

## ABSTRACT

Title of Document: DEVELOPMENT OF DIAGNOSTIC AND PROGNOSTIC METHODOLOGIES FOR ELECTRONIC SYSTEMS BASED ON MAHALANOBIS DISTANCE

Sachin Kumar, Doctor of Philosophy (Ph.D.), 2009

Directed By: Chair Professor, Michael Pecht,  
Department of Mechanical Engineering

Diagnostic and prognostic capabilities are one aspect of the many interrelated and complementary functions in the field of Prognostic and Health Management (PHM). These capabilities are sought after by industries in order to provide maximum operational availability of their products, maximum usage life, minimum periodic maintenance inspections, lower inventory cost, accurate tracking of part life, and no false alarms. Several challenges associated with the development and implementation of these capabilities are the consideration of a system's dynamic behavior under various operating environments; complex system architecture where the components that form the overall system have complex interactions with each other with feed-forward and feedback loops of instructions; the unavailability of failure precursors; unseen events; and the absence of unique mathematical techniques that can address fault and failure events in various multivariate systems.

The Mahalanobis distance methodology distinguishes multivariable data groups in a multivariate system by a univariate distance measure calculated from the

normalized value of performance parameters and their correlation coefficients. The Mahalanobis distance measure does not suffer from the scaling effect—a situation where the variability of one parameter masks the variability of another parameter, which happens when the measurement ranges or scales of two parameters are different.

A literature review showed that the Mahalanobis distance has been used for classification purposes. In this thesis, the Mahalanobis distance measure is utilized for fault detection, fault isolation, degradation identification, and prognostics.

For fault detection, a probabilistic approach is developed to establish threshold Mahalanobis distance, such that presence of a fault in a product can be identified and the product can be classified as healthy or unhealthy. A technique is presented to construct a control chart for Mahalanobis distance for detecting trends and biasness in system health or performance. An error function is defined to establish fault-specific threshold Mahalanobis distance.

A fault isolation approach is developed to isolate faults by identifying parameters that are associated with that fault. This approach utilizes the design-of-experiment concept for calculating residual Mahalanobis distance for each parameter (i.e., the contribution of each parameter to a system's health determination). An expected contribution range for each parameter estimated from the distribution of residual Mahalanobis distance is used to isolate the parameters that are responsible for a system's anomalous behavior.

A methodology to detect degradation in a system's health using a health indicator is developed. The health indicator is defined as the weighted sum of a

histogram bin's fractional contribution. The histogram's optimal bin width is determined from the number of data points in a moving window. This moving window approach is utilized for progressive estimation of the health indicator over time. The health indicator is compared with a threshold value defined from the system's healthy data to indicate the system's health or performance degradation.

A symbolic time series-based health assessment approach is developed. Prognostic measures are defined for detecting anomalies in a product and predicting a product's time and probability of approaching a faulty condition. These measures are computed from a hidden Markov model developed from the symbolic representation of product dynamics. The symbolic representation of a product's dynamics is obtained by representing a Mahalanobis distance time series in symbolic form.

Case studies were performed to demonstrate the capability of the proposed methodology for real time health monitoring. Notebook computers were exposed to a set of environmental conditions representative of the extremes of their life cycle profiles. The performance parameters were monitored in situ during the experiments, and the resulting data were used as a training dataset. The dataset was also used to identify specific parameter behavior, estimate correlation among parameters, and extract features for defining a healthy baseline. Field-returned computer data and data corresponding to artificially injected faults in computers were used as test data.

DEVELOPMENT OF DIAGNOSTIC AND PROGNOSTIC METHODOLOGIES  
FOR ELECTRONIC SYSTEMS BASED ON MAHALANOBIS DISTANCE

By

Sachin Kumar

Dissertation submitted to the Faculty of the Graduate School of the  
University of Maryland, College Park, in partial fulfillment  
of the requirements for the degree of  
Doctor of Philosophy  
2009

Advisory Committee:

Professor Michael Pecht, Chair

Professor Christopher Davis

Professor Donald Barker

Professor Peter Sandborn

Assistant Professor Byeng Dong Youn

## Dedication

This thesis is dedicated to my family and friends, who taught me to never give up.

## **Acknowledgements**

With a deep sense of gratitude, I wish to express my sincere thanks to my advisor, Prof. Michael Pecht, for allowing me to work on this interesting topic and continuously challenging me to set my sights higher. Thanks to Professors Davis, Barker, Sandborn, and Youn for serving on my dissertation committee and for providing helpful comments on the dissertation.

I thank Eli Dolev for the technical discussions, encouragement, and for reviewing my work. I am grateful to The DEI Group, which provides reliability engineering based asset management solutions, for inspiring me to work on prognostics and health management.

I wish to express my sincere gratitude to the CALCE research faculty for providing feedback during morning meetings, notably Dr. Azarian, Dr. Das, and Sony Mathew.

I am thankful to Mark Zimmerman for his help in improving my writing skills and for reviewing my technical papers. I gratefully acknowledge the financial support of the CALCE PHM Consortium for my research.

I wish to express very special thanks to all of my friends at CALCE who have been so helpful to me during my graduate experience. I am grateful to my morning coffee mates Nishad Patil, Aftab Alam, Rubyca Jaai, Hyunseok Oh, Vikram Srinivas, Elviz George, Vasilis Sotiris, and Rishi Raj. I am also thankful for my friends with whom I have shared apartments over the years and never felt away from my home country. Finally, I would like to thank my family members, especially my mother, for her constant support of my education.

# Table of Contents

Acknowledgements.....	iii
Table of Contents.....	iv
List of Tables .....	vi
List of Figures.....	vii
Chapter 1: Introduction.....	1
1.1 Background and Motivation .....	1
1.2 Research Scope and Objectives .....	6
1.3 Dissertation Overview .....	8
Chapter 2: Literature Review.....	10
2.1 Prognostics and Health Management.....	10
2.1.1 The Statistical Reliability-based Approach.....	11
2.1.2 The Life Cycle Load-based Approach .....	12
2.1.3 The State Estimation-based Approach .....	13
2.1.4 The Feature Extraction-based Approach.....	14
2.2 Mahalanobis Distance.....	15
Chapter 3: Experimental Details and Characterization of Performance Parameters ..	25
3.1 Data Collection Software.....	27
3.2 Experimental Setup.....	30
3.3 Data Collection and Analysis Procedure .....	34
3.4 Baseline of Performance Parameters .....	34
3.4.1 CPU Temperature .....	37
3.4.2 Motherboard Temperature .....	39
3.4.3 Videocard Temperature .....	40
3.4.4 Fan Speed.....	41
3.4.5 C2 State.....	42
3.4.6 C3 State.....	43
3.4.7 CPU Usage.....	44
3.4.8 Usage Level .....	45
3.5 Influence of Environmental, Usage and Power Setting on % CPU Usage..	47
3.6 Summary and Conclusions .....	51
Chapter 4: Fault Identification Approach .....	54
4.1 Diagnostic Approach .....	59
4.2 Healthy Baseline using Mahalanobis distance and Empirical model .....	60
4.3 Threshold Determination .....	62
4.3.1 Generic Threshold Determination.....	63
4.3.2 Fault-Specific Threshold Determination.....	64
4.4 Case Study .....	66
4.5 Summary and Conclusions .....	76
Chapter 5: Fault Isolation Approach.....	79
5.1 Faulty Parameter Isolation Methodology.....	83
5.2 Case Study .....	88
5.3 Summary and Conclusions .....	97
Chapter 6: Health Degradation Identification Approach .....	99
6.1 Degradation Identification Methodology.....	101

6.1.1	Bin-width Estimation Technique .....	102
6.1.2	Health Degradation Detection Procedure .....	103
6.2	Case Study .....	109
6.3	Fault and Degradation Detection .....	113
6.4	Results and Discussion .....	116
6.5	Summary and Conclusions .....	119
Chapter 7:	Prognostic Measures using Symbolic Time Series Analysis .....	122
7.1	Symbolization of Time Series.....	128
7.2	System Modeling .....	131
7.3	Health Monitoring Methodology .....	133
7.4	Case Study .....	139
7.5	Summary and Conclusions .....	148
Chapter 8:	Contributions and Future Work .....	151
Bibliography	.....	154

## List of Tables

Table 1: Parameters Monitored in Notebook Computers .....	28
Table 2: Environmental Conditions .....	31
Table 3: Experiments Performed .....	31
Table 4: Correlation Coefficients for Battery Performance Parameters .....	35
Table 5: Correlation Coefficients for Notebook Performance Parameters .....	36
Table 6: Abbreviations Used for Performance Parameters .....	36
Table 7: Statistics for CPU Temperature .....	38
Table 8: Statistics for Motherboard Temperature .....	39
Table 9: Statistics for Video Card Temperature .....	40
Table 10: Fan Speed Characterization .....	42
Table 11: Statistics for CPU State 2 (C2) .....	42
Table 12: Statistics for CPU State 3 (C3) .....	44
Table 13: Statistics for Percentage CPU Usage .....	45
Table 14: Percentage of Alarms Raised by Different Rules .....	72
Table 15: Estimated Values of the Parameters .....	76
Table 16: Orthogonal Design of Experiment .....	91
Table 17: Percentage of Observations Indicating Faulty Parameters .....	96
Table 18: Health Indicators for System .....	109
Table 19: Changes in Average Coefficient .....	143
Table 20: Relative Change in Shannon Entropy for Different Partitions .....	144
Table 21: Relative Changes in Shannon Entropy for Different Word (Symbol Sequence) Length .....	144
Table 22: Prognostics Measures Computed for Field-Returned Computer .....	146

## List of Figures

Figure 1: Baseline construction flow chart.....	27
Figure 2: Frequency chart for CPU temperature.....	37
Figure 3: Probability density of residual CPU temperature for healthy product.....	38
Figure 4: Frequency chart for motherboard temperature.....	39
Figure 5: Frequency chart for videocard temperature.....	40
Figure 6: Frequency chart for fan speed.....	41
Figure 7: Frequency chart for C2 state.....	43
Figure 8: Frequency chart for C3 state.....	44
Figure 9: Frequency chart for CPU usage.....	45
Figure 10: Residual plot of usage level.....	46
Figure 11: Variability of %CPU usage with different power source.....	48
Figure 12: Variability in %CPU usage with different use conditions at room temperature.....	48
Figure 13: Variability of %CPU usage with different usage levels.....	49
Figure 14: Variability of %CPU usage with different environmental conditions.....	49
Figure 15: Percentage CPU usage in different ranges.....	50
Figure 16: Fault detection approach.....	60
Figure 17: Mahalanobis distance calculation using test data.....	60
Figure 18: Baseline establishment methodology.....	61
Figure 19: An approach for defining threshold MD value.....	64
Figure 20: Illustration of threshold value calculation.....	66
Figure 21: Histogram of MD values for healthy population.....	69
Figure 22: Plot of $\lambda$ and likelihood function $f(x, \lambda)$ .....	70
Figure 23: Normal probability plot of transformed variable $x$ .....	71
Figure 24: Control limits for fault identification.....	73
Figure 25: MD value for a baseline and a test system.....	73
Figure 26: Optimal threshold evaluation.....	75
Figure 27: Robustness evaluation of threshold value.....	75
Figure 28: Parameter isolation: threshold determination for parameters ( $\Delta MD_{it}$ ).....	85
Figure 29: Threshold determination.....	86
Figure 30: Parameter isolation (test data evaluation).....	87
Figure 31: Isolating faulty parameter in an observation based on $\Delta MD$ .....	87
Figure 32: $\Delta MD$ values for one observation of training data.....	90
Figure 33: $\Delta MD$ distribution of four different parameters.....	91
Figure 34: MD values corresponding to a fault injection in the videocard experiment.....	93
Figure 35: Comparison of MDs of field-returned computer with baseline.....	95
Figure 36: Degradation methodology using non-parametric method.....	105
Figure 37: Shift in fractional contribution of each bin with time.....	107
Figure 38: Trend of bin's fractional contribution over time.....	108
Figure 39: %CPU usage in different environmental conditions.....	111

Figure 40: %CPU usage in different usage conditions .....	111
Figure 41: MD values for baseline.....	112
Figure 42: Health indicator for baseline .....	113
Figure 43: Sample MD values for test system (healthy).....	114
Figure 44: Sample health indicator for healthy test computer .....	114
Figure 45: Sample MD value for test system (unhealthy) .....	115
Figure 46: Health indicator for unhealthy test computer .....	116
Figure 47: Statistical representation using box plot of MDs .....	117
Figure 48: Mahalanobis distance calculation for training and test data.....	128
Figure 49: Data series conversion into a symbol series.....	130
Figure 50: Schematic of diagnostic and prognostic approaches.....	134
Figure 51: Symbolic time series generation.....	135
Figure 52: Markov state determination.....	137
Figure 53: A representative Markov state model.....	138
Figure 54: MD values of healthy system .....	141
Figure 55: Histogram of MD values for healthy population.....	142

## Chapter 1: Introduction

The desire for diagnostic and prognostic capabilities has been around for as long as humans have operated complex, expensive, and safety-critical equipment. Diagnostic and prognostic capabilities are two of the main constituents of Prognostic and Health Management (PHM) systems. PHM systems have been created to monitor system health, provide early detection of faults, identify failure modes, point out failure precursors, detect degradation, determine remaining useful life, and recommend maintenance/logistic responses [1].

Any industry would like to have the maximum operational availability of their products and systems, minimum periodic inspections, a low number of spares, maximum usage life, accurate part-life tracking, and no false alarms. PHM can make such things possible. PHM is an approach that enables real-time health assessment of a system in its actual application conditions by sensing, recording, and interpreting environmental, operational, and performance-related parameters that are indicative of a system's health [2].

### 1.1 Background and Motivation

The real-time health assessment of electronics has great importance due to its wide range of applications, from a battery safety circuit to system-of-systems readiness. Electronics, as components, subsystems, or products, are an integral part of many systems. Electronics provide functionality and performance through mechanical and electrical controls. Electronic systems with a long life-cycle ensure customer satisfaction and low liability at the manufacturer's end. Increased warranties and the

severe liability of system failures make it desirable to analyze a system's performance in the field and determine its operational availability, which is a new concept that is being incorporated into contracts in the heavy vehicle industry and the airline industry. In 2005, electronics prognostics was identified by Logistic management institute (LMI) as one of the most needed maintenance-related features, and a similar view was observed in the avionics industry [3][4].

The Department of Defense (DoD) outlined the importance of PHM implementation in its DoD 5000.2 policy document for defense acquisition. It states that "program managers shall optimize operational readiness through affordable, integrated, embedded diagnostics and prognostics, embedded training and testing, serialized item management, automatic identification technology, and iterative technology refreshment" [5].

Often, quantification of degradation and fault progression in an electronic system is difficult since not all faults necessarily lead to system failure or functionality loss [1] [2]. In addition, there is a significant lack of knowledge about failure precursors in electronics [6]. With limited failure precursors and complex architecture, it is generally hard to implement a health monitoring system that can directly monitor all the conditions in which fault incubation occurs. The built-in test (BIT) and self-test abilities in a system were early attempts at providing diagnostic capabilities that were incorporated into a system's structure [7]. But the applicability of these capabilities was limited to the failure definition embedded at the system's manufacturing stage, whereas with recent developments in sensor and data analysis capabilities, the implementation of data-driven diagnostic systems that can adapt to

new failure definitions is now possible. The literature review in Chapter 2 shows various PHM approaches based on the data type and system information.

A system's health assessment is made by observing its multiple performance parameters. Dataset of very high dimensions presents an analytical challenge, since all non-trivial data mining and indexing algorithms degrade exponentially with dimensionality [8]. A high-dimensional dataset contains a lot of valuable information, while a lower dimensional measure is easier to comprehend and can be computed quickly. Consideration of correlations among performance parameters is advantageous as an electronic product experiences diverse environmental and use conditions. For example, the capacitance and insulation resistance of a capacitor vary with changes in ambient temperature. The effectiveness of a diagnostic or prognostic procedure increases by incorporating the change in relationship among performance parameters. Each performance parameter changes at a different rate due to changes in ambient conditions.

A multivariate Mahalanobis distance (MD) [9], a unified parameter representative of system health, is used to capture the non-linear dynamics of an electronic system. Motivation of chose the Mahalanobis distance for this work came from its certain attributes and advantages over other available approaches. These attributes and advantages of Mahalanobis distance are as follows:

- It reduces a multivariate system to a univariate system (simplifies system monitoring).
- It is sensitive to inter-variable changes in a multivariate system, because it takes correlations between parameters into account.

- It does not suffer from a scaling effect and uses correlation among parameters in contrast to other distances such as the Manhattan distance, the Euclidean distance, and the Hamming distance.
- It uses correlation matrix for fault diagnosis whereas the Hotelling T-square and the square prediction error indices use covariance. The covariance measure is not scaled on the same level and varies with change in measurement unit.
- It reduces the analytical burden, because MD provides a number after combining information on all performance parameters, whereas other multivariate approach such as multivariate state estimation technique (MSET) provides an estimate for each parameter and needs analytical assessment of each parameter for a system's health determination.
- It provides higher dimensionality reduction compared to the principle component analysis (PCA), which also explains variance-covariance with fewer linear combinations of original parameters. But to calculate the principle components, all original variables are needed.

The MD approach reduces the analytical burden because information on all the performance parameters is combined into a number (i.e., MD), which is utilized for system health assessment. The MD does not suffer from the scaling effect because it uses normalized data. The scaling effect describes a situation where the variability of one parameter masks the variability of another parameter; this happens when the measurement ranges or scales of two parameters are different [10]. The use of correlation among parameters for MD calculation also makes it sensitive to small change in performance parameters.

Betta [11] presented requirements for system monitoring, including establishment of a suitable threshold for fault diagnosis, to perform the continuous comparison of the system under analysis. Liu et al. [12] discussed the need to localize the component or subsystem that is the source of the anomalous behavior.

In electronics, degradation occurs due to several mechanisms, including electromagnetic disturbance [13], electro-migration [14], and corrosion [15]. Yang et al. [16] discussed how a light emitting diode degrades with the increase in duty cycles under the typical dynamic working conditions pulse input.

In the MD-based diagnostic approach, the traditional method of defining a threshold MD value is either based on personal judgment, a trade-off for lowering the economic consequences of misclassifications, or an MD value that corresponds to a known abnormal condition is chosen [17]-[20]. These traditional methods do not provide a generic framework to define a threshold MD value for fault identification. For the health degradation identification, application of MD has been limited [21][22]. The parameters used in these studies were either monotonously increasing or decreasing. Up until now, MD has not been utilized for fault isolation and prognostic purposes.

The Mahalanobis distance has been used for various purposes in many different disciplines. It has been used for: 1) real-time process control, because MD combines all performance parameters into a number, which eliminates the need for analyzing each individual parameter; 2) health monitoring of a complex system whose system dynamics change rapidly, because MD can be computed for multivariate observations in quick succession; 3) qualifying a product for a particular

fault, because a unique threshold for each fault can be established; 4) clustering (unsupervised pattern recognition) where similar objects (measured samples) are grouped together; 5) discriminating a product from established/confirmed good product; 6) locating faults in a system; and 7) estimating probability of impending faults.

Although Mahalanobis distance has been used for various applications, it has some limitations. These limitations are as follows: 1) the MD approach suffers from the masking effect if the training data contains a significant amount of outliers, because MD uses a sample mean and a correlation matrix, which can be influenced by a cluster of outliers; 2) the outliers can shift the sample mean and inflate the correlation matrix in a covariate direction. This is especially true if the  $n/p$  ratio is small, where  $n$  is the number of observations and  $p$  is the number of features; 3) the computation time increases in order of  $O(p^2)$  for  $p$ -dimensionality of feature vectors; and 4) there are no guidelines for treating noise factors in the MD method.

## 1.2 Research Scope and Objectives

The use of a threshold MD that is either based on personal judgment or on an economic trade-off presents a challenge for the applicability of the MD-based diagnostic approach for a new system that has limited failure evidence. A probabilistic approach to define a threshold MD value when system faults are either unknown or known does not exist in the literature.

A diagnostic approach would have wider applicability if it were capable of performing fault isolation in addition to fault detection. The fault isolation capability

enables identification of parameters that may have contributed to a fault. Until now, MD has not been used for fault isolation purposes.

MD has been used to identify health degradation of a few systems. But the performance parameters of these systems were either monotonously increasing or decreasing. An MD-based approach for degradation identification of a system whose performance parameters do not exhibit a monotonous trend is not available.

Until now, MD has not been utilized for any prognostic purposes. Enabling the use of MD to estimate failure probability and time to failure can increase the applicability of the MD approach to new application areas.

The research goals are summarized in the following objectives:

1. Develop a probabilistic approach to define a threshold MD
  - a. to detect unknown faults, and
  - b. to detect known faults.
2. Develop an approach using MD to identify faulty parameters in order to perform fault isolation.
3. Develop an approach using MD to detect system's health degradation.
4. Develop an approach using MD to estimate failure probability and time to failure.

Objective 1 focuses on establishing a threshold MD value in order to classify a system as being healthy or unhealthy. A fault-specific threshold MD value is determined by minimizing an error function such that a product can be qualified against a specific known fault. A control chart for MD values is also constructed to detect trends and biasness in system health.

Objective 2 focuses on developing a method that enables fault isolation by identifying parameters that are associated with a fault. The distribution of residual MD values for each parameter is obtained, and a probabilistic range of each parameter's contribution toward a healthy system's MDs is estimated. These probabilistic ranges are used to identify parameters that are responsible for the anomalous behavior of a system.

Objective 3 focuses on developing a methodology to detect a system's health degradation. A moving window approach is utilized. In each assessment window a histogram of MD values is created to summarize the system's performance. A health indicator, which is a weighted sum of a bin's fractional contribution to the histogram, is calculated. As time progresses, new estimates of the health indicator are obtained and are compared with the degradation threshold value in order to detect degradation.

Objective 4 focuses on defining and estimating prognostic measures for detecting anomalies in a system and predicting if and when the system will become faulty. The MDs are utilized as a time series signal and transformed into symbolic form so that a system's behavior under different environmental and operational conditions can be represented using a hidden Markov model. This Markov model is then used to compute prognostic measures.

### 1.3 Dissertation Overview

The work is organized as follows. Chapter 2, which focuses on background literature, discusses different PHM approaches, previous studies based on Mahalanobis distance, and the mathematics involved in MD calculation. Chapter 3 presents experimental details, data collection, and performance parameter

characterization. The collected data has been used as training data for case studies. Chapter 4 presents an approach for establishing a probabilistic threshold MD value in order to classify a system as being healthy or unhealthy. It also presents the creation of a control chart and an approach to establishing a fault specific threshold MD value.

Chapter 5 presents an approach for isolating a fault by identifying parameters that are associated with that fault. Chapter 6 presents an approach for detecting degradation in a system's health by means of a health indicator estimated by using MD values. Chapter 7 presents an approach for defining and estimating prognostic measures for detecting anomalies in a product and predicting if and when the product will become faulty. Finally, Chapter 8 lists the contributions of this thesis and possible future work.

## Chapter 2: Literature Review

### 2.1 Prognostics and Health Management

Prognostics and health management is a combination of three concepts: enhanced diagnostics, prognostics, and health management. While a system performs its intended functions, enhanced diagnostics estimates the system's health condition and provides a high degree of fault detection and fault isolation capability with a low false alarm rate. Prognostics involves the assessment of a system's actual health condition followed by modeling fault progression, health degradation, performance prediction, and remaining useful life determination. Health management provides the capability to make intelligent, informed, and appropriate decisions about logistic actions based on diagnostics and prognostics information, available resources, and operational demand.

Diagnostic techniques for a system are based on observational data taken from the system's performance and its environment, while prognostic techniques are based on historical data, system knowledge, future usage, and future environmental conditions. Although the goals of diagnostics and prognostics are somewhat different, studying them separately is not practical. This is because prognostic methods are often built on the results of diagnostic methods. The following subsection provides a literature review of work related to electronic prognostics and Mahalanobis distance.

The various models and algorithms for PHM are studied and can be grouped into four different categories based on available data type and system information: (1) statistical reliability-based approaches, (2) life cycle load-based approaches,

(3) state estimation–based approaches, and (4) feature extraction–based approaches. The following section presents the models and algorithms being used in PHM

### 2.1.1 The Statistical Reliability–based Approach

A statistical reliability–based PHM approach is appropriate for systems that have a sensor network that insufficiently monitors health conditions; that have a short life cycle with a low fault rate; are non-critical; and involve low risk. This approach assumes usage and environmental conditions have no effect and that knowledge of failure mechanisms is not required. This approach needs a system’s historical failure data and can be used for legacy systems, since failure and/or inspection data for legacy systems are often available in abundance to be used as input for statistical reliability models. However, for new products accelerated testing is required to obtain failure times. Accelerated testing may cause new or different failure modes to evolve under accelerated conditions.

The Weibull distribution is the most appropriate statistical distribution for analyzing life data [23][24]. The lognormal distribution has also been used in many applications to analyze life data [24][25].

Gebraeel [26] developed a degradation-modeling framework that combined reliability and degradation characteristics of a component’s population with real-time sensory information acquired through condition monitoring. A methodology was provided to compute and update residual life distributions of partially degraded components provided that the degradation model is in exponential form.

### 2.1.2 The Life Cycle Load-based Approach

The life cycle environment of an electronic product consists of the assembly, storage, handling, and use of the product (application and operational loads), including the severity and duration of these conditions [27]. Various life cycle loads are due to environmental conditions such as temperature, humidity, pressure, vibration or shock, chemical environments, radiation, contaminants, and loads due to operating conditions such as current, voltage, and power. These loads may affect the reliability of the product either individually or in combinations with each other.

Mathew et al. [28][29], presented remaining-life assessment of circuit cards inside the space shuttle solid rocket booster (SRB) based on vibration time history from the prelaunch stage to splashdown in conjunction with damage models. Vichare et al. [30][31], performed in-situ health monitoring of notebook computers and estimated the distributions of the load parameters, which were used to estimate damage accumulation and make a remaining life prediction. Tuchband et al. [32] utilized information of the life cycle loads of line replaceable units (LRUs) to estimate the readiness of LRUs to complete a mission. Zhang et al. [33] presented an enhanced prognostic model to predict the remaining useful life of electronic assemblies. The model utilizes environmental loads and in-situ performance measurements in conjunction with two baseline prediction algorithms: life consumption monitoring (LCM) and uncertainty adjusted prognostics (UAP).

In the life cycle load-based approach, damage accumulation models for specific systems and components are formulated considering the usage profile (e.g., fatigue cycle as a function of operating conditions). Damage is often assumed to

accumulate at the same rate for a given stress level irrespective of the past, although experimental results have shown that damage can accumulate in a nonlinear manner [34]. As a result, many nonlinear damage theories have been proposed to account for the nonlinearity in damage accumulation. In general, Miner's rule is recommended for its simplicity, versatility, and reasonable accuracy.

### 2.1.3 The State Estimation–based Approach

State estimation takes all the information collected by sensors and uses it to determine the underlying behavior of a system at any point in time. State estimation–based techniques that can track the gradual degradation of systems can assist in providing intelligent control, detecting faults, and in predicting future faults. There are two parts to the state estimation techniques: modeling and training. The overall approach is to create a model and train the model with the data made available by the health monitoring of a system. Controls provided by an electronic system require reliable real-time estimation of its present state.

Chinnam et al. [35] presented an approach based on the Hidden Markov Model (HMM) for autonomous diagnostics as well as prognostics. Camci et al. [36] used HMM for health-state forecasting.

Pattern recognition algorithms, such as the multivariate state estimation technique (MSET) and sequential probability ratio test (SPRT), are used to identify signal degradation and provide a preliminary indicator of failure in servers. Lopez [37] used electronic prognostics consisting of a continuous system telemetry harness with SPRT and MSET algorithms for electronics prognostics. Urmanov et al. [38]

developed a failure precursor approach for early fault detection and fault prediction in computing servers.

In the state estimation–based approach, by minimizing error between an estimate obtained from a model and measurement, future states can be predicted. The state estimation–based approach has been used successfully as a product maintenance strategy, but it has not been widely used for electronics prognostics.

#### 2.1.4 The Feature Extraction–based Approach

Feature extraction–based PHM approaches derive features directly from routinely monitored systems’ operational data (e.g., calibration, power, vibration and acoustic signal, temperature, current, and voltage). These approaches assume that the data features are relatively constant unless a malfunctioning event occurs in the system. These approaches are based on the theory of pattern recognition and can be implemented at the system level or at the subsystem level. Generally, these techniques work for assessing system-level degradation, since a performance loss typically results from the improper functioning of multiple components and their interactions. These approaches require the availability of sensor information to assess the current health condition of a system or sub-system.

Vichare et al. [31] monitored a time-load signal and processed it to extract the cyclic range ( $\Delta s$ ), cyclic mean load ( $S_{\text{mean}}$ ), rate of change of load ( $ds/dt$ ), and dwell time ( $tD$ ). These outputs are used in fatigue damage accumulation models. Vichare et al. [39] suggested embedding the data reduction and load parameter extraction algorithms into a sensor module to reduce on-board storage space, lower power consumption, and provide uninterrupted data collection over longer durations.

Swanson [40] used data features to track system degradation and provide advance warning. Wu et al. [41] proposed an autoregressive integrated moving average (ARIMA) modeling and forecasting approach based on the Box-Jenkins model to predict the future health status of a machine. Urmanov [42] used an empirical model for remaining useful life prediction. Brown et al. [43][44][45] used a principle feature of a device to define a healthy profile under temperature cycling testing, and later used that profile for remaining useful life prediction. Sotiris et al. [46] used support vector machines and support vector regression to detect the health of multivariate systems based on training data representative of healthy operating conditions.

Certain distance measures are used to classify a system into different groups. Some of the distance measures that have been used quite often are the Euclidean distance [47], the Mahalanobis distance, and the Bayesian distance [48]. Nearest neighbor algorithms are used to combine two closest groups in a new group and are based on distance measures. The Mahalanobis distance method and statistical pattern recognition have been applied in several areas as has been discussed earlier.

## 2.2 Mahalanobis Distance

Mahalanobis distance (MD) calculation was first developed to calculate distances between two groups in multivariate statistics (1936) [49]. In 2000 [9] and 2001 [50], the Mahalanobis-Taguchi system was proposed, in which Taguchi's design of experiment concept, orthogonal array, and signal-to-noise (S/N) ratio were used with the MD method to reduce the number of parameters to be used for MD computation. Thereafter, it has been used in various fields including medicine,

engineering, and economics. Some of the research works involving MD in the field of prognostics and health management are as follows.

Nie et al. [51] found that the MD method was able to detect failures of the multilayer ceramic capacitors (MLCCs) in temperature-humidity-bias (THB) conditions.

Srinivasan et al. [52] used MD to detect network intrusion and reduced the number of false alarms. The result was an improvement over conventional anomaly-based intrusion detection systems. The approach was also capable in early detection of novel attacks.

Musthpa et al. [53] used MD for damage detection in components of avionics. They successfully detected damage 1 mm in size on these components unambiguously.

Sohn et al. [54] developed an MD-based monitoring system that integrates hardware and software components to diagnose welded connections in a steel moment-resisting frame structure. This structure was susceptible to cracking during seismic loading and involved high cost for visual inspection of these joints. The MD of the potential outlier was checked against a threshold value, and the status of the structure was determined based on this comparison. They used a Monte Carlo method to determine the threshold value.

Li et al. [55] proposed MD based hybrid contextual fire detection algorithm using airborne and satellite thermal images. The algorithm essentially treats fire pixels as anomalies in images. It utilizes the local background around a potential fire pixel in order to isolate the fire pixel. This approach improved accuracy up to 28% on

average for some of the images. In addition, they used MD to create fire probability images that were useful for fire propagation modeling.

Aman et al. [56] proposed an MD-based model for detecting cost-prone classes in software development and maintenance. The model helped in improving the testing of object-oriented software programs that include a lot of sub-routines, components, reduced work force requirements, and correspond to other realistic restrictions.

Sung et al. [21] used MD to consolidate multi-dimensional measured values such as discharge current, voltage, and luminance of a plasma display panel and converted them into a standard scale that was used to analyze measured data. It was found that the MD value was useful in evaluating the degree of degradation of the plasma display panel discharge cell.

Chinnam et al. [22] studied the gradual degradation of a drill-bit during the drilling process. Two degradation signals, thrust force and torque, were used to analyze degradation. Ten features (five features per degradation signal) were used to obtain the Mahalanobis distance. These data features were obtained from the holes with “normal” degradation levels. Data from the last hole drilled prior to the tool breakage, representing “abnormal” degradation level, were used for the validation of the measurement scale. Subsequently, the useful features out of the ten under study were identified using orthogonal arrays and signal-to-noise ratios. A threshold value representing the 99th percentile of the MDs from the “normal group” showed significantly superior performance compared to any of the individual features studied.

MD has been widely used for classification purposes, such as Chinese characters [57], handwritten signatures [58], surface roughness [59], object trajectories [60], patterns in cluttered imagery [61], images [62], human emotions [63], and antenna radiation patterns [64].

MD is also used in the manufacturing sector for manufacturing control systems [65], manufacturing process improvement [66], and quality control [50]. MD has also found applicability in service sectors where it is used for consumer vehicle ratings [67], for selectively displaying scenes from a sumo fight on a mobile phone [68], and for building benchmark tools to assess lean manufacturing processes [69].

Research conducted in the last couple of years are more focused on detection and classification problems. In a few cases MD has been used for system degradation identification when the objectives are targeted to a specific problem. The threshold value defined in every case is based on an expert's judgment. The MD has not been explored for prognostic purposes.

### **Mahalanobis Distance Calculation**

The Mahalanobis distance methodology distinguishes multivariable data groups by a univariate distance measure, which is calculated from the measurements of multiple parameters [9][70]. The MD value is calculated using the normalized value of performance parameters and their correlation coefficients, which is the reason for MD's sensitivity.

