Mahalanobis Distance and Projection Pursuit Analysis for Health Assessment of Electronic Systems

Sachin Kumar, Vasilis Sotiris and Michael Pecht
Prognostics Health Management Lab
Center for Advanced Life Cycle Engineering (CALCE)
University of Maryland, College Park, MD 20742
301-405-5323
pecht@calce.umd.edu

Abstract—This paper presents a Mahalanobis Distance and Projection pursuit analysis based prognostic and diagnostic approach for early detection of anomalies in electronic products and systems. These have been used to detect deviations in system performance from normal operation, and are efficient at characterizing products with short field histories. A case study is presented to demonstrate that an “abnormal” system can be distinguished from a “normal” system and that a new system can be characterized based on existing baselines from different computer models.1 2

TABLE OF CONTENTS
1. INTRODUCTION ...............................................................1
2. EXPERIMENTAL SETUP ..................................................1
3. METHODOLOGY TO IDENTIFY ABNORMALITIES IN ELECTRONIC PRODUCTS ................................................3
4. DATA ANALYSIS AND DISCUSSION ..................................5
5. FAULT ISOLATION – DOMINANT VARIABLES USING PROJECTION PURSUIT ANALYSIS.............................7
6. CONCLUSIONS ................................................................7
ACKNOWLEDGEMENT ........................................................7
REFERENCES ......................................................................8
BIOGRAPHY ........................................................................9

1. INTRODUCTION
Prognostics and health management (PHM) is a process of predicting the future reliability of the system by assessing the extent of deviation or degradation of a product from its expected normal operating conditions in a preemptive and opportunistic manner to the anticipation of failures. This can enable continuous, autonomous, real time monitoring of the health conditions of a system by means of embedded or attached sensors with minimum manual intervention to evaluate its actual life-cycle conditions, to determine the advent of failure, and to mitigate system risks. The term “diagnostics” refers to the detection and isolation of faults or failures and “prognostics” refers to the process of predicting a future state (of reliability) of the system based on its current and historic conditions. The aim of failure prognosis is intended to identify and estimate the advancement of fault conditions to system failure.

Quantification of degradation and the progression from faults to failure in electronic products and systems is a challenging task. Gu et. al. [2] identifies six levels of prognostics implementation for electronics, from on–chip packaging to complete systems of systems. They provided an approach for prognostics implementation at the various levels of electronics, based on failure modes, mechanisms and effects analysis.

Zhang et. al. [3] proposed a model to assess intermittent as well as hard failures. The model is a fusion of two prediction algorithms based on life consumption monitoring and uncertainty adjusted prognostics.

Vichare et. al. [1][4][5] proposed methods for monitoring and recording in-situ temperature loads. This includes methods for embedding data reduction and load parameter extraction algorithms into the sensor modules to enable reduction in on-board storage space, low power consumption, and uninterrupted data collection.

Two prognostic approaches based on classification theory are presented in this paper. They are capable of system level anomaly detection in multivariate, data-rich environments. One methodology uses the Mahalanobis Distance (MD) and the other uses a projection pursuit analysis (PPA) to analyze on-line system data. Both approaches are used to monitor the health of the system and identify onsets and periods of abnormalities. The PPA approach is also used to find the contribution of the system parameters as a means of identifying dominant and potentially faulty parameters [8] [11]. Experiments were performed on notebook computers to generate data and validate the analysis approaches. The experimental details, the algorithmic approach to anomaly detection, and a case study are discussed.

2. EXPERIMENTAL SETUP
To demonstrate the feasibility of the proposed methodology, experiments were conducted to define a baseline for healthy systems and to identify specific parameter behavior. Notebook computers were exposed to a set of environmental conditions representative of the extremes of their life cycle profiles. The performance parameters, the fan speed, CPU temperature, motherboard temperature, videocard temperature, %C2 state, %C3 state, %CPU usage, and %CPU throttle were monitored in-situ.
during the experiments. The baseline of healthy systems was used to differentiate unhealthy systems from healthy ones. The proposed anomaly detection methodology was verified by injecting an artificial fault into the system. Results from the study demonstrate the potential of the approach for system diagnostics and prognostics. Operational and environmental ranges and profiles that constitute a “healthy system” were used to replicate the real time usage of the notebook computer. Software was installed on the computer to be used. A set of user activities was defined and simulated using script file to run on notebook computers. An artificial fault was injected into the notebook computers to create and detect any change in system dynamics.

