1996

A neural network based classifier for the identification of simple finger motion

Michael Heinz
San Jose State University

A NEURAL NETWORK BASED CLASSIFIER FOR THE IDENTIFICATION OF SIMPLE FINGER MOTION

A Thesis

Presented to

The Faculty of the Department of Electrical Engineering

San Jose State University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

by

Michael Heinz

December 1996

UMI Number: 1382579

APPROVED FOR THE DEPARTMENT OF ELECTRICAL ENGINEERING

Dr. R. Benjamin Knapp

Dr. Richard Duda

Dr. Essam Marouf

APPROVED FOR THE UNIVERSITY

ABSTRACT

A NEURAL NETWORK BASED CLASSIFIER FOR THE IDENTIFICATION OF SIMPLE FINGER MOTION

by Michael Heinz

The question of whether electromyographic data from a single region of the forearm can be used to distinguish between various simple classes of finger motion is examined. Extensive clustering of data is performed to identify useful features for pattern classification. Sets of neural networks are trained to classify movements from each possible pairing of fingers. A multi-layered neural-fuzzy network is constructed to address the five-finger classification problem.

For Inés

Acknowledgments

I am indebted to many people for their support over the past five years. While it would be impossible to list everyone who has contributed to making this work possible, I would like to thank a few people in particular.

I am grateful to Dr. Joe Serrano and Professors Michael O'Flynn and Essam Marouf for teaching the excellent core courses in my program. I also appreciate Prof. Marouf's efforts as a reader of this thesis.

I would like to thank Prof. Sun Chiao for serving as my graduate advisor, and for allowing me unusual latitude in designing my graduate program. Prof. Chiao neither rubber stamped my plans, nor did he stand in the way of them. Instead, he forced me to justify my reasoning for my program, and always stood behind my choices when I was able to do so.

I appreciate the support of department chairman Prof. Ray Chen in backing my appointment to several teaching positions at San Jose State University. I am also indebted to the National Science Foundation for providing me with additional support by way of my thesis advisor, Prof. Benjamin Knapp.

I would like to thank Peter Vo and Sudagar Nagra for bringing friendship and humor to the bad old days at FSD.

Thanks also to Dave Grote, Mary Starkweather, and Doug Sprock, for friendship and companionship on the cliffs of Yosemite. I especially appreciate Doug's recent patience on Reed's Direct.

Thanks to Ric Vesely, for friendship and for keeping an open mind.

And thanks to Doug and Debbie Miller for many good times, and especially for some memorable wine parties.

Thanks to my long-time friends Dave and Susan Gold for enriching my life in so many ways.

Thanks to my parents and my brothers Mark and Val for so much love and support over the years. I appreciate the kindness that Val, Marsha, Michelle, and Nicole showed me while I stayed with them on my way to ICNN'96.

It is a pleasure to acknowledge the many ways that Prof. Richard Duda has contributed to my program of studies. His excellent class on neural networks gave me my first formal introduction to adaptive systems, and his contributions to my understanding of these systems have been many. I am also grateful for his many useful suggestions regarding this thesis.

I am deeply indebted to my thesis advisor, Prof. Benjamin Knapp, for his guidance, assistance, open-mindedness, and friendship over these past two years. To say that this work would have been impossible without him would be an understatement. I know that, in coming years, thousands of students will benefit from his talent and from the extreme dedication that he brings to his work.

Finally, I want to thank my wife Inés for her continuous love and support while I have weathered the ups and downs of graduate school. She, more than anyone else, made this work possible.

Contents

1. Introduction

1.1 Description of Problem

Over the past several decades, investigators have increasingly exploited the detection and analysis of various biological signals for use in applications such as the medical diagnosis of disorders, control of artificial prosthesis, and the enhancement of human-computer interfaces. One of the areas of investigation has been the use of electromyographic (EMG) signals. Surface EMG signals are the result of the gross electrical activity from a large ensemble of individual muscle fibers, and thus depend upon the properties and activities of both individual nerves as well as entire muscle groups. These signals can be detected using current surface electrode technology, and the speed of modern computer systems makes the real-time analysis and discrimination of the various classes of signals an increasingly viable option. This makes possible, for instance, the real-time control of prosthetics by handicapped individuals, or the enhanced manipulation of virtual reality type environments by computer users. In addition, recent theoretical work in the adaptive processing of signals has greatly increased the arsenal of tools that researchers can apply to such problems. Current efforts include the use of artificial neural networks (ANN), fuzzy logic, and genetic algorithms, in addition to more traditional signal processing tools [1,2,3,4].

This paper outlines the design and testing of a system that attempts to use EMG data from only a single region on the forearm to distinguish which of the hand's five digits is being lifted from a rest position (palm down, arm resting on table). The task of distinguishing between any given pair of fingers is first considered. The more difficult question of being able to distinguish between any one of the five fingers being raised (one at a time) is then considered. It is hoped that these efforts will eventually lead to robust finger motion classification systems that are largely independent of precise positioning of the sensory apparatus.

1.2 History of EMG Research

The study of the electrical nature of the human body has a long history. The German physiologist Emil Heinrich Du Bois-Reymond first reported the measurement of tiny electrical signals caused by the contraction of the muscles in his arm in the year 1849. Working without the benefit of modern electronic amplification equipment, Du Bois-Reymond had to induce a blister on each of his arms, remove the skin, and place his saline-soaked paper electrodes within the resulting wounds in order to bypass the electrical resistance of the skin [5].

Modern researchers (and test subjects) are more fortunate. Current electronic amplifiers, together with modern silver chloride electrodes, make it comparatively easy to measure EMG signals from the surface of the skin. Such signals were first utilized in the 1970's when researchers began to create prostheses that could operate by sensing the contractions in muscle groups. The application of these techniques to unimpaired muscles began to make it possible for even profoundly handicapped individuals to manipulate electronic equipment. One recent example of this occurred in 1993 when researchers at Loma Linda University Medical Center were able to provide a child who was completely paralyzed below the neck with the ability to move objects on a computer screen, keying upon signals detected from the boy's face [5].

1.3 Prior Related Work

ANN's have recently been applied to signal processing tasks in which the pattern classification to be performed is ill-defined or complex. Neural network structures such as the adaptive linear combiner have been used in conjunction with the least-mean-square (LMS) and other adaptive signal processing algorithms to create efficient, adaptive pattern classifiers [6]. Backpropagation and other techniques have greatly improved the supervised learning efficiency of multi-layered ANN's. In many instances, ANN's have proven more robust than rule-based expert systems in dealing with noisy environments, and have been easier to train since they do not require intricate and potentially unreliable heuristics. The intrinsically parallel nature of many neural network architectures means that correspondingly parallel special purpose hardware could be constructed if greater processing speed is required. Finally, ANN's lend themselves to the application of fuzzy logic, which take advantage of the system architect's a priori knowledge of the system [3,4].

It should be noted that ANN's also have disadvantages when used in signal processing. Among them is the "black-box" nature of the ANN [7], which makes it difficult to use the final algorithm to gain any intuitive insight into the system that is being analyzed. Without citing examples, there seem to be instances in the literature in which ANN's are "thrown at" various problems without regard to the rigorous modeling of the system under study, and with no appreciative increase in understanding of the system once the results have been obtained. Additional difficulties with ANN's include the task of finding adequate databases for both training and validation of the network, and the potentially lengthy training time that some networks require [7].

It should be noted that there are other analysis techniques that sometimes produce end products that offer no intuitive insight into the system. In one recent example, a team of researchers at Rice University used genetic algorithms to evolve a program to help control a prosthetic hand. Part of their software analyzes the nerve impulses picked up by three electrodes taped around the wrist and can tell, "with perfect accuracy," which way the subject's thumb is moving. The program reportedly contains a single line so long that it fills an entire page and contains hundreds of nested parenthetical expressions. No one knows why the expression works [8].

While the emphasis of the current work will be to build on recent investigations of the EMG (see for instance [9,10]), it is interesting to note that these techniques are currently being applied to the analysis of other types of biologically generated signals such as the electroencephalogram (EEG) and the electrocardiogram (EKG). In one such example, fuzzy logic was used to classify EEG data from a human subject into the categories of wakefulness and of five different states of sleep [3]. The algorithms succeeded in sorting 1101 epochs of data with a 77% success rate. In another study [11], a neural network classifier (a Kohonen learning vector quantizer) was used to analyze an EEG with the intention of predicting which hand the subject would use to press a switch. A testing set of size 250 was used with a 78% success rate. These and many other examples show the broad appeal of the analysis techniques being considered.

2. Data Acquisition

This section first discusses the hardware used to acquire the EMG data used in the study. The specific data acquisition (sampling rate, arm positions, etc.) are then considered. The filtering and preprocessing of the data are then discussed.

2.1 The BioMuse™

Data acquisition and signal preprocessing were performed using the Biomuse™ system which is manufactured by Biocontrol Systems Inc. The system detects changes in the EMG potential on the surface of the skin through a set of surface electrodes that provide differential signals covering the relevant sensitivity range of 0.5-10,000 μvolts. Although only one pair of differential signals was used in this study, the system is capable of supporting up to four such pairs of signals.

The signal path gain was on the order of 10,000, with initial Nyquist filtering being provided by a fourth-order Butterworth low-pass filter. After the filtering, twelve bit analog to digital conversion was applied to render the data accessible to the BioMuse's™ digital signal processor, a TMS-320C25. The DSP communicates with a PC host via an RS-232 serial connection. Real-time transmission of data to the PC host is limited to 500Hz due to the speed limitations of the RS-232 connection.

2.2 Specific Data Acquisition Used in This Study

The EMG signal was sampled at 500Hz and recorded in epochs of up to 30 seconds. The test subject's arm was resting on a table, and the thumb and each finger were sequentially lifted off of the table to create five different categories of test signal. Four such sets of data were collected from four different positions on the arm: the medial, central, and lateral areas on the bottom of the forearm, and the central part of the top of the forearm (see Figure 1). Subsequent analysis of the data from these four distinct arm positions was carried out independently of each other, as the goal was to identify a single measurement region that would provide enough signal information to determine which digit was being lifted.

