WORKING PAPER
ALFRED P. SLOAN SCHOOL OF MANAGEMENT

A Methodology for Manufacturing Process Signature Analysis

Steven D. Eppinger
Christopher D. Huber
Van H. Pham

#3631-93-MSA November 1993

MASSACHUSETTS
INSTITUTE OF TECHNOLOGY
50 MEMORIAL DRIVE
CAMBRIDGE, MASSACHUSETTS 02139

A Methodology for Manufacturing Process Signature Analysis

Steven D. Eppinger
Christopher D. Huber
Van H. Pham

#3631-93-MSA November 1993

A Methodology for Manufacturing Process Signature Analysis

By
Steven D. Eppinger
Christopher D. Huber
Van H. Pham

Massachusetts Institute of Technology

Abstract

One of the fundamental challenges facing manufacturing engineers today is the achievement of continuous improvement through the implementation of better process control systems. We believe that the improvement of control systems entails the collection of more information about the process and/or more effective use of that information. We present manufacturing process signature analysis in order to construct a relationship between the collected information (process signatures) and the quality of the process output, which can be used for on-line monitoring and control. The general procedure applied in this paper consists of three steps: feature extraction, feature selection, and classification.

We have found that the extracting of large sets of features from signatures is straightforward and that several classification schemes are available, with nueral networks being the most general and powerful method that we have tried. Feature selection, on the other hand, is generally quite difficult for complex data structures. We present several feature extraction methods and show that neural networks can be quite useful in choosing different feature sets. Using a data set from an automated solder joint inspection system, we demonstrate the unique capabilities of neural networks for both feature selection and classification, using more traditional statistical classification techniques as a benchmark.

Keywords:

Manufacturing Process Monitoring
Manufacturing Process Signature Analysis
Feature Selection
Neural Network Applications

1 Introduction

1.1 Motivation

Most manufacturing processes are monitored for output quality, either continuously or on a sampled basis. Process monitoring is used to assess whether or not the process is performing its function adequately so that appropriate corrective action can be taken if necessary. Inputs to the process monitor include one or more measured attributes of the product or process which should reflect the product's ability to perform its intended function.

As we strive to improve the quality of our manufacturing processes, we seek better monitoring and control systems. One approach is to utilize more sophisticated measurements. At every stage of even the simplest manufacturing process, there are many opportunities for quality measurements; involving process inputs (material thickness or temperature), process attributes (feed rate, spindle speed or tool vibration), and measures of the process output (critical dimensions or surface finish). To monitor a larger process or a series of process steps, we could inspect the finished product or assess ultimate customer satisfaction through questionnaires or warranty information. The integration of computers into our manufacturing environment (CIM) can certainly facilitate the collection of these different process measurements but the information by itself is not enough. In fact, without an appropriate procedure for the analysis and evaluation of this information, the resultant flood of data can confound process improvement efforts.

Our approach to process monitoring is called manufacturing process signature analysis. During each cycle, we measure one or more process signals and/or parameters over the duration of the process. We then analyze these measurements to determine the quality of the process iteration which just took place. Ideally, this quality classification would then be fed back through an appropriate controller to close the process control loop.

1.2 Related Signature Analysis Work

Although the focus of our paper is on manufacturing applications, we draw upon research over a broad range of disciplines. In the analysis of human electrocardiogram (ECG) signals, information about the health of a patient is contained in the ECG signature [18].

The ECG is broken down into a series of basic waveform elements called complexes and segments. These simpler elements are analyzed through syntactic pattern recognition techniques where grammars or rules of syntax are used to determine the patient's state of health by classifying the ECG signatures.

Research into computer access security systems employs a signature which consists of the length of time between keystrokes in the entry of a password [1]. A Bayesian classifier is used to authenticate a given password by comparing the current entry against a known signature.

In manufacturing research, success has been achieved using acoustical signals from machining and forming operations (see [3], [13], and [16]). In work involving punch stretching and deep drawing of aluminum sheet metal, an acoustical sensor was placed in direct contact with the sheet metal during the forming process [7]. The energy content, spectral characteristics, and time series behavior of the resulting signature were then used to identify critical transitions in the forming process.

From this literature, we recognize and adopt a general principle for process signature analysis: Parameterize the original signature through the extraction of key features and determine an appropriate classification for the signature based on these extracted features. For the ECG signature, the shape parameters of each component complex (features) are used to determine the patient's health (classification). For password security, keystroke timing is used to authenticate the user's identity. For forming, the energy, spectral characteristics, and time series behavior are used to characterize the forming process. The success of this principle is critically dependent on our ability to locate and extract the appropriate features of the original signature.

In this paper, we extend the feature-extraction strategy for signature analysis to include neural-network-based schemes for feature selection and quality classification. The most important contribution of this paper is our development of a Nueral-Network-based feature selection scheme which can identify the most useful features for a simplified on-line monitoring system.

1.3 Paper Organization

To develop our signature analysis approach in the next section, we begin by explaining a progression of signature classification tools which serve to highlight the specific utility of neural networks and the important role of feature selection in the process. In the following section, a data set from an automated solder joint inspection project, which was the primary vehicle of research, is then described in detail. We then show how the signature analysis methodology is applied to the solder joint inspection data. We conclude with a discussion of the strengths and weaknesses of this approach.

2 Signature Analysis Tools

There are many signature analysis utilities available, from simple statistical process control to artificial neural networks. Selecting the combination of signature analysis tools which are most effective for a given problem requires an understanding of the relationship between signature and quality as well as the capabilities and limitations of the analysis tools which are employed. A progression of classification tools is described below in order of increasing sophistication. Our discussion presents the capabilities and limitations of each tool; successive tools overcome some limitations at the cost of implementation complexity.

2.1 Statistical Process Control

One of the most common process monitoring tools in industry today is statistical process control (SPC). Traditionally, SPC entails sampling the process output at given intervals and then measuring some critical attribute of the sampled parts. The average of these measurements is then plotted on an X-Bar chart while the maximum range of these measurements is plotted on a Range chart. Control limits for these charts are constructed based on the mean and deviation of the "normal" process and the determination of whether or not the process is "in control" is based on whether or not the data points from successive samples fall within these control limits.

When multiple discrete measurements are required to gain an adequate description of a process, we say the signature is *multi-dimensional*. The basic premise of SPC can be extended to accommodate multi-dimensional signatures by expanding the one-dimensional control limits into a *multi-dimensional control limit mask*. Figure 1 illustrates an example of a multi-dimensional signature in the measurement of bias force of a computer's hard disk drive assembly.

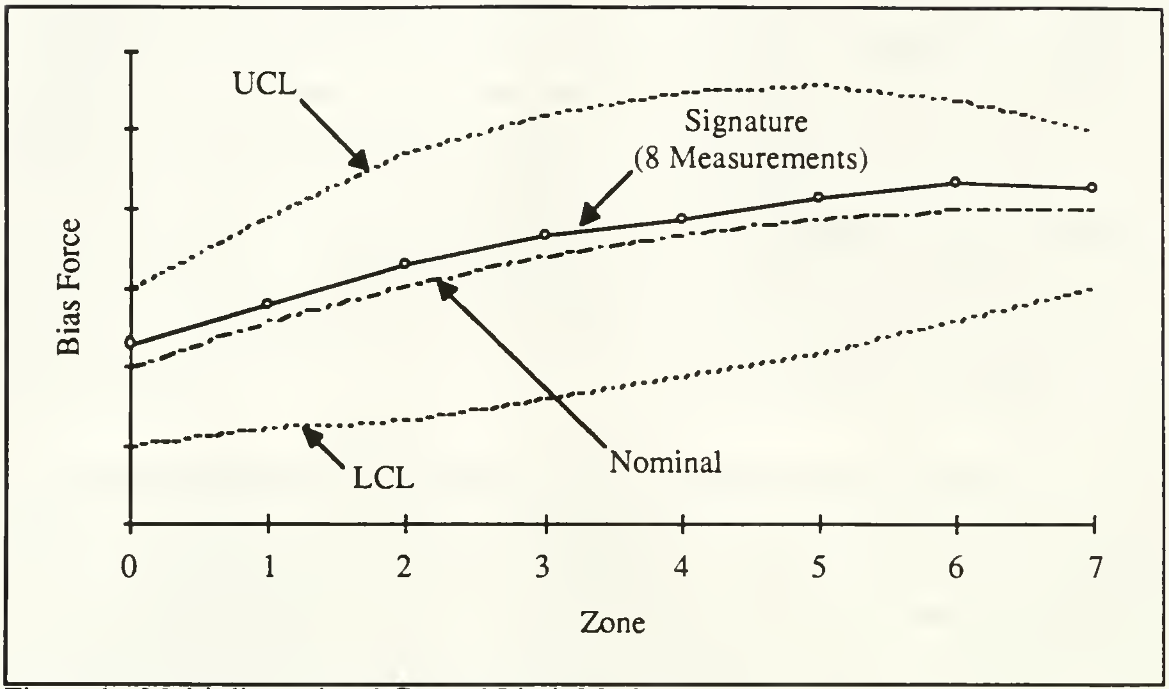

Figure 1 - Multi-dimensional Control Limit Mask

With multi-dimensional SPC, each dimension of the signature has a corresponding pair of control limits determined based on the mean and variance of that dimension, measured for several good quality drives. The set of control limits then constitutes a control limit mask. The accept or reject decision is based on whether or not a given signature falls entirely within the bounds of the control limit mask. A commercial system which utilizes this basic technique was developed at General Motors and is now publicly available through Assurance Technologies Inc. [2], [15].

While this scheme may be suitable for a variety of process signatures, it is not difficult to imagine circumstances where the application of control limits would produce misleading results. Consider the scenario illustrated in Figure 2, where the desired bias force signature follows the form of the nominal signature, and departures from the nominal form constitute failure. The control limit mask does not capture these failures.