A dataset formed by measuring the performance parameters of a healthy product is used as training (or baseline) data in MD calculation. The collection of MD values for a healthy system is known as the Mahalanobis space. The performance

parameters collected from a product are denoted as  $X_i$ , where  $i = 1, 2, \dots, p$ . Here,  $p$  is the total number of performance parameters. The observation of the  $i$ th parameter, on the  $j$ th instance, is denoted by  $X_{ij}$ , where  $i = 1, 2, \dots, p$ , and  $j = 1, 2, \dots, m$ ;  $m$  is the total number of times an observation is made for all parameters. Thus, the  $(p \times 1)$  data vector for the normal group is denoted by  $X_j$ , where  $j = 1, 2, \dots, m$ . Each individual parameter in the data vector is normalized using the mean and the standard deviation of that parameter calculated from the baseline data. Thus, a parameter's normalized values are:

$$Z_{ij} = \frac{(X_{ij} - \bar{X}_i)}{S_i}, \quad i = 1, 2, \dots, p, j = 1, 2, \dots, m, \quad (1)$$

$$\text{where } \bar{X}_i = \frac{1}{m} \sum_{j=1}^m X_{ij} \text{ and } S_i = \sqrt{\frac{\sum_{j=1}^m (X_{ij} - \bar{X}_i)^2}{(m-1)}} \quad (2)$$

Next, the values of the MDs are calculated for a healthy product:

$$MD_j = \frac{1}{p} z_j^T C^{-1} z_j \quad (3)$$

where  $z_j^T = [z_{1j}, z_{2j}, \dots, z_{pj}]$  is a vector comprising  $z_{ij}$ ;  $z_j$  is the transpose of  $z_j^T$ ; and  $C$  is the correlation matrix calculated as:

$$C = \frac{1}{(m-1)} \sum_{j=1}^m z_j z_j^T \quad (4)$$

### **Parameter Selection for Mahalanobis Distance Calculation**

Mahalanobis distance calculation uses the normalized value of parameters and the correlation between parameters. The normalization and the correlation coefficient

computation involve parameters' mean ( $\mu$ ) and standard deviation ( $\sigma$ ), which are measures of the variability or dispersion of a data set. Therefore, one should evaluate the mean and standard deviations and the correlation of parameters before considering a parameter for MD calculation.

A system's performance parameters can be qualitative and quantitative. Qualitative data is not expressed in terms of numbers, but rather by means of a natural language description, and this type of data is also referred to as categorical data. Taguchi and Jugulum have presented an approach for using qualitative data in MD [50]. Quantitative data is expressed in terms of numbers. These parameters can have measurements in different scales and units. Normalization of parameters removes these scaling and unit effects from the data. One should be careful during the computation of the parameter's mean and standard deviations and correlation coefficients between parameters, because these values can corrupt the MD value.

If a parameter is non-varying (i.e.,  $\sigma = 0$ ) then a parameter's normalized value cannot be computed. At the same time, correlation between non-varying parameters and other parameters cannot be estimated. These two situations are undesirable for MD calculation, so only parameters that are varying (i.e.,  $\sigma \neq 0$ ) should be considered. Also, a parameter that has very small variability (i.e.,  $\sigma < \#$ ) should not be considered, as this variability may be due to measurement errors. A parameter that is non-varying provides no information about systems dynamic behavior and such a parameter could be used for other purposes such as to determine whether a system is operational or not.

A parameter's measurement can be a default value or a representation of the

system's performance. Therefore, when analyzing a parameter measurement one should investigate for the parameter's default value, if any. If a parameter's default measurement is combined with a system's performance measurement, the resulting parameter's mean and standard deviations, and correlation coefficient could be misleading and may not reflect the true nature of system dynamics. If the difference between a default value and the smallest value of performance measurement is significantly higher than the difference between two consecutive values of performance measurements, the data set (i.e., population) can be grouped into two: one with a default value, where the number of parameters considered for MD calculation would be one less than total number of parameters considered otherwise, and other with all parameters, where number of parameters considered for MD calculation would be equal to the total number of parameters. A flow chart is presented (Figure 1) to explain the parameter selection procedure for MD calculation. The need to consider parameter analysis before MD calculation is illustrated by a few simulated case studies.

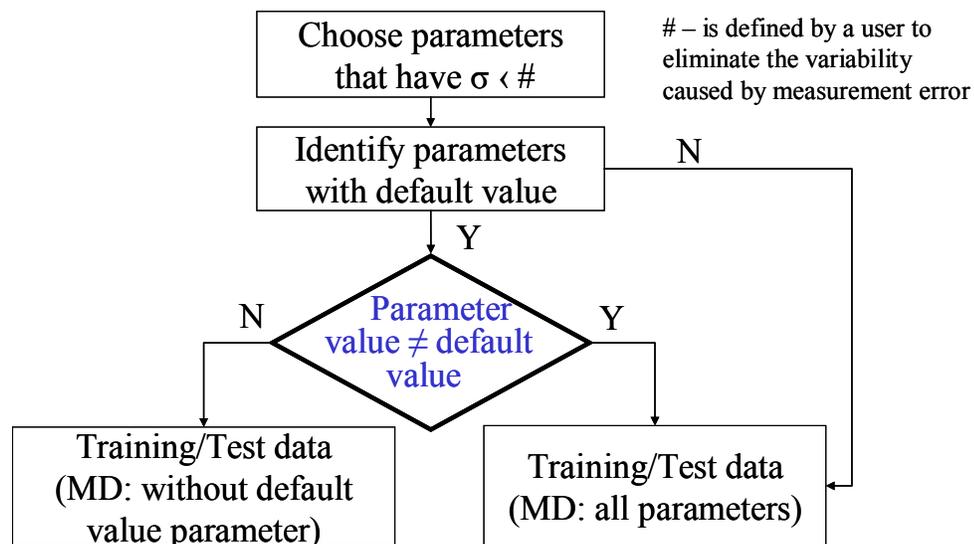
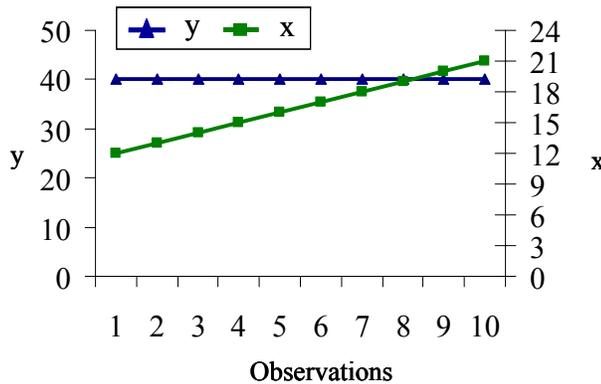


Figure 1: Selection of parameters (training/test data) for MD

**Case I: A parameter is constant (i.e.,  $\sigma = 0$ )**

Two parameters, x and y, are chosen, where y has a constant value and x varies linearly. Standard deviation for y is 0 and correlation coefficient between x and y cannot be computed. Therefore, Mahalanobis distance cannot be computed using parameter x and y since these parameters do not have correlation.

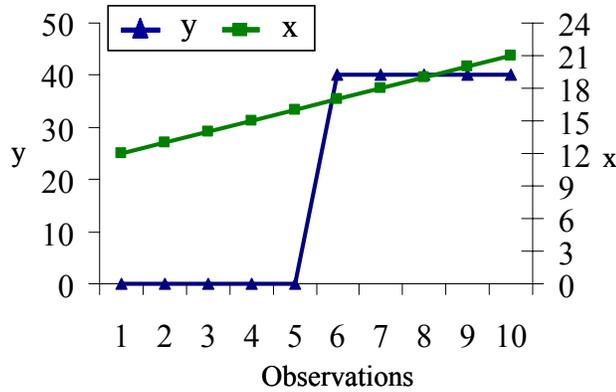


	Mean	SD	Corr.
y	40	0	-
x	17	3	

Here y is equal to 40 and x varies from 12 to 21. Parameters x and y have no correlation.

**Case II: A parameter has one-step increment**

Two parameters x and y is chosen, where y has a step increment at 5<sup>th</sup> observation from 0 to 40 and x varies linearly. The parameter y can be treated as categorical parameter with level 1 (y = 0) and 2 (y = 40) and MD can be computed, or y can be treated as continuous parameter, or y can be considered as parameter that has a default value (=0). MD values do not vary by considering y parameter as categorical or continuous parameter because correlation coefficient remains same and parameters are normalized. But, parameter y has a default value and two data group is formed then MD can not be computed for either group. This highlights the need of parameter analysis before considering them for MD calculation.



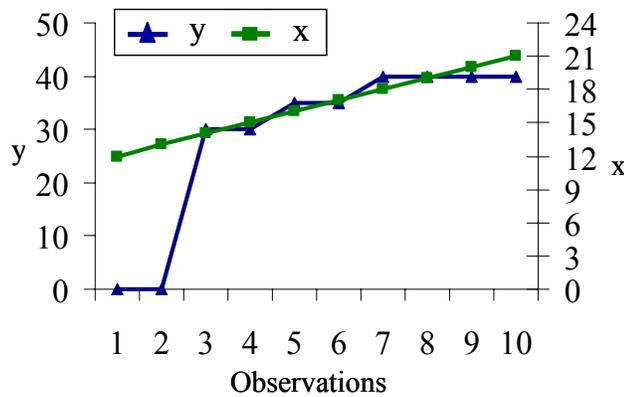
Here y varies from 0 to 40 in one step and x from 12 to 21. MD values are as follows considering all 10 observations for the training.

Observation	1	2	3	4	5	6	7	8	9	10
MD (x, y: 1, 2)	1.4	0.7	0.5	0.7	1.4	1.4	0.7	0.5	0.7	1.4
MD (x, y: 0, 40)	1.4	0.7	0.5	0.7	1.4	1.4	0.7	0.5	0.7	1.4

### Case III: A parameter has more than one step increment

Two parameters x and y are chosen, where y has a larger increment at the 2<sup>nd</sup> observation from 0 to 30 followed by smaller increments to 35 and 40, and x varies linearly. The parameter y can be treated as a categorical parameter with several levels one for each increment, or y can be treated as a continuous parameter, or y can be considered as a parameter that has a default value (=0). MD value varies by changing numeric values corresponding to each step. Considering y as a categorical parameter when y has multiple steps is not advisable, because a number of pseudo parameters will increase, which are equal to the number of levels considered for parameters minus 1. Also, consideration of multiple categorical levels reduces the effect of step size and becomes insensitive to values falling between step ranges. But, if an assumption is made that parameter y has a default value (=0), two data groups are formed. When a parameter has multiple steps, considering steps that are closer for training improves the effectiveness of the MD method. This is illustrated by the MD

values for each observation considering two different training data. Although the first two observations are different from the rest of the observations, categorizing them into training reflects these observations are similar to the rest of the population, which is misleading. Considering that other observations are similar identifies these two observations as dissimilar, which is true, as can be observed from the data plot. Therefore, analyzing each parameter is important before considering them for MD calculations.



Here y varies from 0 to 30 in one step and is followed by smaller increments, x from 12 to 21. MD values are calculated considering two training data set.

Training 1: observation 1 to 10

Training 2: observation 3 to 10

Observation	1	2	3	4	5	6	7	8	9	10
MD (Training 1)	1.6	2	1.2	0.4	0.5	0.1	0.4	0.3	0.8	1.7
MD (Training 2)	<b>125.8</b>	<b>140.6</b>	1.1	1.3	0.5	0.1	1.5	0.5	0.6	1.8

## Chapter 3: Experimental Details and Characterization of Performance Parameters

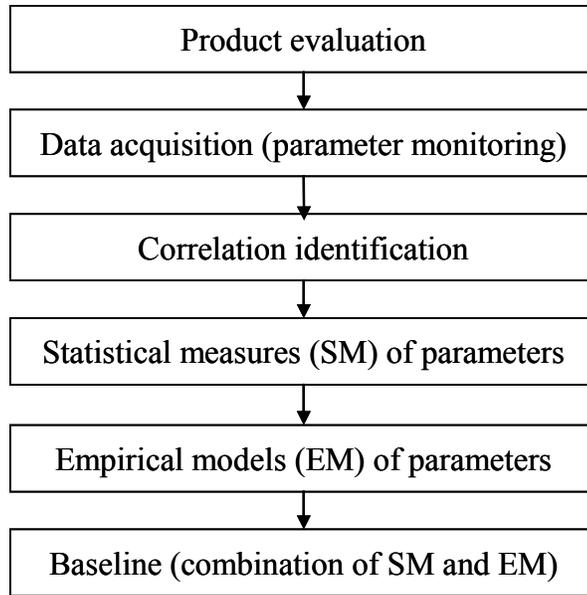
Electronic products can be monitored by assessing their performance indices [30][39]. These indices from a pristine system can define the baseline performance of that system, and the baseline can be used later for identifying degradation or failures. Early detection of a problem based on baseline performance will allow preventative action to be taken in order to avoid problems.

The definition of reliability warrants that a product must perform its intended functionality to be considered reliable under stated conditions. The functionality of a product can be assessed by monitoring its performance parameters. Building knowledge of the performance parameters' variability is essential in order to make informed reliability decisions [71]. To perform prognostics for an electronic product it is necessary to develop an understanding of its performance under various usage and environmental conditions.

In this characterization study, notebook computers, which are complex electronic products, were used to perform experiments under various environmental and usage conditions. Based on the suggested operating conditions of various notebook manufacturers, a range of environmental conditions was chosen for the experiment. Usage conditions were decided based on a report by the U.S. Department of Commerce [72]. A combination of usage conditions was not considered in this work, because there was the assumption that a user does not often use certain types of

applications concurrently. For example, a user would not open Word and Excel files simultaneously. A user generally only accesses one software application at a time.

A flow chart for establishing a baseline is shown in Figure 2. The baseline establishing procedure starts with functional evaluations of a product under consideration. This evaluation process helps to identify the critical components to be monitored, germane performance parameters, and expected outcomes after considering usage conditions, feasibility and limitations, and economic variability. In the data acquisition process a product's performance is monitored using built-in, embedded, or external sensors and stored after performing data cleaning. In this work, performance parameters are monitored and recorded using data collection software. Correlation between parameters is obtained to study the influence of parameters on one another. Such correlation also serves as a pointer for parameter selection during empirical equation development. In the statistical measurement step, a performance parameter's features including mean, standard deviation, distance measures, Eigen values, and other metrics that are extracted to represent a product's health. The variability of each performance parameter is addressed by establishing an empirical relationship between the performance parameters. These equations along with the statistical measures are used to establish the baseline performance of the parameters. The baseline can be used to detect system's deviation from normal operation and for determining prognostic distance.



**Figure 2: Baseline construction flow chart**

### 3.1 Data Collection Software

For this experiment, software was developed to collect real-time performance parameter information from the notebook computers without any user intervention. The software was coded in C++ programming language. The software interacted with the computers' basic input-output system (BIOS) to retrieve performance parameter information. It periodically wrote this information in log files (.txt format). These performance data were grouped into three categories, as shown in Table 1. The software also collected the notebook computers' hardware information, which included each computer's service tag, model number, BIOS version, maximum central processing unit (CPU) speed, video controller, size of the hard drive, hard drive make/model number, and size of the system's random access memory (RAM). The computer's mechanical usage information records how many times a button/key was pressed, an optical disk drive (ODD) was swapped, a battery was inserted and/or

removed, and when docking has occurred. Mechanical usage information was obtained every five minutes.

**Table 1: Parameters Monitored in Notebook Computers**

<b>Parameter</b>	<b>Unit</b>	<b>Frequency</b>
<b>1. Device Information</b>		
Battery's relative state of charge (RSOC)	%	1 min
Battery current	mA	1 min
Battery voltage	mV	1 min
Fan speed	RPM	1 min
LCD brightness	%	1 min
<b>2. Thermal information</b>		
CPU temperature	°C	30 sec
Videocard (i.e., graphic processing unit) temperature	°C	30 sec
Motherboard temperature	°C	30 sec
<b>3. Performance management information</b>		
CPU power state, C1/C2/C3 state	%	5 sec
CPU usage	%	5 sec
CPU throttling	%	5 sec
Memory usage capacity	pages per sec	5 min

A CPU's performance management information includes its power saving states (C1, C2, and C3), CPU usage, and CPU throttling. C1, C2, and C3 are processor power states, commonly known as the power saving states [73]. These states are the processor's sleeping state, where the processor consumes less power and dissipates less heat than in its active state. They represent the percentage time a processor spends in the low-power idle state (i.e., C1, C2, and C3 are a subset of the processor's total idle time). In the C1 power state, the processor is able to maintain the context of the system caches and has its lowest exit latency. The C2 power state has lower power and higher exit latency than the C1 power state. In the C2 power state, the processor is able to maintain the context of the computer's caches. The C3 power state offers improved power savings and higher exit latency over the C1 and C2 states. In the C3 power state, the processor is unable to maintain the coherency of its caches. CPU usage is a measure of how much time the CPU spends on a user's applications and high-level Windows functions, and it is measured in terms of percentage (%CPU). CPU throttling is a feature that adjusts CPU speed in run-time. CPU throttling sets the maximum CPU percentage to be used by any process or service, thereby ensuring that no process can consume all of the CPU's resources at the expense of other users or processes. Often, CPU throttling is performed to accommodate an excessive work request and to manage CPU temperature. The memory usage capacity, measured in pages per second, is the number of requested pages needed to run applications that were not available in the random access memory (RAM) and that had to be read from or written to the hard disk to make room in the RAM for other pages.

## 3.2 Experimental Setup

Experiments were performed on ten identical notebook computers representative of the 2007 state of the art in notebook computer performance and battery life (nearly 3.5 hrs on a single battery). The computers were exposed to a set of environmental and usage conditions representative of the normal life cycle profile and likely extremes. The performance parameters were monitored in situ during the experiment. Operational temperatures for most notebook computers are in the range of 5°C to 45°C; this experiment was conducted in the temperature range of 5°C to 50°C. For the experiment, six different environmental conditions were tested (see Table 2)

The duration for each test was based on the type of power applied. When a computer was powered by an AC adapter (when the battery was fully charged), the test duration was 3.5 hrs. When a computer was powered by an AC adapter (when the battery was fully discharged), the test duration was determined by the time it took for the battery to fully charge. When the battery alone powered the laptop, the test duration was determined by the time it took for the battery to fully discharge. Tests were conducted in a temperature-humidity chamber and in an ambient room environment.

For each temperature/humidity combination, four usage conditions and three power supply conditions were applied. Factorial experiments were designed to study the effect of each factor on the response variable, as well as the effects of interactions between factors. Table 3 lists all 72 experiments. Each computer was powered on for

30 min before starting each experiment, and the computers were kept at room temperature between each test.

**Table 2: Environmental Conditions**

<b>Temperature-Humidity</b>
E1. 5°C with uncontrolled RH
E2. 25°C with 55% RH
E3. 25°C with 93% RH
E4. 50°C with 20% RH
E5. 50°C with 55% RH
E6. 50°C with 93% RH

**Table 3: Experiments Performed**

<b>Power Setting</b>	<b>Usage Level</b>	<b>Environmental Condition</b>
AC adapter (when battery is fully charged)	U1 – U4	E1 – E6
AC adapter (when battery is initially fully discharged)	U1 – U4	E1 – E6
Battery only	U1 – U4	E1 – E6

A set of software for the experiments was installed on the computers, along with Windows XP Professional operating system, Microsoft Office, Front Page,

WinRunner, Spybot, Winamp, Real Player, Visual Studio, Java 5, Minitab, iTunes, Adobe Photoshop, MATLAB, Winzip, and McAfee Antivirus. A script file was written using WinRunner software to simulate user activity. The antivirus application McAfee v8.0 was configured to run on the laptops all of the time.

The same environmental and usage conditions were applied to each of the ten computers to achieve time synchronization between the computers and the software application responses. Each notebook's power mode was always set to ON. The screen saver and hibernation options were disabled to prevent these functions from occurring during the experiment. The wireless capability of the computer was disabled due to the limited wireless connectivity inside the temperature-humidity chambers. Four levels (U1, U2, U3, and U4) of computer usage were chosen:

1. Idle system (U1) - In this category the operating system was loaded, all windows were closed, and user input from the keyboard or mouse and the optical drive was disabled. USB and Firewire peripherals were not attached.
2. Office productivity (U2) - This usage condition was designed to simulate an office environment. The simulator read a Word document and prepared a new Word document. The simulator opened the file explorer and located a file to be opened. It opened a "technology benchmark report" Word document of 88 pages. The simulator read the document, using the arrow keys to page up and page down, and selected a paragraph to copy. The simulator opened a new document from the Word toolbar and pasted the copied section into a new document. The simulator resized both documents to toggle between them. It switched to the original document, read pages, copied additional paragraphs, and pasted them into the new

document. The simulator also typed a new paragraph into the new document. With these activities, the simulator created a five-page document and saved it. Then it saved the file by invoking the “save as file” explorer and providing a file name for the new document. The simulator performed a cleanup by resizing and closing all of the opened documents. It then removed the new files from the desktop and pasted them into another folder. Finally, the simulator closed all of the opened file explorer windows.

3. Media center (U3) – This usage condition was designed to simulate entertainment conditions. The Winamp (v5.24) media player was started from the start menu. The file explorer window was opened in Winamp. MP3 music files were stored on the hard drive and selected to play in Winamp. The music was stopped after 4 min, then the Winamp player window was shut down. The Real media player (v10.5) was started from the start menu. The file explorer window was opened to select video files in Real player. Video files from a DVD were selected by maneuvering through the file explorer window and then played in Real player. Movie screens were resized to full screen. The movie was turned off after 90 min and Real player was closed.
4. Game mode (U4) – In this category, the usage condition was designed to simulate gaming. Quake Arena II was started from the start menu and the single player option was selected to start the game. After an hour of play, the game was stopped and exited.

### 3.3 Data Collection and Analysis Procedure

For each set of test conditions, a time log was maintained. Data were continuously collected in each notebook computer and stored in a separate database. A set of statistical metrics—including the mean, the median, the mode, the standard deviation, the minimum, the maximum, the kurtosis, the skewness, and the 95% confidence interval—were calculated for each parameter for each set of experiments, with their corresponding environmental, usage, and power-setting conditions. Kurtosis and skewness were used to determine the normality of each dataset. The mean values of the performance parameters were used to calculate the Pearson correlation coefficient.

### 3.4 Baseline of Performance Parameters

To create a baseline of system performance, each performance parameter was analyzed. Analysis of the performance parameters revealed that they did not necessarily follow any parametric distribution over the range of experiments. Environmental factors such as temperature, humidity, and applications (software) running on the system have a significant influence on the performance parameters. Therefore, non-parametric methods such as histograms, kernels, orthogonal series estimation, or the nearest neighbor method must be used to estimate the probability density function. Histograms and kernel density were used for this study.

Different power supply sources (battery, AC adapter) had no apparent effect on the performance parameters of the computer, which is discussed later in section 6. To produce histograms for performance parameters, calculate statistics, estimate

correlation coefficients, and derive empirical equations, data was collected from a sample set of 10 computers, which were powered by AC adapters (when the batteries were fully charged) and operated under all four usage conditions (U1 – U4). The correlation coefficient between parameters expresses the strength and direction of a linear relationship between parameters. The correlation coefficients between performance parameters are given in Table 4 and Table 5. In the tables, only significantly correlated (p-value less than 0.05) parameters, and the corresponding correlation coefficients between those parameters are given. Table 4 shows parameters related to the battery performance of the notebook computers, and Table 5 contains parameters related to computer performance.

An empirical equation for each parameter was given in order to calculate the expected values of the parameters. To construct an empirical equation for each performance parameter, the correlated performance parameters, which describe the most variable of the dependent parameters, were considered. Abbreviations used for the different parameters are given in Table 6.

**Table 4: Correlation Coefficients for Battery Performance Parameters**

	Power source	Battery life	RSOC	Current	Voltage	%C2 state
Power source	1	-0.49	-0.75	0.58	-0.27	-
Battery life	-0.49	1	0.51	-0.88	0.63	0.26
RSOC	-0.75	0.51	1	-0.56	0.75	-
Current	0.58	-0.88	-0.56	1	-0.62	-
Voltage	-0.27	0.63	0.75	-0.62	1	-
%C2 state	-	0.26	-	-	-	1

**Table 5: Correlation Coefficients for Notebook Performance Parameters**

	Fan speed	CPU temp	Motherboard temp	Videocard temp	%C2 state	%C3 state	%CPU usage	%CPU throttle
Ambient temp	0.92	0.74	0.96	0.67	0.35	-0.63	0.52	0.43
Ambient humidity	0.25	0.26	0.34	0.23	-	-	-	-
Usage level	-	0.36	-	0.44		-0.62	0.74	-0.48
Fan speed	1	0.78	0.95	0.77	0.48	-0.75	0.61	0.22
CPU temp	0.78	1	0.86	0.98	0.60	-0.81	0.66	-0.22
Motherboard temp	0.95	0.86	1	0.81	0.45	-0.70	0.56	0.23
Videocard temp	0.77	0.98	0.81	1	0.61	-0.85	0.70	-0.33
%C2 state	0.48	0.60	0.45	0.61	1	-0.46	-	-0.30
%C3 state	-0.75	-0.81	-0.70	-0.85	-0.46	1	-0.93	0.20
%CPU usage	0.61	0.66	0.56	0.70	-	-0.93	1	-
%CPU throttle	0.22	-0.22	0.23	-0.33	-0.30	0.20	-	1

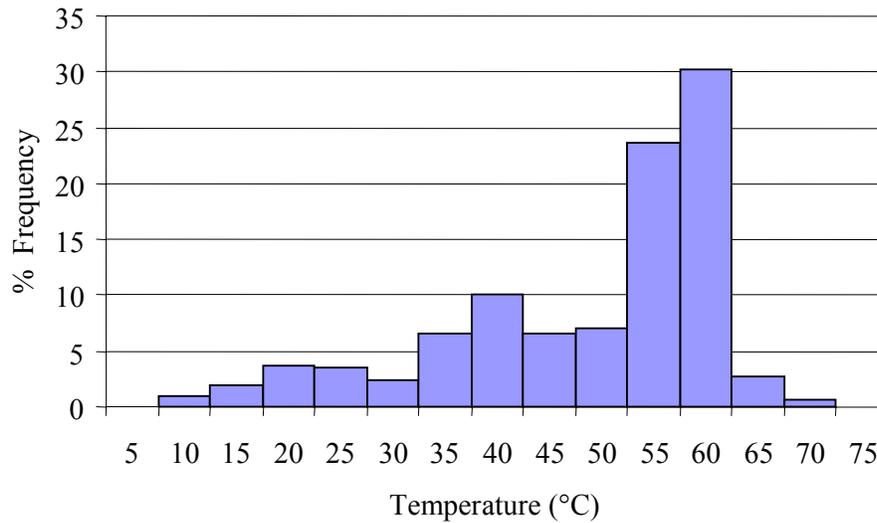
**Table 6: Abbreviations Used for Performance Parameters**

CT = CPU temperature	C3 = % C3 state
MT = Motherboard temperature	C2 = % C2 state
VT = Video card temperature	T = Ambient temperature
FS = Fan speed	H = Ambient humidity
CPU = % CPU usage	CTh = % CPU throttle

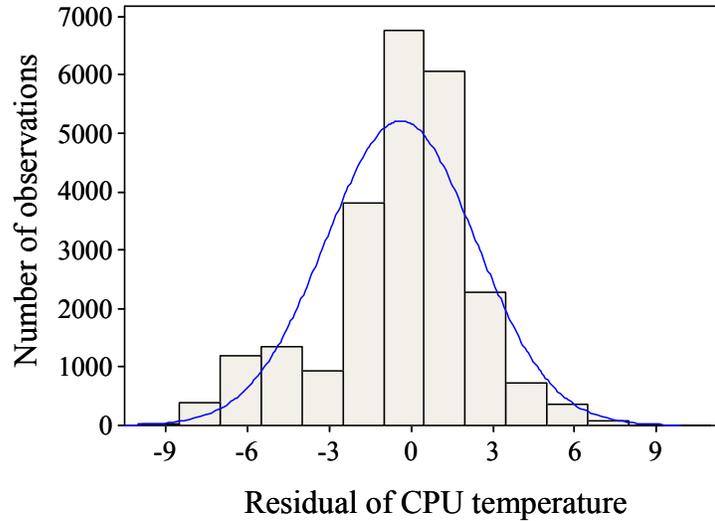
### 3.4.1 CPU Temperature

A histogram for the CPU temperature is presented in Figure 3. Although the computer was ON, the CPU temperature in a few instances was observed to be below room temperature. This was attributed to the ambient temperature. It was verified by the minimum temperature mentioned in the column of the 5°C test condition of Table 7. Means, standard deviations, and the range of CPU temperatures in different ambient temperature conditions are given in Table 7. An empirical equation for the CPU temperature as a function of fan speed, motherboard temperature, and video card temperature is

$$CT = -21.6 - 0.0025*FS + 0.44*MT + 0.87*VT \quad (5)$$



**Figure 3: Frequency chart for CPU temperature**



**Figure 4: Probability density of residual CPU temperature for healthy product**

A histogram of residuals obtained from estimated and observed CPU temperature is presented in Figure 4. Approximately 94% of the variation in CPU temperature is represented by the probability density function of residual data. This suggests that the regression equations can be used for the purpose of comparison, but cannot be relied upon completely. Similar observations were made for other performance parameters as well.

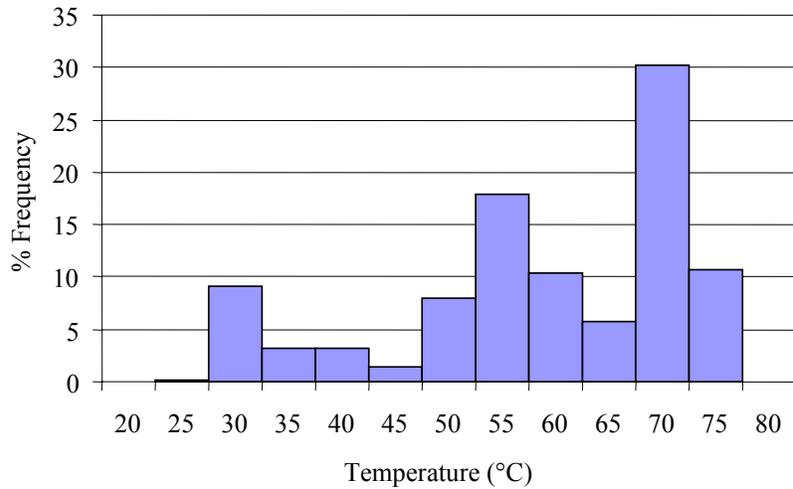
**Table 7: Statistics for CPU Temperature**

	All data points	5°C	25°C	50°C
Mean	46.7	29.6	43.3	54.8
Std Dev	12.7	5.0	4.1	3.5
Minimum	9.0	9.0	22.0	29.0
Maximum	70.0	70.0	70.0	70.0

### 3.4.2 Motherboard Temperature

A histogram for the motherboard temperature is presented in Figure 5. Means, standard deviations, and the range of motherboard temperatures in different ambient temperature conditions are shown in Table 8. An empirical equation for the motherboard temperature as a function of ambient temperature, fan speed, CPU states C2 and C3, and CPU temperature is

$$MT = 9.59 + 0.22 * T + 0.005 * FS + 0.53 * CT - 0.22 * C2 + 0.10 * C3 \quad (6)$$



**Figure 5: Frequency chart for motherboard temperature**

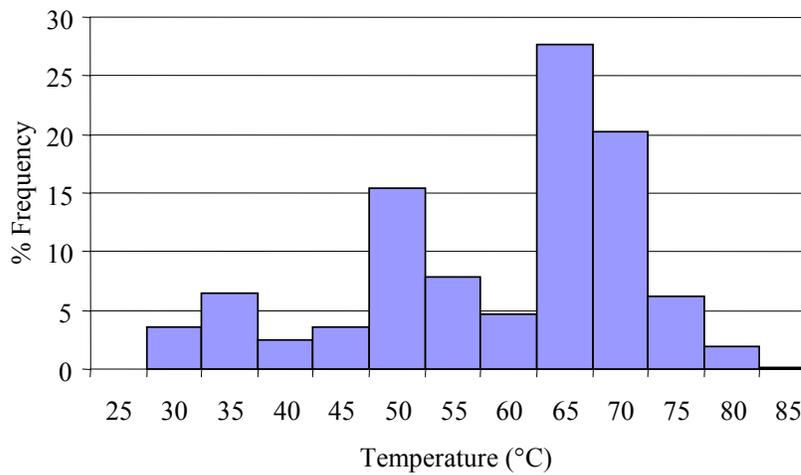
**Table 8: Statistics for Motherboard Temperature**

	All Data Points	5°C	25°C	50°C
Mean	56.8	32.7	53.0	67.4
Std Dev	13.1	5.2	3.2	1.9
Minimum	25.0	25.0	28.0	35.0
Maximum	74.0	52.0	62.0	74.0

### 3.4.3 Videocard Temperature

A histogram for the video card temperature is presented in Figure 6. Means, standard deviations, and the range of videocard temperatures in different ambient temperature conditions are shown in Table 9. An empirical equation for videocard temperature as a function of CPU state C3, CPU temperature, and CPU throttles is

$$VT = 24.6 + 0.81*CT - 0.06*C3 - 0.08*CTh \quad (7)$$



**Figure 6: Frequency chart for videocard temperature**

**Table 9: Statistics for Video Card Temperature**

	All Data Points	5°C	25°C	50°C
Mean	57.0	42.0	53.9	66.1
Std Dev	12.1	19.0	8.8	2.4
Minimum	26.0	26.0	35.0	41.0
Maximum	83.0	76.0	80.0	83.0

### 3.4.4 Fan Speed

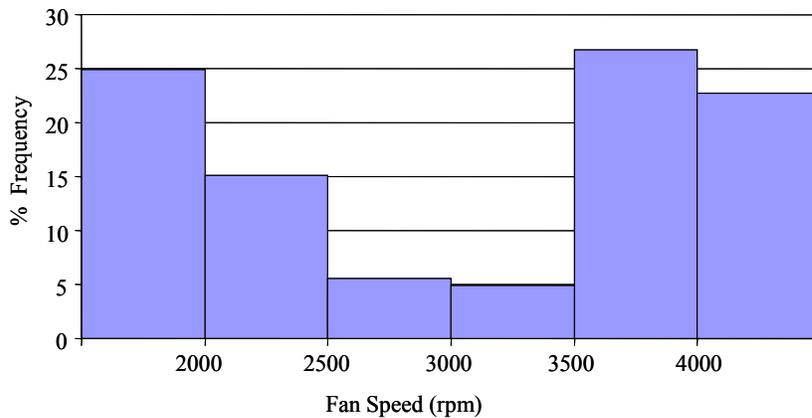
A histogram for fan speed is presented in Figure 7. Fan speed is a parameter that increases and decreases in steps. Fan speed predominantly depends on motherboard temperature but is fine-tuned based on CPU temperature. Fan speed is categorized and shown in Table 10. An empirical equation for fan speed as a function of ambient temperature, CPU temperature, motherboard temperature, percentage CPU usage, CPU state C3, and CPU throttle is

Fan speed

$$= 0 \quad \text{when } T < 25^{\circ}\text{C}$$

$$= 1506 + 26.2 * T - 81.4 * CT + 113 * MT - 10.9 * CPU - 19.5 * C3 - 25.8 * CTh$$

$$\text{when } T \geq 25^{\circ}\text{C} \quad (8)$$



**Figure 7: Frequency chart for fan speed**

Fan speed can be grouped into four categories based on motherboard temperature, and is given in Table 10. In each category, the startup fan speed depends on the CPU temperature.

