ABSTRACT

A KINEMATIC ANALYSIS OF HAND CONFIGURATIONS IN STATIC AND DYNAMIC FINGERSPELLING

by
Gillian B. Sherry

The focus of this study was the investigation of target handshapes in American Sign Language fingerspelling in order to determine whether there was a difference between static canonical structures and structures produced in the context of a movement sequence. This was achieved by measuring the joint angles of a signing hand with an 18-sensor *Cyber*Glove® by Virtual Technologies, Inc.

A discriminant analysis was used to identify targets that occurred at points of minimum angular joint velocity. A multivariate analysis of variance with planned comparisons was then applied to these dynamic data along with the static data to test the hypothesis.

The results showed that there was a significant difference between handshapes produced statically and those produced dynamically, which suggested that a simple, cipher model of static handshapes produced within the context of a movement sequence is not sufficient to account for the production and perception of fingerspelling. These findings may be applied to future research in sign language recognition, so that consideration of the variability of target handshapes, as influenced by the spatiotemporal environment, might be incorporated into future models.

A KINEMATIC ANALYSIS OF HAND CONFIGURATIONS IN STATIC AND DYNAMIC FINGERSPELLING

by
Gillian B. Sherry

A Thesis
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Biomedical Engineering

Department of Biomedical Engineering

May 2004

APPROVAL PAGE

A KINEMATIC ANALYSIS OF HAND CONFIGURATIONS IN STATIC AND DYNAMIC FINGERSPELLING

Gillian B. Sherry

Dr. Richard Foulds, Thesis Advisor Associate Professor of Biomedical Engineering, NJIT	Date
Dr. William Hunter, Committee Member Professor of Biomedical Engineering, NJIT	Date
Dr. Stanley Reisman, Committee Member Professor of Biomedical Engineering, NJIT	Date

Blank Page

BIOGRAPHICAL SKETCH

Author: Gillian B. Sherry

Degree: Master of Science

Date: August 2004

Graduate and Undergraduate Education:

Master of Science in Biomedical Engineering
New Jersey Institute of Technology, Newark, New Jersey, 2004

Bachelor of Science in Electrical & Computer Engineering Technology
New Jersey Institute of Technology, Newark, New Jersey, 2002

Major: Biomedical Engineering

Presentations and Publications:

Sherry, Gillian and Foulds, Richard, "Pattern Recognition Considerations for Continuous Sign Language Recognition", Proceedings of the 29th Northeast Bioengineering Conference, Newark, NJ, March 2003.

To Jacqueline Pridgen Feeney

agus

I gcuimhne Kathleen O'Malley Sherry, Beannacht Dé lena hanam

~~~~~~

Education is not the filling of a pail, but the lighting of a fire.
-William Butler Yeats
~~~~~~

ACKNOWLEDGEMENT

There are many people to whom I am indebted and would like to acknowledge. I am most grateful to Dr. Richard Foulds, my advisor, for giving me the opportunity to pursue research in this very interesting field, and for always being helpful, enthusiastic and supportive of my work. Special thanks to Dr. William Hunter and Dr. Stanley Reisman for actively participating on my committee.

I am very grateful to the Millbank Foundation, the National Science Foundation and the Pfeiffer Foundation for supporting this study, which was conducted in the Neuromuscular Engineering Research Laboratory at NJIT.

I would like to extend particular thanks to my fellow graduate students in the Neuromuscular Engineering Laboratory (Jerome Allen, Corey Birmingham, Don Helt III, Bruno Mantilla, Matthew Noesner and Darnell Simon), whose friendship and humor made the experience much more enjoyable. My thanks also to John Davidson, who graciously helped with the data collection.

My sincerest appreciation to my many teachers throughout my formative years in Limerick for providing me with a strong background, not only in the math and science areas, but also in the arts and literature, without which the pieces of the puzzle could not come together.

Finally, I wish to express my deepest gratitude to my family for their constant encouragement. My sincere thanks to my parents Eileen E. and Francis T. Sherry for their support and sacrifices over the years. My deepest appreciation to my sisters Diane M. Sherry, Keira F. Baheti, and brother-in-law, Umesh S. Baheti, for their friendship.

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES
(Continued)

LIST OF FIGURES
(Continued)

CHAPTER 1

INTRODUCTION

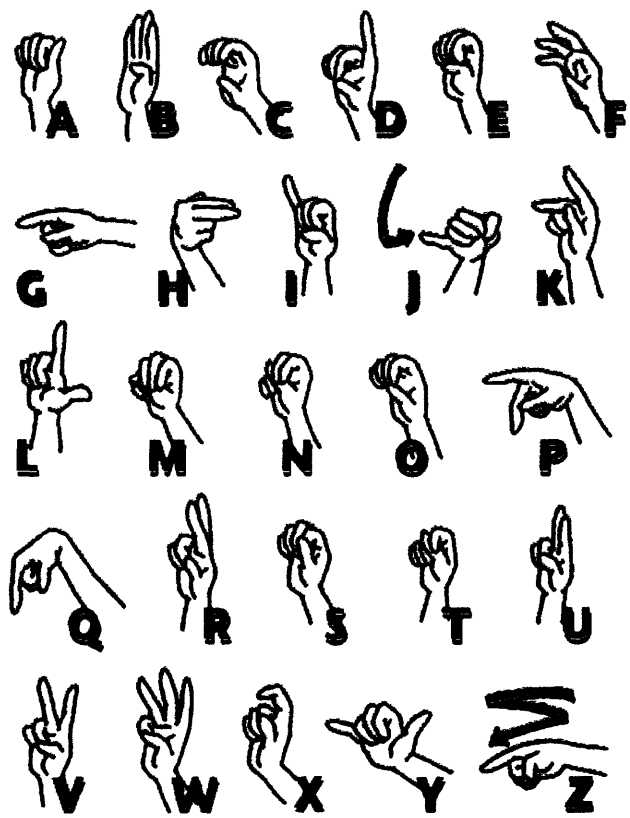
1.1 Background

The visual-gestural languages of the Deaf, their labored history of evolution, their intricacy and richness of linguistic structure, as well as their expression of a unique and challenged culture, have provided a vast array of disciplines with a unique insight into the workings of the human brain. It is through this window of gesture that scientists can explore some of the puzzles of language and cognition and take one step closer to resolving the complex mystery that is man. Research in linguistics, neurophysiology and biomechanics are among a few of the interrelated disciplines that have used signed languages to investigate such topics as phonetics, motor control and sensorimotor integration where the underlying strategies of sign organization, production and perception provide a structured basis for examination.

Natural language may be represented by the three different modes of speech, writing and gesture, and its study must include both cross-modal and cross-linguistic data in order to ensure accuracy. Particular emphasis has been placed on signed languages over the last couple of decades for this very reason, with the most extensive research having been conducted on American Sign Language (ASL). This is the native language of the North American Deaf community where it is the primary mode of communication and is used by more than a half a million people throughout the United States and Canada. This visual medium is composed of signs but also of fingerspelling where the latter represents the manual alphabet of the signed language.

1.2 American Sign Language

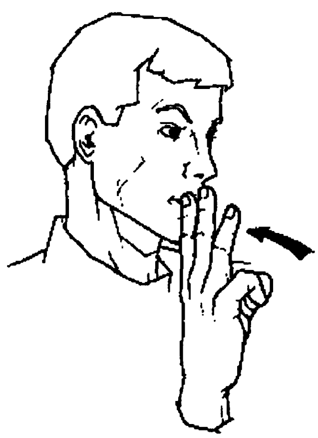
It is important to note that ASL is a natural, autonomous language that is transmitted in signed modality and not merely a medium though which the English language is expressed. There exist regional, social and situational dialects, just as can be found in any spoken language. Complex abstract thought, intellectual argument, wit and poetry are all capable of expression. Fingerspelling, on the other hand, is a tertiary system, which represents the written form of a spoken language, and the American manual alphabet contains the same twenty-six letters as the orthographic form of English. Consequently, it is integrated into ASL, where it is used for transliteration of English words by spelling one letter at a time.



Source: http://www.deafmissions.com/dic/ASLabc.html

Figure 1.1 Manual alphabet of American Sign Language.

Fingerspelling is used to communicate words for which no specific unitary sign exists such as proper names, place names, and new or technical words not yet incorporated into the sign lexicon. Furthermore, it is fully integrated into the sign vocabulary to the extent that hand configurations for some letters actually form the basis for certain signs while incorporating a variation in a particular parameter(s). The sign for "Monday", for example, consists of the letter "M" oriented towards the body and moved in a circular motion, and "water" is signed by forming the letter "W" with the right hand and tapping the mouth with the index finger.



Source: www.lifeprint.com/asl101/pages-signs/w/water.htm

Figure 1.2 ASL sign for "water".

The phonological structure of signs consists of five parameters: hand configuration and orientation, movements of the hands and arms, places of articulation, and non-manual gestures such as body language and facial expression (Stokoe). It is through variation of these parameters that lexical signs are differentiated and communication is achieved. The parameters of fingerspelling are handshape and movement, and transformations of the other features are irrelevant.

1.3 Models of Fingerspelling

The traditional, elementary model of fingerspelling views word production as the sum of its individual letters, or "beads on a string" (Hardcastle and Hewlett), with simple letter-handshape correspondence. However, although the organization of fingerspelling is, by nature, a concatenation of discrete elements, the overall perception of fingerspelling is that of a continuous stream when articulated fluently and in quick succession. This is reflected in Akamatsu's model of fingerspelling which builds on the basic, cipher model by approaching word production and perception as a "movement envelope" whose shape is dictated by the hand configurations of the individual letters, so changes in letters bring about changes in the overall envelope shape. Hanson's study on fingerspelling also found that subjects used an underlying structure to perceive words rather than reading individual letters. Similarly, Zakia and Haber note, "In reading fingerspelling words, a highly experienced reader is not attending to the individual letters, but rather to the total pattern of finger configuration, or at least enough to identify the word." For a non-signer, the best analogy will result from observing the following picture.

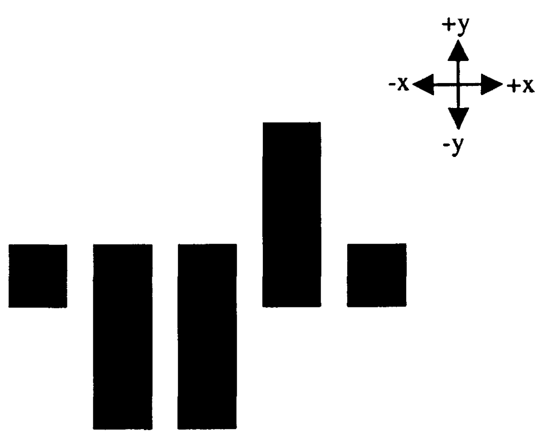


Figure 1.3 Envelope of a written word.

If asked to identify the word represented by the overall shape, rather than examining the individual components in sequence (which in this case would be futile anyway), one might easily read the word "apple." This is interpreted by relying solely on the envelope of the written word. Moreover, it is the relative sizes and locations of the blocks that contribute to the shape of the envelope and are critical to the overall perception of the word. Notice how the two blocks in the middle extend equally in length in the negative y-direction. This could be considered the result of assimilation, which is the tendency of distinguishing features of different components, in this case the length of the letter blocks, to become more like each other, in order to aid in comprehension. Conversely, the extension of the fourth block in the positive y-direction could be attributed to dissimilation of length, whereby distinguishing features become less like each other to facilitate recognition.

Another example of assimilation and dissimilation can be found in the form of caricatures. Here, the characterizing features that are necessary for differentiation are emphasized so that the overall impression is the exaggeration (and blending) of facial features pertinent to recognition. In the same way, it is believed that fingerspelling recognition is largely based on a similar phenomenon except that the envelope is created in the spatiotemporal environment. It is not only the articulatory targets of the hand configurations, but the corresponding transitional movements implicit in Akamatsu's model that contribute to the production and comprehension of a word. This emphasis on transitions is central to Wilcox's dynamic model of fingerspelling which explores the underlying structure that contributes to the coordination and cooperation of articulators in executing a task.

Source: http://www.dimpleart.com

Figure 1.4 Caricatures demonstrating assimilation and dissimilation.

1.4 Hypothesis

In accordance with the model of fingerspelling that is based on targets and transitions, this thesis examined the structure of targets produced in different circumstances and explored the influence of surrounding transitional variables. Targets were characterized in terms of the configurations of the articulators (fingers of the hand) occurring at time points when the overall velocity of the fingers was at a minimum. The handshapes formed by configuring the finger positions were determined by measuring certain joint angles of the signing hand when producing letters in isolation and when executing movement sequences. Admittedly, this was a simplification since the independent motor subsystems, of which the articulatory targets of fingerspelling comprise, also include movement, which is important for differentiating between particular targets, such as the hand configurations for “I” and “J.”

The hypothesis of this thesis is that the target handshapes towards which articulators move are not the canonical structures found when letters are produced statically, but rather are modified versions of the canonical forms, with variable ranges that adjust according to the influence of kinematical and temporal variations in surrounding letters. The study was designed to investigate this hypothesis by examining and comparing the production of hand structure, one of the main motor subsystems of ASL. The anticipated target modification is a direct result of a number of phenomena including assimilation and dissimilation, as mentioned previously. It can also be attributed to another effect that is prominent in fingerspelled and signed words, known as coarticulation, which has its roots in speech and has been investigated since as early as the 19th century by various researchers including Sweet in 1877 and Jespersen in 1897.

1.5 Coarticulation

It is a well-known fact that coarticulation in the auditory medium is manifested when the articulation of a given phoneme partially overlaps in time with the articulation of immediately adjacent speech sounds, thereby affecting its acoustic and articulatory properties. As with most motor activities in which a certain amount of fluency and coordination is required, signed language also displays coarticulatory influences in articulator production. This reflects the neuromuscular production of articulator movement and results from a kinematic variation in the movement of these articulators, as determined by the kinematic characteristics of surrounding elements in the movement sequence.

Reich described the bidirectional effects of coarticulation (carry-over and anticipatory) in fingerspelling as a type of phonological restructuring, which can manifest itself as forward or backward assimilation. The very existence of such a phenomenon refutes the notion of targets in fingerspelling as invariant, static, canonical structures.

This study is an advance over Akamatsu's movement envelope which was primarily based on the sequential transmission of static handshapes, and builds on Wilcox's model of dynamic fingerspelling to propose that the handshapes produced serially are in fact different to their corresponding static formations.

Although previous research has explored the presence and extent of coarticulation in the form of assimilation and dissimilation (Jerde et al.), few studies, if any, have addressed the deviation from canonical forms that results from coarticulation. This question is taken up here through analyses described in the following chapters.

CHAPTER 2

METHODOLOGY

2.1 Overview

The main goal of this study was to compare the target configurations produced along an articulatory trajectory to the same target configurations produced statically, in order to determine whether the static formations remained invariant when produced in context. The trajectories selected were based on a study of coarticulation in fluent fingerspelling, where the stimuli consisted of a set of letter strings, comprising ten words and ten non-words, that were performed by sign language interpreters wearing a *Cyber*Glove® on the signing hand. The 20 strings were distributed equally amongst four distinct categories of same initial letter words, same initial letter non-words, different initial letter words, and different initial letter non-words. Each string sequence contained the trigram I-S-C followed by a vowel. The primary question was whether there was a statistical difference between static and contextual configurations of the abutting letters in the trigram.

Table 2.1 Letter String Categories

Same initial letter	Different initial letter
Words	*Words*
D<u>ISC</u>ARD	CONF<u>ISC</u>ATE
D<u>ISC</u>ERN	V<u>ISC</u>ERAL
D<u>ISC</u>IPLE	OMN<u>ISC</u>IENT
D<u>ISC</u>OVER	PER<u>ISC</u>OPE
D<u>ISC</u>USS	B<u>ISC</u>UIT
Non-words (pronounceable)	***Non-words (not pronounceable)***
BR<u>ISC</u>ANT	QWKX<u>ISC</u>AVB
BR<u>ISC</u>ENSE	GDT<u>ISC</u>EXI
BR<u>ISC</u>IARD	XBTK<u>ISC</u>IGR
BR<u>ISC</u>OZE	HVZS<u>ISC</u>OWJ
BR<u>ISC</u>UDGE	PCLM<u>ISC</u>UDJ

2.2 Experimental Task

Two professional sign language interpreters participated in this study which was approved by the Institutional Review Board at the New Jersey Institute of Technology. Both were female, right-handed with normal hearing, and were proficient in ASL. The subjects were presented with a visual representation of each of the 26 letters in the English alphabet or each of the 20 strings in random order, and were asked to produce the appropriate sign(s). They were instructed to hold the static letters for the duration that the cue remained on the screen and to sign the letter strings in a consistent manner. Because signs for letters are unimanual, this required that the data from one hand be collected. Due to the limited combination of finger positions, many of the signs are very similar and differ only by slight variations in finger position or hand orientation.

Figure 2.1 Data acquisition program interface.

Each trial began with the subject's elbow in a neutral resting position. A break period was provided between the trials, as needed. Upon receiving a letter or string prompt on the lower right corner of the screen, the subject produced the sign(s) representing the letter(s). A total of 20 series were collected for each subject (ten discrete alphabet blocks of 26 letters and ten continuous string blocks of 20 words/non-words). The order of the trials was randomly permuted and cues were presented to the subject one at a time and remained on the screen for several seconds. The intention of the random selection was to minimize anticipatory coarticulation that may have occurred if the signs were produced in consecutive or predictable order. Extra time was incorporated to ensure that samples from the static trials could be easily segmented into their corresponding letters. Letters within strings would be recognized using an automatic recognition procedure which will be discussed later in detail.

2.3 Calibration

It was necessary to calibrate the glove to the subjects' hands due to the range of variability amongst hand sizes and ranges of motion. This was achieved with Virtual Technologies' *Device Configuration Utility* (DCU) software program. In order to convert to appropriate joint angles from raw A/D sensor values the following equation was applied,

$$Angle = Gain * (A/D\ value - Offset)$$

Gain and offset values were set during calibration and were based on a default configuration of hand-geometry in the calibration file. While this hand-geometry portion of the file contained baseline hand structure information, the hand-calibration portion contained the gain and offset parameters required to map the digitized values to the joint angles of the default hand structure. The gain affected the range of joint angles and the offset referred to the difference between the A/D values and the default hand-geometry position (Virtual Technologies, Inc).

Calibration consisted of two phases where each subject was instructed initially to rest the hand with the *Cyber*Glove® flat on the table, and then instructed to make an 'OK' sign and hold the position. The DCU then calibrated the hand to the default file and created a unique calibration for each subject based on her specific hand structure. The following figures illustrate the screens that prompted for glove calibration.

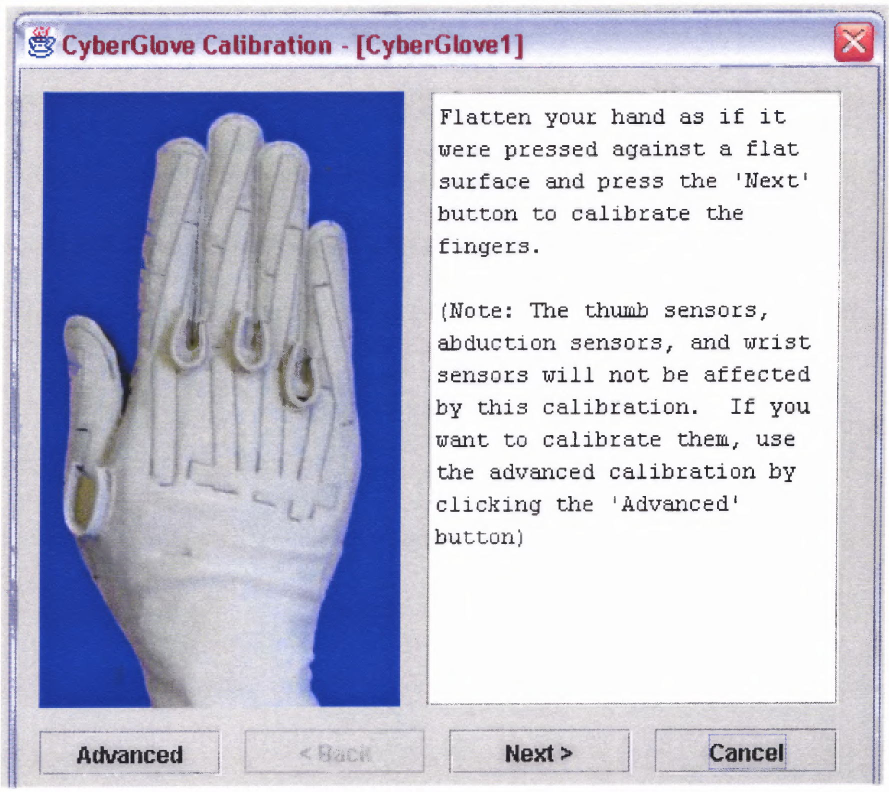


Figure 2.2 *Cyber*Glove® calibration, part I.

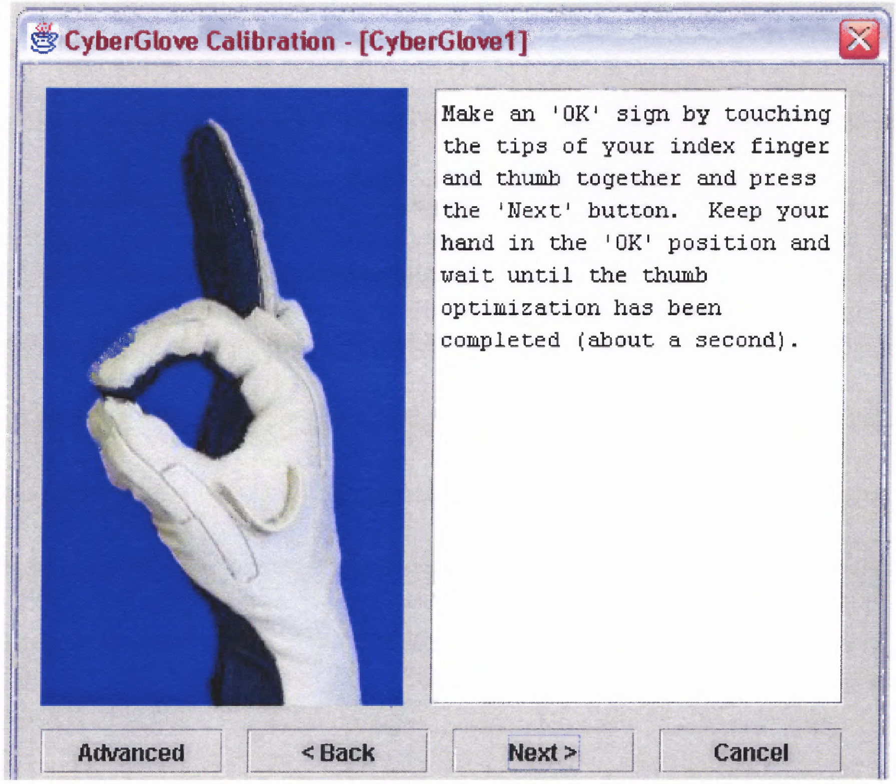


Figure 2.3 *Cyber*Glove® calibration, part II.

2.4 Data Collection

The first requirement in handshape recognition is data acquisition in order to capture the gestures that are made as signs. This project used an instrumented glove as a transducer rather than a camera-based device because gloves allow for concise and accurate data extraction whereas certain data, such as hand orientation, forward-backward motion and finger position information, are difficult to extract from visual images.

Source: Martin, J., "A Linguistic Comparison: Two Notation Systems for Signed Language: Stokoe Notation & Sutton SignWriting."

Figure 2.4 Orientation of finger position.

A right-handed *Cyber*Glove® by Immersion 3D Interaction Corporation (formerly known as Virtual Technologies, Inc) was used to record high-accuracy digital joint angle information. Sensor data were captured at a rate of 55 Hz and were stored on a PC hard disk for off-line analyses. Although 22-sensor *Cyber*Gloves® with covered fingertips are commercially available, as shown in Figure 2.6, the *Cyber*Glove® used in this study was open-fingered and did not possess sensors on the most distal joints of the pinkie through index fingers. It did contain the remaining 18 embedded sensors - two bend sensors on each finger, four abduction sensors, plus sensors measuring thumb crossover, palm arch, wrist flexion and wrist abduction.

The thumb of the *Cyber*Glove® had two bend sensors, which measured the metacarpophalangeal and interphalangeal joints. The metacarpophalangeal joint is located at the point where the thumb connects to the palm and the interphalangeal joint is located between the thumb's proximal and distal phalanges. The remaining four fingers of the *Cyber*Glove® also contained two embedded sensors which measure the metacarpophalangeal and proximal interphalangeal joints. As with the metacarpophalangeal joint sensor of the thumb, these sensors also measured the points at which the fingers connect to the palm. The proximal interphalangeal joints of the fingers are found between the proximal and middle phalanges. The thumb had an additional sensor which measured the degree of rotation of the thumb about the palm. The point of rotation is a saddle point at base of thumb and is called the trapeziometacarpal joint. These joints are illustrated in right-hand side of Figure 2.6 which shows the posterior view of the hand.

An additional sensor was provided on the pinkie finger, which measured the degree of rotation of the pinkie across the palm. This indicates the arch of the palm formed when the hand is in a cupped position. Sensors on the wrist measured pitch, which is the flexion/extension of the wrist in the vertical plane, and wrist yaw, which is the abduction/adduction or sideways flexion of the wrist. Finally, abduction sensors measured the lateral movement of the thumb-index, middle-index, ring-middle, and pinkie-ring fingers.

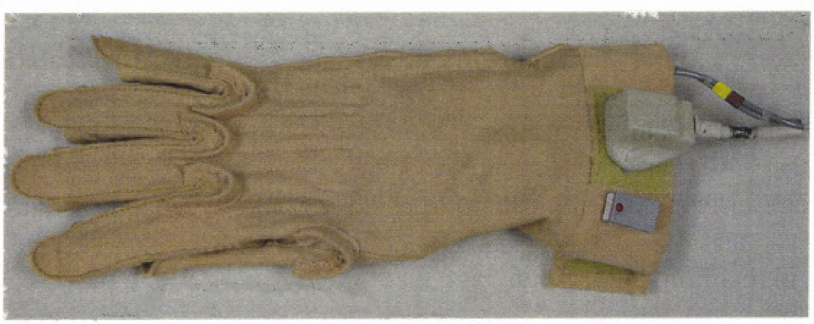


Figure 2.5 *Cyber*Glove® used for data collection.

In sum, these bend sensors measured the following 18 degrees of freedom: the metacarpophalangeal joint angles (MPJ – sensors 2, 5, 7, 10, 13) of the thumb and each of the 4 fingers; the proximal interphalangeal joint angles (PIJ – sensors 6, 8, 11, 14) of each of the 4 fingers; the interphalangeal joint angle (IJ – sensor 3) of the thumb; abduction (ABD – sensors 4, 9, 12, 15) of the thumb-index, middle-index, ring-middle, and pinkie-ring fingers; the trapeziometacarpal joint angle (TMJ – sensor 1) of the thumb; palm arch (PA – sensor 16); wrist pitch (WP – sensor 17); and wrist yaw (sensor 18) with a spatial resolution of 1°, and a temporal resolution of 18ms.

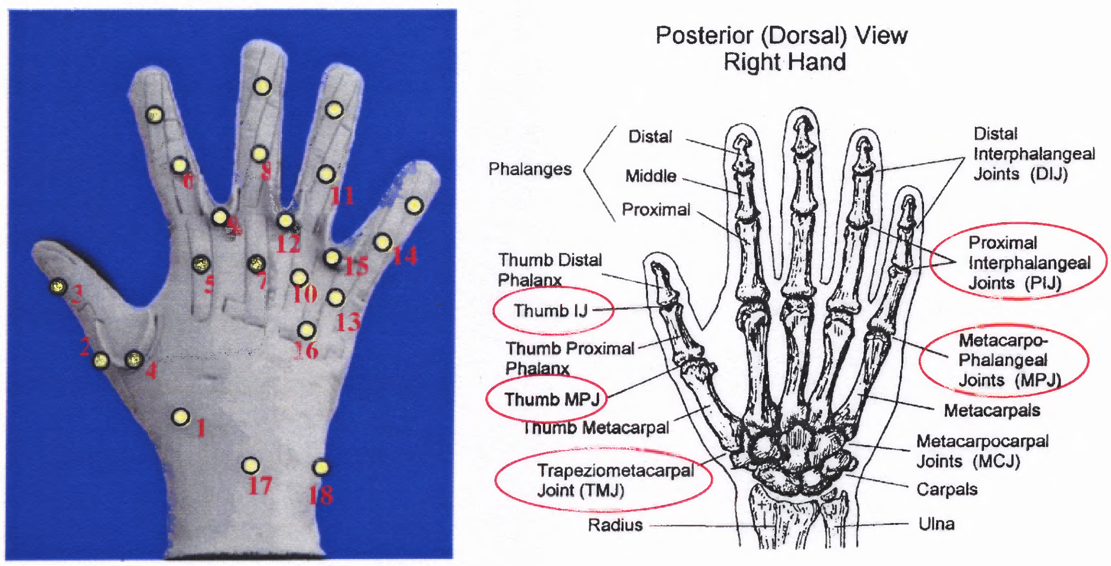


Figure 2.6 *Cyber*Glove® with embedded sensor locations (left) and posterior view of the right hand (right).

The raw analog data were converted to a digital form that could be analyzed. This was accomplished by the data acquisition program, which produced an ASCII output file that was a description of this sequence of features (18 joint angles) and was displayed in degrees. The ASCII file then became the input pattern matrix to the classifier. The angle headers were: TTMJ, TMPJ, TIJ, TABD, IMPJ, IPIJ, MMPJ, MPIJ, MABD, RMPJ, RPIJ, RABD, LMPJ, LPIJ, LABD, PA, WP, and WY, as described in Table 2.2.

Table 2.2 Sensor Descriptions

Sensor	Description
1	Thumb TMJ
2	Thumb MPJ
3	Thumb IJ
4	Thumb-index ABD
5	Index MPJ
6	Index PIJ
7	Middle MPJ
8	Middle PIJ
9	Middle – index ABD
10	Ring MPJ
11	Ring PIJ
12	Ring – middle ABD
13	Pinkie MPJ
14	Pinkie PIJ
15	Pinkie – ring ABD
16	Palm arch PA
17	Wrist pitch WP
18	Wrist yaw WY

In order to facilitate the goal of the project, which was the comparison of static and contextual hand configurations for the specific letters in the trigram (I-S-C), it was necessary to compile feature vectors corresponding to each of these letters from the static alphabet data and from the dynamic string data. Static postures were defined by calculating the moving average of the last 60 samples of each trial (approximately the last second). Dynamic postures were defined using an automatic recognition procedure based on minimum angular joint velocities put forth by Jerde, Soechting and Flanders.

