Classification of Finger Gestures From Myoelectric Signals

by

Peter M. Ju

Submitted to the Department of Electrical Engineering and Computer Science in partial fulfillment of the requirements for the degrees of

Bachelor of Science in Electrical [Computer] Science and Engineering

and

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2000
[June 2000]

Author ....
Department of Electrical Engineering and Computer Science
May 18, 2000

Certified by.......................................
Leslie P. Kaelbling
Associate Professor
Thesis Supervisor

Accepted by.......................................
Arthur C. Smith
Chairman, Department Committee on Graduate Students

Classification of Finger Gestures From Myoelectric Signals

by

Peter M. Ju

Submitted to the Department of Electrical Engineering and Computer Science
on May 18, 2000, in partial fulfillment of the
requirements for the degrees of
Bachelor of Science in Electrical [Computer] Science and Engineering
and
Master of Engineering in Electrical Engineering and Computer Science

Abstract

Electromyographic signals may provide an important new class of user interface for consumer electronics. In order to make such interfaces effective, it will be crucial to map EMG signals to user gestures in real time. The mapping from signals to gestures will vary from user to user, so it must be acquired adaptively. In this thesis, I describe and compare three methods for static classification of EMG signals. I then go on to explore methods for adapting the classifiers over time and for sequential analysis of the gesture stream by combining the static classification algorithm with a hidden Markov model. I conclude with an evaluation of the combined model on an unsegmented stream of gestures.

Thesis Supervisor: Leslie P. Kaelbling
Title: Associate Professor

Acknowledgments

I would like to thank Yoram Singer for being my primary research advisor at AT&T Labs. Without his help, this thesis would not have been possible. I would also like to thank Will Hill for suggesting this research and for numerous helpful discussions. Thanks also to Anna Gilbert for valuable discussions.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In the past, as computers became more powerful, new interfaces were needed to take advantage of that power. Recently, digital devices have tended towards portability as well as increasing in speed. The demand for smaller cellular phones, and personal digital assistants is proof of this trend. Consequently, new interfaces now have to accommodate portability as well as power. Currently, all input schemes for these devices rely on pressing buttons (Walkmans, cellular phones), using a stylus (palm pilots), or speech input (cellular phones). These interfaces are all lacking in certain aspects. The first two methods involves a user removing the device from its storage area (pocket, backpack, etc) and manipulating the device with his hands. This maybe difficult if the storage area is not easily accessible or if the user's hands are occupied. The second method requires the user to speak out loud in order to use the device, which maybe awkward in public or quiet areas.

A new input device has been devised that enables a user to control their portable devices with simple small finger gestures. Electromyographic (EMG) signals are recorded using a wrist-worn input device. When a user performs a finger gesture, the EMG signal is sent to nearby devices using a small-field wireless broadcaster. The devices classify the signal, mapping it to the gesture performed, and then carry out the actions associated with the gesture. For example, a user could touch his index finger to his thumb. The wrist-worn EMG electrode records the signal and sends it to a portable CD player through the broadcaster. The CD takes the EMG signal, and classifies it as an index tap, and then fast forwards the CD.

In order to make such interfaces effective, it will be crucial to map EMG signals to user gestures in real time. The mapping from signals to gestures will vary from user to user, so it must be acquired adaptively. In addition, from time to time for a single user, the position of the electrodes will vary, requiring the classifier to be adapted online.

This thesis explores the creation of a classification system for EMG signals of finger gestures. The goal of this project is to have a classification scheme that can classify streaming EMG data in real-time and that can be retrained and adapted quickly for different people and different days of use. As a starting point, this thesis first compares three methods for static (batch) classification of EMG signals. Then it goes on to explore methods for adapting and retraining the classifiers over time. Finally, the thesis explores augmenting some static classification methods for sequential (real time) analysis of streaming EMG data.

1.1 Background

Electromyographic signals have been extensively studied in the past. These signals are attractive because they are easily measured, and their power has a qualitative relationship with the force of a muscle contraction. However, they are difficult to work with due to the various sources of noise for an EMG signal, and the fact that no quantitative relationship between the power of an EMG signal and the force of a contraction has been found. Surface Electromyography is a wide field, and has a number of applications for many problems.

1.1.1 Surface Electromyography

When a muscle is voluntarily activated, action potentials from the motor cortex propagate down to the spinal cord. Here the action potentials are relayed to motor neurons. Each motor neuron propagates the signal down to all its branches, which are each connected to a muscle fiber. A motor neuron along with all of its branches and muscle fibers is called a motor unit. The combined action potentials of all the fibers in a motor unit is called the motor unit action potential (MUAP). A series of MUAP's is called a motor unit action potential train. An electromyographic (EMG) signal is comprised of a summation of the MUAP trains created when a contraction of a muscle occurs. Action potentials propagate through fat, tissue, and skin. Therefore, a surface electrode placed on the skin can detect these signals. A surface EMG signal is a EMG signal that has been detected by an electrode placed on the surface of the skin.

1.1.2 Attributes of Signal

The characteristics of surface EMG signals have been studied extensively. Since theses signals are a summation of many MUAP's, they can be modeled as a stochastic process approximated by a Gaussian distribution. The majority of the signal energy occurs under 500 Hz with energy peaking at around 50-150 Hz. The voltage can range from 0 to 10 mV peak-to-peak. For short periods of time (around 0.1 seconds), EMG signals are stationary. A stationary signal is a signal whose probability distribution function is independent of time. Traditionally, surface EMG's have been modeled as an AR signal, and experiments have shown that models of order higher than four do not improve perfromance. While the AR model for EMG signals seems to work well for short periods of time, it has been our experience that it does not perform well when the electrodes are slightly shifted in position or between days.

1.1.3 Finger Anatomy

This study looks at classifying figure gestures from surface EMG signals. This is a difficult problem, since the the movements of the fingers are extremely varied and are controlled by a combination of muscle groups. The muscle movements that compose the finger gestures to be classified are the extension and flexion of all the fingers and thumb, and the abduction and adduction of the thumb. If an electrode is placed exactly over every muscle that was involved in the finger gestures, the device would be too bulky and could not be worn as a wrist-watch. However, since EMG signals travel through tissue, an electrode placed in the proper area can receive cross-talk from the various muscles involved in the a gesture. Therefore, identification of finger gestures can still be carried out using two electrodes located around the wrist.

Chapter 2

Previous Work

The majority of EMG research has been related to the design of prosthetics for amputees. A standard method for controlling an artificial limb is to train an amputee to flex the remaining muscles on his limb such that the EMG signals produced are easily distinguishable. These signals are then used as commands for the prosthetic.

Graupe et al. [7] were the first researchers to classify EMG signals with reasonable success. They modeled the EMG signal as autoregressive, and were able to classify with an 85% success rate. They only dealt with the case where a single channel of data was being recorded.

Doerschuck et al. [8] improved upon Graupe's traditional AR approach for EMG signal classification by using the cross-correlation information between the EMG signals of different electrodes. They formulated a model in which scalar AR coefficients were replaced by AR matrices that took into account multiple data channels and their relationships.

Surprisingly, the classical AR technique is still the primary method used to classify EMG signals. However, there have been experiments with other techniques. Instead of using the traditional AR coefficients for EMG signal classification, Kang et al. [11] explore the use of Cepstral coefficients as a feature set. They found that there was a statistically significant improvement in classification correctness when using Cepstral coefficients over AR coefficients. The article also looked at the difference between Cepstral and AR coefficients in feature space and the frequency domain and found that Cepstral coefficients have better cluster separation than AR coefficients. In their experiments, the AR method demonstrated a 78.7% classification rate, and the Cepstral coefficient method demonstrated an 83.7% classification rate when classifying ten distinct gestures with four channels recording data.

Englehart et al. [6] attempted to classify EMG signals using various time-frequency based representations with dimensionality reduction. They used the short-time Fourier transform, the wavelet transform, and the wavelet packet transform as their features for classification. For dimensionality reduction, they used methods such as Euclidean distance class separability and principle components analysis. In their experiments, Englehart et al. achieved a maximum classification rate of 93.7 % using the coiflet wavelet packet family as a feature set, principle components analysis for feature selection, and linear discriminative analysis for classification. They were classifying four gestures, and recorded two channels of data.

The classification results from these experiments look very promising, however, their experimental setup differs considerably from this thesis. Since these studies have been geared towards creating controls for prosthetic limbs, they have focused on large movements using large muscle groups. The goal of this project is to classify small finger gestures controlled by a complicated set of small muscle groups using an unobtrusive sensor placed in a location that is convenient for the user, but highly suboptimal for signal recording.

In pilot experiments, the methods described by Kang et al. [11] and Englehart et al. [6] were implemented, and their performance was quite poor. This motivated the development of additional classification methods, as well as the incorporation of dynamic training and sequential analysis to improve performance.

Chapter 3

Static Classification

The first problem addressed in this thesis is simple supervised classification of single-channel signals into one of four classes. The signals are represented as sequences of real-valued samples of the EMG signal generated during a particular gesture.

In this thesis, two different approaches to classification are explored: Generative model classification and discriminative classification.

A generative approach to signal classification is to assume a model for the signals you wish to classify. This model has parameters which can be set, and different classes have different parameters. Once the model is chosen, and the parameters estimated for each class, a signal is classified by how well it fits each class model. The AR model is the generative classification scheme explored in this thesis. It is the first and most widely used method to classify EMG signals.