Table 1: Environmental conditions

<table>
<thead>
<tr>
<th>Temperature-Humidity</th>
<th>1. 5°C with uncontrolled RH</th>
<th>2. 25°C with 55% RH</th>
<th>3. 25°C with 93% RH</th>
<th>4. 50°C with 20% RH</th>
<th>5. 50°C with 55% RH</th>
<th>6. 50°C with 93% RH</th>
</tr>
</thead>
</table>

Table 2: Experiments Performed

<table>
<thead>
<tr>
<th>Power Setting</th>
<th>Usage Level</th>
<th>Environmental Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>AC adapter (when battery is fully charged)</td>
<td>Antivirus, Window Idle</td>
<td>1 - 6</td>
</tr>
<tr>
<td></td>
<td>Word Processing, Antivirus</td>
<td>1 - 6</td>
</tr>
<tr>
<td></td>
<td>Music, Movie, Antivirus</td>
<td>1 - 6</td>
</tr>
<tr>
<td></td>
<td>Video Game, Antivirus</td>
<td>1 - 6</td>
</tr>
<tr>
<td>AC adapter (when battery is initially fully discharged)</td>
<td>Antivirus, Window Idle</td>
<td>1 - 6</td>
</tr>
<tr>
<td></td>
<td>Word Processing, Antivirus</td>
<td>1 - 6</td>
</tr>
<tr>
<td></td>
<td>Music, Movie, Antivirus</td>
<td>1 - 6</td>
</tr>
<tr>
<td></td>
<td>Video Game, Antivirus</td>
<td>1 - 6</td>
</tr>
<tr>
<td>Battery only</td>
<td>Antivirus, Window Idle</td>
<td>1 - 6</td>
</tr>
<tr>
<td></td>
<td>Word Processing, Antivirus</td>
<td>1 - 6</td>
</tr>
<tr>
<td></td>
<td>Music, Movie, Antivirus</td>
<td>1 - 6</td>
</tr>
<tr>
<td></td>
<td>Video Game, Antivirus</td>
<td>1 - 6</td>
</tr>
</tbody>
</table>

Experiments were performed on ten identical notebook computers, representative of the state-of-the-art in (2007) notebook computer performance and battery life (nearly three and half an hours on a single battery). For the experiment, six different environmental conditions were considered as shown in Table 1. For each temperature/humidity combination, four usage conditions and three power supply conditions were considered. Factorial experiment were designed to study the effect of each factor on the response variable, as well as the effects of interactions between factors on the response variable. Table 2 shows the list of all 72 experiments. Each computer was turned on for 30 minutes before starting the experiment.

The software for in-situ monitoring was installed on the notebook 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. Selection of this software was based on the authors’ discretion and experience. A script file was written using WinRunner software to simulate user activity. Antivirus application McAfee v8.0 was configured to run on the laptop all the time. A set of files (.doc, .mp3, .ppt, .pdf, .xls) was kept in a folder to be used during simulation. Notebook computers were kept at room temperature between each test condition. When the laptop was powered by the AC adapter (when the battery was fully charged), the test duration was 3.5 hours. When the laptop 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 laptop was powered by its battery only, the test duration was determined by the time it took for the battery to fully discharge.

Same usage conditions were applied on all notebook computers to achieve time synchronization between computer and software application responses. The notebook computer’s power mode was always set to ON. The screen saver and hibernation option were disabled to prevent these functions from occurring during an experiment. The wireless capability of notebook computer was disabled due to the limited wireless connectivity inside the temperature-humidity chambers. Four level of notebook computer usage were chosen:

1. Idle system - In this category the operating system is loaded, all windows are closed, user input from the keyboard or mouse, optical drive are disabled. USB or Firewire peripherals are not attached.