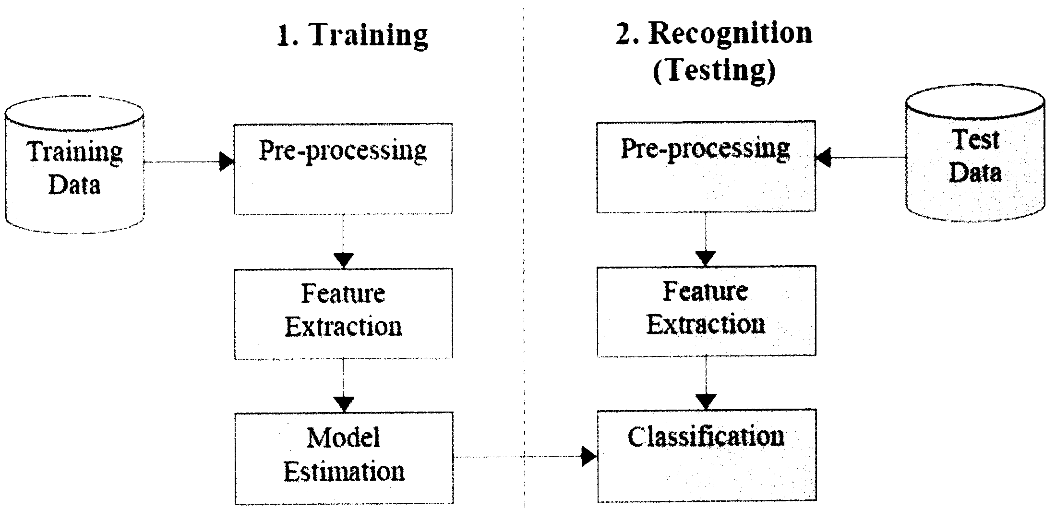
The static letters were manually segmented and extracted. Each letter occurred once in random order in every series. Due to equipment error, only seven usable series of static alphabet were acquired for Subject 1. All ten of Subject 2's alphabet was usable. For each series produced by the subjects, the moving average vectors of the last 60 samples of each of the 26 letter trials were computed and combined to form one static feature matrix for each subject. Let each of the sample averages be represented by an 18-dimensional vector, $x_k = \{x_1, x_2, x_3, \ldots x_{18}\}$ where 18 is the number of joint angles. Let $\{x_1, x_2, x_3, \ldots x_N\}$ be a set of N series for each class, where each member is represented by x_k^i ($k=1, 2, \ldots N_i$, N_i is the number of samples in class i). Therefore N = 7 for Subject 1 and $x_k^i = \{x_1^A, x_2^A, x_3^A, x_4^A, x_5^A, x_6^A, x_7^A, x_1^B \ldots, x_7^Z\}$ and N = 10 for Subject 2 and $x_k^i = \{x_1^A, x_2^A, x_3^A, x_4^A, x_5^A, x_6^A, x_7^A, x_8^A, x_9^A, x_{10}^A, x_1^B, x_2^B \ldots, x_{10}^Z\}$.

10:54:06 AM	Thumb TMJ	Thumb MPJ	Thumb IJ	Thumb Abd.	Index MPJ	Index PIJ	Middle MPJ
10:54:08 AM	107.4708	-61.0114	17.26952	28.18723	-11.4483	29.02604	29.95194
10:54:08 AM	107.4708	-61.0114	17.26952	28.18723	-11.4483	29.02604	29.95194
10:54:08 AM	107.4708	-61.0114	17.26952	28.18723	-11.4483	29.02604	29.95194
10:54:08 AM	107.4708	-61.0114	17.26952	28.18723	-11.4483	29.02604	29.95194
10:54:08 AM	107.4708	-61.0114	17.26952	28.18723	-11.4483	29.02604	29.95194
10:54:08 AM	107.4708	-61.0114	17.26952	28.18723	-11.4483	29.02604	29.95194
10:54:08 AM	107.4708	-61.0114	17.26952	28.18723	-11.4483	29.02604	29.95194
10:54:08 AM	107.4708	-61.0114	17.26952	28.18723	-11.4483	29.02604	29.95194
10:54:08 AM	107.4708	-61.0114	17.26952	28.18723	-11.4483	29.02604	29.95194

Figure 2.7 Sample of the first seven features of an initial trial.

2.5 Pattern Recognition

The process of automatic letter recognition, necessary to define dynamic postures, involved several steps including the pre-processing and feature extraction of the training and testing data sets, model estimation of the training data and classification of the testing data using a linear discriminant analysis (LDA). The training set for each subject was compiled from the joint angles retrieved from the static block. Subject 1 had 7/10 successful trials (with three trials discarded due to equipment error) and Subject 2 had 10/10 successful trials yielding (176x18) and (260x18) feature matrices respectively for the training sets. Pre-processing was initiated in the CLASSIFYSTATIC function by filtering the data at 10Hz using a fourth-order bi-directional Butterworth filter as outlined in the next section. Feature extraction involved isolating the *Cyber*Glove® data from other movement data that had been collected. The start of a movement was taken 25 frames after the cue was initially presented and the next 275 frames were automatically analyzed.



Source: Hansen, J., "A Matlab Project in Optical Character Recognition (OCR)."

Figure 2.8 Pattern classification process.

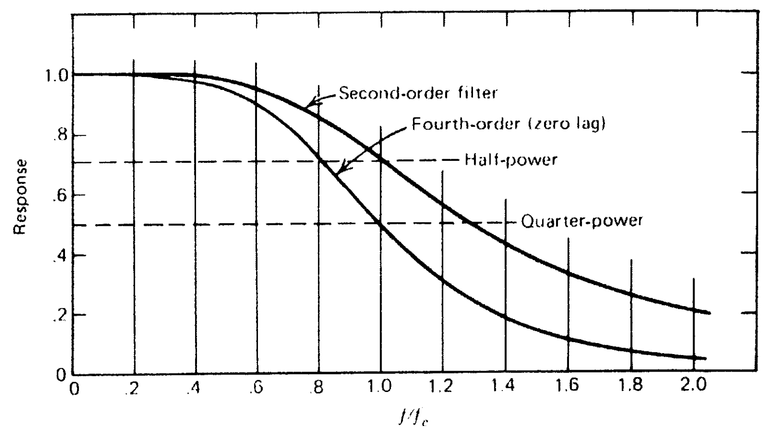
The function then set up for the model estimation and classification of the LDA. The goal of the pattern classification algorithm was to classify joint angles, produced within a string, to letters in the alphabet. Each of the 26 letters corresponded to a different class. The feature matrix was labeled according to the particular class to which the joint angle vectors belonged. An LDA was then performed on the recorded trajectory for classification. The Matlab function for this procedure, CLASSIFYSTATIC, can be referenced in Appendix A.

2.6 Pre-Processing

Before the raw data could be effectively analyzed, a certain amount of pre-processing was required. Pre-analytic processing functions such as analog-to-digital conversion, data compression, and data formatting were performed to the sensor data by the data acquisition program on the front-end for the purposes of enhancing data utilization. Once the raw data was in the format indicated in Figure 2.7, a smoothing operation was performed whereby the signal was passed through a filter to remove any additive noise that may have been present. All signal processing operations were performed in Matlab 6.5 (MathWorks, Inc.) and each task is described henceforth in terms of its Matlab implementation.

The high frequency noise component of the signal was significantly reduced by applying a low-pass digital filter to the kinematic data. A second-order Butterworth filter was chosen because of its shorter rise time and in spite of its overshoot in response to impulse type movements (which was unlikely to pose a problem to the data in question). The filter itself introduces a phase delay to the higher harmonic range of input frequencies passed and this effect was eliminated by applying the filter a second time to the attenuated output signal which was reversed in time. The final output sequence then displayed zero-phase shift due to the canceling of the phase lead and lag. The implementation of a bi-directional filter also had the effect of doubling the sharpness of cutoff, effectively creating a fourth-order filter.

The optimal cutoff frequency was determined by performing a residual analysis on the difference between the filtered and unfiltered signals. The overall cutoff frequency was set at 10Hz because the greatest percentage of signal power was contained below that frequency. This selection was a compromise to minimize noise while simultaneously minimizing distortion. The cutoff frequency for the original second-order filter needed to be adjusted to account for the increased sharpness in cutoff caused by double filtering. The overall cutoff frequency was then equal to 0.802 times the cutoff frequency of the one-way filter (Winter). The Matlab function FILTERDATA can be referenced in Appendix A.



Source: David A. Winter , Biomechanics and Motor Control of Human Movement, 2nd ed. (New York: Wiley-Interscience,1990) 41.

Figure 2.9 Response of bi-directional filter fourth order versus uni-directional second order filter.

2.7 Model Estimation

It was necessary to estimate a model for each class of the training data from the (176x18) and (260x18) feature vectors of each subject. A Gaussian model for the actual probability distribution was assumed because of the nature of the data, which were likely corrupted by normal random processes (which have normal distributions). To verify that this assumption was acceptable, pairwise scatter plots for random samples were constructed. Pairs of variables have a bivariate normal distribution when the sample comes from a multivariate normal distribution, so the scatter plot allowed the association between variables to be examined.

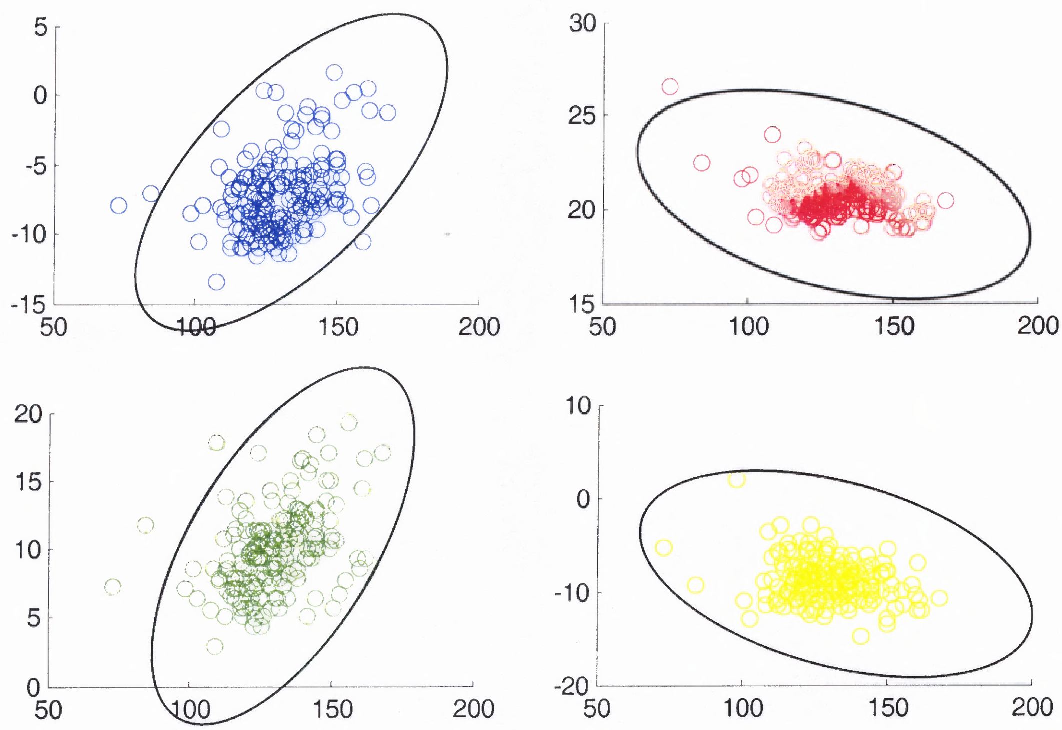


Figure 2.10 Pairwise scatter plots show bivariate normality.

Visual inspection can confirm that elliptical patterns are formed by these pairs of variables, thereby validating the assumption of normality.

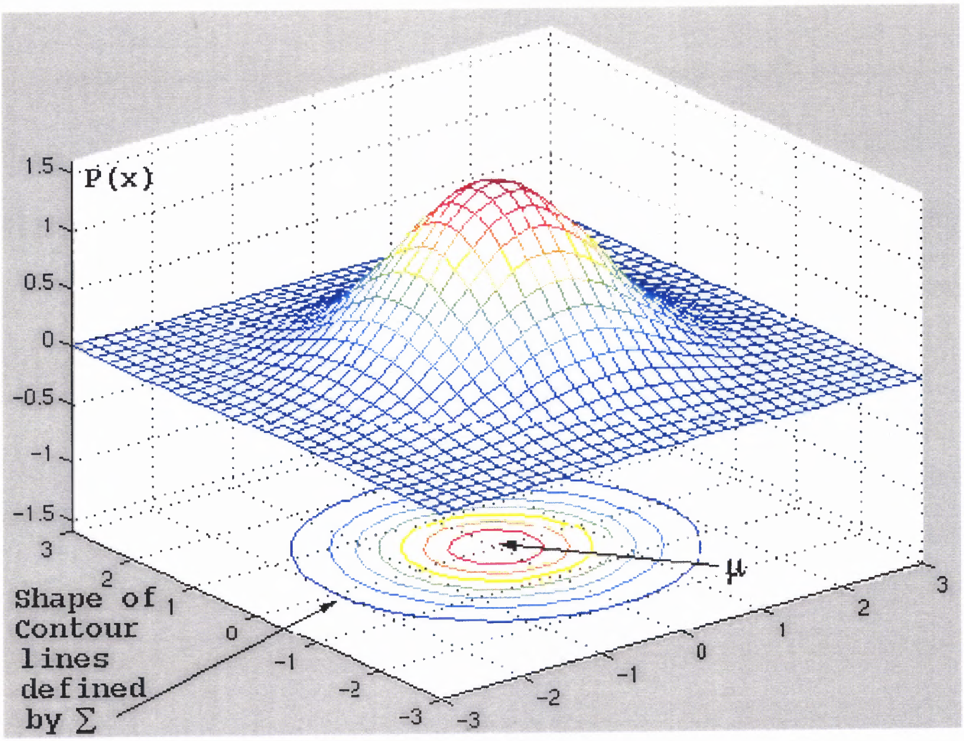
An 18-variate normal distribution for each sample, $X \sim N_{18}(\mu_i, \Sigma_i)$, was considered to provide a satisfactory approximation of the true density, where X is an 18-dimensional row vector, determined by the elements of the mean vector, μ_i (also an 18-dimensional row vector), and the covariance matrix, Σ_i, (an 18x18 positive definite symmetric matrix i.e. none of the eigenvalues are zero). The multivariate normal distribution of its probability density function is of the form,

$$p(x) = \frac{exp(-1/2*(x - \mu_i)'\Sigma_i^{-1}(x - \mu_i))}{\sqrt{(2\pi^d}\,|\Sigma_i|)}$$

where $|\Sigma_i|$ is the determinant of the covariance matrix, and d is the number of dimensions. Because Σ_i is a positive definite matrix, Σ_i^{-1} is positive definite by implication, and therefore,

$$(x - \mu_i)'\Sigma_i^{-1}(x - \mu_i) \geq 0$$

The maximum value of the density occurs at $x - \mu_i = 0$, or $x = \mu_i$ (because of the (-1/2) in the front of the exponent). Multivariate normal data form hyperellipsoids, as defined by this quadratic equation, whose points cluster about the mean, μ, and whose shape is dictated by the eigenvalues (which determine the length), and the eigenvectors (which determine the principal axes) of the covariance matrix, Σ. Figure 2.11 shows a bivariate normal distribution. In this instance the contours are spherical because the covariance matrix is diagonal (covariances (non-diagonal elements) are zero because features are independent) and the variances are equal.



Source: http://www.cs.mcgill.ca/~mcleish/644/normal_header.html

Figure 2.11 Bivariate normal distribution.

The contours of constant density are determined by setting this negative exponent of the Gaussian density to a constant value, C, which represents the distance from the mean group value that has a constant covariance, Σ_i.

$$(x - \mu_i)^t \Sigma_i^{-1} (x - \mu_i) = C$$

This is called the Mahalanobis squared distance or D statistic, after Prasanta Chandra Mahalanobis of India, whose statistical analyses of anthropometric problems led to its discovery. The larger this distance, the smaller the probability density, since the density decreases exponentially with the square of the distance. In two-dimensional space this distance forms an ellipsoid as shown below. Here, the ellipses show the 50%, 95%, and 99% lines of equal probability density, indicating the percentage probability that an unknown sample will fall in that area (Duda et al.).

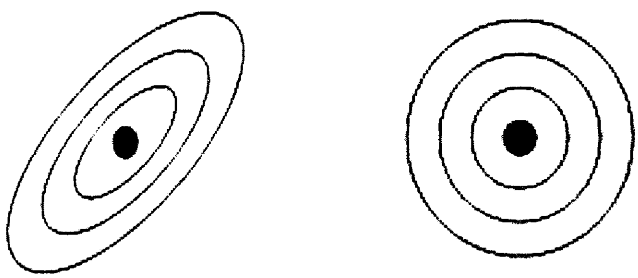
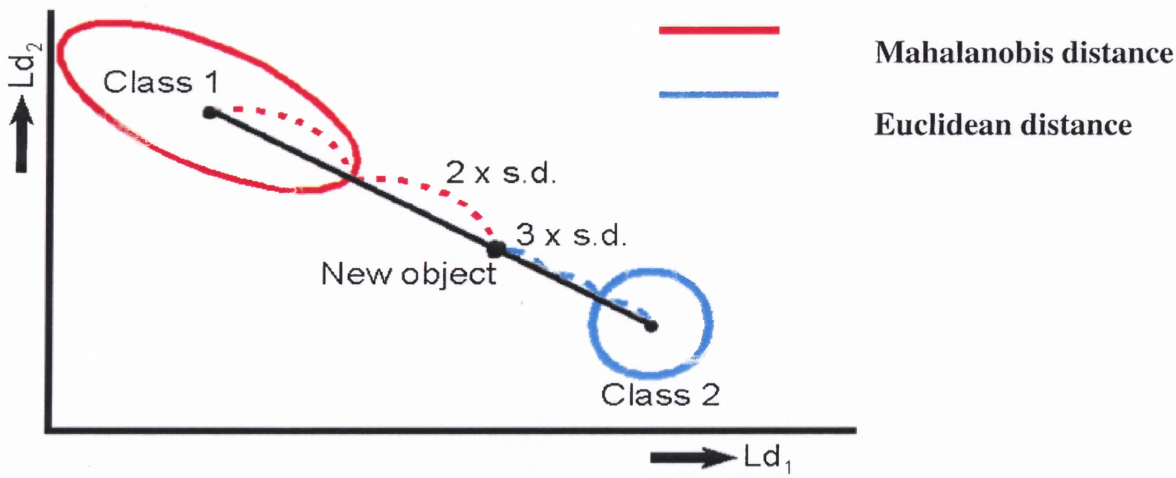


Figure 2.12 Contours of constant Mahalanobis distance (left) and Euclidean distance (right) in two-dimensional space.

Mahalanobis squared distance is a natural choice to measure the distance from an unclassified joint angle vector, x, to a mean vector, μ_i, so classification in this 26-category case was assigned to the group with the shortest Mahalanobis distance between x and μ_i of the training data, while weighing the variation in the unknown sample by the range of variability in that direction. Differences in the sample are weighted the most on the shorter axis and the least when they lie along the axis of elongation. This is preferable to Euclidean distance which measures relative distance by weighing all directions equally, thus failing to take into account the variability of the values in all dimensions. Figure 2.13 illustrates how the distance between the new object and the mean of Class 1 is 2 standard deviations (constant density in the univariate case, equivalent to covariance in the multivariate case) whereas the distance between the new object and Class 2 is 3 standard deviations. The new object is then classified to Class 1 which has the shortest statistical distance.



Source: http://www.cac.sci.kun.nl/people/philipg/poster97/

Figure 2.13 Classification using Mahalanobis and Euclidean distance measures.

Mahalanobis distance looks at variations in within-group (variance) and between-group (covariance) responses alike. The training groups define multidimensional spaces within whose borders an acceptable range of variation lies and into which an unknown vector must fall in order for it to be classified as a member. The separation between classes in LDA space is achieved by minimizing the variance within groups and maximizing the variance between groups.

The LDAM algorithm implements a linear discriminant using the Mahalanobis distance as a classifier. It begins by splitting the (176x18) or (260x18) feature matrix of training data from the static block into the 26 respective classes and calculating 18-component mean vectors for each. Class dependent covariance matrices, weighted by the number of observations in each class of the training set, are then pooled and inverted. The means and inverse pooled covariance matrix create clusters in 18-dimensional space, where the Mahalanobis distances between classes are then computed. Tables 2.3 and 2.4 indicate the normalized Mahalanobis distances between the training and testing sets, where the testing set, for the purpose of illustration, consists of the mean vectors from each class.

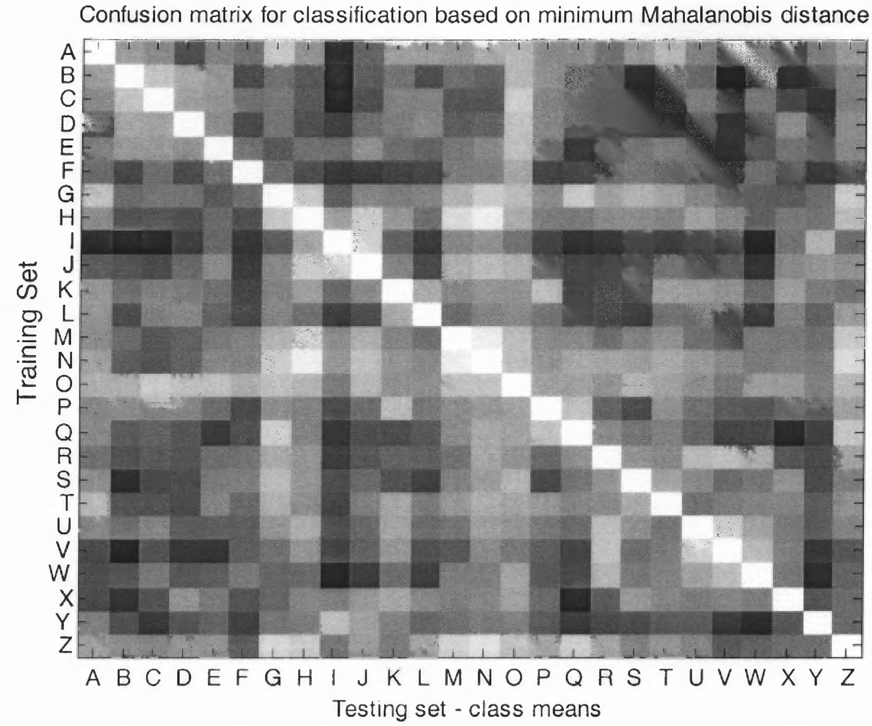
Table 2.3 Mahalanobis Distances Between Training Set and Class Means for Subject 1

	A	B	C	D	E	F	G	H	I	J	K...
A	0	319.77	370.17	541.95	395.73	451.15	194.83	336.97	680.35	486.02	312.47
B	319.77	0	216.18	283.12	323.68	557.82	450.96	518.37	783.11	501.73	429.11
C	370.17	216.18	0	295.56	333.5	434.01	413.77	506.45	696.03	531.58	388.65
D	541.95	283.12	295.56	0	356.31	546.49	422.27	526.54	577.25	441.06	399.63
E	395.73	323.68	333.5	356.31	0	429.12	507.12	445.65	464.1	348.42	451.7
F	451.15	557.82	434.01	546.49	429.12	0	441.37	526.96	621.7	588.26	573.96
G	194.83	450.96	413.77	422.27	507.12	441.37	0	172.18	497.55	365.19	383.59
H	336.97	518.37	506.45	526.54	445.65	526.96	172.18	0	285.77	158.39	295.82
I	680.35	783.11	696.03	577.25	464.1	621.7	497.55	285.77	0	146.43	476.34
J	486.02	501.73	531.58	441.06	348.42	588.26	365.19	158.39	146.43	0	370.27
K	312.47	429.11	388.65	399.63	451.7	573.96	383.59	295.82	476.34	370.27	0
L	324.06	543.82	381.47	362.17	482	559.96	372.81	439.19	588.88	545.21	263.48
M	338.25	443.36	504.57	477.71	415.32	380.57	238.44	127.7	354.17	249.26	368.5
N	272.95	485.58	502.37	474.33	369.25	380.33	161.24	60.24	322.82	218.9	322.42
O	253.24	239.71	160.42	203.63	213.45	258.89	208.43	248.09	405.26	242.83	331.65
P	359.53	335.31	378.33	364.61	446.8	552.52	335.96	387.24	556.66	447.41	207.42
Q	344.23	490.98	466.97	504.01	599.9	530.9	153.91	379.66	644.27	547.03	552.24
R	426.81	537.74	498.21	521.03	436.67	410.47	264.77	321.75	601.79	523.73	485.5
S	314.85	653.51	548.47	533.82	400.8	382.07	210.57	310.53	526.55	435	505.24
T	213.13	502.84	513.51	546.97	376.02	461.01	256.53	336.5	558.28	382.78	262.19
U	358.41	434.7	365.85	499.53	495.33	501.11	319.04	246.97	525.22	443.43	274.23
V	561.33	713.1	518.71	617.46	630.38	485.8	422.08	317.27	557.18	464.03	305.53
W	537.38	513.03	391.18	495.03	488.95	398.08	457.88	444.77	690.81	591.36	428.14
X	543.03	641.81	525.91	315.83	395.37	516.78	456.57	357.26	384.64	313.11	260.05
Y	367.71	519.58	601.07	507.15	474.89	566.89	398.27	381.69	206.58	289.98	378.15
Z	283.67	349.93	365.48	329	331.63	490.98	138.87	181.38	407.08	303.3	392.39

Table 2.4 Mahalanobis Distances Between Training Set and Class Means for Subject 2

	A	B	C	D	E	F	G	H	I	J	K...
A	0	702.23	605.82	661.32	527.51	705.94	276.91	420.67	535.62	556.84	378.84
B	702.23	0	289.59	562.87	667.68	875.98	675.72	580.08	543.35	514.33	571.58
C	605.82	289.59	0	389.61	429.59	667.22	525.16	429.09	311.86	368.11	477.53
D	661.32	562.87	389.61	0	451.62	691.13	347.41	374.24	335.6	339.98	172.43
E	527.51	667.68	429.59	451.62	0	637.65	601.53	501.37	372.21	338.79	478.3
F	705.94	875.98	667.22	691.13	637.65	0	644.98	701.2	619.65	673.62	667.99
G	276.91	675.72	525.16	347.41	601.53	644.98	0	260.22	444.91	426.42	339.35
H	420.67	580.08	429.09	374.24	501.37	701.2	260.22	0	376.23	397.83	266.52
I	535.62	543.35	311.86	335.6	372.21	619.65	444.91	376.23	0	89.82	372.88
J	556.84	514.33	368.11	339.98	338.79	673.62	426.42	397.83	89.82	0	394.14
K	378.84	571.58	477.53	172.43	478.3	667.99	339.35	266.52	372.88	394.14	0
L	511.87	669.4	448.53	214.3	587.94	714.87	319.38	573.46	565.88	525.47	295.21
M	489.21	512.4	517.51	350.54	527.74	540.86	269.46	246.82	366.39	396.91	279.19
N	415.24	753.67	540.17	447.74	568.47	533.72	251.9	107.2	456.72	489.91	306.68
O	445.73	669.69	343.66	299.94	331.98	481.64	317.56	294.97	266.14	330.96	343.38
P	418.36	548.73	417.61	236.26	477.6	723.97	424	295.25	375.04	451.46	75.859
Q	396.51	742.95	519.88	457.81	608.88	746.13	229.58	372.33	439.04	589.13	440.3
R	493.05	867.5	580.34	530.45	489.87	585.05	320.63	231.38	544.47	600.3	418.11
S	341.33	914.3	728.8	606.52	298.5	542.17	387.61	512.6	461.36	448.29	515.57
T	275.73	776.83	545.05	532.67	402.22	606.87	441.83	413.35	416.58	455.92	273.65
U	496.74	622.62	317.19	345	365.26	635.53	350.53	141.68	387.26	460.29	250.81
V	799.98	1075.7	697.06	563.84	587.9	654.7	586.69	482.44	615.45	736.54	422.85
W	711.39	639.81	643.37	560.05	518.43	536.47	563.29	650.27	802.13	816.64	513.85
X	611.39	675.05	468.75	188.67	393.17	739.45	584.37	541.02	423.6	394.22	247.95
Y	357.74	675.9	534.24	475.54	697.97	680.51	449.7	446.79	300.86	371.88	371.62
Z	425.34	462.2	353.55	236.7	458.54	752.92	272.44	253.15	318.16	339.71	243.8

The confusion matrices in Figure 2.14 are grayscale colormaps of the tables of the training versus testing sets, where white indicates minimum Mahalanobis distance and black indicates maximum distance. They provide a summary of the extent to which the Mahalanobis distances could predict the class to which a handshape would be classified. The white blocks along the main diagonal illustrate how each mean vector along the x-axis has the minimum Mahalanobis distance to the corresponding class of the y-axis, and so would be classified accordingly.



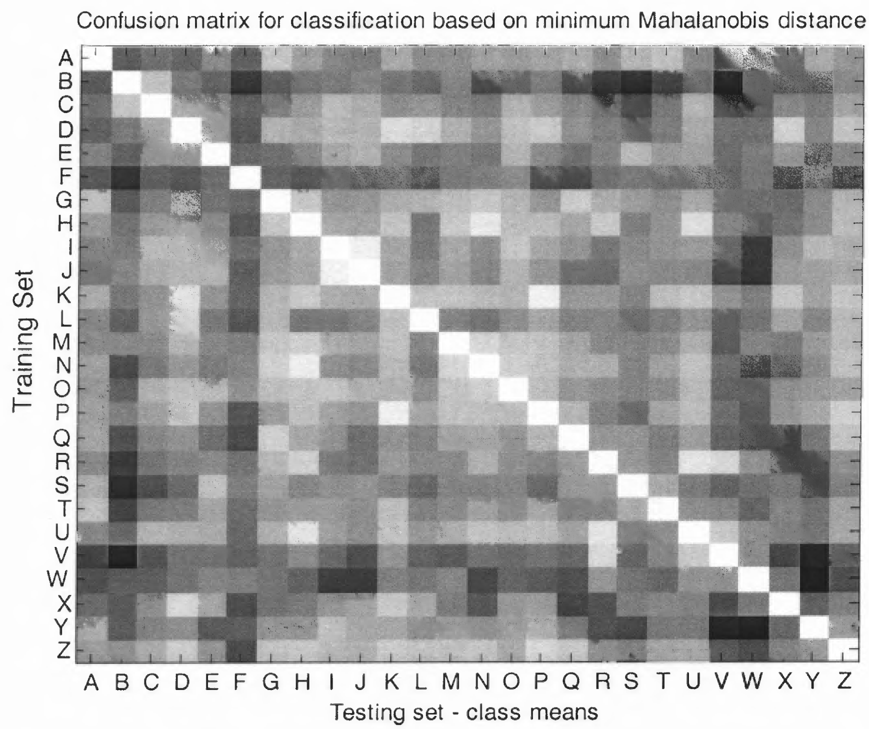


Figure 2.14 Confusion matrix for training set versus testing set of class means for Subject 1 (above) and Subject 2 (below).