In a discriminative classification scheme, no assumptions are made about the signal. An algorithm directly tries to classify the signals without the creation of an intermediate model. The discriminative method tested in this thesis involves two parts. The first part is to take training data and project it onto another (possibly overcomplete) basis; possible bases could be the Gabor space or a wavelet packet decomposition. The second step is to take the resulting coefficients from the transform and use them as features for a machine learning algorithm to weigh, and create a classifier with. This system is essentially determining the most important coefficients to use to classify a signal. It is analogous to systems such as matching pursuit [4], best basis [5], and basis pursuit [3] which try to find the best projection to encode a signal. The learning algorithm used is iterative scaling (IS) which is discussed below.

In this thesis, three classification methods were implemented and tested: a simple nearest neighbor method on unprocessed data as a base case, a variation of Doerschuck's AR technique, and the new discrimative method based on iterative scaling.

3.1 Nearest Neighbor

As a baseline system, a simple nearest-neighbor classifier was implemented using cross-correlation between signals as the distance metric. In the static classification setting you are given a set of training signals $\bar{x}_k = (x_k[1], x_k[2], \ldots, x_k[n])$, each of length N, and associated classes c_k. For a test signal $\bar{u}$, compute an inverse distance between the test and training signals as

$$D(\bar{x}_k, \bar{u}) = \sum_{n=1}^{N} x_k[n]\, u[n] \; .$$

The class assigned to signal $\bar{u}$ is the class, c_k, of the signal $\bar{x}_k$ for which $D(\bar{x}_k, \bar{u})$ is maximized. In order to remove errors due to imprecise segmentation of the signals, calculate the cross-correlation between each $\bar{x}_k$ and many shifts of $\bar{u}$, and then assign to signal $\bar{u}$ the class of $\bar{x}_k$ for which $D(\bar{x}_k, \bar{u})$ is maximized. The comparison of shifts of the same signal can be done efficiently using the fast Fourier transform in $N \log(N)$ time per test signal [23].

3.2 Autoregressive Models

The second approach tested was the standard method for classifying EMG signals, which is a generative approach using AR models. The AR model seems to be able to capture many of the important attributes of the EMG signals. A pth-order AR signal is modeled by the equation:

$$x[n] = \sum_{i=1}^{p} a_i x[n-i] + e_\sigma[n], \tag{3.1}$$

where $\bar{x} = (x[1], x[2], ..., x[N])$ is an AR signal, $\bar{e}_\sigma = (e_\sigma[1], e_\sigma[2], ..., e_\sigma[N])$ is zero-mean white noise, and $a_1...a_p$ are the AR coefficients.

The value of a pth-order AR signal at time sample n is a weighted sum of the previous p samples of the signal with white noise added to it. To fully specify an AR model, the AR coefficients and the variance of the noise need to be determined. Solving for the AR coefficients can be formulated as a matrix equation, and is equivalent to a least squares problem:

$$\underbrace{\begin{bmatrix} x[p] & x[p-1] & x[p-2] & \cdots & x[1] \\ x[p+1] & x[p] & x[p-1] & \cdots & x[2] \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x[N-1] & x[N-2] & x[N-3] & \cdots & x[N-p] \end{bmatrix}}_{X} \underbrace{\begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_p \end{bmatrix}}_{a} = \underbrace{\begin{bmatrix} x[p+1] \\ x[p+2] \\ \vdots \\ x[N] \end{bmatrix}}_{b},$$

where N is the length of the signal.

Therefore, to find the least squares solution for a, calculate:

$$a = (X^T X)^{-1} X^T b.$$

After estimating the AR coefficients, the variance must be estimated to fully specify the AR model. Filter the signal, $\bar{x}$, with the AR coefficients to calculate the estimated signal:

$$\hat{x}[n] = \sum_{i=1}^{p} a_i x[n-i].$$

Then, subtract the estimated signal, $\hat{x}$, from the actual signal, $\bar{x}$, to calculate the error signal. From the error signal you can then estimate the variance of the noise:

$$\sigma^2 = \sum_{n=1}^{N} \frac{(\hat{x}[n] - x[n])^2}{N}.$$

A classification scheme can be created using these AR model estimates by building autoregressive (AR) models from the training data of each class, then classifying new signals according to how far they deviate from the AR models of the classes. The AR coefficients are estimated from the training signals using a least squared error estimation method to minimize the difference between the actual signals and those that would have been generated by the model,

$$\sum_{j} \sum_{n=1}^{N} (x_j^k[n] - \sum_{i=1}^{p} a_i^k x_j^k[n-i])^2 \ ,$$

where $\bar{x}_j^k$ ranges over training signals for a class k. For each class, k, Solve the following equation for a:

$$\underbrace{\begin{bmatrix}
x_1^k[p] & x_1^k[p-1] & x_1^k[p-2] & \cdots & x_1^k[1] \\
x_1^k[p+1] & x_1^k[p] & x_1^k[p-1] & \cdots & x_1^k[2] \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
x_1^k[N-1] & x_1^k[N-2] & x_1^k[N-3] & \cdots & x_1^k[N-p] \\
x_2^k[p] & x_2^k[p-1] & x_2^k[p-2] & \cdots & x_2^k[1] \\
x_2^k[p+1] & x_2^k[p] & x_2^k[p-1] & \cdots & x_2^k[2] \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
x_2^k[N-1] & x_2^k[N-2] & x_2^k[N-3] & \cdots & x_2^k[N-p] \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
x_J^k[p] & x_J^k[p-1] & x_J^k[p-2] & \cdots & x_J^k[1] \\
x_J^k[p+1] & x_J^k[p] & x_J^k[p-1] & \cdots & x_J^k[2] \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
x_J^k[N-1] & x_J^k[N-2] & x_J^k[N-3] & \cdots & x_J^k[N-p]
\end{bmatrix}}_{X}
\underbrace{\begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_p \end{bmatrix}}_{a}
=
\underbrace{\begin{bmatrix}
x_1^k[p+1] \\ x_1^k[p+2] \\ \vdots \\ x_1^k[N] \\ x_2^k[p+1] \\ x_2^k[p+2] \\ \vdots \\ x_2^k[N] \\ \vdots \\ x_J^k[p+1] \\ x_J^k[p+2] \\ \vdots \\ x_J^k[N]
\end{bmatrix}}_{b},$$

Then for each class, calculate the AR-model predicted signals of the training signals by filtering the training signals through the AR coefficients for their class:

$$\hat{x}_j^k[n] = \sum_{i=1}^{p} a_i^k x_j^k[n-i],$$

where $\hat{x}_j^k[n]$ is the AR-model predicted signal for the jth training signal of the kth class and a_i^k is the ith AR coefficient for the kth class.

Then estimate the variance for each class:

$$\sigma_k^2 = \sum_{j=1}^{J}\sum_{n=1}^{N} \frac{(\hat{x}_j^k[n] - x_j^k[n])^2}{JN},$$

Once models have been estimated for every class, to classify a signal, $\bar{u} = (u[1], u[2], ..., u[N])$, filter it with each class' AR coefficients:

$$\hat{u}^k[n] = \sum_{i=1}^{p} a_i^k u[n-i].$$

where $\hat{u}^k$ is the estimated signal of $\bar{u}$ with respect to the kth class.

Classify the signal, $\bar{u}$, by calculating its negative log likelihood with respect to each class:

$$L(\bar{u}|k) = N \log\sqrt{2\pi\sigma_k^2} + \sum_{n=1}^{N} \frac{(\hat{u}^k[n] - u[n])^2}{2\sigma_k^2},$$

where $L(\bar{u}|k)$ is the likelihood of the sequence, $\bar{u}$, occurring if it was of class k. Classify the signal as the class whose model produces the smallest log likelihood.

Log likelihood is useful for determining the class of an entire sequence, however it is not very useful when trying to classify a signal at every time step. Class probablity curves makes it possible to calculate the latter. A class probability curve is an estimate of the probability of a class at every time step. They can be calculated by using the following equation:

$$p_k[n] = \frac{P_k(u[n]|U^{n-1})p_k[n-1]}{\sum_{k=1}^{K} P_k(u[n]|U^{n-1})p_k[n-1]},$$

where K is the number of classes, $p_k[n]$ is the probability that gesture k is taking place, given data up to and including $u[n]$, $U^{n-1} = (u[1], u[2], \cdots, u[n-1])$ and $P_k(u[n]|U^{n-1})$ is the probability of sample, $u[n]$, given U^{n-1} and gesture k.

The difficult part of calculating $p_k[n]$ is determining $P_k(u[n]|U^{n-1})$. However, if the signal is AR, the value at time sample n is the estimated signal with respect to a given gesture, $\hat{u}^k[n]$, with a Gaussian noise element added. Therefore, $P_k(u[n]|U^{n-1})$ becomes a Gaussian with mean $\hat{u}^k[n]$ and variance σ_k^2. A simple method of producing a class estimate at every time step is to take the class probability curves and find the class with the maximum probability at every step.