2. Office productivity - In this category, the usage condition is designed to simulate an office work environment. The simulator work is designed to read a word document as well as prepare a new word document. The simulator opens the file explorer and locates a file to be opened. The simulator opens a “technology benchmark report” word document of 88 pages and size of 2.6MB. The simulator reads through the document and uses arrow keys to move page up, page down and selects a paragraph to copy. The simulator opens a new document from the word toolbar and pastes thee copied section to a new document. The simulator resizes both documents to make it easy to
The Mahalanobis Distance (MD) methodology is a process of distinguishing data groups [6][9]. The MD measures distances in multi-dimensional spaces by considering correlations among parameters. The distance is sensitive to the correlation matrix of the healthy group. Health of a system is defined by the Mahalanobis space to monitor the condition of a multi-dimensional system. Two statistical indices, the Hotelling Squared (T^2) and squared prediction error (SPE) are used in the PCA. The SPE statistic is related to the residuals of process variables. The SPE physically tests the fit of the process variables. The SPE is a reliable indicator to a change in the correlation structure of the process variables. The SPE physically tests the fit of new data to the established PCA models and is efficient at identifying outliers from the PCA model [7]. The Hotelling Squared statistic is computed as:

\[ T^2 = (X - \bar{X})^T S^{-1} (X - \bar{X}) \]

where, \( X \) is the data vector, \( \bar{X} \) is the mean vector, and \( S^{-1} \) is the inverse of the covariance matrix.

The SPE statistic is computed as:

\[ SPE = (X - \hat{X})^T (X - \hat{X}) \]

where, \( \hat{X} \) is the predicted value.

Next, the values of the MDs are calculated for the normal items using:

\[ MD = \frac{1}{m} z_j^T C^{-1} z_j \]

where, \( z_j \) is a vector comprised of \( z_{ij} \), \( C = \sum_{j=1}^{n} z_j z_j^T \) is the correlation matrix, and \( n \) is the number of observations.

Next, the test system is considered to determine its health. The MDs for the test system are calculated after the test system is considered to determine its health. The resulting MD values from the test system are compared with the MD values from the normal or healthy system to determine test system’s health.

### Projection Pursuit Analysis

The Projection Pursuit Analysis uses a Principal Components Analysis (PCA), least squares optimization (LS) and a Singular Value Decomposition (SVD) treatment of the data. PCA is used in a wide array of applications to reduce a large data set to a smaller one while maintaining the majority of the variability present in the original data. It’s also very useful in providing compact representation of temporal and spatial correlations in the fields of data being analyzed. PCA facilitates a multivariate statistical control to detect when abnormal processes exist and can isolate the source of the process abnormalities down to the component level.

Two statistical indices, the Hotelling Squared (T^2) and squared prediction error (SPE) are used in the PCA. The SPE statistic is related to the residuals of process variables that are not illustrated by the PCA statistical model, and is a reliable indicator to a change in the correlation structure of the process variables. The SPE physically tests the fit of new data to the established PCA models and is efficient at identifying outliers from the PCA model [7]. The Hotelling Squared statistic is computed as:

\[ T^2 = (X - \bar{X})^T S^{-1} (X - \bar{X}) \]

where, \( X \) is the data vector, \( \bar{X} \) is the mean vector, and \( S^{-1} \) is the inverse of the covariance matrix.

The SPE statistic is computed as:

\[ SPE = (X - \hat{X})^T (X - \hat{X}) \]

where, \( \hat{X} \) is the predicted value.
T\(^2\) score measures the Mahalanobis distance from the projected sample data point to the origin in the signal space defined by the PCA model.

The primary objectives of principal component analysis are data summarization, classification of variables, outlier detection, early warning of potential malfunctions and isolation of fault. PCA seeks to find a few linear combinations which can be used to summarize the data with a minimal loss of information. Let \(X = x_1, x_2, x_3, ..., x_n\) be an \(m\) - dimensional data set describing the system variables. The first principal component is the linear combination of the columns of \(X\), i.e. the variables, which describes the greatest variability in \(X\). \(t_1 = Xp_1\) subject to \(|p_1| = 1\). In the \(m\) - dimensional space \(p_1\) defines the direction of greatest variability, and \(t_1\) represents the projection of each sample data point onto \(p_1\). The second principal component is the linear combination defined by \(t_2 = Xp_2\) which has the next greatest variance subject to \(|p_2| = 1\) and subject to the condition that it is orthogonal to the first principal component, \(t_1\) [10]. Essentially PCA decomposes the original signal \(X\) as

\[
X = TP^T = \sum_{i=1}^{m} t_i p_i^T
\]  

(3)

where \(p_i\) is chosen to be an eigenvector of the covariance matrix of \(X\). \(P\) is defined as the principal component loading matrix and \(T\) is defined to be the matrix of principal component scores. The loadings provide information as to which variables contribute the most to individual principal components, and can help isolate the dominant faulty variables. In the approach used in this paper, each variable is separately scaled to zero mean and unit variance.