**Table 10: Fan Speed Characterization**

Category	Motherboard temperature (°C)		Fan speed (RPM)		CPU temperature (°C)		Sub-speed in group
	min	max	min	max	min	max	
1	15	50	0	0	7	64	NA
2	51	55	2422	2561	33	63	12
3	56	58	2859	3463	41	62	12
4	59	72	3903	4031	41	64	3

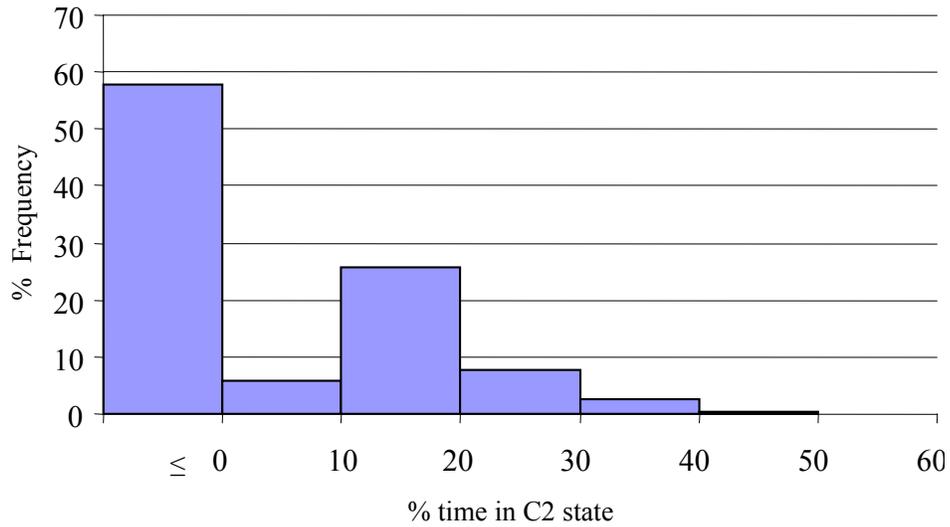
### 3.4.5 C2 State

A histogram for the CPU state (C2) is presented in Figure 8. Means, standard deviations, and the range of the CPU state C2 in different ambient temperature conditions are shown in Table 11. An empirical equation for CPU state C2 as a function of ambient temperature, fan speed, CPU state C3, CPU temperature, motherboard temperature, and videocard temperature is

$$C2 = 0.52*T + 0.01*FS + 0.97*CT - 2.56*MT + 0.72*VT + 0.35*C3 \quad (9)$$

**Table 11: Statistics for CPU State 2 (C2)**

	All Data Points	5°C	25°C	50°C
Mean	7.5	4.6	6.1	9.3
Std Dev	6.1	7.3	5.8	5.8
Minimum	0	0	0	0
Maximum	62	59	62	57

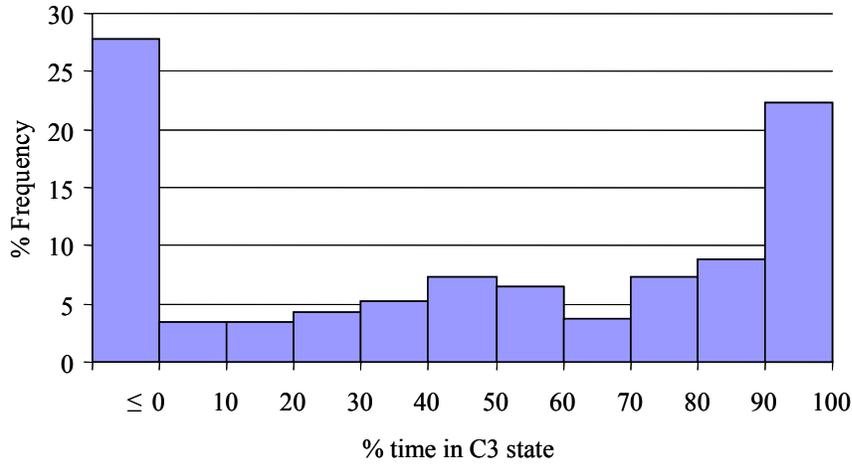


**Figure 8: Frequency chart for C2 state**

### 3.4.6 C3 State

A histogram for the CPU state (C3) is presented in Figure 9. Means, standard deviations, and the range of CPU states, C3, in different ambient temperature conditions are shown in Table 12. An empirical equation for the CPU state C3 as a function of ambient temperature, fan speed, CPU state C2, CPU usage, CPU throttle, motherboard temperature, and videocard temperature is

$$C3 = 109 - 0.007*FS + 1.15*MT - 1.09*VT - 1.34*C2 - 0.83*CPU - 0.34*CTh \quad (10)$$



**Figure 9: Frequency chart for C3 state**

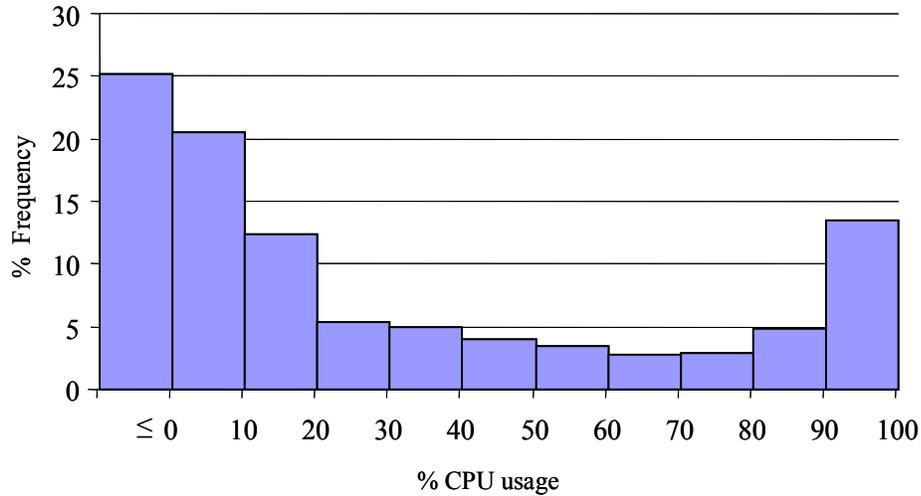
**Table 12: Statistics for CPU State 3 (C3)**

	All Data Points	5°C	25°C	50°C
Mean	47.3	72.5	65.6	26.7
Std Dev	32.8	35.2	33.4	16.4
Minimum	0	0	0	0
Maximum	100	99	100	99

### 3.4.7 CPU Usage

A histogram for the CPU usage is presented in Figure 10. Means, standard deviations, and the range of percentage CPU usage in different ambient temperature conditions are shown in Table 13. An empirical equation for the percentage CPU usage as a function of the CPU states C2 and C3 and CPU throttle is

$$\%CPU\ usage = 96.2 - 2.09*C2 - 0.94*C3 - 0.15*CTh \quad (11)$$



**Figure 10: Frequency chart for CPU usage**

**Table 13: Statistics for Percentage CPU Usage**

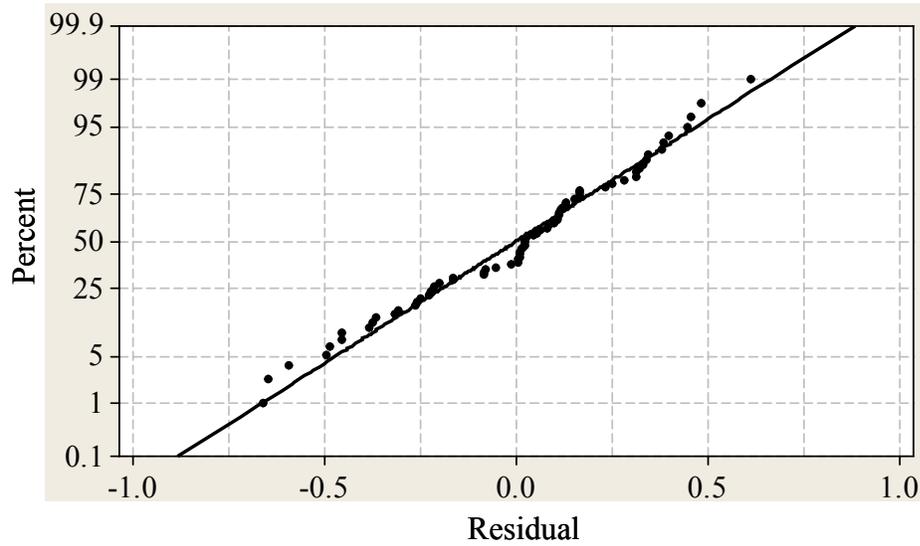
	All Data Points	5°C	25°C	50°C
Mean	31.2	14.2	17.8	45.8
Std Dev	28.7	22.2	22.5	28.3
Minimum	0	0	0	0
Maximum	100	100	100	100

### 3.4.8 Usage Level

The usage level for a notebook computer did not show a strong linear relationship with any individual performance parameter, but showed a weak correlation with several parameters. Therefore, for usage level a non-linear empirical relationship was defined as a function of various performance parameters. It was also found that the usage load on the computer could not be established by just knowing the name of the application running on the system. Characterization of the usage level is based on ambient temperature, humidity, and CPU parameters such as CPU states

C2 and C3, CPU usage, and CPU throttle. Approximately 94% of the variation in usage level is represented by the probability density function of residual data. A normality plot of the residual of the usage level is shown in Figure 11, which shows that the residuals were following normal distribution. Usage level for this study was discretized into four values, and the estimated value was rounded to the nearest integer value for comparisons. The empirical equation for the usage level is non-linear in nature and can be expressed as

$$\begin{aligned}
 \text{Usage Level} = & 8.65 - 0.006 * H + T * (0.17 + 0.0004 * C2 * C3 - 0.0004 * C2 * CPU - \\
 & 0.0014 * C2 * CTh - 0.0001 * C3 * CPU) - T^2 * (0.005 - 0.0001 * CPU - 0.00002 * C2 * CTh) \\
 & - CPU * (0.319 - 0.034 * C2 + 0.0012 * C2^2 - 0.0001 * C3^2 - 0.00002 * C3 * CPU) + \\
 & C2^2 * (0.0200 - 0.0015 * C3 + 0.002 * CTh) - 0.0009 * C3^2 \quad (12)
 \end{aligned}$$



**Figure 11: Residual plot of usage level**

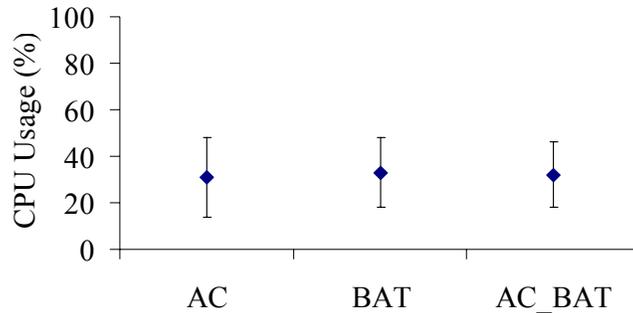
### 3.5 Influence of Environmental, Usage and Power Setting on % CPU

#### Usage

A system's performance parameters will often respond uniquely to environmental factors. Analysis was performed on the experimental data to determine the effects of different environmental and usage conditions on performance characteristics. Total CPU usage is a measure of how much time the CPU spends on user applications and high-level operating system (Windows) functions. Even when the CPU usage is 0% the CPU is still performing basic system tasks, like responding to mouse movements and keyboard input. The total CPU usage measures the amount of time the CPU spends on all tasks, including Windows. This is very useful when evaluating system performance problems based on a specific program. %CPU usage in this study was measured by the operating system, and collected by monitoring software.

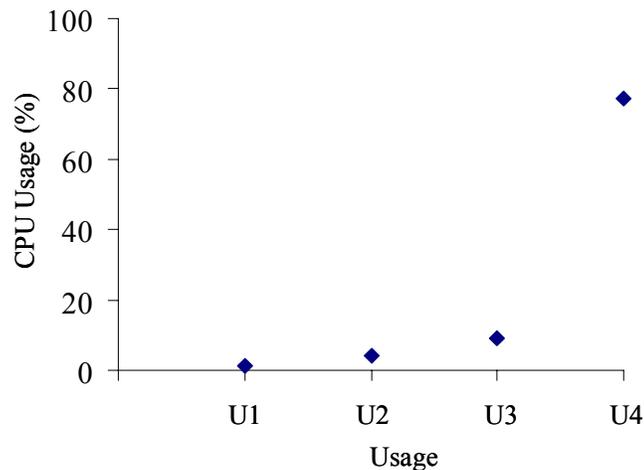
The notebook computers were powered by any one of the three possible power supply methods: AC adapter - while battery was fully charged, battery, and AC adapter - while battery was initially discharged. To neutralize the effect of different environmental and usage conditions, data from all of these conditions were considered together to observe the effect of the power source. Variations in average CPU usage among different power states were not more than 3% and fell within one standard deviation for each power state. The mean and one standard deviation are plotted in Figure 12. The figure shows that average CPU usage (%) did not depend on the power source of the computer, although the spread in CPU usage (%) in the AC adapter condition was greater. To capture more variability in the performance

parameters, the data corresponding to the AC adapter power setting conditions were analyzed.

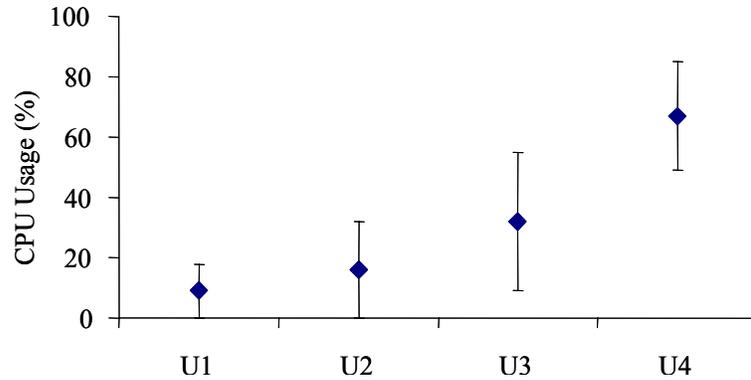


**Figure 12: Variability of %CPU usage with different power source**

The change in CPU usage with different usage/load conditions at room temperature is shown in Figure 13. Figure 14 shows the variability in CPU usage by metric mean and standard deviation as a function of the usage level in the entire range of environmental conditions. This validated the assumption that it is necessary to have different usage/load conditions to baseline the health of a product.

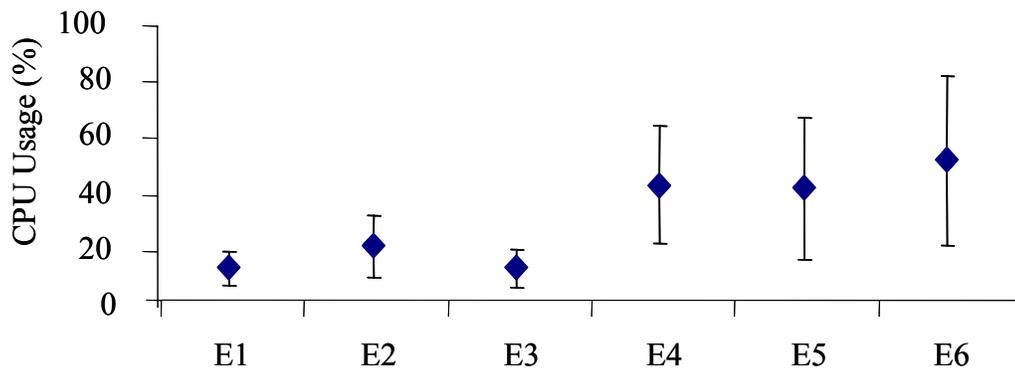


**Figure 13: Variability in %CPU usage with different use conditions at room temperature**



**Figure 14: Variability of %CPU usage with different usage levels**

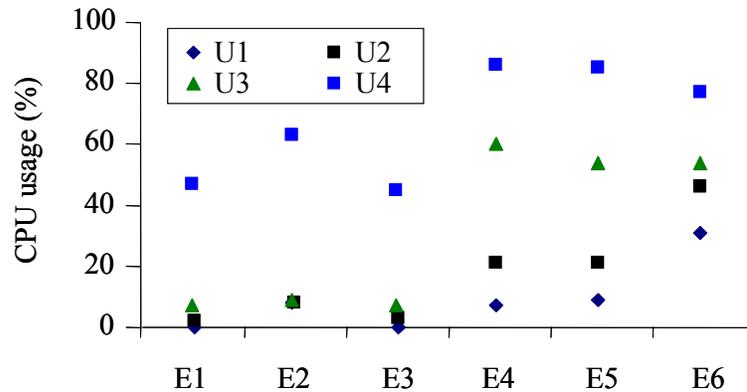
Figure 15 shows the CPU usage metric mean and standard deviation as a function of different environmental conditions. Similarly, this validated the assumption that it was necessary to have different environmental conditions to baseline the health of a product. Again, because the spread was larger for certain usage conditions, it may be possible to select conditions preferable to baseline healthy conditions.



**Figure 15: Variability of %CPU usage with different environmental conditions**

The combined effects of different usage levels under various environmental conditions on percentage CPU usage are presented in Figure 16. Use level 4 differs significantly from the other usage levels, regardless of environmental conditions.

Except for high temperature and humidity conditions, all of the use conditions show results very close to each other. This study found that it is possible to use 50°C/20%RH as a baseline for the CPU usage test to cover the full range of environmental and usage conditions of the notebook computer.



**Figure 16: Percentage CPU usage in different ranges**

Based on the characterization and behavior of each performance parameter under various environmental and usage conditions, system diagnostics can be performed using univariate techniques, such as time series, or by a go/no-go decision based on the differences in the parameters or by using a multivariate distance-based approach, such as the Mahalanobis distance method. The multivariate approach mentioned above considers the correlation of parameters and transforms the multivariate problem into a univariate problem, which is easier to interpret for decision-making and prognostic purposes. For system prognostics, multivariate methods in conjunction with the time series technique can be used.

### 3.6 Summary and Conclusions

This chapter outlines an approach to baseline a commercial electronics product by considering an electronic product's life-cycle profile. This approach could be applied to any system and emphasizes utilizing embedded sensors. It also highlights the need to understand the correlation between and the variability of performance parameters.

To assess environmental and usage conditions associated with electronic products for prognostics, automated program scripts were written to perform typical user activities. These scripts also provided an opportunity to expose all of the computers to a similar workload during the experiment. These scripts reduced the uncertainty that could have arisen due to variations in user activity. The user activity was simulated by defining different usage levels.

The experiments were designed to evaluate variations and trends as well as determine the greatest value a parameter can attain in various usage and environmental conditions. The variability of performance parameters was defined using an empirical relationship as a function of other performance parameters.

In this study, no external sensors were used in order to avoid any possibility of electromagnetic interference, electrostatic discharge, and change in the failure mechanism. Software was built to collect data from the system BIOS, where information from different embedded sensors was collected. This software interfered minimally with system performance.

Analysis of the experimental data revealed that several computer performance parameters were significantly correlated. However, the battery performance

parameters were not significantly correlated to notebook computer performance parameters except the C2 state parameter. This made sense because one would expect longer battery life if the CPU operated in the power saving mode. Various performance parameters were dependent on the ambient temperature, humidity, and product usage. The performance parameters were multi-modal in nature, and a parametric method could not be used for density estimation of performance parameters over the entire range of the experiment. The presence of multiple modes was also observed in performance parameters of an electronic assembly [7]. One should analyse the data before assuming a certain probability density function for the component or system for the entire range of applications in various environments. Non-parametric methods, such as histograms, kernels, orthogonal series estimation, and the nearest neighbour method, can be used to estimate the probability density function for multi-modal data.

A linear empirical model for each performance parameter is defined in the paper. These empirical models construct a simple formula that will predict what value will occur for a parameter of interest when other related parameters take given values. This relationship can assist in system diagnostics, because a probability density function of the residuals (i.e., the differences between expected and observed parameters) obtained for a healthy product can be used for comparison. The probabilistic estimate of deviation from healthy condition of a product would give an indication of the severity of an abnormality. The baseline approach that provides the ability to estimate deviation also sets a platform for performing prognostics.

Using empirical equation developed in the paper can provide estimated value of parameters and parameters' residual value can be obtained by taking difference between the estimated and the observed value. A time-series of parameters' residual can be formed by associating time information to these residuals. A complete history on a system's health can be built by collecting information on systems operation in its entire life cycle. For real-time prognostic assessment, a Markov state model from the various states of a system health can be developed.

In this study, notebook computers were chosen as representatives of complex electronic products. This approach has more significance for a product or system that does not go through hardware changes due to mission criticality and cost, such as satellites and aircraft. For a product that has many variants because of using similar components with different capacities, a new baseline should be established. However, the analysis of data collected on the variants of a product after subjecting it to a set of experiments may provide a scaling factor for each parameter of interest. These scaling factors can be used for defining baseline of other variants of a product instead of running the entire suite of experiments.

## Chapter 4: Fault Identification Approach

Quantification of degradation and fault progression in an electronic system is difficult since not all faults necessarily lead to system failure or functionality loss [1][2]. In addition, there is a significant lack of knowledge about failure precursors in electronics [6]. With limited failure precursors and complex architecture, it is generally hard to implement a health monitoring system that can directly monitor all the conditions in which fault incubation occurs.

The health of a system is a state of complete physical, structural, and functional well-being and not merely conformance to the system's specifications. A health assessment of electronic products can be performed at the product level, at the assembly level, or at the component level [74]. The health assessment procedure should also consider various environmental and usage conditions in which a product is likely to be used.

The built-in test (BIT) and self-test abilities in a system were early attempts at providing diagnostic capabilities incorporated into a system's own structure. Gao and Suryavanshi have catalogued applications of BIT in many industries including semiconductor production, manufacturing, aerospace, and transportation [7]. BIT system applicability is limited to the failure definition embedded at the system's manufacturing stage, whereas with developments in sensor and data analysis capabilities, the development and implementation of data-driven diagnostic systems that can adapt to new failure definitions are now possible.

Today, a product's health can be assessed in many ways, including by monitoring changes in its performance parameters, which are used to characterize a

system's performance; by monitoring canaries (structures that have equivalent circuitry but are calibrated to fail at a faster rate than the actual product); and by estimating accumulated damage based on physics-of-failure modelling [75]. Performance parameter analysis uncovers the interactions between performance parameters and the influence of environmental and operational conditions on these parameters. In the absence of fault-indicating parameters, health assessment can be performed by combining 1) damage estimate information obtained from physics-based models that utilize data from environmental and operating conditions, and 2) failure precursor information extracted from data-driven models [76]. A product's historical data on intermittent failures (i.e., failures that cannot be reproduced in a laboratory environment [77]) should be included in a product's health assessment.

Sun Microsystems developed the Continuous System Telemetry Harness (CSTH) for collecting, conditioning, synchronizing, and storing computer systems' telemetry signals [78]. The Multivariate State Estimation Technique (MSET) provides an estimate of each parameter, and these estimates are later used for decision-making using the Sequential Probability Ratio Test (SPRT) and hypothesis testing. The Mahalanobis Distance (MD) approach considered in this chapter is a distance measure in multi-dimensional space that considers correlations among parameters [9]. The use of the MD approach over the MSET will reduce the analytical burden, because the MD approach provides a number for a system's health determination after combining information on all performance parameters, whereas MSET provides an estimate for each parameter and needs analytical assessment of each parameter for a system's health determination.

Other distance-based approaches that have been used for diagnostics and classification include Manhattan distance, Euclidean distance, Hamming distance, Hotelling T-square, and square prediction error. Manhattan distance is the distance between two points measured along axes at right angles. It has been used to classify text via the N-gram approach [79]. Euclidean distance is the straight-line distance between two points and can be calculated as the sum of the squares of the differences between two points. The Hotelling T-square and square prediction error are used in principal component analysis for representing statistical indices [80]. The Hotelling T-square is a measure that accounts for the covariance structure of a multivariate normal distribution and is computed in reduced model space, which is defined by a few principal components (i.e., the number of principal components used is less than the number of original parameters) [81]. The squared prediction error (SPE) index is a measure that is computed in the residual space that is not explained by the model space [82].

The Manhattan distance, Euclidean distance, and Hamming distance do not use correlation among parameters and suffer from a scaling effect, in contrast to Mahalanobis distance. The scaling effect describes a situation where the variability of one parameter masks the variability of another parameter, and it happens when the measurement ranges or scales of two parameters are different [10]. In order to remove the scaling effect (i.e., eliminate the influence of measurement units) the data should be normalized. The Hotelling T-square and the square prediction error indices are calculated in reduced dimensions (i.e., information loss) and use covariance as opposed to a correlation matrix, which is one reason to consider using MD for fault

diagnosis. MD calculation uses the normalized values of measured parameters, which eliminates the problem of scaling. MD also uses correlation among parameters, which makes it sensitive to inter-parameter “health” changes. For example, consider a set of multi-parameter points that are equidistant (i.e., estimated by Euclidean distance) from a sphere around a location. This location is defined by the arithmetic mean of those points in multi-dimension space. The Mahalanobis distance stretches this sphere to even off the respective scales of the different dimensions and account for the correlation among the parameters.

The performance data of some electronic systems are multi-dimensional, such as multi-functional radio-frequency communication devices, infrared imaging cameras, and hybrid silicon complementary metal oxide semiconductor (CMOS) circuits [83]. While a high-dimensional dataset contains a lot of valuable information, one-dimensional measures are easier to comprehend and can be computed in quick succession.

Consideration of correlations among performance parameters is advantageous as an electronic product experiences diverse environmental and uses conditions. For example, the capacitance and insulation resistance of a capacitor vary with changes in ambient temperature. The effectiveness of a diagnostic procedure increases by incorporating the change in relationship among performance parameters. This is because each performance parameter changes at a different rate with changes in ambient conditions.

In an MD-based diagnostic approach, a healthy baseline and a threshold MD value are needed to classify a product as healthy or unhealthy. In the MD-based

diagnostic approach, traditional methods to define a threshold MD value are either based on personal judgment or traded off to lower the economic consequences of misclassifications, or an MD value is given that corresponds to a known abnormal condition [17]-[20]. These traditional methods do not provide a generic framework to define a threshold MD value for fault identification. The proposed diagnostic method does not require the definition of a faulty product during training and fault isolation, unlike other methods such as clustering and supervised neural networks that require a-priori knowledge of the types of faults during training [84]. When unforeseen types of faults occur, supervised neural networks or clustering approaches may fail to deliver correct decision on systems health [84].

The MD approach suffers from the masking effect if the training data contains a significant amount of outliers [85]. This is because MD uses a sample mean and a correlation matrix that can be influenced by a cluster of outliers. These outliers can shift the sample mean and inflate the correlation matrix in a covariate direction. This is especially true if the 'n/p' ratio is small, where 'n' is the number of observations and 'p' is the number of features. Another issue is related to the computation time needed to reach  $O(p^2)$  for p-dimensionality of feature vectors [86].

This chapter provides a probabilistic approach for defining warning and fault threshold MD values in order to improve upon the traditional approaches where threshold MD values are decided by experts. Since MD values do not follow any distribution and have positive values, a Box-Cox transformation was applied to the MD values to obtain a normally distributed transformed variable. The transformed variable was used to construct a control chart and to define threshold values to detect

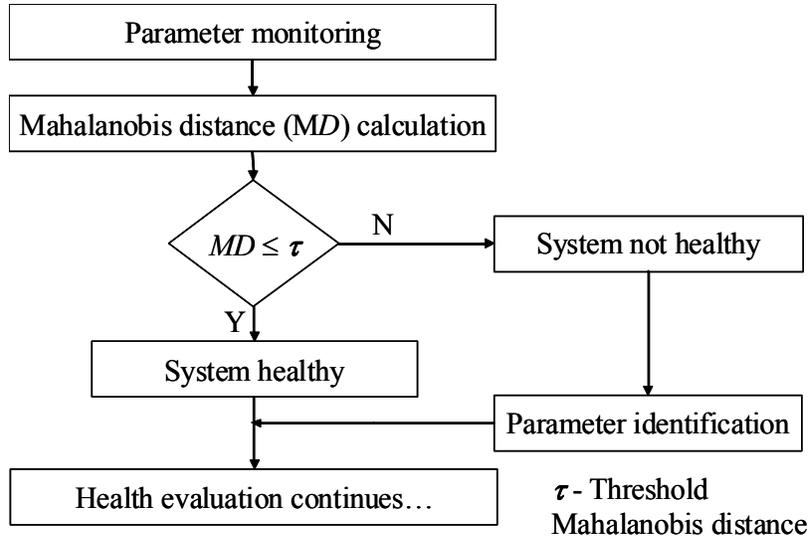
faults. An optimized MD value, using an error function, was obtained to qualify a product against a particular fault. The residual, which is the difference between a parameter's estimated and observed values, was calculated to isolate faulty parameters. A product's health was classified by comparing its MD value, which was computed for each observation, with a threshold MD value.

For fault diagnosis, Betta [11] presented requirements including system monitoring; establishment of a suitable threshold; and estimation of residuals, which can be obtained by the continuous comparison of the system under analysis with another system or by taking the differences between measured and expected quantities. The following section illustrates an MD-based diagnostic approach that meets these requirements, including the creation of a healthy baseline from measured data, an approach to define a threshold for fault detection, and a residual-based approach to identify faulty parameters.

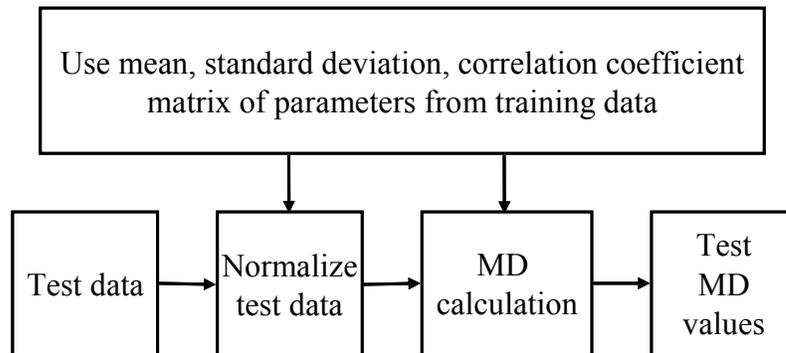
#### 4.1 Diagnostic Approach

The anomaly detection approach (Figure 17) starts with performance parameter monitoring. For a test product, the MD value for each observation is calculated using the performance parameters' mean, standard deviation, and a correlation coefficient matrix that is obtained from the training data (Figure 18). The calculated MD value is then compared with a threshold MD value ( $\tau$ ), which is established from a baseline to classify the product as being healthy or unhealthy. Then, if the product were to be classified as unhealthy, further processing would be performed to isolate the faulty parameter(s) in order to establish reasons for the fault.

The process to define the baseline and the threshold MD values is discussed in the following sections.



**Figure 17: Fault detection approach**

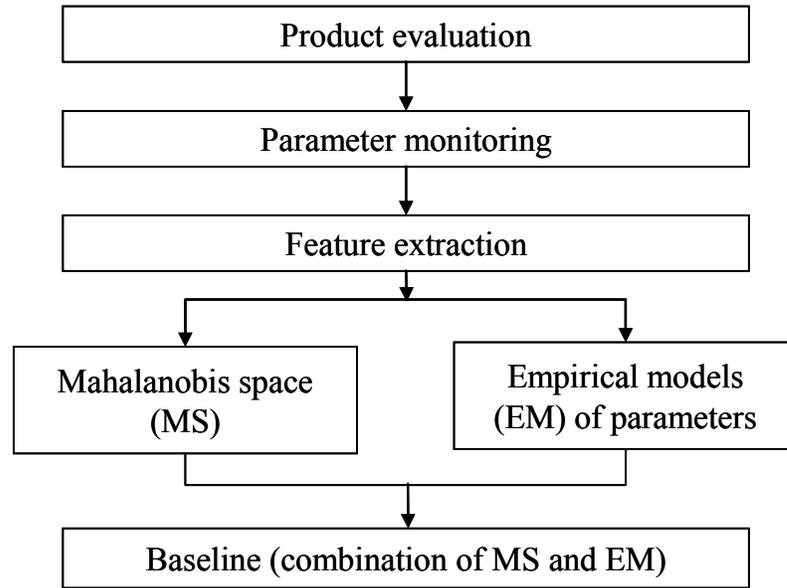


**Figure 18: Mahalanobis distance calculation using test data**

#### 4.2 Healthy Baseline using Mahalanobis distance and Empirical model

A product's performance range is defined by measurements made of its performance parameters in different operating conditions. The combination of performance parameters can be summarized by a distance measure. A baseline

consists of an MD profile, a threshold MD value, and the empirical models of performance parameters. The process of constructing a baseline is shown in Figure 19.



**Figure 19: Baseline establishment methodology**

The baseline construction process starts with the functional evaluation of a product. Based on a failure modes, mechanisms, and effects analysis (FMMEA) of a product, parameters that represent product performance should be selected for monitoring [1]. These parameters are monitored during the operation of a set of healthy products under various environmental, operational, and usage conditions. The collected information on parameters forms a data set that is used to train and calculate the statistical features of each parameter. For MD calculation, performance parameter data are normalized and a correlation coefficient matrix is formed. The correlation coefficient between two parameters expresses the linear dependency of one parameter on the other and the direction of the dependency. The MD values corresponding to each observation in the training data are calculated, and this group of MD values

forms the Mahalanobis space (MS). From the MS, the min-max range, mean, and standard deviation of MD values are obtained to explain the variability of a healthy product's performance in terms of MD values.

Empirical models of performance parameters are developed in the absence of analytical models. Training data are used to compute the correlation coefficients between different parameters and identify parameters to be used for empirical models. The linear modeling approach was chosen because of its simplicity and effectiveness without losing much model-fitting accuracy. One can use non-linear models for parameter estimation, but non-linear models need relatively complex learning algorithms to fit the underlying relationship among parameters [87]. In our application, the training data, collected under various operation conditions of a set of healthy products, are linear. Thus, a linear model for each performance parameter is developed as a function of other related performance parameters (Chapter 3). Linear models are considered appropriate due to their simplicity and considerable fit (i.e., >90%) to the experimentally collected data. These models are used for isolating parameters that are behaving far differently from expectations.