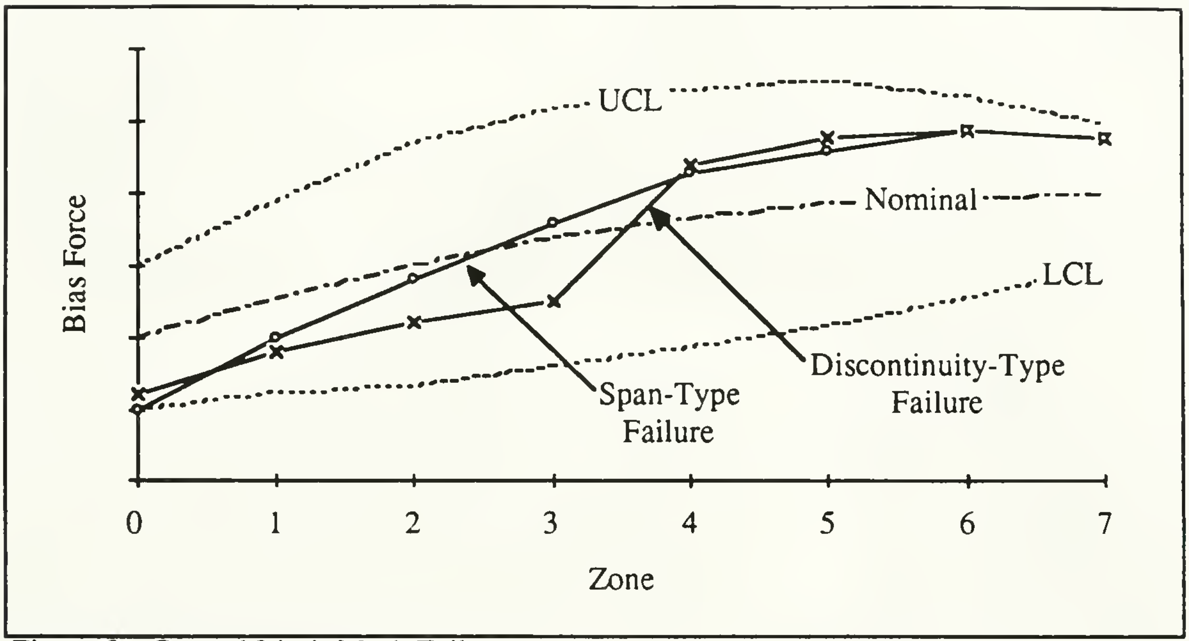

Figure 2 - Control Limit Mask Failure

2.2 Feature Extraction

We can circumvent the above difficulties through *feature extraction.* The purpose of feature extraction is to reduce the amount of data we must process without discarding useful information. Consider the bias force signature shown in Figure 3. In this case, two features are extracted from the original signature: the slope of the best fit line and the range of the signature's extreme points. By plotting the slope versus the range for many samples of hard disk assemblies, we arrive at a *feature space* scatter plot as shown in Figure 4. Three signature classes become apparent by the clustering of the data points. Class 1 in the figure corresponds to good signatures which follow the form of the nominal curve (both range and slope are low), Class 2 (higher range) corresponds to signatures with a discontinuity, and Class 3 (higher slope and range) corresponds to signatures which span the control limit range without a discontinuity.

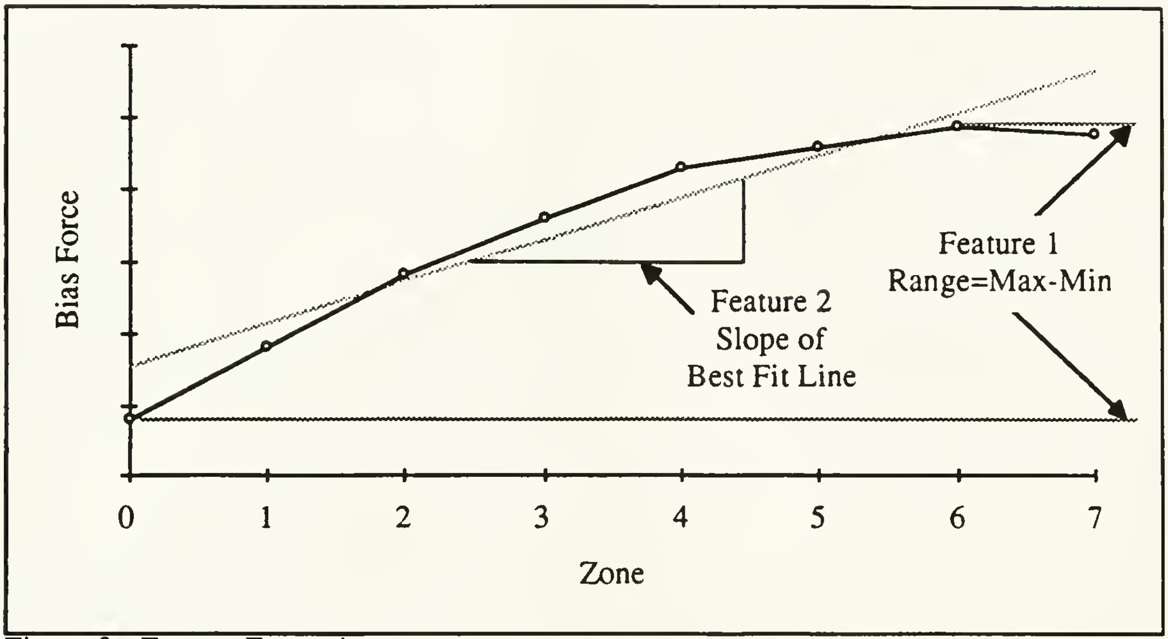

Figure 3 - Feature Extraction

In the bias force example, we extracted two features and the resulting feature space plot is two-dimensional. There are also two one-dimensional feature spaces associated with each individual feature. These one-dimensional spaces, shown opposite the axes in Figure 4, are simpler to interpret; however, the clustering of the classes we observed in the two-dimensional space may not be so apparent. As the number of features grows larger, the number of feature spaces increases tremendously. For a signature described by 23 features, as in the case of the data set used for this research, there are over 8 million distinct feature spaces. Before we address the question of which of these 8 million feature spaces is most effective for the given problem, a formulation for the quality decision function within the context of feature space must be examined.

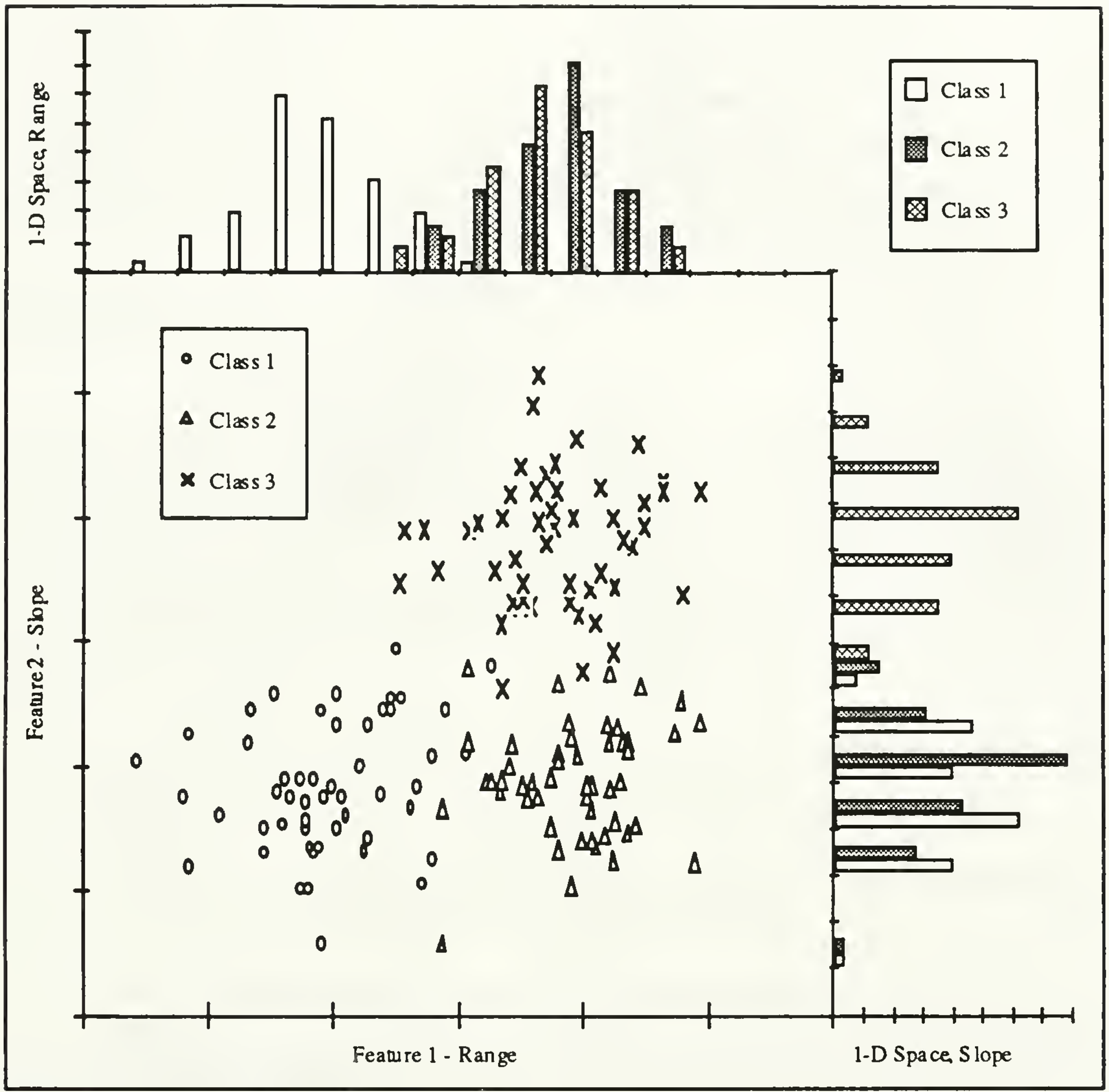

Figure 4 - Feature Space

2.3 Linear Statistical Classification

Classification within the context of feature space requires a mathematical formulation fundamentally different than the control limits of SPC. The objective in this case is to formulate a function whose output indicates the signature class. Note that unlike the accept/reject decision of SPC, we can now consider multiple classes of process output. In this section we describe the traditional linear statistical classifier.

2.3.1 Bayes Classifier

In geometric terms, a *linear statistical classifier* parameterizes each class of data with its mean and covariance[1]. Based on these parameters, an ellipse is constructed about each class of data, as shown in Figure 5. The common secant between two ellipses (i.e. two classes of data) then describes the linear decision function.

