

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

[46] R. O. Duda, P. E. Hart and D. G. Stork, *Pattern Classification (second edition)*, John Wiley & Sons, 2001.

[47] I. T. Jolliffe, *Principle Component Analysis*, Springer-Verlag, New York, 1986.

[48] D. L. Donoto and I. M. Johnstone, "Ideal spatial adaptation by wavelet shrinkage," *Biometrika*, vol. 81, no. 3, pp. 425-455, September, 1994.

[49] D. L. Donoto and I. M. Johnstone, "Adapting to unknown smoothness via wavelet shrinkage," Technical Report, Dept. of Statistics, Stanford University, July 20, 1994.

[50] D. L. Donoto and I. M. Johnstone, G. Kerkyacharian, and P. Dominique, "Wavelet shrinkage: asymptopia," *Journal of the Royal Statistical Society Series B*, vol. 57, no. 2, pp. 301-369, 1995.

[51] D. L. Donoto, "De-noising by soft-thresholding," IEEE Transactions on Information Theory, vol. 41, no. 3, pp. 613-627, May, 1995.

[52] I. Daubechies, *Ten Lectures on Wavelets*, SIAM, Philadelphia, PA, 1992.

[53] G. Deslauriers, and S. Dubuc, "Symmetric iterative interpolation processes," Constructive Approximation, vol. 5, no. 1, pp. 49-68, 1989.

[54] C. Herley, "Exact interpolation and iterative subdivision schemes," IEEE Transactions on Signal Processing, vol. 43, no. 6, pp. 1348-1359, June, 1995.

[55] W. Sweldens, "The lifting scheme: a custom-design construction of biorthogonal wavelets," Technical Report, no. IMI 1994:7, Dept. of Mathematics, University of South Carolina, 1995.

[56] W. Sweldens, "The lifting scheme: a construction of second generation wavelets," Technical Report, no. IMI 1995:6, Dept. of Mathematics, University of South Carolina, 1995.

[57] J. T. Hackos and J. C. Redish, *User and Task Analysis for Interface Design*, John Wiley & Sons, New York, 1998.

[58] A. Marcus, N. Smilonich and L. Thompson, *The Cross-GUI Handbook for Multiplatform User Interface Design*, Addison-Wesley Publishing Company, New York, 1995.

[59] *Creating Graphical User Interfaces*, The Mathworks, Natick, MA, 2002.