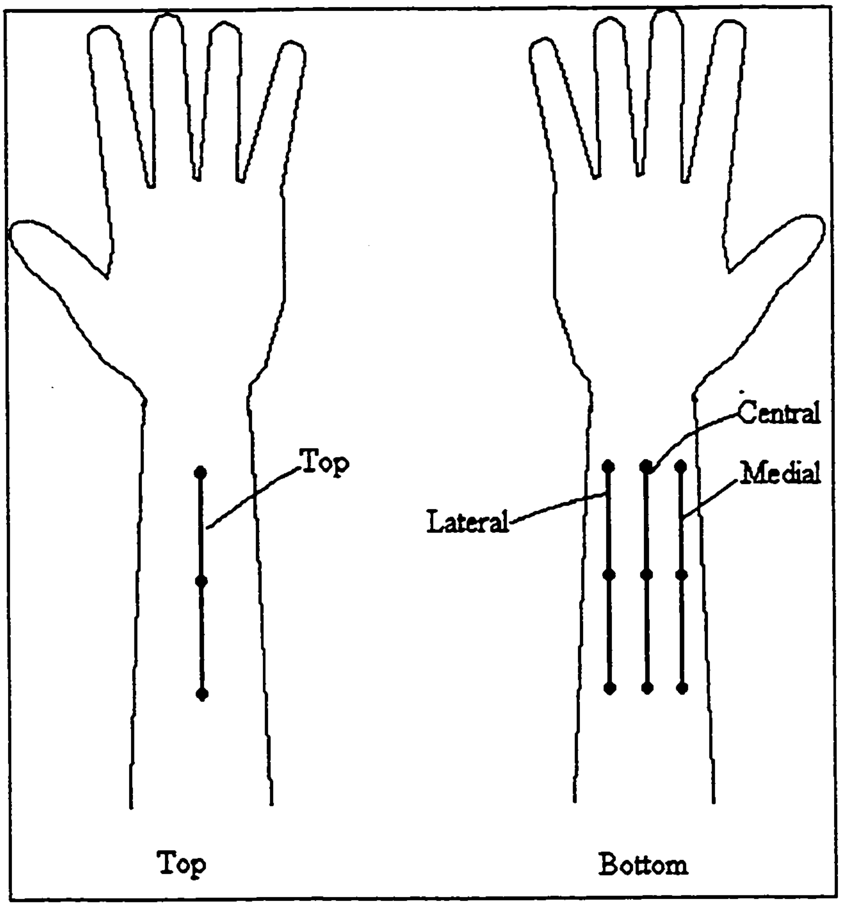

Figure 1: Data Sampling on Right Arm

2.3 Filtering and Preparation of the Data

Spectral information, as provided by application of the Fourier Transform to raw EMG data, was chosen as the basis of the feature set to be used for signal classification. This choice of feature sets is in no way unique, and was partly biased by the author's background in physics. The scientific literature is filled with analyses concerning the complicated way in which the attenuation of electromagnetic radiation through various media depends strongly on the frequency of the radiation (see for instance [12]). It was hoped that the attenuation of the various frequencies of electrical signals, the superposition of which provided the detected EMG data, would cause a sufficiently rich variation in the different classes of signals. Mathematically, the choice is reasonable insofar as any periodic signal can be completely characterized by its Fourier characteristics. Of course, the degree to which a short interval of EMG data (0.2 seconds, in this case), can be considered periodic, much less stationary, is variable.

Data associated with each digit and each measurement position was preprocessed by being separated into 100-point (0.2 second), non-overlapping segments that were then Fourier transformed and low-pass filtered to obtain estimates of their spectral content. The choice of 100 points was made empirically based on early results that suggested that this would be a large enough frame size to obtain a frequency resolution that provides a sufficiently diverse set of features for accurate classification. This choice of frame size was also small enough to reduce the non-stationary effects of the signal. The resulting transformed data sets contained data points for each of the 20 possible finger/measurement combinations. Each data point contained 51 features, corresponding to the frequencies from 0Hz (DC) through 250Hz (the sampling Nyquist frequency) in 5Hz intervals. Separate portions of this data were used for training and for testing of the finished networks. In the case of the five-finger classifier system, a separate validation data set was used to measure the final system performance, since the test sets were used to determine the point at which system training should be terminated.

Additional processing of the data took the form of smoothing in the frequency domain using a three-tap moving average filter (with "wrap around" at either end of the 51 feature "data point"). This also was an empirical choice based on the observation that the resulting 51-tuples provided pairs of features that gave generally better clustering results as measured by the metric discussed in the next chapter. Quantitatively, circular convolution by a moving average filter in the frequency domain is equivalent to multiplication in the time domain by a portion of a sinc function. One could argue that similar improvements in clustering might be obtained by attenuating in some way the magnitudes of the data points in the 100-point raw data, as would happen if those points were scaled by the lobes of the discrete sinc function. The author offers no statement to the effect that this approach was uniquely appropriate, but rather that it was useful in improving the overall separation of the various classes of data in this system. Qualitatively, the filter had the effect of rounding the sharp edges (high frequency changes) of the features in the frequency domain. Examples of the same thumb data in its raw, Fourier transformed, and final smoothed forms are shown in Figures 2-4.

3. Data Clustering and Feature Selection

This section considers the application of data clustering algorithms to the problem of selecting useful features for solving the finger classification problem. The application of clustering to pattern recognition is first discussed. The specific clustering algorithms, K-means and nearest neighbors, are examined, and the results for the 2- and 5-finger