### 4.3 Threshold Determination

In this section, a probabilistic approach is presented to determine two types of threshold MD values. First, a generic threshold for detecting any type of fault or anomaly present in a product based on the MDs obtained from the training data is determined. Second, a fault-specific threshold for detecting the presence of a particular fault based on historical data related to a particular fault is determined. The second threshold can be considered a second-tier fault isolation process.

### 4.3.1 Generic Threshold Determination

An approach for determining a generic threshold—an MD value—for fault diagnosis is shown in Figure 20. The MDs are always positive, but they do not generally follow a normal distribution. The Box-Cox power transformation can be used to transform a variable that has positive values and does not follow a Normal distribution into a normally distributed transformed variable [88]. The Box-Cox transformation is defined as follows:

$$\begin{aligned} x(\lambda) &= \frac{(x^\lambda - 1)}{\lambda} & \lambda \neq 0 \\ x(\lambda) &= \ln(x) & \lambda = 0 \end{aligned} \quad (13)$$

where the vector of data observations is  $x = x_1, \dots, x_n$  and  $x(\lambda)$  is the transformed data.

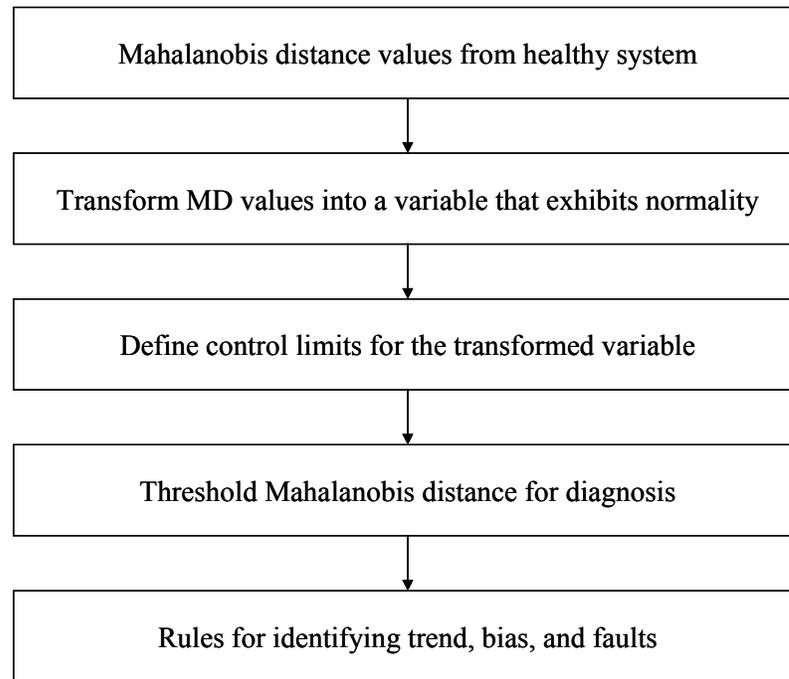
The power  $\lambda$  is obtained by maximizing the logarithm of the likelihood function (14):

$$f(x, \lambda) = -\frac{n}{2} \ln \left[ \sum_{i=1}^n \frac{(x_i(\lambda) - \bar{x}(\lambda))^2}{n} \right] + (\lambda - 1) \sum_{i=1}^n \ln(x_i) \quad (14)$$

where

$$\bar{x}(\lambda) = \frac{1}{n} \sum_{i=1}^n x_i(\lambda) \quad (15)$$

The normality of  $x(\lambda)$ , a transformed variable, is confirmed by plotting it into a normal plot. The mean ( $\mu_x$ ) and standard deviation ( $\sigma_x$ ) of the transformed variable are used to determine the control limits of an x-bar chart. A threshold value corresponding to the warning limit ( $\mu_x + 2\sigma_x$ ) and a threshold value corresponding to a fault alarm ( $\mu_x + 3\sigma_x$ ) are defined. Since higher MD values are of concern from an “unhealthiness” perspective, the upper portion of the control chart is of importance for identifying changes in system health. Rules from quality control, including bias and variance identification, can be used [89].



**Figure 20: An approach for defining threshold MD value**

#### 4.3.2 Fault-Specific Threshold Determination

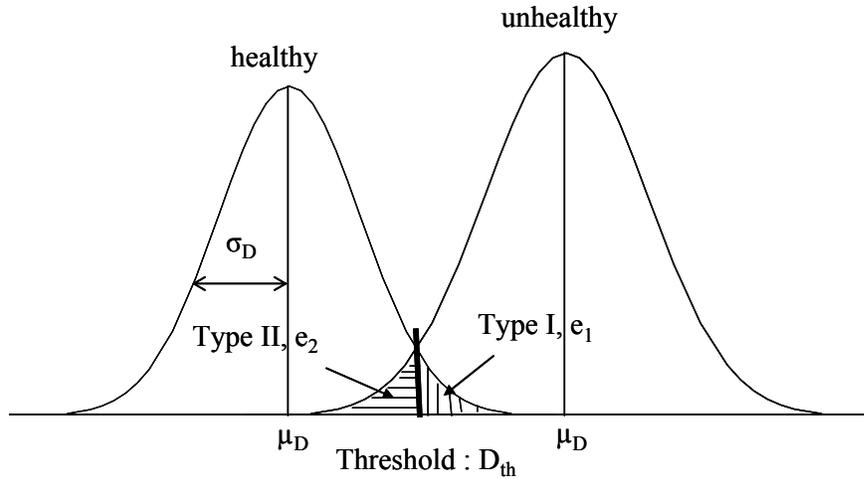
A normally distributed transformed variable, which corresponds to MD values, can be used to determine Type I and Type II errors [19]. A Type I error, often referred to as a false positive, is a statistical error made in testing the health of a product where the product is healthy but is incorrectly determined to be unhealthy. A Type II error, often referred to as a false negative, is a statistical error made in testing the unhealthiness of a product where a product is determined to be healthy when it is not. Figure 21 illustrates Type I and Type II errors using a variable's distribution for a healthy and an unhealthy system, where the healthy distribution is defined from the training data, and the unhealthy distribution is defined from the data representing a specific fault in a system.

For a known fault, an optimal transformed variable can be defined such that

the combined error (i.e., the sum of Type I and Type II errors) remains minimal (i.e., the shaded region in Figure 21), and an MD value corresponding to the optimal transformed variable ( $x$ ) is calculated. For a healthy product, the probability of having MD values higher than the threshold value is the number of observations that produce an MD value higher than the threshold MD value divided by the total number of observations for a healthy product. Similarly, for an unhealthy product, the probability of having an MD value less than the threshold value is the number of observations that produce MD values less than the threshold MD value divided by the total number of observations for an unhealthy product. The threshold value ( $\tau_x$ ) of a transformed variable for detecting a known anomaly is established using the following error function ( $\varepsilon$ ):

$$\varepsilon(\tau_x) = \frac{e_1}{n_h} + \frac{e_2}{n_u} \quad (16)$$

where  $\tau$  is the threshold,  $e_1$  is the number of observations classified as unhealthy in the healthy population ( $n_h$ ), and  $e_2$  is the number of observations classified as healthy in the unhealthy population ( $n_u$ ). The threshold value is obtained by minimizing the error function (i.e., by choosing a different value for  $\tau_x$ ).



**Figure 21: Illustration of threshold value calculation**

#### 4.4 Case Study

Experiments were performed on ten state-of-the-art (2007) notebook computers that were produced by the same manufacturer. As part of the test plan, it was necessary to assess the performance of the products under various environmental and usage conditions. The computers used for this study were exposed to different environmental and usage conditions during the experiments, and their performance parameters were monitored in-situ. Since not all conditions could be tested, certain extreme and nominal conditions were included. The software usage conditions—a set of computer users activities representative of typical computer uses—were defined [72]. These usage conditions were executed through a script file, where all user activities were encoded.

To study the variability in performance parameters, experiments were conducted under six different environmental conditions, as shown in Table 2. The test temperature range was from 5°C to 50°C, which was wider than the specified

operating and storage temperature range of the computer in order to include variation in operating conditions beyond the manufacturer-specified range. In each environmental (temperature-humidity combination) condition, four usage conditions and three power supply conditions were considered [82]. The test duration depended upon the way the computer was powered. When the computer was powered by an AC adapter and the battery was fully charged (relative state of charge (RSOC) = 100%), the test ran for 3.5 hrs. When the computer was powered by an AC adapter when the battery was fully discharged (i.e., RSOC < 4%), the test duration was determined by the time the battery took to fully charge (RSOC = 100%). When the computer was powered by its battery only, the test duration was determined by the discharge (RSOC < 4%) time. The tests were conducted in a temperature-humidity chamber and in a room-ambient environment. Table 3 shows all 72 experiments. Each computer was turned on for 30 minutes before the experiment was started. The computers were kept at room temperature between each test for 30 minutes.

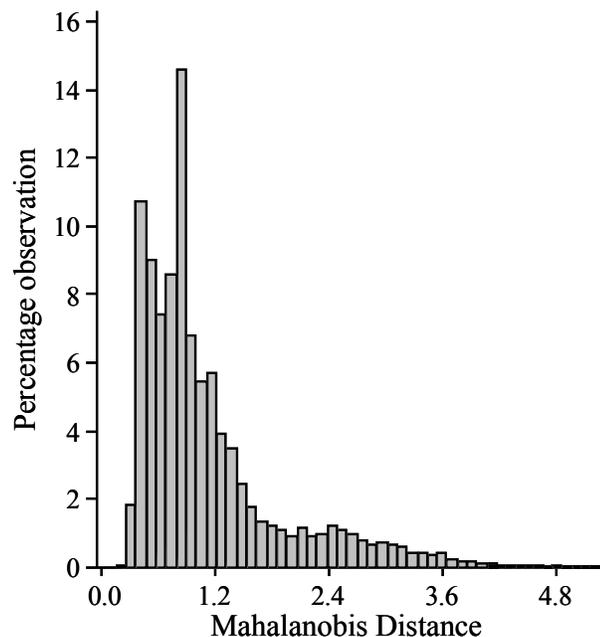
The correlation coefficients among performance parameters were calculated. Only significant correlation coefficients (for which the Pearson probability was less than 0.05) between two performance parameters are shown in Table 5. The training data was formed by eight correlated performance parameters (listed in Table 5). The parameters measured were fan speed (speed of a cooling fan in rpm), CPU temperature (measured on the CPU die), motherboard temperature (measured on the top surface of the printed circuit board near the CPU), videocard temperature (measured on the graphics processor unit), %CPU usage (measure of how much time the processor spends on a user's applications and high-level Windows functions), and

%CPU throttle (measure of the maximum CPU percentage to be used by any process or service, thereby ensuring that no process can consume all of the CPU's resources at the expense of other users or processes). The parameters C2 and C3 are power saving states of the CPU in which the processor consumes less power and dissipates less heat than in the active state [90]. C2 and C3 represent the percentage time a processor spends in the low-power idle state and are a subset of the processor's total idle time. In the C2 power state, the processor is able to maintain the context of the computer's caches. The C3 power state offers improved power savings and higher exit latency over the C2 state. In a C3 power state, the processor is unable to maintain the coherency of its caches. All the parameters mentioned in Table 5 were sampled at different rates: CPU operation at every 5<sup>th</sup> second, and temperatures and fan speed at every 30<sup>th</sup> second.

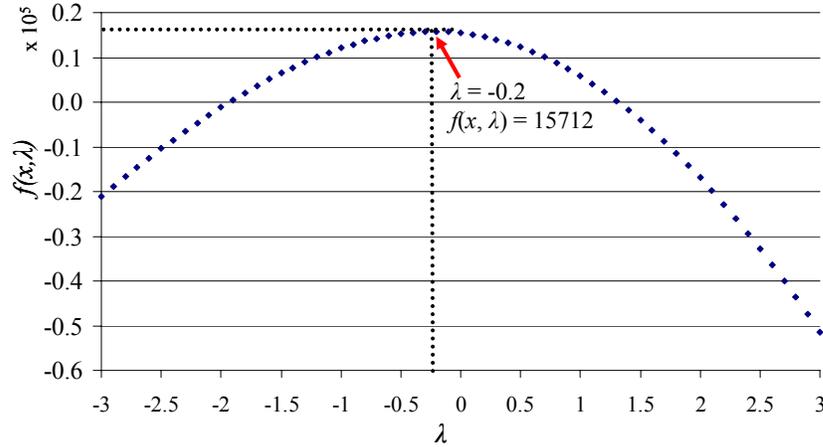
The Mahalanobis distance for each observation in the training dataset was calculated using Equation (3). According to the flow chart shown in Figure 19, a healthy baseline was defined using MD values and empirical models of performance parameters. The training data was comprised of approximately 25,000 observations. The distribution of these MD values corresponding to the training data is shown in Figure 22. Empirical models for each performance parameter were developed as functions of other performance parameters using training data. The "residuals" of each parameter were calculated by subtracting the estimated value from the observed value. For example, an empirical model for CPU temperature as a function of fan speed, motherboard temperature, and video card temperature is presented in Equation 6.

The residual analysis of CPU temperature indicated that a probability density plot of CPU temperature residuals (Figure 4) represents 94% of the variability in the CPU temperature. The residual analysis indicated that the mean residual for fan speed was up to 500 rpm. Similarly, the mean residual for the CPU temperature was 5°C, and for the motherboard and videocard temperature it was 8°C. Similar empirical models for other parameters have been developed [91].

Two types of threshold MD values were determined. First, a generic threshold to detect faults at the product level was developed, and second, a specific threshold for detecting the presence of a particular fault was developed. For generic threshold value determination, the Box-Cox transformation was applied on the training MD values, and an optimized value of  $\lambda$  ( $= -0.2$ ) was obtained by maximizing the likelihood function defined earlier. The plot of  $\lambda$  and the likelihood function  $f(x, \lambda)$  is shown in Figure 23.

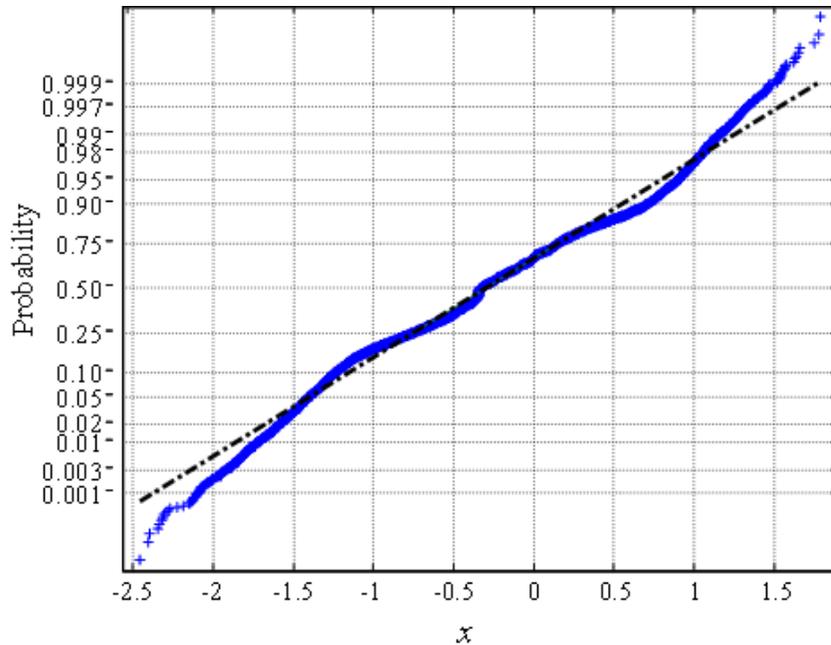


**Figure 22: Histogram of MD values for healthy population**



**Figure 23: Plot of  $\lambda$  and likelihood function  $f(x, \lambda)$**

The optimal  $\lambda$  value was used to obtain a normally distributed transformed variable ( $x$ ) from the training MD values. The normal probability plot of  $x$  is shown in Figure 24. A control chart for fault identification was developed where control limits were calculated using the mean and standard deviations of the transformed variable,  $x$ . A warning limit and a fault limit corresponding to  $\mu_x + 2\sigma_x$  and  $\mu_x + 3\sigma_x$  were defined. For fault identification, two rules were used: first, one or more points fall above the fault limit (i.e.,  $\mu_x + 3\sigma_x$ ), and, second, two (or three) out of three consecutive points are within the fault and warning limits (i.e., Zone A). From the training data only one data point fell above the fault limit, and 1.5% of the data fell in Zone A (i.e., above the warning limit). The threshold MD values corresponding to the warning limit and fault limit were 3.4 and 9.1, respectively. Quality control rules were applied to determine the bias and trend in the data along with the identification of faults [89]. Data exhibits a trend if six (or more) consecutive points are increasing or decreasing. Data exhibits biasness if nine (or more) consecutive points fall on one side of the central line (i.e.,  $\mu_x$ ). Since MD values increase with abnormality, data that fall above the central line of the control chart are of more concern.



**Figure 24: Normal probability plot of transformed variable  $x$**

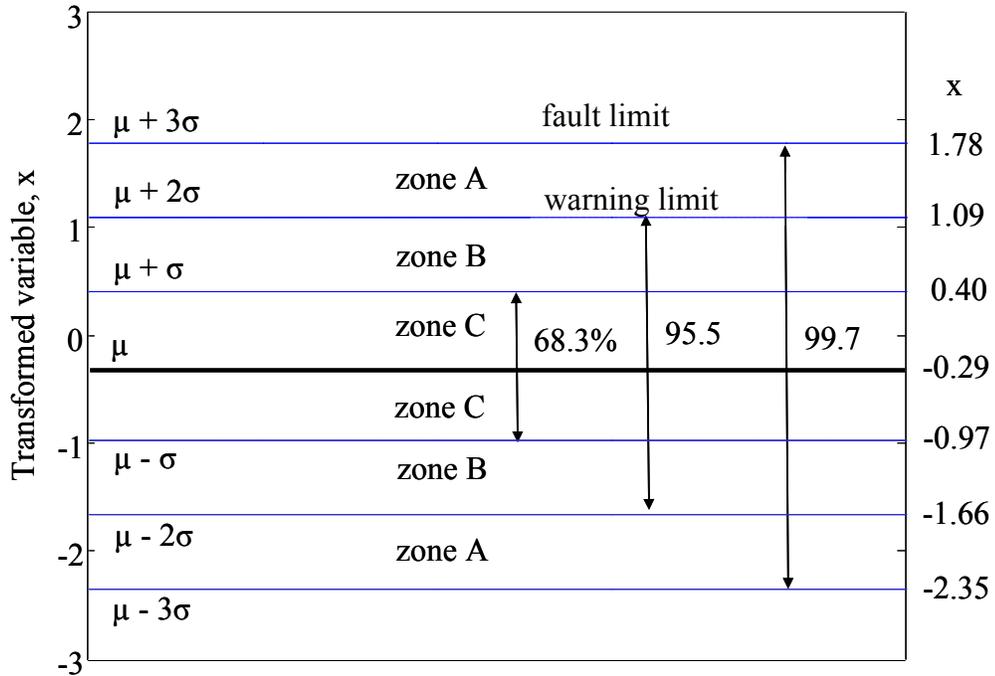
A set of data obtained from one test notebook computer, which was field-returned, were plotted on the control chart constructed for the transformed value ( $x$ ), and the quality control rules were applied. The observations made (Table 14) were as follows: 62.1% of the data were above the failure alarm limit ( $\mu_x+3\sigma_x$ ) (i.e., 62.1% of the data indicates the presence of faults in the test system) in comparison with 0% for a healthy system. And 37.8% of the data were within Zone A (i.e., 37.8% of the data indicates the tendency of the test system to be faulty) in comparison with 1.5% for a healthy system. This also indicates that 99.9% of the data were above the warning limit,  $\mu_x+2\sigma_x$ , in comparison with 1% for a healthy system. At 96% of the time, the data indicated the presence of a trend in the test system in comparison to 2% for a healthy system. All test data (i.e., 100%) were on one side of the average, which that indicated the presence of biasness in the test system in comparison to 33% for a healthy system. The marginal difference between a healthy system and the test system

suggested that the test system had problems, which is identified using the fault isolation approach.

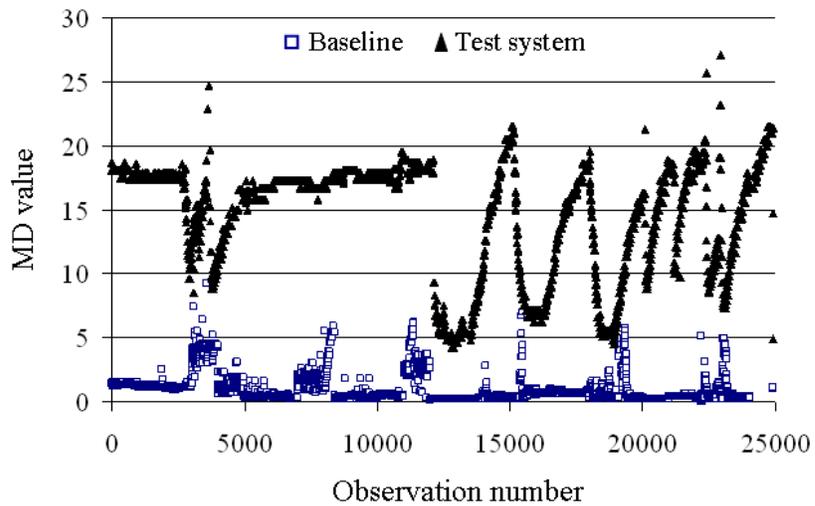
**Table 14: Percentage of Alarms Raised by Different Rules**

No.	Rules	Healthy	Test
1	One or more points fall outside control limits	0	62.1
2	Two (or three) out of three consecutive points are in Zone A	1.5	37.8
3	Six (or more) consecutive points are increasing or decreasing	2	96.0
4	Nine (or more) consecutive points are on one side of the average	33	100

The MD values corresponding to the baseline (i.e., healthy) and the test computer are shown in the Figure 26. Both sets of MD values obtained from the training and test data sets were transformed into normally distributed variables. To detect a specific fault, a threshold MD value corresponding to that fault was defined using the error function approach discussed earlier.



**Figure 25: Control limits for fault identification**



**Figure 26: MD value for a baseline and a test system**

An optimal threshold MD value,  $\tau$ , was calculated by minimizing the error function (Equation (16)). The amount of error,  $\varepsilon$ , considering different MD values is shown in Figure 27. In this study, the optimal MD threshold value,  $\tau$ , was 4.70, and

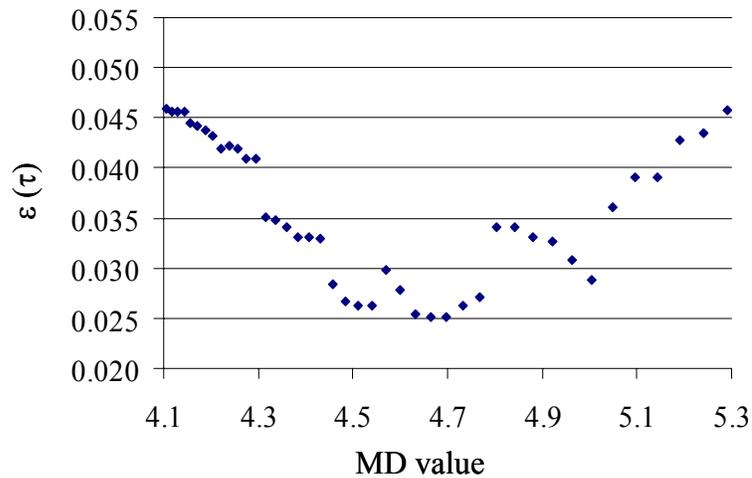
the error,  $\varepsilon(\tau)$ , was 0.025 (i.e., 2.5% misclassification, where 1.8% was contributed by the training data and 0.7% was from the test data). Higher misclassification of training data suggests that the defined threshold value was conservative in nature because the healthy product was misclassified more than the unhealthy product.

The validity of the defined threshold value was evaluated by calculating the misclassification of training data and test data at various threshold values (Figure 28). The graph in Figure 28 indicates that lowering the threshold value resulted in an increase in the number of observations from the training data being classified as faulty (misclassification of healthy data as unhealthy data increased). Similarly, increasing the threshold value resulted in an increase in the number of observations being classified as healthy from the test data (misclassification of unhealthy data as healthy data increased). Large-percentage changes in misclassification were not observed even after changing (increasing or decreasing) the threshold value of MD by 10%, and so the threshold value can be considered robust.

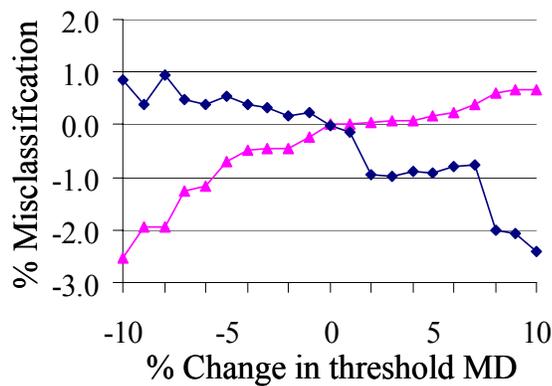
The performance parameter residuals were used to isolate the parameters that were responsible for the drift in the health of the test computers. A few test data samples are presented in Table 15, where the parameters measured ( $M$ ) and estimated ( $E$ ) values are shown. From the residual analysis, it was observed that the residual of the fan speed was greater than expected in 90% of the instances, and in 10% of the instances the residual of the temperature parameters were greater than expected. The fan was judged faulty based on the residual analysis, and this judgment was verified by investigation of raw data.

The case study demonstrated that the methodology presented was capable of

identifying faults. A baseline generated from experimental data can be used to successfully analyze the onset of a fault and the eventual failure of a similar computer. The diagnostic approach can be applied to any product, but the case study results (and the baseline) cannot be extrapolated to all products and their variations. It would be expected that the product developers would develop baselines for their products of interest.



**Figure 27: Optimal threshold evaluation**



**Figure 28: Robustness evaluation of threshold value**

**Table 15: Estimated Values of the Parameters**

Observation number	MD	Fan speed		CPU temp		Motherboard temp		Videocard temp	
		M	E	M	E	M	E	M	E
4001	7	0	17	54	44	51	43	50	60
5001	11	0	1167	65	57	60	50	60	73
6001	9	0	903	65	60	62	58	62	68
7001	9	0	757	66	60	62	58	62	69
8001	9	0	1043	66	60	62	57	62	70
9001	10	0	1184	67	61	63	53	63	71
...	...		...	...	...	...	...	...	...
12001	10	0	1410	67	61	63	64	63	70

#### 4.5 Summary and Conclusions

This chapter presents a data-driven diagnostic approach that utilizes Mahalanobis distance (MD). Instead of using expert-opinion-based threshold MD value, a probabilistic approach was developed to establish threshold MD values in order to classify a product as being healthy or unhealthy. An error function was defined and minimized in order to determine a reference MD value to identify the presence of a specific fault in a product. Once faults are detected, a set of specific threshold values developed using the residuals of the performance parameters can be used for isolating known faults. This chapter demonstrated that the distribution of the

residuals of performance parameters can be used to isolate parameters that exhibit faults.

This chapter presented an approach to construct an MD control chart from a system's performance data. The control chart enables continuous monitoring of a system's health using the MD value calculated from the system's performance data. This MD control chart concept can also be used by the manufacturing industry for continuous process monitoring instead of following several performance parameter control charts.

Rules for detecting faults and observing trends and biases in a system's performance are presented in this chapter. The ability to identify trend and biasness in the data will enable the development of new tests to identify flawed system and processes. The ability to detect trends and biasness in system health by observing a control chart constructed for MD values will allow for the detection of changes in a product's health before it experiences failure.

The case study on notebook computers demonstrates that the approach to define threshold MD value is a major improvement. The defined thresholds were able to detect faults in a product with 99% accuracy. In a known fault condition, a specific threshold was defined, which classified a product with 97.5% accuracy (i.e., 2.5% error). The residuals analysis of the performance parameters identified the fan as a problem 90% of the time. The temperature parameters that are correlated to the fan operation identified as a problem 10% of the time. The results demonstrated that the suggested approach for defining a threshold MD value for the diagnostic approach

was able to identify faults. The residual-based parameter isolation approach identified the cause of the problem.

MD is a good health measure that summarizes multiple monitored parameters that are correlated as a number. With the modifications presented in this chapter, MD will benefit manufacturers in controlling the quality of their products and processes on-line or off-line.

## Chapter 5: Fault Isolation Approach

System application areas call for enhanced safety, reliability, and maintainability in ways that reduce their vulnerability to serious failures and failures of their host systems. The increasing complexity of systems and their use in a wide range of environmental conditions increases the burden on system designers and maintainers to develop effective tools to provide a system's real time health assessment [92]. Prognostics and health management (PHM) involves continuous, autonomous, real-time monitoring of a system's health by means of embedded or attached sensors with a minimal level of manual intervention in order to evaluate the system's actual life-cycle conditions, determine the advent of failure, and mitigate system risks [1] [33]. For health management and the development of fault-tolerant systems, realization of system failure, recognition of failure sites, and quantification of damage are essential [31]. An effective health management tool's basic requirements include identification of faults and failures and fault isolation [12].

A fault is an unexpected deviation from acceptable behavior of at least one performance property of a system. Failure is a permanent interruption of a system's functionality or a time lag in the system's expected functionality under specified operating conditions. Realization of a system failure is not always easy because some faults may not lead to the loss of functionality of a system; therefore, definition and characterization of faults is required in the fault detection process. Fault isolation is the determination of the type and location of a fault, and it follows fault detection

[93][12][94]. Identification of performance parameters is essential for locating the faulty components that are responsible for a system's failure, because one or more performance parameters might provide precursors to the failure [39]. Identification of parameters that are indicating faults is an important step in the fusion prognostics approach [96]. These parameters are then utilized for damage estimation through physics-of-failure models and for identifying the presence of a trend in a system's performance. For better estimation of the remaining useful life of a system, the output from the PoF and data-driven methods can be combined [97]. The combined output will have better estimates and less uncertainty.

Fault isolation methods are broadly grouped into two classes: model-based and data-based. Model-based methods are generally functionality-dependent: system functionality is modeled into mathematical form and residuals obtained from model are connected to specific faults [98][99]. These methods are based on a deterministic process model that must be correct for a system's proper functionality. This approach is suitable for isolating specific known faults. On the other hand, data-based methods rely on performance parameter measurements. For fault detection, the definitions of thresholds are based on the measurements made during a system's healthy operation. For fault isolation, different abnormal health states of a system are defined using data collected when a system was operating abnormally in order to distinguish different faults. Fault isolation is accomplished by comparing a system's current health state to the known regions of the system's healthy state space. Some data-based approaches consider the contribution of particular states to the overall shift from healthy states to identify and isolate faults [100][101].

The data-based fault isolation approach includes dimensionality reduction techniques, state-based techniques, regression techniques, and techniques based on distance measures. One of the dimensional reduction techniques, principle component analysis (PCA), has been used for handling large amounts of data [102]. In the PCA approach, reduction in dimensionality results in information loss [103][102]. In this technique, the contributions of different parameters to a distance measure are analyzed to identify faulty parameters. However, a parameter contributing more does not indicate that the parameter is exhibiting anomalous behavior because these are the parameter's contributions in the variability of an observation.

State-based techniques define state models for parameters in order to capture linear or nonlinear characteristics of a system's performance parameters. The analysis of residuals obtained from parameters' observed and estimated values from the state model are used for characterizing faults [104]. In this characterization process, a model is defined for each fault type to isolate the fault. However, these fault-specific models are only useful for identifying specific known faults. The regression model of performance parameters is another approach to obtain a parameter's estimate, but it often fails to control the measurement error and intervening variables [105].

Distance-based methods have been used for fault detection. The Mahalanobis distance (MD) has been used for fault detection in computers [106][82]. In addition, MD has also been used for fault isolation where data from both healthy and unhealthy systems were used to isolate faults [107].

This chapter presents a new fault isolation approach using residual MD. This work on fault isolation is different from and improvement over earlier work presented

in previous chapter. Here individual parameters contribution to system health (i.e., represented by MD) is utilized to estimate residual for each parameters [107]. Whereas the previous chapter, the difference between estimated and observed value of each parameter were used to calculate residuals for each parameter. The MD is selected because of the two major advantages: first, MD reduces a multivariate system to a univariate system without losing information; and second, MD is sensitive to inter-variable changes in multivariate system parameters. It includes all observed performance parameters in defining the health of a system (i.e., no information loss in the health estimation process). The proposed method uses only health data during training and fault isolation does not require information on faulty conditions, so it does not depend on any specific fault type. The first stage in this method is to create a baseline using MDs to detect faults. The second stage is to create residual bounds using residual MDs to identify faulty parameters. The approach defines the threshold bounds on MD residuals for parameters, which are estimated from healthy data. The residual for each parameter (i.e., impact of each parameter on MD) is obtained by performing experimental analysis planned by the Design of Analysis (DoA) based on Design of Experiments (DoE). The design of experiment concept has been used to determine parameters that contribute most to Mahalanobis distance [108].

Experiments were performed on computers in order to validate the proposed method. Two kinds of threshold were generated from the healthy population for the proposed fault isolation method, and a case study was conducted to demonstrate and validate the approach. The experimental details, the algorithmic approach to fault

isolation, and three case studies are discussed in later sections of this chapter. MD has been discussed in previous chapters and DoA is briefly described below.

### Design of Analysis

In this chapter, a new analysis process namely Design of Analysis (DoA) is developed for identifying the influence of an input parameter on an output. DoA provides a structured, organized method for determining the relationship between different input parameters affecting a process and the output of that process. In this approach, makes observations on parameters influence are made methodically as directed by a systematic table, which is based on the Design of Experiments (DoE) [108]. The systematic table is designed by a set of parameter combinations, which is varied systematically. In order to perform fault isolation, DoA implements “one change at a time” because it allows to make a judgment on the significance of an input variables for the output. Each set of parameter combination in the systematic table of DoA is referred as an analytical run in the chapter. The parameters that influence the process output (i.e., MD) are identified.