One important feature of the PCA model is that it gives information about the noise structure of the original data which means that it can tell something about variables that do not dominate on the variance level but are indeed degrading or faulty. Consequently, it is desirable to exclude less influential principal components from the signal space defined by the PCA model, which leads to the decomposition of \(X\) into the signal and residual subspaces. The signal subspace is intended to capture the variables that are contributing to any abnormal process variability and the residual subspace will complement this by examining the variables that are effectively over-shadowed by dominant variables in the signal subspace. It’s important to note that faulty variables aren’t always the ones that exhibit the greatest variability. An example of this phenomenon is presented in the data analysis and discussion section of this paper.

**Principal Component Subspace Decomposition**

Subspace decomposition into Principal Components can be accomplished using singular value decomposition of matrix \(X\). The SVD of data matrix \(X\), is expressed as \(X = USV^T\), where \(S = \text{diag}(s_1, ..., s_m) \in \mathbb{R}^{n \times m}\), and \(s_1 > s_2 > ... > s_m\). The two orthogonal matrices \(U\) and \(V\) are called the left and right eigen-matrices of \(X\). Based on the singular value decomposition, the subspace decomposition of \(X\) is expressed as:

\[
X = X_s + X_r = UsVs^T + UsSrV^T
\]  

(4)

The signal space \(S_s\) is defined by the PCA model and the residual subspace \(S_r\) is taken as the residual space [11]. The diagonal \(S_s\) are the singular values \(\{s_1, ..., s_k\}\), and \(\{s_{k+1}, ..., s_m\}\) belong to the diagonals of \(S_r\). The set of orthonormal vectors \(U_r = [u_1, u_2, ..., u_k]\) form the bases of signal space \(S_s\). The projection matrix \(P_s\) onto the signal subspace is given by:

\[
P_s = U_sU_s^T
\]  

(5)

The residual subspace is the orthogonal complement of the signal subspace and the projection of the original data onto it can be expressed as:

\[
P_r = I - P_s
\]  

(6)

Any vector \(X\) can be represented by a summation of two projection vectors from subspaces \(S_s\) and \(S_r\):

\[
X = X_s + X_r = Px + (I - P_s)x
\]  

(7)

The subspace decomposition can also be accomplished by the eigen analysis of the correlation matrix of \(X, C\), which is expressed as follows, where the columns of \(U\) are actually the eigenvectors of \(C\), and the eigen values of \(C\) are the squared singular values of the diagonal matrix \(S\). The eigen values provide a measure of the variance of each of the eigenvectors and determine the selection of the principal components and the number of principal components to choose.

\[
C = (1/n)XX^T = (1/n)USU^T
\]  

(8)

**Fault Detection Using PPA**

From the normal historical data one can derive the nominal normal system behavior statistics, mean, variance and from the above analysis the signal and residual subspaces. From the subspaces, we extract some statistics to describe the data distributions in two subspaces[12]. One is the Hotelling \(T^2\) which measures the variation of each sample is the signal subspace. For a new sample vector \(x\), it is expressed as:

\[
T^2 = x^TU_sS^2U_s^Tx
\]  

(9)

where \(S\) is the covariance of \(X\), and is equal to \(U^TU\). Another statistic, the squared prediction error (SPE), indicates how well each sample conforms to the PCA model, measured by the projection of the sample vector onto the residual space

\[
SPE = ||P_sx||^2 = r = ||(I - P_s)x||^2
\]  

(10)

The process is considered normal if
\[ \text{SPE} \leq \delta^2 \text{ and } T^2 \leq \tau^2 \]  

where \( \delta^2 \) and \( \tau^2 \) are the control limits for the SPE and \( T^2 \) statistics, respectively, given a 1-\( \alpha \) confidence level. These limits assume that \( x \) follows a normal distribution and \( T^2 \) follows a \( \chi^2 \) distribution with \( k \) degrees of freedom, where \( k \) is defined to be the cut off for the number of principal components used in the PCA model. Because SPE is a measure of the deviation in the residual space, it can be used to identify when the current operation deviates from the expected in terms of parameters that are not dominant but still abnormal. On the other hand, the \( T^2 \) will be more sensitive to the regular fluctuations that move the process away from normal based on the projections in the model subspace. The two statistics function independently in this analysis, although a combined index has been developed for process monitoring. Yue and Qin [13] proposed a convenient alternative for merging the information from SPE and \( T^2 \). For the purpose of this analysis though, each statistic will be examined separately.