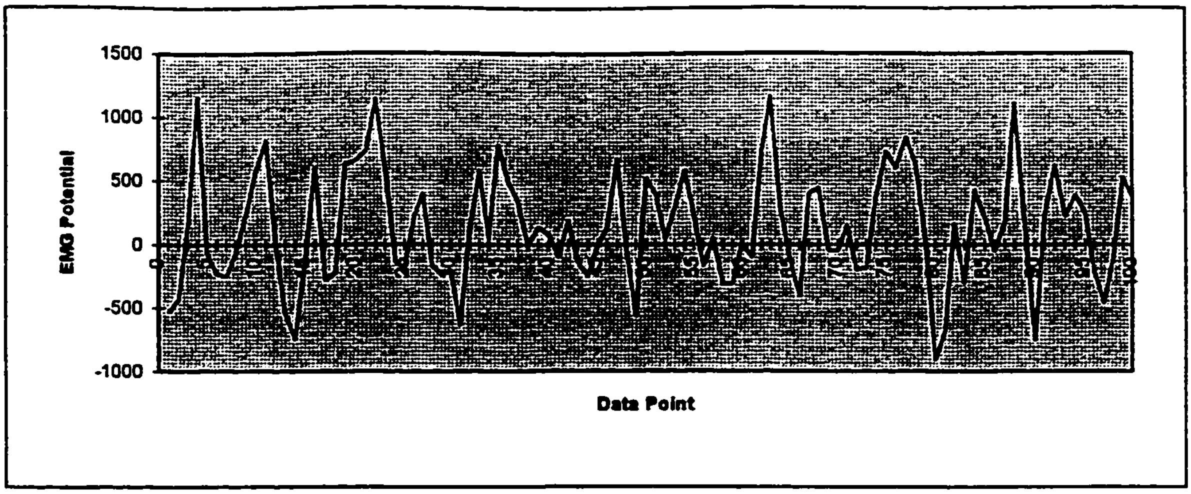

Figure 2: Raw Thumb Data - Bottom Center of Forearm, Dr. Benjamin Knapp

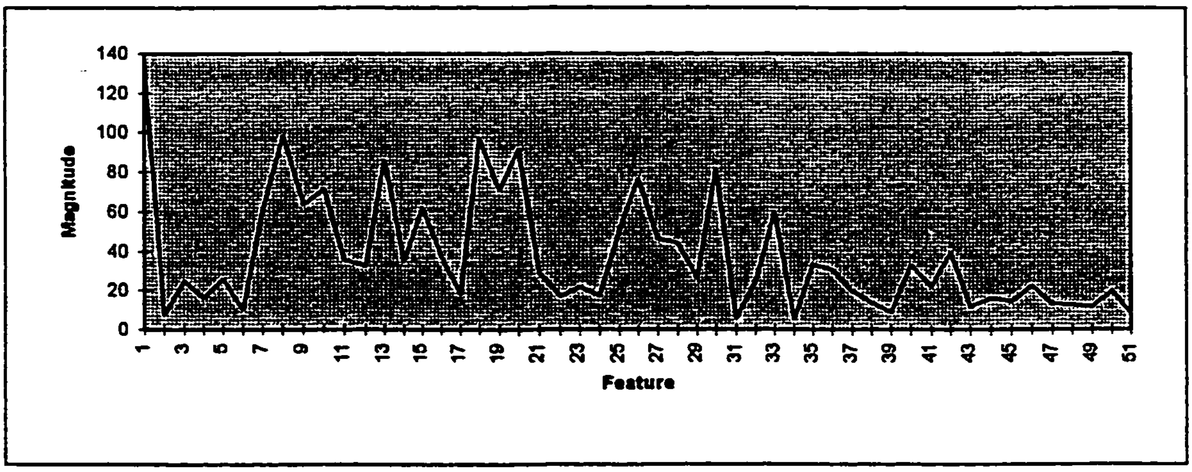

Figure 3: Thumb Data After Fourier Transform

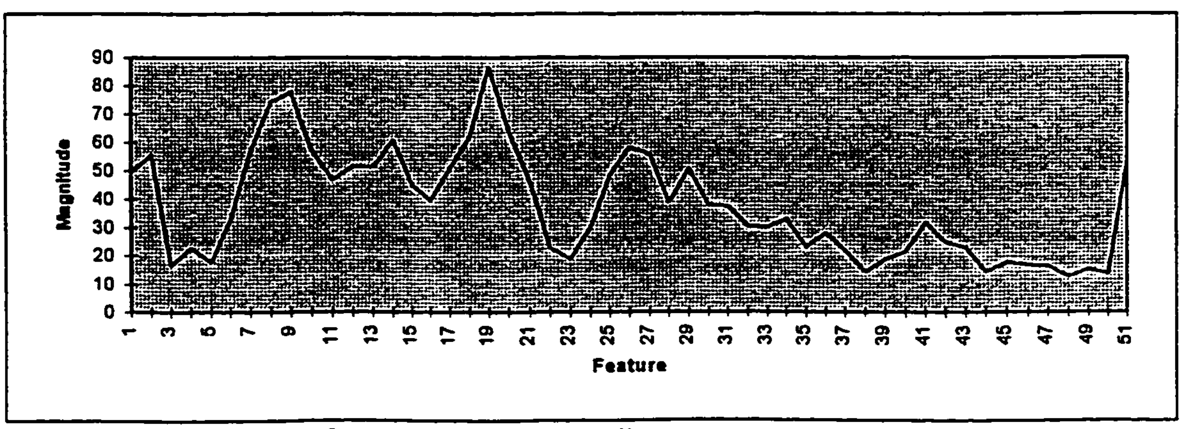

Figure 4: Thumb Data After Moving Average Filter

classification problems are given. A reasonably general metric used to measure the success of the clustering is presented, and its shortcomings are discussed.

3.1 Use of Clustering in Pattern Recognition

Cluster analysis is the process of classifying objects into sets that are relevant in the context of a particular problem. Ideally, an astute application of clustering techniques to a set of training data leads to the objects being organized into an efficient representation that accurately characterizes the complete population being considered. Clustering algorithms are an example of unsupervised learning, since no correct classifications need be supplied in order to train the clustering algorithm. This fact was key to the application of clustering to the problem at hand. The high dimensionality of the processed data (51 features) made the direct visualization of the data impossible. Since the number of combinations in which small numbers of features could be selected was large, clustering was used extensively to assess the success with which a given set of features separated the relevant classes of data.

3.2 Clustering Algorithms

The K-means and nearest neighbors algorithms are thoroughly discussed in the literature, and will only be briefly summarized here. Data sets that favor each algorithm will be considered.

3.2.1 K-means Clustering Algorithm

The K-means algorithm [13] requires that the user supply the number of clusters the algorithm is to deliver. This implies that the analyst has a priori knowledge of the number of clusters into which the data should naturally be separated, or that some test of the clusters' validity will be brought to bear. The algorithm first assigns one data point to each cluster arbitrarily, with that point trivially becoming the cluster center, and then assigns each of the remaining data points to the cluster with the closest cluster center. The new cluster centers are then computed based on the new cluster memberships, and reassignment of each point's cluster memberships is once again performed based on which cluster center is closest to that point. The process continues until no further changes occur, and is guaranteed to terminate. A cluster center is taken to be the center of mass of the points in the cluster, with all points equally weighted, and all distances are taken (in our case) to be Euclidean. K-means runs fairly quickly, with termination in data sets of 200 points typically occurring in six to eight passes through the data set. (It is interesting to note, however, that optimal K-means clustering, relative to some user specified performance metric, is an NP-complete problem [14]. The author also found, to his surprise, that the algorithm can, in rare cases, return fewer clusters then were requested.)

Because K-means cluster centers migrate to the center of mass of dense sets of points, the algorithm tends to perform well on sets of data in which each class of data is well separated in space. Examples include classes that can be easily separated by a line, plane, or hyperplane (Figure 5). In contrast, the algorithm typically performs poorly when classes of data are radially symmetric, and thus share common centers of mass (Figure 6).

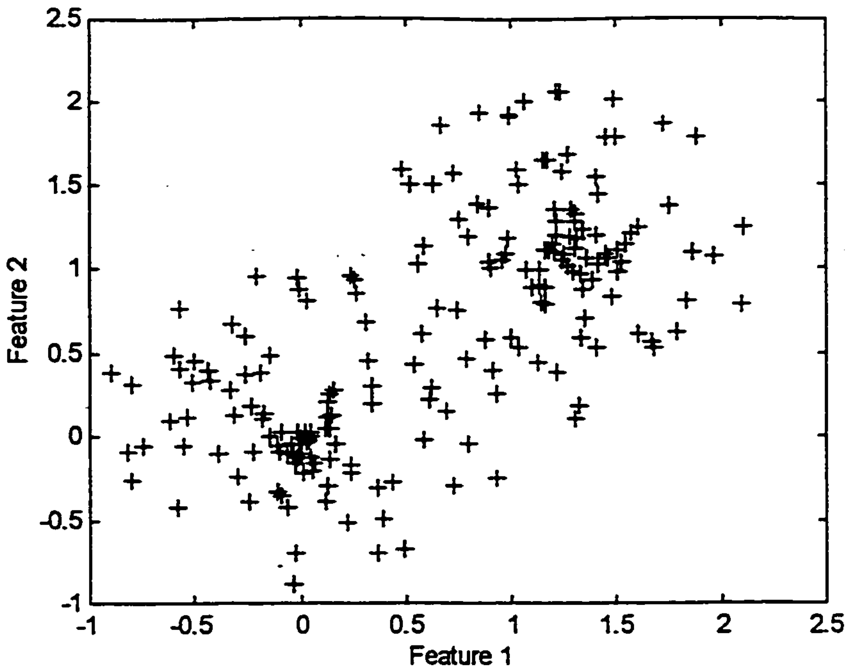

Figure 5: Data well conditioned for K-means, but poorly conditioned for Nearest Neighbors

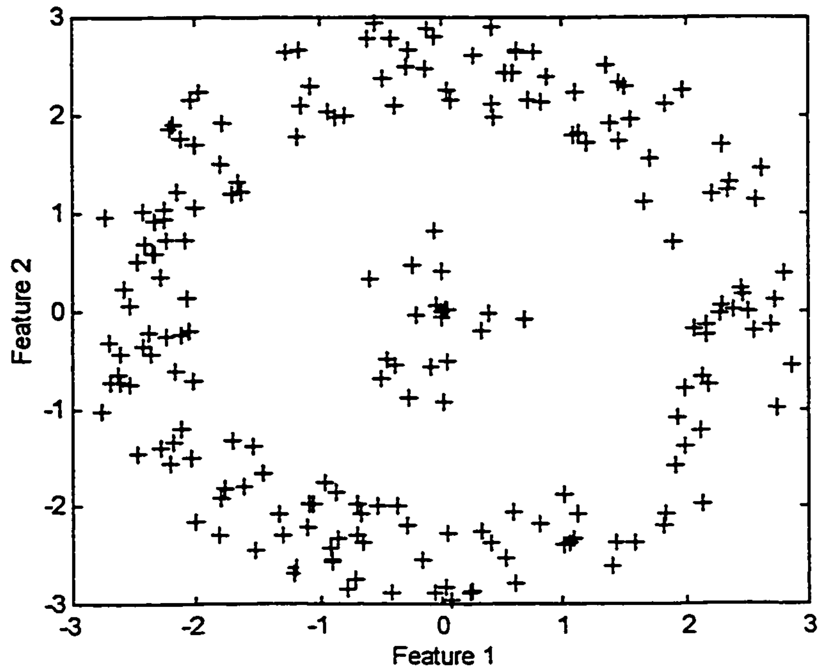

Figure 6: Data well conditioned for Nearest Neighbors, but poorly conditioned for K-means

3.2.2 Nearest Neighbors Clustering Algorithm

The nearest neighbors algorithm requires that the user supply a parameter (t) that constitutes the maximum distance between two points for which they will still be considered to be members of the same cluster. The algorithm then begins with a single point, collects all points that belong to the same cluster as that point, and then starts a new cluster and continues the process until no points are left.

To obtain some user specified number of clusters using this algorithm, it may be necessary to try a number of different t's, with larger values producing fewer clusters and vice versa. This search can render the algorithm computationally expensive. In general, the t that provides some given number of clusters is not unique. Also, two different t's may deliver the same number of clusters, but with somewhat different cluster memberships. Some pathological data sets may not be able to be clustered using this algorithm, all though this rarely occurs in practice. An example would be the vertices of a cube, which this algorithm could separate into one or eight (trivial) clusters, but no number of clusters in between.

If the data to be clustered does not lend itself well to this algorithm, then the number of clusters produced may be extremely sensitive to small changes in t, as would be the case with the data in Figure 5. However, the algorithm can perform extremely well on classes of data that have some spatial separation between them, regardless of the relative proximity of the data sets (as in Figure 6). The algorithm also performs well when dealing with data sets that have a smaller intrinsic dimensionality then the space in which they are embedded (Figure 7).