Equation 3.1 explains the single channel description that Graupe et al. used to classify EMG signals. Doerschuk et al. extended AR modeling to a multichannel technique that accounted for cross-correlation among different channels. This framework is described by the equation:

$$\underbrace{\begin{bmatrix} x_1[n] \\ x_2[n] \\ \vdots \\ x_M[n] \end{bmatrix}}_{X[n]} = \sum_{i=1}^{p} \underbrace{\begin{bmatrix} a^i_{1,1} & a^i_{1,2} & \cdots & a^i_{1,M} \\ a^i_{2,1} & a^i_{2,2} & \cdots & a^i_{2,M} \\ \vdots & \vdots & \vdots & \vdots \\ a^i_{M,1} & a^i_{M,1} & \cdots & a^i_{M,M} \end{bmatrix}}_{A^i} \underbrace{\begin{bmatrix} x_1[n-i] \\ x_2[n-i] \\ \vdots \\ x_M[n-i] \end{bmatrix}}_{X[n-i]} + e[n],$$

where M =the total number of channels, X is an M by L matrix with each row representing a channel signal, $X[n]$ is a M vector representing all the channel values at time sample ,n, $A^1 \cdots A^p$ are the AR matrices and $e[n]$ is white noise. The matrix entries $a_{j,k}$ represent the cross-talk between the different channels.

Determining the AR coefficients,$A^1 \cdots A^p$, for the multi-channel case can also be formulated as a matrix problem:

$$Xa = b,$$

where

$$X = \begin{bmatrix} x_1[p] & \cdots & x_1[1] & x_2[p] & \cdots & x_2[1] & \cdots & x_M[p] & \cdots & x_M[1] \\ x_1[p+1] & \cdots & x_1[2] & x_2[p+1] & \cdots & x_2[2] & \cdots & x_M[p+1] & \cdots & x_M[2] \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_1[N-1] & \cdots & x_1[N-p] & x_2[N-1] & \cdots & x_2[N-p] & \cdots & x_M[N-1] & \cdots & x_M[N-p] \end{bmatrix},$$

$$a = \begin{bmatrix} a_{1,1}[1] & a_{1,2}[1] & \cdots & a_{1,M}[1] \\ a_{1,1}[2] & a_{1,2}[2] & \cdots & a_{1,M}[2] \\ \vdots & \vdots & \vdots & \vdots \\ a_{1,1}[p] & a1,2[p] & \cdots & a_{1,M}[p] \\ a_{2,1}[1] & a_{2,2}[1] & \cdots & a_{1,M}[1] \\ a_{2,1}[2] & a_{2,2}[2] & \cdots & a_{1,M}[2] \\ \vdots & \vdots & \vdots & \vdots \\ a_{2,1}[p] & a2,2[p] & \cdots & a_{1,M}[p] \\ \vdots & \vdots & \vdots & \vdots \\ a_{M,1}[1] & a_{M,2}[1] & \cdots & a_{M,M}[1] \\ a_{M,1}[2] & a_{M,2}[2] & \cdots & a_{M,M}[2] \\ \vdots & \vdots & \vdots & \vdots \\ a_{M,1}[p] & aM,2[p] & \cdots & a_{M,M}[p] \end{bmatrix},$$

$$b = \begin{bmatrix} x_1[p+1] & x_2[p+1] & \cdots & x_M[p+1] \\ x_1[p+2] & x_2[p+2] & \cdots & x_M[p+2] \\ \vdots & \vdots & \vdots & \vdots \\ x_1[N] & x_2[N] & \cdots & x_M[N] \end{bmatrix},$$

Specifying the rest of the model and classifying with it is analogous to the single channel case, however, the variance is replaced with a covariance matrix and the Gaussian distribution used to calculate the log likelihood and probability curves becomes a multivariate Gaussian.

3.3 Discriminative Methods and Iterative Scaling

Auto-regressive models provide a simple and efficient approach for the problem of classification of *stationary* signals. However, the conditional probability model induced by auto-regressive modeling is deficient. This is mainly due to the fact that the AR method is a *generative* approach. The AR models for the different gestures contains information that best describes each class, which does not necessarily coincide with information that best distinguishes classes from one another. The AR models might concentrate on information common to all the gestures. The class probability estimates are computed (using Bayes' rule) from the individual AR models. Hence, deficiency of the AR models result in probability-"smeared" estimates.

The third classification technique tested in this thesis is a *discriminative* approach that instead of calculating $P(x|y)$ as in the generative approach, directly estimates the class-conditional probabilities $P(y|x)$. The approach is based on a parameter estimation technique called iterative scaling (IS) [17, 18].

3.3.1 Discriminative Methods

Consider the problem of separating two different groups of points plotted on a two dimensional grid. There are many ways you could try to approach this problem, but the first idea you would probably try is to draw a line such that one set of points is on one side of the line, and the other set of points is on the other side of the line. This is exactly the approach that certain machine learning algorithms take. More formally, the goal of some machine learning (ML) algorithm is to find a hyperplane in feature space that best separates two classes. The separating hyperplane is a weighted sum of the features given to the machine learner, and therefore is also known as a linear threshold function:

$$\alpha_0 + \alpha_1 * x_1 + \alpha_2 * x_2 + \cdots + \alpha_N * x_N = 0$$

where $\alpha_0 \cdots \alpha_N$ are the weights, and $x_1 \cdots x_N$ are the feature values.

In order to classify with a linear threshold function, input the features of the item you wish to classify into the equation, if it is less that zero, it is on one side of the hyperplane, and therefore one class, and if it is greater than zero it is on the other side of the hyperplane, and therefore the other class.

3.3.2 Iterative Scaling

Iterative Scaling is a machine learning technique that allows us to create class-conditional probabilities. It uses a conditional exponential model of the form:

$$P_{\bar{\alpha}}(y|x) = \frac{1}{Z(x)} \exp\left(\sum_{i=1}^{l} \alpha_i f_i(x,y)\right). \tag{3.2}$$

Here $y \in \{1, \dots, 4\}$ is the gesture (finger) index. The functions $f_i(x,y)$ are called features and can take real values. Discussed below are methods for choosing features. α_i is a real-valued weight associated with f_i. Denote by $\bar{\alpha}$, the vector of weights $(\alpha_1, \dots, \alpha_l)$. $Z(x)$ is a normalization constant ensuring that $P(y|x)$ sums to one, $Z(x) = \sum_{y=1}^{4} \exp\left(\sum_{i=1}^{l} \alpha_i f_i(x,y)\right)$. Given a training set of signal-gesture pairs $\{x_k, y_k\}_{k=1}^{m}$, the log-likelihood of the training set for a particular value of $\bar{\alpha}$ is

$$\begin{aligned} L(\alpha) &= \sum_{k=1}^{m} \log\left(P_{\bar{\alpha}}(y_k|x_k)\right) \\ &= \sum_{k=1}^{m} \sum_{i=1}^{l} \alpha_i f_i(x_k, y_k) - \log(Z(x_k)) \ . \end{aligned}$$

Note that the log-likelihood is non-linear due to the term $\log(Z(\cdot))$. Therefore, it is not possible to have an analytical form for the vector of weights $\bar{\alpha}$ which maximizes the empirical log-likelihood. Iterative scaling performs additive updates to the weights, $\alpha_i \rightarrow \alpha_i + \delta_i$, guaranteeing that the log-liklihood is increased on each iteration. This thesis uses a recent variant of iterative scaling called the improved iterative scaling [18]for which there is a closed form solution for the update vector $\bar{\delta}$. Let C be a constant, defined below; then the update is

$$\delta_i = \frac{1}{C} \log(Q_i) \ , \tag{3.3}$$

where

$$Q_i = \frac{\sum_k f_i(x_k, y_k)}{\sum_k \sum_{y=1}^{4} P_{\bar{\alpha}}(y|x_k) f_i(x_k, y)} \ . \tag{3.4}$$

(Note that the value of $\bar{\delta}$ depends on the current probability estimates $P_{\bar{\alpha}}(y|x)$.) It is possible to show that this iterative process converges to the *global* maximum of the log-likelihood.

It remains to describe the choice of features for finger gesture classification. Normally, iterative scaling uses features that are binary functions from an instance-class pair (x, y) onto a 1 or 0. However, it can be shown that iterative scaling will also converge using real-valued features. To achieve a closed form update given by Eqs. (3.3) and (3.4), the features need to satisfy the following constraints:

$$\forall x, y : \quad f_i(x, y) \geq 0 \quad \text{and} \quad \sum_{i=1}^{l} f_i(x, y) = C \ . \tag{3.5}$$

In other words, all the feature values must be positive, and for every instance-class pair, the features must sum to the same value. In practice, the first constraint can be satisfied by subtracting the minimum feature value (over the training set) from all the features. The second constraint can be satisfied by adding a new *virtual* feature that forces all the features to sum up to a single constant:

$$\tilde{f}(x, y) = C - \sum_{i=1}^{l} f_i(x, y),$$

where $C = \max_{x,y} \sum_{i=1}^{l} f_i(x, y)$ is set in order to satisfy the conditions given by Eq. (3.5). It is straightforward to verify that the any weighted combination of both f_i's and $\tilde{f}$ is still linear in f_i's. Therefore, the resulting model induces a conditional class probability as given by Eq. (3.2).

For features this thesis experimented with various time-frequency decompositions including long and short time Fourier transform, and various wavelet families, of the raw EMG signal. After transforming each channel of the EMG signal, the classification scheme takes the transform's coefficients and feeds them as real-valued features to iterative scaling. Since each coefficient is a scalar, the classification scheme needs to expand it to a 4-component vector in order to to use it as a feature f_i in the form given by Eq. (3.2). For brevity and concreteness drop the feature index i and assume that c is a transform coefficient of an EMG signal for a gesture from the second class. Then define a feature f based on c as follows,

$$c \mapsto f \stackrel{\text{def}}{=} (0, c, 0, 0) .$$

In other words, a transform coefficient c generated from a signal from class i defines a feature f that "abstains" on all the classes but the class i for which its prediction is of "strength" c. This construction is reminiscent of the specialists model used in on-line prediction [19, 21] and weak hypotheses that can abstain in the context of boosting [20]. This particular form forces each feature to extend predictions only for the class from which it was derived.