**Fault Isolation and Contribution Plots**

After a fault has been detected, there are several methods that can be used to determine the critical system parameters. Contribution plots continue to be a widely used for fault diagnosis. During monitoring of the system each new observation that is projected on the model and residual subspaces will have a unique impact onto each subspace respectively as discussed earlier. The impact is quantified by calculating the contributions to the SPE and \( T^2 \). The larger the contribution of an individual parameter, the more likely it is that the parameter is the reason for the changes or faults. The contribution of the \( m \)th parameter to the SPE is found by taking the squared residual associated with that parameter \( x_m \) by:

\[ \text{SPE}_m = r_m^2 \]  

The contribution of all parameter to \( T^2 \) is given (in terms of the SVD) by:

\[ T^2 = ||XUS^{-1/2}U^T|| \]  

**4. Data Analysis and Discussion**

The experiments were performed on ten new notebook computers with an assumption that these systems are representative of normal/healthy systems. The MD value obtained from these datasets called Mahalanobis space is used to identify anomalies present in system. Five thousand data points are selected from the experiments performed at CALCE. An effort was made to demonstrate the capability of MD method to detect anomalies present in the test notebook computer and characterization of this computer model based on the baseline experiments. In Figure 1, experimental data are used as a baseline to identify the anomalies present in a test abnormal notebook computer. From Figure 1, the test system shows problems from the beginning that is verified by observing the data file of test notebook computer in which the fan was not operating and the three monitored temperatures were high by \( \sim 10^\circ \text{C} \). The drop in MD value at 2700th data point is an indication that the computer was shutdown and then restarted, which caused a temporary drop in these temperatures. Very high values of MD at and around 3500th data point are due to the different correlation structure for CPU usage, CPU throttle and CPU state variables. Since the MD value is very sensitive to the correlation of different parameters, the MD value corresponding to an abnormal observation is high. Thus, it can be inferred that the MD value is a good measure to identify anomalies present in the system.

![Figure 1: Comparison of MD values of abnormal test computer with baseline](image)

Based on our results, we have seen that a test computer can be characterized using experimental data representative of “healthy” computers (of different model). This leads us to claim that a comprehensive baseline can be used to characterize (the health) of new computers without the need to conduct a whole set of experiments for that new computer (regardless of the model). One can observe that the MD values for the test (normal) notebook computer model are separated from the MD values corresponding to the baseline experiments. Statistical metrics corresponding to the analysis are given in Table 3.

**Table 3: Statistics of MD value based on experimental data**

<table>
<thead>
<tr>
<th>System/Stats</th>
<th>Baseline</th>
<th>Test Normal</th>
<th>Test Abnormal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>0.91</td>
<td>4.27</td>
<td>17.65</td>
</tr>
<tr>
<td>Std Dev</td>
<td>0.65</td>
<td>1.401</td>
<td>5.14</td>
</tr>
<tr>
<td>1st Quartile</td>
<td>0.53</td>
<td>3.19</td>
<td>16.01</td>
</tr>
<tr>
<td>3rd Quartile</td>
<td>1.03</td>
<td>5.32</td>
<td>19.87</td>
</tr>
</tbody>
</table>

The table gives the simple statistics of MD values for different notebook computer models used as the baseline, test normal and test abnormal. Average MD values for the test normal system are higher by \( \sim 3.5 \) as compared to MD values for the baseline. The abnormal test system shows very high MD values as compared to the baseline as well as
test normal system baseline. The average difference between MD a value for test normal and test abnormal is ~13. This indicates that the baseline can be used for anomalies detection across computer models, assuming the operating conditions are similar.