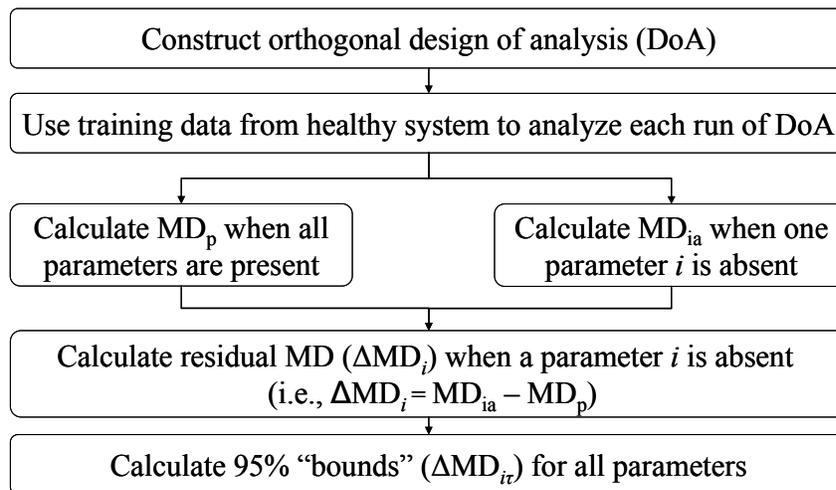
## 5.1 Faulty Parameter Isolation Methodology

The first step in the proposed method is to identify the fault, and the second step is to isolate the fault by identifying the parameter indicating the fault. The fault detection approach (Figure 17) starts with the monitoring of the system’s performance parameters under different life cycle scenarios. A system’s life cycle includes different environmental and usage conditions. In the data acquisition process, a product’s performance is monitored using built-in, embedded, or external

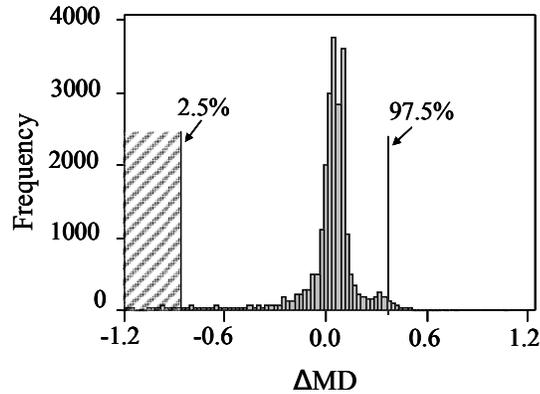
sensors, and data is stored after data scrubbing. The performance parameters collected during a healthy system's operation under different usage conditions makes healthy training data. For fault detection, a system's health is classified by taking the MD value corresponding to each observation (i.e., performance parameter vector) and comparing it to a threshold MD value ( $\tau$ ), which is determined from a set of training (healthy) data. In a situation where a system is classified as unhealthy, which is based on a sequence of continuous faulty observations, further investigation is performed to identify the fault type using the fault isolation approach.

The fault isolation approach discussed here is based on the residual MD value. The MD value for each observation is calculated for every analytical run decided in DoA. The residual MD value for each parameter is obtained by taking the difference between MDs corresponding to two sets of analytical run, which are different in terms of a parameter's absence. The distribution of residual MDs is formed from training (i.e., healthy) data and is used to decide the threshold residual MD value ( $\Delta MD_\tau$ ) for each parameter. The threshold defining procedure (Figure 29) starts with constructing an orthogonal analytical run in DoA by considering "one change at a time." The training data for these experiments are extracted from the healthy data set collected after monitoring healthy systems. For each analytical run the MD value for each observation is calculated, where  $MD_p$  represents an MD calculated considering that all parameters are present, and  $MD_{ia}$  is an MD where parameter  $i$  is absent. Test systems are monitored and analyzed in a similar way. The residual MD value for each parameter corresponding to each observation is compared with each parameter's residual threshold.

The residual MD corresponding to a parameter  $i$  is represented by  $\Delta MD_i$ . The  $\Delta MD_i$  is obtained by subtracting the  $MD_p$  from  $MD_{ia}$  for each observation. The distribution of  $\Delta MD_i$  for a parameter represents that parameter's contribution to MD. A threshold value ( $\Delta MD_{it}$ ) corresponding to a 95% bound on the  $\Delta MD_i$  for each parameter is defined as an expected range of  $\Delta MD$  for parameter  $i$ . Figure 30 illustrates the distribution and threshold bound for a parameter. The upper bound (i.e.,  $+\Delta MD$ ) on  $\Delta MD$  indicates changes in performance but no loss in information, whereas the lower bound (i.e.,  $-\Delta MD$ ) indicates a loss in information, because the MD value should increase when there is an abnormality. Since negative  $\Delta MD$  represents a loss in information due to the absence of a parameter during MD calculation, a parameter with a  $\Delta MD_i$  below the lower bound of the threshold range ( $\Delta MD_{it}$ ) is picked as the potentially faulty parameter.

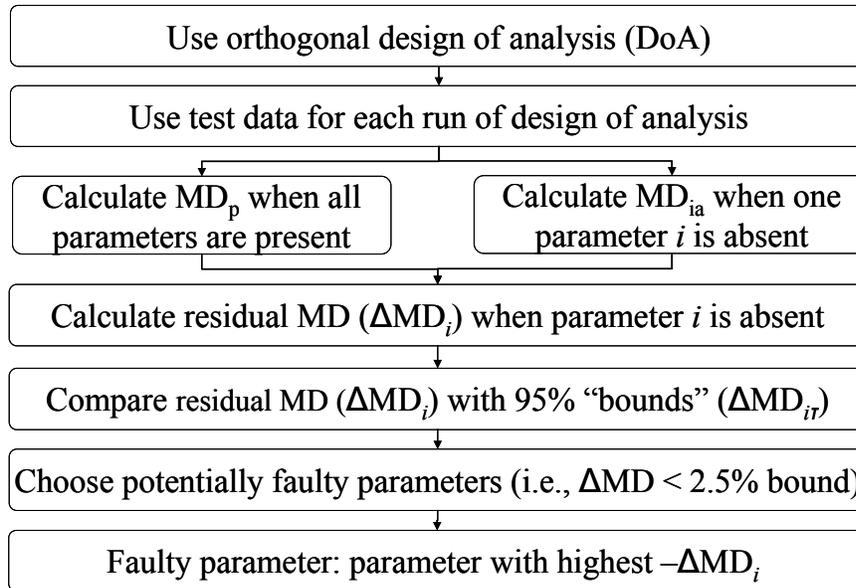


**Figure 29: Parameter isolation: threshold determination for parameters ( $\Delta MD_{it}$ )**

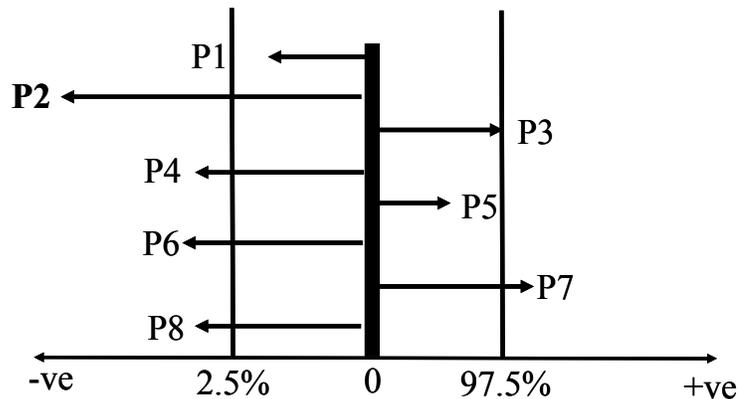


**Figure 30: Threshold determination**

The procedure for isolating a faulty parameter for a test system is shown in Figure 31. Every observation from a test system is analyzed for each parameter. The  $\Delta MD_i$ , where  $i$  represent a parameter, for each test observation is calculated and compared with the parameter's threshold,  $\Delta MD_{it}$ . The parameter with a  $\Delta MD_i$  below the lower bound of the threshold range ( $\Delta MD_{it}$ ) is picked as the candidate for the faulty parameter, and the highest  $-\Delta MD_i$  is determined to be the faulty parameter. Figure 32 indicates that several parameters are candidates for being considered faulty (P2, P4, P6, and P8), and a parameter that has the highest negative  $\Delta MD$  is chosen as the faulty parameter (P2). Other parameters are evaluated as well in order to validate the faulty parameter identification.



**Figure 31: Parameter isolation (test data evaluation)**



**Figure 32: Isolating faulty parameter in an observation based on  $\Delta MD$ .**

To identify the faulty parameter, the residual MD for that parameter is compared with the threshold residual MD. A parameter is determined to be faulty if the probability estimate of being faulty is significantly higher than other parameters. Expert knowledge is used to determine a faulty parameter if several parameters are identified as being equally faulty. This is another approach to understanding system failure.

## 5.2 Case Study

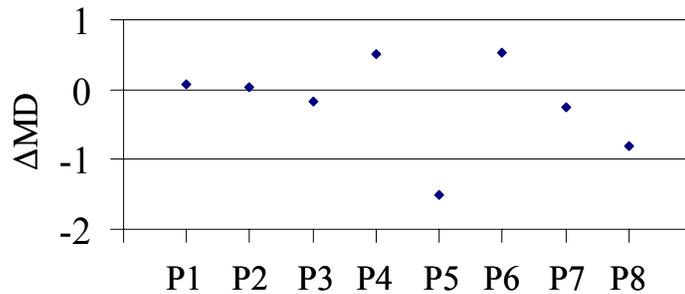
This fault detection and faulty parameter isolation methodology was applied to computers in order to demonstrate and validate the suggested method. Experiments were conducted on ten brand new identical computers with the assumption that these computers were representative of healthy systems. The data from these computers were used to define a healthy baseline and to study the parameters' behaviors. The parameters measured were fan speed (speed of a cooling fan in rpm, P1), CPU temperature (measured on the CPU die, P2), motherboard temperature (measured at the top surface of the printed circuit board near the CPU, P3), videocard temperature (measured on the graphics processor unit, P4), %CPU usage (measure of how much time the processor spends on a user's applications and high-level Windows functions, P5). The C2 (P6) and C3 (P7) are power saving states of the CPU where the processor consumes less power and dissipate less heat than in the active state [30]. They represent the percentage time a processor spends in the low-power idle state and are a subset of the processor's total idle time. In the C2 power state, the processor is able to maintain the context of the computer's caches. The C3 power state offers improved power savings and higher exit latency over the C2 state. In the C3 power state, the processor is unable to maintain the coherency of its caches. The %CPU throttle (P8) is a measure of the maximum CPU percentage to be used by any process or service, thereby ensuring that no process can consume all of the CPU's resources at the expense of other users or processes. The CPU operation was measured at every 5<sup>th</sup> second, and temperatures and fan speed were measured at every 30<sup>th</sup> second. These parameters are referred to as P1 through P8 in the following discussion.

The experiment was designed to replicate the real-time usage of computers. The computers were exposed to six environmental conditions, as shown in Table 2. For each temperature-humidity combination, four usage conditions and three power supply conditions were considered. A set of user activities was defined to execute different usage conditions of the computers. Details on the experimental setup and training data collection can be found in the Chapter 2. In total, 72 experiments were conducted. The same usage conditions were applied simultaneously to all computers in order to achieve time synchronization between computer and software application responses. The computer's power mode was always set to ON. The screen saver and hibernation options were disabled to prevent these functions from occurring during the experiment.

The Mahalanobis distance (MD) values were obtained for the experimental (i.e., training) datasets. These values were used to determine a threshold MD value (=5.8) for a healthy system [107]. The threshold was used to compare the MD values of test systems to determine the health state or presence of faults. Once the presence of a fault was established, the next objective was to isolate the parameter that was responsible for the fault in the computer.

For this study, since eight parameters were being analyzed, nine analytical runs (Table 16) were defined. The first analytical run included all eight parameters for MD calculation. Subsequently, eight (A2 to A8) more analytical runs were defined by eliminating one parameter at a time. Each cell entry in Table 2 represents a parameter's status in the MD calculation: "1" means that a parameter was included, and "0" means that a parameter was excluded from the MD calculation. Subtracting

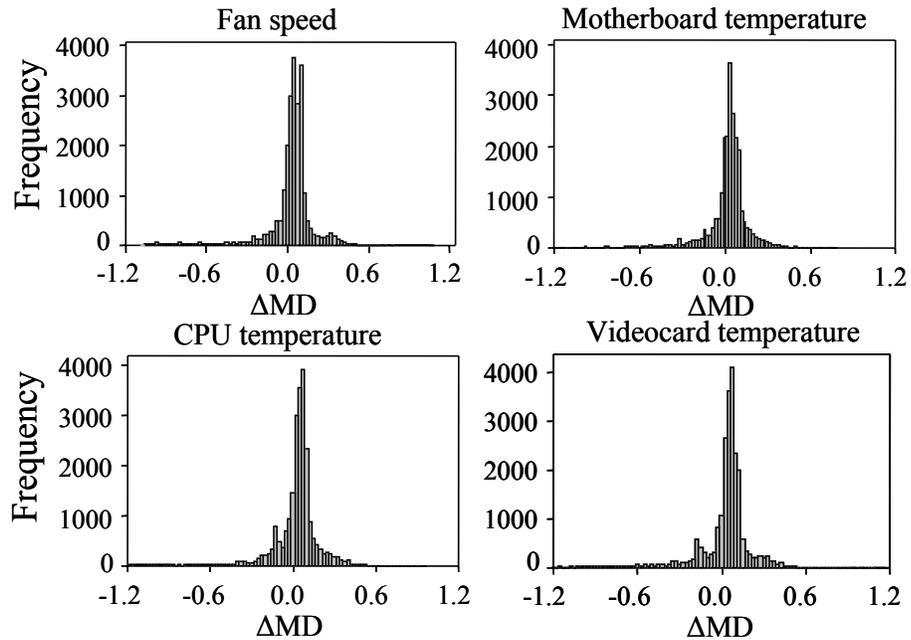
the first analytical run's MD from other analytical experiments' MD values gave the quantitative measure ( $\Delta MD_i = MD_{ia} - MD_p$ ) of a parameter's contribution to MD value. For each observation, the  $\Delta MD$ s resulting from the absence of different parameters are shown in Figure 33. For every run, the MD corresponding to each observation was calculated, which gave the  $\Delta MD$  for the parameters. From the training data, the  $\Delta MD$  for each observation was calculated and sample distributions of the  $\Delta MD$ s for four parameters are shown in Figure 34. The  $\Delta MD$ s followed Normal distribution, and a 95% confidence range for  $\Delta MD$  was estimated to define a threshold boundary (as shown in Figure 30) for each parameter in order to isolate the faulty parameters [107]. A parameter that had the highest negative  $\Delta MD$  was identified as the faulty parameter (Figure 32) in each observation. A parameter that was identified as the highest number of times as being faulty was isolated as the faulty parameter.



**Figure 33:  $\Delta AMD$  values for one observation of training data**

**Table 16: Orthogonal Design of Experiment**

Analysis	Performance Parameters							
	P1	P2	P3	P4	P5	P6	P7	P8
A1	1	1	1	1	1	1	1	1
A2	0	1	1	1	1	1	1	1
A3	1	0	1	1	1	1	1	1
A4	1	1	0	1	1	1	1	1
A5	1	1	1	0	1	1	1	1
A6	1	1	1	1	0	1	1	1
A7	1	1	1	1	1	0	1	1
A8	1	1	1	1	1	1	0	1
A9	1	1	1	1	1	1	1	0

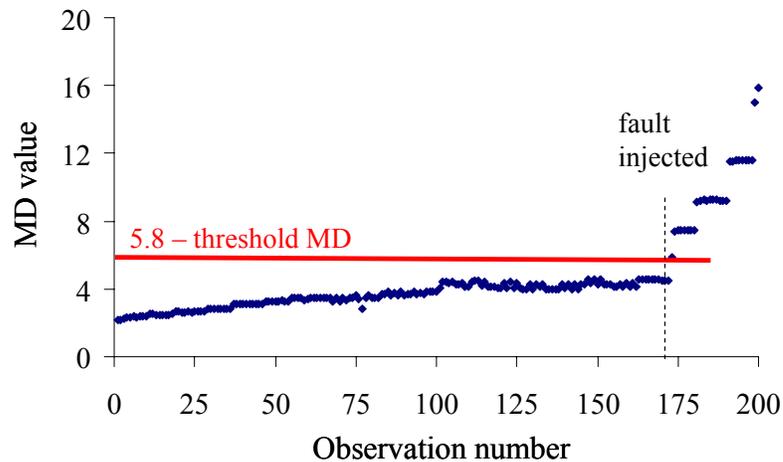


**Figure 34:  $\Delta\text{AMD}$  distribution of four different parameters**

In a lab setup controlled experiments were conducted where two known faults were injected into the computer and monitored data were analyzed by the approach discussed earlier to verify that the proposed methodology was able to isolate parameters related to these faults. Data from one field returned computer was analyzed as well to verify the approach. In the first controlled experiment, the videocard was externally heated using a kapton flex heater (1in × 1in placed on top of the videocard chip). In the second experiment, a gap was created between the CPU and the heat pipe used to transfer heat from the CPU to the heat exchanger. These two scenarios were created to simulate problems with the thermal interface material used for the CPU and videocard. In these two cases, one would expect the proposed approach to detect videocard temperature and CPU temperature parameters as faulty. In all these cases, more than 95% of the test data's MD values were higher than the threshold MD (=5.8), which indicated that the test computer had an anomaly.

The experimental procedure of externally heating the videocard was as follows: computer was turned ON for a while to stabilize its performance, and then the heater was powered ON to heat the videocard. As the computer was turned ON, its components started heating up gradually but stabilized in a while and this behavior was reflected in MD values as well (Figure 35). The MD values increased up to the 100<sup>th</sup> observation and then it stabilized by the 170<sup>th</sup> observation. Thereafter the external heater was powered ON and the MD values started increasing. The first objective was to check that the anomaly detection methodology identified the problem at the system level when the heater was turned ON. Then the parameter that was related to this anomaly was identified. For the experiment presented in Figure 35

an anomaly was identified at the 173<sup>rd</sup> observation and the videocard temperature parameter was isolated immediately. The experiment was repeated three times to check the repeatability of the results and the computer was turned OFF after each trial to let it cool down.



**Figure 35: MD values corresponding to a fault injection in the videocard experiment**

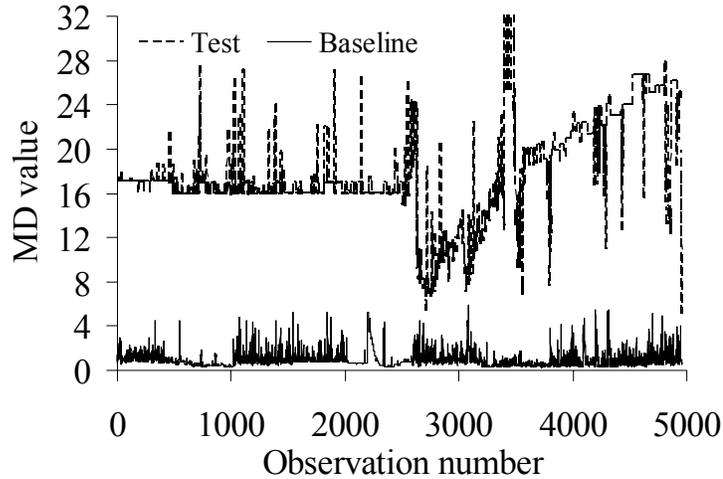
A similar procedure was followed for the second controlled experiment. In this experiment the computer was turned ON for a while to stabilize its performance parameters, and then a gap between the CPU and the heat pipe was created by unfastening the screw used to secure the heat pipe on top of the CPU. The assumption was that by doing this the thermal conductivity would be reduced and the CPU temperature would increase. For the experiment, trend in MD values was similar to Figure 35. The CPU temperature was identified as the faulty parameter in this case. The experiment was repeated three times to check for repeatability. In each experiment the computer's performance parameter information was collected for 5 minutes after turning the heater ON and after creating the gap between the CPU and

the heat pipe. The percentage observation indicating faulty parameters is presented in Table 17. The percentage observation was computed as the ratio of the number of observations indicating faulty parameters to the total observations considered during a test.

To check the validity of the proposed approach, five thousand data points from a field-returned computer were analyzed. Figure 36 shows the MDs of healthy computers (lower plot) and the test computer (above plot). The MD plot indicates that the computer had a problem right from the beginning of the period during which data was collected. A drop in MD value at around the 2700<sup>th</sup> observation was due to the rebooting of the computer, and the MD value gradually increased again. The proposed approach identified the presence of a fault and identified two parameters—fan speed and videocard temperature—as the faulty parameters since these two parameters had a higher percentage of observations (73% and 24%, respectively) indicating them as faulty parameters (Table 17). Of these two parameters, fan speed was more likely to be faulty because if the videocard temperature was increasing the fan should have been operating more, and the correlation ( $\approx 0.8$ ) between these two parameters also supports that. Based on the system knowledge the fan was adjudged as faulty and it was verified with the project sponsor.

Throughout the experiments, the computer was functionally fine even with the presence of a fault. In the experiment on the videocard, videocard temperature was identified as the faulty parameter with 95% confidence. In this case, CPU temperature also rose and was detected as faulty. However, the percentage difference between these two parameters of being faulty was significantly, and the videocard temperature

was adjudged as the faulty parameter. In the other experiment where a gap between the CPU and the heat pipe was created, the CPU temperature parameter was identified as faulty with 100% confidence.



**Figure 36: Comparison of MDs of field-returned computer with baseline**

In the analysis of data from the field-returned computer, 73% of the observations indicated fan speed as the faulty parameter, 24% of the observations indicated videocard temperature as the faulty parameter, and 3% of the observations indicated CPU temperature as the faulty parameter.

A fan failure could raise the temperature of the CPU, motherboard, and videocard. The reason for the identification of some temperature parameters as faulty lies in the cooling mechanism used for the computer, the correlation of temperature parameters with ambient conditions, and percentage CPU usage. The CPU has its own thermal management schemes and controls its temperature rise to protect the chip, whereas the videocard and the motherboard do not have such thermal management. The motherboard temperature is highly correlated ( $\approx 0.96$ ) to ambient

temperature. If the computer had been used in room temperature, the motherboard temperature would not have changed significantly and would not have been identified as faulty due to the presence of other temperature parameters. Another reason that

**Table 17: Percentage of Observations Indicating Faulty Parameters**

Experiments Parameters	Videocard heating			Gap between CPU and the heat pipe			Field- returned computer
	Trial 1	Trial 2	Trial 3	Trial 1	Trial 2	Trial 3	
P1 - Fan speed	0	0	0	0	0	0	73
P2 - CPU temperature	6	4	4	100	100	100	3
P3 - Motherboard temperature	0	0	0	0	0	0	0
P4 - Videocard temperature	94	96	96	0	0	0	24
P5 - %C2 state	0	0	0	0	0	0	0
P6 - %C3 state	0	0	0	0	0	0	0
P7 - %CPU usage	0	0	0	0	0	0	0
P8 - %CPU throttle	0	0	0	0	0	0	0

contributed to identifying the videocard temperature parameter as faulty was the restart of the computer at the 2,700<sup>th</sup> data point (Figure 36). Here, a drop in MD value was observed because the computer shutdown by itself and was then restarted manually. This resulted in temporary fan start-up and a drop in three temperatures. However, these temperatures were above the nominal/expected value of the CPU and videocard temperatures during that period. Therefore, these temperatures were identified as anomalous. Further investigation of the data set also indicated that the fan was not functioning well, and it was verified by the company that the fan was not working properly. Therefore, it can be affirmed that this new approach did identify the faulty parameter of the computer.

### 5.3 Summary and Conclusions

The chapter presented a new approach for isolating parameters that indicate system faults. The approach expands the applicability of Mahalanobis distance (MD) from fault detection to fault isolation. The main advantage of the approach, over other approaches, is that it includes all the parameters for MD calculation (i.e., no information loss due to dimensionality reduction), does not require data from unhealthy systems and does not need to create a MD scale for fault isolation. Rather, a threshold bound for each parameter based on healthy training data is defined.

A set of experiments was conducted to establish the “healthy” or “normal” operation from a set of notebook computers subjected to a range of usages and environmental conditions. The MD-based faulty parameter identification method was used to identify the fault and associated parameters. Two different types of faults

were injected in order to verify the fault isolation approach presented in this chapter. In these cases, the methodology succeeded in identifying the correct faulty parameter with high confidence of 95% and more. A field-returned computer was evaluated as well, and a parameter associated with cooling component was identified as faulty 73% of times and probability of cooling component being faulty increased to 95% after considering the experts knowledge. These results were verified by manual analysis of the data files. Faults were detected within a minute of fault injection, and parameter associated with the fault was identified subsequently. The results showed that the suggested approach provides quick fault detection capability at a system level and isolates parameter associated with faults for in-situ or offline analysis.

Identification of parameters will assist in root cause analysis of the anomalies. Identification of parameter associated with fault will help in identifying failure modes, failure site, and the critical failure mechanisms acting within the product. These parameters are will be used for damage estimation through Physics-of-failure models, quantification of product's performance shift and establishment of trends in the performance shift. The results obtained from two methods will be fused to estimate accurate remaining useful life with less uncertainty.

## Chapter 6: Health Degradation Identification Approach

A considerable body of knowledge exists on prognostics and health management of mechanical and structural systems, particularly with respect to establishing precursors (such as changes in vibration signatures and variations in acoustic level due to wear) for detecting degradation [109][110]. Degradation in electronics is much more difficult to detect and analyze than degradation in most mechanical systems and structures due to the complex architecture of electronics, the interdependency of component functionality, and the lack of monitoring sensors due to the miniaturization of most electronic devices and products.

Techniques for diagnostics and prognostics in electronic products include the use of statistical models [1], physics-of-failure-based life-consumption models [27], state estimation models [111], and data-driven models [112]. These techniques can be applied at any of the six levels of electronics defined by Gu et al.[74]: 1) chip and on-chip sites; 2) parts and components that cannot be disassembled; 3) interconnects and circuit boards; 4) enclosures and chassis; 5) entire electronic product; and 6) multi-electronic products and external connections between them. Degradation identification at any of these six levels can reduce an electronic system's reliability.

Anomalies in a product may have been caused during the manufacturing process, or they may evolve over time due to wear-out. Identification of anomalies and their progression provides advance warning and precursors for failures; enables condition-based maintenance in place of scheduled maintenance; and reduces the life cycle cost of equipment by decreasing inspection costs, downtime, and inventory.

The objective of this chapter is to provide a method to detect degradation in an electronic product's performance. A multivariate distance measure, Mahalanobis distance (MD) [9][82], is used to represent product performance in a reduced dimension. MD was selected for two reasons: first, MD reduces a multivariate system to a univariate system; and second, MD is sensitive to inter-variable changes in a multivariate system [21][22].

In a situation where a product's monitored performance parameters exhibit trends, the derived distance measure, MD, will also show a trend. However, there is also a need to translate product performance parameters that do not exhibit trends and are correlated to the product's health. Due to these two constraints, MD values alone cannot be directly used as an indicator of health. A novel method is proposed that defines a new metric, a health indicator that assesses the changes in a histogram of MD values over time. This health indicator can be used to reliably detect degradation and diagnose faults.

A histogram is a graphical representation of a frequency distribution in which the height of the bins represents the observed frequencies. The choice of bin-width primarily controls the representation of the actual data. Our health indicator is defined using the histogram bins' fractional contributions (FC) and a weight assigned to the bins. Any change in a bin's FC results in a change in the health indicator, which is used to distinguish unhealthy products from healthy ones.

The binning process reduces noise by grouping MDs that represent similar performance. The "sudden" presence of an MD value in a bin that is highly weighted would indicate an abrupt change in a system's performance. The tracking of a bin's

FC is useful in detecting no-fault-found (NFF) types of faults (faults that suddenly appear and then disappear, often associated with intermittent failures) and also in identifying trends.

The following section describes the degradation identification methodology. The section includes details on the estimation of MD, optimal bin-width for constructing a histogram, health indication, and a threshold for degradation detection. A case study is then presented to demonstrate the capability of the proposed methodology to perform real-time product monitoring. Ten computers were tested in a set of environmental and operational conditions. Eight performance parameters were monitored. The collected data was used to define a healthy baseline and threshold for the computers. A computer with an artificially injected fault was also monitored. The methodology was validated by testing the data with the injected fault against the healthy data.

## 6.1 Degradation Identification Methodology

The physical degradation of a product can result in shifts in performance parameters and can be analyzed by monitoring these parameters. MD provides an opportunity to unite performance parameters' data and their correlated interactions. A product's health over time is then represented by an MD time-series. To determine the health of any other test product, MDs for the test product are calculated after the performance parameters are standardized using the mean and standard deviation of the baseline data. The resulting MDs from the test product data are compared with a threshold MD of the healthy product to determine the test product's health. A time series of MDs can be formed by associating time information with the data, which

enables visualization of the test product's health progression over time.

### 6.1.1 Bin-width Estimation Technique

The distribution of data can be represented by parametric models or non-parametric methods. Standard parametric models include normal, lognormal, exponential, and Weibull distribution models, which are described by parameters such as the mean and standard deviation and are good for modal data. For multi-modal data, fitting a standard parametric function can result in errors, and fitting multiple parametric functions cannot provide a unique solution. Various non-parametric methods such as histograms, kernels, orthogonal series estimation, and the nearest neighbor method are therefore used to estimate the density function without assuming any parametric structure [39][113].

In this chapter, a histogram approach is used to represent the frequency distribution in which the height of the bins represents the occurrences of the observed values. The choice of bin-width primarily controls the representation of the actual data. Smaller bin-widths may provide excessive information (under-smoothing), but larger bin-widths may provide too little information (over-smoothing of the true distribution). Once the bin width is calculated, the histogram is defined. Histograms are based on an equally spaced bin-width,  $h_n^*$ , where  $n$  denotes the sample size. For the normal distribution, the approximate optimal bin-width is

$$h_n^* = 3.49sn^{-1/3} \quad (17)$$

where  $s$  is sample standard deviation and  $n$  is the sample size.

In cases where data do not follow Gaussian distribution and are multimodal in nature, a non-parametric density estimation technique can be used to determine the

underlying distribution. Kernel density estimates are well-suited for these types of data. Commonly used kernel functions include uniform, Gaussian, triangle, Epanechnikov, and bi-weight. A kernel function is generally chosen based on the ease of computation. For example, the Gaussian kernel is used for its continuity and differentiability. The optimal bin width for the Gaussian kernel is obtained by

$$h^{opt} = 1.06\hat{s}n^{-1/5} \quad (18)$$

where  $\hat{s}$  is the estimate of standard deviation. The Gaussian kernel can be used for non-Gaussian data, since the kernel is used only as a local weighting function [39]. Kernel functions are also called window functions, and Equation (18) is used for optimal bin width estimation of a histogram.

To construct a histogram, a sufficient amount of data is needed in a time window to avoid either under-smoothing or over-smoothing the distribution. A window size is chosen based on the sampling rate of observations, the sufficiency in representing the changes in a system, and the time allowed for detecting degradation. A very small window size is dominated by recent observations, while a large window size is dominated by aged observations.

### 6.1.2 Health Degradation Detection Procedure

A sequential flowchart of the degradation detection methodology is shown in Figure 37. The process involves the construction of a healthy baseline using a healthy product's performance parameters (i.e., the data collected should represent the product's performance under different environmental and operational conditions).

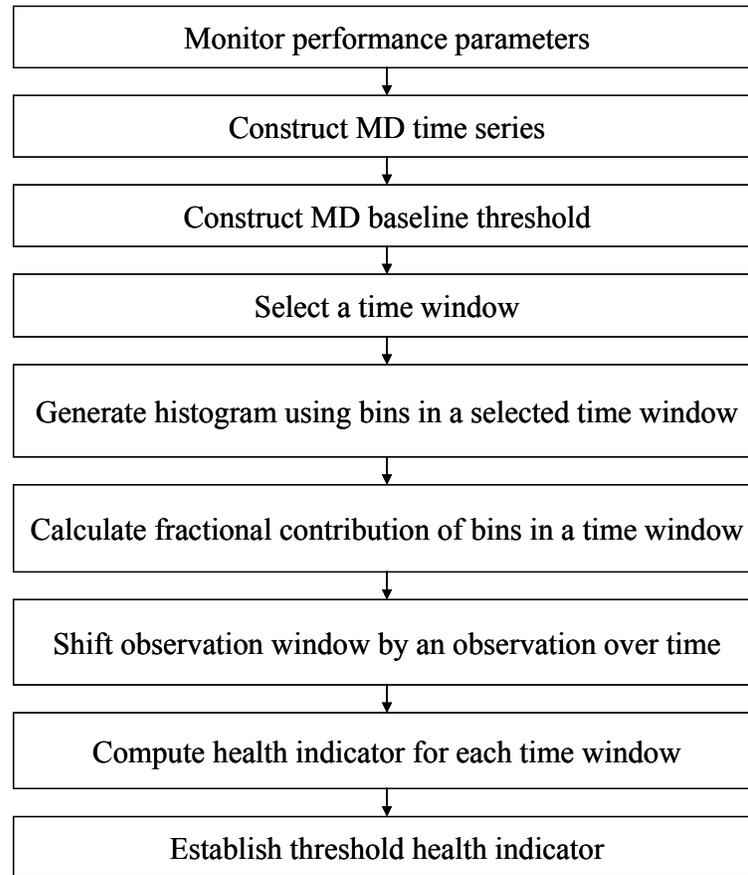
The product's performance parameter observations are transformed into a distance measure, the MD. The MD value is always positive and increases when a

product deviates from its normal healthy state. Thus, a smaller distance value indicates better health for a product. A time-series of MDs is obtained by associating time information with each distance calculation. MDs corresponding to all baseline data are estimated, which forms the Mahalanobis space. A baseline threshold that corresponds to the 95<sup>th</sup> percentile is determined from the healthy set of MDs to distinguish faulty or unhealthy products. The 95<sup>th</sup> percentile represents the confidence level in correctly classifying health. As found, the distribution of MD has a long right tail, and increasing percentiles may include extreme MD values that are due to some intermittent event or error in measurement. Before deciding the percentile to be used, one must examine data distribution.

A time window is chosen and a set of MDs, which reflects a variety of observations in that time window, is used to construct an MD histogram using Equation (18). Data points in lower bins represent a healthy product, while data in higher bins represent a product's deviation from a healthy state, i.e., a degraded state. The histogram approach reduces noise (introduced by measurement error and calculation round off) by grouping MDs that correspond to similar health states [113].

The distribution of MDs can be represented by a histogram after estimating optimal bin width from the MD's dispersion for a healthy product and the number of observations in a time window. The histogram bins will span from zero to the threshold MD value plus one extra bin. The extra bin is used to accommodate any MD value that is greater than the threshold MD value. The use of a threshold MD value takes care of extreme changes in the amplitude of MD time series signals due to intermittent events.

In the case of product degradation, performance parameters could either rise or fall relative to the values obtained for the parameters in the product's normal (i.e., healthy) condition. In either case, the MD increases. As time progresses, the product degrades, the contributions from the lower bins decrease, and the contributions from the higher bins increase.