2.8 Classification

The CLASSIFYSTATIC algorithm, implemented in Matlab, was used to assign consecutive feature vectors performed in a string to the class with the minimum Mahalanobis distance. In each case, the classifier was trained with the static block of the performing subject. The program first prompted the user for subject and string selection. Once acquired, training and testing data were pre-processed by passing forward and backward through a fourth-order Butterworth digital low-pass filter, and features corresponding to the 18 joint sensors from the *Cyber*Glove® were extracted. Input to the linear discriminant analyzer, such as number of classes and class size, was compiled according to the subject in question and classification of the entire string was then performed. Finally, feature vectors, corresponding to the time points at which the velocities of summed joint angles fell to a minimum, were classified.

2.8.1 Bayesian Decision Theory

Bayesian Decision Theory is a methodology that uses a priori information about the data to solve pattern recognition problems by applying decision rules that use probability distributions. Bayes Formula states that the probability (posterior, $P(\omega_j|x)$) that a pattern is in class ω_j and contains the feature vector, x, can be calculated from prior information (right-hand side of equation),

$$P(\omega_j|x) = \frac{p(x|\omega_j)\,P(\omega_j)}{p(x)}$$

where $p(x|\omega_j)$ is the likelihood that ω_j is the true class for a large value of $p(x|\omega_j)$, $P(\omega_j)$ is the prior probability, and $p(x)$ is the evidence or scale factor that ensures that the posterior probabilities sum to one (Duda et al.).

Let ω_1 represent the state of nature where a hand configuration produces the letter 'A'. Let x represent the 18-component feature vector of joint angles that are produced in the formation of the letter. The expression $p(x|\omega_1)$ represents the class-conditional probability density function for the feature vector, x, given that the state of nature is an 'A'. Finding the posterior probability means finding the probability of a hand configuration pattern belonging to a particular letter class, given that a certain vector of joint angles is observed. If such a probability were known, then for some observed feature vector, the hand configuration could be classified by comparing the probabilities that each letter class had such a feature vector ($P(\omega_j|x)$). According to the Bayes Decision Rule, the feature vector should then be classified to the class with the highest posterior probability to minimize the probability of error. Because $p(x)$ is irrelevant to the decision-making process, in that it simply ensures that the sum of prior probabilities is equal to one, this rule becomes based on the combination of the states of nature (prior probabilities, $P(\omega)$) and the likelihoods (conditional probabilities, $p(x|\omega)$ so that an unknown sample is classified to ω_i if $p(x|\omega_i)P(\omega_i) > p(x|\omega_j)P(\omega_j)$, otherwise the sample is classified to ω_j.

Hand configuration patterns in this study were classified by computing different discriminant functions, $g_i(x)$, for each ω_i, such that $\{\omega_1,\ldots\omega_{26}\}$ is the finite set of 26 states of nature of classes, and selecting the class corresponding to the largest discriminant.

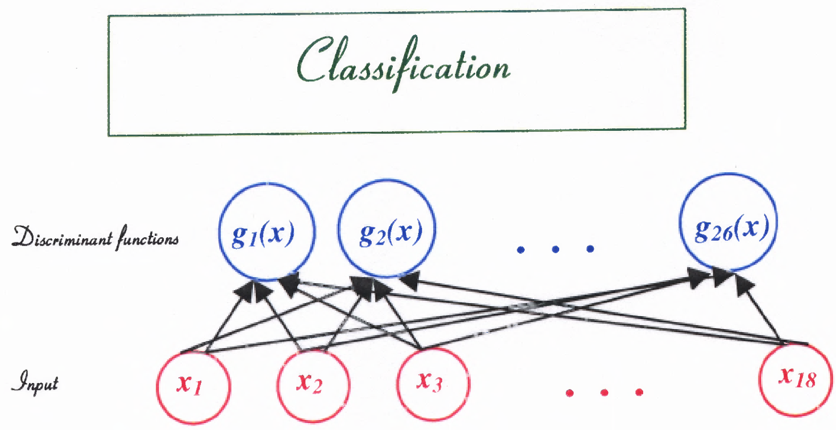


Figure 2.15 Functional structure of pattern classifier.

The Bayes Decision Rule was used as a discriminant function, such that,

$$g_i(x) = P(\omega_i|x) = p(x|\omega_j)P(\omega_j)$$

$$= \ln p(x|\omega_j) + \ln P(\omega_j)$$

The maximum discriminant function corresponds to the maximum posterior probability and the classifier assigned the hand configuration to the class with the maximum posterior probability. The multivariate normal density, assumed for the movement data analyzed here, had multivariate normally distributed conditional densities $p(x|\omega_i) \sim N(\mu_i, \Sigma)$ and prior probabilities $P(\omega_i)$. Since,

$$p(x|\omega) = \frac{\exp(-1/2*(x - \mu_i)^t \Sigma^{-1}(x - \mu_i))}{\sqrt{(2\pi^n |\Sigma|)}}$$

the discriminant function became,

$$g_i(x) = \ln\left(\frac{\exp(-1/2*(x - \mu_i)^t \Sigma_i^{-1}(x - \mu_i))}{\sqrt{(2\pi^d\,|\Sigma_i|)}}\right) + \ln P(\omega_j)$$

$$= -\frac{1}{2}(x - \mu_i)^t\, \Sigma^{-1}(x - \mu_i) - \frac{d}{2}\ln 2\pi - \frac{1}{2}\ln|\Sigma| + \ln P(\omega_j)$$

where Σ was the pooled covariance matrix and μ_i was the mean of class ω_i, i = 1,...26. Since the prior probabilities were assumed to be the same for all classes, the last three terms were eliminated as extra constants resulting in,

$$g_i(x) = -\frac{1}{2}(x - \mu_i)^t\, \Sigma^{-1}(x - \mu_i)$$

This quadratic discriminant function is, of course, the squared Mahalanobis distance, and the optimal decision rule for classification is that the feature vector, x, should be classified to the category which has this distance at a minimum. This is based on Bayes Decision Rule of classifying according to the maximum posterior probability (maximum discriminant function) which is proportional to the probability density function, of which the Mahalanobis distance is the negative exponent. The smaller this distance, the larger the probability density, and the larger this distance, the smaller the probability density, since the density decreases exponentially with the square of the distance. Consequently, classification to the class with minimum Mahalanobis distance means classifying to the class with the maximum probability density, which in turn means classifying to the class with the maximum posterior probability.

Geometrically, the effect of the decision rule is to divide the feature space into 26 decision regions which are defined by hyper ellipsoidal clusters, centered about different mean vectors, and separated by decision boundaries. These boundary surfaces, in the case of 18-dimensional feature vectors, take the form of 17-dimensional hyperplanes, and are defined by the linear equations $g_i(x) = g_j(x)$ for the two classes with the highest posterior probabilities. Expansion of the quadratic form of the discriminant function results in an equivalent sum of linear discriminant functions,

$$g_i(x) = w_i^t x + w_{i0}$$

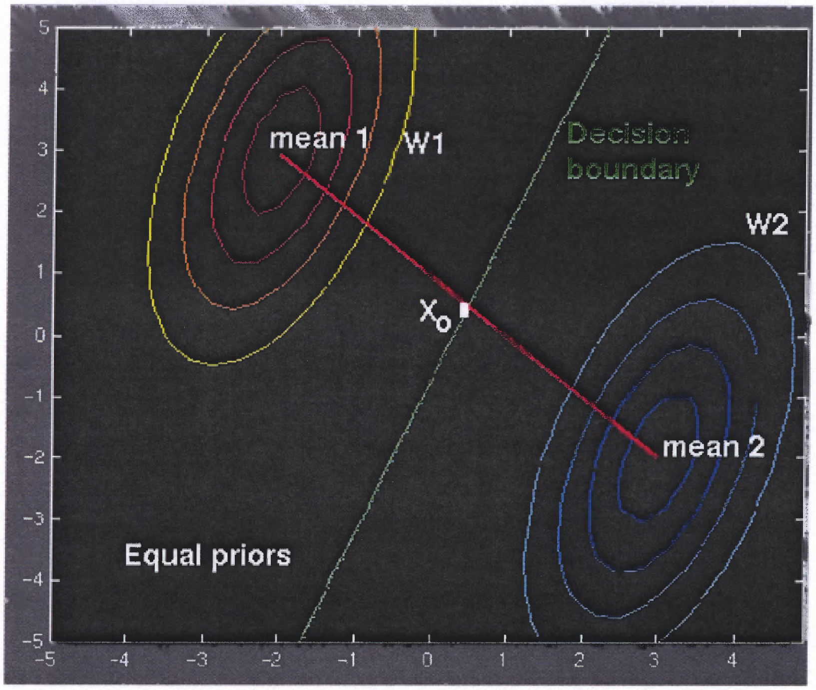
where w_{i0} is the weight factor,

$$w_{i0} = \frac{-1}{2}\mu_i^t \Sigma^{-1} \mu_i + \ln P(w_i)$$

and w_i is the threshold or bias,

$$w_i = \Sigma^{-1} \mu_i$$

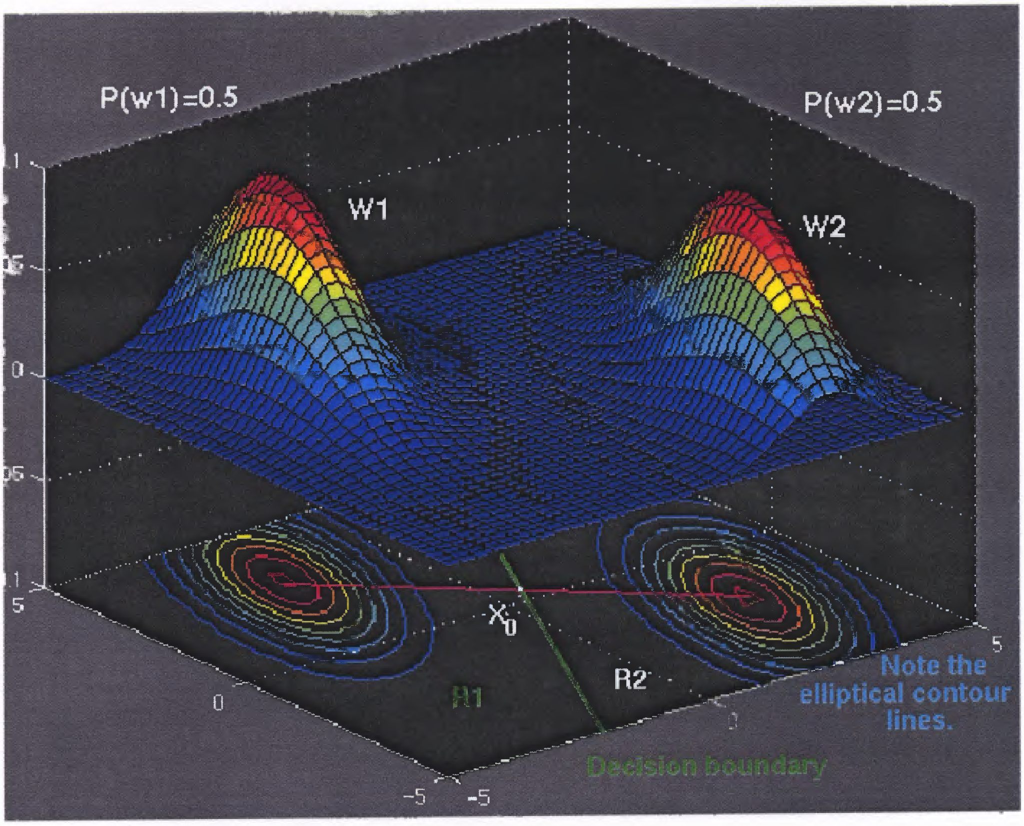
The orientations of the surfaces of these decision boundaries are determined by the weight vector, w_i, so that its points are of equal distance to the contour lines surrounding the means in each class, and the locations are determined by the bias, w_{i0}. Unequal prior probabilities also bias the decision in favor of the a priori more likely class.



Source: http://www.cs.mcgill.ca/~mcleish/644/normal_header.html

Figure 2.16 Decision boundary for bivariate Gaussian distributions.

This is illustrated in Figures 2.16 and 2.17 for bivariate normal distributions. Here, the covariance matrix is not diagonal and the variances are not equal causing the rotation and the elliptical shape of the contours. Notice how the prior probabilities are the same, causing the decision boundary, which runs through the point x_0, to fall half-way between the two means. Because the probability densities are bivariate, the decision boundary is a 1-dimensional plane, or line. Here, the decision line is tilted as per the weight factor, so that it is equidistant to the Mahalanobis space constructed by W1 and W2. Observations are classified according to the decision region into which they fall.



Source: http://www.cs.mcgill.ca/~mcleish/644/normal_header.html

Figure 2.17 Probability densities and decision regions for bivariate Gaussian distributions.

Using these linear discriminant functions for pattern classification, the set of feature vectors, corresponding to the angular displacement of joints caused by the execution of the string "PERISCOPE" by Subject 4, was classified at every time point in the recorded trajectory and this is illustrated by the confusion matrix in Figure 2.18. The upper graph of the figure is a plot of the summed absolute velocity of joint angles. The Mahalanobis distances between the vectors produced at every time frame in the string and the mean vectors for each class are represented visually in grayscale, where black corresponds to the maximum distance and white corresponds to the minimum distance.

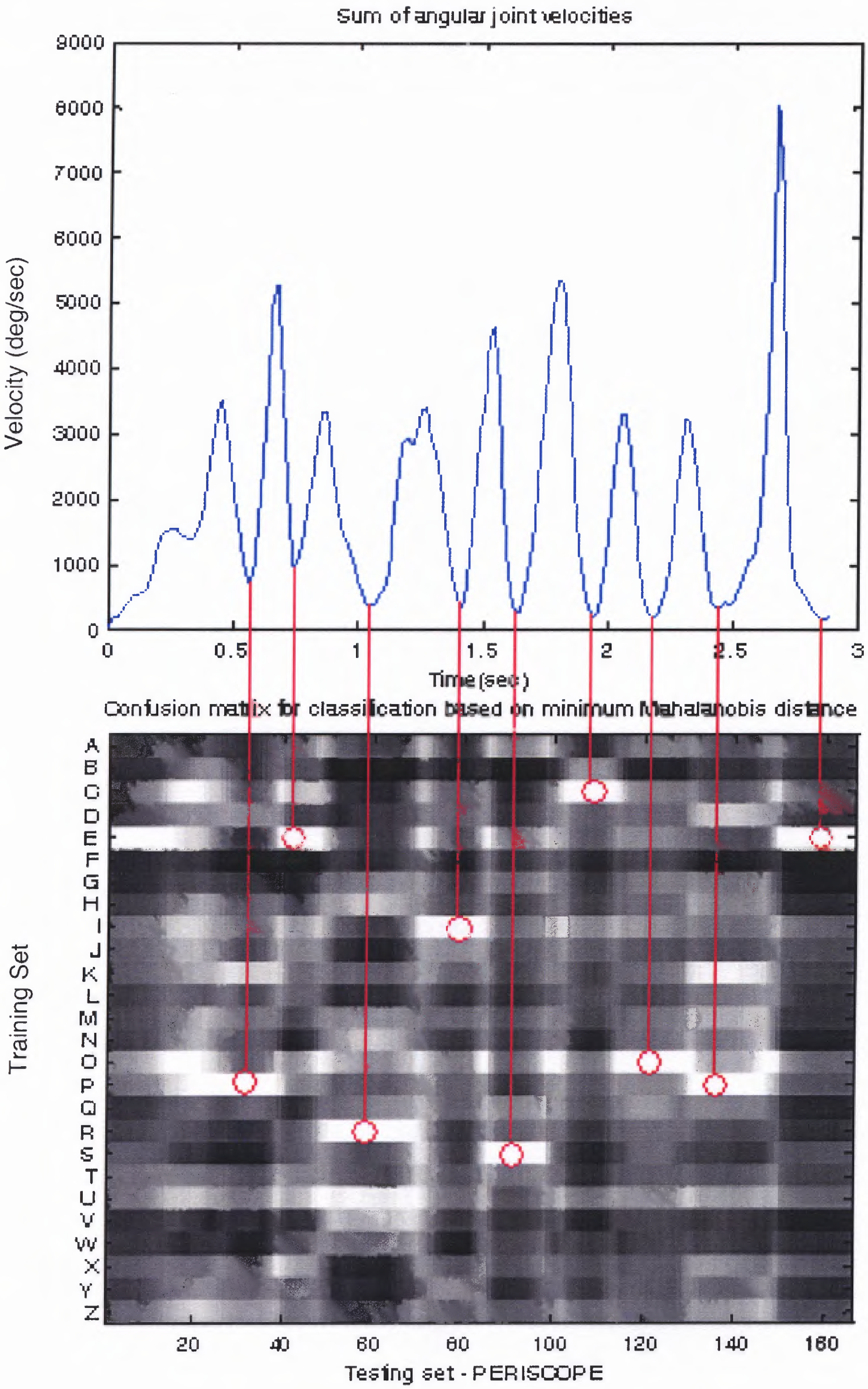


Figure 2.18 Classification of entire string using linear discriminant classifier.

The minimum distances indicate the categories to which a linear discriminant classifier would assign an unknown vector. Comparison of the times at which such distances occur (as circled in red) with the plot of summed angular joint velocity, strongly suggests that the classification of target letters in the string occurs at points of minimum velocity. Notice the distinct switching between black and white indicating the transition between maximum and minimum velocities.

2.8.2 Calculation of Angular Joint Velocities

In order to detect these points of minimum velocity, the angular displacement of joints produced within the recorded trajectory was first differentiated and smoothed to obtain the instantaneous velocity of the individual joints. Angular joint velocities were derived from the angular displacement data by numerical differentiation using a central difference algorithm. This three-point method uses a Taylor series of points to find derivatives.

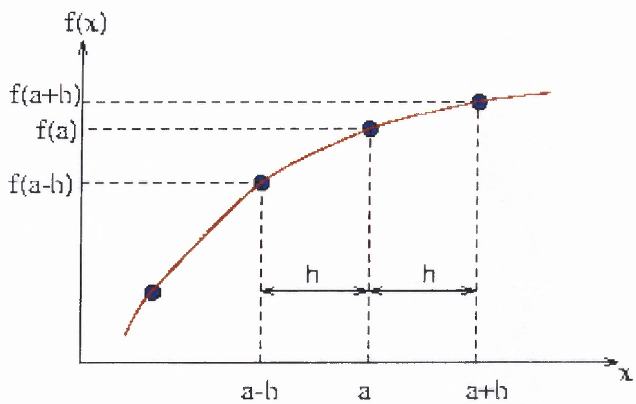


Figure 2.19 Graphical representation of function f(x) to be differentiated at x = a.

The centered formula for the first derivative is based on three points centered about the point $x=a$, as represented by the Taylor series expansion, given as follows,

$$f(x) = f(a) + \Delta x \frac{df}{dx}\Big|_{x=a} + \frac{(\Delta x)^2}{2!}\frac{d^2 f}{dx^2}\Big|_{x=a} + \frac{(\Delta x)^3}{3!}\frac{d^3 f}{dx^3}\Big|_{x=a} + K$$

$$\Delta x = h = x - a$$

$$f(x) = f(a) + f'(a)(x\text{-}a) + f''(a)\frac{(x\text{-}a)^2}{2!} + f'''(a)\frac{(x\text{-}a)^3}{3!} + K$$

Assuming the data are equally spaced, the points can be represented by the following equations,

$$f(a+h) = f(a) + f'(a)h + f''(a)\frac{h^2}{2!} + f'''(a)\frac{h^3}{3!} + K \qquad (2.1)$$

$$f(a) = f(a) \qquad (2.2)$$

$$f(a\text{-}h) = f(a) - f'(a)h + f''(a)\frac{h^2}{2!} - f'''(a)\frac{h^3}{3!} + - K \qquad (2.3)$$

For central differentiation, equation 2.3 is subtracted from equation 2.1 to give,

$$f'(a) = \frac{f(a+h) - f(a\text{-}h)}{2h} + Error$$

This method has greater accuracy and lower error than the forward or backwards difference methods. The three-point methods do, however, introduce end effects at the beginning and end of the analyzed waveform, so a different three point method was required to evaluate the value of $f'(a)$ at the endpoints. By using more terms to model the first derivative, these effects were eliminated and the equation became,

$$f'(a) = \frac{-3f(a) + 4f(a+h) - f(a+2h)}{2h} + Error$$

for the first point and,

$$f'(a) = \frac{-3f(a) + 4f(a-h) - f(a-2h)}{2h} + Error$$

for the endpoint. Once the angular velocities were calculated for each of the 18 measured joints throughout the trajectory, the absolute values of these velocities were summed together and passed to the next function for minima detection. Code for the velocity calculations was executed within the Matlab function ABSSUMVEL, which can be referenced in Appendix A.

Local minima of the summed angular joint velocities were calculated with an off-the-shelf Matlab function, LMIN, called by LOCAL_MINIMA. Through trial and error, a default filter level was selected which determined the number of passes of the running average filter required to eliminate small peaks. Once these were removed, the function calculated the values of the local minima and more importantly, the time points at which they occurred.

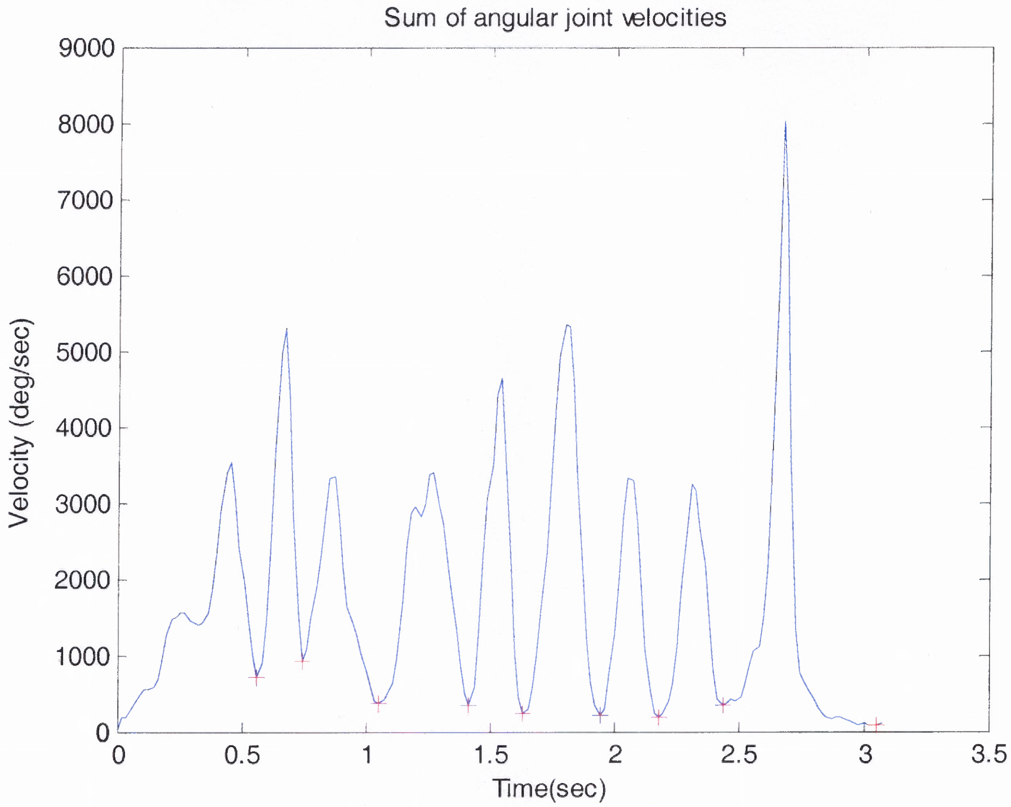


Figure 2.20 Summed angular joint velocities and local minima for string "PERISCOPE".

Figure 2.19 shows the same plot included in Section 2.8.1 of summed angular joint velocity produced by Subject 2 when executing the hand configurations for the string "PERISCOPE". The plot now also includes the points marked as local minima, as determined by the LOCAL_MINIMA function. Instead of feeding the entire sequence of consecutive hand configurations recorded every 18ms to the LDA for classification, only the feature vectors performed at the time points of minimum velocity were classified. Figure 2.20 shows these nine minima to have minimum Mahalanobis distances from the classes corresponding to the letters P, E, R, I, S, C, O, P and E, respectively.

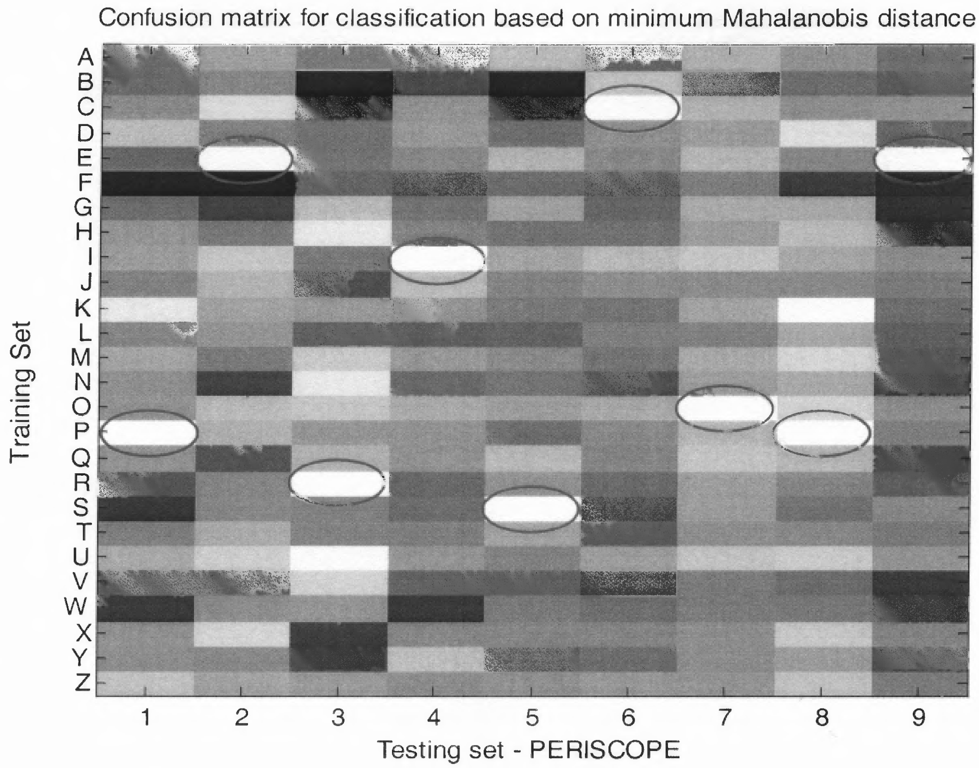


Figure 2.21 Confusion matrix of Mahalanobis distances of local minima for string "PERISCOPE".

Notice how P and K at the first and eighth minima have the shortest distances, indicating a similarity in configuration of joint angles. This is verified by Table 2.5 from which the confusion matrix is derived.

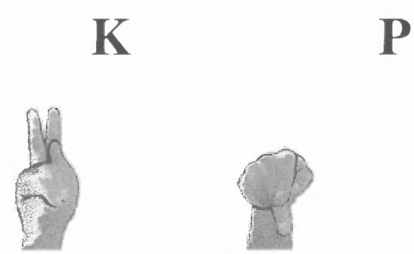


Figure 2.22 Similar joint angle configuration for K (left) and P (right), differing only in orientation.

Table 2.5 Mahalanobis Distances Between Training Set and Local Minima for String 'PERISCOPE' Performed by Subject 2

	P	E	R	I	S	C	O	P	E
A	549.37	342.26	483.83	553.65	261.92	571.53	262.71	354.47	485.44
B	550.47	26.13	933.08	660	844.89	212.57	632.57	506.84	582.83
C	411.28	137.99	598.02	369.75	618.03	0	252.71	363.41	347.63
D	243.59	458.52	499.89	357.85	520.63	389.42	325	102.12	519.38
E	563.24	0	500.64	431.02	275.04	410.37	235.58	396.35	0
F	879.34	866.34	566.77	662.87	532.8	635.46	450.05	696.84	772.97
G	505.61	676.82	245.5	494.82	320.49	530.39	266.66	268.64	743.45
H	357.25	473.1	101.24	445.45	490.53	486.49	326.52	229.07	585.08
I	337.25	216	486.08	0	319.45	241.42	201.42	288.24	353.31
J	438.82	268.89	634.52	235.83	425.63	383.39	384.07	382.66	427.48
K	91.509	323.79	359.9	380.48	469.01	485.31	316.87	20.482	484.59
L	444.72	373.52	571.03	588.28	557.28	455.67	447.18	298.16	553.85
M	310.84	478.5	216.33	351.67	409.03	481.12	297.74	149.67	574.23
N	381	680.72	110.17	543.52	502.55	630.02	325.86	280.9	665.44
O	365.3	243.16	226.03	295.34	346.36	298.72	0	159.37	359.08
P	0	265.31	368.23	373.37	515	411.92	249.94	0	421.31
Q	365.5	618.92	278.76	391.82	253.78	449.42	211.16	255.27	639.87
R	612.94	429.97	0	549.37	364.93	551.48	332.08	414.6	556.29
S	717.02	420.23	430.53	518.38	0	663.34	325.42	471.52	390.71
T	434.52	278.84	430.1	439.46	320.14	591.19	368.86	391.36	446.32
U	305.76	178.88	38.813	352.49	448.39	324.52	212.98	178.98	345.32
V	596.26	585.34	182.07	557.52	569.04	674.18	457.63	497.89	686.72
W	727.56	395.06	384.95	734.96	476.45	518.08	463.56	455.66	591.36
X	392.11	167.45	700.27	430.37	439.82	446.33	379.5	195.4	412
Y	400.83	432.59	627.08	245.49	596.16	526.53	380.8	330.1	607.33
Z	221.55	329.41	355.74	412.25	401.55	361.61	350.43	205.13	440.15

2.9 Data Matrix Assembly

The next step was to find the local minima for each of the ten trials of 20 strings for each subject. CLASSIFYSTATIC was called for each string replication and the cut points for the local minima of letters on either side of the trigram were recorded. (See Subject 1 Cuts and Subject 2 Cuts in Appendix B.) ASSIGNLETTERS was a Matlab script called for each string and was used to assign every trial's feature vectors for I, S and C to its own feature matrix. The static trials for each subject were first saved to the data matrices of I-Observations, S-Observations and C-Observations. These are represented by the black shaded areas in Figure 2.22.