The data from the test normal computer are used as a baseline to identify the anomalies present in test abnormal computer. A separation was observed between MD values of normal and abnormal system. This strengthens the argument made earlier that the MD method can be used for anomaly detection. Statistical metrics of MD values for this are given in Table 4.

Table 4: Statistics of MD value based on Test normal data

<table>
<thead>
<tr>
<th>System/Stats</th>
<th>Test Normal</th>
<th>Test Abnormal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>0.83</td>
<td>10.72</td>
</tr>
<tr>
<td>Std Dev</td>
<td>1.16</td>
<td>3.13</td>
</tr>
<tr>
<td>1st Quartile</td>
<td>0.24</td>
<td>8.07</td>
</tr>
<tr>
<td>3rd Quartile</td>
<td>0.88</td>
<td>13.31</td>
</tr>
</tbody>
</table>

Table 4 gives the simple statistics of MD values for the normal and abnormal test systems. Average MD values of the test normal system are ~0.8. The average difference between MD values of the test normal and test abnormal is ~10. By looking at the statistics for the MD values for the different systems, it can be observed that the difference in average MD value for the test normal computer in both cases are equivalent to the difference in MD values for the abnormal computer and the trend for the MD values in both cases is similar. This illustrates that the baseline can be used for characterization of a new computer model. This will allow us to characterize a new notebook computer regardless of the model and reduce the time for analysis. The following paragraphs discuss Projection Pursuit analysis approach for system diagnostics.

The goal of the Projection Pursuit analysis was to use Hotelling $T^2$ and SPE statistics from a healthy computer and successfully classify and detect faults in a new computer of the same model. Two “healthy” baseline sets of $T^2$ and SPE statistics were derived from two sources: one from a CALCE baseline set, based on 10 healthy computers of a different model, and the other based on one computer of the same model. From the analysis, the known faulty computer was identified as abnormal based on both comparisons. Figure 2 and Figure 3 are used as example plots to show the analysis results by plotting the SPE and Hotelling $T^2$ statistics for the above two scenarios versus the number of sample points. The lower pink line indicates baseline “healthy” values for each statistic and the upper blue line indicates the test values for each statistic for the abnormal test computer. It is clear from the plots that the abnormal test computer statistics are different from the baseline computer for both scenarios. The first five principal components were used to form the model space and the remaining three for the residual space.

From these results we see that the variability of the process in both the PCA model and residual subspaces can be used to capture abnormal system behavior. The detection is based on the geometry of the problem whose dimensions are established by the PCA model and residual subspaces. The subspaces as discussed are constructed from the “healthy” data and represent a fixed frame of reference used to compare incoming new observations. New observations are taken as a point in the multi-dimensional space and are projected onto the PCA model and residual subspaces respectively.

With the projection the new observation is reduced from its original dimension $R_1 \times m$ to the lower dimension of the PCA model, $R_1 \times k$, where $k$ is the number of principal components used to form the model subspace. If the projection of the observation falls within the statistical control limits $\tau^2$ and $\delta^2$, of the model and residual subspaces respectively, then it is taken as normal or “healthy”, otherwise it’s treated as abnormal or “un-healthy”. Faults can be masked by the PCA model. This can occur for example when the new computer starts to exhibit abnormal behavior yet the variability of test data in both the model and residual subspaces fall within the “healthy” control limits for the system.

For the baseline, the statistics are modeled with patterns that are similar for both comparisons. One of the explanations for this is that the baseline for the computers captures the necessary range and variability of normal operating conditions of such computer models. Without the use of the control limits this analysis is left to identify the presence of
abnormalities between test and training data and also to identify the critical system parameters.

5. Fault Isolation – Dominant Variables Using Projection Pursuit Analysis

The model space is designed to capture the data that varies the most, whereas the residual space is designed to capture the data that does not vary but contributes to a faulty state. The residual space can therefore detect changes in the distribution from variables that are degrading or have faults and are not effecting the variance. Below are the principal components for the entire subspace S. Each principal component is composed of the eight parameters with a particular weighting as shown in Table 5. The model/signal subspace is composed of the first four principal components. This was chosen based on iterative experimentation to best capture the faults. The decision of how many principal components are chosen to represent the model/signal space is based on experience and understanding of the data at hand. There are computational/statistical techniques that can provide estimates for the selection of the number of PCs to optimized results. The remaining columns span the residual subspace.