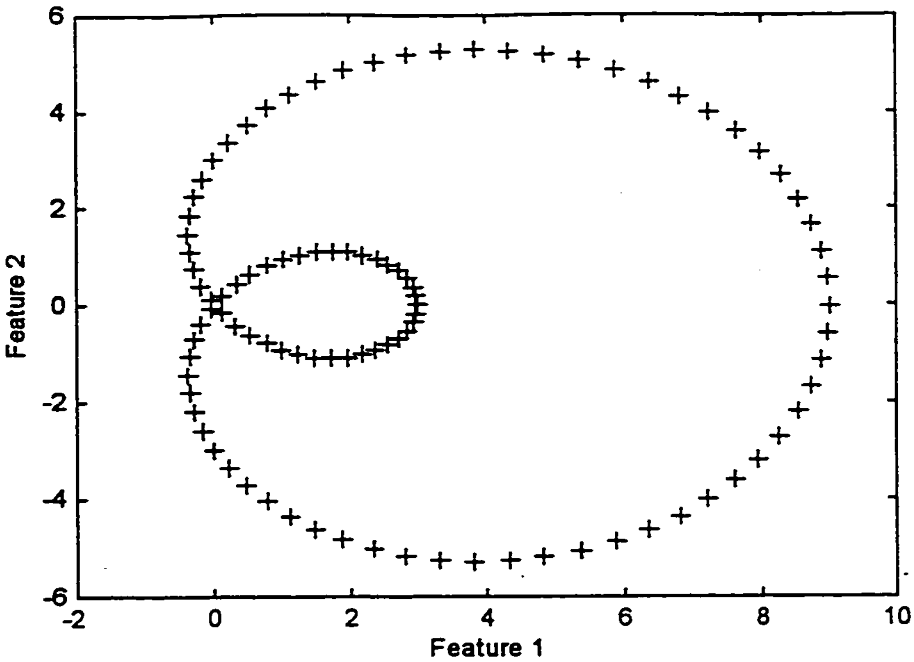

Figure 7: Data with Intrinsic Dimensionality 1

3.3 The Clustering Metric Used in This Study

Just as there is no all purpose clustering algorithm that fills every need, there is no general metric that can be used to judge the success of clustering in all situations. The degree to which clustering succeeds has to be evaluated in terms of the context of the problem that is to be solved. Nevertheless, in this study many thousands of sets of data were clustered and an automated way of accessing the clustering was required. Since in all cases the data came from two classes with each point's correct membership already known, the question of how well the clustering has succeeded in separating the data can be considered from two points of view: the simple labeling of the points (correctly or incorrectly classified), and the location of the incorrectly classified points. The classification of the points can be addressed in a straight-forward manner by choosing the most favorable of the two possible mappings between the two classes of input data and the two sets returned by the clustering algorithm. Specifically, if set A contains M data points and set B contains N data points, and if clustering results in m points in set A being correctly classified and n points in set B being correctly classified, then the metric returns $\max[(m+n)/(M+N), 1-(m+n)/(M+N)]$. Thus, if Figure 8 contains the correct classification of two sets of data that are to be clustered as one set of points, then Figures 9 and 10 contain equivalent classifications of this data. The metric that was used in this study would return 0.9 for both of these classifications, on a scale that ranges from 0.5 (because the two classes contain equal numbers of points) to 1.0 for a perfect classification of the data.

It is important to note that this metric ignores the geometrical aspects of incorrectly classified data. The metric would, for instance, return 0.9 as the score for the classification illustrated in Figure 11, just as it would for Figures 9 and 10. Most observers would judge the manner in which the data is classified in Figure 11 to be qualitatively inferior to that of Figures 9 and 10, and the degree to which this difference matters again depends on what is important in the system under study. Given the large number of data sets to be analyzed and essentially no prior information on the distribution of the data, it seemed difficult to tailor a metric to take into account these subtleties. However, as will be discussed in detail in later sections, it was found that the clustering was fairly successful in generating useful data classifications (many results above 0.9 for 2-finger classification and above 0.8 for 5-finger classification) and that the data sets separated reasonably well in space (favorable to K-means).

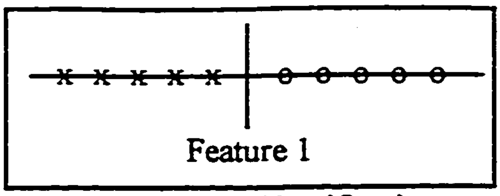

Figure 8: Correct Classification of Data

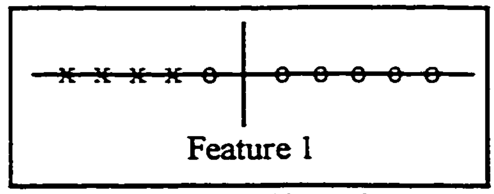

Figure 9: Sample Clustering - Metric Returns Score of 0.9

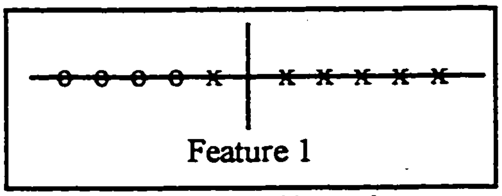

Figure 10: Sample Clustering - Metric Returns Score of 0.9

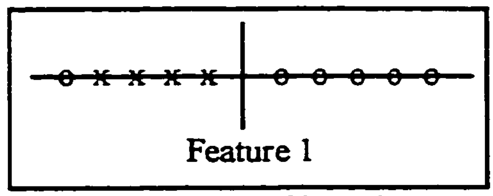

Figure 11: Sample Clustering - Metric Returns Score of 0.9

3.4 Results of the Clustering Runs

3.4.1 Two-Finger Clustering

In order to select effective sets of features to provide input to the neural network classifiers, extensive efforts were made to cluster the data using both the smoothed and un-smoothed data sets (see section 2.3), and using both the K-means and nearest neighbor algorithms. In the case of classification between pairs of fingers, the training data from all four arm positions was grouped into ten sets corresponding to all combinations of data from five fingers taken two sets at a time. These ten sets of data were then clustered using all combinations of both one feature and two feature pairings (51 cases for one feature, 1275 for two features). A total of 2*2*4*10*(51+1275) = 106,080 clustering runs were performed. In this way, a search was made for what would hopefully be near optimal sets of features that could be used to discriminate between any given pair of digits.

In all cases, the K-means classification results proved to be superior to the nearest neighbor classification results, and the smoothed data classified as well or better than the un-smoothed data. Clustering by pairs of features provided results superior to the clusters generated from individual features, as would reasonably be expected. Because of the large dimensionality of the feature space, exhaustive searches of feature sets containing more than two features would not be practical, although selective searches by genetic or other means might prove effective.

General results of the clustering runs indicated that the central location on top of the forearm provided the most favorable separation of the five classes of signals when considered in pairs. Favorable sets of features, selected from among the best clustering data, and recorded from this location, are summarized in Table 1. These features were selected as the inputs to the neural network classifier described in section 4.1.

3.4.2 Five-Finger Clustering

In the case of classification between all five fingers, the training data was grouped into five sets corresponding to all combinations of data from one finger versus the other four. Because of the results discussed in section 3.4.1, clustering was performed using only the K-means algorithm, and only using smoothed data. Once again, all combinations of one feature and two feature pairs were considered, and data from all four arm positions was analyzed. A total of 5*4*(51+1275) = 26,520 clustering runs were performed. General results of the runs indicated that the central area on the bottom of the forearm was marginally superior to the other sites, and all further work was performed with data from only that location.

4. Supervised Learning - Neural and Fuzzy Classification of Data

This study was primarily intended to address the application of neural and neural-fuzzy networks to the EMG classification problem. The literature is filled with many variations of neural network architectures (see, for instance, [7]), and this study was once again not intended as a survey of all alternatives, but rather as an investigation of one set of choices. The neural network structure used in the 2-finger classification problem is discussed in section 4.1, and the neural-fuzzy system constructed for 5-finger classification is discussed in section 4.2.

4.1 Two-Finger Classification Efforts - Radial Basis Function Neural Networks

The radial basis function network (Figure 12) is a two-layer neural network. The hidden layer contains neurons that produce a significant non-zero response only when the

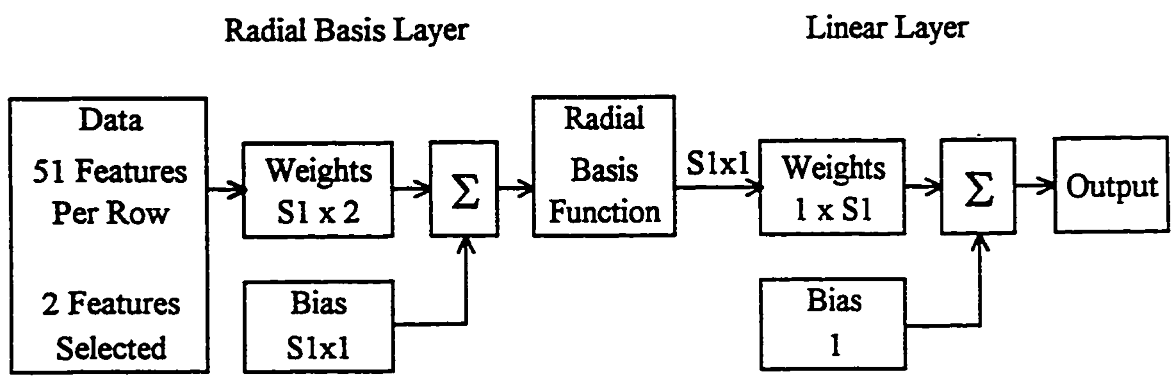

Figure 12: Two-Finger Radial Basis Network Classifier

input falls within a small region of the input space. In this study, the transfer function for each hidden neuron was a Gaussian transfer function of the form [15]:

$$\text{radbas}(n,b) = e^{-(b*n)^\wedge 2}$$

as implemented in the MATLAB Neural Network Toolbox. The output layer of the network consists of a linear combination of the outputs of the hidden layer neurons.

The only design constants in a radial basis network are the number of neurons used and the spread (b) of the Gaussian transfer functions. A variety of spreads were tested on the training data in an effort to optimize the performance of the network on each 2-finger classification problem. The number of hidden neurons used and the placement of these neurons in the input space was controlled using an orthogonal least squares learning algorithm available in the MATLAB toolbox [15] and first discussed in [16]. The algorithm chooses each radial basis function center one by one in such a way that each new center maximizes the increment to the energy of the desired output and does not suffer numerical ill-conditioning problems. In this way, the algorithm requires only one pass of the training data and the choice of centers is directly linked to the reduction of error signals.