In ASSIGNLETTERS, the user was prompted to enter the number of the subject whose data would follow. The user was then instructed to enter the string to be assigned to the data matrices, as well as the number of replications of that string that would be entered. Rejected movements included those in which the target letter was deemed to have been unsuccessfully classified. Unsuccessful classification did not necessarily imply that the letter was classified to the incorrect class. Rather, there were instances where the expected letter sequence simply did not occur, and in particular, the expected letter was completely omitted. For Subject 1, 4/200 samples were rejected (2%) for Observation C, 6/200 samples were rejected for Observation I (3%), and 2/200 samples were rejected for Observation S (1%). For Subject 2, 23/200 samples were rejected (11.5%) for Observation C, 1/200 samples were rejected for Observation I (0.5%), and 1/200 samples were rejected for Observation S (0.5%). In total, 2% and 4.16% of trials were rejected for Subjects 1 and 2 respectively.

Once all usable replications of each string were selected, the local minima of the truncated strings were classified using CLASSIFYMINIMA (a modification of the CLASSIFYSTATIC program). ASSIGNLETTERS was called 20 times for each subject and each time a context matrix was created as illustrated in Figure 2.22. From these three-dimensional matrices the data matrices for the target letters of interest were formed. These I-Observations, S-Observations and C-Observations were (201x18), (204x18) and (203x18) dimensional matrices for Subject 1 and (209x18), (209x18), (187x18) dimensional matrices for Subject 2, and included both static and contextual feature vectors to be used for analysis.

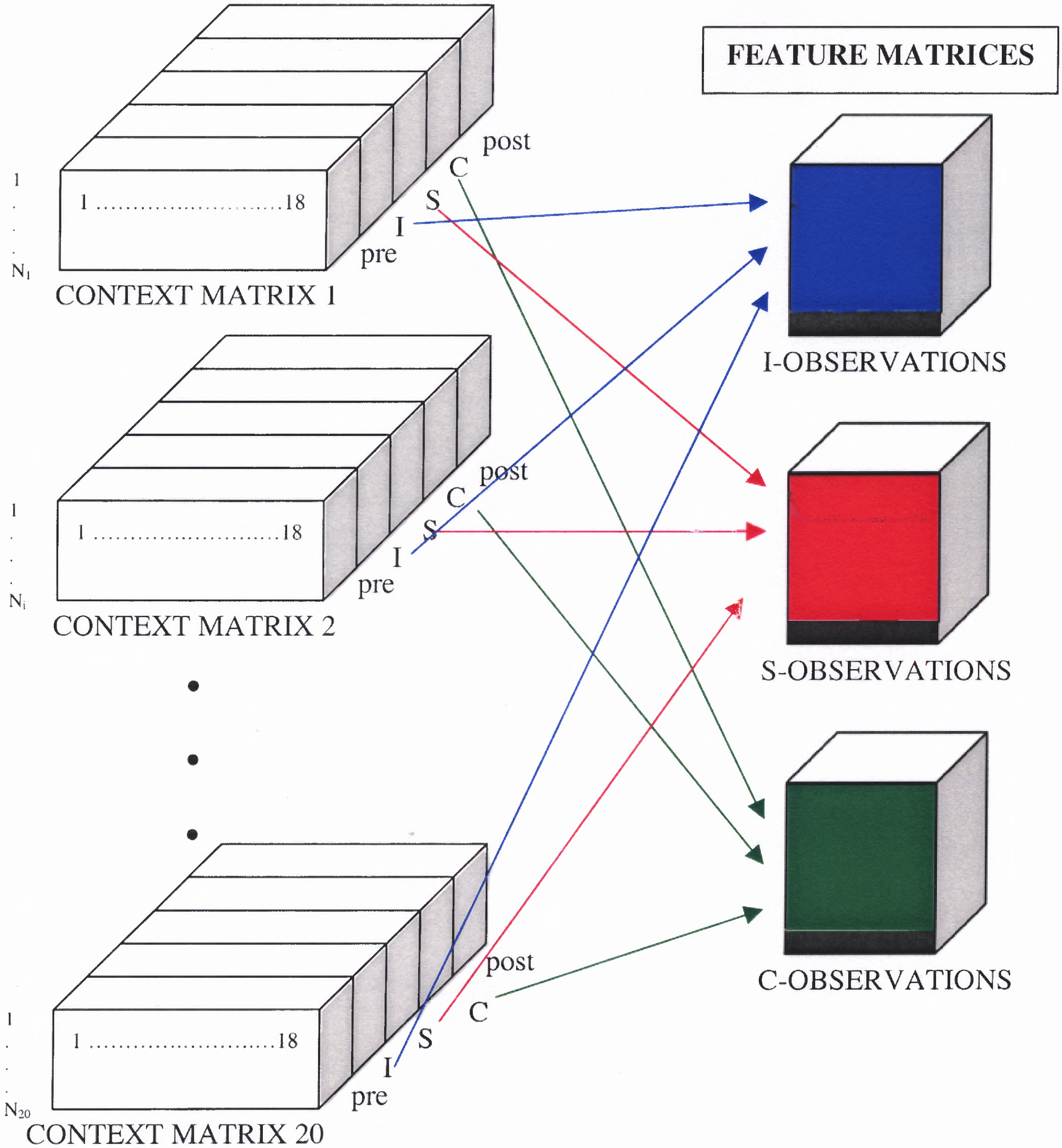


Figure 2.23 Formation of data matrices for multivariate analysis.

CHAPTER 3

RESULTS

Post-hoc multivariate analyses were performed on the three data matrices for each subject. A Multivariate Analysis of Variance (MANOVA) was the statistical test used to assess the likelihood that the mean vectors of each of the 21 groups (20 dynamic, 1 static) were taken from the same sampling distribution of means, and to explore how static versus context letters influenced the production of joint angles. The main goal of the test was to compare the mean vectors of the 18-dimensional hand configurations of the letters produced statically with those same letters produced dynamically.

3.1 Hypothesis Testing

The null hypothesis was that there is no statistical difference between populations from the static and context groups, as indicated by the sample data. Realistically, there is always a difference, due to the nature of random processes and imperfect sample data. The real question was whether the effect was trivial, as determined by predefined confidence limits. The statistical test itself simply tested for significance, trivial or not. The level of confidence in the significance was represented by the p-value associated with the multivariate statistic, and a confidence level of 0.05 was allowed in this study for a 5% chance of finding a significant difference between groups when, in fact, no difference existed. The multivariate statistic produced by the test was based on a comparison of the error variance-covariance matrix and the variance-covariance matrix of the hypothesis (effect).

3.1.1 Basic Assumptions – Multivariate Normality

In order to perform the MANOVA, it was necessary that certain basic statistical assumptions be verified. The first was that the residuals or prediction errors were sampled from a normal distribution. For this study, it was assumed that all observations were sampled from an 18-variate normal distribution with (1x18) mean vector, μ, and (18x18) covariance matrix, Σ. To verify this, an off-the-shelf program, MULNORMTEST (Trujillo-Ortiz et al.), was used. The output of this program combined a graphic approach with a probability output that the null hypothesis of normally distributed errors was true. The chi-squared quantile-quantile plot, which graphed the chi-squared percentiles against the percentiles of the I, S, and C observations' ordered squared Mahalanobis distances, are shown in Figures 3.7 to 3.14. See Appendix A.11 for Matlab source code.

Table 3.1 Test for Multivariate Normality

	I_subject1	I_subject2	S_subject1	S_subject2	C_subject1	C_subject2
n	201	209	204	209	203	187
Variables	18	18	18	18	18	18
Slope	0.5260	0.4010	0.2728	0.2634	0.4024	0.5092
t	-39.0190	-30.0498	-47.0571	-38.4782	-43.2796	-33.8333
Pr	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000

Since the independent variables consisted of groups, the normality assumption implied that the dependent variables must be normally distributed within each group. The within group dependent variables for each subject were found to follow a multivariate normal distribution, and so the Mahalanobis distances followed a chi-squared distribution with 18 degrees of freedom. The graphs below illustrate the multivariate normality of the set of observations as a whole and the outputs of the program (significance level of 0.05), as per Table 3.1, confirmed that this assumption was tenable.

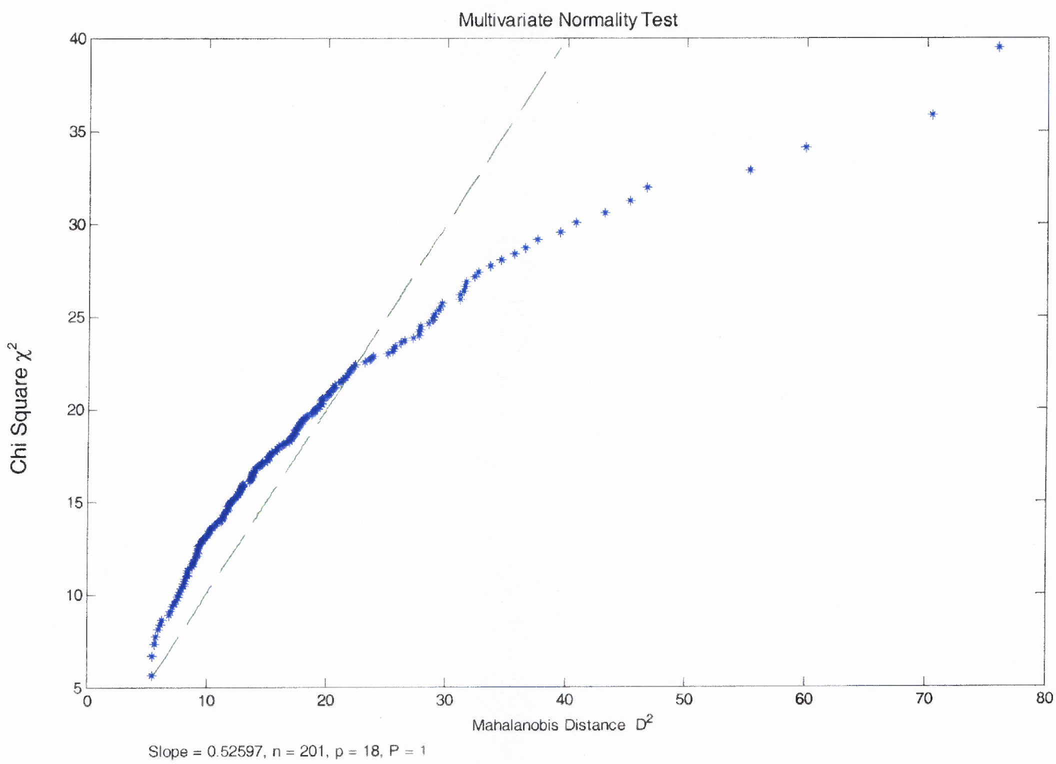


Figure 3.1 Chi-squared quantile-quantile plot of I observations for Subject 1.

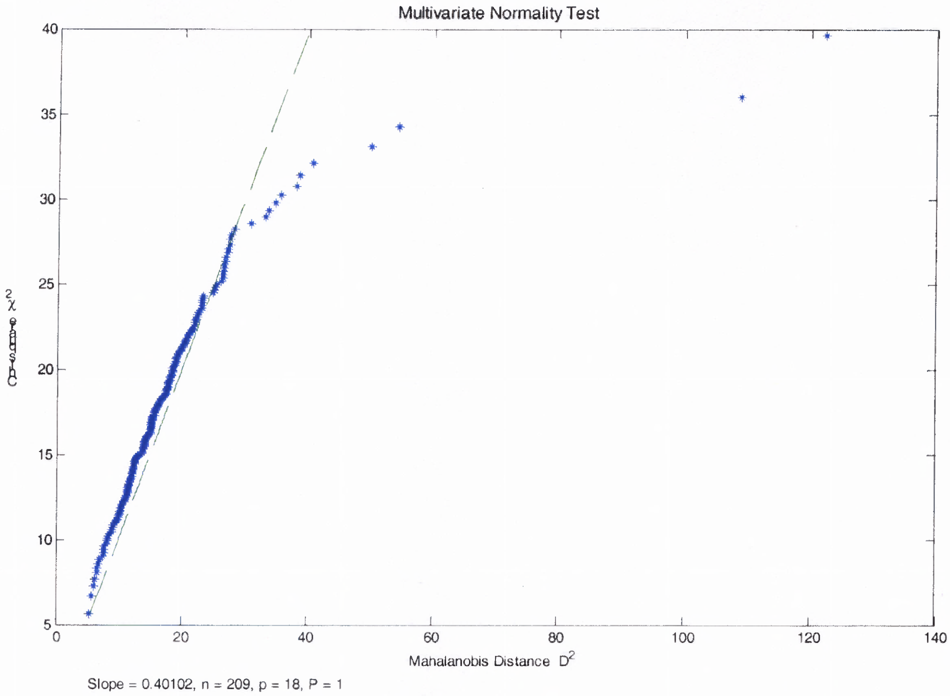


Figure 3.2 Chi-squared quantile-quantile plot of I observations for Subject 2.

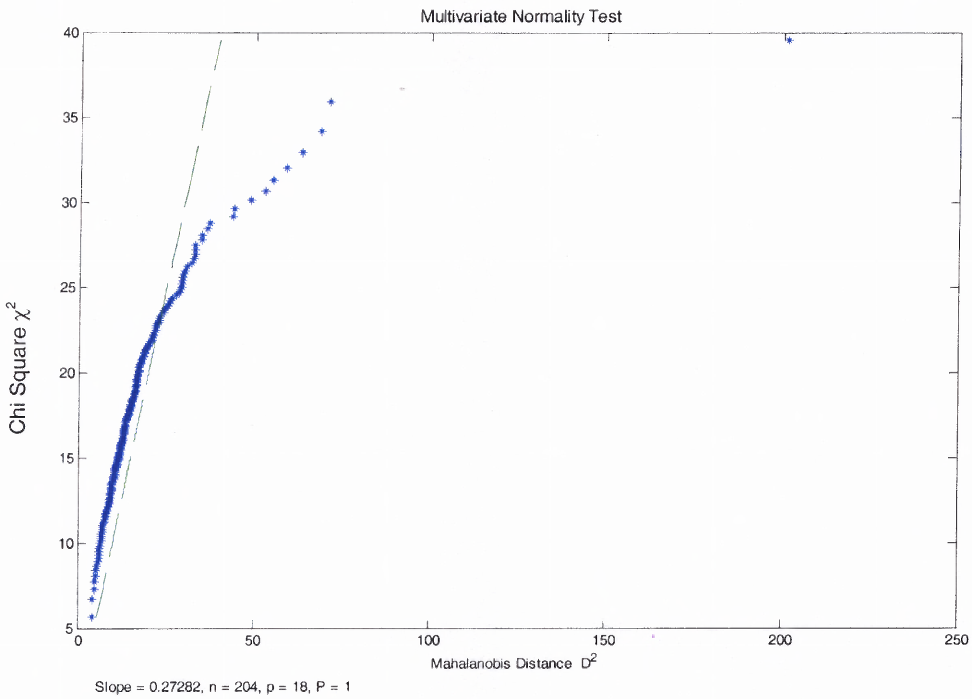


Figure 3.3 Chi-squared quantile-quantile plot of S observations for Subject 1.

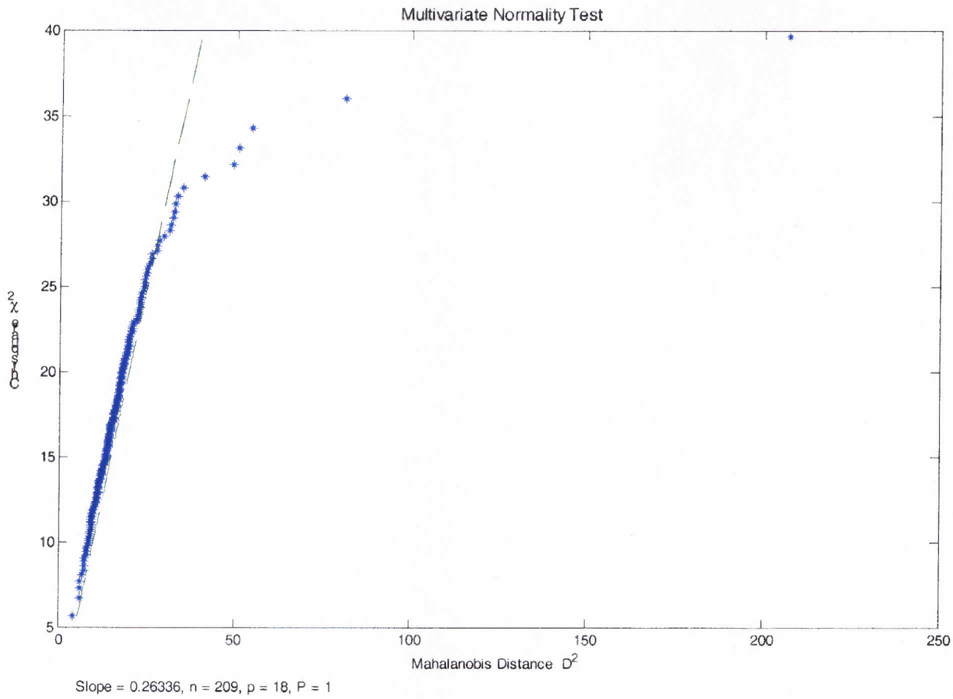


Figure 3.4 Chi-squared quantile-quantile plot of S observations for Subject 2.

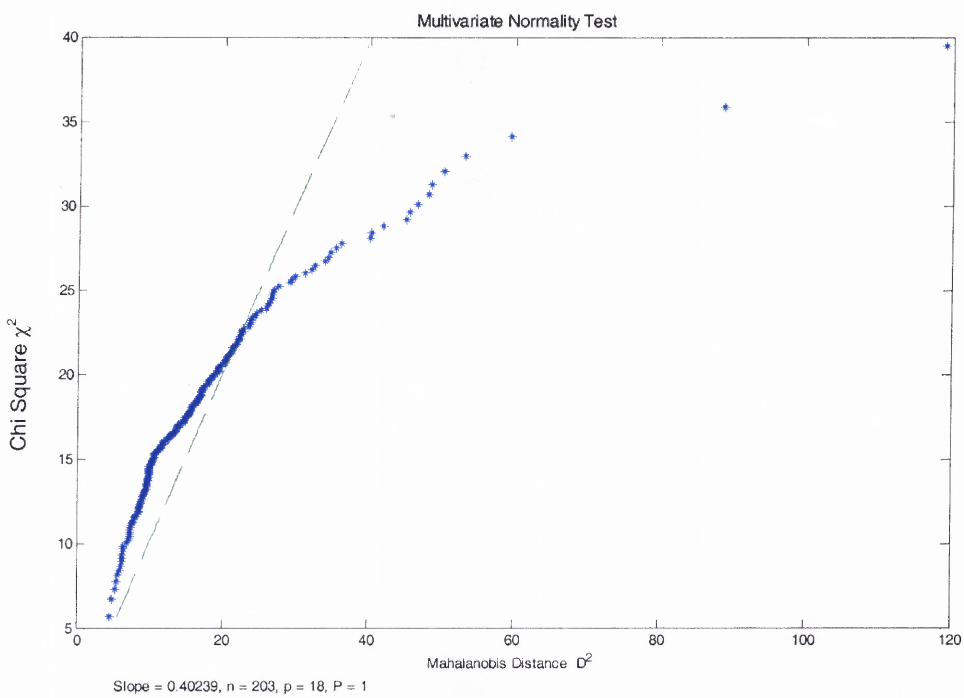


Figure 3.5 Chi-squared quantile-quantile plot of C observations for Subject 1.

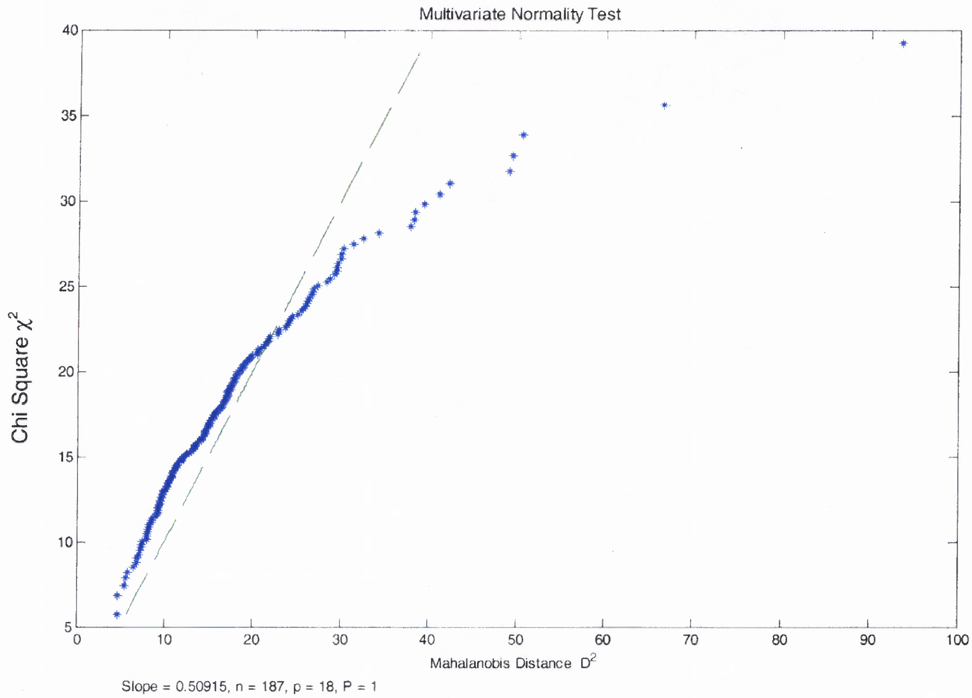


Figure 3.6 Chi-squared quantile-quantile plot of C observations for Subject 2.

3.1.2 Detection of Outliers

The second assumption was that errors, which were significantly different from the average, were transformed or removed. MANOVA involves finding the canonical variate or linear combination of original dependent variables that yields the largest differences between groups. Figures 3.7 to 3.12 show the grouped scatter plots of the first two canonical variables of I, S, and C observations for Subjects 1 and 2, so that the separation between the 21 groups is emphasized. Visual inspection may indicate the presence of outliers, which often appear as clusters with only one member. In each of the figures, the observations with the greatest Mahalanobis distances from its group mean were within 3 standard deviations of the group mean, and were not considered to be outliers.

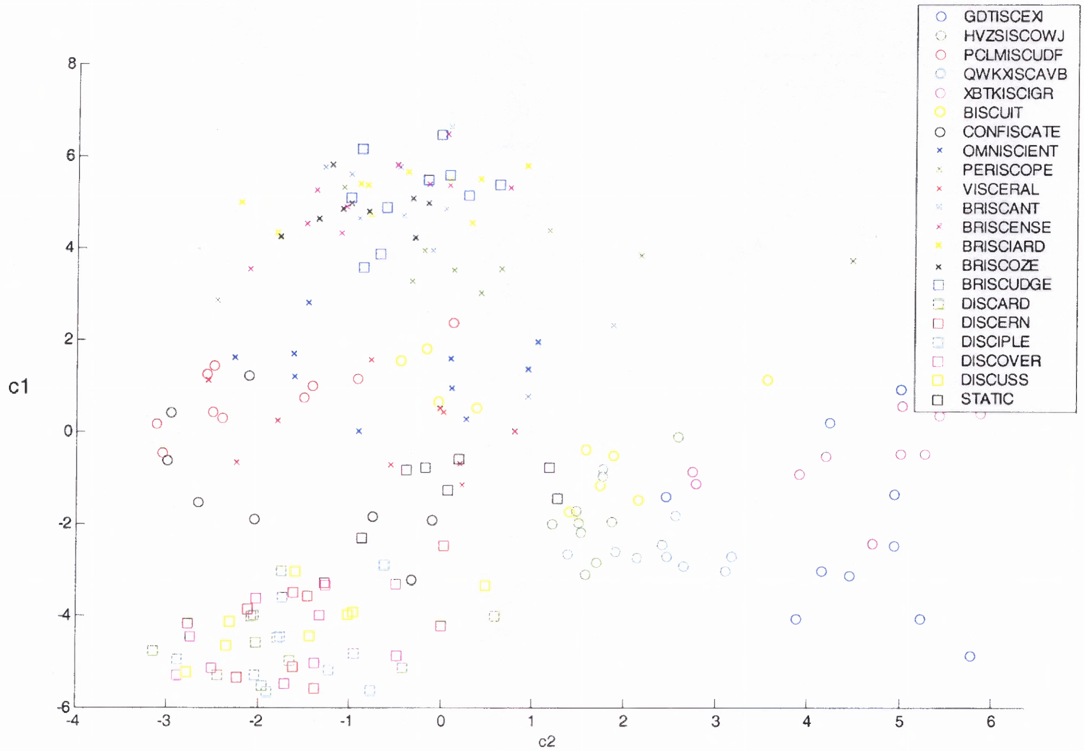


Figure 3.7 Scatter plot of first two canonical variables of I observations for Subject 1.

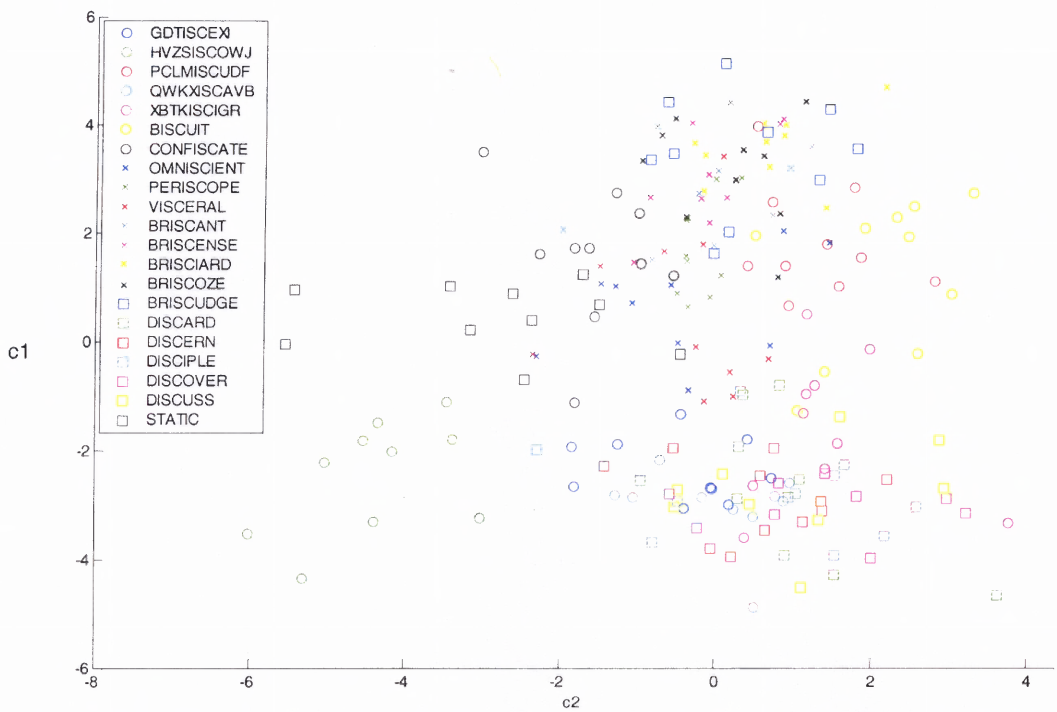


Figure 3.8 Scatter plot of first two canonical variables of I observations for Subject 2.

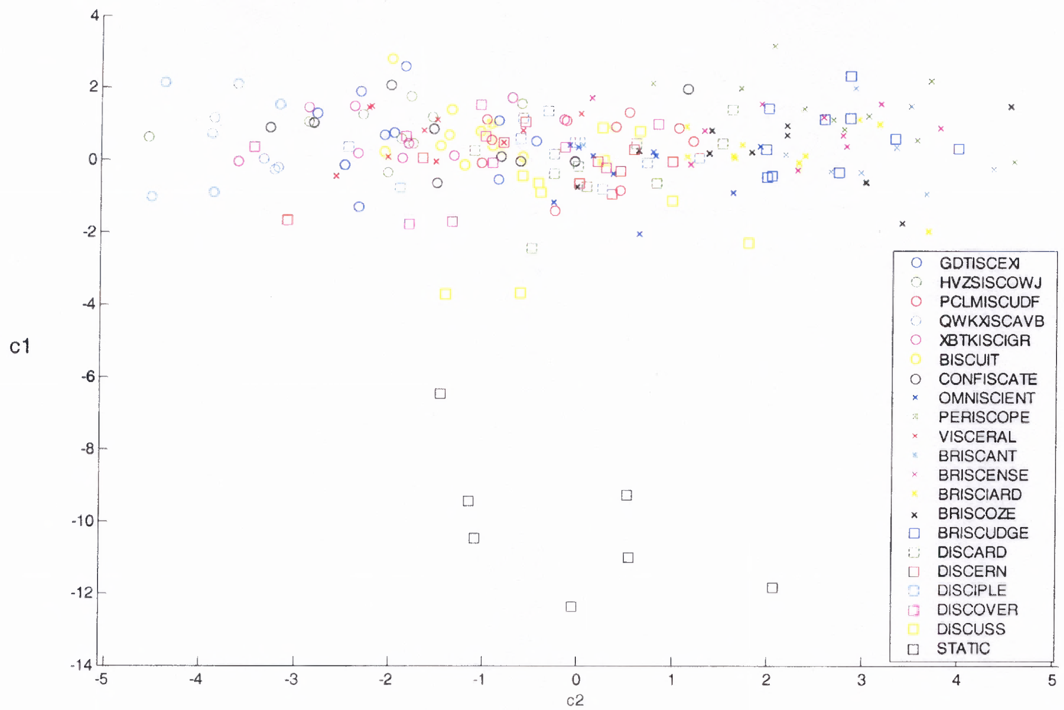


Figure 3.9 Scatter plot of first two canonical variables of S observations for Subject 1.