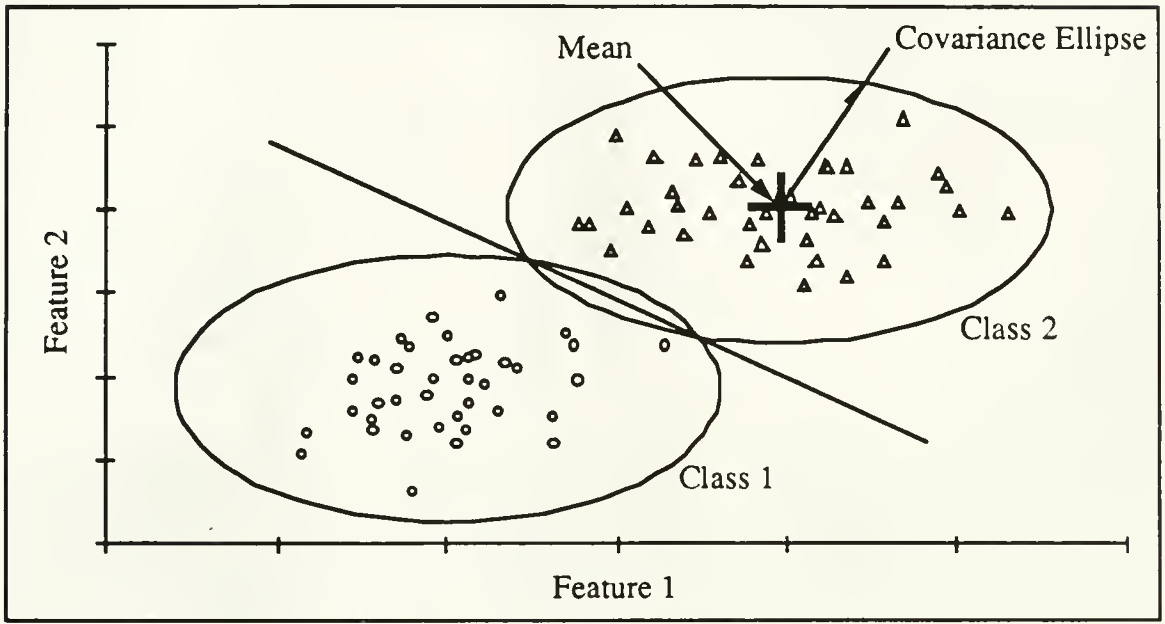

Figure 5 - Construction of a Linear Statistical Classifier (Bayes)

In mathematical terms, the construction of a linear statistical classifier (Bayesian) begins with the assumption that each class of data can be described by a normal probability density function [17]. Given this assumption, the likelihood that a given signature comes from a given class ω_i, is defined as follows:

$$p(x|\omega_i) = \frac{1}{(2\pi)^{n/2}|C_i|^{1/2}} \exp\left[-\frac{1}{2}(x-m_i)^T C_i^{-1}(x-m_i)\right] \quad i = 1 \ldots M$$

where:

x = Feature vector for a given signature.
ω_i = The ith class of data.
n = The number of features.
C_i = The covariance matrix for class i.
m_i = The mean vector for class i.
M = The number of classes.

[1] The geometric interpretation given here does not attempt to capture all the rich detail of statistical classification, rather its purpose is to provide an intuitive understanding of the essence of statistical classification.

For a one-dimensional feature vector, the above expression reduces to the following:

$$p(x|\omega_i) = \frac{1}{\sqrt{2\pi}\,\sigma_i}\exp\left[-\frac{1}{2}\left(\frac{x-m_i}{\sigma_i}\right)^2\right] \quad \mathrm{i} = 1 \ldots \mathrm{M}$$

where:

σ_i = The standard deviation of class i.

The classification decision for a signature characterized by a feature vector involves calculating the likelihood function for each of the possible classes and then selecting the class with the largest corresponding likelihood: Classify signature, $\boldsymbol{x}$, as class ω_i, if $p(x|\omega_i) > p(x|\omega_j)$ for i ≠ j). It is important to note that the application of a linear statistical classifier is restricted to those problems which conform to its underlying assumptions:

1. Normality: Each feature in the feature vector is normally distributed about some mean.
2. Linear Separability: Each class of output is linearly separable from all other classes.

In geometric terms, these assumptions can be restated as follows:

1. Normality: In drawing the ellipse around each class, a normal distribution is assumed.
2. Linear Separability: In drawing the line through the common secant, linear separability of classes is assumed.

2.3.2 Prediction of Error

We would like to quickly evaluate how much class-discerning information a set of signatures carries, without constructing and testing a complete linear statistical classifier. In predicting the success rate of the linear classifier we can also estimate the utility of each feature in the overall classifier performance. To do this efficiently, we do not examine every individual signature, rather we evaluate the data set as a whole and predict the *expected* contribution of each feature. In general, such prediction sacrifices accuracy for speed.

Our prediction of error begins with a measure of feature quality based on a set of features' ability to separate the classes called the *Mahalanobis Distance* which is defined as follows [17]:

$$FQ_{ij} = \left(m_i - m_j\right)^T C_{ij}^{-1} \left(m_i - m_j\right)$$

$$C_{ij} = \frac{1}{2}(C_i + C_i)$$

where:

FQ_{ij} = Mahalanobis distance between class i and class j (the feature quality with respect to class i and class j).
m_i = The mean vector for class i.
m_j = The mean vector for class j.
C_{ij} = The effective covariance matrix for class i and class j.
C_i = The covariance matrix for class i.
C_j = The covariance matrix for class j.

In one dimension (for one feature), the above expressions can be reduced to the following:

$$FQ_{ij} = \frac{\left(m_i - m_j\right)^2}{\sigma_{ij}^2}$$

$$\sigma_{ij} = \frac{1}{2}\left(\sigma_i + \sigma_j\right)$$

where:

σ_{ij} = The effective standard deviation for class i and class j.
σ_i = The standard deviation of class i.
σ_j = The standard deviation of class j.

In essence, the Mahalanobis distance is a feature quality score based on the signal to noise ratio between two classes of data. The "signal" being the distance between the classes in feature space, and the "noise" being the variation or spread of each class. From this feature quality score, a prediction of error can be calculated as follows:

$$P(e) = 1 - \operatorname{erf}\left(\sqrt{FQ/4}\right)$$

where:

$$\operatorname{erf}\left(x\right) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}y^2\right) dy$$

Note that in addition to the assumptions of a linear statistical classifier (normality and linear separability), the prediction of error as defined above is subject to a third assumption: *the covariance matrices for each class of data are approximately equal.* In calculating the feature quality score between two classes of data, we use an *effective* covariance matrix which is a simple averaging of the covariance matrices for each individual class. If the covariance matrices are not similar, then the effective covariance matrix does not accurately describe the spread of the class' respective distributions.

2.4 Neural Network Pattern Classification

All of the classification tools presented thus far have their basis in normal, linear statistical theory. As such, any application to which these tools can be reliably applied is restricted to normal and linear domains. However, many process measurements are not so well behaved. While these ill-behaved signatures may have significant content (class discerning information), traditional classification tools are not able to extract this content. We now turn to artificial neural networks as an alternative computational classification technique. Artificial neural networks make classification decisions by a fundamentally different mechanism so they are not encumbered by the rules of normal, linear statistics. There are many sources which can provide a brief introduction to artificial neural networks, see for example [8] or refer to [14] for more details.

Like the Bayes classifier, a neural network produces a mathematical function whose input is the process signature (the feature vector x) and whose output is the classification decision. The manner in which we arrive at the mathematical function is fundamentally different though. The formulation of a neural network also begins with a training set of signatures with known classifications. Each signature is applied to the input nodes of a network of simple processing elements. Numerical values propagate through the network via nonlinear, weighted connections, and eventually result in a classification decision at the output nodes. In all likelihood, this decision will be incorrect (the output values will not correspond to the known classifications) because the weights within the network structure are initially randomized.

At this point, a learning algorithm steps in and compares the actual network output with the desired output. Then based on the error, the weights within the network structure are modified. After many iterations, the network eventually 'learns' to distinguish among signatures from the various classes. Note that in formulating a decision function with a neural network, no assumptions are made regarding the underlying distribution of the data or linear separability. Instead, the motivating force behind the learning algorithm is the minimization of error[2]. As such, a neural network will attempt to create the line or surface which most effectively separates the classes.

After the network has been trained, each new signature for classification, characterized by a feature vector *x*, is simply applied to the input nodes of the network. The output values determine which class the signature best matches. The classification decision function therefore is the non-linear mapping determined in the network training stage.

2.4.1 The Perceptron

The fundamental processing element in the neural network classifier used for this study is the perceptron first introduced by Frank Rosenblatt [12]. A perceptron calculates the weighted sum of its inputs and passes the result through a non-linear thresholding function as shown in Figure 6. The thresholding function shown in the figure is a simple signum function. Some of the other common threshold functions which are used in neural networks include the hyperbolic tangent and the sigmoid. It is the non-linear threshold function which allows a neural network to extend the reach of pattern classification into the domain of generalized non-linear functions.

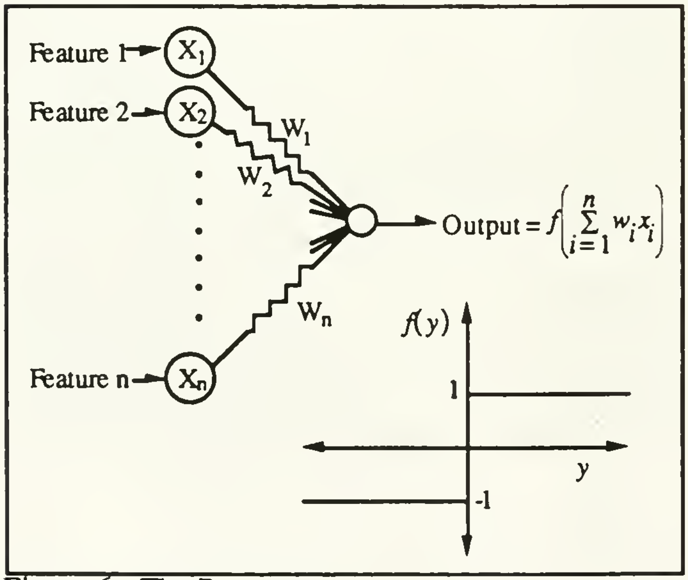

Figure 6 - The Perceptron

[2] Note that neural networks can be confused by the presence of local minima. One solution to this problem is to simply train multiple networks with different intial starting weights. Alternatively, gaussian noise can be added to the neural network inputs which will serve to "bump" the solution out of local minima [9].