In the experiments reported below, the discriminative classifier uses the first $N/2$ coefficients of the N-point discrete Fourier transform of each of the two channels of data, giving a total of N features. Since the signals are real, the DFT is symmetric. Therefore, only half the coefficients are needed. In order to classify a new EMG signal, you first project the signal onto the feature space, expand the projection as described above, and then weight the features according to the weights assigned by IS. Then classify the signal as belonging to the class which has the highest likelihood value.

Chapter 4

Sequential Decoding

In a real application of the EMG device, the classifier will not be given isolated segments of the signal to classify. Rather, it will be given a sequence of voltage samples recorded from a sequence of gestures, for which the system will have to produce a classification at every sample. The second part of the thesis deals with modifying the static AR classification technique so that it can work in real-time. The AR classification technique is used instead of the IS technique because it can be implemented more efficiently than the IS technique. More importantly, since the AR technique is a generative approach, it is easily incorporated into the sequential decoding framework of this thesis. However, future research will look at incorporating IS estimates into a sequential decoding framework.

With the AR method, sample classifications are produced by finding the estimated value at a given sample by weighting the previous p samples with the AR coefficients for each of the classes,

$$\hat{x}^k[n] = \sum_{i=1}^{p} a_i^k x[n-i] \; .$$

Then, the likelihood of an observation $x[n]$ given that it was generated by the model for class k is

$$P(x[n]|k) = \frac{1}{\sqrt{2\pi\sigma_k^2}} \exp\left(\frac{(\hat{x}^k[n] - x[n])^2}{2\sigma_k^2} \right) \; .$$

You can then classify a sample as the class model with the highest probability. However, this approach does not incorporate timing information, which can help classification. When a person taps a finger, the smallest duration for that gesture is approximately 50 to 100 milliseconds. If you are sampling at 1 kHz, the smallest duration for a gesture is 75 to 100 samples, and therefore the probability for a change in gesture is at most 1 out of 75. The naive approach does not take into account the history of the signal, making its classification results very noisy, jumping from to state to the next and back again in rapid succession.

In this thesis, two methods are tested for incorporating timing information into AR sequence decoding. The first method is a modification of Doershuck et al. probablity curve estimation. The second method is a new technique involving the use of hidden Markov models.

4.1 Doerschuk Probablity Curves

Doerschuck et al. [8] provide an alternative method for calculating class probabilities at each time step. Let $p_k[n]$ denote the probability that gesture k is taking place at time step n. Then, using Bayes' rule, you can calculate $p_k[n]$ using the following recursive equation,

$$p_k[n] = \frac{P(x[n]|k)\, p_k[n-1]}{\sum_{k=1}^{4} P_k(x[n]|k)\, p_k[n-1]} \, ,$$

where $P(x[n]|k)$ is the probability that the sample $x[n]$ occuring given class k, and defining $p_k[0] = 1/4$. From these class probability estimates, report the class that has the maximum probability at each time step. Information about the history of the signal is only propagated through the conditional class probability estimates $p_k[n-1]$.

The problem with this method is that it assumes that the whole sequence up to and including the sample you want to classify is one gesture. It assumes that if you are in a state then it is very probable that you will stay in that state. However, even though the assumption is a close approximation to reality, it is not correct and causes problems. In order to switch states, you have to build up enough evidence to prove that the entire sequence up to the sample being classified is the new gesture, and not the previous gesture. This results in a significant delay in the estimated onset time of a new gesture.

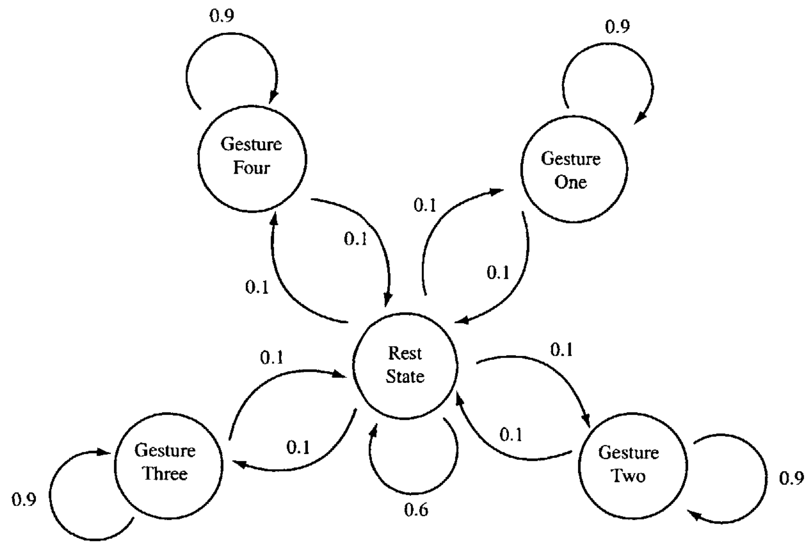

Figure 4-1: Simple HMM Model For Gesture Sequence

4.2 Hidden Markov Models

In order to incorporate timing and state transition information into the sample classification system, this thesis combines a very simple hidden Markov model (HMM) with the AR classification model. A Hidden Markov Model (HMM) consists of three major elements: An underlying set of states, a transition probability distribution that depends only on the previous state, and an observation probability distribution that depends only on the current state. For example, figure 4-1 shows a simple HMM model for the EMG voltage signal produced by a sequence of gestures. The model involves a rest state, and one state for each gesture. The transition distribution for the model allows the rest state to switch with some probability to each of the gesture states, or to itself, and allows each of the gestures states to switch back to the rest state or itself. Finally, the HMM model's observation distribution for each state is based on the AR model for each gesture:

$$P(v_k|q_j) = \frac{1}{\sqrt{2\sigma^2}} e^{\frac{(\hat{u}^j(t) - v_k)}{2\sigma^2}} \tag{4.1}$$

where $P(v_k|q_j)$ is the observation probability distribution given state j, $\hat{u}^j(t)$ is the estimated signal at time t using AR model j and v_k is the observation value.

Formally, a discrete observation HMM consists of:

T = length of the observation sequence (in the EMG example, the length of the EMG signal)

N = number of states (gestures) in the model

M = the number of observation symbols (voltages)

$Q = \{q_1, q_2, \cdots, q_N\}$ the set of states (gestures)

$V = \{v_1, v_2, \cdots, v_M\}$ the set of observation symbols (voltages)

$A = \{a_{ij} \mid a_{ij} = P(q_j \text{ at } t+1 \mid q_i \text{ at } t)\}$ state transition probability distribution

$B = \{b_j(k) \mid b_j(k) = P(v_k \mid q_j)\}$ observation symbol probability distribution in state j (Voltage Probability distribution based on the residual of the AR model)

and

$\pi = \{\pi_i \mid P(q_i \mid t = 1)\}$ initial state distribution

The benefit of an HMM model is that it brings with it a notion of state. The traditional AR probability model discussed does not have any notion of state associated with it, so when it is given a EMG signal to classify, the model assumes that the entire signal is one gesture. With an HMM model, the signal is not assumed to be one gesture, and is considered a sequence of gestures.

The HMM model used in the experiment is shown in Figure 4-2 . It consists of five major groups of states: a rest group, an index group, a middle group, a ring group and a pinky group. These groups each represent a possible gesture being performed. Each group contains 50 timing states. Since no gesture lasts for less than 50 samples, the timing states are used to ensure that the best state sequence is relatively smooth, and only allow the sequence to exit a state every 50 samples. The observation probabilities for this HMM are calculated using the same method as in the naive sequence decoding method. The Viterbi algorithm is then used to determine the most probable state sequence given the data.

Currently, the transistion probabilities are manually set because there are no sequences of gestures collected in a task setting. In future research, Baum-Welch algorithm [24] will be used to tune the model parameters based on actual sequences generated in particular tasks or by particular users.