4.2 Neural-Fuzzy Based Five-Finger Classifier

The five-finger classification network used in this study is shown in Figure 13. Each of the five neural network blocks (NN1, etc.) is a radial basis network with an architecture of the type described in section 4.1. However, instead of each neural network being trained on a two-finger classification problem, each network was trained to distinguish one digit versus all others. For instance, NN1 was trained to distinguish the thumb versus any of the other four fingers. Thumb-like responses were trained to produce the value 1, while other responses were trained to produce the value -1. Each of the five networks was provided with input information from two features determined through clustering to be favorable to that particular classification problem. In initial efforts, the output of the five neural networks was evaluated directly, with the maximum value being taken as an indicator of which digit the network was choosing. Thus, if NN2 produced an output of 0.8 and the other outputs were -0.3, -1.1, 0.3, and -0.7, the classifier would conclude that the index finger was being raised.

Experiments with the network described above led to the discovery that many of the output ranges on the neural networks (NN1 - NN5) were skewed relative to the ideal mapping range of [-1 1]. While many of the networks were doing a reasonable job of mapping their respective digit to values that were positive with respect to the other four digits, they were producing mapping ranges with end points that varied widely from network to network.

In an effort to correct the mapping problem while taking advantage of the knowledge that relative output values represent relative confidence as to which digit is being detected, a second layer of fuzzy networks was implemented as shown in Figure 13. The fuzzification of each of the five inputs to each network occurred using a sigmoid function reflecting the (ideally) monotonic change in confidence as one moves through the range of input values. As with the neural-network layers, the fuzzy layers were each trained to give an output response reflecting whether that network's corresponding digit had been raised or whether one of the other digits had been raised. For instance, FN1 was trained to produce an output of 1 for thumb data, and an output of 0 for all other digits. The outputs of all five networks were then examined, and the network with the largest value was taken as the winning network.

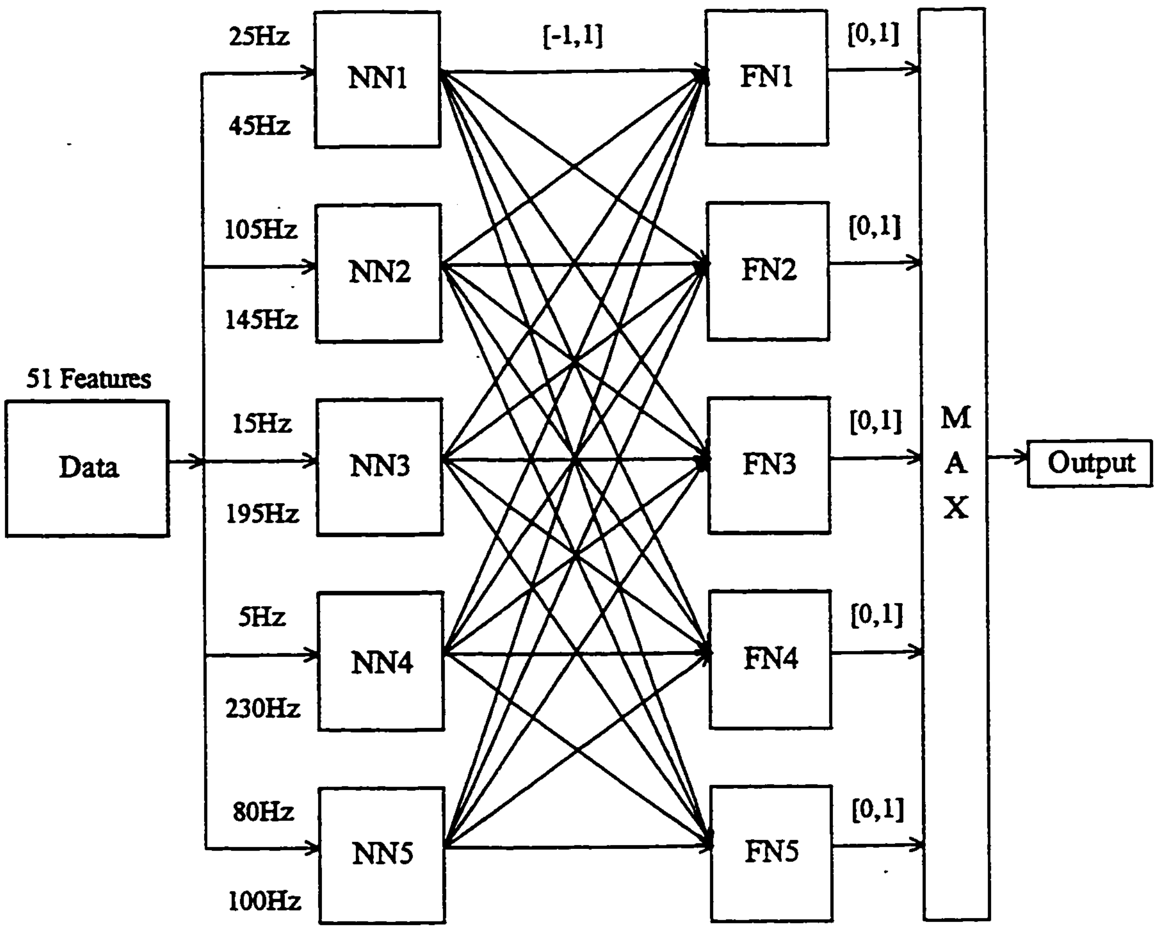

Figure 13: Five-Finger Neural-Fuzzy Classifier

Each of the neural-fuzzy networks (FN1 - FN5) was implemented as a Sugeno network and trained using an adaptive-network-based fuzzy inference system (ANFIS) [17], as implemented in the MATLAB Fuzzy Logic Toolbox. ANFIS allows the user to use the language-based rules and user-provided input functions that make fuzzy logic so powerful in situations where the user has some knowledge of the system under study (the outputs of NN1 - NN5). At the same time, it allows the network to be trained using learning algorithms such as backpropagation in the same way that neural networks can be trained.

If Gaussian functions are used as the basis functions in an ANFIS system, then it is interesting to note that the resulting ANFIS system can be shown to be functionally equivalent to a radial basis function neural network (RBFN). Thus, all of the learning and optimization techniques that have been developed for RBFN's can potentially be applied to ANFIS.

5. Implementation and Results

5.1 Two-Finger Classification Efforts

Ten radial basis networks were designed using the features tabulated in Table 1. Each network contained a hidden layer of radial basis functions and a linear layer of output neurons. Each network was trained on an appropriate set of 20 vectors containing data taken from the two fingers that that network was supposed to classify. The radial basis layers were created one neuron at a time, with each new neuron based on the input vector that would result in lowering the network error the most [16]. The ten networks were then tested using independent data files, each containing 20 test vectors. The results are summarized in Table 1.

5.2 Five-Finger Classification Efforts

The five-finger classifier was implemented as shown in Figure 13, and consisted of five radial basis function neural networks and five fuzzy networks. The fuzzy network layer was introduced in an effort to eliminate problems stemming from the skewed output ranges of the neural networks, as discussed in section 4.2. Each of the neural networks

Classifier		Features (frequencies in Hz)		Percent Correctly Classified
Thumb	Finger 1	10	15	90%
Thumb	Finger 2	115	150	85%
Thumb	Finger 3	5	205	90%
Thumb	Finger 4	70	75	60%
Finger 1	Finger 2	10	195	100%
Finger 1	Finger 3	235	240	100%
Finger 1	Finger 4	185	210	100%
Finger 2	Finger 3	105	165	85%
Finger 2	Finger 4	105	135	90%
Finger 3	Finger 4	205	230	80%

Table 1: Neural Network Features and Performance - Two Finger Case

was trained using features appropriate for their individual classification task, with training and test sets consisting of 50 vectors from each digit (250 total training and test vectors). The input feature choices for all five neural networks are summarized in Table 2. The neural networks and fuzzy networks were trained separately, with the fuzzy networks using the processed neural network output as their inputs. A separate set of validation data was used to measure the performance of the final network, as the test data was used to determine the point at which the training of the networks should end. (Note that this validation set was not necessary with the two-finger classifier. The variable parameters consisted of the width of the basis functions and the desired error on the training set, and each was adjusted to minimize the error of the training set. The test set remained independent and was in no way used to build the networks.)

Table 2 summarizes the results for the training, checking, and validation data. As a cross check of the results, the networks were retrained using the validation data as training and checking data, and using the training and checking data as validation data. The results of those runs are summarized in Table 3.

Classifier (Finger vs. All Others)	Features (Frequencies in Hz)		Percent Correctly Classified: Training Data	Checking Data	Validation Data
Thumb	25	45	98%	86%	34%
Finger 1	105	145	52%	28%	23%
Finger 2	15	195	82%	66%	10%
Finger 3	5	230	52%	22%	1%
Finger 4	80	100	92%	92%	87%

Table 2: Neural-Fuzzy Network Features and Performance - Case 1

Classifier (Finger vs. All Others)	Features (Frequencies in Hz)		Percent Correctly Classified: Training Data	Checking Data	Validation Data
Thumb	25	45	94%	96%	1%
Finger 1	105	145	76%	42%	27%
Finger 2	15	195	50%	46%	82%
Finger 3	5	230	70%	52%	18%
Finger 4	80	100	66%	74%	47%

Table 3: Neural-Fuzzy Network Features and Performance - Case 2

As can be seen in tables 2 and 3, the results of the five-finger classifier appear very marginal. While in some cases the validation data performed fairly well (80+ percent), most of the network's performance was poor. In particular, finger three in case one and the thumb in case two were almost completely misclassified. Finger three in case one was largely misclassified as the thumb (69%), and the thumb in case two was largely misclassified as finger 1 (63%). Given that the training, test, and validation data were all collected during the same session, this large disparity in network performance suggests that a single channel of data may not have enough information to support a robust five-finger classification system.