2.4.2 Multi-Layer Feedforward Neural Networks

The neural network structure used for this study is a multi-layer feedforward neural network which utilizes the back propagation learning algorithm. A general schematic of this network is shown in Figure 7. The *input layer* has one node for each feature extracted from the raw signature. Succeeding layers of the network consist of one or more perceptron nodes. The *output layer*, where the classification decision emerges, also consists of one or more perceptron nodes. The actual number of output nodes depends on the number of possible classes in the data set as well as the manner in which we desire to code the different classes. For instance, only one output node is needed for a two class problem where an output of positive one corresponds to the first class and an output of negative one corresponds to the second class. In problems which involve a larger number of classes, we could assign one output node to each possible class or we could encode each class as a binary number thereby reducing the number of output nodes needed to identify each distinct class. The layers in between the input and output layers are referred to as *hidden layers*. The lines between the various nodes in the network represent the weighted connections through which the processing elements communicate.[3]

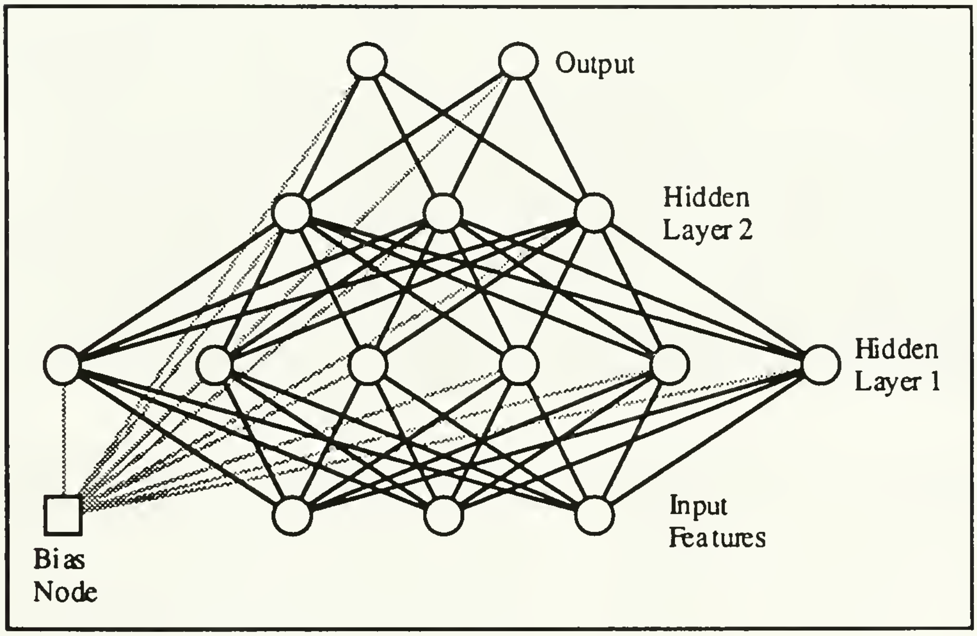

Figure 7 - The Multi-Layer Feedforward Neural Network

[3] The bias node in Figure 7, acts like an input layer node except that it is held at a constant value. The bias node allows the neural network to shift a decision function away from the origin in feature space.

The neural network structure described above, in essence, represents a complex non-linear function. The learning algorithm adjusts the parameters of the non-linear function until the classification error is minimized. It is important to note that, given a sufficiently complex topology, a neural network could eventually learn to correctly classify with zero errors. In general, this is not a desirable result as the neural network is functioning as a look up table (memorizing) instead of a generalized classifier. Consider an analogy where we wish to fit a curve to a set of ten data points. We could fit a tenth order polynomial so that each of the ten points falls on the curve and then extrapolate the curve to predict the value of succeeding points. In general, the tenth order polynomial will not accurately capture the trend in the data. The usual course of action in this case is to fit a curve to the ten points using a much lower order polynomial in hopes of capturing the general trend of the data as opposed to an exact representation of the first ten points. The same principle applies to neural networks, excessive complexity in the network structure prevents the network from making generalized decisions. Several authors offer guidelines on the selection of network structure including [14], [6], and [10].

2.4.3 Neural Networks in the Presence of Non-Linearity and Non-Normality

Figure 8 displays a hypothetical situation where a non-linear decision function is required. The straight line represents the linear decision function of a statistical classifier and the non-linear separation represents the decision function which a neural network would attempt to find. A non-normal case is illustrated in Figures 9 and 10. Note that the outliers in the distributions invalidate the assumption of normality. A statistical classifier constructs a decision function at the intersection of the normal density function for each class. In contrast, a neural network avoids these difficulties by ignoring the assumptions and instead moves the decision function with each iteration of the learning process until the error is minimized.