Each variable is represented by each respective row of matrix $[S]$. The first row shows the contributions of the fan speed, and the rest show the CPU temperature, motherboard temperature, video card temperature, %C2 state, %C3 state, %CPU usage and %CPU throttle from top to bottom in matrix $[S]$.

Table 5 – Principal Component of subspace S and parameter contribution

<table>
<thead>
<tr>
<th>$[S]$</th>
<th>PC1</th>
<th>PC2</th>
<th>PC3</th>
<th>PC4</th>
<th>PC5</th>
<th>PC6</th>
<th>PC7</th>
<th>PC8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fan Speed</td>
<td>0.999</td>
<td>0.019</td>
<td>0.001</td>
<td>0.001</td>
<td>0.001</td>
<td>0.002</td>
<td>0.004</td>
<td>0.001</td>
</tr>
<tr>
<td>CPU Temp</td>
<td>0.005</td>
<td>0.076</td>
<td>0.042</td>
<td>0.048</td>
<td>0.000</td>
<td>0.484</td>
<td>0.856</td>
<td>0.153</td>
</tr>
<tr>
<td>Mother board Temp</td>
<td>0.002</td>
<td>0.035</td>
<td>0.077</td>
<td>0.079</td>
<td>0.004</td>
<td>0.523</td>
<td>0.158</td>
<td>0.830</td>
</tr>
<tr>
<td>Video card Temp</td>
<td>0.000</td>
<td>0.048</td>
<td>0.107</td>
<td>0.093</td>
<td>0.002</td>
<td>0.670</td>
<td>0.490</td>
<td>0.537</td>
</tr>
<tr>
<td>%C2 State</td>
<td>0.001</td>
<td>0.060</td>
<td>0.018</td>
<td>0.091</td>
<td>0.994</td>
<td>0.001</td>
<td>0.002</td>
<td>0.001</td>
</tr>
<tr>
<td>%C3 State</td>
<td>0.015</td>
<td>0.730</td>
<td>0.384</td>
<td>0.554</td>
<td>0.102</td>
<td>0.005</td>
<td>0.025</td>
<td>0.011</td>
</tr>
<tr>
<td>%CPU Usage</td>
<td>0.013</td>
<td>0.661</td>
<td>0.271</td>
<td>0.690</td>
<td>0.028</td>
<td>0.114</td>
<td>0.019</td>
<td>0.000</td>
</tr>
<tr>
<td>%CPU Throttle</td>
<td>0.001</td>
<td>0.130</td>
<td>0.872</td>
<td>0.439</td>
<td>0.033</td>
<td>0.166</td>
<td>0.038</td>
<td>0.005</td>
</tr>
</tbody>
</table>

From the decomposition of $[S]$ we can see that the model space variations should be dominated by the fan speed followed by %C3 state, %CPU throttle and usage. In the residual subspace the temperature components are dominant. We expect that the temperature variables to be highly dominant. Changes in the temperature are expected in turn to be less obvious to changes in system variance and should contribute to the shape of the multivariate data distribution. Such a distribution can be modeled as Gaussian mixtures, but in general a hard task. Intuitively, if the fan speed is not functioning, we expect that the temperature of the system will rise and become abnormally high. This is at first hand validated by the dominance of the temperatures components as observed in the residual subspace in $[S]$. Mathematically, this is also validated through the parameter contribution plots to the $T^2$ and SPE respectively as illustrated in the contribution plots shown in Figure 4. The contribution plots tell us which parameter is contributing the most to the projection onto each subspace.

It is shown that on the model space the fan speed is highly dominant and varies the most in terms of standard deviation. This phenomenon masks the effect on parameters that are also exhibiting abnormalities but are overpowered by dominant parameters such as the fan speed. The residual space statistic, SPE, captures the inverse information and identifies the parameters that are indeed abnormal but are not dominating in terms of variance. Also interesting is the fact that the mathematics validate our intuition that because the fan isn’t functioning properly, the temperature sensors would be experiencing unusual readings. Note that these results are based on picking the model space using k=4, that is the first four PCs in matrix $[S]$. The selection of more PCs for the model space and consequently fewer PCs for the residual space will change the results slightly. If all eight PCs are used to construct the model space then the SPE will be rendered ineffective although the results for the Hotelling $T^2$ will improve. Even though the results from the Hotelling $T^2$ improve with the selection of more PCs the information available through the SPE is lost. There are ways to select the optimum number of PCs necessary to optimize the information captured from both subspaces, often the selection is purely based on experience or experimentation, although there are statistical methods such as the maximum likelihood estimator (MLE) which can estimate the optimum number of PCs to use.