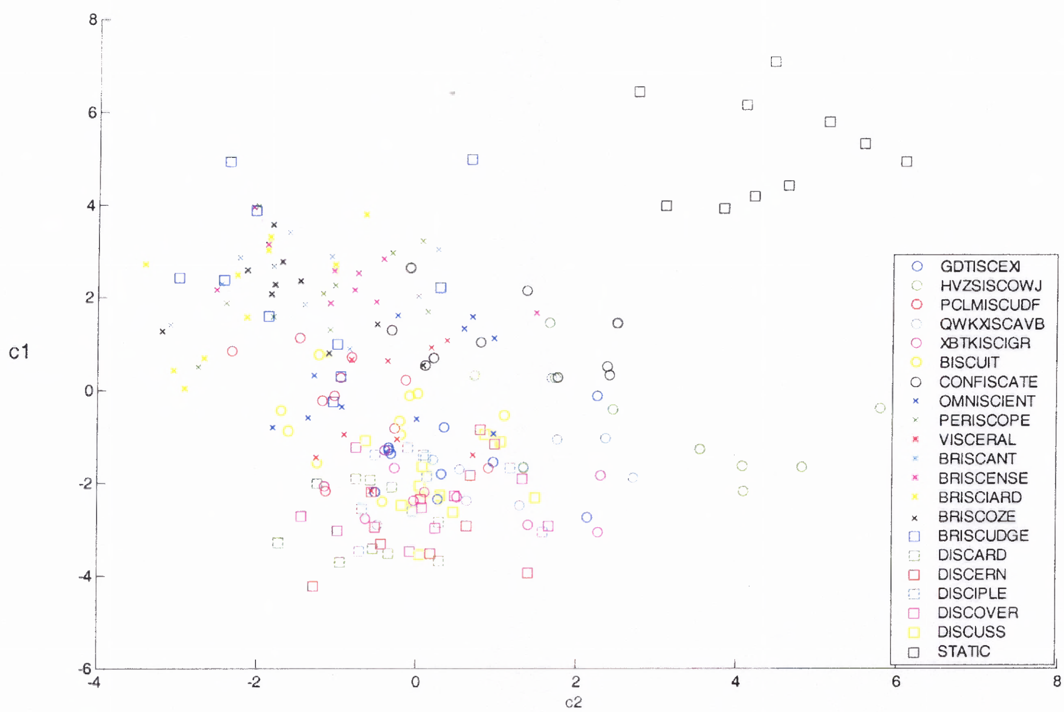


Figure 3.10 Scatter plot of first two canonical variables of S observations for Subject 2.

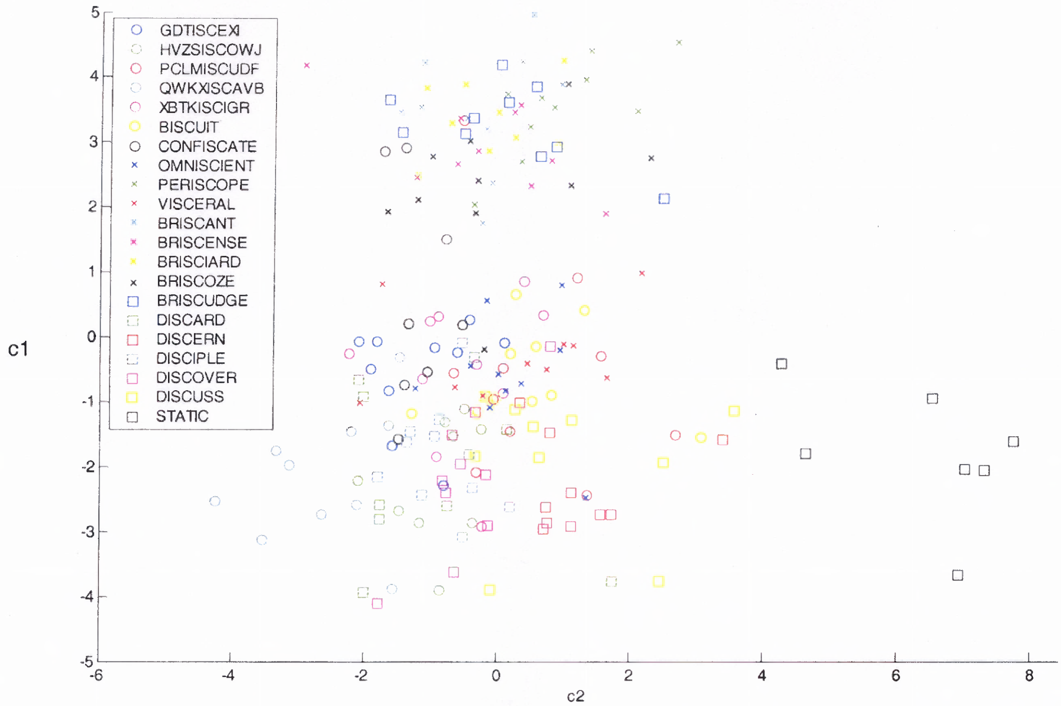


Figure 3.11 Scatter plot of first two canonical variables of C observations for Subject 1.

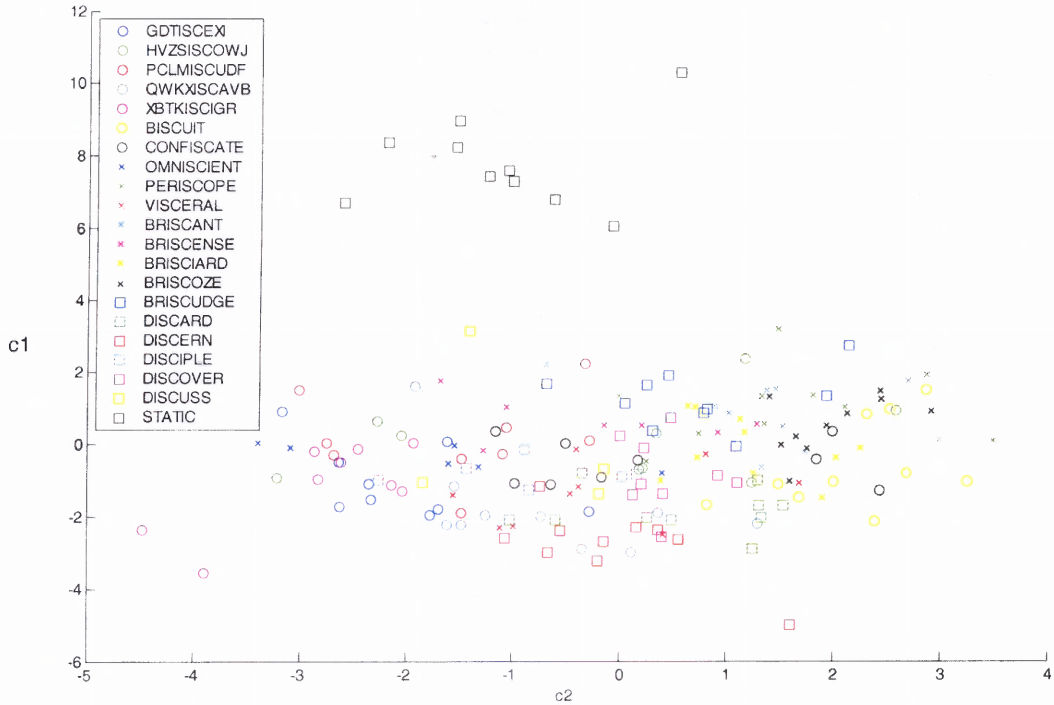


Figure 3.12 Scatter plot of first two canonical variables of C observations for Subject 2.

3.1.3 Homogeneity of Variance

The third usual assumption was that the underlying errors were all uncorrelated with identical variances (homoscedastic, in regression terminology) in the different groups of the design. To test for homogeneity of residual variances, which is whether the standard deviations are significantly different from each another, the values of the dependent variables for each group of independent variables were first converted to dispersion variables, and then a MANOVA was performed on these dispersion variables. The *p*-value for the *F*-test then determined the significance, where the level of significance was set to $\alpha = 0.05$. Appendix C shows the SAS code (PROC GLM) to perform the usual ANOVA test for equal group means followed by Levene's test for homogeneity of variances, which is considered to be the standard.

For each dispersion variable, it computes the difference between the value of the dependent variable and its mean and performs a one-way analysis of variance on those differences. The samples from the populations under consideration are assumed to be independent and the populations under consideration are assumed to be approximately normally distributed.

The null hypothesis of Levene's test, which is that group variances are within random sampling of each other, was accepted for each set of observations, since the p values were greater than the 0.05 cutoff for most of the variables. Even though the variances for all variables proved not to be within sampling error, it was assumed that the MANOVA test would be sufficiently robust to compensate for the slight heteroscedasticity. This is a valid assumption unless the variances are extremely different or the number of groups is large, which was not the case here.

3.1.4 Homogeneity of Covariance

Bartlett's chi-square test was performed to examine the homogeneity of covariance. The results for each data set were highly significant indicating equality of covariance matrices. Regardless, MANOVA would also have been sufficiently robust if this assumption had been violated. The SAS code to request this test is included in the Appendix C under the PROC DISCRIM section which uses the POOL=TEST option to request Bartlett's modification of the likelihood ratio test of the homogeneity of the within-group covariance matrices. As an example of the SAS output, Figure 3.13 shows the results for Bartlett's test for homogeneity of covariance for C observations of Subject1. Since the chi-square values are significant at the 0.01 p level, the within covariance matrices were assumed to be equal.

Chi-Square	DF	Pr > ChiSq
6876.104834	3420	<.0001

Figure 3.13 Test of homogeneity of within-class covariance matrices.

3.1.5 Independence of Observations

The observations were randomly sampled and the joint angle measurements for any one observation were independent from the measurements for all other observations. This is known as the "independence of observations" assumption. There is little room for violation of this assumption, so if it is violated MANOVA should not be conducted.

3.2 MANOVA Theory

The overall logic of MANOVA is to compare two different estimates of the population covariance based on covariance matrices and mean vectors of the groups. The first estimate is the average covariance matrix within the groups (mean squares within groups or error mean squares) and the second is the covariance matrix of the means between the groups (mean squares between groups or hypothesis mean squares). A ratio of the mean squares of the hypothesis divided by the mean squares of the error is called the F statistic and this value is expected to be close to 1.0, or the identity matrix, under the null hypothesis, implying that the mean vectors are the same and that the observations are sampled from the same normal distribution.

$$F = \frac{MS_{hypothesis}}{MS_{error}}$$

As the value of the F statistic increases, so too does the likelihood that the null hypothesis is false. These multivariate F values are obtained with tests such as Pillai's criterion (pooled effect variances), Hotelling-Lawley's trace (pooled ratio of effect variances to error variances), Wilk's lambda (pooled ratio of error variances to effect variance plus error variance) and Roy's largest root (upper bound/ maximum eigenvalue for F statistic). Should any of the assumptions have been violated, Pillai's criterion provides the most conservative analysis.

3.3 MANOVA Design

The MANOVA was performed using the Statistical Analysis System (SAS) program, version 8.2 for Windows. Traditionally, results of a MANOVA test would indicate equality of mean vectors across all groups, without providing any indication as to which mean vectors differed, or which dependent variables were responsible for the inequality. The F statistic simply indicates whether the means of all groups for an ANOVA effect are within random sampling error of each other. Because the null hypothesis for this study required that one specific group be compared to the others, it was necessary to overcome this limitation of traditional MANOVA. This was accomplished through the use of a priori contrast coding; available in contemporary MANOVA testing and implemented in SAS.

The MANOVA was performed on the I, S, and C observations matrices, where the rows consisted of (1x18) vectors of independent observations, and the columns of joint angles represented the dependent variables. These observations were organized into 21 groups corresponding to 20 context groups and 1 static group, and this matrix of groups was the oneway ANOVA factor representing the categories of independent variables. The MANOVA would uncover the direct effect of these independent variables on the dependent variables. Because each group had a different number of independent variables, for reasons mentioned in the previous chapter, the design was unbalanced. Consequently, the PROC GLM (General Linear Model) procedure was used in SAS which compensated for the non-orthogonal nature of the design.

3.4 General Linear Model

MANOVA is a specific case of the general linear model, whose fundamental equation is,

$$Y = X\,\Theta + U$$

where Y is a matrix of observations of dependent variables, X is a design matrix of independent variables as determined by predictors, Θ is a matrix of parameter estimates, and U is a matrix of prediction errors. The general linear equation for multivariate hypothesis tests in MANOVA, where the null hypothesis states that the elements of the parameter matrix are zero, is as follows,

$$H_0 : X\,\Theta\,Y^t = 0$$

$$or \qquad H_0 : L\,\beta M = 0 \text{ (as per SAS notation)}$$

To test this hypothesis of no effect, such that there is no difference between means, the model expresses the 21 group means as linear functions of the overall mean, μ, plus a deviation, α, from this mean. Algebraically, this is equivalent to,

$$H_0 : \overline{Y}_1 = \overline{Y}_2 = \ldots = \overline{Y}_{21}$$

$$\Rightarrow H_0 : \mu + \alpha_1 = \mu + \alpha_2 = \ldots = \mu + \alpha_{21}$$

Knowing the variables of the dependent and independent matrices (L and M, respectively), the parameters of the null hypothesis (β, which corresponds to μ and α) are then solved in order to determine if the deviations from the overall mean are zero. This parameter matrix is estimated by performing a multivariate analysis of variance. The F statistic then indicates the likelihood that the group mean vectors are within sampling error and were drawn from a single normal distribution, but does not indicate which groups contribute to this conclusion. This is addressed with contrast coding.

3.5 Contrast Analysis

In order to determine which group mean vectors are different, multiple t-tests or univariate F tests could be performed for the variables of means in each level of the MANOVA factor. However, this would increase the chance of rejecting the null hypothesis of no mean difference (false positive or Type I error), when in fact there is no mean difference and the hypothesis should be accepted. Instead, a contrast analysis was performed based on the a priori hypothesis that the mean group vector of the static group was statistically different to the average mean group vector of the context groups. This was a planned comparison where the data was deliberately coded in advance to test the natural contrast hypothesis that the joint angles formed by the production of static letters were sampled from a different population than the joint angles formed by the production of context letters. Directly addressing a hypothesis for planned comparison in general linear models is accomplished by creating linear combinations of independent variables from the original independent variables.

These new variables are created by a coding procedure that uses contrast weights to compare the groups of independent variables in accordance with the new hypothesis. The contrasts represent linear combinations of the parameters and are used to test for differences among the levels of the factor (the 21 groups). The dependent variables are then analyzed using these new independent variables.

$$\begin{pmatrix}\text{New}\\ \text{Independent}\\ \text{Variables}\end{pmatrix} = \begin{pmatrix}\text{Matrix}\\ \text{for}\\ \text{Contrast}\\ \text{Codes}\end{pmatrix}\begin{pmatrix}\text{Levels}\\ \text{of}\\ \text{Independent}\\ \text{Variables}\end{pmatrix}$$

This can also be expressed as,

$$X^* = L\,X$$

where X* is the matrix of new independent variables, X is the matrix of old independent variables, and L is the design matrix of independent variables.

In this study, the null hypothesis stated that the mean for the static group of observations was equal to the average mean of the 20 context groups of observations. Algebraically, this is equivalent to,

$$H_0 : \bar{Y}_{21} = \frac{\bar{Y}_1 + \bar{Y}_2 + \ldots + \bar{Y}_{20}}{20}$$

$$\Rightarrow H_0 : \mu + \alpha_{21} = \frac{\mu + \alpha_1 + \ldots + \mu + \alpha_{20}}{20}$$

Multiplying each side by 20 gives,

$$-1\alpha_1 - 1\alpha_2 - 1\alpha_3 \ldots - 1\alpha_{20} + 20\alpha_{21} = 0$$

$$-1\alpha(GDTISCEXI) - 1\alpha(HVZSISCOWJ) - 1\alpha(PCLMISCUDF) \ldots - 1\alpha(DISCUSS) + 20\alpha(STATIC) = 0$$

So, to test the null hypothesis that the 20 context groups did not differ from the static group, the contrast code created a new variable by assigning numeric weights to the levels of the factor, such that the sum of the weights was equal to zero,

$$\begin{bmatrix} \text{CONTEXT} \\ \text{STATIC} \end{bmatrix} = (-1\ -1\ -1\ -1\ -1\ -1\ -1\ -1\ -1\ -1\ -1\ -1\ -1\ -1\ -1\ -1\ -1\ -1\ -1\ -1\ +20) * \alpha \begin{pmatrix} \text{GDTISCEXI} \\ \text{HVZSISCOWJ} \\ \text{PCLMISCUDF} \\ \text{QWKXISCAVB} \\ \text{XBTKISCIGR} \\ \text{BISCUIT} \\ \text{CONFISCATE} \\ \text{OMNISCIENT} \\ \text{PERISCOPE} \\ \text{VISCERAL} \\ \text{BRISCANT} \\ \text{BRISCENSE} \\ \text{BRISCIARD} \\ \text{BRISCOZE} \\ \text{BRISCUDGE} \\ \text{DISCARD} \\ \text{DISCERN} \\ \text{DISCIPLE} \\ \text{DISCOVER} \\ \text{DISCUSS} \\ \text{STATIC} \end{pmatrix}$$

This is equivalent to

$$X^* = L\,X$$

where X* is the matrix of the new independent variables Context and Static, L is the row vector of coefficients or contrast codes (design matrix of independent variables), and X is the matrix of old independent variables. Performing a MANOVA on the new independent variables implies that the null hypothesis, as represented by,

$$H_0 : L\beta M = 0$$

would test whether the average deviation of the 20 context groups from the overall mean were equal to the deviation of the static group from the overall mean, such that L is the (1x21) vector of contrast codes (design matrix of independent variables), M is the matrix of dependent variables, and β is a (22x18) dimensional matrix of parameter estimates, where the first row contains the overall means for each of the dependent variables, and the remaining rows contain the deviations of the variable from the group mean.

3.6 SAS Results

A sample of the SAS code to perform the multivariate analyses is located in Appendix C. The output of the analysis programs on the I, S, and C observations for Subjects 1 and 2 first displayed the univariate ANOVAs for each of the 18 joint angles (dependent variables). This was followed by the results of the tests for homogeneity of variance and covariance. Next, the PRINTE option in the MANOVA statement displayed the elements of the error Sums of Squares and Cross Products (SSCP) matrix. The diagonal elements of this matrix were the error sums of squares from the corresponding univariate analyses. The PRINTE option also displayed the partial correlation matrix associated with the error matrix. The Type III sums of squares were calculated by default which adjusted its calculation to account for the unbalanced design. Otherwise the total sum of squares would not equal the hypothesis sum of squares and the error sum of squares. Finally, the MANOVA using planned comparisons was implemented by using the CONTRAST statement, under PROC GLM.

Four multivariate tests (Wilks' Lambda, Pillai's Criterion, Hotelling-Lawley Trace and Roy's Greatest Root) were computed, and all were based on the characteristic roots and vectors of $E^{-1}H$, which is the product of the inverse of the error SSCP matrix and the hypothesis SSCP. These roots and vectors were also displayed along with the tests. All four statistics were transformed to variates that had F distributions under the null hypothesis.

The univariate results for the dependent variables showed mixed levels of significance throughout the data sets. There was no evidence of differences between means for some of the joints and favorable evidence for differences in others. It was possible that some of the results may have been false positives or negatives. This was cleared up by the multivariate analysis, which confirmed that there was, indeed, an overall difference between the group mean vectors of the joint angle configurations of the hands, and angles of the letters produced statically were significantly different from the average of the angles of the letters produced within a string. In each case, the four tests all gave the same results for the contrast, since it had only one degree of freedom. The following tables summarize the results of the contrast hypothesis coding.

Table 3.2 MANOVA Results for Static versus Context Groups of I Observations

MANOVA Test Criteria and Exact F Statistics for the Hypothesis of No Overall Static vs Context Effect H = Contrast SSCP Matrix for Static vs Context E = Error SSCP Matrix I Observations										
	Subject 1 (S=1 M=8 N=80.5)					**Subject 2 (S=1 M=8 N=84.5)**				
Statistic	Value	F	DFn	DFd	Pr > F	Value	F	DFn	DFd	Pr > F
Wilks' Lambda	0.50596823	8.84	18	163	<.0001	0.44777024	11.72	18	171	<.0001
Pillai's Trace	0.49403177	8.84	18	163	<.0001	0.55222976	11.72	18	171	<.0001
Hotelling-Lawley Trace	0.97640868	8.84	18	163	<.0001	1.23328823	11.72	18	171	<.0001
Roy's Greatest Root	0.97640868	8.84	18	163	<.0001	1.23328823	11.72	18	171	<.0001

Table 3.3 MANOVA Results for Static versus Context Groups of S Observations

MANOVA Test Criteria and Exact F Statistics for the Hypothesis of No Overall Static vs Context Effect
H = Contrast SSCP Matrix for Static vs Context
E = Error SSCP Matrix
S Observations

	Subject 1 (S=1 M=8 N=82)					**Subject 2 (S=1 M=8 N=84.5)**				
Statistic	Value	F	DFn	DFd	Pr > F	Value	F	DFn	DFd	Pr > F
Wilks' Lambda	0.19583673	37.87	18	166	<.0001	0.26804964	25.94	18	171	<.0001
Pillai's Trace	0.80416327	37.87	18	166	<.0001	0.73195036	25.94	18	171	<.0001
Hotelling-Lawley Trace	4.10629449	37.87	18	166	<.0001	2.73065223	25.94	18	171	<.0001
Roy's Greatest Root	4.10629449	37.87	18	166	<.0001	2.73065223	25.94	18	171	<.0001

Table 3.4 MANOVA Results for Static versus Context Groups of C Observations

MANOVA Test Criteria and Exact F Statistics for the Hypothesis of No Overall Static vs Context Effect H = Contrast SSCP Matrix for Static vs Context E = Error SSCP Matrix C Observations										
	Subject 1 (S=1 M=8 N=81.5)					**Subject 2 (S=1 M=8 N=73.5)**				
Statistic	Value	F	DFn	DFd	Pr > F	Value	F	DFn	DFd	Pr > F
Wilks' Lambda	0.33528625	18.17	18	165	<.0001	0.19728394	33.68	18	149	<.0001
Pillai's Trace	0.66471375	18.17	18	165	<.0001	0.80271606	33.68	18	149	<.0001
Hotelling-Lawley Trace	1.98252615	18.17	18	165	<.0001	4.06883629	33.68	18	149	<.0001
Roy's Greatest Root	1.98252615	18.17	18	165	<.0001	4.06883629	33.68	18	149	<.0001

The F statistics found using all four of the tests were identical and highly significant. It is clear from the MANOVA that differences between the means of the static and context groups are highly likely, as indicated by the F values in the table and their associated p levels, which imply that the probabilities of obtaining such differences by chance are extremely low (less than 0.01%). The p value for the F is the area under the F distribution with n and d degrees of freedom from F to positive infinity. Hence, for Subject 2, C observations, it is the probability of observing an F of 33.68 or greater from an F distribution with (18, 149) degrees of freedom. Because the confidence level was set to 0.05, these low p-values suggested that the null hypothesis of no difference must be rejected.

3.7 Matlab Results

The Statistics Toolbox in Matlab also allows for multivariate analysis. The function MANOVA1 produces a number of output structures including the stats structure which contains the field GMDIST. This is a measurement of the Mahalanobis distance between each pair of group means. The following confusion matrices are visual representations of the multivariate distances between means for I, S, and C of Subjects 1 and 2. The colormaps are grayscale, where the maximum distance is represented by black and the minimum distance is represented by white. Figures 3.14 and 3.16 shows the Mahalanobis distances between each of the 21 groups for the I observations for each subject. This is followed in Figures 3.15 and 3.17 by a dendrogram of the same thing, where the heights of the connecting lines illustrate the distance between connected groups. The context groups are represented by the first 20 columns and rows and the static group is represented by the final column and row. Notice the lightly colored square blocks, highlighted with the red circles. These represent the categories of Same Initial Letter Word (Discard, Discern, Disciple, Discover, Discuss) and Same Initial Letter Non-Word (Briscant, Briscense, Brisciard, Briscoze, Briscudge). The fact that the distances within these groups are similar and minimal, by visual inspection, indicates the presence of carryover coarticulation, since the only difference between the hand configurations of each block is the "D" or "R" that occurred prior to the "I". Interestingly, the mean vectors of the hand configurations for the I's produced after the D's in the Same Initial Letter Word (SILW) category were further from the mean vectors of the hand configurations for the I's produced after the R's in the Same Initial Letter Non-Word (SILN) category, as indicated by the black blocks. The others are arbitrary in accordance with the fact that the letters performed prior to the "I" were different. It is interesting to note that the mean vectors between groups were not maximally distant from the mean vector of the static group in the case of the I observations.

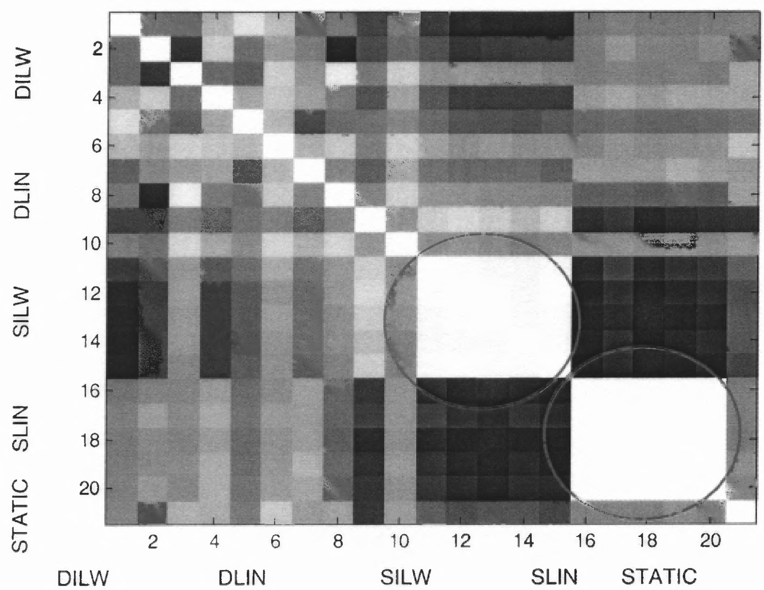


Figure 3.14 Confusion matrix of Mahalanobis distances between group means for Subject 1, I observations.

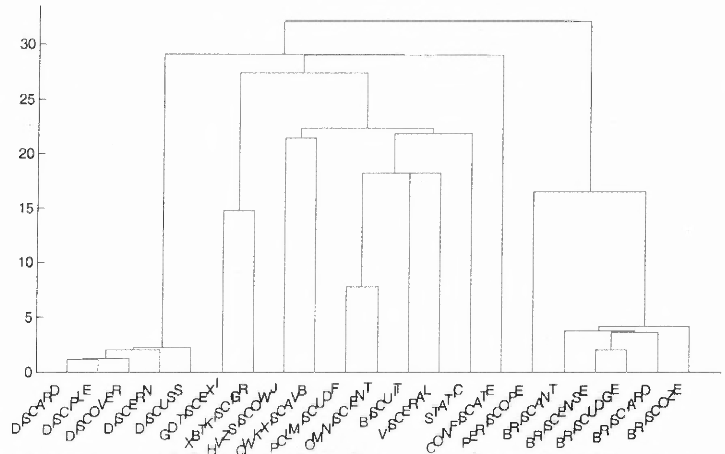


Figure 3.15 Dendrogram of Mahalanobis distances between group means for Subject 1, I observatior

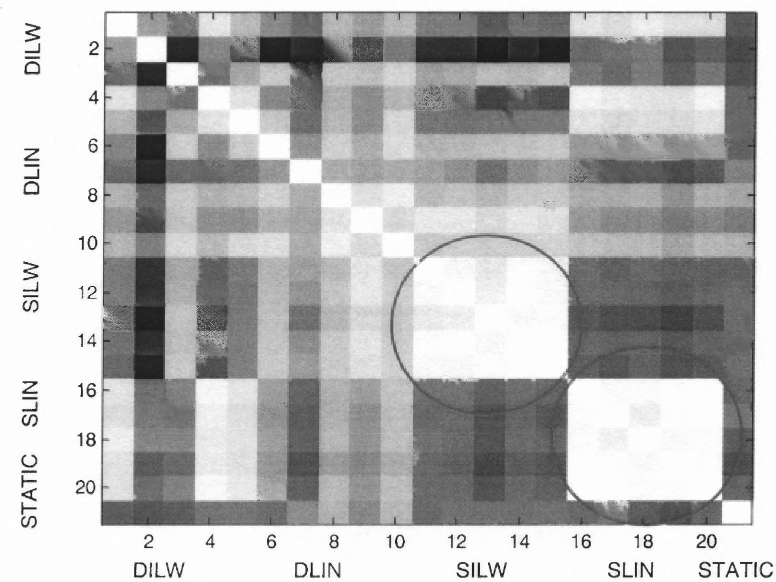


Figure 3.16 Confusion matrix of Mahalanobis distances between group means for Subject 2, I observations.

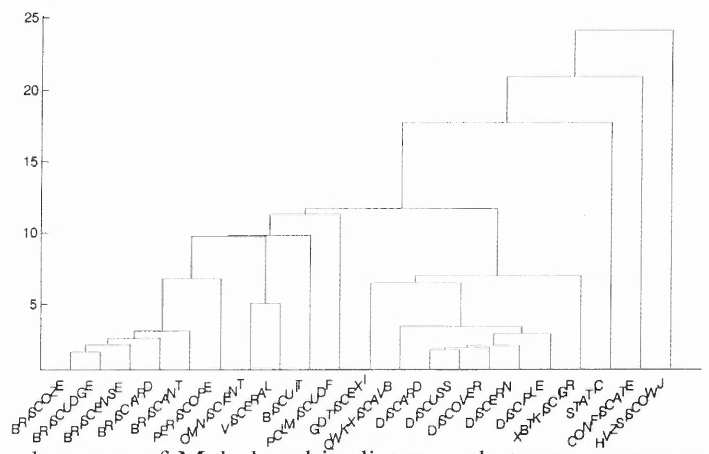


Figure 3.17 Dendrogram of Mahalanobis distances between group means for Subject 2, I observations.

For the S observations (as represented by the next four figures), which correspond to the second letter in the trigram, the effect of the same letter prior to the "I" can still be seen by the white blocks. At this point the "D" and the "R" were executed two letters before the "S". This gives some indication as to the domain of coarticulation.