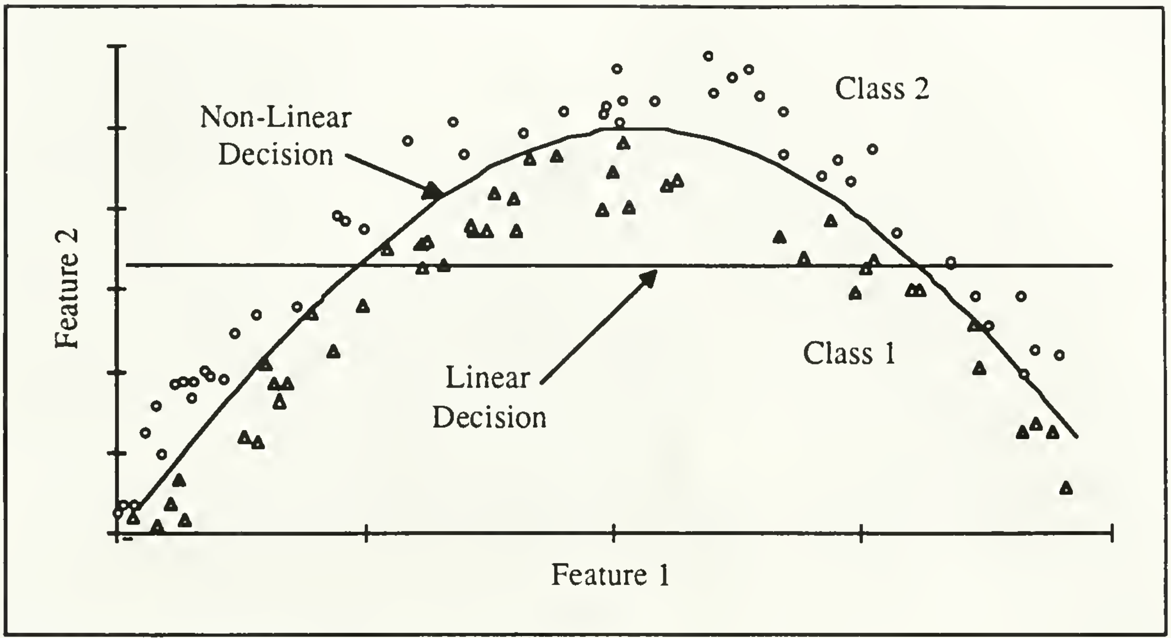

Figure 8 - Non-Linearity in the Feature Space

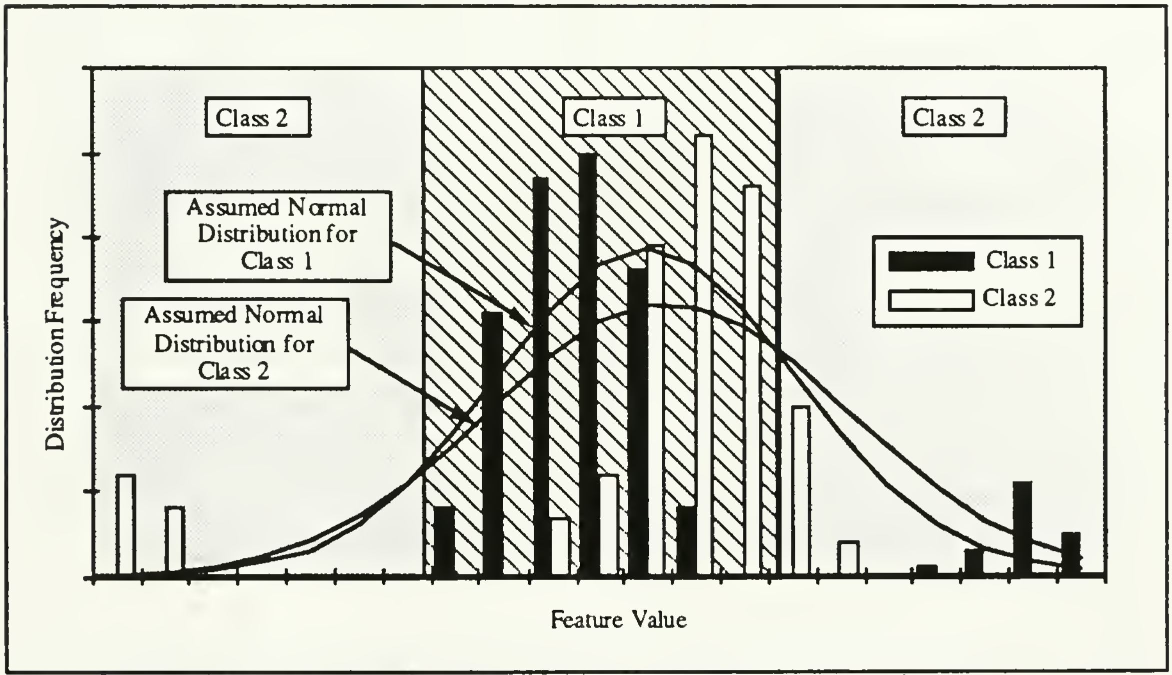

Figure 9 - Non-Normality and Statistical Classification Decision Regions

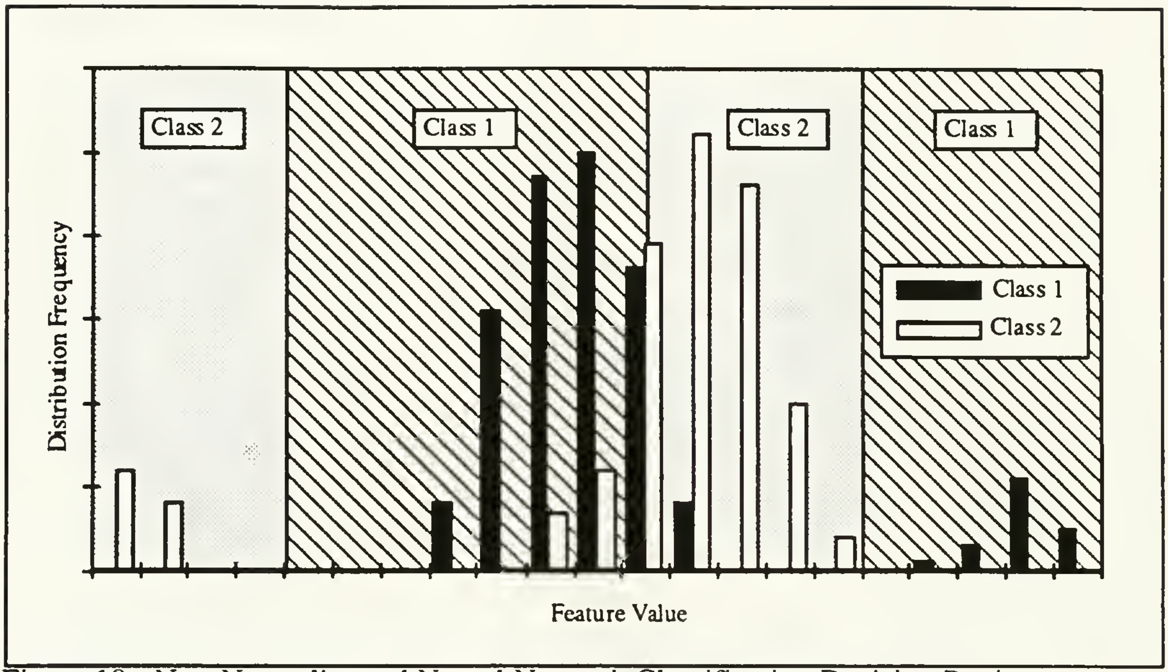

Figure 10 - Non-Normality and Neural Network Classification Decision Regions

2.5 Feature Selection

Given a perfect understanding of a process and its corresponding signatures, we could theoretically parameterize the signature with a finite set of descriptive features without discarding signature content. In reality, feature extraction may involve a significant amount of 'guess work' in proposing features as we must contend with a rudimentary understanding of the problem. It is often necessary to parameterize the signature with an abundance of features in the hopes that a handful will yield useful information. The difficulty associated with this approach is that the remaining features which do not add useful information add complexity to the classification problem. Clearly, if our goal is the development of an on-line process control tool, the useless features should be eliminated.

Feature selection is the process by which we eliminate those features which do not contribute toward the goal of discriminating among the different classes of output. Four different heuristic algorithms were investigated during the course of this research:

1. One-Dimensional Prediction of Error
2. One-Dimensional Statistical Classification
3. One-Dimensional Neural Network Classification
4. Multi-Dimensional Neural Network First Layer Weights

A description of each of these selection algorithms follows, along with their respective advantages and drawbacks.

2.5.1 One-Dimensional Prediction of Error

The one-dimensional prediction of error scheme, unlike the other schemes, is limited to two-class problems.[4] Within the context of a two-class problem, the utility of each feature is examined individually through normal linear statistics according to the following algorithm:

1. Calculate the mean and standard deviation for each feature within each class.
2. Calculate the one-dimensional Mahalanobis distance between classes for each feature.
3. Calculate a prediction of misclassification error for each feature.
4. Rank the features based on the prediction of misclassification error.

The predictions which result from the algorithm above represent the misclassification error that would be incurred if the given feature was the only information available (interaction among features is ignored). In the event that the data set conforms to the three assumptions associated with a prediction of error, normality, linear separability, and similar covariance among classes, this scheme provides the least costly (in terms of computation time) evaluation of the feature set.

2.5.2 One-Dimensional Statistical Classification

With one-dimensional statistical classification, like one-dimensional prediction of error, each feature is examined individually but in this case the data are applied to a statistical classifier. The steps in the one-dimensional statistical classification algorithm are outlined below:

1. Calculate the mean and standard deviation for each feature within each class.
2. Apply each signature in the data set to the one-dimensional likelihood function for each class.
3. Classify each signature according to magnitude of the likelihood function for each class.
4. Determine the misclassification error by comparing the computed classifications against the known signature classifications.
5. Rank the features based on the misclassification error for each feature.

[4] Recall that prediction of error, as defined earlier, is based on the Mahalanobis distance between *two* classes of data.

The one-dimensional statistical classification scheme has several advantages over the one-dimensional prediction of error scheme, including:

1. Statistical classification is not restricted to two class problems.
2. The assumption of similar covariance among classes is eliminated.
3. The actual data are applied to the classifier which can help to expose deviation from normal, linear statistics.

The first two advantages are self-explanatory but the third deserves closer examination. The prediction of error scheme employs the assumptions of normal, linear statistics and then decides the misclassification error based on what the data *would* look like if it conformed to these assumptions. The statistical classifier scheme employs the same assumptions but instead of relying on those assumptions, they are tested against the actual data. Deviation from the assumptions of normal, linear statistics can then be exposed in the form of higher misclassification rates. The prediction of error scheme, though less costly in terms of time and computing resources, is more easily fooled as it makes only indirect use of the data in arriving at its conclusion.

2.5.3 One-Dimensional Neural Network Classification

With the one-dimensional neural network scheme, like the preceding schemes, the utility of each feature is examined in isolation from the other features. This scheme is similar to the classification scheme with the exception that a neural network classifier is used in place of normal, linear statistics. The algorithm is given below:

1. Train a one-input neural network for each feature.
2. Determine the misclassification error of each one-input neural network by comparing the actual output against the desired output.
3. Rank the features based on the misclassification error for each feature.

The neural network structure which was used for this purpose is defined as follows:

1. Back propagation learning algorithm.
2. Hyperbolic tangent thresholding function.
3. One input node for the feature being evaluated.
4. Four perceptron nodes in the first hidden layer.
5. Two perceptron nodes in the second hidden layer.
6. The number of output nodes depends on the how many classes of output exist. For a two-class problem, one output node is sufficient.

The primary advantage of a neural network classifier is its independence from normality and linearity issues. Minimization of error underlies the iterative learning algorithm of neural networks, with no assumption of normality. Furthermore, the thresholding function in each processing element allows the neural network to twist and contort the decision function in a manner which is not available through traditional statistics.

2.5.4 First Layer Weights in Multi-Dimensional Neural Network Classification

Each selection scheme presented thus far, including one-dimensional neural network classification, evaluates the utility of each feature in isolation from the other features. Interactions between features are ignored. In contrast, the first layer weight scheme takes a global view of the full feature set. The first layer weight algorithm is illustrated below:

1. Train a multi-dimensional neural network using the full feature set as the input to the network.
2. Determine the sum of the absolute values of the first layer weights associated with each feature.
3. Rank the features based on the sum of the first layer weights.

The neural network structure which was used for the first layer weight scheme is defined as follows:

1. Back propagation learning algorithm.
2. Hyperbolic tangent thresholding function.
3. N input nodes (one for each feature).
4. 2N perceptron nodes in the first hidden layer.
5. N perceptron nodes in the second hidden layer.
6. The number of output nodes depends on the how many classes of output exist. For a two class problem, one output node is sufficient.

The significance of the first layer weights in a multi-dimensional neural network can best be explained by reexamining the general schematic of a neural network as shown in Figure 11. In this schematic, all of the lines representing the weighted connections between the input layer and the first hidden layer have been removed except for those associated with the feature of interest. With the first layer weight scheme, the importance of a given feature is measured by the sum of the absolute values of the first layer weights shown in bold. In the case that all of these weights are zero, or nearly zero, the feature is effectively filtered from the classification decision. Conversely, if these weights are large, the feature will have a greater affect on the classification decision thus the rationale behind the first layer weight scheme. Like the one-dimensional neural network classification scheme, the first layer weight scheme is able to recognize non-normality and non-linearity with the added advantage of recognizing interaction among features.

Note that the first layer weight scheme does not explicitly address the presence of redundant features though we would expect the first layer weights associated with redundant features to be similar. While redundant features will not hinder the accuracy of the network, elimination of these features would further reduce its computational complexity. To further simplify the network, we must examine the weights associated with each feature and identify any similarities. In order to verify redundancy, we must then eliminate all but one of the redundant features and retrain the network. If the misclassification error is unchanged, redundant features have been successfully identified and removed.