**Figure 37: Degradation methodology using non-parametric method**

The fractional contribution (FC), which is defined as the number of observations in a bin divided by the total number of observations, is then calculated for each bin. A moving time window approach, wherein a fixed time interval moves from left to right, is used to examine a product's health progress with time, and the

fractional contribution of each bin is estimated as the time window moves forward [114]. Any change in the product's health will be reflected in the change of each bin's contribution in a successive time window. For each time window, a health indicator is estimated. A threshold health indicator corresponding to the 95<sup>th</sup> percentile is determined from the healthy set of health indicators to identify the degraded state of a product's health.

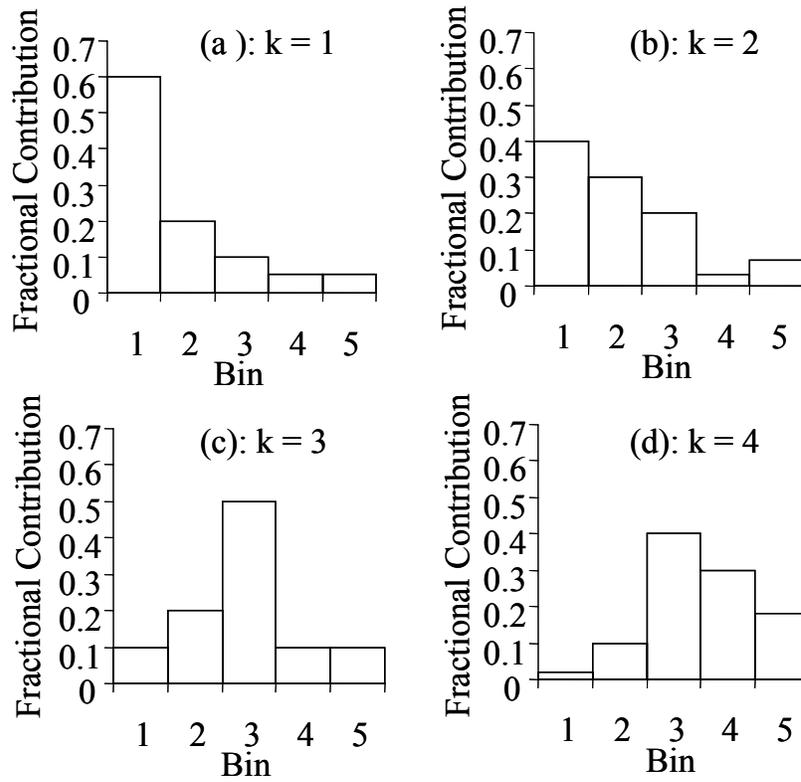
The health indicator is based on the weighted sum of each bin's contribution, and a weight factor is assigned to each bin. Sequentially higher weight factors are assigned to bins that contain higher MD values compared to bins that contain lower MD values, where histogram bins are arranged in ascending order of bin range. The rationale behind using increasing weight factors lies in the fact that a higher MD indicates a deviation in a system's performance that is larger than what was expected. Events that could occur momentarily are weighted higher, because such events represent more risk of a product's failure and less response time. Assigning equal weight would dilute the impact of events that caused the higher MDs, and may cause them to go those events unnoticed. Assigning weight in reverse order would work as well, where decision-making criteria would be a health indicator value that is greater than the threshold value of the health indicator. Assigning weight in a random manner would also give a health indicator value, but it would not provide any logical insight into the system's health or other events. In summary, assigning sequentially higher or lower weights to the bins on the right side enhances the ability to detect degradation. In this study, weights were assigned in increasing order.

An overall health indicator at each instance of time is defined by the weighted

sum of the fractional contributions of each bin in a time window. The health indicator ( $H_k$ ) at time  $k$  is estimated by

$$H_k = \sum_{l=1}^b (w_l x f_{c_{kl}}) \quad (19)$$

where  $k$  represents time,  $l=1, \dots, b$ ;  $b$  is the total number of bins;  $w_l$  represents the weight assigned to a bin; and  $f_{c_{kl}}$  is the fractional contribution of the  $l^{th}$  bin at time  $k$ .



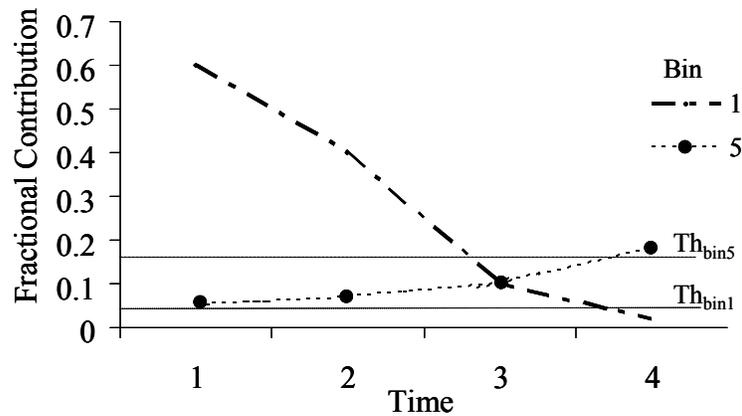
**Figure 38: Shift in fractional contribution of each bin with time**

As an illustration, a histogram of simulated data is shown in Figure 38. At each time shift, a new histogram is generated. Figure 38 also shows a sequential change in a product's health (37(a), 37(b), 37(c), and 37(d)) over time. As time progresses, the contributions from lower order bins reduces while the contributions from higher order bins increases. The fractional contribution trend of a bin can also

be utilized for a product’s health diagnosis, and the bin that has higher slopes can be used as a failure or degradation precursor. The contribution trends of bins 1 and 5 are shown in Figure 39. These trends can be used to establish a unique threshold (such as  $Th_{bin1}$  and  $Th_{bin5}$ ) for each bin’s contribution in order to track changes in a product’s particular health state for diagnostic purposes.

The threshold could be based on the 95<sup>th</sup> percentile of the bin contributions from the training data evaluation. The only limitation is that a particular bin may not come up during evaluation of the test system. Therefore, the first and last bins along with two or three in between should be considered when observing only bin trends.

The health indicator estimated for the simulated data (Figure 38) varies from  $w_l=1$  (i.e., weight of bin 1) to  $w_l=5$  (i.e., weight of bin 5). For this data, a health indicator of 1 represents a healthy system and 5 represents a degraded system (i.e., the weight of the right-most bin if all data fell into this bin,  $fc=1$ ). A basic rule is that the smaller the health indicator, the better the product’s health. In order to be 95% confident about a product’s health, a threshold value for a health indicator corresponding to the 95<sup>th</sup> percentile should be defined to detect degradation.



**Figure 39: Trend of bin’s fractional contribution over time**

**Table 18: Health Indicators for System**

Bin ( $l$ )	weight ( $w_l$ )	% contribution by bins ( $fc_{kl}$ )			
		$k = 1$	$k = 2$	$k = 3$	$k = 4$
1	1	0.6	0.4	0.1	0.02
2	2	0.2	0.3	0.2	0.1
3	3	0.1	0.2	0.5	0.4
4	4	0.05	0.03	0.1	0.3
5	5	0.05	0.07	0.1	0.18
Health Indicator ( $H_k$ )		1.75	2.07	2.9	3.52

Table 18 shows the health indicator calculation of the simulated data as time progresses and the fractional contribution of the bin changes. Degradation of a product's health can be identified as the health indicator starts to increase from 1.75 (closer to 1, healthy) to 3.52 (closer to 5, degraded).

After establishing a threshold value for the health indicator from the healthy training product, information such as the correlation matrix, mean, standard deviation, time window, bin width, and bin weight used for the training data are used to evaluate test products. The health indicator of a test product is estimated in a way that is similar to how the health indicator was evaluated for a healthy product.

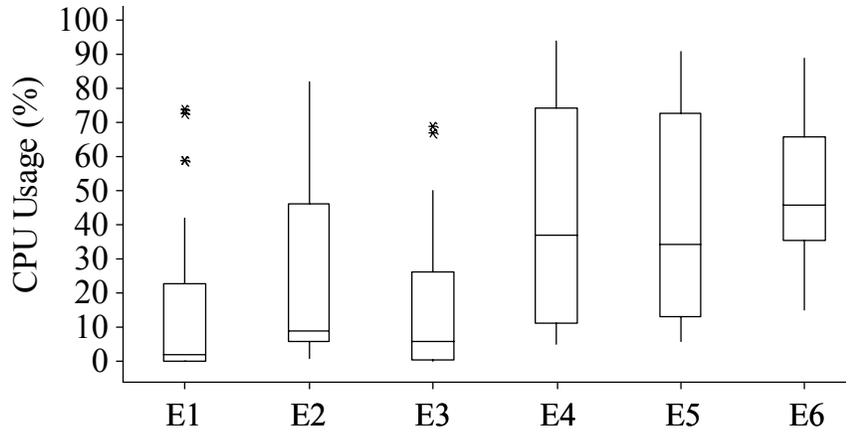
## 6.2 Case Study

A case study is presented to demonstrate the feasibility of the proposed non-parametric degradation methodology. A set of healthy computers was used to

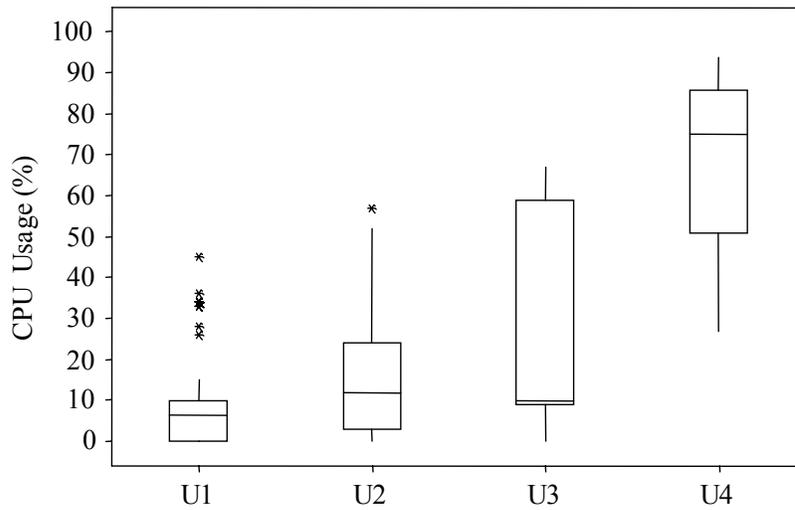
construct a healthy baseline and threshold for fault detection (i.e., threshold MD) and degradation identification (i.e., threshold health indicator). The degradation identification was verified for a computer where a fault was induced.

For constructing a healthy baseline, a set of experiments was designed to replicate the real-time usage of computers. The computers were exposed to six environmental conditions. For each temperature-humidity combination, four usage conditions and three power supply conditions were considered. A set of user activities was defined to execute four different usage conditions on the computers. Details on the experimental setup and training data collection can be found in the authors' previous publications [91]. In total, 72 experiments were conducted. The range of %CPU consumption is shown below over different environmental conditions (Figure 40) and usage conditions (Figure 41). This baseline was used to differentiate unhealthy products from healthy ones.

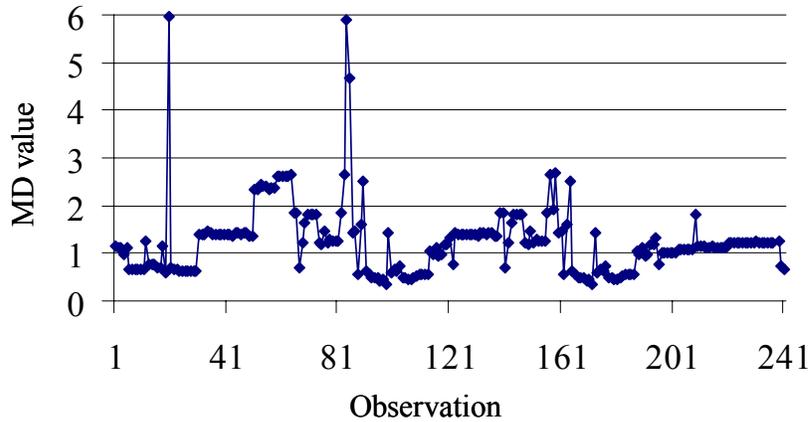
Twenty-five thousand observations were randomly chosen to represent the full spectrum of computer usage under different environmental conditions to construct the baseline. These data points were used to define the characteristic of each performance parameter. MD values corresponding to them were obtained using Equation (3) to create a baseline for the healthy computers. Variability in the MD values for the baseline is shown in Figure 42. This baseline was used to identify anomalies in the computers and detect their degradation.



**Figure 40: %CPU usage in different environmental conditions**



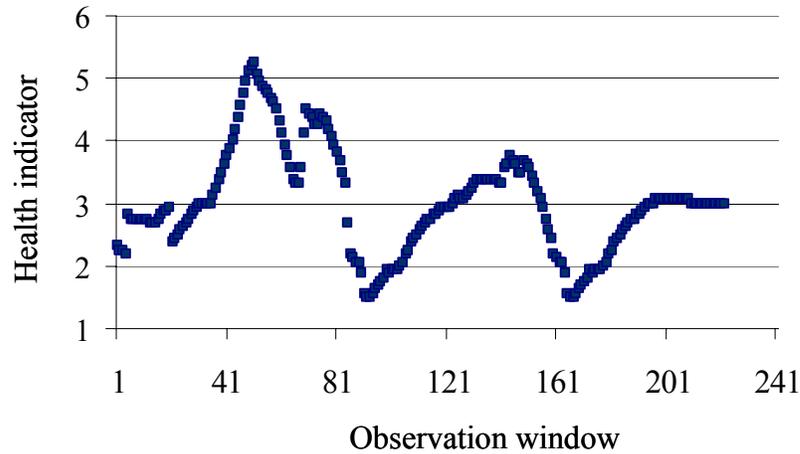
**Figure 41: %CPU usage in different usage conditions**



**Figure 42: MD values for baseline**

Since MD cannot be negative, a one-sided limit was chosen for the threshold limit. The distribution of the baseline MDs was a highly skewed distribution. Therefore, the threshold MD value was estimated from the cumulative distribution of MD values. From the baseline MDs, a threshold MD value equal to 5.8 was defined, which corresponded to the 99th percentile of MDs (i.e., 99% of MDs fell below the threshold MD).

A time window of 1 min was chosen to create a histogram and estimate the health indicator. The time window included ten instances of CPU performance measures, two instances of temperature measures, two instances of fan speed measures, and one instance of brightness and battery information. A change in temperature and fan speed was of importance because the fault was injected into the fan.



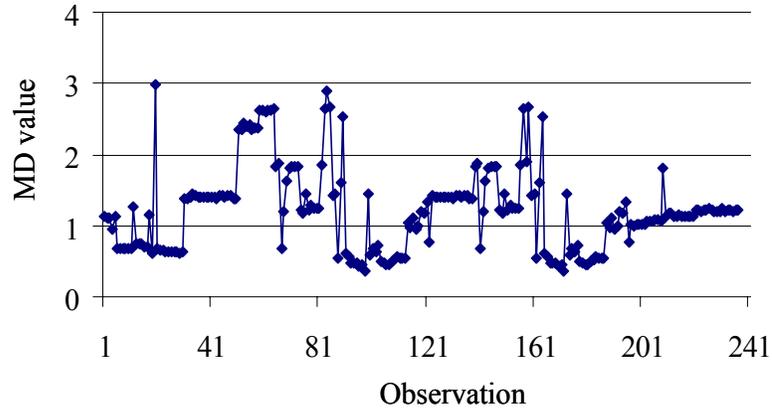
**Figure 43: Health indicator for baseline**

The standard deviation of the baseline MDs was used to estimate optimal bin width of the histogram using Equation (6). By considering 16 observations per minute, the optimal bin width was estimated (= 0.50 MD). Based on the threshold MD value and the optimal bin width, twelve bins were created, which included a bin that corresponded to the threshold MD value and anything greater than that. In the time window, the contribution of each bin was calculated and the health indicator was estimated. The health indicator plot for the baseline is shown in Figure 43. A threshold health indicator equal to 7.05 was defined for degradation identification, which corresponded to the 99th percentile of the health indicator.

### 6.3 Fault and Degradation Detection

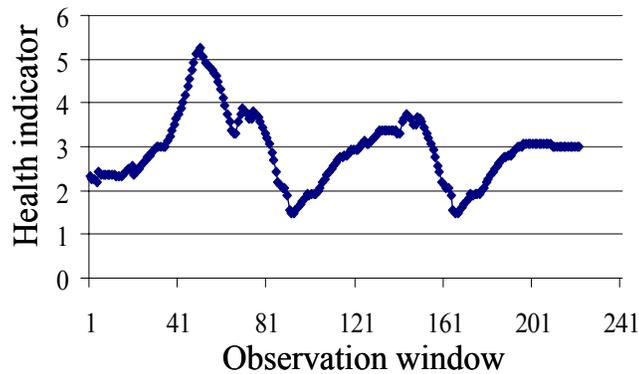
The methodology was verified with a test computer. The test computer model was the same model as the computers used for baseline construction. The test computer was powered by an AC adapter and subjected to the U1 condition in a room environment (approximately 25°C/55% RH) for observation of its healthy state. MD

values corresponding to the healthy condition of the test computer are shown in Figure 44.



**Figure 44: Sample MD values for test system (healthy)**

The health indicator for the healthy test computer was calculated for each time window using the number of bins and bin weight determined for the baseline. The health indicator plot of the healthy test computer is shown in Figure 45.

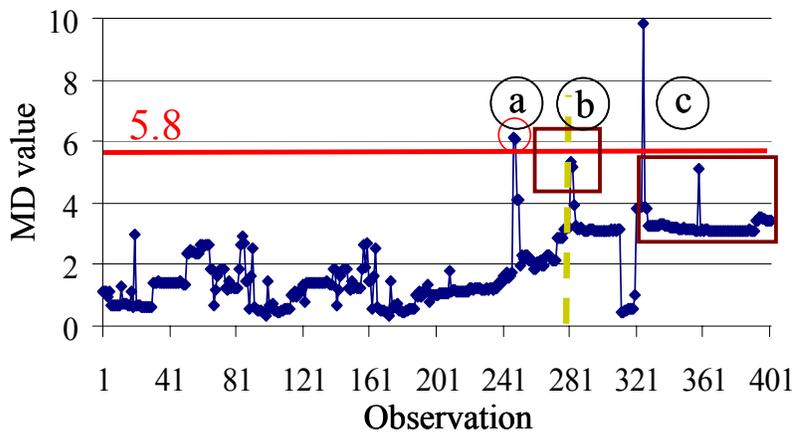


**Figure 45: Sample health indicator for healthy test computer**

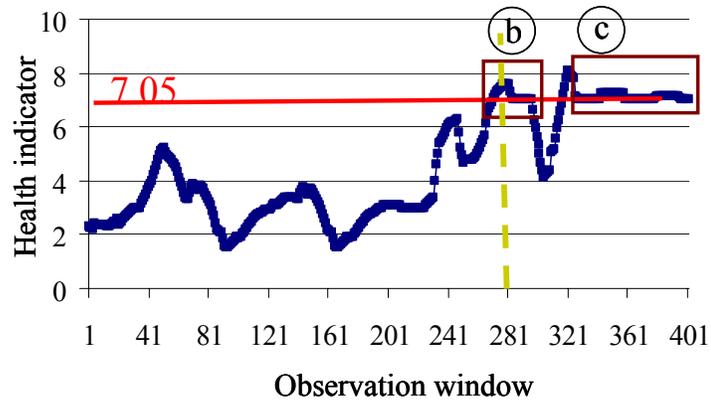
The presence of a few jumps in the MD values in the MD plots (Figure 42 and Figure 44) indicated that the test computer had some variability, but this did not necessarily indicate faults in the test computer. These jumps were due to the fact that

MD is sensitive to changes in correlation of performance parameters, and not all parameters were refreshed at the same time. As a result, the measurement of a few parameters was carried over to the next sweep of observation, and the sudden change in a particular parameter resulted in jumps in MD. With background knowledge of the data collection technique, a few jumps in MD value are not of concern, but a change in the trend of MD values is.

After turning on the test computer in the same experimental condition, a fault in the fan was induced after a short time period (242nd observation). The MD values of the test computer after fault injection are shown in Figure 46. The health indicator for each time window was calculated by constructing histograms from the MD values of the unhealthy test computer, where the number of bins and the bins' weights that were used were determined earlier. The health indicator plot is shown in Figure 47, and the difference in MD and the health indicator could be observed beyond the 321<sup>st</sup> observation.



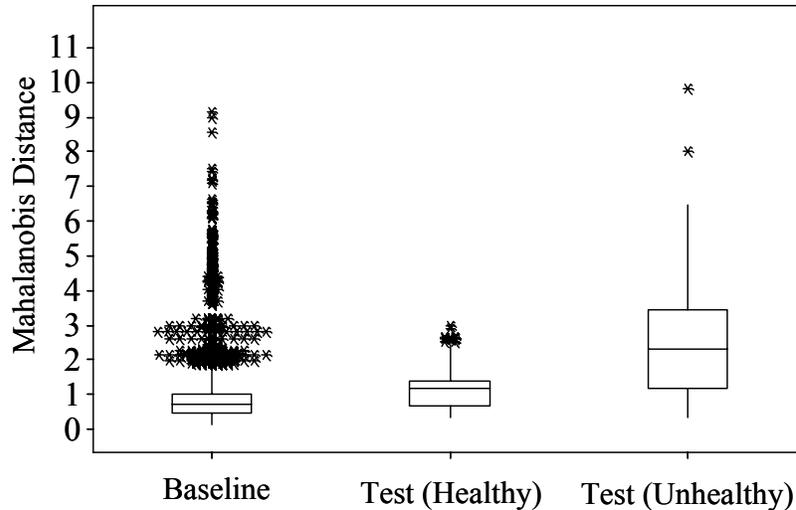
**Figure 46: Sample MD value for test system (unhealthy)**



**Figure 47: Health indicator for unhealthy test computer**

## 6.4 Results and Discussion

The statistics of MDs for the baseline, the test computer in a healthy state, and the test computer in a degraded state are given in Figure 48. Outliers in Figure 48 are represented by “\*”. The minimum and mean of MD changed by an equal amount ( $\sim 0.2$ ), and a similar inter-quartile range suggested that the test computer and the computers used for the baseline were similar but had some inherent differences either due to manufacturing or to the components used. After injecting a fault, the MD mean increased by 50%, and the inter-quartile range also increased by 100% in comparison to the healthy state of the test computer.



**Figure 48: Statistical representation using box plot of MDs**

As the test computer was turned on, various temperature parameters started to increase and MD values increased simultaneously (Figure 44) and so the health indicator (Figure 45). A fault was injected at the 242nd observation after a few instances of the fan starting and stopping, which computer’s BIOS regulates. The fan start and stop events resulted in the fall and rise of the MD values and the health indicator, but they fell below the threshold limit and had a data plot similar to what was observed for the baseline data.

The MD value went above the threshold MD value at the 249th observation, indicated by “a” (Figure 46), which was the 7th observation after fault injection. A change in computer performance was noticed within a time window (14 observations ~ 1 min). The successive MD values were higher in comparison to the healthy test computer but fell below the threshold MD value. This can be attributed to the MD estimation procedure that estimates an MD value for each observation and does not include the immediate history of system performance. Failure to include immediate history of system performance can be viewed as positive attribute of MD estimation

approach if one is concerned with fault detection only, and is not concerned about degradation.

For detecting degradation, inclusion of the immediate history would add value, and the health indicator estimation makes use of that by considering the historical information. In our case study, the health indicator started to increase with the fault injection and continued to increase with successive observations. As the temperature parameters raised, so did the health indicator, and at the 272nd observation, “b”, (2 min after fault injection) the health indicator exceeded its threshold limit. During this period, an increasing trend in MD values with a small slope and few spikes was observed (Figure 46). These spikes were accounted for in the cumulative sum of weighted bins for health indicator estimation.

The effectiveness of the health indicator was more evident after the 330<sup>th</sup> observation, “c”, when MD values were well below the threshold MD and the health indicator was hovering near the threshold health indicator. Even small changes in MD values over time resulted in a higher health indicator. The larger MDs fell into higher bins that had higher weights assigned to them. Observations from a case study suggest that the MD values provide an indication of a sudden change, while the health indicator identifies the aftereffect of fault injection.

Analysis of the fractional contribution of bins from the healthy test computer showed that higher order bins were not present up to the 242nd observation. As a fault was injected at the 242nd observation, the higher bins (bin # 7 and up) started to show up in a fractional contribution plot, while bins 1 to 6 started to fade out. Shifts in the bins’ appearances were due to changes in the computer’s health. A trend in the

contribution of each bin can be monitored as a precursor of degradation, and ten consecutive higher MD values (i.e., more than 50% contribution coming from bins 7 and above) could also be used as criteria for decision making on degradation.

## 6.5 Summary and Conclusions

A new degradation detection methodology is presented for products for which multiple parameters indicative of performance and operation were monitored in time-series. Mahalanobis distances (MD) are utilized to transform the multi-dimensional problem into a univariate problem. A time window is selected for assessment, and a histogram of MD values is calculated for each time window by binning MD values in bins of optimal bin-widths. A new metric, a health indicator, was defined as the weighted sum of the histogram bins. Based on healthy training data for the product a threshold value for the health indicator was established. This value was compared to the health indicator values during operation in order to detect degradation.

Variation in MD values is expected due to the sensitivity of MD to the variability in individual parameters. A few incidences of MD being greater than a threshold MD value do not always mean that a product is unhealthy or degraded. The variability in the MD values was neutralized by grouping them into a histogram. The health indicator utilizes immediate history to account for recent changes in a system's performance. Over time, combining information on recent history into a health indicator reduces the number of false alarms, especially when performance parameters do not exhibit trends.

The case study on computers demonstrated the usefulness of the degradation detection methodology in detecting anomalies and degradation. In this study a

specific health indicator was created using a histogram created from MDs and the weight assigned to each histogram bin. The MDs were estimated from observations collected in a time window and the size of the time window was decided based upon the data collection frequency and the diagnostic requirements. Weights were assigned to each histogram bin in an increasing order. The MD indicated a change in the computer's performance at the 249th observation. Within 2 minutes, at the 272nd observation, the health indicator crossed the threshold value, thus indicating degradation. The health indicator kept on indicating degradation even though MD values fell below the threshold value after the 330th observation. The case study demonstrates the robust nature of the health indicator metric for indicating degradation and the advantages of using this metric over using MD values directly.

The threshold value for the health indicators of a product is determined from the product's healthy baseline data. This value is dependent on the product specifications (components, ratings, duty cycle, etc.) and the known or expected use conditions of the product. This often limits the applicability of one set of threshold value across different products or product families. For example, computer products made for the gaming market would have different threshold values compared to computers for regular office use. Manufacturers can develop procedures to characterize product threshold values during the product development cycle. These values can then be updated with incremental changes in product design and features.

The present approach extends the applicability of Mahalanobis distance for degradation identification. The method can be extended to any system whose performance can be monitored under various life cycle conditions. The user's risk

acceptance criteria would define the time window size and the assignment of weights to different bins. In the decision making process, assignment of a higher weight to larger MD values reduces the smoothing nature of the weighted sum and provides intuitive meaning to numbers. The use of health indicators would reduce false negative and false positive types of errors in comparison to the use of MDs. Sensitivity analysis of health indicators will be presented in further publications.

## Chapter 7: Prognostic Measures using Symbolic Time Series

### Analysis

The real-time health assessment of electronics has great importance due to its wide range of possible applications, ranging from a battery's safety circuit to system-of-systems readiness. A system-of-systems is a complex structure composed of a large number of subsystems. A subsystem is made up of several components. All of these components of the overall system have complex interactions with each other, with feed-forward and feedback loops of instructions. Many practical systems are nonlinear and exhibit periodic, quasi-periodic, or chaotic behavior during a system's healthy operation. An incipient fault may cause changes in system behavior and translate into changes in a system's performance parameters. These changes define a system's different health conditions, providing a means to detect anomalies and estimate probable future health conditions.

Quantification of degradation and fault progression in an electronic system is difficult since not all faults necessarily lead to system failure or functionality loss [1][2]. Anomalies in a system can evolve over time or arise due to a manufacturing defect. Identification of these anomalies and their progressive trends can provide: 1) advance warning or precursors of failures; 2) condition-based maintenance instead of scheduled maintenance; 3) reduction in the life-cycle cost of equipment by decreasing inspection costs, downtime, and inventory; and 4) optimization of design and qualification cycles of new systems as well as logistical support of fielded and future

systems. Techniques for diagnostics and prognostics in electronic systems include the use of statistical models [24], physics-of-failure-based life-consumption models [115], system models [116], and data-driven models [112].

In general, a product's health can be assessed by observing multiple performance parameters of the product. Data sets of very high dimensions present an analytical challenge since all non-trivial data mining and indexing algorithms degrade exponentially with dimensionality [8]. While a high-dimensional data set contains a lot of valuable information, a lower dimensional measure is easier to comprehend and can be computed in quick succession. Considering the correlations among performance parameters is advantageous because an electronic product experiences diverse environmental and use conditions. For example, the capacitance and insulation resistance of a capacitor vary with changes in ambient temperature. The effectiveness of a diagnostic procedure increases by incorporating a change in relationship among performance parameters, because each performance parameter changes at a different rate due to changes in ambient conditions.

A multivariate Mahalanobis distance (MD) measure [9], a unified parameter representative of system health, is used to capture the non-linear dynamics of an electronics system. The use of the MD approach reduces the analytical burden because information on all the performance parameters is combined into a number (i.e., MD), which is utilized for the system's health assessment. MD does not suffer from the scaling effect because it uses normalized data. The scaling effect describes a situation where the variability of one parameter masks the variability of another parameter, and it happens when the measurement ranges or scales of two parameters

are different [10]. The use of correlation among parameters for MD calculation also makes it sensitive to small changes in performance parameters. This sensitivity is desired for a system whose operational domain is small (i.e., normal variation in performance parameters is small). For a system whose operational domain is large, MD's sensitivity results in noise for a time series analysis and needs to be reduced in order to make better decisions. The interaction of the dynamical system with its environment is another source of small variation in the parameters. In any automatic fault detection system, these types of extrinsic effects can be easily confused with the true intrinsic changes. To deal with this noisy domain data, an approach that can reduce sensitivity to small variations should be used.

Noisy information can hide the true nature of measured data and lead to the inaccurate classification of a system's health and health progression [117]. Symbolization of measurements minimizes the effects of noise and measurement error so that accurate quantification of system health becomes possible [118]. Symbolic representations of real-value measurement enable the use of algorithms and data structure from text processing and bioinformatics [119]. Symbolization of real-value measurement is achieved by representing each discrete range of measurements by a symbol. Representation of time series greatly affects the ease and efficiency of data analysis. A number of time-series' symbolic representation approaches have been introduced in the literature. The Discrete Fourier Transform (DFT) was the first approach used to reduce the dimensionality of time series [120]. The Fourier coefficients resulting from DFT do not guarantee optimal reduction. The Discrete Wavelet Transform (DWT) is used for time series sequences whose lengths are an

integral power of two [121]. However, the wavelet coefficients obtained from DWT are not generally optimal. Singular Value Decomposition (SVD) is also used for data reduction [122]. However, it is not adaptive to local variations in data.

Piecewise Aggregate Approximation (PAA) reduces time series data by dividing them into equal-sized “frames” and recording the mean value of the data points that fall within each frame [122]. Typically, the times and locations at which a parameter shows the greatest uncertainty are of most interest and are generally the most likely to yield the greatest information about system dynamics and health. The averaging of data in a frame results in the loss of sensitivity to variance. Adaptive Piecewise Constant Approximation (APCA) represents an improvement over PAA on the issue of the over-smoothing of time series data as it places a single frame in an area of low activity and many frames in areas of high activity [123]. Symbolic Aggregate Approximation (SAX) is an extension of the PAA approach [119]. A symbol is assigned to each frame of PAA where a frame for a real-time series is obtained by dividing the normalized time series into equal-sized areas. The Continuous Wavelet Transform (CWT) provides a time-frequency representation of the time series signal and uses the multi-resolution technique by which different frequencies are analyzed with different resolutions [124]. The CWT and the Shannon entropy concept are used to reduce the dimensionality of time series.

Shannon entropy is essentially a compact metric of uncertainty, and high uncertainty is taken to indicate high information content [124]. Shannon entropy offers at least two important advantages. First, it is a formal measure of information content, based on mathematical communications theory. Second, it is a relatively

comprehensive and robust estimator of uncertainty. The variance, for instance, works best for data that has Gaussian distribution. Both the variance and rank-based equivalents, such as the inter-quartile range, are poorly suited to data that has bimodal or multimodal distribution. The range has no such restrictions, but it generates absolutely no knowledge about the relative likelihood of different values lying between the minimum and maximum. Shannon entropy is free of such burdens, as it succinctly captures the informational implications of the observed frequency distribution while making no assumptions about the overall shape of that distribution.

The objective of this chapter is to develop an approach for analyzing a multivariate system and defining prognostic measures to be used for detecting anomalies and estimating probability that a system will be in a bad condition. The MD efficiently summarizes multivariate data into univariate data. Associating observation time information with MD values forms an MD time series, which can be used for system diagnostics and prognostics. To reduce noise and characterize self-similar behavior over a wide range of time scales in the MD time series, continuous wavelet transform is used [126]. The wavelet transforms represent functions that have discontinuities and sharp peaks and provide a wavelet coefficient as result of the transformation. The wavelet coefficient plot is partitioned and a symbol is assigned to each partition. The symbolic time series is generated from the MD time series and a sequence of symbols is used to define system states such that maximum information on system behavior can be revealed. A non-linear dynamic Markov model is then developed from the symbolic representation of system dynamics to differentiate the states of the system [111][126]. Symbol sequence statistics are used as the model

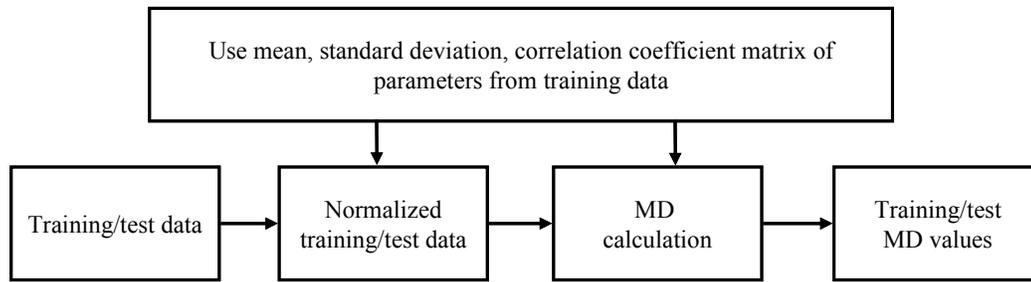
strategy, and the Markov state model is used to represent system dynamics constructed for system diagnostics and prognostics.