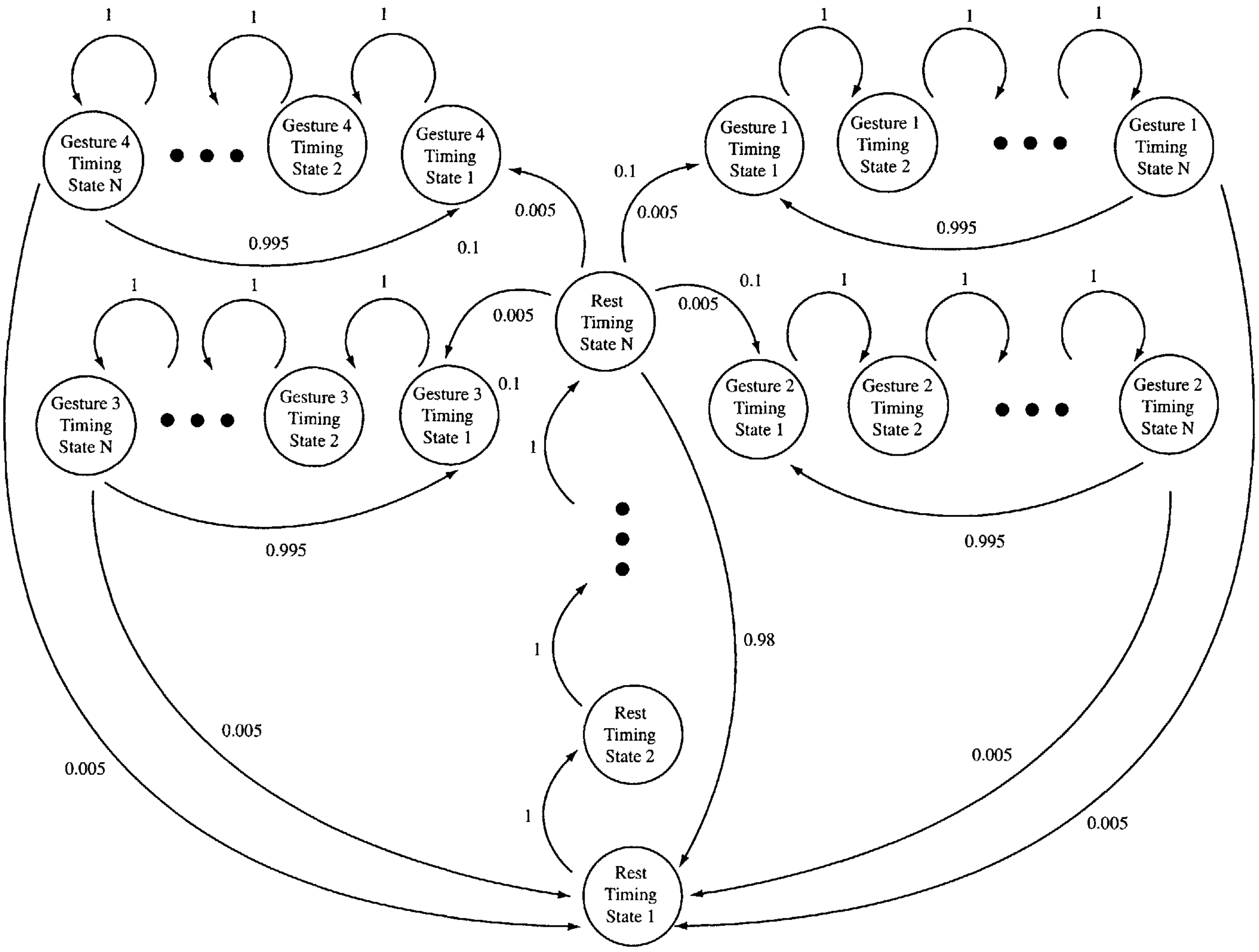

Figure 4-2: The HMM model used for sequence decoding

Chapter 5

Adaptation

The final part of the thesis is to test possible adaptation schemes to increase the speed and efficiency of training the classifier. Due to slight shifts in the position of the EMG recorder and changes in the environment and physiology of the user, a classifier trained on one day may not perform well on data collected from a different day. One possible mode for gathering additional training data is to ask the user to go through a short sequence of calibration gestures each time the device is placed on the wrist. These calibration gestures will provide an additional set of classified training examples.

One obvious strategy would be to save the original training data, add this new data, and re-run the batch training algorithm. But the device may not be capable of storing all previous data, and may require that the adaptation in reaction to the calibration data be very efficient.

This thesis uses a stochastic gradient descent (Widrow-Hoff) rule to adapt the weights of the AR model to minimize the mean squared error. Suppose you have an initial weight vector, $\mathbf{w}$, and you are given a new training example, $(\mathbf{x_t}, y_t)$, at each time t. You want to modify the weight vector after each example is given such that by example T:

$$L(\mathbf{w_T}) = \sum_{t=1}^{T} (\mathbf{w_T}\dot{\mathbf{x}}_{\mathbf{t}} - y_t)^2$$

is minimized, where T is the current example number, $\mathbf{w_T}$ is the weight vector after the T-th example, $\mathbf{x_t}$

is the input vector for example t, and y_t is the output value for example t.

In other words, you want to be able to modify the weights after each new training example, and minimize the error for all the training examples, performing as well as the batch case. Using the gradient descent heuristic, you can create an update rule that can achieve this:

$$w_{t+1,j} = w_{t,j} - \eta \frac{\delta L}{\delta w_j}$$

where $w_{t,j}$ is the j-th weight after the t-th example, η is the learning rate set by the user, and $\frac{\delta L}{\delta w_i}$ is the partial derivative of the loss function with respect to the j-th weight.

Applying the gradient descent rule to the AR classifier is relatively straightforward. In the AR case you want to minimize:

$$\sum_{t=1}^{T}(\sum_{i=1}^{p} a_i x[t-i] - x[t])^2$$

where $a_1 ... a_p$ are the AR coefficients or the weights, $x[n]$ is the signal, $x[t-1], ..., x[t-p]$ are the input values for the t-th example, and $x[t]$ is the output value for the t-th example.

In order to find the update rule, take the loss function, which in the AR case is the squared error:

$$L = (\sum_{i=1}^{p} a_i x[t-i] - x[t])^2,$$

and take the partial derivative relative to each AR coefficient and get:

$$\frac{\delta L}{\delta a_j} = 2(\sum_{i=1}^{p} a_i x[t-i] - x[t])x[t-j].$$

Therefore, the update rule is this case is:

$$a_{t+1,j} = a_{t,j} - \eta(\sum_{i=1}^{p} a_i x[t-i] - x[t])x[t-j].$$

However, this update rule works for the scenario where updates are performed after every new sample.

In reality, when the device gets a training signal, it is getting many samples at one time. In order to devise an update rule for this case, just modify the loss function:

$$L = \sum_{n=1}^{N} (\sum_{i=1}^{p} a_i x[n-i] - x[n])^2$$

Then take the partial derivative once again and get the update rule:

$$a_{t+1,j} = a_{t,j} - 2\eta \sum_{n=1}^{N} (\sum_{i=1}^{p} a_{t,i} x[n-i] - x[n]) x[n-j]$$

In experiments, a learning rate of $1/(t^{1.2})$ was used. The larger the learning rate, the more emphasis is placed on the new data. The learning rate is due to Widrow and Hoff [22]. Theoretically, in order for the Widrow-Hoff adaptation algorithm to converge to batch performance, $\sum_{t=1}^{\infty} \eta^2$ must converge to a finite value. From empirical studies, it has been shown that a learning rate, $\eta = 1/(t^{1+\delta})$ works well in practice.

In addition to updating the AR coefficients, in order to completely update an AR model, you must also update the estimated variance:

$$\sigma_{t+1}^2 = (1 - \frac{1}{t+1})\sigma_t^2 + \frac{\xi_{t+1}}{t+1},$$

where

$$\xi_{t+1} = \frac{\sum_{n=1}^{N} (\sum_{i=1}^{p} a_{t+1,i} x_{t+1}[n-i] - x_{t+1}[n])^2}{N}.$$

The two-channel AR adaptation rule is an extension of the one-channel rule, with the addition of having to update four times more coefficients, and a bivariate Gaussian instead of a univariate Gaussian.

Chapter 6

Experiments

The experiments for this thesis are devised to determine which classification method works best for static classification of EMG signals, whether or not the adaptation method is a feasible replacement for batch learning, and whether the HMM method for sequence decoding performs better than previous sequence decoders. The results from these experiments will direct future research for this system.

6.1 Equipment

Delsys Inc's Bagnoli-2 EMG System is a two-channel surface EMG acquisition system. It consists of two active differential surface electrodes, one main amplifier unit, and one reference electrode cable. The maximum output voltage range is 5 volts, and the channel frequency response is 20 Hz to 500 Hz.

Data acquisition was performed with the NI-DAQ Card from Nation Instruments, and EMGWorks signal acquisition and analysis software. Data analysis was performed in Matlab.

6.2 Determination of Electrode Placement

The placement of the surface EMG electrodes is influenced by both interface issues and signal processing issues. In terms of interface design, the device should be relatively unobtrusive, and comfortable for people to wear. From the signal processing perspective, the electrodes should be placed in an area that gives us enough information to classify the signals. After measurements were recorded from various electrode placements, it was decided to place two electrodes around the wrist. One electrode was positioned on the palm side of the arm, approximately an inch below the wrist joint, on the pinky-half of the hand. The other electrode was placed on center of the non-palm side of the arm, approximately an inch below the wrist joint. Ideally, for the best signal to noise ratio, the electrodes would be placed on the muscles at the base of the forearm, near the elbow. However, reasonably good signals can be picked up from the wrist area, and this allows for the electrodes to be held in a device that looks like a wristwatch, which people are comfortable wearing.

6.3 Gestures

The number and types of gestures to be classified is also influenced by both interface and signal processing issues. The group of finger gestures need to be easy and intuitive for a user to perform, and easy to distinguish using a classifier. Initially, two sets of gestures were decided upon to classify. The first set consists of 4 gestures: touching the tip of the thumb to the tip of the index, middle finger, ring finger, and pinky finger. The second set of gestures involves using your thumb and index finger as a mouse. Imagine having a tiny trackball located at the index finger joint closest to your knuckle. There are 4 gestures involved in this set: the motions associated with moving the imaginary trackball up, down, left, and right. Eventually, the task was limited to classifying the first set. Originally, finger taps were used. However, in an attempt to make the signal as stationary as possible, the gestures were changed from taps to maintained touching of the fingers to the thumb. The final set of gestures recorded from the subjects were:

- A two-second press of the tip of the index finger to the tip of the thumb.
- A two-second press of the tip of the middle finger to the tip of the thumb.
- A two-second press of the tip of the ring finger to the tip of the thumb.
- A two-second press of the tip of the pinky finger to the tip of the thumb.