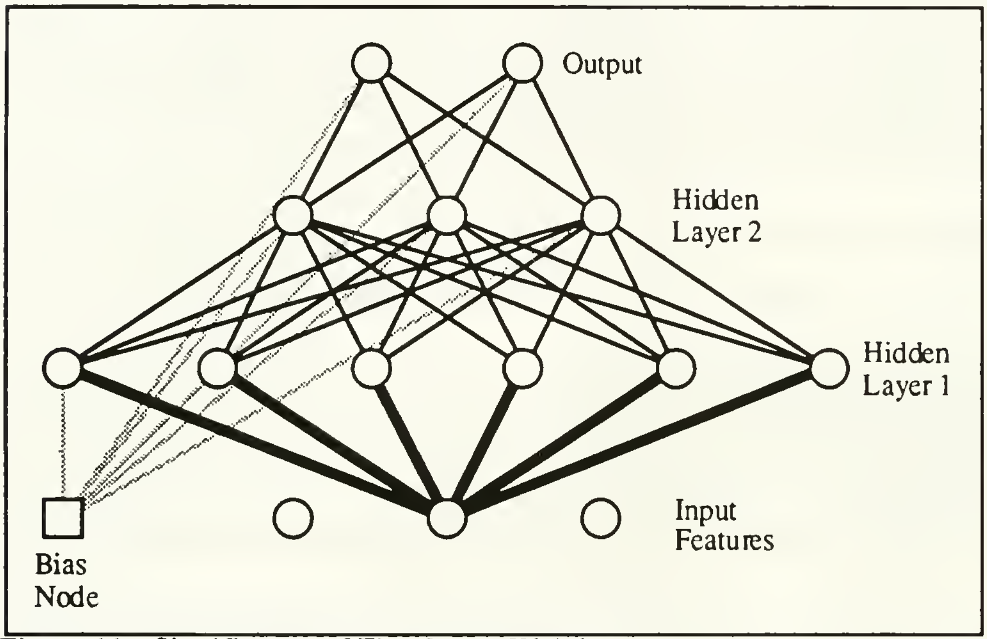

Figure 11 - Significance of First Layer Weights

2.5.5 Selecting the Right Features

Each of the feature selection schemes described above simply ranks the utility of the features. The question of how many features to employ in an actual classifier remains. One solution to this problem is to simply test a succession of classifiers, adding additional features with each test. As soon as the performance of the classifier does not improve significantly with the addition of new features, feature selection is complete. Alternatively, the feature selection scores can be examined and the feature cutoff can be placed where there is a significant drop in the feature selection score. For this research, we were interested in directly comparing the four feature selection schemes' abilities to find the beat features, so an arbitrary cutoff of the best five features for each scheme was used in our comparison.

2.6 Choosing a Classifier

Once an appropriate subset of features has been selected, the task of constructing a classifier remains. In the event that a given data set conforms to the assumptions of normality and linear separability, there is no need to expend additional time and energy developing a neural network classifier. A neural network and a statistical classifier will arrive at the same decision function for well behaved data sets. In the event that a given data set does *not* conform to the assumptions of normality and linear separability, a neural network is a highly effective solution. While the amount of time required to *develop* a neural network is significantly greater than a statistical classifier, the cycle time of the actual classifier is not necessarily significantly greater. If there is any question regarding the integrity of the data with respect to the assumptions of normal, linear statistics, a neural network is worth investigating.

3 Automated Solder Joint Inspection

The data set which was the primary vehicle of research for this paper came from an automated solder joint inspection project at Digital Equipment Corporation. The goal of automation in this case is the replacement of manual visual inspection methods for surface mount electronic components (SMD technology). With surface mount technology, printed circuit boards are screened with solder paste and then the components are placed onto the board using automated equipment. The boards are then heated to re-flow the solder creating the lead bonds. Although surface mount technology allows the electronics industry to continue reducing the size of components, new problems have been introduced. In particular, very high lead density causes manual visual inspection methods to become less and less effective (as measured by higher misclassification error). Inefficiency drives the need for redundant inspect and test which, in turn, provides the impetus for automated inspection methods.

3.1 Modes of Solder Joint Failure

Among common problems associated with the SMD assembly process is the application of incorrect solder paste volume. Insufficient or excess solder conditions have been correlated with premature failure of the solder bond due to thermal cycling of the circuit. Electronic testing of the circuit boards can detect more severe classes of defects such as short and open circuits, but cannot guarantee that the board will remain defect free over the life of the component.

The automated solder joint inspection system at Digital [4] is illustrated in Figure 12. A laser is directed at a single component lead while two infrared camera sensors monitor the temperature of the solder joint as it is heated and then cools. These signals are digitized and stored for off-line analysis. Eventually these data would be utilized on-line with the results of our signature analysis scheme.

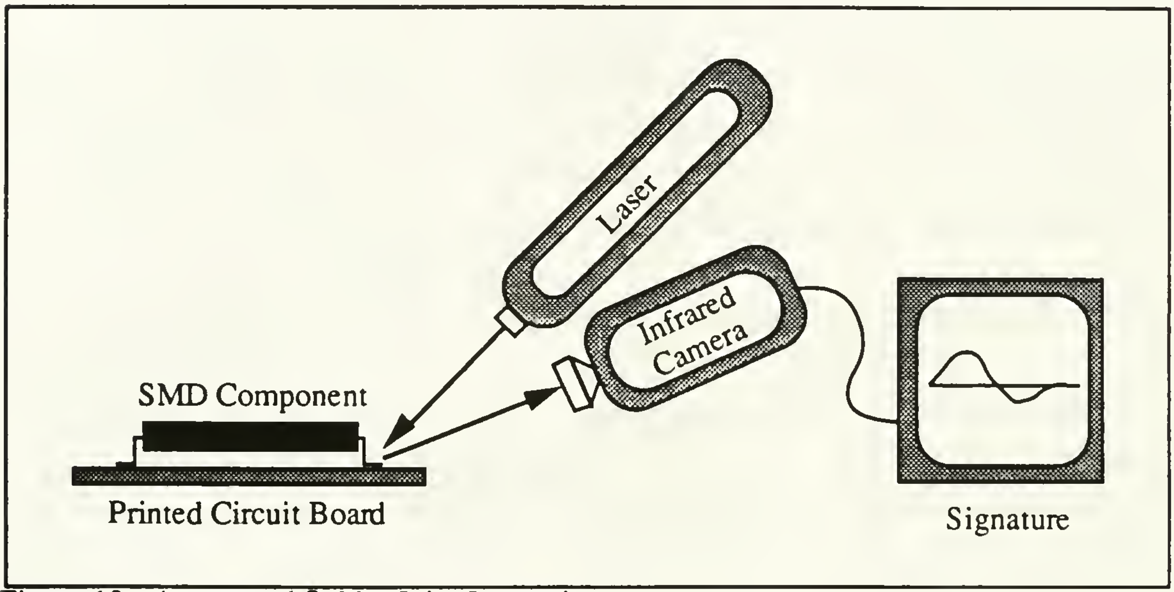

Figure 12 - Automated Solder Joint Inspection

These data represent the time-derivative of the lead temperatures as observed by the two different cameras. Each camera extracts energy in a different IR wavelength band. The result is a pair of thermal signatures for the solder joint where each signature consists of a discrete time series of 500 data points as shown in Figure 13. The physics underlying the success of this inspection method involves the heat transfer characteristics of the solder joints. The hypothesis is as follows: If the lead is well bonded, the board will conduct heat quickly away from the lead. If the connection is insufficient, the heat transfer will be slower, and the lead will heat excessively. If the solder mass is excessive, the joint will also retain more heat, and will heat up slower. Accurate physical models of this phenomenon have not been proven.

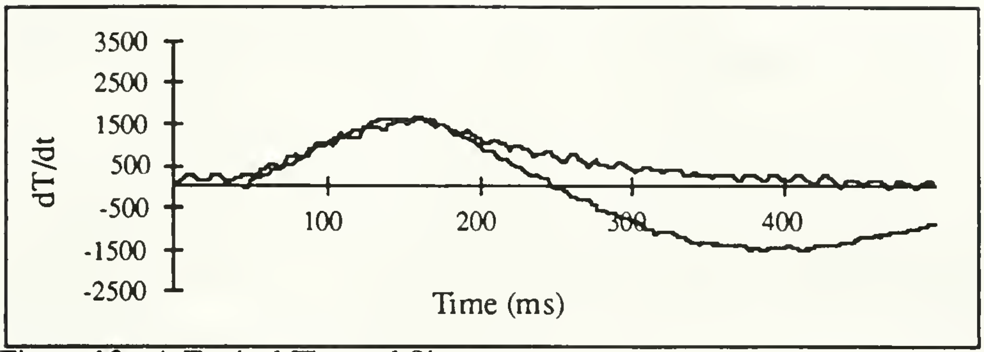

Figure 13 - A Typical Thermal Signature

3.2 Data Collection

The data for this study were collected by specially fabricating a series of circuit boards under the three solder paste conditions: excess, normal, and insufficient[5]. SMD components were placed onto the solder pads, and then the boards were heated in the conventional re-flow process. Finally, each of the solder joints was inspected using the laser/infrared system described above. The solder joint data set consisted of 3510 signatures, 1510 from solder joints with excess solder, 1688 from solder joints with insufficient solder, and 312 from solder joints with normal solder. Given three classes of output, the signature analysis methodology can be applied to four distinct classification problems, three problems associated with distinct class pairs and one problem associated with all three classes. From a manufacturing standpoint, a classifier which can distinguish between insufficient and good solder conditions but ignores the existence of excess solder conditions is of marginal utility. From a research standpoint however, each of these four problems can be employed in our exploration of the principles of the signature analysis methodology. As such, all four classification decisions (excess-good, excess-insufficient, insufficient-good, and excess-insufficient-good) were investigated during the course of this research.

3.3 Feature Extraction

For each signature, 23 numerical features were extracted from the raw data. In developing our list of candidate features, we drew first from our physical understanding of the process to define appropriate features. As an example, heat transfer theory

[5] These data were part of a larger experiment at Digital comparing the performance of several automated solder joint inspection schemes and also other defect types which are not included in our data set.

suggests that the area under the temperature derivative curve up to the first zero crossing (see Figure 13) should be proportional to the total amount of heat absorbed by the solder joint during the measurement. The remaining features are a mixture of statistical descriptions and empirical relationships. For this example, feature extraction was conducted on an intuitive basis though more interesting alternatives are available. Of particular interest is a method of feature extraction based on wavelet decomposition. For more information on this alternative see [11].

3.4 Feature Selection

Each of the four feature selection schemes described above was investigated in order to determine the success of each scheme and to test our understanding of the behavior of each scheme. The one-dimensional prediction of error scheme was applied to only the two-class problems associated with distinct class pairs (excess-good, excess-insufficient, and insufficient-good) whereas the remaining schemes were applied to all four classification problems. Each feature selection scheme produced a feature ranking for each applicable classification problem. An evaluation of these rankings was carried out by selecting the top five features in each ranking and then calculating a prediction of error, statistical misclassification error, and neural network misclassification error for each five feature subset.