Another possibility is that the signal may have some non-stationary properties, and that the location of the cluster centers may be drifting with time. This may be a reasonable hypothesis, since the biological system in question will be subject to fatigue and other effects. One piece of evidence that potentially supports this hypothesis is the fact that several of the fuzzy networks showed immediate, in some cases monotonic, increases in the errors associated with the checking data. Since the only difference between the training and checking data was the interval of time under consideration, this suggests that the system had somehow changed during the ten seconds that the training (first five seconds) and checking data (next five seconds) were recorded.

6. Conclusions

6.1 Performance of Two-Finger Classifier

As can be seen in Table 1, many of the classifiers operated at a reasonably high level of performance. Only the thumb/finger 4 classifier performed poorly, with test data delivering a 60% performance level. Clustering results indicate that many of the performances would be greatly improved with analysis of EMG signals from one of the other arm locations. For instance, the thumb/finger 4 classifier would likely work better on data taken from the lateral part of the lower forearm, and the finger 2/finger 3 classifier would respond best to data taken from either the medial or central part of the lower forearm. It may be necessary to abandon the use of only a single EMG signal if one expects to get simultaneously high performance from all ten of these classifiers.

General results of the clustering runs indicated that the central location on top of the forearm was the single most favorable data sampling location when considering all ten two-finger classification problems. However, other locations might favor a particular two-finger classifier.

6.2 Performance of Five-Finger Classifier

While the training and test data received mixed but favorable results, the validation data results tended to be poor. The reason for this is not clear. It may be that a single channel of EMG data is inadequate to build a robust five-finger classifier. Another possibility is that the biological system may produce non-stationary signals, so that the cluster centers are drifting in the feature space. If so, it may be possible to build an adaptive network that tracks these changes and compensates for them. Further work will be necessary to determine if a reliable five-finger classifier can be constructed.

General results of the runs indicated that the central area on the bottom of the forearm was marginally superior to the other sites.

7. Areas For Future Work

The analysis and use of EMG data is still in its infancy. This and other studies have necessarily limited themselves to very specific lines of investigation, and many other approaches remain unexplored. A few of the ways in which this study might be extended and improved are briefly discussed in the following sections.

7.1 Examination of the Robustness of the Results From This Study

This study was based on data from only two test subjects, and with data taken from only a small number of test sessions. A more thorough study might make use of cross validation of the existing data. Also, it would be worth while to examine how well the systems discussed above operate on a larger number of test subjects, and with small variations in placement of the electrodes on the subjects' arms.

7.2 Use of Adaptive Signal Processing to Track Non-Stationary Signals

If the poor performance of the five-finger classifier can be attributed to non-stationary effects in the EMG signals, then it may be possible to build an adaptive classifier that tracks these changes and compensates for them.

7.3 Multi-Signal Analysis (Several Areas of Arm Sampled Simultaneously)

A logical extension to this work would be the use of data sampled simultaneously from several points on the arm. As was previously noted, a single source of data appears to be inadequate to solve the 5-finger classification problem, and multi-channel data might provide a sufficiently rich information source.

7.4 Use of Other Features

The features in this study were limited to spectral information obtained by way of the Fourier Transform. Many other feature sets are possible, including features that correlate signals from more than one data channel.

7.5 More Complex Finger Motions and Analysis of Strength of Motion

The analysis of other finger and hand motions could be explored. In addition, previous studies have used envelope detection to create a control signal proportional to the strength of the muscle contractions [10], and an analogous study could be made on finger strength.

7.6 Fuzzy Clustering

Fuzzy clustering could be used instead of the "crisp" clustering algorithms used in this study.

7.7 Statistical Pattern Recognition

The mathematical tools of statistical pattern recognition could be brought to bear upon these classification problems, perhaps offering more insight into the properties of the system. In particular, one might build a classifier that makes use of the relative Mahalanobis distances to each cluster center. A rigorous mathematical analysis of these signals might be related to existing models in physiology and neuro-anatomy.

8. Some Personal Thoughts on Pattern Recognition

After working part-time for two years on issues related to pattern recognition, I almost feel as if I know less than when I started. The remarkable array of tools that researchers now have includes the many techniques of statistical pattern recognition, neural networks, fuzzy systems, and evolutionary tools such as genetic algorithms. In addition, special purpose pattern recognition hardware is finally beginning to see its day. To cite one line of research, Carver Mead and his colleagues at the California Institute of Technology have begun putting large scale analog neural networks on silicon using modern VLSI tools (see, for instance, [18]). Their chips seek to use the natural, powerful computational primitives available from semiconductor devices to circumvent current digital bottlenecks that prevent the real-time solution of some pattern recognition problems.

For my part, I have had the pleasure to watch the development of an extraordinarily powerful adaptive system by way of my son, Sam, who was born about the time that I began this work, and who is now two years old. In that span of time, I have watched him develop from a new-born infant into an individual who harbors abilities that defy analysis by all of the world's researchers. One is tempted to take to heart a recommendation by Hecht-Nielsen (p. 51, [7]): "Everyone in neurocomputing should have a basic knowledge of neuroscience, if for no other reason than to be properly humbled by the super-advanced alien technology used in the construction of brains." It seems likely to me that our approaches to understanding so-called intelligent computing are still symptomatic, addressing the superficial classification problems that we have been able to characterize, while some underlying theory that might really explain cognition remains yet to be discovered. If we are able to use evolutionary computational techniques to build machines that seem intelligent, one wonders if we will be intelligent enough to really understand our creations.

References

[1] Renato, M. E. Sabbatini, "Neural Networks for Classification and Pattern Recognition of Biological Signals," *Proceedings of the 15th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 265-266, 1993.

[2] Costa, J. D., and Gander, R. E., "Classification of Muscle Contraction Levels With an Artificial Neural Network," *Proceedings of the 15th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 259-20, 1993.

[3] Hu, Jung, "EEG Pattern Recognition Using Fuzzy Logic," Master's Thesis, San Jose State University, 1991.

[4] Hu, J., and Knapp, R. B., "Electroencephalogram Pattern Recognition Using Fuzzy Logic," *Proceedings of the 25th Conference on Signals, Systems, and Computation*, Vol. 2, pp. 805-807, November, 1991.

[5] Lusted, H. S., and Knapp, R. B., "Controlling Computers With Neural Signals," *Scientific American*, pp. 82-87, October, 1996.

[6] Widrow, Bernard, and Stearns, Samuel D., *Adaptive Signal Processing*, Prentice-Hall, 1985.

[7] Hecht-Nielsen, Robert, *Neurocomputing*, Addison Wesley, 1990.

[8] Gibbs, W. W., "Technology and Business: Programming With Primordial Ooze," *Scientific American*, pp. 48-50, October, 1996.

[9] Putnam, William, and Knapp, R. Benjamin, "Real Time Computer Control Using Pattern Recognition of the Electromyogram," *Proceedings of the 15th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 1236-1237, 1993.

[10] Putnam, William, Master's Project, San Jose State University, 1993.

[11] Flotzinger, D., Pfurtscheller, G., Neuper, Ch., Mohl, W., Berger, H., "Classification of Non-averaged EEG Data by Learning Vector Quantization and the Impact of Signal Preprocessing," *Proceedings of the 15th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 263-264, 1993.

[12] Jackson, J. D., *Classical Electrodynamics*, pp. 290-292, Wiley, 1975.

[13] Hush, D. R., Horne, B. G., "Progress in Supervised Neural Networks - What's New Since Lippmann?," *IEEE Signal Processing Magazine*, January, 1993, pp. 8-39.

[14] Garey, M. R., and Johnson, D. S., *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, New York, NY, 1979.

[15] Demuth, H., and Beale, M., *Neural Network Toolbox*, The Math Works Inc., pp. 6-1 through 6-12, 1994.

[16] Chen, S., Cowan, C. F. N., and Grant, P. M., "Orthogonal Least Squares Learning Algorithm for Radial Basis Function Networks," *IEEE Transactions on Neural Networks*, pp. 302-309, March, 1991.

[17] Jang, J. S. R., "ANFIS: Adaptive-Network-Based Fuzzy Inference System," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 23, No. 3, May/June, 1993.

[18] Koch, C., and Mathur, B., "Neuromorphic Vision Chips," *IEEE Spectrum*, May, 1996, pp. 38-46.

Appendix 1 - Clustering Results

This appendix contains a partial listing of the results of the clustering performed to support the five-finger classifier. The first page, 1 versus not 1, displays the results of clustering of thumb data versus data from fingers 1 through 4. Results are shown for the four areas of the arm where data was taken, and for clustering using 1 and 2 features. The results show the features and metc score (section 3.3) for the 20 most favorable runs. The other pages show similar results for the other four digits.

Clustering in the first chart was performed using 50 data points from the thumb and 50 data points taken from the collective data of the other four digits. As this left 150 data points from fingers 1 through 4 unused, the clustering was performed 4 separate times. Each time a different subset of the available data was used. The feature choices for the five-finger classifier were made by considering the collective results of these clustering runs. Only one of these four runs is shown here for each digit. The other tables are available from the author.