6.4 Subjects

The EMG signal is effected greatly by fat and tissue content of a person. Therefore, it was necessary to retrieve data from a wide range of body types. The study had 21 subjects total: four female, nineteen male. Their body weights ranged from 140 to 200 lbs, and their ages ranged from 22 to 50.

6.5 Collection Method

A data collection package was created using WinBatch, and the EMG data collector. Using the WinBatch scripting language, a script was written that would relay to the test subject what gesture they would perform. The script would then start recording the EMG signal. The signals were sampled at 1024 samples/sec for 2 seconds per gesture. The signals were then segmented into examples of length 512 samples.

During a data collection session, a subject performed the 4 gestures 15 times each. Three subjects participated in one data collection session per day over the course of eight days in order to determine classification rate across sessions. 18 subjects participated in one data collection session. After the data collection, the data was divided into test and training data.

6.6 Preliminary Data Analysis

In order to determine whether the signals recorded have any distinguishable characteristics, different representations of the signal were examined.

When the raw surface EMG signal was examined, there were not many distinguishing features between the different gestures. The only noticeable feature is the relative amplitudes of the different signals (see figure 3.1). The middle finger has the most power, followed by the ring finger, then the pinky finger, and finally the index finger. However, the relative amplitude changes from data collection session to data collection session, and from person to person.

The Fourier transform of the EMG signals conformed to previous results. The majority of the frequency content is under 400 Hz, peaking at around 50 Hz to 200 Hz (see figure 3.2). Therefore, it is possible to

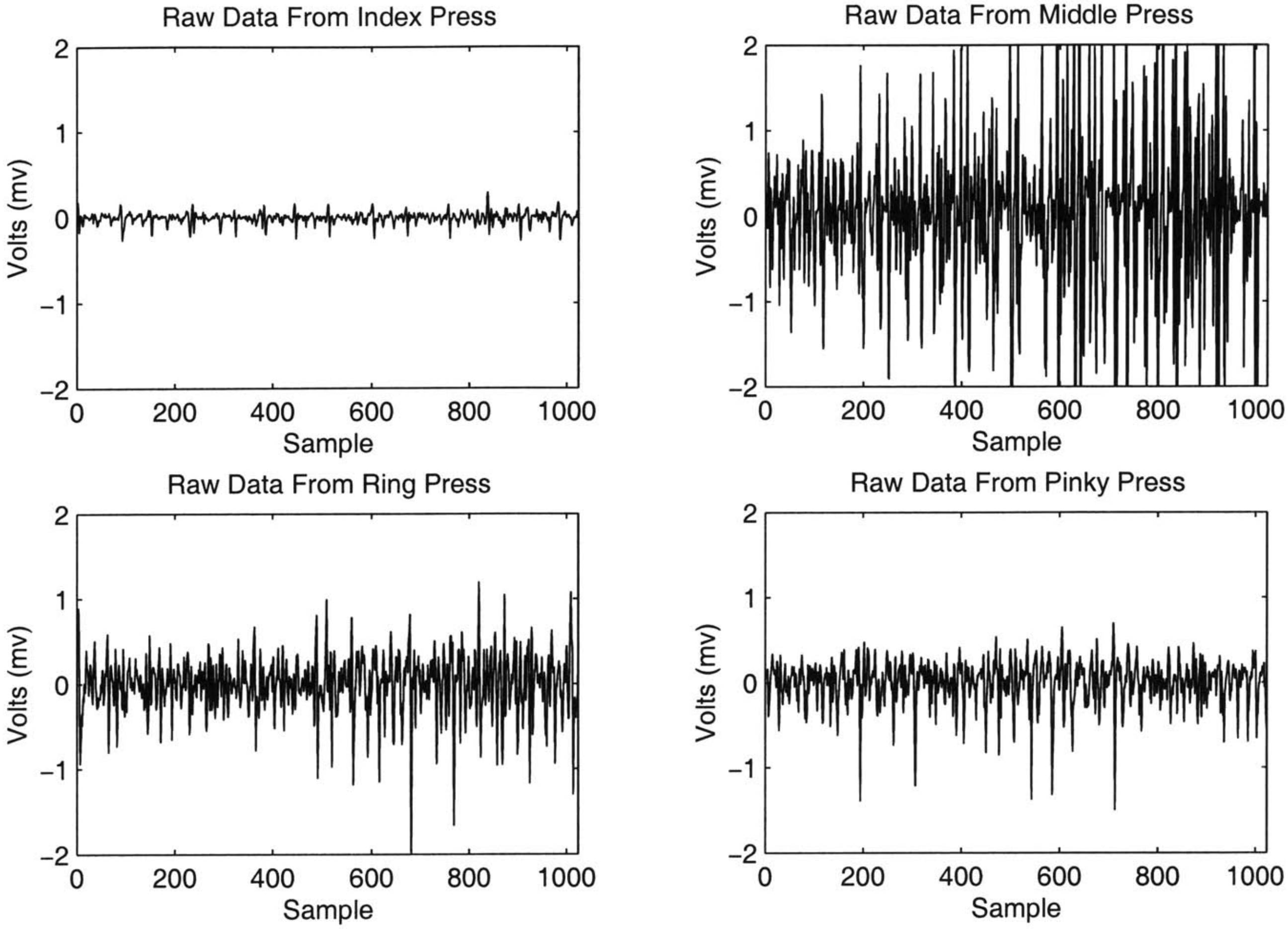

Figure 6-1: Graphs of the Raw Signal Voltages

sample the signals at 1024 samples/sec without loss of information.

The spectrograms of the signals did not show any distinguishing features bewteen the different gestures (see figure 3.3). The spectrogram parameters used were a 128-sample FFT, and a window of 64 samples, with an overlap of 54 samples.

6.7 Classification Experiments

Three major classification schemes were tested in the experiments: nearest neighbor, AR model classification, and IS classification. For AR classification, a fourth order AR model was used. For IS classification, the signals were Fourier transformed, and the resulting frequency coefficients were used as features for the IS