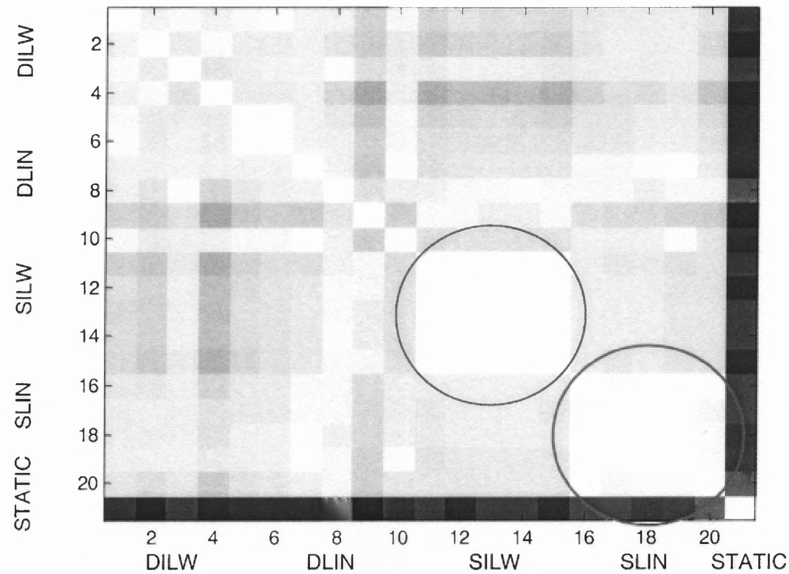


Figure 3.18 Confusion matrix of Mahalanobis distances between group means for Subject 1, S observations.

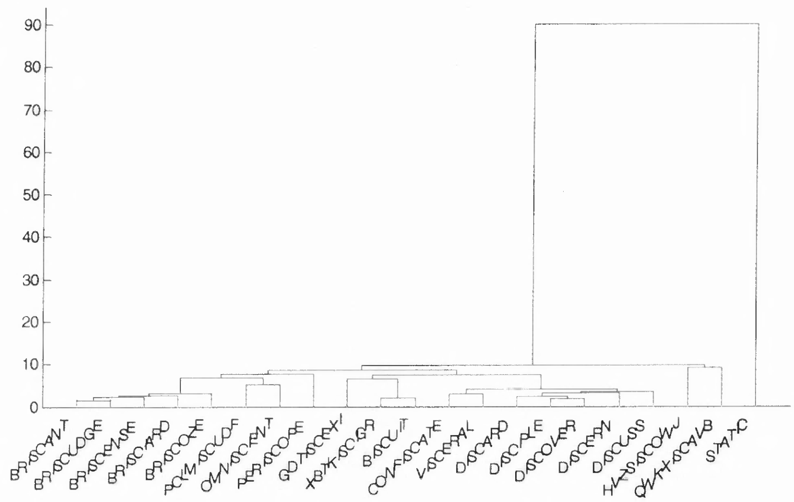


Figure 3.19 Dendrogram of Mahalanobis distances between group means for Subject 1, S observations.

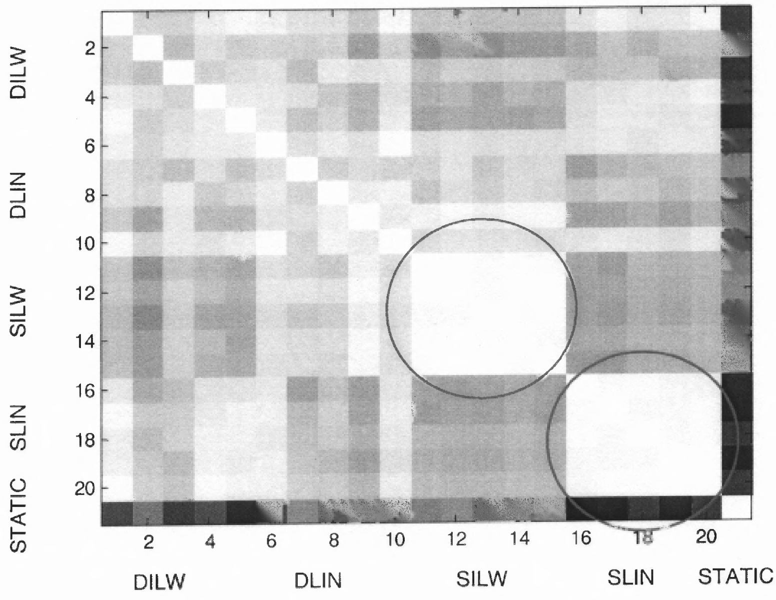


Figure 3.20 Confusion matrix of Mahalanobis distances between group means for Subject 2, S observations.

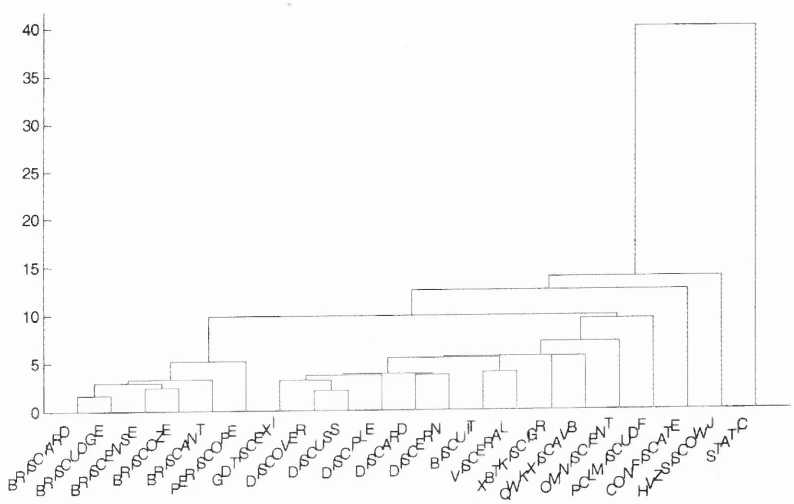


Figure 3.21 Dendrogram of Mahalanobis distances between group means for Subject 2, S observations.

For the C observations (as represented by the last figures), which correspond to the third letter in the trigram, the effect of the same letter prior to the "I" can still be seen by the white blocks. At this point the "D" and the "R" were executed three letters before the "C". Again, this may indicate the domain of coarticulation as these hand configurations were all performed after the same letter, "S". The effect is less obvious in the data for Subject 2, which may mean that the domain is signer dependent.

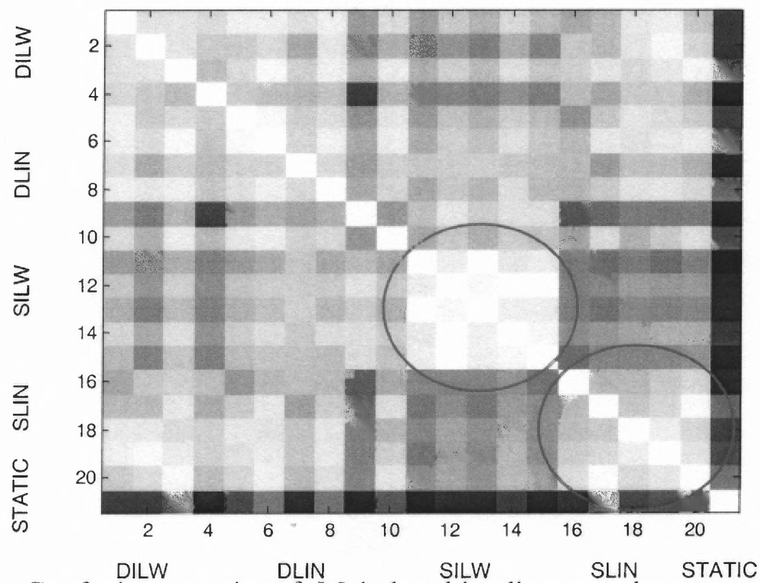


Figure 3.22 Confusion matrix of Mahalanobis distances between group means for Subject 1, C observations.

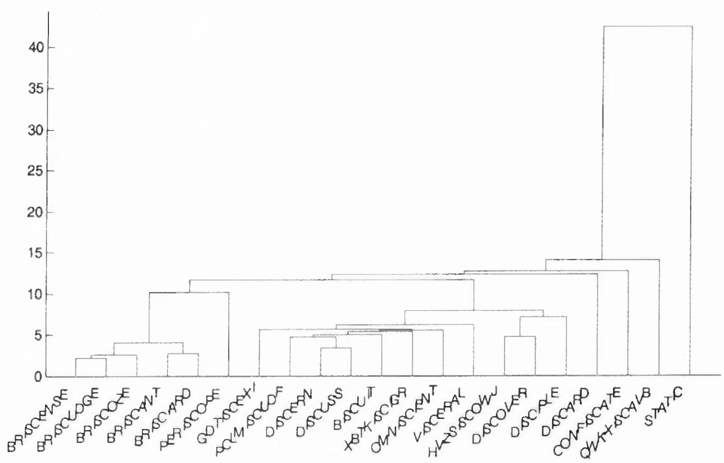


Figure 3.23 Dendrogram of Mahalanobis distances between group means for Subject 1, C observations.

The dark stripes along the 21st dimensions illustrate the vast difference between the means of the static and context groups as opposed to the differences among the context groups.

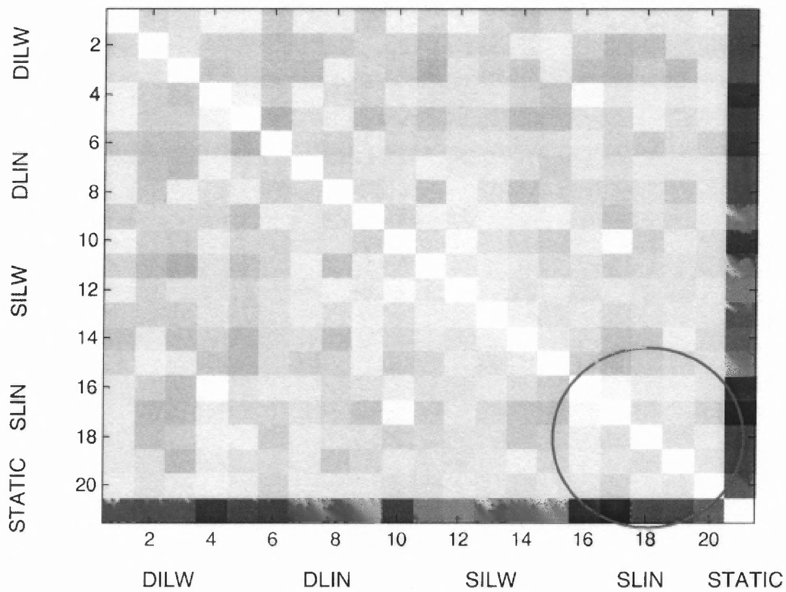


Figure 3.24 Confusion matrix of Mahalanobis distances between group means for Subject 2, C observations.

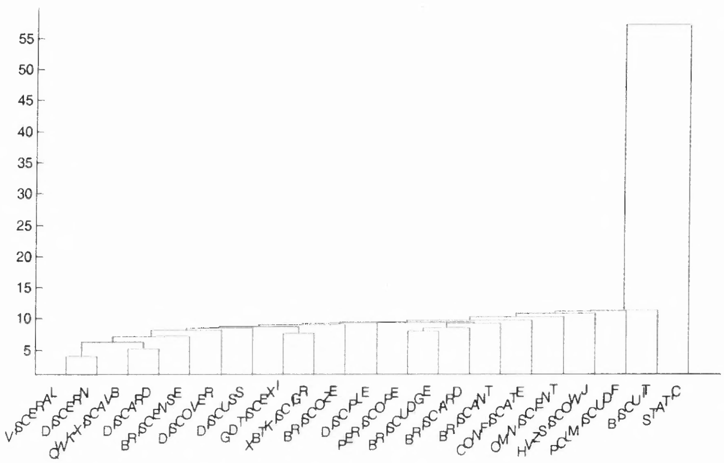


Figure 3.25 Dendrogram of Mahalanobis distances between group means for Subject 2, C observations.

CHAPTER 5

DISCUSSION

The study provided favorable evidence that the simple, cipher model of static handshapes produced within the context of a movement sequence is not sufficient to account for the production and perception of fingerspelling. Rather, the results suggest that it may be insightful to use targets that are based on dynamic configurations in order for a fingerspelling model to be more accurate. Such target modification can be attributed to the presence of coarticulation in the signs, which are influenced by the kinematic characteristics of surrounding signs. The coarticulation is likely made manifest as assimilation and dissimilation in order to emphasize and reduce differences between finger positions, thereby aiding in the letter recognition. Past research has revealed that joints of the thumb and wrist tend to show assimilation whereas the proximal interphalangeal joints of the middle and index fingers tend to show dissimilation (Jerde).

The findings are in keeping with Wilcox's prediction that "if learners [of fingerspelling] are looking for static, canonical, invariant hand configurations, then it is little wonder why understanding fingerspelling is so difficult. They are looking for something that simply is not there." He made this assessment on the basis of a study he conducted to determine the temporal variation in the production of targets versus transitions. He found that the difference between the duration of targets and transitions in dynamic fingerspelling was significant, $F_{(1,291)} = 473.294$, $p < .001$.

This emphasizes the importance of transitions in their contribution to the overall comprehension of fingerspelling. Modifications of this study might consider an examination of transitions in conjunction with targets. Even in a target-oriented scheme, detection of targets may well occur during the approach or retreat phases, centered about the time points of minimum summed joint velocity. Rather than examining the handshape at these points alone, an improved scheme might account for this range of configurations so that joint angles within a certain range could be detected for examination.

A number of factors may have influenced the automatic recognition of letters, and in turn, the accuracy of the data on which the MANOVAs were performed. Namely, bias and transposition errors may have decreased the accuracy of the data. There may have some bias introduced into the data as a result of the fit of the glove, worn by the subjects during data collection. The open-fingered design of the 18-sensor glove results in a looser fit, particularly among female subjects with smaller physiological measurements of the hand. This may have affected the accuracy of the joint angle information picked up by the sensors.

Additionally, transposition errors may have prevented classification of letters in the predicted order, which limited automatic post-hoc analyses. Transposition errors refer to the switching of elements of a sequence and are considered one of the "cardinal characteristics of serial order processes" (Averbeck et al.). Another aspect that limited the automatic letter recognition was the sensitivity of the program to biphasic speed profiles, where surrounding transitions about a target reflect the adjustment of finger positions causing a dip in velocity. Such points were sometimes interpreted as local minima, resulting in classifications at time points other than when targets were reached.

CHAPTER 4

CONCLUSION

The goal of this thesis was to conduct a scientific investigation into a specific aspect of sign organization with the intention of gaining greater insight into ASL's phonological structure. The main objective was to identify a significant amount of variability in underlying sign structure, and in conclusion, the findings of the study offer some evidence in favor of this theory. The difference between static and dynamic sign configurations was found to be significant for each letter that was analyzed ($F_{I1(18,163)} = 8.84$, $p < .0001$, $F_{I2(18,171)} = 11.72$, $p < .0001$, $F_{S1(18,166)} = 37.87$, $p < .0001$, $F_{S2(18,171)} = 25.94$, $p < .0001$, $F_{C1(18,165)} = 18.17$, $p < .0001$, $F_{C2(18,149)} = 33.68$, $p < .0001$).

These results, in consonance with earlier studies, indicate that the simple, cipher model of static handshapes produced within the context of a movement sequence is not sufficient to account for the production and perception of fingerspelling, and that a model of dynamic handshapes would be more appropriate. Although it cannot be assumed, on the basis of two subjects, that findings can be applied to the entire population it is reasonable to assert that the results provide favorable evidence in that direction with further investigation in order.

CHAPTER 6

FUTURE DIRECTION

The overall goal was to facilitate further progress toward fluent sign and fingerspelling recognition as well as to gain a greater understanding of the manifestation of the organizational levels of language production and visual-spatial functions. Data concerning handshape variability in dynamic fingerspelling may be an invaluable resource for testing theories on sign language production and perception in the future. Although important, this study offers limited discussion in the neurological basis for the organization of sign and further research is called for to investigate this directly observable language as a window into the brain and cognition.

APPENDIX A

MATLAB SOURCE CODE

This appendix contains the annotated Matlab source code for all the programs referenced throughout this thesis. They were written to aid in the analysis of data collected.

A.1 ABSSUMVEL.M

```
function [abs_summed_vel] = abssumvel(string)
% ABSSUMVEL Calculates the absolute summed velocity of a string


% Initialize
global TIME;
x = string;
[nrow,ncol] = size(x);


% Calculate the first derivative of angular position i.e. angular velocity
for i = 2:nrow - 1
   for k = 1:ncol
      ang_vel(i, k) = (x(i + 1, k) - x(i - 1, k)) / (TIME(i + 1) - TIME(i - 1));
   end
end
```

```
% A different three point method exists which allows us to evaluate the value of y' at the
endpoints.
for i = 1
 for k = 1:ncol
 ang_vel(i, k) = (- 3*(x(i, k)) + 4*(x(i + 1, k)) - x(i + 2, k) ) / 2*(TIME(i + 1) - TIME(i));
 end
end

% Notice that this formula uses -1 to replace 1 for the final endpoint
for i = nrow
 for k = 1:ncol
 ang_vel(i, k) = (- 3*(x(i, k)) + 4*(x(i - 1, k)) - x(i - 2, k) ) / -(2*(TIME(i) - TIME(i - 1)));
 end
end

% Take the absolute values of all velocities
for i = 1:nrow
        for k = 1:ncol
    if (ang_vel(i, k) < 0)
        abs_ang_vel(i, k) = -ang_vel(i, k);
    else
        abs_ang_vel(i, k) = ang_vel(i, k);
    end
```

```
    end
end

% Sum these absolute values and return
abs_summed_vel = sum(abs_ang_vel, 2);
```

A.2 PLOT_SAV.M

```
function[] = plot_SAV(string)

% PLOT_SAV  Plots sum of angular joint velocities


% Initialize

global TIME;

x = string;


% Calculate the absolute summed velocity of the string x

asv = abssumvel(x);

[nrow_asv,ncol_asv] = size(asv);


% Plot sum of angular joint velocities against time

figure, plot(TIME(1:nrow_asv),asv);

hold on;

xlabel('Time(sec)');

ylabel('Velocity(deg/sec)');

title('Sum of angular joint velocities');
```

A.3 PLOT_SAV*AND*MINIMA.M

```
function[lmval, indd, abs_summed_vel] = plot_savANDminima(string, nmin)

% PLOT_MINIMA   Plots both the sum of angular joint velocities and all local
%               minima, unless the number of minima required is specified.
%               Returns a vector of absolute summed velocities a well as all
%               the local minimum values and their corresponding indices.


% Initialize
global TIME;
x = string;


plot_SAV(x)
hold on;


% Return values of local minima and their respective indices and plot
if(nargin >= 2)
   filtlevel = 10;
   [lmval indd abs_summed_vel] = local_minima(x);
   while (length(indd) > nmin)
         filtlevel = filtlevel + 1;                % ensures that only nmin minima are found
         [lmval indd abs_summed_vel] = local_minima(x, filtlevel);
   end
   while (length(indd) < nmin)                     % ensures that only nmin minima are found
```

```
            filtlevel = filtlevel - 1;
            [lmval indd abs_summed_vel] = local_minima(x, filtlevel);
    end
else
    [lmval indd abs_summed_vel] = local_minima(x);
end

% Plot against time
plot(TIME(indd), abs_summed_vel(indd),'r+');
```

A.4 LMIN.M

```
function [lmval,indd]=lmin(xx,filt)

%LMIN Finds local minima in vector X, where LMVAL is the output
%       vector with minima values, INDD is the corresponding indices
%       FILT is the number of passes of the small running average filter
%       in order to get rid of small peaks.  Default value FILT =0 (no
%       filtering). FILT in the range from 1 to 3 is usually sufficient to
%       remove most small peaks
%
%*************************************************|
%       Serge Koptenko, Guigne International Ltd., |
%       phone (709)895-3819, fax (709)895-3822     |
%--------------06/03/97---------------------------|


x=xx;
len_x = length(x);
fltr=[1 1 1]/3;
if nargin <2, filt=0;
else
        x1=x(1); x2=x(len_x);
        for jj=1:filt,
        c=conv(fltr,x);
        x=c(2:len_x+1);
```

```
        x(1)=x1;
        x(len_x)=x2;
        end
end

lmval=[];
indd=[];
i=2;                        % start at second data point in time series

while i < len_x-1,
if x(i) < x(i-1)
        if x(i) < x(i+1)        % definite min
                lmval =[lmval x(i)];
                indd = [ indd i];

        elseif x(i)==x(i+1)&x(i)==x(i+2)    % 'long' flat spot
                i = i + 2;                  % skip 2 points

        elseif x(i)==x(i+1)     % 'short' flat spot
                i = i + 1;                  % skip one point
        end
end
i = i + 1;
```

```
end


if filt>0 & ~isempty(indd),
    if (indd(1)<= 3)|(indd(length(indd))+2>length(xx)),
        rng=1;                          %check if index too close to the edge
    else rng=2;
    end


    for ii=1:length(indd),
        [val(ii) iind(ii)] = min(xx(indd(ii) -rng:indd(ii) +rng));
        iind(ii)=indd(ii) + iind(ii)  -rng-1;
    end
indd=iind; lmval=val;
else
end
```

A.5 LOCAL_MINIMA.M

```
function[lmval, indd, abs_summed_vel] = local_minima(string, filt)

% LOCAL_MINIMA  Calculates the absolute summed velocity of a string and its

%               local minima values and the indices where these occur


x = string;


abs_summed_vel = abssumvel(x);


if (nargin == 1)

   filt = 10;                      % number of passes of the small running average filter

end


% Return values of local minima and their respective indices

[lmval, indd]=lmin(abs_summed_vel, filt);
```

A.6 FILTERDATA.M

```
function [filtered_data] = filterData(data, fc_butter)

% FILTERDATA Smooths data with a bi-directional zero-phase-lag Butterworth filter.
%       The original filter is a second order, which is applied twice, creating an effective
%       fourth order filter. Unless otherwise specified, the overall cutoff frequency is set
%       to 10Hz.

% Initialize
global FS;
order = 2;

if (nargin < 2)
    fc_butter = 10;                 % original filter then has a cutoff of 10 / 0.802 = 12.2Hz
end                                 % fc_orig = fc_butter/0.802, fc_butter = fc_orig*0.802

nyquist = 1/2;

% normalized cutoff frequency = cutoff freq/(FS/2)
[b,a] = butter(order, (fc_butter/0.802)/(FS*nyquist), 'low');

filtered_data = filtfilt(b,a,data);
```

A.7 CLASSIFYSTATIC.M

```
% CLASSIFYSTATIC script prompts the user to select an Excel string to be classified
%       and to select the subject to whom the string belongs. The entire string is classified
%       and the confusion matrix of Mahalanobis distances is displayed to illustrate
%       classification. Summed angular velocity for the joints are calculated and plotted.
%       All local minima are calculated, plotted and classified and confusion matrix is
%       plotted.

% Initialize
global FS;

% Prompt for subject selection. Check for valid entry
subject = input('Classify string for subject 2 or 4? ');
if (subject == 2)
   inp = xlsread('S2train');
elseif (subject == 4)
   inp = xlsread('S4train');
else
   error('Invalid subject number. Please start over.');
end

% Load static hand postures for relevant subject as training data into variable inp
```

```
% and isolate 18 CyberGlove joint angles
% Static hand postures consist of moving average of final 60 samples of each trial
inp = inp(:,1:18);
inp_filtered = filterData(inp);

% Prompt for file selection.
[filename, pathname] = uigetfile('*.xls', 'Select an EXCEL string for classification to be
trained on static postures');
string = xlsread([pathname,filename]);
string = string(25:300,3:20);
string_filtered = filterData(string);
x = string_filtered;
[nrow_s,ncol_s] = size(x);

% Set up for discriminant analysis
ncls = 26;      % number of classes is 26 (letters)

% Allocate each of the samples to the class to which it belongs
if (subject == 2)
  for i = 1:ncls
    clsinfo(1,1+((i-1)*7):i*7) = i;
  end
elseif (subject == 4)
```

```
    for i = 1:ncls
        clsinfo(1,1+((i-1)*10):i*10) = i;
    end
end

% Class size = Number of samples for each of the the 26 letter classes of the alphabet
if (subject == 2)
    for i = 1:ncls
        clsize(1,i) = 7;
    end
elseif (subject == 4)
    for i = 1:ncls
        clsize(1,i) = 10;
    end
end

% Classify entire string and display confusion matrix
[letter] = ldam(inp_filtered,ncls,clsinfo,clsize,x);

% Plot the summed angular joint velocity and the local minima
[lmval, indd] = plot_savANDminima(x);

% Classify data at local minima into correct classes based on subject's
```

```
% static training data. (NOTE that string consists of 25:200 of original
% data)
[letter] = ldam(inp_filtered,ncls,clsinfo,clsize,x(indd,1:ncol_s));
```

A.8 LDAM.M

```
function [letter, min_mahal, mvect] = ldam(train,ncls,clsinfo,clsize,pred)

% Linear Discriminant Analysis-Mahalanobis

% Assigns ungrouped items to closest group center, using the Mahalanobis

% distance measure (i.e., minimum distance classifier).

% Author          Ron Shaffer

% Modified        Gillian Sherry

%

% letter          letter classification

% mvect           mean vectors for each class

% pcov            pooled covariance matrix

% tscor           training set mahalanobis distance scores

% pscor           prediction set mahalanobis distance scores

% tmislist        training set misclassified list

% pmislist        prediction set misclassified list

% dist_sum        summation of mahalanobis distances between classes

% tpct            training percentage correct

% ppct            prediction percentage correct

% train           training set data

% ncls            number of classes

% clsinfo         classification of each member of the training set (vector)

% clsize          class sizes (vector)

% pred            prediction set data
```

```
% Revisions    Verison 1.0 4/18/96 (original code)
%              Version 1.1 4/19/96 fixed bug in pooled covariance calculation
%              Version 1.2 5/28/96 fixed bug in calculation of prediction % correct
%              Version 1.3 6/17/96 Added calculation of the sum of between-class
distances
%              Version 1.4 6/17/96 Return distance and pct correct
%              Version 1.5 6/26/96 fixed bug in calculation of mean vectors and
%                                  pooled cov caused by having data classes with
%                                  different #s of patterns.
%              Gillian Sherry fixed bug in computing training results by adding tmislist =
0
%
% Note: Based on MINITAB function (DISCRIMINANT) and Fortran subroutine
% LDA by F. Murtagh available from the Statlib FTP site.
%
[nrow_t,ncol_t] = size(train);
[nrow_p,ncol_p] = size(pred);
dist_sum = 0;

% Split training data into respective classes
count = 1;
covsum = zeros(ncol_t,ncol_t);
for i = 1:ncls
```

```
        for j = 1:clsize(i)
                % Fill data(temporarily) with contents of each training class
                data(j,1:ncol_t) = train(count,1:ncol_t);
                count = count + 1;
        end
        mvect(i,1:ncol_t) = mean(data);          % Compute mean of that class
        covmat = cov(data)*(clsize(i)-1);        % Compute covariance matrix of that class
        covsum = covsum + covmat;                % Sum of covariance matrices for all classes
        clear data;
end

% Compute pooled covariance matrix
pcov = covsum/(sum(clsize)-ncls);
c = inv(pcov);

% Mahalanobis distance between classes
% (x-m)'C(x-m) = m.d.^2
for i = 1:ncls-1
        for j = i+1:ncls
                temp = (mvect(i,:)-mvect(j,:)) * c * (mvect(i,:)-mvect(j,:))';
                dist_sum = dist_sum + temp;
        end
end
```

```
% Now compute training results

tmislist = 0;

count = 1;

missed = 1;

tcor(1:ncls) = zeros(size(1:ncls));

for i = 1:ncls

    for j = 1:clsize(i)

        for k = 1:ncls

        tscor(count,k) = (mvect(k,:)-train(count,:))*c*(mvect(k,:)-train(count,:))';

        end

        temp(1:ncls) = tscor(count,1:ncls);

        [junk,winner] = min(temp);

        if (winner == clsinfo(count))

            tcor(i) = tcor(i) + 1;

        else

            tmislist(missed) = j;

            missed = missed + 1;

        end

        count = count + 1;

    end

end

for k = 1:ncls

    temp = (tcor(k)/clsize(k))*100;
```

```
end

% Now compute prediction results
pcor = 0;
missed = 0;
for i = 1:nrow_p
        for k = 1:ncls
                pscor(i,k) = (mvect(k,:)-pred(i,:))*c*(mvect(k,:)-pred(i,:))';
        end
        temp(1:ncls) = pscor(i,1:ncls);
        [junk,winner] = min(temp);
end
temp = sum(tcor);
tpct = (temp/nrow_t)*100;

% Normalize the noise array to zero by subtracting the smallest value from all points.
clear temp;
temp = zeros(nrow_p,ncol_t);
[min_value, min_index] = min(pscor, [], 2);
for i = 1:nrow_p
        for k = 1:ncls
        temp(i,k) = pscor(i,k) - min_value(i,1);        % shows the min m.d. more clearly
by making the min zero
```

```
        end
    end

    letter = min_index;
    min_mahal = temp;

    %Produce a grayscale colormap with negated distances so that white
    %represents 0 and black represents -1.
    temp = temp';
    figure, colormap('gray');
    imagesc(-temp);
    axis([0.5 (nrow_p + 0.5) 0.5 (ncls + 0.5)]);
    get(gca,'YTick');
    set(gca,'YTick',[1:1:ncls]);
    set(gca,'YTickLabel',{'A','B','C','D','E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T','U','
    V','W','X','Y','Z'});
    get(gca,'XTick');
    set(gca,'XTickLabel',{'A','B','C','D','E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T','U','
    V','W','X','Y','Z'});
    title('Confusion matrix for classification based on minimum Mahalanobis distance');
```

A.9 ASSIGNLETTERS.M

```
% ASSIGNLETTERS script prompts for selection of an Excel string to be
% assigned to the feature matrix for that subject.
% Perform analysis on one string (all replications) at a time.

% context(nreps, nvar, nletters) = temporary holding block
% mean_context = zeros(5,18,40,2);
% std_context = zeros(5,18,40,2);
% cuts = zeros(10, 2, 40, 2);

% Enter the subject in question
subject = input('Which subject(2 or 4)? ');
% 1 or 2 which correspond to subjects 2 or 4 respectively
if (subject == 2)
   sub = 1;
elseif (subject == 4)
   sub = 2;
else
   error('Only subjects 2 and 4 are being analyzed')
end

% Enter the string number in question
strnum = input('What string number(0:39)? ');                % choice of 40 strings
```

```
if (0 > strnum > 39)
    error('String number must be between 1 and 40.')
end
strnum = strnum + 1;

% Enter the number of replications of that string
rep = input('How many replications for this string? ');
if (1 > rep > 10)
    error('Maximum of 10 replications per string.')
end

context = zeros(rep, nvar, nletters);
cutsC(:, :, strnum, sub) = tempcut(:, :);

for j = 1:rep
        [filename, pathname] = uigetfile('*.xls', ['Select EXCEL string ', num2str(strnum), ', 
replication ', num2str(j)]);
        string = xlsread([pathname,filename]);
        string = string(25:300,3:20);
        fstring = filterData(string);                      % filter the string
        index = cutsC(j,1,strnum,sub):cutsC(j,2,strnum,sub);          % index the filtered data
        [indd, letter] = classify_minima(fstring(index,:),subject);    % indd is returned as an 
index into the cut string
```

```
        for i = 1:3
            indd(i) = index(1) + indd(i);
        end
        minima = [index(1) indd(1) indd(2) indd(3) index(length(index))];
        for k = 1:nletters
                context(j,:,k) = fstring(minima(k),:);
        end
end
%count = 1;
C_observations(count:(count + rep - 1),:) = context(:,:,4);
count = count + rep;

% Add the static observations to the feature matrix
% C_observations1(197:203,:) = staticC(1:7,1:18);
% C_observations(178:187,:) = staticC(1:10,1:18,2);
```

A.10 MULNORTEST.M

```
function [n, p, bl, t, P] = Mulnortest(X,alpha)

%   Multivariate Normality Statistical Test.

%   Modified by: Gillian Sherry, April 2004

%   Inputs:

%       X - multivariate data matrix.

%       alpha - significance level (default = 0.05).

%   Output:

%       n - sample-size.

%       p - variables.

%       b1 - estimated sample slope.

%       t - observed Student's t statistic used to test any deviation from

%          a expected slope of 1.0 (b1 = 1.0), which means a deviation of the

%          multivariate normality.

%       P - probability that null Ho: is true.

%       Figure - plot of the ordered Mahalanobis distances along with the

%             corresponding chi-square values, as well as the expected

%             straight line.

%

% On the literature there are available several tests of the multivariate normality.

% Among them is the graphic approach based on a chi-square quantile-quantile plot of

% the observations' squared Mahalanobis distances. Besides the graphic q-q approaching,

% in this file we proposes an alternative statistical test to this.
```

% Created by A. Trujillo-Ortiz and R. Hernandez-Walls

% Facultad de Ciencias Marinas

% Universidad Autonoma de Baja California

% Apdo. Postal 453

% Ensenada, Baja California

% Mexico.