6. Conclusions

A set of experiments were conducted to establish the “healthy or normal” operation on a set of notebook computers subjected to range of usages and environmental conditions. A test computer was then subjected to field use conditions and evaluated in-situ using Mahalanobis Distance, and Projection Pursuit analysis techniques. The Projection Pursuit analysis method was also used to identify
key parameters for root cause analysis of anomalies. This study emphasizes that the defined baseline can be used to characterize a new computer model. This will allow us to characterize a new notebook computer regardless of the model and reduce the time for analysis.

In this study, PPA and MD were independently used to identify the similarity of new observations to healthy data. PPA performed this analysis in a reduced dimension based on an optimization criterion (maximum variance). It was also found that PPA can identify the faulty parameters based on the data whereas MD requires an understanding of the system. The strength of PPA lies in the ability to decompose the signal and extract additional information not originally available, used to identify faults in the system. PPA overcomes masking effects when working with highly correlated data. The strength of the MD method is that it preserves all the information available because it does not reduce the original dimensionality of the data. The drawbacks of using just the MD method are that it cannot be used directly for fault identification and it is susceptible to masking effects.

With the MD results and our understanding of the system functionality, four critical parameters were empirically identified: the fan speed and the three temperature components (CPU temperature, motherboard temperature, and videocard temperature). In parallel, in the PPA approach, the principal component space also identified the fan speed as the most dominant, and from the residual principal component space three temperature parameters were identified to be dominant, mathematically confirming the earlier empirical conclusion.

The cross-validated result shows that these two algorithms can be used for fault detection and isolation. The MD method can be used for quick fault detection at a system level and fault isolation can be made if related fault to MD signatures are available. PPA can also be used when system faults are not known and where critical parameters need to be identified.

**ACKNOWLEDGEMENT**

This work is sponsored by the members of the CALCE Prognostics and Health Management Consortium at the University of Maryland, College Park.

**REFERENCES**


BIOGRAFÍA

Sachin Kumar recibió la B.S. en Metalurgia de la Ingeniería de la Universidad de Bihar e Instituto de Tecnología y la M.Tech. en Ingeniería de Reliability de la Universidad de Kharagpur. En la actualidad está persiguiendo el Ph.D. en Ingeniería de Mecánica en la Universidad de Maryland, College Park. Sus intereses de investigación incluyen reliability, system pronósticos electrónicos, y monitoring de salud y uso de sistemas.


Michael Pecht tiene un B.S. en Acústica, un M.S. en Ingeniería Eléctrica y un M.S. y un Ph.D. en Ingeniería de Mecánica, Universidad de Wisconsin en Madison. Es ingeniero registrado, Fellow de IEEE y Fellow de ASME. Ha recibido el Premio 3M a la Investigación en Embalaje Electrónico, el Premio IEEE por el liderazgo en el Estándar de Reliability, y el Premio IMAPS William D. Ashman Memorial Achievement por su contribución en la Reliability Analysis. Ha escrito más de veinte libros sobre el desarrollo, uso y el manejo de la cadena de suministro de productos electrónicos. Ha servido como jefe de editorial del IEEE Transactions on Reliability por ocho años y en el consejo asesor de IEEE Spectrum. Ha sido el jefe de editorial de Microelectronics Reliability por once años y editor asociado del IEEE Transactions on Components and Packaging Technology. Ha sido profesor titular en el Centro de Ingeniería de Vida Avanzada (CALCE) y el Consorcio de Productos y Sistemas Electrónicos de la Universidad de Maryland. Ha liderado un equipo de investigación en el área de pronósticos y formó el Consorcio de Gestión de Salud y Pronósticos en la Universidad de Maryland. Ha consultado para más de 50 empresas electrónicas internacionales, proporcionando expertise en planificación estratégica, diseño, prueba, pronósticos, IP y el manejo de riesgo de productos electrónicos y sistemas.