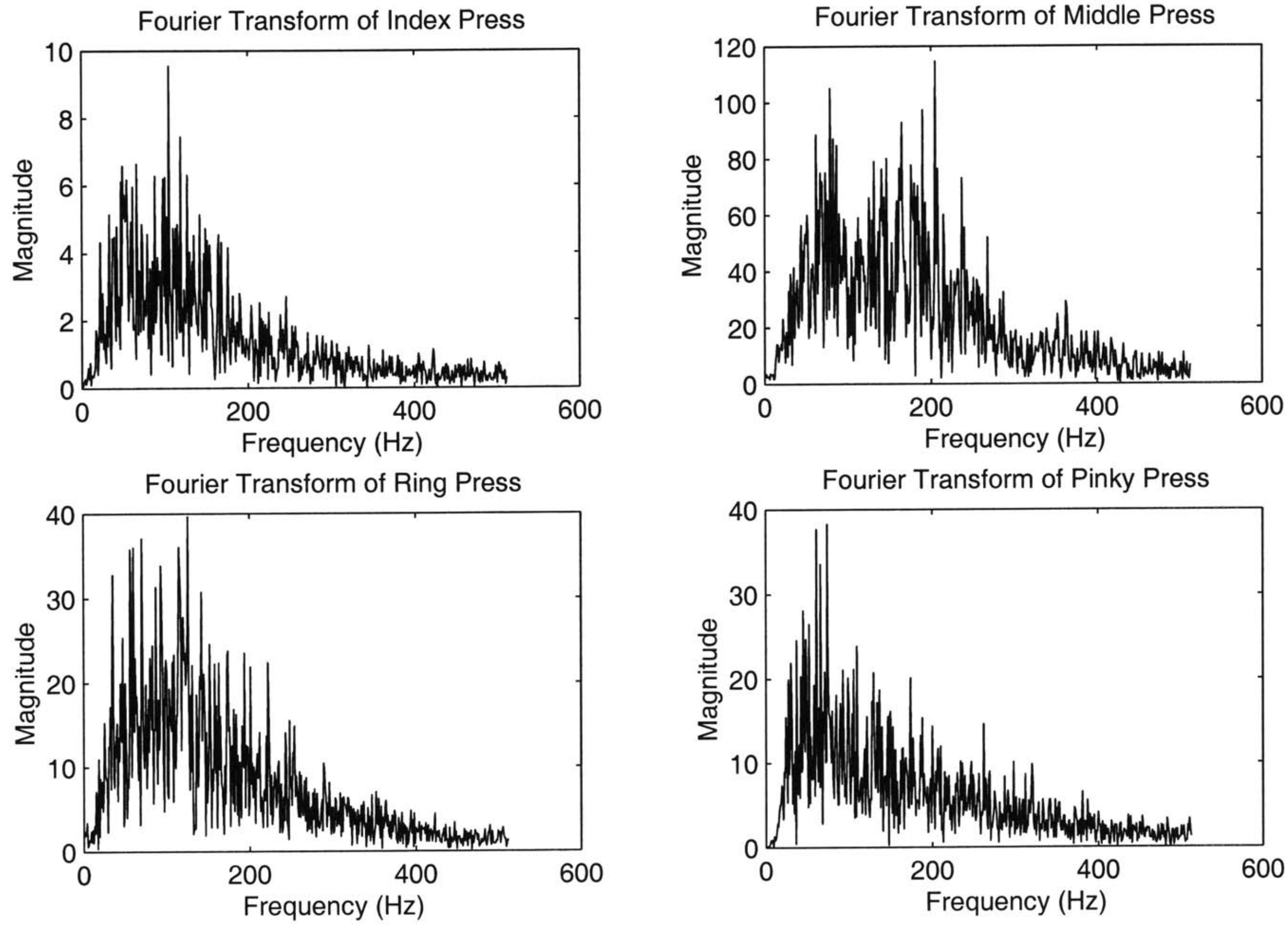

Figure 6-2: Graphs of the Fourier Transform of the Signals

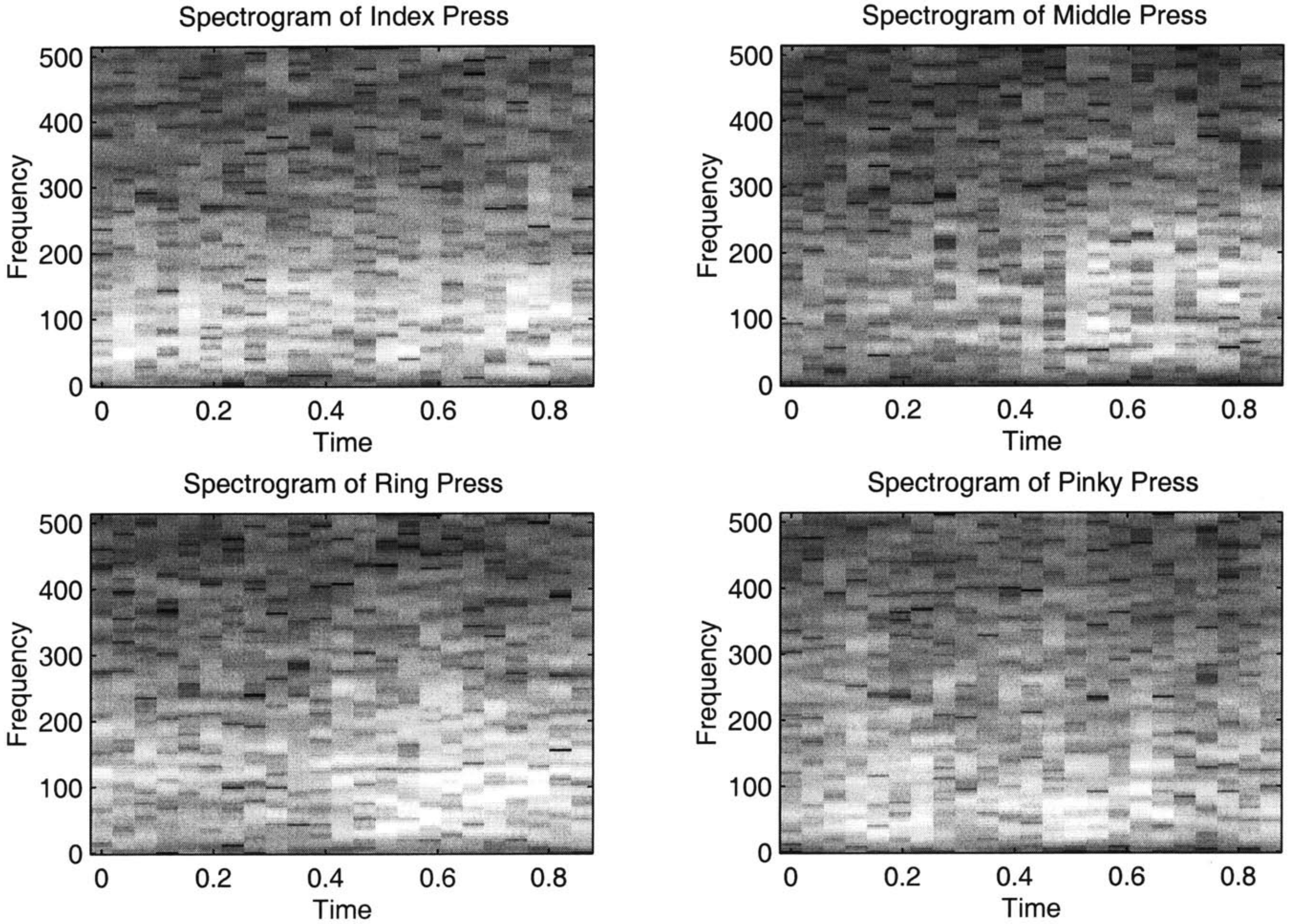

Figure 6-3: Graphs of the Spectrograms of the Signals

algorithm. The IS algorithm ran for 1000 iterations.

Preliminary classification tests were performed on other classification methods such as nearest neighbor classification with cepstral coefficients, and wavelet coefficients, Linear Discriminate Analysis with wavelet packet coefficients and iterative scaling with various wavelet transforms. However, these methods performed quite poorly on the data, and therefore they were not fully tested.

The main purpose of the classification experiments is to determine the classification rates of the various schemes. In addition, the classification experiments will give information on the necessity of incorporating training and retraining into the system. Therefore, five types of classification experiments were performed on the three schemes: Base Case, Pairwise Across People, Across People, Pairwise Across Sessions, Across Sessions.

6.7.1 Base Case Experiment

The purpose of the base case experiment is to test the general effectiveness of each classification technique. For this experiment, the testing and training data are taken from the same subject during the same collection session. The data from a collection session is divided into 28 training examples (seven examples per class) and 28 test examples (seven examples per class). Each example has a length of 512 samples.

6.7.2 Pairwise Across People

The purpose of the pairwise across people experiment is to measure the average similarity between subjects. In this experiment, data is taken from one subject to produce 98 (14 examples per class) training examples of length 512 samples. Then a classifier trained from the training examples is used to classify 98 examples taken from a different subject. This experiment is performed pairwise on all subjects.

6.7.3 Across People Experiment

The purpose of the across people experiment is to determine whether increasing the number of subjects used for training of a classifier will improve the classification rate. In this experiment, the data from one subject is used to create 98 test examples. Then five different classifiers are created. The first classifier is trained on the data from one subject, the second classifier is trained on the data from two subjects, and so on. Then the test data is classified by the five classifiers and classification rates are compared.

6.7.4 Pairwise Across Sessions Experiment

The purpose of the pairwise across sessions experiment is to measure the average similarity between different sessions collected from the same subject. In this experiment, data is taken from one session to produce 98 (14 examples per class) training examples of length 512 samples. The a classifier trained on this data is used to classify 98 examples taken from data collected during a different session with the same subject. The experiment is performed on every pair of sessions, and then averaged.

6.7.5 Across Sessions Experiment

The purpose of the across sessions experiment is to determine whether increasing the number of sessions used for training a classifier will improve the classification rate. In this experiment, the data from one collection session is used to create 98 test examples. Then five different classifiers are trained. The first classifier is trained on the data from one session, the second classifier is trained on the data from two sessions, and so on. Then the test data is classified by the five classifiers and classification rates are compared.

6.8 Adaptation Experiments

Experiments are also performed to compare various retraining techniques for the WAMI system. For the adaptation experiment, AR coefficients are trained and calculated from 98 training examples (14 per class) collected from one session, and then 98 test examples signals collected from a different session are classified. Then four signals (one for each class) are taken from the test session data, and used to update the AR coefficients, using five different methods:

- No Adaptation - The weights are not changed. This is used as the base case test.
- Completely Retrain - The AR coefficients are completely recalculated using the original training data and new examples.

- New Data Retrain - The AR coefficients are recalculated using only the new data given, and none of the original training data.
- Always Adapt - The AR coefficients are updated using the gradient descent learning rule.
- Conditional Adapt - The AR coefficients are updated using the gradient descent learning rule only if the classifier misclassifies the new examples.
- New Data Adapt - The AR coefficients are initially set by training on the new data for the first iteration, and then are adapted using gradient descent and the remaining new data.

Then the remaining test signals are classified, and another set of signals are taken from the test set and used to update the AR coefficients. This procedure is performed for 8 iterations and the results are plotted.

6.9 Sequence Decoding Experiments

The final set of experiments compare the performance of HMM sequence decoding of EMG signals and Doerschuk's probability curves for sequence decoding. Both the HMM model and Doerschuk's model are used to calculate the most probable state sequence for the EMG signal of a sequence of four different gestures. The results from both models are then compared to the actual state sequence and the error is calculated.

Chapter 7

Results

The results from the experiments show that EMG classification of finger gestures is possible. However, they also show that a good deal of research and fine tuning needs to be performed on the classifiers before they are accurate enough for the device.

7.1 Base Case Static Classification Experiments

The results for the base case experiments are summarized in figure B.1. As expected, time domain nearest neighbor performed the worst with an average classification rate of 40.14% (10.34%). The AR technique performed at a 73.35% (20.48%) classification rate, and the IS method had a similar classification rate of 77.75% (11.16%). Significance tests show that the success rates of the AR and IS methods are significantly larger than the nearest neighbor method with an alpha of .05. However, the AR and IS methods are not significantly different from each other. Further tests are needed to determine whether the IS method is significantly different than the AR method.