3.5 Signature Analysis Results

The results of the signature analysis computations discussed in preceding sections of this paper are presented in Table 2. The four column groups in this table correspond to the four distinct classification problems while the rows correspond to the misclassification errors for individual features, feature subsets, and the full feature set. For example, the results from the excess-insufficient problem are in the first column group: The prediction of error for the feature Speak_1, acting alone, was 32% while the statistical and neural network classifiers had misclassification errors of 20% and 18% respectively. For the same problem, the first layer weight scheme produced a subset of five features which resulted in a prediction of 17% error and a statistical and neural network misclassification errors of 17% and 14% respectively. The next few sections of this paper are devoted to an explanation of the characteristics and significance of the results table.

Feature		Excess-Insufficient			Insufficient-Good			Excess-Good			Excess-Good Insufficient	
Description		P(e)	Class	NNet	P(e)	Class	NNet	P(e)	Class	NNet	Class	NNet
1	Speak 1	32%	20%	18%	26%	18%	17%	50%	67%	DNC	51%	32%
2	Speak 2	37%	21%	17%	26%	17%	DNC	48%	66%	DNC	52%	30%
3	Peak Ratio	48%	44%	DNC	49%	61%	DNC	49%	64%	DNC	76%	DNC
4	Peak Time 1	47%	48%	DNC	30%	29%	26%	43%	62%	DNC	58%	DNC
5	Peak Time 2	46%	49%	DNC	30%	31%	DNC	42%	58%	DNC	59%	DNC
6	Mean 1	42%	48%	42%	43%	65%	DNC	45%	53%	DNC	71%	53%
7	Mean 2	28%	21%	20%	28%	20%	16%	49%	47%	DNC	39%	35%
8	Variance 1	19%	16%	16%	20%	18%	15%	48%	43%	DNC	36%	31%
9	Variance 2	43%	50%	DNC	39%	55%	DNC	48%	66%	DNC	80%	DNC
10	Sig. RMS 1	20%	16%	16%	21%	17%	13%	48%	50%	DNC	39%	29%
11	Sig. RMS 2	30%	19%	18%	27%	19%	DNC	49%	62%	DNC	52%	31%
12	Zero Xing 1	44%	50%	DNC	45%	62%	DNC	47%	53%	DNC	73%	DNC
13	Zero Xing 2	43%	41%	41%	29%	37%	DNC	37%	50%	DNC	64%	50%
14	Area Above 1	27%	17%	18%	27%	16%	15%	50%	40%	DNC	34%	31%
15	Area Above 2	27%	22%	21%	27%	20%	18%	50%	55%	DNC	48%	35%
16	Sig Min 1	39%	30%	18%	24%	23%	15%	49%	67%	DNC	56%	DNC
17	Sig Min 2	49%	50%	DNC	44%	48%	DNC	46%	60%	DNC	69%	DNC
18	Min Time 1	49%	45%	DNC	45%	35%	DNC	48%	65%	DNC	60%	DNC
19	Min Time 2	38%	39%	39%	41%	28%	DNC	47%	56%	DNC	49%	48%
20	Peak of Diff	50%	52%	DNC	43%	46%	24%	46%	69%	DNC	70%	DNC
21	Min of Diff	39%	32%	DNC	26%	26%	DNC	49%	67%	DNC	58%	DNC
22	Zero Xing Diff	50%	49%	DNC	50%	66%	DNC	50%	67%	DNC	79%	DNC
24	Erms2	45%	45%	DNC	47%	71%	DNC	46%	67%	DNC	81%	DNC
Best 5 Features												
	1-D Prediction of Error	17%	22%	17%	19%	38%	15%	34%	54%	27%	N/A	N/A
	1-D Statistical Classification	17%	18%	14%	16%	25%	12%	46%	57%	DNC	59%	28%
	1-D Neural Net Classification	17%	18%	14%	18%	33%	12%	N/A	N/A	N/A	63%	29%
	First Layer Weights	17%	17%	14%	15%	26%	9%	33%	44%	26%	57%	26%
Full Feature Set		14%	21%	13%	9%	24%	9%	24%	56%	22%	52%	27%

Table 2 - Signature Analysis Classification Results

3.5.1 Generation of Data Sets for Analysis

In the formulation and training of the various classification tools, random subsets of the full data set were used. These subsets consisted of 500 excess signatures, 500 insufficient signatures, and 200 good signatures. For the statistical tools (prediction of error and statistical classification), one subset was used for training and testing. For neural network classification, two subsets were drawn from the full data set, one for training and one for testing. The reason for using two different subsets for training and testing was suggested earlier: improper selection of the neural network structure can cause the neural network to act as a look up table instead of a generalized decision function. By training the neural network with one data set and testing it with another, we can verify the formulation of a generalized decision function.

3.5.2 Explanation of Anomalies in Results Table

In several instances, neural network classification was unsuccessful as the neural network was not able to converge on a decision function. The DNC entries in Table 2 denote instances where the neural network did not converge. Whenever the inputs to a neural network do not contain sufficient content (class discerning information), a neural network learns to select whichever class it saw most frequently during training regardless of input. For the excess-good classification problem, the neural networks were trained with 500 excess signatures and 200 good signatures. In this case, a neural network which did not converge learned to classify every signature as excess because it was presented with more excess signatures during training. In actuality, this corresponds to a misclassification error rate of 29% (200/700), however, we enter DNC in the results table. Note that in almost every case where the neural network could not converge, such as the feature Speak_1 in the excess-good problem, the statistical prediction of error (50%) and statistical classification error (67%) were even less useful.

The N/A entries in Table 2 denote two situations where misclassification error could not be calculated. In the first situation, the individual neural net feature selection scheme failed for the excess-good classification problem because none of the individual neural networks was able to converge on a decision function. In the second instance, the prediction of error scheme failed for the excess-insufficient-good classification problem because prediction of error was not defined for the three class problem. As a result, these schemes could not offer a ranking of features.

3.5.3 Feature Content and Signature Content

For the excess-insufficient problem, the Sig Min_2 feature produced prediction, statistical, and neural network misclassification errors of 49%, 50%, and DNC, respectively. Clearly, this feature cannot offer any more than a random guess as to the appropriate classification. This is only one of many examples where the various classification tools were not able to perform any better than the toss of a coin. These high misclassification errors occur due to the lack of signature content. The greatest challenge in the development of a process monitoring system involves what in this paper we call signature content, feature quality, or simply observability. In the extreme case, if we choose an inappropriate sensor measurement which does not contain the information needed to predict quality, then an entirely different measurement scheme must be chosen. The techniques presented in this paper cannot remedy this situation as demonstrated by the excess-good classification problem.

3.5.4 Neural Network Classification vs. Statistical Tools

An interesting scenario arises though when a neural network classifier is able perform with reasonable accuracy while the statistical tools fail. In these cases, the neural network has found a better non-linear or non-normal decision function which was invisible to traditional classification tools. As an example, in the insufficient-good problem, the first layer weight scheme produced a subset of features which generated a 9% error rate with the neural network and a 26% error rate with the statistical classifier. Given the same information, the neural network was able to perform significantly better.

Throughout the table of results, the neural network misclassification error is lower than both the statistical prediction of error and the statistical misclassification error except in a few instances where the various error rates are similar to within a one or two percent noise factor[6]. This is a direct result of the neural networks iterative learning process and non-linear decision functions. While statistical tools are confounded by ill-behaved data, neural networks exploit these differences. Conversely, when the assumptions of normality and linear separability are fairly accurate, the performance of the statistical tools and neural networks are similar. If the assumptions of normal, linear statistics hold

[6] Recall that random subsets of data were drawn from the full data set for the development of each tool. The noise factor is the result of the slight differences in each random subset of data.

true, neural networks and statistical tools will arrive at the same decision function - there is no need to implement a neural network classifier.

3.5.5 Full Feature Set Outperforms Feature Subsets

The neural networks which employed the full feature set invariably performed better than any individual feature and better than any subset of features. This suggests that neural networks are able to employ interactions among features thereby generating a decision which is better than the sum of its parts yet is still able to effectively filter those features which do not add useful information to the problem. As an example, consider the insufficient-good problem. In this case, the best individual feature (Sig. RMS_1) produced a 13% misclassification error whereas neural network classification with the full feature set produced a 9% misclassification error. Note that this was accomplished even though more than half of the features had no signature content at all (DNC). In the event that the feature extraction process entails a significant amount of guesswork, we can be assured that a neural network will filter poor guesses. Note that this is not the case with the statistical tools. In the excess-insufficient problem, the best individual feature (Sig. RMS_1) produced a statistical misclassification error of 16% whereas the full feature set could do no better than 21%. Those features which lack content, while filtered out of the neural network decision function, serve to confuse statistical decision functions.

3.5.6 Evaluation of Feature Selection Schemes

In evaluating the various feature selection schemes, we are most interested in those schemes which selected features with the most content. In all four classification problems, the first layer weight scheme produced the lowest neural network misclassification error. In fact, the first layer weight scheme produced a subset of features which performed nearly as well as the neural network classifiers which employed the full feature set. Consider the insufficient-good problem, the first layer weight scheme produced a subset of features with a neural network misclassification error of 9% which was the same error produced by the full feature set. The remaining tools and schemes produced misclassification errors which ranged from 12% to 38%. By eliminating the useless features we gain significant computational advantages without sacrificing classification accuracy.

3.5.7 Normality Issues

In order to evaluate the effect of normality on the various classification tools, normality scores were calculated for each feature in each class. These scores are presented along with the results from the excess-insufficient classification problem in Table 3. The normality score calculation consists of the Pearson Product-Moment correlation between the given variable (a particular feature from a particular class) and the corresponding normal scores (i.e. standardized z-score).

In the event that a given data set conforms to the assumptions of normal linear statistics, we would expect that the prediction of error, statistical misclassification error, and neural network misclassification error would reflect similar results [5]. The results shown in Table 3 for the features Mean_2, Sig. RMS_1, Zero Xing_2, and Area Above_2 (features 7,10,13, and 15 respectively) tend to support this hypothesis. Conversely, we might expect that poor normality scores would produce results where the statistical errors were significantly higher than neural network errors. In fact, the results do not corroborate this hypothesis. Consider the feature Sig. RMS_2 (feature 11). In this case, the normality scores for the excess and insufficient classes are .82 and .68 respectively yet the statistical and neural network misclassification errors are 19% and 18% respectively. This can be explained by noting that poor normality implies that the normal probability model does not accurately represent the data and any subsequent decision function *may or may not* separate the data. Low statistical misclassification error under these circumstances cannot be taken as reliable.