1 versus not 1																			
Medial					Central					Lateral					Top				
1 feature		2 features			1feature		2 features			1 feature		2 features			1 feature		2 features		
15	0.600	18	28	0.775	21	0.625	8	18	0.750	10	0.725	7	32	0.900	20	0.600	7	18	0.750
33	0.625	18	30	0.775	26	0.625	9	14	0.750	13	0.725	29	32	0.900	45	0.625	8	15	0.750
42	0.625	5	18	0.775	14	0.625	9	24	0.750	47	0.750	5	32	0.900	48	0.625	8	18	0.750
24	0.625	5	29	0.775	49	0.625	10	11	0.750	26	0.775	32	35	0.900	13	0.625	5	9	0.750
16	0.625	6	18	0.775	29	0.625	5	42	0.750	22	0.775	9	31	0.900	4	0.650	8	19	0.750
30	0.625	18	31	0.775	6	0.650	10	12	0.750	7	0.775	32	36	0.900	6	0.650	8	51	0.750
21	0.650	7	18	0.775	8	0.650	10	19	0.750	9	0.775	9	32	0.900	7	0.650	5	10	0.750
45	0.650	2	18	0.775	47	0.650	6	12	0.775	35	0.775	6	33	0.900	11	0.650	9	10	0.750
28	0.650	10	18	0.775	48	0.650	7	13	0.775	15	0.775	9	33	0.925	50	0.650	9	19	0.750
25	0.650	4	18	0.775	44	0.675	10	42	0.775	18	0.775	32	50	0.925	16	0.650	3	10	0.750
13	0.650	18	32	0.775	9	0.675	10	45	0.775	44	0.800	32	51	0.925	17	0.650	9	34	0.775
34	0.675	18	33	0.775	46	0.675	10	46	0.775	49	0.800	7	33	0.925	18	0.650	9	45	0.775
23	0.700	3	18	0.775	3	0.700	11	27	0.775	30	0.800	28	34	0.925	19	0.650	9	50	0.775
12	0.700	18	34	0.775	4	0.700	11	28	0.775	43	0.800	29	33	0.925	3	0.675	10	18	0.775
17	0.700	12	14	0.775	11	0.700	12	27	0.775	23	0.800	29	34	0.925	49	0.675	3	19	0.775
20	0.700	1	18	0.775	12	0.700	8	11	0.775	50	0.825	32	42	0.925	10	0.675	9	16	0.775
14	0.725	18	35	0.775	5	0.725	5	44	0.775	33	0.825	32	45	0.950	8	0.700	3	9	0.775
22	0.750	3	22	0.775	7	0.725	12	26	0.800	16	0.825	23	32	0.950	9	0.700	6	9	0.800
29	0.775	18	36	0.775	10	0.725	9	12	0.800	20	0.825	8	32	0.950	15	0.700	10	17	0.800
18	0.775	22	29	0.800	45	0.725	7	18	0.850	32	0.850	6	32	0.950	14	0.725	9	17	0.825

2 versus not 2																				
Medial					Central					Lateral					Top				
1 feature		2 features			1feature		2 features			1 feature		2 features			1 feature		2 features		
45	0.700	35	50	0.875	24	0.725	21	25	0.850	4	0.700	7	48	0.825	47	0.775	4	44	0.950
43	0.700	44	50	0.875	18	0.725	21	27	0.850	46	0.700	20	39	0.850	7	0.800	3	16	0.950
37	0.725	35	37	0.875	46	0.725	21	35	0.850	34	0.700	5	21	0.850	45	0.800	3	48	0.950
30	0.725	45	47	0.875	41	0.750	21	36	0.850	32	0.700	2	21	0.850	46	0.800	4	50	0.950
38	0.725	46	47	0.875	23	0.750	21	37	0.850	38	0.700	6	7	0.850	15	0.800	4	16	0.950
10	0.725	46	49	0.875	27	0.750	21	42	0.850	18	0.700	6	8	0.850	16	0.800	6	39	0.950
31	0.725	39	49	0.900	43	0.750	19	21	0.850	27	0.725	21	33	0.850	40	0.800	3	38	0.950
39	0.725	40	46	0.900	36	0.750	21	44	0.850	39	0.725	6	21	0.850	3	0.825	4	37	0.950
48	0.750	44	47	0.900	32	0.750	22	29	0.850	28	0.725	21	34	0.850	50	0.825	3	44	0.950
50	0.750	47	51	0.900	19	0.750	22	41	0.850	11	0.725	21	41	0.850	44	0.825	37	44	0.975
29	0.750	35	40	0.900	26	0.775	22	44	0.850	47	0.725	21	42	0.850	28	0.825	37	47	0.975
36	0.775	1	47	0.900	37	0.775	22	45	0.850	7	0.750	21	46	0.850	49	0.850	37	48	0.975
27	0.775	35	45	0.925	22	0.775	13	21	0.850	33	0.750	6	38	0.850	39	0.850	4	45	0.975
44	0.775	47	50	0.925	28	0.775	23	31	0.850	19	0.750	21	51	0.850	18	0.850	4	47	0.975
34	0.800	35	49	0.925	42	0.775	23	33	0.850	20	0.750	1	21	0.850	36	0.875	4	48	0.975
35	0.825	44	49	0.925	35	0.775	15	29	0.850	22	0.775	21	27	0.875	14	0.875	4	49	0.975
49	0.825	35	39	0.925	45	0.775	23	34	0.850	8	0.775	21	38	0.875	37	0.925	4	51	0.975
40	0.825	35	46	0.950	21	0.800	19	28	0.850	15	0.775	21	39	0.875	38	0.950	6	49	0.975
46	0.850	35	47	0.975	34	0.800	20	22	0.850	6	0.800	8	21	0.875	48	0.950	6	50	0.975
47	0.900	35	48	0.975	29	0.825	22	30	0.875	21	0.825	4	21	0.875	4	0.975	6	40	1.000

<table>
<tr><th colspan="20">3 versus not 3</th></tr>
<tr><th colspan="5">Medial</th><th colspan="5">Central</th><th colspan="5">Lateral</th><th colspan="5">Top</th></tr>
<tr><th colspan="2">1 feature</th><th colspan="3">2 features</th><th colspan="2">1feature</th><th colspan="3">2 features</th><th colspan="2">1 feature</th><th colspan="3">2 features</th><th colspan="2">1 feature</th><th colspan="3">2 features</th></tr>
<tr><td>13</td><td>0.725</td><td>4</td><td>31</td><td>0.875</td><td>37</td><td>0.700</td><td>2</td><td>41</td><td>0.850</td><td>3</td><td>0.600</td><td>2</td><td>35</td><td>0.675</td><td>35</td><td>0.725</td><td>1</td><td>28</td><td>0.850</td></tr>
<tr><td>16</td><td>0.725</td><td>31</td><td>36</td><td>0.875</td><td>38</td><td>0.700</td><td>2</td><td>33</td><td>0.850</td><td>25</td><td>0.600</td><td>1</td><td>33</td><td>0.675</td><td>8</td><td>0.725</td><td>3</td><td>28</td><td>0.875</td></tr>
<tr><td>23</td><td>0.725</td><td>31</td><td>37</td><td>0.875</td><td>34</td><td>0.700</td><td>30</td><td>41</td><td>0.850</td><td>21</td><td>0.600</td><td>2</td><td>49</td><td>0.675</td><td>9</td><td>0.725</td><td>28</td><td>39</td><td>0.875</td></tr>
<tr><td>21</td><td>0.725</td><td>1</td><td>31</td><td>0.875</td><td>27</td><td>0.725</td><td>1</td><td>38</td><td>0.850</td><td>7</td><td>0.600</td><td>2</td><td>36</td><td>0.675</td><td>10</td><td>0.725</td><td>21</td><td>29</td><td>0.875</td></tr>
<tr><td>6</td><td>0.750</td><td>5</td><td>31</td><td>0.875</td><td>32</td><td>0.725</td><td>30</td><td>45</td><td>0.850</td><td>34</td><td>0.600</td><td>3</td><td>33</td><td>0.675</td><td>40</td><td>0.725</td><td>28</td><td>41</td><td>0.875</td></tr>
<tr><td>7</td><td>0.750</td><td>31</td><td>38</td><td>0.875</td><td>40</td><td>0.725</td><td>1</td><td>33</td><td>0.850</td><td>29</td><td>0.600</td><td>3</td><td>45</td><td>0.675</td><td>15</td><td>0.725</td><td>24</td><td>28</td><td>0.875</td></tr>
<tr><td>27</td><td>0.750</td><td>31</td><td>39</td><td>0.875</td><td>36</td><td>0.725</td><td>33</td><td>38</td><td>0.850</td><td>36</td><td>0.600</td><td>4</td><td>33</td><td>0.675</td><td>45</td><td>0.725</td><td>2</td><td>28</td><td>0.875</td></tr>
<tr><td>48</td><td>0.750</td><td>31</td><td>41</td><td>0.875</td><td>1</td><td>0.750</td><td>42</td><td>48</td><td>0.850</td><td>15</td><td>0.600</td><td>4</td><td>45</td><td>0.675</td><td>47</td><td>0.725</td><td>9</td><td>28</td><td>0.875</td></tr>
<tr><td>25</td><td>0.775</td><td>31</td><td>43</td><td>0.875</td><td>2</td><td>0.750</td><td>42</td><td>51</td><td>0.850</td><td>39</td><td>0.600</td><td>5</td><td>33</td><td>0.675</td><td>6</td><td>0.750</td><td>28</td><td>48</td><td>0.875</td></tr>
<tr><td>30</td><td>0.775</td><td>27</td><td>31</td><td>0.875</td><td>33</td><td>0.750</td><td>1</td><td>45</td><td>0.850</td><td>22</td><td>0.625</td><td>2</td><td>45</td><td>0.675</td><td>38</td><td>0.750</td><td>28</td><td>32</td><td>0.875</td></tr>
<tr><td>32</td><td>0.775</td><td>28</td><td>31</td><td>0.875</td><td>39</td><td>0.750</td><td>43</td><td>47</td><td>0.850</td><td>30</td><td>0.625</td><td>1</td><td>45</td><td>0.675</td><td>32</td><td>0.750</td><td>28</td><td>49</td><td>0.875</td></tr>
<tr><td>9</td><td>0.775</td><td>31</td><td>45</td><td>0.875</td><td>35</td><td>0.750</td><td>2</td><td>39</td><td>0.850</td><td>46</td><td>0.625</td><td>2</td><td>46</td><td>0.675</td><td>34</td><td>0.750</td><td>28</td><td>34</td><td>0.875</td></tr>
<tr><td>46</td><td>0.775</td><td>30</td><td>31</td><td>0.875</td><td>30</td><td>0.750</td><td>2</td><td>31</td><td>0.850</td><td>8</td><td>0.650</td><td>6</td><td>32</td><td>0.675</td><td>48</td><td>0.750</td><td>28</td><td>50</td><td>0.875</td></tr>
<tr><td>8</td><td>0.800</td><td>31</td><td>46</td><td>0.875</td><td>47</td><td>0.775</td><td>45</td><td>51</td><td>0.850</td><td>35</td><td>0.650</td><td>6</td><td>40</td><td>0.675</td><td>29</td><td>0.775</td><td>28</td><td>35</td><td>0.875</td></tr>
<tr><td>14</td><td>0.800</td><td>31</td><td>47</td><td>0.875</td><td>41</td><td>0.775</td><td>1</td><td>41</td><td>0.875</td><td>28</td><td>0.650</td><td>6</td><td>45</td><td>0.700</td><td>33</td><td>0.775</td><td>21</td><td>26</td><td>0.875</td></tr>
<tr><td>15</td><td>0.800</td><td>31</td><td>35</td><td>0.875</td><td>42</td><td>0.775</td><td>1</td><td>34</td><td>0.875</td><td>40</td><td>0.650</td><td>3</td><td>51</td><td>0.700</td><td>49</td><td>0.800</td><td>28</td><td>45</td><td>0.900</td></tr>
<tr><td>29</td><td>0.800</td><td>31</td><td>49</td><td>0.875</td><td>51</td><td>0.775</td><td>41</td><td>51</td><td>0.875</td><td>1</td><td>0.675</td><td>2</td><td>48</td><td>0.700</td><td>31</td><td>0.800</td><td>28</td><td>46</td><td>0.900</td></tr>
<tr><td>31</td><td>0.825</td><td>2</td><td>31</td><td>0.900</td><td>31</td><td>0.775</td><td>2</td><td>42</td><td>0.900</td><td>33</td><td>0.675</td><td>1</td><td>2</td><td>0.725</td><td>27</td><td>0.825</td><td>28</td><td>47</td><td>0.900</td></tr>
<tr><td>47</td><td>0.825</td><td>6</td><td>30</td><td>0.925</td><td>45</td><td>0.800</td><td>2</td><td>45</td><td>0.900</td><td>45</td><td>0.675</td><td>2</td><td>51</td><td>0.725</td><td>41</td><td>0.825</td><td>28</td><td>36</td><td>0.900</td></tr>
<tr><td>1</td><td>0.900</td><td>6</td><td>31</td><td>0.925</td><td>46</td><td>0.825</td><td>2</td><td>40</td><td>0.900</td><td>2</td><td>0.800</td><td>1</td><td>3</td><td>0.750</td><td>28</td><td>0.875</td><td>28</td><td>37</td><td>0.900</td></tr>
</table>