% atrujo@uabc.mx

% And the special collaboration of the post-graduate students of the 2002:2

% Multivariate Statistics Course: Karel Castro-Morales, Alejandro Espinoza-Tenorio,

% Andrea Guia-Ramirez, Raquel Muniz-Salazar, Jose Luis Sanchez-Osorio and

% Roberto Carmona-Pina.

%

% November 2002.

% $Updated: June 10, 2003$

%

% References:

%

% Johnson, R. A. and Wichern, D. W. (1992), Applied Multivariate Statistical Analysis.

% 3rd. ed. New-Jersey:Prentice Hall. pp. 158-160.

% Stevens, J. (1992), Applied Multivariate Statistics for Social Sciences. 2nd. ed.

% New-Jersey:Lawrance Erlbaum Associates Publishers. pp. 247-248.

```
if nargin < 2,
   alpha = 0.05;  %(default)
end;

if nargin < 1,
   error('Requires at least one input arguments.');
end;

mX = mean(X); %Means vector from data matrix X.
[n,p] = size(X);
difT = [];

for j = 1:p;
   eval(['difT=[difT,(X(:,j)-mean(X(:,j)))];']);
end;

S = cov(X);
D2T = difT*inv(S)*difT';
D2 = sort(diag(D2T));  %Ascending squared Mahalanobis distances.

Pr = [];
for i = 1:n;
   eval(['pr' num2str(i) '=(i-0.5)/n;'])
```

```
    eval(['x= pr' num2str(i) ';'])
    Pr = [Pr,x];  %Corresponding sampling percentiles.
end;

X2 = [];
for i = 1:n
   eval(['X2' num2str(i) '=chi2inv(pr' num2str(i) ',p);'])
   eval(['x= X2' num2str(i) ';']);
   X2=[X2,x];  %Expected chi-square distribution with p degrees of freedom, associated
               %to the sampling percentiles.
end;

X = D2;
Y = X2';
%Test of the straight line by the least squares fitting method.
X = [ones(size(X)) X];
b = inv(X'*X)*(X'*Y);
b1 = b(2,1); %Unbiased estimation of slope.
Ye = X*b; %Expected Y values.
e = Y-Ye; %Estimation of the fitted residuals.
SCRes = e'*e;  %Sum of squares of the fitted residuals.
[rb,cb] = size(b);
v2 = n-rb;  %Degrees of freedom of the fitted residuals.
```

```
CMRes = SCRes/v2;  %Residuals mean square (random variance).

varb = CMRes*inv(X'*X);

EEb = diag(sqrt(varb));

EEb1 = EEb(2,1);  %Slope standard error.

t = (b1-1)/EEb1;  %Observed Student's t statistic assuming a slope expected value of 1.0.

P = 1-tcdf(t,v2);  %Probability that null Ho: is true, P = tcdf(X,V) computes Student's t
cdf at each of the values in X using the corresponding degrees of freedom in V.


fprintf('-----------------------------------------------------\n');

disp(' Sample-size    Variables    Slope       t       P')

fprintf('-----------------------------------------------------\n');

fprintf('%8.i%13.i%14.4f%10.4f%8.4f\n',n,p,b1,t,P);

fprintf('-----------------------------------------------------\n');

fprintf('With a given significance level of: %.2f\n', alpha);


if P >= alpha;

  fprintf('Assumption of multivariate normality is tenable.\n\n');

else

  fprintf('Assumption of multivariate normality is not tenable.\n\n');

end;


X = X(:,2);

plot(X,Y,'*',Y,Y,'--');
```

```
title('Multivariate Normality Test','FontSize',12);
xlabel('Mahalanobis Distance  D^2');
ylabel('Chi-square  \chi^2');
text(5.2,1.5,['Slope = ',num2str(b1),', ','n = ',num2str(n),', ','p = ',num2str(p), ', ' , 'P =
',num2str(P)]);
clear all;
```

APPENDIX B

SEGMENTATION POINTS

This appendix contains time points at which the strings were segmented for the analysis.

These corresponded to the local minima of the letters on either side of the ISC trigram.

B.1 SUBJECT 1

Subject 1	Cuts	Classified as
ISC- Different initial letter non-words		
16 GD**TISCE**XI – 7 4 **20 9 19 3 5** 24 9		
GDTISCEXI1	66 150	7 4 **20 9 19 3 5** 24 9
GDTISCEXI2	62 136	7 4 **20 25 19 3 5** 24 9
GDTISCEXI3	86 167	15 7 4 **20 10 19 3 5** 24 10
GDTISCEXI4	59 128	7 4 **20 9 19 3 5** 24
GDTISCEXI5	68 137	7 4 **20 25 19 3 5** 24
GDTISCEXI6	64 130	7 4 **20 5 19 3 5** 15
GDTISCEXI7	93 170	3 7 4 **20 10 19 3 5**
GDTISCEXI8	85 168	7 4 20 **9 19 3 3 5**
GDTISCEXI9	67 159	7 4 **20 10 19 3 5**
GDTISCEXI10	64 131	7 4 **20 25 19 3 5** 24 9

Misclassified...	as...	#times/10
9	25, 10, 5	3, 3,1

18 HVZ**SISCO**WJ – 8 22 26 **19 9 19 3 15** 23 10		
HVZSISCOWJ1		5 5 21 22 22 12 11 19
HVZSISCOWJ2	89 167	21 22 26 **19 15 19 3 15** 23 10
HVZSISCOWJ3	83 148	21 22 26 **19 15 19 3 19** 23
HVZSISCOWJ4	82 145	21 22 26 **19 15 19 3 15** 23
HVZSISCOWJ5	87 153	21 22 26 **19 10 19 3 19...**
HVZSISCOWJ6	88 155	21 22 26 **19 15 19 3 15** 23 15 10
HVZSISCOWJ7	93 161	21 22 26 19 **15 15 19 15 15** 23 10
HVZSISCOWJ8	114 190	21 26 26 19 **19 15 19 15 15** 23 10
HVZSISCOWJ9	138 212	7 21 22 26 26 **19 10 19 3 15** 23 10
HVZSISCOWJ10	92 162	21 22 26 **19 15 19 3 15** 23 10 10 15

Misclassified...	as...	#times/10
9	15, 10	8,2
3	15	1

26 PCL**MISCU**DF – 16 3 12 **13 9 19 3 21** 4 6

PCLMISCUDF1	90	156	16 3 12 **13 9 19 3 21** 4 6 6 6
PCLMISCUDF2	84	151	16 3 12 **13 10 19 3 21** 4 3 6 6
PCLMISCUDF3	96	173	16 3 12 **15 10 19 3 21** 21 4
PCLMISCUDF4	92	169	16 3 12 **7 10 19 3 21** 4 6
PCLMISCUDF5	84	185	2 3 12 **15 10 19 3 21** 4 4 15
PCLMISCUDF6	87	166	11 3 12 **13 10 19 3 21** 4 6 6
PCLMISCUDF7	103	197	3 11 3 12 **15 10 19 3 21** 4 6
PCLMISCUDF8	95	171	3 2 3 12 **15 10 19 3 21** 4 6
PCLMISCUDF9	88	165	15 16 3 12 **15 10 19 3 21** 21 6 3
PCLMISCUDF10	80	164	16 3 12 **13 10 19 3 21** 4 6 6

Misclassified...	as...	#times/10
9	10	9
13	15, 7	5, 1
16	2, 11	1, 2

34 QWK**XISCA**VB – 17 23 11 **24 9 19 3 1** 22 2

QWKXISCAVB1	99	164	7 17 22 11 **24 9 19 3 1** 22 2 2
QWKXISCAVB2	112	188	17 23 11 **24 9 19 3 1** 22 2 2
QWKXISCAVB3	98	190	17 23 11 **24 9 19 3 1** 22 2 2 2
QWKXISCAVB4	105	188	7 17 23 11 **24 9 19 3 1** 22 2 2
QWKXISCAVB5	116	183	17 22 23 11 **24 9 19 3 1** 22 2 7
QWKXISCAVB6	114	196	17 23 11 **24 9 19 3 1** 22 2 2
QWKXISCAVB7	154	228	17 23 11 24 **10 19 19 3 1** 22 2
QWKXISCAVB8	115	190	3 17 23 11 **4 10 19 3 1** 22 2 2
QWKXISCAVB9	97	170	15 17 23 11 **24 9 19 3 1** 22 2 2
QWKXISCAVB10	85	150	17 23 11 **24 9 19 3 1** 2 22 2 2

Misclassified...	as...	#times/10
22	23	1/2
24	4	1

39 XBT**KISCI**GR – 24 2 20 **11 9 19 3 9** 7 18

XBTKISCIGR1	97	180	22 24 2 20 **11 9 19 3 9** 7 18 18
XBTKISCIGR2	95	172	4 2 20 **11 10 19 3 10** 7 1 18 18
XBTKISCIGR3	107	189	3 24 2 20 **11 10 19 3 9** 7 19 18
XBTKISCIGR4	84	153	24 24 2 20 **11 10 19 3 9** 3 7 15 18 23
XBTKISCIGR5	98	157	15 4 2 20 **11 10 19 3 10** 7 15 18
XBTKISCIGR6	75	140	12 24 2 20 **11 10 19 3 10** 7 19 18
XBTKISCIGR7	83	150	15 24 2 20 **11 10 19 3 10** 7 18 18
XBTKISCIGR8	82	163	4 2 20 **11 10 19 3 10** 7 19 21 21
XBTKISCIGR9	89	148	12 24 2 20 **11 10 19 3 15** 7 1 18
XBTKISCIGR10	93	177	24 2 20 **11 10 19 3 10** 7 3 21 21

Misclassified...as... #times/10

24	4	3
9	10 – 10, 15	9 – 6, 1

ISC Different initial letter words

0 **BISCUIT** - **2 9 19 3 21** 9 20

BISCUIT1	109 170	3 9 10 10 **2 9 19 3 21** 20 20 20
BISCUIT2	48 119	**2 10 19 19 3 21** 10 20
BISCUIT3	11 91	**2 9 19 3 21** 10 20
BISCUIT4	53 147	3 **2 10 19 19 3 21** 10 20 7 7
BISCUIT5	41 114	2 **2 10 19 3 21** 21 10 20 7 7
BISCUIT6	21 87	**2 10 19 3 21** 10 20 7 7
BISCUIT7	19 102	**2 10 19 3 21** 10 20 7 7
BISCUIT8	34 113	2 **2 10 19 3 21** 10 20 3 2 3
BISCUIT9	28 112	**2 10 19 3 21** 25 20 7 7
BISCUIT10	20 79	**2 10 19 3 21** 10 20 7

Misclassified...as...		#times/10
9	10 – 10, 25	8 - 8,1

6 CON**FISCATE** – 3 15 14 **6 9 19 3 1** 20 5

CONFISCATE1	86 162	3 15 14 **6 9 19 3 1** 20 5 5 5
CONFISCATE2	79 171	19 7 15 **15 19 1 3 1** 20 5 2 15
CONFISCATE3		3 15 7 10 10 15 19 15 15 15 9 19
CONFISCATE4	136 230	3 15 7 15 6 6 **9 19 1 3 1** 20 5
CONFISCATE5	109 178	15 9 **15 9 19 3 1** 20 5 15 15 3
CONFISCATE6	79 156	15 **6 9 19 3 1** 20 5 5 5 15 15
CONFISCATE7	92 163	15 7 **6 9 19 3 1** 20 5 15 15
CONFISCATE8	80 149	15 **6 9 19 3 1** 20 5 5 15 15
CONFISCATE9	77 173	15 **6 9 19 1 1** 1 20 5 15 15
CONFISCATE10	65 120	3 15 15 **15 9 19 3 1** 20 5 15 15

Misclassified...as...		#times/10
14	7, 15, 9	
3	1	
6	15	3
9	10, 15	

25 OM**NISCIENT** – 15 13 **14 9 19 3 9** 5 14 20

OMNISCIENT1	79 154	3 13 **21 9 19 3 9** 5 20 21 20 11
OMNISCIENT2	84 172	15 14 15 **8 10 19 3 10** 5 5 20 20
OMNISCIENT3	83 180	15 15 10 **10 10 19 3 15** 5 15 20 20
OMNISCIENT4	75 154	15 15 **8 10 19 3 15** 5 5 15 20
OMNISCIENT5	77 151	15 15 **14 10 19 3 10** 5 20 20 20
OMNISCIENT6	66 130	15 15 15 **8 10 19 3 15** 5 14 20
OMNISCIENT7	77 138	15 15 **15 15 19 3 15** 5 15 20 20 15
OMNISCIENT8	119 195	15 15 15 10 8 8 **10 19 19 3 15** 5 1 20

OMNISCIENT9	99	170	15 8 **10 15 19 15 10** 5 8 20 3 3
OMNISCIENT10	84	144	15 15 15 13 14 **10 3 19 3 10** 5 14 20

Misclassified...	as...	#times/10
14	8, 21 – 15, 8, 1	5,1 – 3, 1, 1
9	10 – 10, 15	8 – 4, 5
3	15	1

27 PE**RISCOP**E – 16 5 **18 9 19 3 15** 16 5

PERISCOPE1	48	125	16 19 **18 9 19 3 15** 16 5 5 5 5
PERISCOPE2	72	134	16 19 **18 10 19 3 15** 16 5 5 5 5
PERISCOPE3	47	109	2 19 **21 10 19 3 15** 16 5 3 3
PERISCOPE4	43	102	2 15 **18 10 19 3 15** 11 5 5 5 5
PERISCOPE5	50	113	16 19 **21 10 19 3 15** 16 5 5 15 15
PERISCOPE6	58	119	2 19 **21 10 19 3 15** 11 5 15 15 15
PERISCOPE7	47	110	11 19 **18 10 19 3 15** 11 5 3 3 15
PERISCOPE8	69	124	15 2 19 **18 10 19 3 15** 11 5 15 15
PERISCOPE9	47	130	16 19 **21 20 19 3 15** 16 5 5 3 3
PERISCOPE10	47	105	11 19 **21 10 19 3 15** 11 5 3 15 3

Misclassified...	as...	#times/10
5	19, 15	9, 1
9	10, 20	8, 1
16	2, 11 -11	4, 2 -5
18	21	5

38 **VISCER**AL – **22 9 19 3 5** 18 1 12

VISCERAL1	26	87	**22 9 19 3 19** 18 20 12 12 12 12 12
VISCERAL2	85	162	22 10 3 10 **22 10 19 3 19** 18 1 12 12
VISCERAL3	13	85	**22 10 19 3 19** 18 1 12 12 12 12 12
VISCERAL4	37	93	22 **22 9 19 3 19** 18 1 4 12 12 7
VISCERAL5	17	69	**22 10 19 3 19** 18 19 12 26 7 7
VISCERAL6	14	98	**22 10 19 3 19** 18 20 12 3 15 3 15
VISCERAL7	25	95	**22 10 19 3 19** 18 19 12 12 15 7 7
VISCERAL8	16	80	**22 10 19 3 19** 22 19 12 12 12 7
VISCERAL9	22	98	**22 9 19 3 19** 18 19 12 3 3 3 2
VISCERAL10	32	89	22 **9 19 19 3 19** 18 1 12 12 7

Misclassified...	as...	#times/10
5	15	10
1	19, 20	4, 2
9	10	6
18	22	1

ISC Same inital letter non-words

1 B**RISCA**NT – 2 **18 9 19 3 1** 14 20

BRISCANT1	37	110	2 **18 10 19 3 19** 19 21 14 20 20
BRISCANT2	57	137	5 2 **2 10 19 3 1** 20 20 20 20 20
BRISCANT3	38	106	2 **2 9 19 3 1** 20 20 20 7 7
BRISCANT4	40	118	2 **2 9 19 3 1** 20 20 20 20 7 7
BRISCANT5	34	105	2 **2 10 19 3 1** 7 20 15 15 15 15
BRISCANT6	29	81	2 **2 10 19 3 1** 20 20 20 20 20 20
BRISCANT7	55	111	2 2 **10 15 19 3 1** 20 20 20 1 1
BRISCANT8	41	108	2 **18 10 19 3 1** 15 20 11 20 7 7
BRISCANT9	111	168	3 2 2 10 **15 15 15 3 1** 20 20 7
BRISCANT10	34	110	2 **18 10 19 3 20** 1 14 20 20 20 7

Misclassified...	as...	#times/10
9	10	8
1	20, 19	1,1

2 B**RISCE**NSE – 2 **18 9 19 3 5** 14 19 5

BRISCENSE1	19	128	**2 10 19 3 5** 10 19 5 5 5 5
BRISCENSE2	52	125	2 3 **18 10 19 3 5** 8 19 5 5 5
BRISCENSE3	50	133	2 18 **10 19 19 3 5** 14 19 5 5 5
BRISCENSE4	34	108	2 **2 9 19 3 19** 14 19 5 15 15 15
BRISCENSE5	52	125	2 2 **18 10 19 3 5** 8 19 5 15 15
BRISCENSE6	31	90	2 **2 10 19 3 5** 8 19 5 15 15 15
BRISCENSE7	53	126	2 2 **18 10 19 3 5** 8 19 5 5 15
BRISCENSE8	30	95	2 **2 10 19 3 5** 15 5 5 5 15 15
BRISCENSE9	53	115	2 **2 10 19 3 5** 15 19 5 5 5
BRISCENSE10	40	103	3 2 **2 10 19 3 5** 14 5 5 5 15

Misclassified...	as...	#times/10
14	10,8, 15	1, 4, 2
9	10	9
5	19	1

3 B**RISCI**ARD – 2 **18 9 19 3 9** 1 18 4

BRISCIARD1	76	137	3 2 18 **10 3 19 3 10** 1 18 4 4
BRISCIARD2	43	110	2 **18 10 19 3 10** 1 18 4 4 4 4
BRISCIARD3	100	173	2 10 15 2 **18 10 19 3 10** 1 18 4
BRISCIARD4	42	114	2 **18 10 19 3 10** 1 18 4 15 4 3
BRISCIARD5	83	150	3 2 9 **18 10 19 3 15** 1 18 4 4
BRISCIARD6			2 18 10 19 1 15 19 3 19 3 15 1
BRISCIARD7	47	122	2 2 **18 10 19 3 15** 15 1 18 4 4
BRISCIARD8	51	118	3 3 2 **2 10 19 3 15** 1 18 4 4
BRISCIARD9	36	117	2 **2 10 19 3 10** 1 18 4
BRISCIARD10	32	100	2 **2 10 19 3 10** 1 18 4 4 4 2

Misclassified...as... #times/10

9 10 – 10, 15 10 - 6, 4

4 B**RISCOZ**E – 2 **18 9 19 3 15** 26 5

BRISCOZE1	95	154	2 25 3 2 **18 18 19 3 19** 26 5 5
BRISCOZE2	39	117	2 **18 10 19 3 15** 26 5 5 5 5 5
BRISCOZE3	32	106	2 **2 10 19 3 15** 26 5 5 5 15 15
BRISCOZE4	65	132	2 2 **18 10 19 3 15** 26 5 5 15 15
BRISCOZE5	38	116	2 **18 10 19 3 15** 26 5 5 5 15 15
BRISCOZE6	46	113	2 **18 10 19 3 15** 12 5 15 15 15
BRISCOZE7	32	104	2 **2 10 19 3 15** 26 5 5 5 15 15 15
BRISCOZE8	67	135	3 2 **2 10 19 3 15** 26 5 5 5 15 15
BRISCOZE9	32	103	2 **2 10 19 3 15** 26 5 15 15 15
BRISCOZE10	45	110	2 **2 10 19 3 15** 26 5 15 15 15 15

Misclassified...	as...	#times/10
15	19	1
9	10, 18	9, 1

5 B**RISCU**DGE – 2 **18 9 19 3 21** 4 7 5

BRISCUDGE1	45	113	2 **18 10 19 3 21** 4 7 5 5 5 5 5
BRISCUDGE2	31	99	2 **18 26 19 3 21** 21 21 4 5 7 5
BRISCUDGE3	57	141	2 **2 10 19 3 21** 21 4 4 7 5
BRISCUDGE4	33	105	2 **18 10 19 3 21** 4 5 7 5 5 5 15 15
BRISCUDGE5	36	98	2 **18 10 19 3 21** 4 5 7 5 15 15 15
BRISCUDGE6	29	93	2 **18 10 19 3 21** 4 7 5 5 5 5 5
BRISCUDGE7	35	109	2 **18 10 19 3 21** 4 7 5 15 15
BRISCUDGE8	46	110	2 **18 10 19 3 21** 21 4 4 7 5 5 5 15
BRISCUDGE9	46	128	2 **2 10 19 3 21** 12 4 15 7 5 15
BRISCUDGE10	33	106	2 **18 10 19 3 21** 21 4 7 5 5 15 15

Misclassified...	as...	#times/10
9	10, 26	9, 1

ISC Same initial letter words

8 **DISCARD – 4 9 19 3 1** 18 4

DISCARD1	45	110	3 4 **9 19 3 3 1** 18 4 4
DISCARD2	16	88	**4 15 19 3 1** 18 4 4
DISCARD3	66	143	18 4 **9 19 1 3 1** 18 4 3 4 3 15 15
DISCARD4	23	128	**4 9 19 3 1** 18 4 4 7 7
DISCARD5	17	95	**4 15 19 3 1** 18 4 7 7
DISCARD6	23	119	**4 15 19 15 1** 18 15 4 15 15 15
DISCARD7	35	126	15 **4 15 19 3 1** 18 4 3 4 2 3
DISCARD8	32	108	4 **4 15 19 3 1** 18 4 4 7
DISCARD9	32	102	3 **4 15 19 3 1** 18 4 4 3 3
DISCARD10	16	99	**4 15 19 3 1** 18 4 4

Misclassified...as...		#times/10
9	15	7
3	15	1

9 **DISCER**N – **4 9 19 3 5** 18 14

DISCERN1	13	69	**4 9 19 3 19** 18 14 14 14
DISCERN2	20	82	**4 15 19 3 19** 22 14 14
DISCERN3	20	87	3 **4 15 19 3 19** 22 14 14 14
DISCERN4	29	93	4 **4 10 19 3 19** 18 8 8 15 15
DISCERN5	15	84	**4 15 19 3 19** 18 14 14 7 7 7
DISCERN6	21	83	4 **4 15 19 3 19** 18 8 8 8 15 15
DISCERN7	32	110	3 **4 15 19 3 19** 18 10 10 8 8 8
DISCERN8	24	96	**4 15 19 3 19** 18 8 8 8 8 15
DISCERN9	22	100	**4 15 19 3 19** 5 18 8 8 15 15
DISCERN10	28	96	4 **4 15 19 3 19** 22 14 14 14 8

Misclassified...as...		#times/10
5	19	10
9	15, 10	8, 1
18	22	3

10 **DISCIP**LE – **4 9 19 3 9** 16 12 5

DISCIPLE1	21	86	**4 10 19 3 9** 16 12 5 5 5 5
DISCIPLE2	16	103	**4 15 19 3 15** 11 12 5 5 5
DISCIPLE3	44	105	4 4 **10 15 19 3 15** 16 12 5 15
DISCIPLE4	30	114	3 **4 9 19 3 10** 16 12 5 1 1 1
DISCIPLE5	25	87	**4 15 19 3 15** 16 5 15 15 15
DISCIPLE6	19	81	3 **4 15 19 3 10** 16 12 5 15 15
DISCIPLE7	31	96	4 **4 15 19 3 15** 11 12 5 15 3 15
DISCIPLE8	15	74	**4 15 19 3 15** 11 12 5 15 15 15
DISCIPLE9	21	85	3 **4 15 19 15 15** 16 12 5 3 3 3 15
DISCIPLE10	21	80	**4 15 19 3 15** 11 12 5 5 5 5 5

Misclassified...as...		#times/10
9	10, 15 – 10, 15	2, 7 – 2, 7
3	15	1
16	11	4

11 **DISCO**VER – **4 9 19 3 15** 22 5 18

DISCOVER1	20	93	4 **4 9 19 3 22** 20 18 18 18 18 18
DISCOVER2	15	65	**4 9 19 3 19** 22 19 22 22 22
DISCOVER3	24	88	**4 9 19 3 3** 22 22 19 22 18 18 18 22 7
DISCOVER4	26	100	3 **4 15 19 3 15** 22 19 18 18 18 18 7
DISCOVER5	41	95	7 **4 9 19 3 19** 22 19 22 22 7 7 7 7 7 7
DISCOVER6	22	98	**4 15 19 3 22** 19 18 18 7 7 7 7 7 7

DISCOVER7	77	161	4 4 15 **4 25 19 15 15** 22 19 18 3 15
DISCOVER8	17	88	**4 15 19 3 15** 22 19 18 15 15 15
DISCOVER9	20	92	**4 15 19 3 15** 22 19 18 18 11 7 7 7
DISCOVER10	21	96	4 **4 15 19 3 15** 22 19 18 20 7 7 7

Misclassified...	as...	#times/10
9	15, 25	5, 1
3	15	1
15	19	2

12 **DISCUSS** – **4 9 19 3 21** 19 19

DISCUSS1	57	110	9 **19 19 19 3 19** 21 21 19 19 19 3 15
DISCUSS2	66	136	4 19 19 **19 19 19 3 21** 19 19 19 19 15
DISCUSS3	37	103	3 **9 15 19 3 21** 21 19 19 19 19 19 19
DISCUSS4	68	152	4 9 **3 19 19 3 21** 19 19 19 19
DISCUSS5	12	79	**4 15 19 3 21** 19 19 15 15 15 15 15
DISCUSS6	43	136	15 4 **15 19 18 3 21** 19 19 19 19 19
DISCUSS7	23	101	**4 10 19 3 21** 19 15 19 19 19 19
DISCUSS8	17	108	**4 15 19 3 21** 19 19 19 19 19 19
DISCUSS9	47	127	3 4 **10 19 19 3 21** 19 19 19 19 19
DISCUSS10	19	92	4 **4 15 19 15 21** 19 19 19 19 15 15

Misclassified...	as...	#times/10
9	15, 10	4, 2
3	15	1

B.2 SUBJECT 2

Subject 2	Cuts		Classified as
ISC- Different initial letter non-words			
16 GD**TISCE**XI – 7 4 **20 9 19 3 5** 24 9			
GDTISCEXI1	73	127	7 4 **1 9 19 3 5** 24 9 9 9 9 9
GDTISCEXI2	68	130	7 4 **7 9 19 2 5** 24 9 9 9 9 9
GDTISCEXI3	64	120	15 7 4 **7 9 19 3 5** 24 9 9 9 99
GDTISCEXI4	67	116	7 7 4 7 **5 9 19 2 5** 24 9 9 9 99
GDTISCEXI5	67	129	7 4 **7 9 19 2 5** 24 9 9 9 9 9 9 9
GDTISCEXI6	65	135	7 4 4 **7 9 19 3 5** 24 9 9 9 9 9
GDTISCEXI7	57	107	7 15 4 **26 9 19 3 5** 24 9 9 9 9
GDTISCEXI8	48	110	7 4 **7 9 19 3 5** 24 9 9 9 9 9 99
GDTISCEXI9	55	104	7 4 7 **24 9 19 3 5** 24 9 9 9 9 9
GDTISCEXI10	60	125	7 15 4 **7 9 19 2 5** 24 9 9 9 9 9