In the following section, details on the Mahalanobis distance method are presented regarding the transformation of multivariate data into univariate data. Following that, symbolic time series, the Markov model, and health monitoring concepts are discussed. A case study on a notebook computer is presented to demonstrate the capability of the proposed methodology in “real time” product monitoring.

#### Training and Test Data’s Mahalanobis Distance Calculation

The Mahalanobis distance methodology distinguishes multivariable data groups by a univariate distance measure that is calculated from the measurements of multiple parameters. The MD value is calculated using the normalized value of performance parameters, and their correlation coefficients, which is the reason for MD’s sensitivity [9][127].

The MD calculation process for training and test data is shown in Figure 49. The mean and standard deviation of each parameter are calculated from the training data and are used for normalizing each observation of the training and test data sets. Likewise, the correlation coefficient matrix obtained from the training data is used to calculate the MD values for both the training and test data sets.



**Figure 49: Mahalanobis distance calculation for training and test data**

At each time step, an MD value is calculated for an observation, thus forming a time series of MD values. This time series constructs a domain that represents a healthy system and represents variation in a system’s health over time. This domain can be used to identify changes in system health, and the domain can be updated as well with new system information. This time series is used to extract system features and patterns for diagnostics and prognostics. Since the MD value calculation is based on correlation coefficients among parameters, a high amount of variation arises with small changes in parameters. The small variation in parameters generates noise in the MD time series, which means it needs to be filtered. The following section discusses filtering and the conversion of this time series into a symbolic time series, which is used to develop the Markov state of the system.

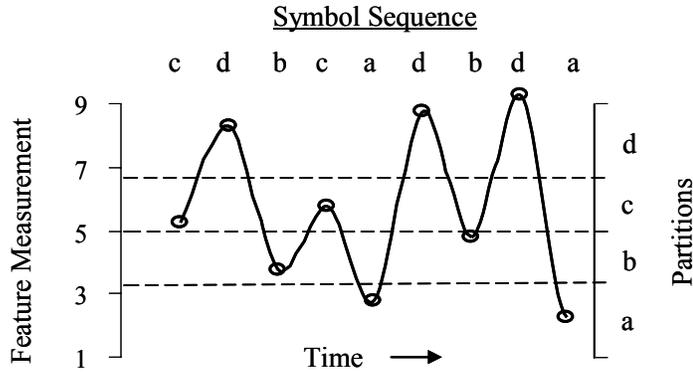
## 7.1 Symbolization of Time Series

It is generally not possible to repeat observations under the same conditions in a multivariate complex system, which has multiple components that interact with each other together with feed-forward and feedback information loops. Sometimes noisy data is observed because the measurements are irregularly sampled or exhibit non-stationary behavior. In order to study a complex system’s dynamic behavior,

capturing statistical regularities in a group of patterns, for which models are inferred from samples of the observation, is preferred over using a physics-based model, which may account for only averaged behavior of the physical process [128]. The use of statistical inferences makes time-series data generated from sensors more valuable. Several approaches to the transformation of real-value time series data into symbolic series are available [111].

The symbolic time series is generated from discretization of a real valued time-series measurement into a finite set of values [118][119][122]. The first step in this process is to define symbolic regions by introducing partitions in the real value data space, such that a specific symbol is assigned to a range of real values. This partitioning reduces the signal's variability in a region by representing it through a single symbol, which is also known as the coarsening of information. This process also improves the signal-to-noise ratio. The partitions are defined such that the occurrence of each symbol representative of a partition has the same probability.

It is observed that a small change in a system's performance parameters results in different MD values. It is assumed that a small range of MD values represents similar conditions in a system. The variability within that range is considered noise. In this chapter, an MD time series is converted into a symbolic time series. This conversion process captures large-scale features and reduces noise. A representative partitioning of a system's measured features, a real value time series, and a set of symbols used for its symbolizing (i.e., conversion of continuous measurement into symbols) is shown in Figure 1. Details on the feature-space partitioning are presented in a later section.



**Figure 50: Data series conversion into a symbol series**

A system’s behavior (or states) can be represented by a number of symbol sequences (i.e., combination of symbols such as “abc”, “acb”). A sequence is referred to as a word. The relative frequencies of words are used to construct a symbol sequence histogram (SSH). The SSH represents the overall dynamics embedded in a given time series and can be used to compare data sets or system behavior. SSH also provides information regarding “forbidden symbol sequences,” i.e., symbol sequences with zero or a significantly low probability of occurrence. With good partition, this symbolic sequence retains essentially all of the information contained in the original time series, and once coded, the dynamics of a system are indistinguishable from other information sets. This optimal symbol sequence length is achieved by maximizing normalized Shannon entropy. Shannon Entropy,  $H$ , is calculated using Equation (20)

$$H = -\frac{1}{\log_2 N} \sum_i p_i \log_2(p_i) \quad (20)$$

with  $0 \cdot \log_2(0) = 0$  and where  $N$  is the total number of observed symbol sequences with length  $l$  (i.e., the number of sequences of length  $l$  with non-zero probability);  $i$  is

an index for the sequences of length  $l$ ; and  $p_i$  is the probability of the  $i^{\text{th}}$  symbol sequence. The choice of  $\log = \log_2$  reflects the fact that the information is measured in bits.

The symbol sequence is used to represent any possible variation over time, depending on the number of symbols used and the sequence lengths. This approach does not make any assumption about the nature of the patterns, and it works equally well for linear and nonlinear phenomena. The symbolic sequence identifies features that do not emerge in the analysis of the original time series. For example, such features can be periodic repetitions of a symbol (interpreted as repeating episodes of structural change) or some kind of stationary behavior. Once the symbol sequence is generated, the symbol sequence statistics, reflecting the probability of occurrence of different symbol sequences are calculated. The symbol sequences' transition probabilities are computed from the sequences' traversal over time through the symbolic series. This information is sufficient to create a Markov state model that can effectively express system behavior. Details on the Markov model are presented in the following section.

## 7.2 System Modeling

Hidden Markov models (HMMs) are used to analyze both the time dependent evolution of a system and the steady state of a system [129]. HMMs have been widely applied to speech recognition [130], character recognition [131], texture analysis [132], and so on. In HMMs, typically the current state depends on the immediately preceding state but is not influenced by the immediately preceding observation. In our

approach, the current state depends on both the immediately preceding state and the immediately preceding observation because system states are defined by symbol sequences that share few symbols.

The HMM for a system is defined by various system states representative of a system's behavior (or health), state probability, and transition probability (probability of a system transitioning from one state to another) [129]. In the context of Markov models the matrix of transition probabilities (or transition matrix),  $E(t_1, t_2)=[e_{ij}(t_1, t_2)]$ , is the core of any HMM. Each entry in a transition matrix shows the probability that the system, being in a state “ $i$ ” at time  $t_1$ , will be in the state “ $j$ ” at time  $t_2$ . A Markov model is represented as the four tuple  $\langle s, S, W, E \rangle$ , where  $s$  is the start state,  $S$  is the set of states,  $W$  is the set of state probabilities, and  $E$  is the set of transitions.

At regularly spaced discrete times, the system undergoes a change of state and possibly back to the same state according to a set of probabilities associated with the state. The time instances are associated with state changes as  $t = 1, 2, \dots, n$ , and denote the actual state at time  $t$  as  $q_t$ . A full probabilistic description of the system state requires specification of the current state (at time  $t$ ), and all of its predecessor states. For the special case of a discrete first order Markov chain, the probabilistic description includes the current and predecessor state, i.e.

$$P[q_t = s_i | q_{t-1} = s_i, q_{t-2} = s_k \dots] = P[q_t = s_i | q_{t-1} = s_i] \quad (21)$$

Furthermore, consider processes in which the right-hand side of the above mentioned equation is independent of time, thereby leading to the set of state transition probabilities ‘ $e$ ’, of the form

$$e_{ij} = P[q_t = s_j | q_{t-1} = s_i], \quad 1 \leq i, j \leq N \quad (22)$$

with the state transition coefficients having the properties  $e_{ij} \geq 0$  and  $\sum e_{ij} = 1$  since they obey standard stochastic constraints.

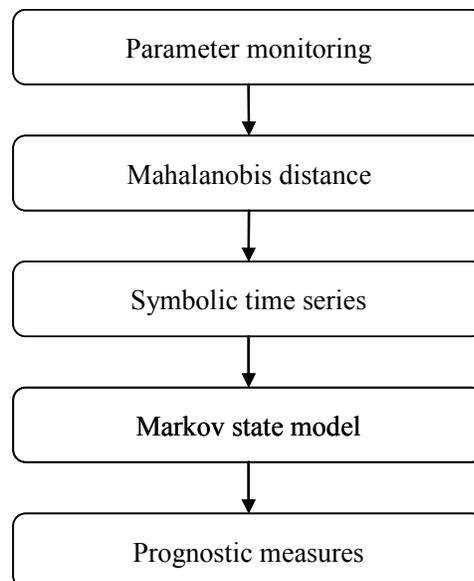
In this chapter, the Markov state model is used to determine 1) the probable system states, 2) the initial state probability, 3) the state transition probabilities, 4) average time of stay in a particular state, 5) the probability of reaching a state given its present state information, and 6) reduction in time to reach an undesired state. The following section discusses diagnostic and prognostic methodology using Mahalanobis Distance, the symbolic time series, and the Markov state model.

### 7.3 Health Monitoring Methodology

A system's health monitoring process includes the study of its functional considerations; operational and performance parameter monitoring; data feature extraction; representation of system dynamics; and identification of diagnostic and prognostic measures. These measures can be used for fault diagnosis and prognosis. It is assumed that a system under consideration has been studied for its functionality, and parameters that represent its operational environment and performance have been identified for monitoring. This chapter uses Mahalanobis distance as a data feature that is transformed into symbolic form in order to reduce noise and create Markov model for representing system dynamics and defines few measures for diagnostics and prognostics.

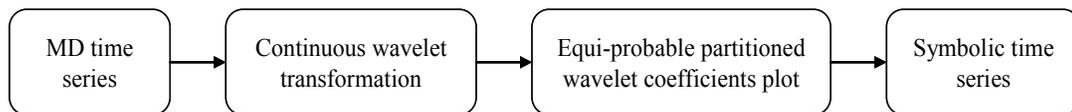
A sequential flow chart of the proposed health monitoring methodology is shown in Figure 51. The process starts with monitoring system parameters. The range

of values and the behavioral pattern of each parameter under different environmental and operational conditions are obtained. Correlation and interdependency between these parameters, also known as correlation coefficients, are calculated. These correlation coefficients and normalized parameters are used to calculate MD value for each observation, as discussed in earlier section. The MD is calculated from monitored parameters to reduce multivariate data into univariate form and by including time information with MD values. A MD time series is created. The collection of MD values for a healthy product/system is used to determine a threshold MD value, which can be used to detect any anomalies that arise over time. This provides a preliminary measure that detects sudden changes in MD values due to extreme changes in system health. The MD time series that fall within the threshold are used for further analysis. Due to inherent product variability and measurement noise, these MD values sometimes vary a lot and may provide wrong impression about product health.



**Figure 51: Schematic of diagnostic and prognostic approaches**

The next step involves creation of a symbolic time series (shown in Figure 52) from the MD time series such that noise can be reduced in the MD time series. The wavelet transform, which provides a precise and unifying framework for the analysis and characterization of a signal at different scales, is used [133]. Temporal analysis is often performed with a contracted, high-frequency version of the prototype wavelet, while frequency analysis is performed with a dilated, low-frequency version of the prototype wavelet. Since the original signal or function can be represented in terms of wavelet expansion (using coefficients in a linear combination of the wavelet functions), data operations can be performed using just the corresponding wavelet coefficients. The wavelet coefficient is the measure of the similarity of the original signal to the prototype function, called an “analyzing wavelet” or “mother wavelet”



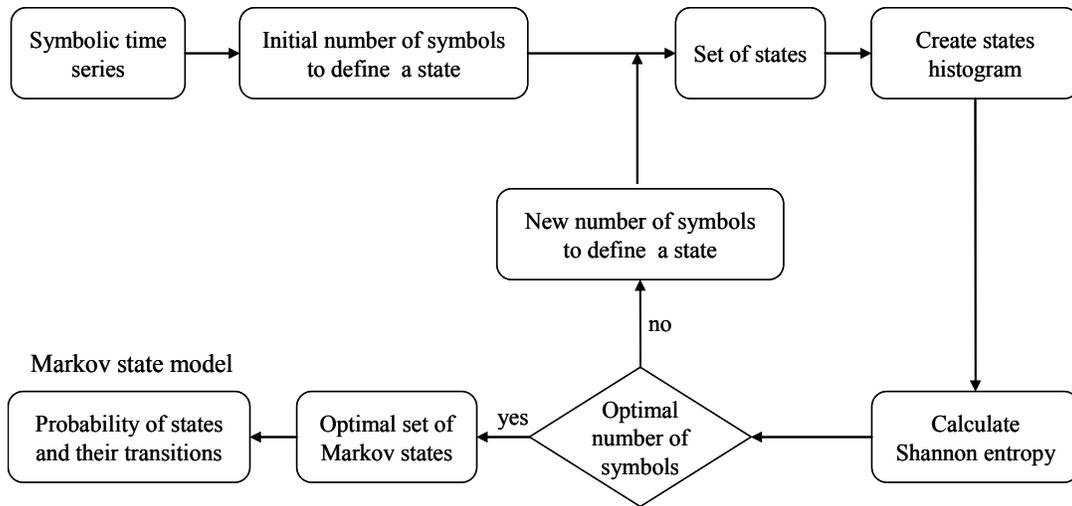
**Figure 52: Symbolic time series generation**

The wavelet transformation uses a family of wavelet functions (i.e., mother wavelet) and its associated scaling functions to decompose an original signal. In this chapter, a continuous wavelet transform is applied to MD time series to get wavelet coefficients at different scales and time shifts. The high scales correspond to a non-detailed global view (of the signal), and low scales correspond to a detailed view. Similarly, in terms of frequency, low frequencies (high scales) correspond to global information of a signal (that usually spans the entire signal), whereas high frequencies (low scales) correspond to detailed information of a hidden pattern in the signal (that usually lasts a relatively short time). The scale and coefficient at each time shift

represents product health. The higher coefficient means better similarity between the mother wavelet and the original signal. The stacked plot of a maximum coefficient, at each time, over different scales is generated. The stacked plot is partitioned such that each partition (i.e., symbol) is equi-probable (i.e., each partition has same number of coefficients). This results in segments with more information having a finer partition while segments with coarser information would have a wider partition [134]. The partitioning process is iterative, and the optimal number of partitions is influenced by the number of symbols to be used for representing a state.

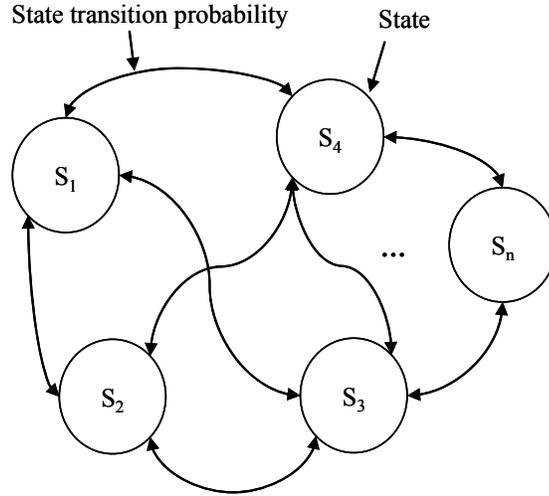
To create a Markov state model we need information on the Markov states and the transition matrix. The steps involved in determining Markov states from the symbolic sequence are shown in Figure 53. A symbol sequence of length “ $l$ ” forms a state of the system. The number of states in a Markov model is decided by the number of symbols used for a state, their possible combinations, and the total number of symbols used in partitioning. A histogram from the probability of different states is built, where the probability of each state is calculated from the symbolic time series. This state probability is used to calculate the Shannon entropy. This process is iterated to determine optimal sequence length, which is used to define Markov states. Relative improvement in the Shannon entropy over different sequence lengths is used to determine the number of symbol sequences to be used for representing a state. The number of segments on a coefficient plot is an iterative process, and the optimal number of segments is the amount that provides the minimum relative improvement in the Shannon entropy over other partitions. Minimum relative improvement ensures the minimum influence of noise on the symbolizing process. A symbol (numeric or

alphabetic) is assigned to each partitioned segment. The symbol sequence is generated using the original MD time series and a partitioned plot.



**Figure 53: Markov state determination**

The Markov state model is defined by the initial probability of each state and their transition probabilities. Initial state probability is the relative frequency of a state (i.e., symbol sequence) in a symbolic time series. Transition probability is the probability of reaching one state from another state when traversing through a symbolic time series. In addition, the distribution of times spent in a particular state is derived. A representative Markov state model is shown in Figure 54. Based on system information and data analysis, Markov states can be classified as normal, degraded, and bad. At any given instance, a system can be present in any one of the Markov states.



**Figure 54: A representative Markov state model**

The time to reach an end state, given that a system is in a particular (initial) state, can be estimated by analyzing the chain of states sequence that a system follows from the initial state to an end state [135]. A backward iterative equation 23, 24, and 25 to compute remaining time is as follows where starting state is  $i$  and some end state is  $j$ .

$$TT_{j-1} = (1 - a_{j-1,j}) * k\text{-th percentile of } d_{j-1} + a_{j-1,j-1} * k\text{-th percentile of } d_{j-1} \quad (23)$$

$$TT_{j-2} = (1 - a_{j-2,j-1}) * TT_{j-1} + a_{j-2,j-2} * k\text{-th percentile of } d_{j-2} \quad (24)$$

$$TR_i = k\text{-th percentile of } d_i + TT_i \quad (25)$$

where  $TR_i$  is the remaining time a system has at the  $i$ th state to reach a system state  $j$ ;  $TT_i$  is the time to travel from the  $i$ th state of a system;  $a_{ii}$  is the self state probability;  $a_{i,i+1}$  is state transition probability from  $i$ th to  $(i+1)$ th state of a system; and  $d_i$  is the distribution of the stay duration in state  $i$ . In case of a single training dataset, a system's expected stay in any state could be based on the percentile value (i.e.,  $k$ th percentile of  $d_i$ ) estimated from the distribution of the system's stay in that state. The

equations use probabilistic time to stay in a particular state, where the distribution of a state's stay duration ( $d_i$ ) and the state's transition probability ( $a_i$ ) are defined from the healthy training data. One can estimate remaining useful life (i.e., time to reach the end state from any state) if time-to-failure data is used to create a Markov model.

The measures that can be used for system prognosis include identification of new non-zero states, changes in state probabilities, changes in the amount of time a system stays in a state, changes in the time to reach a particular state, changes in the probability of reaching a particular state, and time to reach a particular state. These probabilistic measures are calculated from the Markov state model and the system's present state information. Knowledge of the probability density function of stay duration in any state and knowledge of the time to transition from one state to another makes it possible to estimate the amount of time a system takes to reach a particular state from its current state. This information provides lead-time information about the system that is equivalent to remaining useful life. The methodology discussed is explained using the following case study.

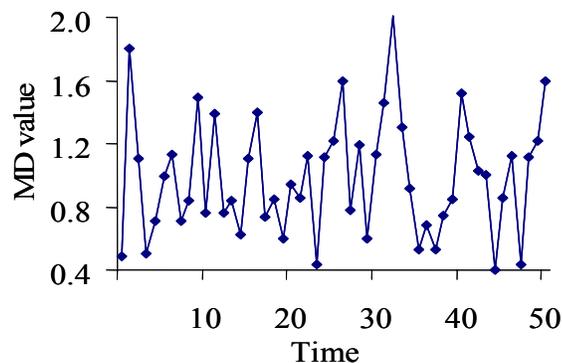
#### 7.4 Case Study

Experiments were performed on ten identical notebook computers that were manufactured by the same company. As part of the test plan, it was necessary to assess the performance of the products under different environmental and usage conditions. Since not all conditions could be tested, certain extreme and nominal conditions were included. The software usage conditions—a set of computer users' activities representative of typical computer uses—was defined [72]. These usage conditions were executed through a script file, where all user activities were encoded.

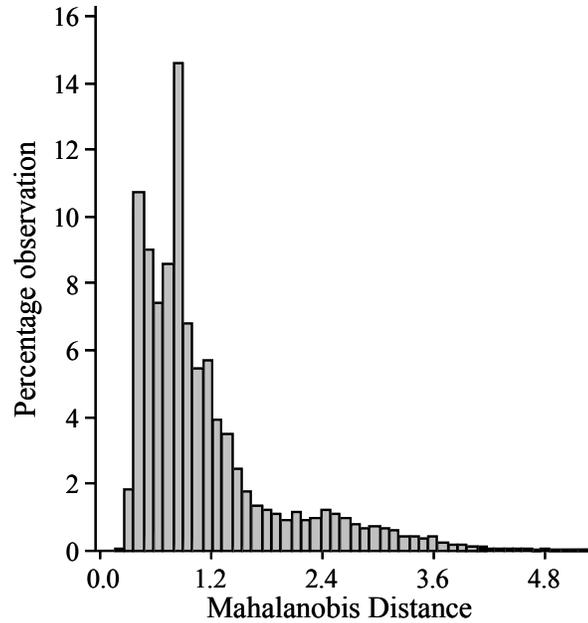
To study the variability in performance parameters, experiments on ten computers were conducted under six different environmental conditions, as shown in Table 2, and measurements were made in-situ. The test temperature range was from 5°C to 50°C, which was wider than the specified operating and storage temperature range of the computer in order to include variation in operating conditions beyond the manufacturer-specified range. In each environmental (temperature-humidity combination) condition, four usage conditions and three power supply conditions were considered [91]. The test duration depended upon the way the computer was powered. When the computer was powered by an AC adapter and the battery was fully charged (relative state of charge (RSOC) = 100%), the test ran for 3.5 hrs. When the computer was powered by an AC adapter when the battery was fully discharged (i.e., RSOC < 4%), the test duration was determined by the time the battery took to fully charge (RSOC = 100%). When the computer was powered by its battery only, the test duration was determined by the discharge (RSOC < 4%) time. The tests were conducted in a temperature-humidity chamber and in a room-ambient environment. Table 3 shows all 72 experiments. Each computer was turned on for 30 minutes before the experiment was started. The computers were kept at room temperature between each test for 30 minutes.

The training data was formed by eight correlated performance parameters. The parameters measured were fan speed (speed of a cooling fan in rpm), CPU temperature (measured on the CPU die), motherboard temperature (measured on the top surface of the printed circuit board near the CPU), videocard temperature (measured on the graphics processor unit), %CPU usage (measure of how much time

the processor spends on a user's applications and high-level Windows functions), and %CPU throttle (measure of the maximum CPU percentage to be used by any process or service, thereby ensuring that no process consumes all of the CPU's resources at the expense of other users or processes). The parameters C2 and C3 are power saving states of the CPU in which the processor consumes less power and dissipates less heat than in the active state. C2 and C3 represent the percentage time a processor spends in the low-power idle state and are a subset of the processor's total idle time. In the C2 power state, the processor is able to maintain the context of the computer's caches. The C3 power state offers improved power savings and higher exit latency over the C2 state. In a C3 power state, the processor is unable to maintain the coherency of its caches. All the parameters were sampled at different rates: CPU operation every 5 seconds, and temperatures and fan speed every 30 seconds.



**Figure 55: MD values of healthy system**



**Figure 56: Histogram of MD values for healthy population**

The training data was composed of approximately 24,012 observations. The correlation coefficients among performance parameters were calculated. The Mahalanobis distance for each observation of the training dataset was calculated using Equation (3). As discussed earlier, a small change in performance parameters results in a bigger change in MD value. For illustration purposes, the variability in training data MD values is shown in Figure 55, and MDs distribution (i.e., histogram) is shown in Figure 56. Right-skewed distribution suggests that a large amount of information on system dynamics can be grouped. However, small variations in values may contain noise that need to be filtered for better representation of system dynamics and health assessment.

The Daubechies wavelet transform was applied to the MD time series data to remove noise and extract features from the data, such as trends, discontinuities, and self-similarities. Wavelet coefficients were obtained as output of wavelet transforms

at different scales and time shifts. An average of the largest coefficients over the scales at different time shifts was calculated. A wavelet transform, db4, which had the largest change in average coefficient (Table 19), was selected for transforming the MD time series.

**Table 19: Changes in Average Coefficient**

db2	db3	db4	db5	db6	db7
0.029	0.077	0.099	0.095	0.084	0.066

A stacked plot of the largest coefficients over different scales at different time shifts was generated. The number of equi-probable partitions of coefficient plots was such that the dense region had more partitions than the sparse region. A symbol was assigned to each partition, such as 1, 2, 3, 4, 5, 6, 7, and 8, were assigned from the lowest partition to the highest partition in ascending order on the coefficient axis of the stacked plot. Table 20 presents the relative improvement in Shannon entropy corresponding to different partitions (i.e., 5, 6, 7, and 8) and word length (i.e., 2, 3, 4 and 5). Based on the minimum relative change in Shannon entropy and word length, a number of partitions were selected for the stacked coefficient plot. The relative change in Shannon entropy is defined as change in entropy divided by old entropy (i.e.,  $\Delta E/E1$ , where  $\Delta E = E1 - E2$ ).

**Table 20: Relative Change in Shannon Entropy for Different Partitions**

Word length		2	3	4	5
Number of division	8	0.159	0.223	0.202	0.170
	7	0.155	0.237	0.260	0.242
	6	0.177	0.206	0.274	0.271
	5	0.200	0.342	0.361	0.366

To create a hidden Markov model different states are defined by symbol sequences, known as a word. The optimal word length is determined by the Shannon entropy maximization. Table 21 presents the relative improvement in Shannon entropy for different word lengths. A word length of two symbols, which shows the largest improvement in the Shannon entropy for all partitions, was selected to define different Markov states. The Markov states were represented by all symbol sequences formed from the two-symbols sequence. After determining the optimal word length, the optimal number of partitions, which was seven, was identified from Table 20. The coefficient plot was segmented into seven partitions.

**Table 21: Relative Changes in Shannon Entropy for Different Word (Symbol Sequence) Length**

Partitions used		8	7	6	5
Word length	2	0.75	0.73	0.70	0.68
	3	0.63	0.64	0.63	0.60
	4	0.48	0.50	0.51	0.51

The number of Markov states increases significantly as the symbol sequence length increases and the number of symbols used in the problem increases. An increase in the number of possible states reduces the initial probability of each state significantly and does not improve the results. Therefore, the window size should be such that the number of Markov states remains manageable. In this study, the total number of symbols used was 7, a window size of 2 was chosen, and this resulted in 49 Markov states. The number of non-zero states was 39, and 10 states had never been visited. The initial state probability and transition probability estimates were calculated from the symbolic time series. Each state's initial probability, time spent in a state, and transition time from one state to another state were calculated. This information formed the baseline for comparing system health over time.

For validation purposes, data from a field-returned notebook computer, which had an issue with its fan and would stop functioning after a while, was used. The temperature of the computer's components increased as the fan speed dropped. The amount of data (i.e., 24012) used from field-returned computer was equal to the training data. The prognostic measures mentioned earlier were calculated. Deviations (Table 22) from the baseline information were indicative of changes in the computer's performance and health. The first prognostics measure to indicate a behavior change in the system was observed as a reduction in the number of non-zeros states (i.e., system visits new states); the second measure was a change in state probability; the third measure was a change in the time for which a system stays in a particular state (i.e.,  $t-t_0$ ); the fourth measure was the change in the travel time from one state to another (i.e.,  $\Delta T_{a \rightarrow b}$ ); the fifth measure was a change in the probability of

reaching from one state to the another (i.e.,  $\Delta P_{a \rightarrow b}$ ); and the sixth measure was the relative time reduction in reaching one state from another ( $\Delta t/t$ ).

**Table 22: Prognostics Measures Computed for Field-Returned Computer**

Reduction in number of non-zero states in test system	3
Change in state probability (state :77)	$p-p_0 = 0.202$
Change in time a system stays in a state (state :77)	$t-t_0 = 4460$ unit
Difference in time unit taken to reach state 77 from 11	$\Delta T_{77 \rightarrow 11} = 73$
Difference in probability of reaching a particular state 77 from 11	$\Delta P_{77 \rightarrow 11} = 0.475$
Time to reach a particular state 77 from 11	178

For prognosis, the objective was to find the probability of a product reaching a bad state and the time it takes to reach that state. The Markov states are defined by symbols, which are represent partitions of a wavelet coefficient plot. A wavelet coefficient plot is formed by plotting wavelet coefficient over different wavelet scale. A wavelet coefficient expresses the closeness of the wavelet function to the data array. Since the wavelet functions are compact, the wavelet coefficients only measure the variations of a small region around the data array. The “localized” nature of the wavelet transform allows identification of spikes in data. The localization also implies that a wavelet coefficient at one location is not affected by the coefficients at another location in the data. This makes it possible to remove “noise” from a signal simply by discarding the lowest wavelet coefficients. Selection of the largest coefficient at each observation reduces the possibility of noise in the transformed data. Out of the coefficient population, larger coefficients in comparison to smaller coefficients

represent spikes in the original data. The state defined by 11 corresponds to the smaller coefficients of the population; and 77 corresponds to the larger coefficients of the population, because the coefficient plot is partitioned from 1 to 7 in the ascending order of coefficients. For this reason, state '11' was assumed good and '77' as bad in this case study.

Next, the time that the system would take to reach the faulty state from a good state and the probability of reaching a faulty state was calculated. In this study, for the field-returned computer the time to reach a faulty state from a good state was 73 time units earlier than the time to reach state 77 from 11 in the system's healthy condition. For a healthy system the probability of these two states together  $P_{(11|77)}$  was 0.0168 where the initial probability of state 77 was 0.14. For an unhealthy system,  $P_{11}$  was 0.033, and the probability of  $P_{77}$  was 0.51, assuming that all other states probabilities had not changed (i.e., the probability of reaching a bad state increased by 0.37). When changes in all of the states' probabilities were considered, the probability to reach a faulty state from a good state increased by 0.47. This result demonstrates that the probability estimate of going into a bad state based on any state without considering other transition states gives a conservative estimate. From this one can make a conservative probabilistic estimate of a system to reach a particular state based on the system's originating state, transition probability matrix, and initial state probability information. This probabilistic estimate can be used as a prognostic measure for a system. The remaining time to failure was not been computed in this study as time to failure data was not collected. Nevertheless, from training data the time to reach a state 77 from another state 11 has been computed: 178 time units. The time to reach

state 77 from 11 was computed using Equations 23, 24, and 25. The 99th percentile value from the duration distribution ( $d_i$ ), which was defined for the system's stay in a state, was used, and for state 11 the 99<sup>th</sup> percentile of the stay duration was 91 time units. Whereas, from the training data set actual time the system took to reach state 77 from state 11 were 185 time units. The computed time to reach a particular state was less than the actual time, which indicates a conservative estimate of the time to reach a particular state. This demonstrates that the presented approach can be used for prognostics purposes, and a remaining time estimate can be made provided that the distribution of the system's stay duration at different Markov states is available.

## 7.5 Summary and Conclusions

This chapter presents a unique approach to perform prognosis of multivariate systems using Mahalanobis distance (MD). The approach extends the utility of MD from outlier classification and fault detection to prognostics. This work applies symbolic time series analysis to assess a multivariate system's health advancement over the previous work that uses symbolic time series to analyze univariate system.

First, the MD reduces a multivariate system to a univariate system by taking into account the correlation among multiple variables. Due to the sensitivity of MD to variation in performance parameters, a pattern or trend could not be observed in the MD series. The absence of trends and the presence of noise in MD is also due to the wide healthy range of the performance parameters. In the absence of a clear trend or pattern in the MD time series, a wavelet transform is applied to extract signal features without losing time information. The wavelet transformation has not been used before on MD because MD has not previously been treated as a time series.

A symbolic time series is created from the MD signal using the optimal number of symbols. The optimal number of symbols is determined by partitioning the coefficient plot. MD time series is converted into a symbolic time series to reduce noise in the MD series. Finally, the optimal number of states representing system behavior is determined. The dynamic nature of the system is modeled through a Markov state model, which can be used to compute measures for anomaly detection and prognostics. The Markov model makes it possible to perform prognostic measurements, such as identification of new non-zero states, changes in state probabilities, changes in the time a system spends in a state, changes in the time to reach a particular state, changes in the probability of reaching a particular state, and relative changes in probable time to reach a state. The Markov model allows the system to go back to their previous states, and therefore consideration of unidirectional system progress is not necessary. The Markov model represents system behavior better than any regression fit that defines system health. Anomaly measurements provide a tool for system diagnostics. Time to reach a particular state provides prognostic measure.

A case study was performed on notebook computers. The computers were subjected to different usage and environmental conditions, such that the healthy behavior of the computer was observed and data was collected and used as a training set. The computer's performance parameter variability and interdependency were examined as well. The MD time series were obtained from these data points and were represented in symbolic form. A Markov model was constructed from these symbolic representations of the training data, and baseline measures were computed. A field-

returned computer data was evaluated, and various prognostic measures were computed. The results presented in the paper indicate that these measures are good measures for anomaly detection and for estimating the probability of a system's reaching a bad state. An approach to estimating on time to reach a particular state based on the distribution of a system's stay duration at different Markov states is presented. Time to reach a particular state can also be used as a prognostic measure, and this measure can be used as an input in the logistic scheduling process.

This method is unique in the sense that it combines several measurements into a distance measure. The consideration of correlation coefficients between performance parameters enables this approach to be used to detect uncorrelated changes in parameters. The approach can be extended to any system, and it is especially valuable when the physics of failure for a system is unknown. The use of symbolic transform reduces the noise, which means data from lower resolution sensors can also be used because of the partitioning made in the coefficient plot. The Markov state model and time distribution of system's stay in a particular state allows estimation of the time to reach a particular state provided the system's current state information. This capability allows one to compute remaining useful life using a Markov model. .