4 versus not 4																			
Medial					Central					Lateral					Top				
1 feature		2 features			1feature		2 features			1 feature		2 features			1 feature		2 features		
22	0.600	2	11	0.800	33	0.625	2	49	0.925	43	0.750	4	23	0.875	45	0.625	7	16	0.725
33	0.600	11	35	0.800	18	0.625	3	51	0.925	39	0.750	1	23	0.875	33	0.625	31	43	0.725
20	0.600	11	38	0.800	12	0.650	4	8	0.925	47	0.750	22	27	0.875	31	0.625	2	40	0.725
3	0.625	11	28	0.800	35	0.650	2	43	0.925	18	0.750	2	49	0.875	17	0.625	31	47	0.725
45	0.625	11	29	0.800	20	0.650	5	9	0.925	2	0.775	22	34	0.875	48	0.625	2	34	0.725
46	0.625	11	17	0.800	48	0.675	2	44	0.925	45	0.775	22	35	0.875	26	0.625	1	34	0.725
9	0.625	1	11	0.800	49	0.675	1	46	0.925	24	0.775	23	27	0.875	2	0.650	36	38	0.750
31	0.650	11	39	0.800	25	0.700	2	30	0.925	28	0.775	2	23	0.875	43	0.650	36	39	0.750
5	0.650	11	40	0.800	6	0.750	6	9	0.925	21	0.775	5	23	0.875	35	0.650	27	36	0.750
10	0.650	11	30	0.800	3	0.775	2	48	0.950	35	0.775	23	32	0.875	46	0.650	36	41	0.750
16	0.650	11	31	0.800	7	0.775	2	50	0.950	48	0.775	23	46	0.900	39	0.650	37	40	0.750
17	0.650	11	32	0.800	5	0.800	1	47	0.950	41	0.775	36	46	0.900	47	0.650	31	35	0.750
44	0.675	3	11	0.800	10	0.800	2	32	0.950	19	0.775	36	48	0.900	41	0.675	38	46	0.750
12	0.675	11	33	0.800	11	0.800	2	34	0.950	49	0.800	36	49	0.900	34	0.675	38	47	0.750
14	0.675	10	13	0.800	51	0.800	2	42	0.950	22	0.800	37	48	0.900	40	0.675	31	40	0.750
15	0.675	11	41	0.800	1	0.850	2	4	0.950	38	0.800	2	50	0.900	49	0.700	1	49	0.750
4	0.700	11	42	0.800	4	0.850	4	51	0.950	20	0.800	1	37	0.900	16	0.700	31	45	0.750
13	0.700	5	11	0.825	9	0.850	2	46	0.950	37	0.825	23	28	0.900	32	0.700	31	48	0.750
43	0.700	4	11	0.825	8	0.875	2	47	0.950	23	0.875	36	41	0.925	36	0.725	36	40	0.775
11	0.800	6	11	0.825	2	0.900	1	4	0.975	36	0.875	37	49	0.925	38	0.725	36	49	0.775

5 versus not 5																			
Medial					Central					Lateral					Top				
1 feature		2 features			1feature		2 features			1 feature		2 features			1 feature		2 features		
19	0.600	9	27	0.725	42	0.625	4	17	0.750	5	0.675	7	50	0.850	24	0.600	21	32	0.750
24	0.600	6	14	0.725	18	0.625	2	19	0.750	26	0.675	9	10	0.875	28	0.600	10	21	0.750
3	0.625	13	14	0.725	36	0.650	11	18	0.750	38	0.675	9	11	0.875	6	0.600	21	33	0.750
5	0.625	14	33	0.725	49	0.650	11	20	0.750	22	0.700	9	27	0.875	45	0.600	21	34	0.750
6	0.625	14	34	0.725	40	0.650	12	16	0.750	21	0.700	3	9	0.875	11	0.600	2	21	0.750
26	0.625	14	35	0.725	20	0.650	1	45	0.750	40	0.700	9	32	0.875	47	0.600	21	35	0.750
8	0.625	3	27	0.725	3	0.675	17	19	0.750	4	0.700	1	9	0.875	14	0.600	21	36	0.750
40	0.625	14	44	0.725	4	0.675	17	20	0.750	14	0.700	9	35	0.875	15	0.600	21	37	0.750
50	0.625	8	9	0.725	44	0.675	18	19	0.750	23	0.700	9	37	0.875	48	0.600	7	21	0.750
1	0.650	1	14	0.725	46	0.675	18	20	0.750	18	0.700	9	38	0.875	17	0.600	21	26	0.750
28	0.650	8	14	0.725	29	0.675	1	51	0.750	20	0.700	9	47	0.875	18	0.600	8	21	0.750
27	0.650	14	45	0.725	41	0.675	19	39	0.750	6	0.725	9	48	0.875	22	0.625	21	38	0.750
13	0.650	14	46	0.725	16	0.675	19	40	0.750	25	0.725	4	9	0.875	51	0.625	21	28	0.750
15	0.650	9	14	0.725	15	0.700	32	51	0.750	15	0.725	9	49	0.875	25	0.625	21	29	0.750
16	0.650	14	47	0.725	1	0.725	2	45	0.775	11	0.750	9	50	0.875	46	0.625	21	39	0.750
51	0.675	14	48	0.725	2	0.725	1	31	0.775	24	0.750	5	8	0.875	13	0.625	21	40	0.750
18	0.675	10	14	0.750	47	0.725	6	17	0.775	10	0.775	2	9	0.875	20	0.625	21	41	0.750
25	0.700	14	23	0.750	51	0.725	15	19	0.775	8	0.825	9	51	0.875	27	0.650	21	30	0.750
9	0.700	9	23	0.750	17	0.725	19	36	0.775	7	0.850	9	23	0.900	19	0.675	21	31	0.750
14	0.725	45	51	0.775	19	0.725	49	51	0.800	9	0.875	7	10	0.900	21	0.750	1	21	0.750

Appendix 2 - Computer Codes

A number of computer codes were developed in the course of this research. These codes run in the MATLAB environment and include the following:

KMEANS: This code runs the K-means algorithm on matrices with arbitrary numbers of features and for arbitrary numbers of clusters.

NEIGHBOR: This code runs the nearest neighbor algorithm on matrices with arbitrary numbers of features for a user specified t.

KNEIGH: This code runs the nearest neighbor algorithm on matrices with arbitrary numbers of features for a user specified number of clusters. The code searches for a t that will produce the given number of clusters.

METC and METSQER: These codes provide metrics for helping to judge the success of clustering. METC computes the metric discussed in section 3.3. METSQER computes the square error (sum of the squares of the distances from each point to its cluster center) for each cluster and the average square error (square error divided by number of points in cluster) for each cluster. It also computes the average square error across all clusters.

SRCHITER: This code automates the use of kmeans and kneigh when the user wishes to run the codes repeatedly on sets of features selected from high dimensional data. For instance, the code could be used to run K-means on 10-dimensional data where the features are being considered 2 at a time and the results are to be judged using the metc algorithm.

Copies of these computer codes, including a MATLAB graphical user interface (GUI) that provides access to some of these tools, may be obtained by contacting

Professor Benjamin Knapp at San Jose State University's Department of Electrical Engineering.