Misclassified...	as...	# times/10
3	2	4

20	1, 7, 26	1, 8, 1

18 HVZ**SISCO**WJ – 8 22 26 **19 9 19 3 15** 23 10

HVZSISCOWJ1	106	153	21 8 21 26 26 **19 9 19>>15** 23 *9 9* 10
HVZSISCOWJ2	93	141	7 8 21 26 **19 9 19 3 15** 23 *9 9* 10 10
HVZSISCOWJ3	110	171	8 22 26 26 **19 9 19 3 15** 23 *9 9* 10 10
HVZSISCOWJ4	133	181	8 22 26 26 26 26 **19 9 19 3 15** 23 *9 9* 10
HVZSISCOWJ5	116	176	14 8 22 26 26 **19 25 19 3 15** 23 9 9 10
HVZSISCOWJ6	122	176	6 8 21 26 26 **19 9 19 3 15** 23 *9 9* 10
HVZSISCOWJ7	104	154	15 8 21 26 **19 25 19 3 15** 23 9 9 10
HVZSISCOWJ8	119	179	8 14 22 26 26 15 **19 9 19 3 15** 23 *9 9* 10
HVZSISCOWJ9	109	152	8 8 8 22 26 **19 9 19 3 15** 23 *9 9* 10 10
HVZSISCOWJ10	133	198	15 8 21 22 26 26 **19 25 19 2 15** 23 9 10

Misclassified…as…		# times/10
22	21	4
9	25	2

26 PCL**MISCU**DF – 16 3 12 **13 9 19 3 21** 4 6

PCLMISCUDF1	68	125	11 11 3 12 **13 9 19 3 21** 4 6 6 6 6 6 6
PCLMISCUDF2	131	209	5 16 3 12 13 9 **5 19 3 3 21** 4 6
PCLMISCUDF3			7 16 3 12 **13 9 19 >> 21** 4 6 6 6 6 6
PCLMISCUDF4	69	133	16 16 3 12 **13 9 19 3 3** 4 6 6 6 6 6 6
PCLMISCUDF5	67	119	16 3 12 **13 25 19 2 21** 4 6 6 6 6 6 6 6
PCLMISCUDF6	69	136	16 3 12 **13 9 19 3 15** 4 6 6 6 6 6 6
PCLMISCUDF7	96	160	15 16 3 12 13 **21 9 19 15 15** 4 6 6 6 6
PCLMISCUDF8	107	158	3 16 12 3 12 **13 25 19 3 21** 4 6 6 6 6
PCLMISCUDF9	123	172	15 16 3 12 13 13 **13 25 19 3 15** 4 6 6
PCLMISCUDF10	71	127	16 3 12 **13 25 19 3 15** 4 6 6 6 6 6 6 6

Misclassified…as…		# times/10
21	15, 3	4, 1
3	2	1
16	11	1
9	25	4

34 QWK**XISCA**VB – 17 23 11 **24 9 19 3 1** 22 2

QWKXISCAVB1	115	177	17 23 11 26 **24 9 19 2 1** 22 2 2 2 2
QWKXISCAVB2	107	165	7 17 15 23 11 4 **24 9 19 3 1** 22 2 2 2
QWKXISCAVB3	86	143	7 17 23 11 **24 9 19 3 1** 22 2 2 2 2 2
QWKXISCAVB4	91	154	17 23 11 **24 9 19 2 1** 22 2 2 2 2 2 2
QWKXISCAVB5	118	170	17 23 11 11 24 24 **26 9 19 3 1** 22 2
QWKXISCAVB6	75	139	17 23 11 **24 9 19 3 1** 22 2 2 2 2 2
QWKXISCAVB7	94	161	17 15 23 11 **24 9 19 3 1** 22 2 2 2 2
QWKXISCAVB8	79	147	17 15 23 11 **24 9 19 3 1** 22 21 2 2 2

QWKXISCAVB9	97	150	17 23 11 24 **24 9 19 3 1** 22 2 2 2 2 2
QWKXISCAVB10	81	136	15 17 22 11 **24 9 19 3 1** 22 2 2 2 2 2

Misclassified...	as...	# times/10
3	2	2
23	22	1

39 XBT**KISCI**GR – 24 2 20 **11 9 19 3 9** 7 18

XBTKISCIGR1	74	135	24 2 1 **11 25 19 3 9** 7 18 18 18 18 18
XBTKISCIGR2	81	155	24 2 20 **11 9 19 3 9** 7 18 18 18
XBTKISCIGR3	107	171	5 24 2 7 14 **11 9 19 2 25** 7 18 18 18
XBTKISCIGR4	89	165	24 2 20 **11 9 19 3 9** 7 22 18 18
XBTKISCIGR5	80	137	5 24 2 20 **11 9 19 3 25** 7 18 18 18 18
XBTKISCIGR6	109	165	5 24 2 1 11 **9 9 19 3 9** 7 18 18 18
XBTKISCIGR7	70	132	24 2 20 **11 9 19 15 9** 7 18 18 18 18
XBTKISCIGR8	103	174	15 24 2 20 11 **11 9 19 3 9** 7 18 18
XBTKISCIGR9			4 24 2 20 **11 25 19 >> 25** 12 18 18
XBTKISCIGR10	67	118	24 2 1 **11 25 19 3 25** 7 18 18 18 18

Misclassified...	as...	# times/10
9	25	3-4
20	1	3

ISC Different initial letter words

0 **BISCUIT** - **2 9 19 3 21** 9 20

BISCUIT1	16	78	**2 9 19 2 21** 2 26 20 20 20
BISCUIT2	12	83	**2 9 19 2 21** 2 20 20 20 20
BISCUIT3	15	91	**2 9 19 2 8** 2 1 11 11 11 11
BISCUIT4	21	94	**2 25 19 2 8** 2 20 26 26 26
BISCUIT5	24	91	2 **2 2 19 2 21** 2 20 20 20
BISCUIT6	17	82	**2 9 19 2 21** 2 11 11 11 11
BISCUIT7	11	98	**2 9 19 3 21** 2 26 26 26 26
BISCUIT8	20	91	**2 2 19 3 21** 2 11 20 20 20
BISCUIT9	15	76	**2 25 19 2 21** 2 11 20 20
BISCUIT10	34	92	2 **2 15 19 2 8** 2 11 20 20

Misclassified...	as...	# times/10
3	2	8
9	2, 25, 15	10, 2, 1
20	11, 26	5, 3

6 CON**FISCATE** – 3 15 14 **6 9 19 3 1** 20 5

CONFISCATE1	82	132	3 15 8 6 **19 9 19 2 1** 20 5 5
CONFISCATE2	67	121	3 15 14 **6 9 19 3 1** 11 5 5 5
CONFISCATE3	73	143	3 15 14 6 **6 9 19 3 26** 16 5 5

CONFISCATE4	71	148	3 15 14 **6 9 19 2 1** 7 5 5 5 5
CONFISCATE5	76	130	3 15 14 6 **6 25 19 2 1** 7 15 5
CONFISCATE6	86	144	3 15 14 6 **6 9 19 1 2** 11 5 5
CONFISCATE7	79	134	3 15 14 6 **9 9 19 3 1** 1 16 5 5
CONFISCATE8	66	122	3 1 8 **6 25 19 3 1** 1 5 5 5
CONFISCATE9	87	167	6 3 3 15 14 **6 9 19 3 26** 5 5 5
CONFISCATE10	59	107	3 15 7 **6 25 19 3 1** 1 5 5 5 5

Misclassified...	as...	# times/10
14	8	2
3	2	4
1	26	2
9	25	3
20	7, 16,11	2,2,2

25 OMN**ISCI**ENT – 15 13 **14 9 19 3 9** 5 14 20

OMNISCIENT1	40	104	15 13 **7 9 19 2 25** 5 20 20
OMNISCIENT2	55	111	15 13 **8 25 19 3 25** 5 14 1 20
OMNISCIENT3	61	115	15 13 13 **14 9 19 2 25** 5 15 11 20
OMNISCIENT4	49	111	15 13 **13** 19 **9 19 25** 5 1 11 20
OMNISCIENT5	86	144	3 15 13 **7 25 19 25** 5 1 1 1 1 1
OMNISCIENT6	68	110	3 15 13 **7 9 19 25** 5 1 11 1 1 1
OMNISCIENT7	54	137	15 13 13 13 9 19 19 2 8 11 20
OMNISCIENT8	74	142	15 15 13 7 **19 9 19 3 25** 5 14 11 1 1 1
OMNISCIENT9	89	142	15 15 13 13 14 **9 19 19 3 25** 5 1 20
OMNISCIENT10	56	117	15 13 **8 13 19 15 25** 5 1 11 20

Misclassified...	as...	# times/10
14	7, 8, 13 - 1,11?	
3	2	2
9	13, 25 – 25	2, 2 - 9

27 PE**RISCO**PE – 16 5 **18 9 19 3 15** 16 5

PERISCOPE1	60	119	11 5 **18 13 19 3 15** 16 16 5 5
PERISCOPE2	51	112	26 4 24 **18 9 19 3 15** 11 16 5
PERISCOPE3	48	105	16 5 **18 9 19 3 15** 11 5 5 5
PERISCOPE4	52	117	16 5 **18 9 19 3 15** 11 5 5 5
PERISCOPE5	42	96	16 5 **18 9 19 2 15** 11 5 5 5
PERISCOPE6	72	138	15 16 5 **18 9 19 2 7** 7 5 5 5
PERISCOPE7	59	121	16 5 **18 9 19 3 15** 16 5 5 5
PERISCOPE8	60	136	16 5 **18 9 19 3 15** 11 5 5 5
PERISCOPE9	35	96	11 5 **18 13 19 3 15** 11 5 5
PERISCOPE10	43	94	11 5 **18 13 19 3 15** 16 5 5

Misclassified…	as…	# times/10
16	11	3, 6
9	13	3

38 **VISCERAL – 22 9 19 3 5** 18 1 12

VISCERAL1	10	66	**22 9 19 2 5** 21 1 12 12 12
VISCERAL2	36	89	22 **9 9 19 2 7** 18 1 12 12 12
VISCERAL3	17	73	**22 9 19 2 5** 18 1 12 12 12
VISCERAL4	12	66	**22 9 19 2 5** 21 1 12 12 12
VISCERAL5	12	63	**22 25 19 2 5** 13 1 12 12
VISCERAL6	21	101	**22 9 19 2 5** 21 1 12 12 12
VISCERAL7	18	75	**22 9 19 2 5** 18 1 12 12 12
VISCERAL8	14	74	**22 9 19 3 5** 21 1 12 12 12
VISCERAL9	13	60	**22 9 >> 3 5** 18 1 12 12 12
VISCERAL10	15	77	**22 25 19 2 5** 18 1 12 12

Misclassified…as…		# times/10
18	21, 13	4, 1
5	7	1
3	2	8
9	25	2

ISC Same inital letter non-words

1 B**RISCANT** – 2 **18 9 19 3 1** 14 20

BRISCANT1	24	76	2 **18 25 19 3 1** 1 20 20 20
BRISCANT2	36	94	2 **18 9 19 3 1** 14 1 1 1 7 7
BRISCANT3	46	113	2 18 **18 25 19 3 7** 1 1 1 1
BRISCANT4	26	86	2 **18 13 19 3 1** 8 20 20 20
BRISCANT5	58	112	2 18 **19 9 19 2 1** 14 1 1 11
BRISCANT6	27	101	2 **18 9 19 2 1** 11 20 20 20
BRISCANT7	44	119	2 **18 9 19 3 1** 1 1 11 1 1 1
BRISCANT8	35	93	2 **18 13 19 2 1** 14 11 1 1 1
BRISCANT9	46	100	2 **18 13 19 2 1** 8 1 1 1 1 1
BRISCANT10	30	77	2 **18 13 19 3 1** 8 1 1 1 1 1

Misclassified…as…		# times/10
9	25, 13	2, 4
20	1	7-8
14	7, 8, 11	1-2, 3, 3

2 B**RISCENSE** – 2 **18 9 19 3 5** 14 19 5

BRISCENSE1	37	109	2 **18 2 19 2 5** 8 19 5 5 5
BRISCENSE2	31	87	2 **18 9 19 2 5** 8 19 5 5 5
BRISCENSE3	83	131	2 18 25 19 **19 19 19 3 5** 8 19 5
BRISCENSE4	28	79	2 **18 9 19 >> 5** 8 19 5 5 5 5
BRISCENSE5	29	80	2 **18 2 19 3 5** 1 19 5 5 5

BRISCENSE6	43	105	2 **18 9 19 3 5** 1 19 5 5 5
BRISCENSE7	46	95	18 18 **18 9 19 >> 5** 15 19 5 5
BRISCENSE8	35	83	2 **18 25 19 >> 5** 15 19 15 5 5
BRISCENSE9	24	71	2 **8 13 19 2 5** 1 19 5 5 5 5
BRISCENSE10	20	65	2 **18 13 19 3 5** 1 19 5 5 5

Misclassified...	as...	# times/10
3	2	3
14	8, 15, 1	4, 2, 4
18	8	1
9	2, 13	2, 2

3 **BRISCI**ARD – 2 **18 9 19 3 9** 1 18 4

BRISCIARD1	38	104	2 **18 9 19 3 9** 1 18 4 4 4 4
BRISCIARD2	69	123	18 2 18 **2 19 19 3 25** 1 18 4
BRISCIARD3	31	112	2 **18 9 19 3 25** 1 8 4 4 4 4
BRISCIARD4	32	100	2 **18 13 19 3 25** 1 18 4 4 4
BRISCIARD5	33	~~79~~80	4 2 **18 13 19 2 25** 1 14 4 4
BRISCIARD6	68	133	3 2 18 18 **18 9 19 3 9** 1 18 4
BRISCIARD7	43	118	2 **18 9 19 3 9** 1 14 3 4 4 4
BRISCIARD8	31	92	2 **18 13 19 15 9** 1 18 4 4 4
BRISCIARD9	26	76	2 **18 13 19 3 25** 18 21 4 4 4
BRISCIARD10	22	75	2 **18 13 19 3 9** 1 21 4 4 4

Misclassified...	as...	# times/10
9	13 – 25	5 - 5
3	2, 15	1 , 1

4 **BRISCO**ZE – 2 **18 9 19 3 15** 26 5

BRISCOZE1	29	90	2 **18 25 19 3 15** 26 15 5 5 5
BRISCOZE2	43	94	2 **18 25 19 3 15** 26 26 26 *15* 5
BRISCOZE3	29	82	2 **18 25 19 3 15** 7 26 17 *15* 5
BRISCOZE4	59	127	2 **18 2 19 3 15** 26 26 17 26 *15* 5 5 5 5
BRISCOZE5	**39**	**104**	2 18 9 19 2 15 **26 26 26 *15* 5**
BRISCOZE6	**31**	**87**	**2** 18 9 19 3 15 **26 26 26 *15* 5**
BRISCOZE7	36	99	2 **18 9 19 3 15** 26 26 26 *15* 5
BRISCOZE8	33	86	2 **18 25 19 3 15** 26 26 26 *15* 5
BRISCOZE9	37	92	2 **18 13 19 3 15** 26 26 26 *15* 5
BRISCOZE10	31	76	2 **18 13 19 2 15** 26 15 5 5 5

Misclassified...	as...	# times/10
9	25, 2, 13	3, 1, 2

5 **BRISCU**DGE – 2 **18 9 19 3 21** 4 7 5

BRISCUDGE1	44	106	3 2 **18 9 19 3 21** 4 7 5 5 5
BRISCUDGE2	33	90	2 **18 9 19 3 21** 4 7 13 5 5
BRISCUDGE3	34	89	2 **18 13 19 3 21** 4 7 *15* 5 5
BRISCUDGE4	30	97	2 **18 13 19 3 21** 4 7 *15* 5 5
BRISCUDGE5	46	106	2 18 **9 19 3 3 21** 4 7 *15* 5 5
BRISCUDGE6	57	138	2 **18 9 19 3 21** 4 7 *15* 5 5 5
BRISCUDGE7	42	99	2 **18 9 19 3 21** 4 7 5 5 5 5
BRISCUDGE8	88	151	3 2 18 **9 19 19 3 21** 4 15 7 15 5
BRISCUDGE9	27	80	2 **18 13 19 3 15** 4 7 5 5 5
BRISCUDGE10	26	88	2 **18 13 19 3 21** 4 7 5 5 5

Misclassified...as...		# times/10
9	13	4
21	15	1

ISC Same initial letter words

8 **DISCARD – 4 9 19 3 1** 18 4

DISCARD1	24	98	**4 9 19 3 1** 14 21 4 4 4
DISCARD2	17	75	**4 9 19 2 1** 21 4 4 4 4 4
DISCARD3	14	66	**4 9 19 2 1** 18 4 4 4 4 4
DISCARD4	17	74	**4 9 19 3 1** 18 22 4 4 4
DISCARD5	22	77	4 **24 9 19 2 1** 21 3 4 4
DISCARD6	28	102	4 **4 9 19 3 1** 21 4 4 4 4
DISCARD7	26	99	**4 9 19 2 1** 21 21 4 4 4
DISCARD8	24	85	**4 9 19 3 1** 18 21 4 4 4
DISCARD9	45	87	4 15 **15 9 19 2 1** 21 4 4
DISCARD10	20	88	**4 9 19 3 1** 21 4 4 4 4 4

Misclassified...as...		# times/10
18	21	6
3	2	5

9 **DISCERN – 4 9 19 3 5** 18 14

DISCERN1	11	71	**4 9 19 2 15** 18 7 7 7 7
DISCERN2	32	88	4 **19 9 19 2 5** 18 14 14 14
DISCERN3	29	115	**4 9 19 2 5** 18 14 14 14 14
DISCERN4	17	89	**4 9 19 3 5** 18 7 7 7 7 7 7
DISCERN5	32	99	4 **9 19 19 2 5** 14 14 14 14
DISCERN6	20	79	**4 9 19 2 5** 18 14 14 14 14
DISCERN7	41	86	4 9 **15 15 19 2 5** 1 14 14 14
DISCERN8	28	81	4 **24 9 19 2 5** 18 7 14
DISCERN9	16	65	**4 9 19 2 5** 18 14 14 14
DISCERN10	31	84	4 **24 9 19 2 5** 7 7 7 14 14

Misclassified...as...		# times/10
3	2	9

5	15	1
14	7	4

10 **DISCI**PLE – **4 9 19 3 9** 16 12 5

DISCIPLE1	50	98	4 19 **19 9 19 >> 25** 11 5 5 5 5
DISCIPLE2	20	68	15 **4 9 19 15** 11 12 15 5 5 5
DISCIPLE3	20	94	**4 9 19 3 >>11** 7 26 5 5 5 5 5
DISCIPLE4	19	114	**4 9 19 3 25** 11 26 5 5 5 5 5
DISCIPLE5	33	85	4 **24 9 19 15 15** 11 16 5 5 5 5
DISCIPLE6	57	123	4 4 **4 9 19 3 25** 11 12 5 5 5 5
DISCIPLE7	21	122	**4 9 19 3 9** 15 16 12 15 5 5 5
DISCIPLE8	51	104	14 4 **24 9 19 3 25** 16 12 15 5
DISCIPLE9	28	79	4 **4 25 19 3 25** 15 11 12 15 5
DISCIPLE10	21	106	**4 9 19 >>15** 11 12 5 5 5 5 5 5

Misclassified…	as…	# times/10
9	25	5
16	11	6
12	26	2
3/9?	15	3

11 **DISCO**VER – **4 9 19 3 15** 22 5 18

DISCOVER1	10	70	**4 9 19 3 15** 22 18 18 18 18
DISCOVER2	24	81	**4 9 19 3 15** 22 19 18 18 18
DISCOVER3	32	96	**4 9 19 2 15** 22 18 18 18 18
DISCOVER4	13	70	**4 9 19 3 15** 22 24 18 18 18
DISCOVER5	15	69	**4 9 19 >>15** 11 24 18 18 18
DISCOVER6	19	87	**4 9 19 3 15** 22 9 18 18 18
DISCOVER7	20	76	**4 9 19 2 15** 22 24 18 18 18
DISCOVER8	22	90	**4 9 19 3 15** 22 18 18 18 18
DISCOVER9	20	74	**4 9 19 3 15** 22 24 18 18 18
DISCOVER10	25	90	4 **4 9 19 3 >>22** 18 18 18 18

Misclassified…	as…	# times/10
3	2	
5	9, 24, 19	

12 **DISCU**SS – **4 9 19 3 21** 19 19

DISCUSS1	16	73	**4 9 19 2 21** 19 19 19 19 19
DISCUSS2	11	64	**4 9 19 >>21** 19 19 19 19 19
DISCUSS3	40	97	4 **9 19 15 3 21** 21 19 19 19
DISCUSS4	16	80	**4 9 19 >> 21** 19 19 19 19
DISCUSS5	13	87	**4 9 19 >> 21** 19 19 19 19
DISCUSS6	16	72	**4 9 19 3 21** 19 19 19 19
DISCUSS7	13	74	**4 24 9 19 >> 21** 19 19 19

DISCUSS8	46	102	4 24 **9 19 1 3 15** 19 19 19
DISCUSS9	18	80	**4 25 19 14 8** 19 19 19 19
DISCUSS10			**4 9 9 19 8** 19 19 19 19 19

Misclassified…	as…	# times/10
3	2	1
21	15	1
9	25	1

APPENDIX C

SAS FILE

SAS Code for MANOVA

```
dm "output;clear;log;clear";
ODS HTML body=     "skulls-body.html"
         contents="skulls-contents.html"
         frame=    "skulls-frame.html"
         page=     "skulls-page.html"
         headtext=""
         anchor="skulls";
OPTIONS PS=55 LS=80 PageNo=1 NoDate
        FORMCHAR='|----|+|---+=|-/\<>*';
TITLE1 'C_observations for Subject 1';
DATA C_observations1;
SET C_observations1;
      LABEL        j1 = 'Thumb rotation'
                   j2 = 'Thumb MPJ'
                   j3 = 'Thumb IJ'
                   j4 = 'Thumb abd'
                   j5 = 'Index MPJ'
                   j6 = 'Index PIJ'
                   j7 = 'Middle MPJ'
                   j8 = 'Middle PIJ'
                   j9 = 'M-I abd'
                   j10= 'Ring MPJ'
                   j11= 'Ring PIJ'
                   j12= 'R-M abd'
                   j13= 'Pinkie MPJ'
                   j14= 'Pinkie PIJ'
                   j15= 'P-R abd'
                   j16= 'Palm arch'
                   j17= 'Wrist pitch'
                   j18= 'Wrist yaw';
OPTIONS NOCENTER NONUMBER NODATE LINESIZE=64;
TITLE2 'Using PROC DISCRIM to test for Homogeneity of Covariances';
/* The pool = test statement carries out Bartlett's
test for equality of covariance matrices.  An alpha value
of 0.05 is used, given by slpool = 0.05. The priors proportional
statement below just
requests that we use prior probabilities for each group that
are proportional to sample sizes.  */
PROC DISCRIM DATA=work.C_observations1 POOL=Test SLPOOL=.05 WCOV PCOV;
CLASS Group;
PRIORS proportional;
VAR j1 j2 j3 j4 j5 j6 j7 j8 j9 j10 j11 j12 j13 j14 j15 j16 j17 j18 ;
Run;
TITLE2 MANOVA for C_comparisons Subject1;
/* The hovtest = levene statement carries out Levene's
test for equality of variance matrices. The multivariate analysis
of variance (MANOVA) is carried out by the MANOVA H=Group statement
```

```
which performs a MANOVA on the Group and Error as well as on the Groups
defined by the CONTRAST statements. */
PROC GLM DATA=work.C_observations1 ORDER=data;
CLASS Group;
MODEL j1 j2 j3 j4 j5 j6 j7 j8 j9 j10 j11 j12 j13 j14 j15 j16 j17 j18 =
Group;
CONTRAST 'Static vs Context' Group -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1 -1 -1 20;
MANOVA H=Group / PRINTE PRINTH SUMMARY;
MEANS Group / HOVTEST=levene;
RUN;
ODS HTML Close;
quit;
```

REFERENCES

Akamatsu, C.T., "The Acquisition of Fingerspelling in Pre-school Children." *Unpublished doctoral dissertation*, University of Rochester, 1982.

Averbeck, B.B., Chafee, M.A., Crowe, D.A., and Georgopoulos, A.P., "Parallel processing of serial movements in prefrontal cortex." *Proceedings of the National Academy of Sciences*, 99, 13172 - 13177, 2002.

Box, G.E., "Non-normality and Tests on Variance." *Biometrika*, 40, 318 - 335, 1953.

Carey, G., (1998), "Multivariate Analysis of Variance (MANOVA) I: Theory." Retrieved January 2004 from http://ibgwww.colorado.edu/~carey/p7291dir/handouts/manova1.pdf

Carey, G., (1998), "Multivariate Analysis of Variance (MANOVA) II: Practical Guide to MANOVA and ANOVA for SAS." Retrieved January 2004 from http://ibgwww.colorado.edu/~carey/p7291dir/handouts/manova2.pdf

Carey, G., (1998), "The General Linear Model: Theory." Retrieved January 2004 from http://ibgwww.colorado.edu/~carey/p7291dir/handouts/glmtheory.pdf

Daniloff, R.G. and Hammarberg, R.E., "On defining coarticulation." *Journal of Phonetics*, 1, 239 – 248, 1973.

Duda, R.O., Hart, P.E., Stork, D.G., *Pattern Classification*, 2nd ed. New York: Wiley-Interscience, 2001.

Emmorey, K., "Sign language: a window into the brain, language, and cognition." *Encyclopedia of Human Behavior*, San Diego, C.A.: Academic Press, 1994.

Hansen, J., "A Matlab Project in Optical Character Recognition (OCR)." Retrieved October 2003 from http://www.ele.uri.edu/~hansenj/projects/ele585/OCR/

Hanson, V.L., "When a word is not the sum of its letters: Fingerspelling and spelling." *Teaching American Sign Language as a second/foreign language*, Ed. Caccamise, F. Garretson, M. and Bellugi, U., Silver Spring, 1982.

Hardcastle, W.J., and Hewlett, N., eds., *Coarticulation: Theory, Data and Techniques*, Cambridge: Cambridge University Press, 1999.

Jerde T.E., Soechting J.F., and Flanders M., "Coarticulation in fingerspelling." *Journal of Neuroscience*, 23, 2383 - 2393, 2003.

Katz, William F., "Anticipatory coarticulation and aphasia: implications for phonetic theories." *Journal of Phonetics*, 28, 313 – 334, 2000.

Liddell, S.K., "Some current theoretical approaches to the study of American Sign Language." *Semiotica,* 31, 345 - 368, 1980.

Martin, J., "A Linguistic Comparison: Two Notation Systems for Signed Language: Stokoe Notation & Sutton SignWriting." Retrieved November 2002 from http://signwriting.org/library/pdf/pdf/sw0032%20Comp%20Stokoe%20Sutton.pdf

MathWorks, Inc., (2003), *MatLAB® 6.5: The Language of Technical Computing*, Retrieved August, 2003 from: http://www.mathworks.com/products/matlab/

Mcleish, E., "Bayesian Theory with Gaussian Distributions." Retrieved February, 2004 from http://www.cs.mcgill.ca/~mcleish/644/main.html

Meirovitch, L., *Introduction to Dynamics and Control*, New York: John Wiley & Sons, 1985.

Ostry D.J., Gribble P.L., & Gracco V.L. "Coarticulation of jaw movements in speech production: is context sensitivity centrally planned?" *Journal of Neuroscience*, 16, 1570 - 1579, 1996.

Poizner, H., Klima, E.S., Bellugi, U., and Livingston, R.B., "Motion analysis of grammatical processes in a visual-gestural language." *Event Cognition: An Ecological Perspective*, Ed. McCabe, V. and Balzano, G.J., Hillsdale, N.J.: Lawrence Erlbaum Ass., 1986.

Poizner, H., Klima, E., and Bellugi, U., *What the Hands Reveal About the Brain*, Cambridge, MA: MIT Press, 1987.

Reich, P.A., "Visible distinctive features." Ed. A. Makkai and V.B. Makkai, *The First LACUS Forum 1974,* Columbia, S.C.: Hornbeam Press, 1974.

Romero, J., "Articulatory blending of lingual gestures." *Journal of Phonetics*, 24, 99 – 111, 1996.

Rumelhart, D.E., and Norman, D.A., "Simulating a skilled typist: a study of skilled cognitive-motor performance." *Cognitive Science*, 6, 1 - 36, 1982.

Santello, M., Flanders, M., Soechting, J.F., "Postural hand synergies for tool use." *Journal of Neuroscience*, 18, 10105 – 10115, 1998.

Santello, M., Flanders, M., Soechting, J.F., "Patterns of hand motion during grasping and the influence of sensory guidance." *Journal of Neuroscience*, 22, 1426 – 1435, 2002.

SAS Institute, Inc., (1999-2001), *SAS Software Release 8.2*, Retrieved February 2004 from http://www.sas.com/software/index.html

Schieber, M.H., "Individuated finger movements of rhesus monkeys: a means of quantifying the independence of the digits." *Journal of Neurophysiology*, 65, 1381 – 1391, 1991.

Soechting J.F, and Flanders, M., "Organization of sequential typing movements." *Journal of Neurophysiology*, 67, 1275-1290, 1992.

Speers, D.L., "Representation of American Sign Language for Machine Translation." *Unpublished doctoral dissertation*, Georgetown University, 2001.

Stokoe, W.C., *Sign Language Structure: An Outline of the Visual Communication System of the American Deaf*, Buffalo, N.Y.: University of Buffalo, 1960.

Terleski, J., "Handwriting Recognition using Matlab." Retrieved August 2003 from http://www.contrib.andrew.cmu.edu/~jterlesk/robotics/writing/writing.html

Thermo Galactic, "Discriminant Analysis, The Mahalanobis Distance", Retrieved February 2004 from http://www.galactic.com/Algorithms/discrim_mahaldist.htm

Trujillo-Ortiz, A., Hernandez-Walls R., Castro-Morales K., Espinoza-Tenorio A., Guia-Ramirez A., Muniz-Salazar R., Luis Sanchez-Osorio J.and Carmona-Pina R., (2002) "Mulnortest: Multivariate normality statistical test. A MATLAB file." Retrieved April 2004 from http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=2746&objectType=FILE.

Tyrone, M.E, Kegl, J., and Poizner, H., "Interarticulator co-ordination in Deaf singers with Parkinson's disease." *Neuropsychologia,* 37, 1271-1283, 1999.

Virtual Technologies, Inc., *Cyber*Glove® *Reference Manual*, Palo Alto, CA, 1998.

Wood, S. A. J. "Assimilation or coarticulation? Evidence from the coordination of tongue gestures for the palatalization of Bulgarian alveolar stops." *Journal of Phonetics*, 24, 139 - 164, 1996.

Wilcox, S., *The Phonetics of Fingerspelling*, Philadelphia: John Benjamins, 1992.

Winter, D.A., *Biomechanics and Motor Control of Human Movement*, 2nd ed., New York: Wiley-Interscience, 1990.

Zakia, R.D. and Haber, R.N., "Sequential letter and word recognition in deaf and hearing subjects." *Perception and Psychophysics*, 9, 110 – 114, 1971.