## Chapter 8: Contributions and Future Work

The thesis presents a probabilistic approach to establish threshold Mahalanobis distance (MD) instead of using expert opinion-based threshold MD value to classify a product as being healthy or unhealthy. An error function is defined and minimized such that a reference MD value can be determined, which is used to identify the presence of a specific fault in a product. An approach to construct an MD control chart is presented. The MD control chart enables identification of trends and biases in a system's performance during continuous health monitoring. The ability to identify trends and biasness in the data will enable the devising of new tests to identify flawed systems and processes. The MD control chart concept can also be used by the manufacturing industry for continuous process monitoring instead of following the control charts of several performance parameters.

The thesis presents a new approach for isolating parameters that indicate system faults. The approach expands the applicability of Mahalanobis distance from fault detection to fault isolation. The main advantage of the approach over other approaches is that it does not require data from unhealthy systems and does not need to create an MD scale for fault isolation. Rather, a threshold bound for each parameter is based on healthy training data. Identification of parameters will assist in root cause analysis of the anomalies. Identification of parameters associated with a fault will help in identifying failure modes, failure sites, and the critical failure mechanisms acting within the product. These parameters can be used for damage estimation

through physics-of-failure models, quantification of the product's performance shift, and establishment of trends in the performance shift. The results obtained from the two methods can be fused to estimate accurate remaining useful life with less uncertainty.

This thesis presents a health degradation detection methodology for a system that has multiple performance parameters. A new metric, a health indicator, was defined as the weighted sum of the histogram's bars (or bins) created from MD values. A time window is selected for assessment, and a histogram of MD values is calculated for each time window by binning MD values in bins of optimal bin-widths. The health indicator utilizes the immediate history of the product to account for recent changes in a system's performance. A few incidences of MD being greater than a threshold MD value do not always mean that a product is unhealthy or degraded. The variability in the MD values is neutralized by grouping them into a histogram. Over time, combining information from recent history into a health indicator reduces the number of false alarms, especially when performance parameters do not exhibit trends. The present approach extends the applicability of Mahalanobis distance for degradation identification. The method can be extended to any system whose performance can be monitored under various life cycle conditions. The user's risk acceptance criteria would define the time window size and the assignment of weights to different bins. In the decision making process, assignment of a higher weight to larger MD values reduces the smoothing nature of the weighted sum and provides intuitive meaning to numbers.

This thesis presents an approach for using symbolic time series for analyzing a multivariate system's health and also defines several prognostic measures. The dynamic nature of the system is modeled through a Markov state model, which enables computation of prognostic measurements, such as identification of new non-zero states, changes in state probabilities, changes in the time a system spends in a state, changes in the time to reach a particular state, changes in the probability of reaching a particular state, and relative changes in probable time to reach a faulty state. The state model developed considers that systems can go back to their previous states after maintenance or changes in usage conditions, and so there is no need to consider unidirectional system progress. The state model represents system behavior better than any regression fit that defines system health, especially for electronic products that experience failure not due to wear-out mechanisms.

Future work that can be performed includes: (1) sensitivity analysis of MD with respect to performance parameters, (2) sensitivity analysis of residual MD values in the fault isolation approach, (3) health indicator trending such that health degradation can be modeled in order to estimate time before system failure, and (4) characterization of system states in terms of system health.

## Bibliography

- [1] M. Pecht, *Prognostics and Health Management of Electronics*, New York: Wiley-Interscience, 2008.
- [2] N. Vichare and M. Pecht, "Prognostics and Health Management of Electronics," *IEEE Transactions on Components and Packaging Technologies*, vol. 29, no. 1, pp. 222–229, Mar. 2006.
- [3] D. Cutter and O. Thompson, "Condition-Based Maintenance Plus Select Program Survey," Tech. Rep. LG301T6, Jan. 2005.
- [4] L. V. Kirkland, T. Pombo, K. Nelson, and F. Berghout, "Avionics Health Management: Searching for the Prognostics Grail," *IEEE Aerospace Conference*, Big Sky, MT, pp. 3448-3454, Mar. 6-13, 2004.
- [5] Department of Defense, *Performance Based Logistics: DoD 5000.2 Policy Document*. Washington, DC: Defense Acquisition Guidebook, Ch. 5.3, Dec. 2004.
- [6] K. Janasak and R. Beshears, "Diagnostics to Prognostics—A Product Availability Technology Evolution," *53rd Annual Reliability & Maintainability Symposium (RAMS)*, Orlando, FL, pp. 113–118, Jan. 22-25, 2007.
- [7] R. X. Gao and A. Suryavanshi, "BIT for Intelligent System Design and Condition Monitoring," *IEEE Transactions on Instrumentation and Measurement*, vol. 51, no. 5, pp. 1061–1067, Oct. 2002.
- [8] L. G. Abril, F. Velasco, J. Ortega, and F. Cuberos, "A New Approach to Qualitative Learning in Time Series," *Expert System with Application*, vol. 36, no. 3, pp. 9924–9927, 2009.
- [9] G. Taguchi, S. Chowdhury, and Y. Wu, *The Mahalanobis–Taguchi System*, New York: McGraw-Hill, 2001.
- [10] G. Blöschl, "Scaling Issues in Snow Hydrology," *Hydrological Processes*, vol. 13, no. 14–15, pp. 2149–2175, 1999.
- [11] G. Betta and A. Pietrosanto, "Instrument Fault Detection and Isolation: State of the Art and New Research Trends," *IEEE Transactions on Instrumentation and Measurement*, vol. 49, no. 1, pp. 100–107, Feb. 2000.
- [12] J. Liu, D. Djurdjanovic, K. A. Marko, and J. Ni, "A Divide and Conquer Approach to Anomaly Detection, Localization and Diagnosis," *Mechanical Systems and Signal Processing*, vol. 23, pp. 2488–2499, 2009.
- [13] T. Takahashi, H. Okaniwa, T. Sakusabe, and N. Schibuya, "A Study on Performance Degradation of Digital Electronic Equipment under Electromagnetic Disturbance," *IEICE Transactions on Communications*, Vol. E90–B, No.6, pp. 1338 – 1343, June 2007.

- [14] T. Miyazaki and T. Omata, "Electromigration Degradation Mechanism for Pb-free Flip-chip Micro Solder Bumps," *Microelectronics and Reliability*, Vol. 46, No. 9-11, pp. 1898-1903, Sept. – Nov. 2006.
- [15] J. Quitmeyer, "Inhibiting Metal Corrosion: Identifying Common Causes of Metal Degradation is the key to Developing and Employing Preventive Measures," *Metal Finishing*, vol. 106, no. 3, pp. 33-39, 2008.
- [16] L. Yang, J. Hu, and M. W. Shin, "Degradation of High Power LEDs at Dynamic Working Conditions," *Solid-State Electronics*, vol. 53, no. 6, pp. 567-570, June 2009.
- [17] T. Riho, A. Suzuki, J. Oro, K. Ohmi, and H. Tanaka, "The Yield Enhancement Methodology for Invisible Defects using the MTS+ Method," *IEEE Transactions on Semiconductor Manufacturing*, vol. 18, no. 4, pp. 561–568, Nov. 2005.
- [18] L. Abdesselam and C. Guy, "Time-Frequency Classification Applied to Induction Machine Faults Monitoring," *32nd IEEE Annual Conference on Industrial Electronics, IECON 2006*, pp. 5051–5056, Nov. 2006,.
- [19] O. Schabenberger and F. J. Pierce, *Contemporary Statistical Models for the Plant and Soil Sciences*, First edition, CRC Press, 2001.
- [20] S. Shou-kui, W. Xiao-feng, and S. Xi-jing, "Discrimination Methods for the Classification of Breast Cancer Diagnosis," *International Conference on Neural Networks and Brain, 2005. ICNN&B '05*, vol. 1, pp. 259–261, Oct. 2005.
- [21] Y. Sung, K. Ikeda, C. Honda, and T. Sakoda, "Monitoring of an MgO Degradation in AC Plasma Display Panel using Multivariate Analysis Method," *Japanese Journal of Applied Physics, Part 1: Regular Papers and Short Notes and Review Papers*, vol. 45, no. 10 B, pp. 8104-8108, Oct 21, 2006.
- [22] B. Rai, R. B. Chinnam, and N. Singh, "Prediction of Drill-Bit Breakage from Degradation Signals Using Mahalanobis-Taguchi System Analysis," *International Journal of Industrial and Systems Engineering*, vol. 3, no. 2, pp. 134-148, 2008.
- [23] R. A. Ion and P. C. Sander, "Early Reliability Prediction in Consumer Electronics Using Weibull Distribution Functions," *Reliability and Maintainability Symposium*, pp. 43-47, Jan. 24-27, 2005.
- [24] D. Das and M. Pecht, Reliability Concepts, Chapter 2, In M. Pecht (Ed.), *Product Reliability, Maintainability, and Supportability Handbook*, 2nd edition, New York: CRC Press, pp. 19 – 30, 2009.
- [25] W. Nelson, *Accelerated Testing, Statistical Models, Test Plans, and Data Analysis*. New York: Wiley, 1990.
- [26] N. Gebraeel, "Sensory-Updated Residual Life Distributions for Components with Exponential Degradation Patterns," *IEEE Transactions on Automation Science and Engineering*, vol. 3, no. 4, pp. 382-393, Oct. 2006.

- [27] A. Ramakrishnan and M. Pecht, "Life Consumption Monitoring Methodology for Electronic Systems," *IEEE Transactions on Components and Packaging Technologies*, vol. 26, no. 3, pp. 625-634, Sep. 2003.
- [28] S. Mathew, D. Das, M. Osterman, M. Pecht, and R. Ferebee, "Prognostic Assessment of Aluminum Support Structure on a Printed Circuit Board," *International Journal of Performability Engineering*, vol. 2, no. 4, pp. 383-395, Oct. 2006.
- [29] S. Mathew, D. Das, M. Osterman, M. Pecht, R. Ferebee, and J. Clayton, "Virtual Remaining Life Assessment of Electronic Hardware Subjected to Shock and Random Vibration Life Cycle Loads," *Journal of the IEST*, vol. 50, no. 1, pp. 86-97, Apr. 2007.
- [30] N. Vichare, P. Rodgers, V. Evely, and M. Pecht, "In-Situ Temperature Measurement of a Notebook Computer - A Case Study in Health and Usage Monitoring of Electronics," *IEEE Transactions on Device and Materials Reliability*, vol. 4, no. 4, pp. 658-663, Dec. 2004.
- [31] N. Vichare, P. Rodgers, V. Evely, and M. Pecht, "Environment and Usage Monitoring of Electronic Products for Health Assessment and Product Design," *International Journal of Quality Technology and Quantitative Management*, vol. 4, no. 2, pp. 235-250, 2007.
- [32] B. Tuchband and M. Pecht, "The Use of Prognostics in Military Electronic Systems," *Proceedings of the 32nd GOMAC Tech Conference*, Lake Buena Vista, FL, pp. 157-160, Mar. 19-22, 2007.
- [33] G. Zhang, C. Kwan, R. Xu, N. Vichare, and M. Pecht, "An Enhanced Prognostic Model for Intermittent Failures in Digital Electronics," *IEEE Aerospace Conference*, Big Sky, MT, pp. 1-8, Mar. 3-10, 2007.
- [34] Y. Zhou and P. K. Mallick, "A Non-linear Damage Model for the Tensile Behavior of an Injection Molded Short E-glass Fiber Reinforced Polyamide-6,6," *Materials Science and Engineering A*, vol. 393, no. 1-2, pp. 303-309, Feb. 2005.
- [35] R. B. Chinnam and P. Baruah, "Autonomous Diagnostics and Prognostics through Competitive Learning Driven HMM-Based Clustering," *Proceedings of the International Joint Conference on Neural Networks*, pp. 2466-2471, July 20-24, 2003.
- [36] F. Camci and R. B. Chinnam, "Hierarchical HMMs for Autonomous Diagnostics and Prognostics," *International Joint Conference on Neural Networks. IJCNN '06*, pp. 2445-2452, July 16-21, 2006.
- [37] L. Lopez, "Advanced Electronic Prognostics Through System Telemetry and Pattern Recognition Methods," *Microelectronics Reliability*, vol. 47, no. 12, pp. 1865-1873, Dec. 2007.
- [38] A. Urmanov and K. Gross, "Failure Avoidance in Computer Systems," *59th Meeting of the Society for Machinery Failure Prevention Technology*, Virginia Beach, VA, Apr. 18-21, 2005.

- [39] N. Vichare, P. Rodgers, and M. Pecht, "Methods for Binning and Density Estimation of Load Parameters for Prognostics and Health Management," *International Journal of Performability Engineering*, vol. 2, no. 2, pp. 149-161, Apr. 2006.
- [40] D. C. Swanson, "A General Prognostic Tracking Algorithm for Predictive Maintenance," *IEEE Aerospace Conference*, Big Sky, MT, pp. 2971 - 2977, Mar. 10-17, 2001.
- [41] W. Wu, J. Hu, and J. Zhang, "Prognostics of Machine Health Condition using an Improved ARIMA-based Prediction Method," *2nd IEEE Conference on Industrial Electronics and Applications, ICIEA 2007*, pp. 1062 – 1067, 23-25 May 2007.
- [42] A. Urmanov, "Electronic Prognostics for Computer Servers," *53rd Annual Reliability & Maintainability Symposium (RAMS)*, Orlando, FL, pp. 65-70, Jan. 22-25, 2007.
- [43] D. Brown, P. Kalgren, C. Byington, and R. Orsagh, "Electronic Prognostics - A Case Study using Global Positioning System (GPS)," *IEEE Autotestcon*, New York, Sep. 26-29, 2005.
- [44] R. Orsagh, D. Brown, M. Roemer, T. Dabnev, and A. Hess, "Prognostic Health Management for Avionics System Power Supplies," *IEEE Aerospace Conference*, Big Sky, MT, Mar. 5-12, 2005.
- [45] D. Brown, P. Kalgren, and M. Roemer, "Electronic Prognostics - A Case Study Using Switched-Mode Power Supplies (SMPS)," *IEEE Instrumentation & Measurement Magazine*, vol. 10, no. 4, pp. 20-26, Aug. 2007.
- [46] V. Sotiris and M. Pecht, "Support Vector Prognostics Analysis of Electronic Products and Systems," *Artificial Intelligence for Prognostics, AAAI Fall Symposium Series*, Arlington, VA, Nov. 9-11, 2007.
- [47] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, New York: John Wiley, 1994.
- [48] P. Kontkanen, J. Lahtinen, P. Myllymaki, and H. Tirri, "An Unsupervised Bayesian Distance Measure," *Advances in Case-Based Reasoning, Proceedings of the 5th European Workshop on Case-Based Reasoning (EWCBR-2000)*, Springer-Verlag, vol. 1898, pp. 148-160, 2000.
- [49] P. C. Mahalanobis, "On the Generalized Distance in Statistics," *Proceedings of National Institute of Science (India)*, vol. 12, pp. 49-55, 1936.
- [50] G. Taguchi and R. Jugulum, *The Mahalanobis-Taguchi Strategy*, New York: John Wiley and Sons, 2002.
- [51] L. Nie, M. Azarian, M. Keimasi, and M. Pecht, "Prognostics of Ceramic Capacitor Temperature-Humidity-Bias Reliability using Mahalanobis Distance Analysis," *Circuit World*, vol. 33, no. 3, pp. 21-28, 2007.
- [52] N. Srinivasan and V. Vaidehi, "Reduction of False Alarm Rate in Detecting Network Anomaly using Mahalanobis Distance and Similarity Measure,"

- International Conference on Signal Processing Communications and Networking, ICSCN 2007, pp. 366-371, 2007.
- [53] F. Mustapha, K. Worden, S. G. Pierce, and G. Manson, "Damage Detection Using Stress Waves and Multivariate Statistics: An Experimental Case Study of an Aircraft Component," *Strain*, vol. 43, no. 1, pp. 47-53, Feb. 2007.
- [54] H. Sohn, C. R. Farrar, M. L. Fugate, and J.J. Czarnecki, "Structural Health Monitoring of Welded Connections," *The First International Conference on Steel & Composite Structures*, Pusan, Korea, June 14-16, 2001.
- [55] Y. Li, A. Vodacek, R. L. Kremens, A. Ononye, and C. Tang, "A Hybrid Contextual Approach to Wildland Fire Detection Using Multispectral Imagery," *IEEE Transactions On Geoscience And Remote Sensing*, vol. 43, no. 9, pp. 2115-2126, Sep. 2005.
- [56] H. Aman, N. Mochiduki, and H. Yamada, "A Model for Detecting Cost-Prone Classes Based on Mahalanobis-Taguchi Method," *IEICE Transactions on Information and Systems*, vol. E89-D, no. 4, pp. 1347-1358, 2006.
- [57] G. Chen, H.G. Zhang, J. Guo, "Efficient Computation of Mahalanobis Distance in Financial Hand-written Chinese Character Recognition," *Proceedings of the Sixth International Conference on Machine Learning and Cybernetics*, Hong Kong, 19-22 Aug.2007.
- [58] J. Galbally, J. Fierrez, and J. Ortega-Garcia, "Classification of Handwritten Signatures Based on Name Legibility," *Proceedings of SPIE - The International Society for Optical Engineering, Biometric Technology for Human Identification IV*, 2007.
- [59] P. J. McKerrow, and B. E. Kristiansen, "Classifying Surface Roughness with CTFM Ultrasonic Sensing," *IEEE Sensors Journal*, vol. 6, no. 5, pp. 1267-1279, October, 2006.
- [60] A. Naftel and S. Khalid, "Classifying Spatiotemporal Object Trajectories using Unsupervised Learning in The Coefficient Feature Space," *Multimedia Systems*, vol. 12, no. 3, pp. 227-238, Dec. 2006.
- [61] M. I. Elbakary and M. S. Alam, "Pattern Recognition In Multi-Band Imagery using Stochastic Expectation Maximization," *Proceedings of SPIE - The International Society for Optical Engineering, Optical Information Systems IV*, pp. 63110W, 2006.
- [62] X. Gigandet, M. B. Cuadra, A. Pointet, L. Cammoun, R. Caloz, and J. Thiran, "Region-Based Satellite Image Classification: Method and Validation," *Proceedings of International Conference on Image Processing, ICIP*, pp. 832-835, 2005.
- [63] Y. Wang and L. Guan, "Recognizing Human Emotion from Audiovisual Information," *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, pp. 1125-1128, 2005.

- [64] A. Bekaryan, H. J. Song, H. P. Hsu, J. Schaffner, and R. Wiese, "Objective Metric for Antenna Patterns Comparison Using Mahalanobis-Taguchi-Gram-Schmidt Method," 66th IEEE Vehicular Technology Conference, VTC-2007, pp. 902-904, Oct. 2007.
- [65] S. Hayashi, Y. Tanaka, and E. Kodama, "A New Manufacturing Control System using Mahalanobis Distance for Maximizing Productivity," IEEE Transactions on Semiconductor Manufacturing, vol. 15, no. 4, pp. 442-446, Nov. 2002.
- [66] T. Riho, A. Suzuki, J. Oro, K. Ohmi, and H. Tanaka, "The Yield Enhancement Methodology for Invisible Defects Using the MTS+ Method," IEEE Transactions on Semiconductor Manufacturing, vol. 18, no. 4, pp. 561-568, Nov. 2005.
- [67] E. A. Cudney, K. Paryani, and K. M. Ragsdell, "Applying the Mahalanobis-Taguchi System to Vehicle Handling," Concurrent Engineering, vol. 14, no. 4, pp. 343-355, Dec. 2006.
- [68] K. Komiya, K. Suzuki, and H. Ikeda, "Specific Video Scenes Enlarged and Displayed on Mobile Phone Set," IEEE Transactions on Consumer Electronics, vol. 53, no. 1, pp. 5-10, Feb. 2007.
- [69] J. Srinivasaraghavan and V. Allada, "Application of Mahalanobis Distance as a Lean Assessment Metric," International Journal of Advanced Manufacturing Technology, vol. 29, no. 11-12, pp. 1159-1168, Aug. 2006.
- [70] A. Fraser, N. Hengartner, K. Vixie, and B. Wohlberg, "Incorporating Invariants in Mahalanobis Distance Based Classifiers: Application to Face Recognition," International Joint Conference on Neural Networks (IJCNN), Portland, OR, USA, July 2003.
- [71] R. Smith and R. K. Mobley, "Rules of Thumb for Maintenance and Reliability Engineers," Butterworth-Heinemann, October 2007.
- [72] J. Day, A. Janus, and J. Davis, "Computer and Internet Use in the United States: 2003," US. Census Bureau, October 2005.
- [73] Hewlett-Packard, Intel, Microsoft, Phoenix, and Toshiba, "Advanced Configuration and Power Interface Specification," Revision 3.0a, December 30, 2005, available at <http://www.acpi.info/DOWNLOADS/ACPIspec30a.pdf>, last accessed on June 4, 2009.
- [74] J. Gu, N. Vichare, T. Tracy, and M. Pecht, "Prognostics Implementation Methods for Electronics," 53rd Annual Reliability & Maintainability Symposium (RAMS), Orlando, FL, pp. 101-106, Jan. 22-25, 2007.
- [75] Z. Liu, D. S. Forsyth, J. P. Komorowski, K. Hanasaki, and T. Kirubarajan, "Survey: State of the Art in NDE Data Fusion Techniques," IEEE Transactions on Instrumentation and Measurement, vol. 56, no. 6, pp. 2435-2451, Dec. 2007.
- [76] M. Baybutt, C. Minnella, A. E. Ginart, P. W. Kalgren, and M. J. Roemer, "Improving Digital System Diagnostics Through Prognostic and Health

- Management (PHM) Technology,” IEEE Transactions on Instrumentation and Measurement, vol. 58, no. 2, pp. 255–262, Feb. 2009.
- [77] D. Thomas, K. Ayers, and M. Pecht, “The ‘Trouble Not Identified’ Phenomenon in Automotive Electronics,” *Microelectronics Reliability*, vol. 42, pp. 641–651, 2002.
- [78] K. C. Gross, K. W. Whisnant, and A. Urmanov, “Electronic Prognostics through Continuous System Telemetry,” *Proceeding of the 60th Society for Machinery Failure Prevention Technology Meeting*, Virginia Beach, VA, Apr. 2006.
- [79] L. Khreisat, “A Machine Learning Approach for Arabic Text Classification using N-Gram Frequency Statistics,” *Journal of Informetrics*, vol. 3, no. 1, pp. 72–77, Jan. 2009.
- [80] S. Bouhouche, M. Lahreche, A. Moussaoui, and J. Bast, “Quality Monitoring using Principal Component Analysis and Fuzzy Logic Application in Continuous Casting Process,” *American Journal of Applied Sciences*, vol. 4, no. 9, pp. 637–644, 2007.
- [81] K. Choi, S. Singh, A. Kodali, K. R. Pattipati, J. W. Sheppard, S. M. Namburu, S. Chigusa, D. V. Prokhorov, and L. Qiao, “Novel Classifier Fusion Approaches for Fault Diagnosis in Automotive Systems,” *IEEE Transactions on Instrumentation and Measurement*, vol. 58, no. 3, pp. 602–611, Mar. 2009.
- [82] S. Kumar, V. Sotiris, and M. Pecht, “Health Assessment of Electronic Products using Mahalanobis Distance and Projection Pursuit Analysis,” *International Journal of Computer, Information, and Systems Science, and Engineering*, vol. 2, no. 4, pp. 242–250, Fall 2008.
- [83] J. Ahn, H. Kim, K. J. Lee, S. Jeon, S. J. Kang, Y. Sun, R. G. Nuzzo, and J. A. Rogers, “Heterogeneous Three-Dimensional Electronics by Use of Printed Semiconductor Nanomaterials,” *Science*, vol. 314, no. 5806, pp. 1754–1757, 2006.
- [84] S. Wu and T. Chow, “Induction Machine Fault Detection using SOM-based RBF Neural Networks,” *IEEE Transaction on Industrial Electronics*, vol. 51, no. 1, Feb 2004, pp. 183-194.
- [85] P. J. Rousseeuw and B. C. van Zomeren, “Unmasking Multivariate Outliers and Leverage Points,” *Journal of the American Statistical Association*, vol. 85, no. 411, pp. 633–639, Sep. 1990.
- [86] F. Sun, S. Omachi, N. Kato, H. Aso, S. Kono, and T. Takagi, “Two-stage Computational Cost Reduction Algorithm Based on Mahalanobis Distance Approximations,” *Proceedings of the 15th International Conference on Pattern Recognition*, vol. 2, pp. 696–699, Sep. 2000.
- [87] T. Mitchell, *Machine Learning*, First edition, McGraw Hill Higher Education, 1997.

- [88] G. Box and D. Cox, "An Analysis of Transformations," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 26, no. 2, pp. 211–252, 1964.
- [89] L. S. Nelson, "Technical Aids," *Journal of Quality Technology*, vol. 16, no. 4, pp. 238-239, Oct. 1984.
- [90] Hewlett-Packard, Intel, Microsoft, Phoenix, and Toshiba, "Advanced Configuration and Power Interface Specification," Revision 3.0a, December 30, 2005, available at <http://www.acpi.info/DOWNLOADS/ACPIspec30a.pdf>, Last accessed on May 29, 2009.
- [91] S. Kumar and M. Pecht, "Baseline Performance of Notebook Computers under Various Environmental and Usage Conditions for Prognostics," *IEEE Transactions on Components and Packaging Technologies*, vol. 32, no. 3, pp. 667–676, Sept. 2009.
- [92] A. K. S. Jardine, D. Lin, and D. Banjevic, "A Review on Machinery Diagnostics and Prognostics Implementing Condition-Based Maintenance," *Mechanical Systems and Signal Processing*, Vol. 20, No. 7, pp. 1483–1510, Oct. 2006.
- [93] S. N. Huang and K. K. Tan, "Fault Detection, Isolation, and Accommodation Control in Robotic Systems," *IEEE Transactions on Automation Science and Engineering*, vol. 5, no. 3, pp. 480–489, July 2008.
- [94] B. J. Ohran, J. Rau, P. D. Christofides, and J. F. Davis, "Plant-wide Fault Isolation using Nonlinear Feedback Control," *Industrial and Engineering Chemistry Research*, vol. 47, no.12, pp. 4220–4229, June 18, 2008.
- [95] N. Vichare and M. Pecht, "Enabling Electronic Prognostics using Thermal Data," *Proceedings of the 12th International Workshop on Thermal Investigation of ICs and Systems*, Nice, Côte d'Azur, France, September 27–29, 2006.
- [96] R. Jaai and M. Pecht, "Fusion Prognostics," *13th Australian International Aerospace Congress (AIAC 13)*, Melbourne, March 9-12, 2009.
- [97] K. Goebel, N. Eklund, and P. Bonanni, "Fusing Competing Prediction Algorithms for Prognostics," *IEEE Aerospace Conference*, Big Sky, MT, March 2006.
- [98] H. Chen, G. Jiang, C. Ungureanu, and K. Yoshihira, "Failure Detection and Localization in Component Based Systems by Online Tracking," *Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, Chicago, Illinois, USA, August 21-24, 2005.
- [99] H. Wang, Z. Song, and P. Li, "Fault Detection Behavior and Performance Analysis of Principal Component Analysis Based Process Monitoring Methods," *American Chemical Society*, vol. 41, pp. 2455–2464, 2002.
- [100] H. H. Yue and S. J. Qin, "Reconstruction-Based Fault Identification Using a Combined Index," *American Chemical Society*, vol. 40, pp. 4403-4414, 2001.

- [101] M. A. Demetriou, "A Model-Based Fault Detection and Diagnosis Scheme for Distributed Parameter Systems: A Learning Systems Approach," *ESAIM–Control Optimization and Calculus of Variations*, vol. 7, pp. 43–67, 2002.
- [102] P. M. Frank, S. X. Ding, and T. Marcu, "Model-Based Fault Diagnosis in Technical Processes," *Transactions of the Institute for Measurement and Control*, vol. 22, pp. 57–101, 2000.
- [103] D. Antory, G. W. Irwin, U. Kruger, and G. McCullough, "Improved Process Monitoring Using Nonlinear Principal Component Models," *International Journal of Intelligent Systems*, vol. 23, no. 5, pp. 520–544, May 2008.
- [104] M. Kettunen, P. Zhang, and S.J. Jounelaa, "An Embedded Fault Detection, Isolation and Accommodation System in a Model Predictive Controller for an Industrial Benchmark Process," *Computers and Chemical Engineering*, vol. 32, pp. 2966–2985, 2008.
- [105] J. L. Green, G. Camilli, and P. B. Elmore, *Handbook of Complementary Methods in Education Research*, Lawrence Erlbaum Associates, Washington, D.C., 3rd Edition, 2006.
- [106] S. Kumar, V. Sotiris, and M. Pecht, "Mahalanobis Distance and Projection Pursuit Analysis for Health Assessment of Electronic Systems," *IEEE Aerospace Conference*, Big Sky, MT, March 2008.
- [107] S. Kumar, E. Dolev, M. Pompetzki, and M. Pecht, "A Residual Estimation Based Approach for Isolating Faulty Parameters," *IEEE Aerospace Conference*, Big Sky, MT, March 2009.
- [108] E. B. Martin, A. J. Morris, and J. Zhang, "Process Performance Monitoring Using Multivariate Statistical Process Control," *IEEE Proceedings on Control Theory Application*, vol. 143, no. 2, pp. 132-144, March 1996.
- [109] E. P. Carden and P. Fanning, "Vibration Based Condition Monitoring: A Review," *Journal of Structural Health Monitoring*, vol. 3, no. 4, pp. 355–377, 2004.
- [110] P. Chang, A. Flatau, and S. Liu, "Review Paper: Health Monitoring of Civil Infrastructure," *Journal of Structural Health Monitoring*, vol. 3, no. 3, pp. 257–267, 2003.
- [111] C.S. Daw, C. E. A. Finney, and E. R. Tracy, "A Review of Symbolic Analysis of Experimental Data," *Review of Scientific Instrument.*, vol. 74, no. 2, pp. 915-930, 2004.
- [112] S. Chin, A. Ray, and V. Rajagopalan, "Symbolic Time Series Analysis for Anomaly Detection: A comparative evaluation," *Signal Process*, vol. 85, pp. 1859–1868, 2005.
- [113] A. Wrangsjö and H. Knutsson, "Histogram Filters for Noise Reduction," *Proceedings of the SSAB Symposium on Image Analysis*, March 2003.

- [114] A. Singhal and D. E. Seborg, "Pattern Matching in Multivariate Time Series Databases using a Moving-Window Approach," *Industrial & Engineering Chemistry. Research*, vol. 41, no.16, pp 3822–3838, 2002.
- [115] J. Gu, D. Barker and M. Pecht, "Prognostics Implementation of Electronics under Vibration Loading," *Microelectronics Reliability*, vol. 47, no. 12, pp. 1849–1856, Dec. 2007.
- [116] S. Narasimhan and G. Biswas, "Model-Based Diagnosis of Hybrid Systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, vol. 37, no. 3, pp. 348–361, 2007.
- [117] S. Markovitch, *Information Filtering: Selection Mechanisms in Learning Systems*, Ph.D. Thesis, EECS Department, University of Michigan, 1989.
- [118] C. Finney, J. B. Green, and C. S. Daw, "Symbolic Time-series Analysis of Engine Combustion Measurements" SAE Paper No. 980624, 1998.
- [119] J. Lin, E. Keogh, S. Lonardi, and B. Chiu, "A Symbolic Representation of Time Series with Implications for Streaming Algorithms," In *Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*, New York, NY, 2003, pp. 2–11.
- [120] R. Agrawal, C. Faloutsos, and A. Swami, "Efficient Similarity Search in Sequence Databases" In *Proceedings of the 4th International Conference of Foundations of Data Organization and Algorithms (FODO)*, Chicago, Oct. 1993, pp. 69-84.
- [121] K. Chan and A.W. Fu, "Efficient Time Series Matching by Wavelets," *Proceedings of the 15th IEEE International Conference on Data Engineering*, Sydney, Australia, 1999, pp. 126–133.
- [122] E. Keogh, K. Chakrabarti, M. Pazzani, and S. Mehrotra, "Dimensionality Reduction for Fast Similarity Search in Large Time Series Databases," *Journal of Knowledge and Information Systems*, vol. 3, pp. 263–286, 2000.
- [123] E. Keogh, K. Chakrabarti, M. Pazzani, and S. Mehrotra, "Locally Adaptive Dimensionality Reduction for Indexing Large Time Series Databases," In *proceedings of ACM SIGMOD Conference on Management of Data*. Santa Barbara, CA, May 21-24, 2001, pp. 151–162.
- [124] A. Ray, "Symbolic Dynamic Analysis of Complex Systems for Anomaly Detection," *Signal Processing* vol. 84, no. 7, pp.1115–1130, 2004.
- [125] C. E. Shannon, "A Mathematical Theory of Communication," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 5, no. 1, pp. 3–55, Jan. 2001.
- [126] M. Markou and S. Singh, "Novelty Detection: A Review Parts 1 & 2," *Signal Process*, vol. 83, no. 12, pp.2481–2521, 2003.
- [127] S. Kumar and M. Pecht, "Approach to Fault Identification for Electronic Products using Mahalanobis Distance," *IEEE Transactions on Instrumentation and Measurement*, accepted for publication, IM-09-1984R2.

- [128]R. Badii and A. Politi, *Complexity Hierarchical Structures and Scaling in Physics*, Cambridge, U.K.: Cambridge University Press; 1st edition, Aug. 21, 2008.
- [129]L. R. Rabiner, “A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition,” *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.
- [130]B. H. Juang and L. R. Rabiner, “Hidden Markov Models for Speech Recognition,” *Technometrics*, vol. 33, no. 3, pp. 251–272, Aug. 1991.
- [131]S. Huang, M. Ahmadi, and M. A. Sid-Ahmeda, “A Hidden Markov Model based Character Extraction Method,” *Pattern Recognition*, vol. 41, no. 9, pp. 2890–2900, Sept. 2008.
- [132]H. Hassanpour and P. M. Farahabadi, “Using Hidden Markov Models for Paper Currency Recognition,” *Expert Systems with Applications*, vol. 36, no. 6, Pages 10105–10111, Aug. 2009.
- [133]I. Daubechies, “Orthogonal bases of Compactly Supported Wavelets,” *Communications on Pure and Applied Mathematics*, vol. 41: no. 7, pp. 909–996, 1988.
- [134]X. Z. Tang, E. R. Tracy, A. D. Boozer, A. Debrauw, and R. Brown, “Symbol Sequence Statistics in Noisy Chaotic Signal Reconstruction,” *Physical Review E*, vol. 51, no. 5, pp. 3871–3889, 1995.
- [135]P. Baruah and R. B. Chinnam, “HMMs for Diagnostics and Prognostics in Machining Processes,” *International Journal of Production Research*, vol. 43, pp. 1275–1293, 2005.