Feature		Excess-Insufficient			Normality	
Description		P(e)	Class	NNet	E	I
1	Speak 1	32%	20%	18%	0.70	0.90
2	Speak 2	37%	21%	17%	0.60	0.55
3	Peak Ratio	48%	44%	DNC	0.89	0.87
4	Peak Time 1	47%	48%	DNC	0.40	0.91
5	Peak Time 2	46%	49%	DNC	0.43	0.54
6	Mean 1	42%	48%	42%	0.67	0.60
7	Mean 2	28%	21%	20%	0.95	0.95
8	Variance 1	19%	16%	16%	0.95	0.86
9	Variance 2	43%	50%	DNC	0.32	0.25
10	Sig. RMS 1	20%	16%	16%	0.97	0.91
11	Sig. RMS 2	30%	19%	18%	0.82	0.68
12	Zero Xing 1	44%	50%	DNC	0.72	0.70
13	Zero Xing 2	43%	41%	41%	0.95	0.98
14	Area Above 1	27%	17%	18%	0.99	0.89
15	Area Above 2	27%	22%	21%	0.98	0.96
16	Sig Min 1	39%	30%	18%	0.51	0.78
17	Sig Min 2	49%	50%	DNC	0.30	0.26
18	Min Time 1	49%	45%	DNC	0.79	0.90
19	Min Time 2	38%	39%	39%	0.81	0.60
20	Peak of Diff	50%	52%	DNC	0.33	0.50
21	Min of Diff	39%	32%	DNC	0.46	0.59
22	Zero Xing Diff	50%	49%	DNC	0.75	0.71
23	Erms2	45%	45%	DNC	0.59	0.48
Feature Selection Schems						
1-D Prediction of Error		17%	22%	17%		
1-D Statistical Classification		17%	18%	14%		
1-D Neural Net Classification		17%	18%	14%		
First Layer Weights		17%	17%	14%		
Full Feature Set		14%	21%	13%		

Table 3 - Excess-Insufficient Classification and Normality

4 Conclusion

4.1 Summary of Contributions

The improvement of process control systems is of critical importance to manufacturing firms today. In pursuit of rapid improvement, we must make effective use of the wealth of process information now made available by on-line sensors and computation. The signature analysis methodology presented in this paper is offered as a formal approach to this challenge. The three-step analysis method (feature extraction, feature selection, and classification) accepts complex process signatures and their corresponding quality classifications as input and produces a relationship between the signatures and the quality of the process output.

In applying this method to a data set from an automated solder joint inspection system, artificial neural networks were found to offer significant performance advantages over traditional classification tools, yet this performance was achieved at the expense of significant computational resources. In order to reduce the computational cost of classification, we must reduce the complexity of the classification problem by employing only the most useful of the features. We have termed this the *feature selection problem*. Once again we can turn to the unique abilities of artificial neural networks. In particular, the first layer weights in the neural network topology were found to be effective in identifying which components of the input signature contain the most useful information. Using the first layer weights, we were able to identify a subset of features which offers classification rates comparable to the full feature set. Note that this result also suggests that a neural network which employs the full feature set is able to effectively filter those features which lack content.

Although we were able to demonstrate the advantage that neural networks have over traditional classification tools, the misclassification error remains unacceptably high for the example data set. This difficulty cannot be resolved due to the lack of quality content in the particular signatures. Our analysis suggests that either more or different information is needed to more accurately monitor the quality of the solder joint.

4.2 Our Vision for Manufacturing Process Signature Analysis

The signature analysis methodology employed in this paper, along with the specific feature selection and classification tools, represents only one component of a larger signature monitoring procedure which can be defined as follows:

Step One:	Establishment of quality metrics
Step Two:	Selection of process measurement sensors
Step Three:	Signature analysis (feature extraction, feature selection, and classification)
Step Four:	Implementation of on-line monitoring/control

Steps One and Two represent the inputs to the signature analysis methodology described in this paper. We must first establish quality metrics so that we can definitively classify each instance of the process output. Without accurate classifications, we cannot build a detailed relationship between the process output and the corresponding signatures. Second, we must decide how to install sensors for process measurements (type, quantity, and location of sensors). As demonstrated in this paper, signatures which lack content cannot contribute to an accurate classification decision. In general, effective sensor selection is dependent on an understanding of the process in question. Given that such understanding is often limited, we would expect sensor selection to be modified after signature analysis has begun. The third step of this procedure involves the application of the signature analysis tools presented in this paper: feature extraction, feature selection, and classification. If, given the application of these tools, we discover that the signatures lack content, we must reevaluate sensor selection and configuration. Finally, in step four, we apply the results of the preceding steps in the form of a signature monitoring system. We can also make the quality classification data available for on-line control, for operator intervention, or both.

4.3 Future Research Directions

The first research question we must address in developing a process monitoring system is simply, how can we definitively quantify the quality of the process output? In order to answer this question, we must develop the means by which we translate product specifications into quality metrics so that we can identify the quality of the process output without ambiguity. In essence, the measurement of quality metrics serve as the calibration standards by which we tune our classification tools. If the calibration standards are not accurate, the classification tools will not be accurate. Alternatively, we might try to build a relationship between the process signatures and the most significant process variables. This might entail a design-of-experiments approach where we vary the pertinent process variables (i.e. dull tool to sharp tool) under a variety of sensor configurations so that we can guarantee a complete yet concise view of the process variables in the data set used for training. Under this scheme, multiple classes of process output would be defined according to the variables used in the experiment. Note that in correlating process signatures to process variables, we are assuming a direct relationship between the process variables and the quality of the process output.

Once the training data are available we must still formulate a relationship between the process signatures and their corresponding classifications. As demonstrated in this paper, artificial neural networks offer many advantages over traditional statistical classification tools in this capacity. Further research in this area could result in more efficient network training algorithms, better feature extraction and selection methods, or more accurate decision functions. In fact, given the tremendous volume of research conducted in the field of neural networks, the investigation of new technologies with respect to the manufacturing signature analysis problem would by itself be a formidable task.

Finally, consideration must be given to the application of these new technologies in the manufacturing environment. This includes both the development of software tools which allow manufacturing engineers to exploit the technology as well as the implementation of neural network hardware which could facilitate real-time control of complex processes. The long term vision of our research consists of a signature analysis toolbox which includes a variety of sensors, signature analysis tools, and the ability to export the "blueprint" of a signature monitoring system. Ideally, a manufacturing engineer would be able to approach a new process with this toolbox and collect a variety of signatures and then subject these signatures to a variety of analysis tools. The result of this analysis would be the blueprint for an on-line process monitoring system.

4.4 Acknowledgment

Funding for this research was provided by Digital Equipment Corporation and by the MIT Leaders for Manufacturing Program.

References

[1] S. Bleha, C. Slivinsky, and B. Hussien. “Computer-Access Security Systems Using Keystroke Dynamics”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*. vol. 12, no. 12, December 1990, pp. 1217-1222.

[2] R.N. Boggs. “System Analyzes Processes to Eliminate Defective Parts.” *Design News*. May 18, 1992, pp. 91-92.

[3] G. Chryssolouris, P. Sheng, and F. von Alvensleben. "Process Control of Laser Grooving Using Acoustic Sensing" *Journal of Engineering for Industry* vol. 113, no. 3, August 1991.

[4] J.E. Goldsberry and C. Burke. “Characterization of the Two Color Infrared Bond Inspection Method.” Digital Equipment Corporation, Technical Resource Group, May 1990.

[5] F. Kanaya and S. Miyake. “Bayes Statistical Behavior and Valid Generalization of Pattern Classifying Neural Networks”, *IEEE Transactions on Neural Networks*. vol. 2, no. 4, July 1991, pp. 471-475.

[6] E.D. Karnin. “A Simple Procedure for Pruning Back-Propagation Trained Neural Networks”, *IEEE Transactions on Neural Networks*. vol. 1, no. 2, June 1990, pp. 239-242.

[7] S.Y. Liang and D.A. Dornfeld. “Characterization of Sheet Metal Forming Using Acoustic Emission”, *Journal of Engineering Materials and Technology*. vol. 112, no. 1, 1990, pp. 44-51.

[8] R.P. Lippmann. “An Introduction to Computing with Neural Nets”, *IEEE ASSP Magazine*. vol. no. April 1987, pp. 4-22.

[9] K. Matsuoka. “Noise Injection into Inputs in Back-Propagation Learning”, *IEEE Transactions on Systems, Man, and Cybernetics*. vol. 22, no. 3, June 1992, pp. 436-440.

[10] K.G. Mehrotra, C.K. Mohan, and S. Ranka. “Bounds on the Number of Samples Needed for Neural Learning”, *IEEE Transactions on Neural Networks*. vol. 2, no. 6, November 1991, pp. 548-558.

[11] O. Rioul and M. Vetterli. “Wavelets and Signal Processing.” *IEEE Signal Processing Magazine*. October 1991, pp. 14-38.

[12] F. Rosenblatt. *Principles of NeuroDynamics*. Washington, D.C., Spartan Books, 1951.

[13] J. Rotberg, E. Lenz, and S. Braun. “Vibration-Based Drill Wear Monitoring”, *Manufacturing Review*. vol. 3, no. 1, March 1990, pp. 60-65.

[14] R. Schalkoff. *Pattern Recognition - Statistical, Structural and Neural Approaches*. New York, John Wiley & Sons, Inc., 1992.

[15] Assurance Technologies Inc., 503D Highway 70 East, Garner, North Carolina, 27529.

[16] R. Teti and D. Dornfeld. "Modeling and Experimental Analysis of Acoustic Emission from Metal Cutting", *Journal of Engineering for Industry*. vol. 111, no. 3, August 1989, pp. 229-237.

[17] J.T. Tou and R.C. Gonzalez. *Pattern Recognition Principles*. Reading, Addison-Wesley Publishing Company, 1974.

[18] P. Trahanias and E. Skordalakis. "Syntactic Pattern Recognition of the ECG", *IEEE Transactions on Pattern Analysis and Machine Intelligence*. vol. 12, no. 7, July 1990, pp. 648-657.