7.2 Across People Static Classification Experiments

The pairwise across people experiments show that there is very little similarity between the EMG signals of various people. The results are summarized in figures B.5 - B.7. Though certain groups of people are

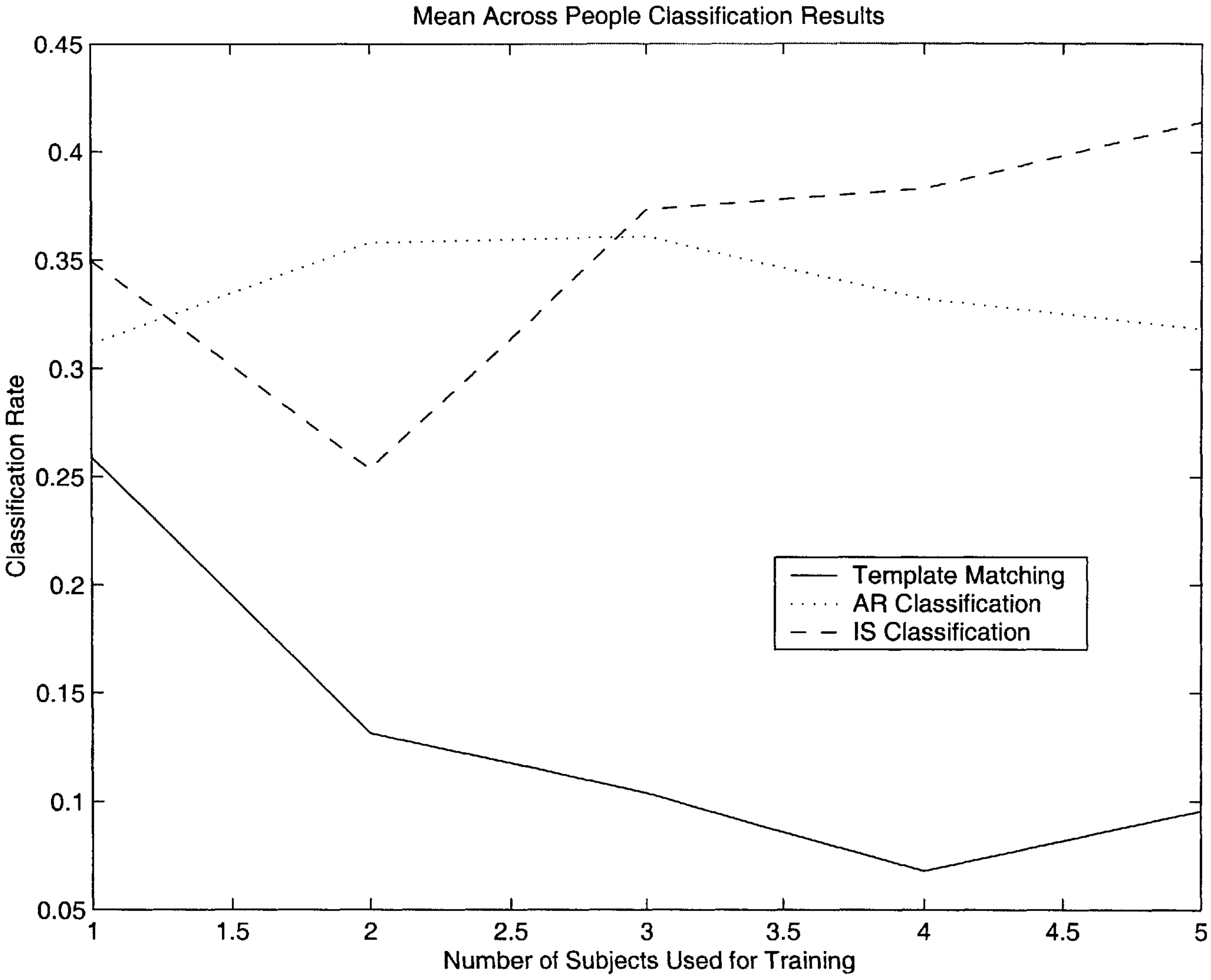

Figure 7-1: Classification Rate vs. Number of People represented in Training Data

similar to each other, on average the signals differ greatly from person to person. For nearest neighbor the classification rate was 25.71% (6.85%). For AR classification the rate was 33.20% (14.12%), and for IS classification the rate was 30.89% (13.13%). In addition, the across people experiments showed that adding more people to the training data does not improve classification rate as can be seen in figure 7-1. The results for the across people experiment are summarized in figures B.2 - B.4.

7.3 Across Days Static Classification Experiments

The results for the pairwise across days experiment are summarized in figures B.17 - B.25. They show a significant drop in classification performance in relation to the base case experiments. Nearest neighbor dropped from 40.14% to a classification rate of 37.60% (6.99%). AR classification dropped from 73.35% to 58.22% (15.11%), and IS classification dropped from 77.75% to 49.77% (16.49%). These results confirm that slight shifts in the position of the EMG recorder cause large changes in the recorded EMG signals. There are still similarities between the signals of two different sessions since the classifiers are still well over chance. However, the across sessions experiments showed that adding more sessions to the training data does not improve classification rate, as can be seen in Figure 7-2. The results of the across sessions experiments are summarized in figures B.8 - B.16.

7.4 Adaptation Experiments

The adaptation experiments are very promising. The gradient descent algorithm performs just as well as batch retraining on the new data only, and better than the batch case when trained on new and old data. This is a very good sign, in that you get all the speed and memory advantages of the gradient descent algorithm along with the performance of the batch algorithm. The results are summarized in Figure 7-3.

7.5 Sequence Decoding Experiments

The sequence decoding experiments show the HMM model outperforming the traditional probability model on every test case. The average error rate for the HMM model was 84.49%, while the error rate for the probability model was 72.85%. The HMM model is significantly different from the Doerschuk model with an alpha of .05. Graphs show some examples of the actual state sequence, compared to the HMM state sequence and the probability model state sequence. As you can see from the example graphs (Figures 7-4 7.5), the HMM state sequence is quicker to realize that a gesture begins and ends.

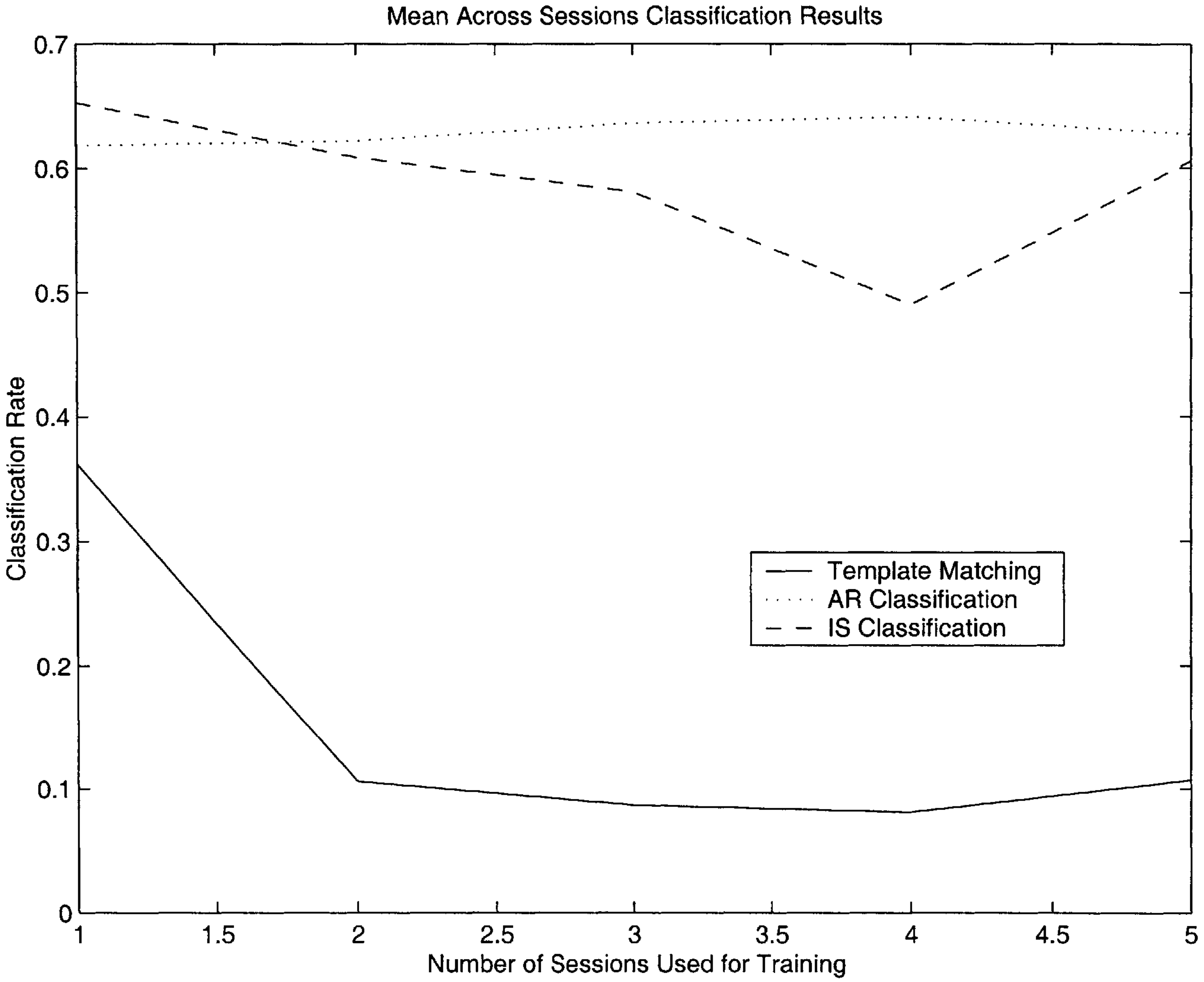

Figure 7-2: Classification Rate vs. Number of People represented in Training Data

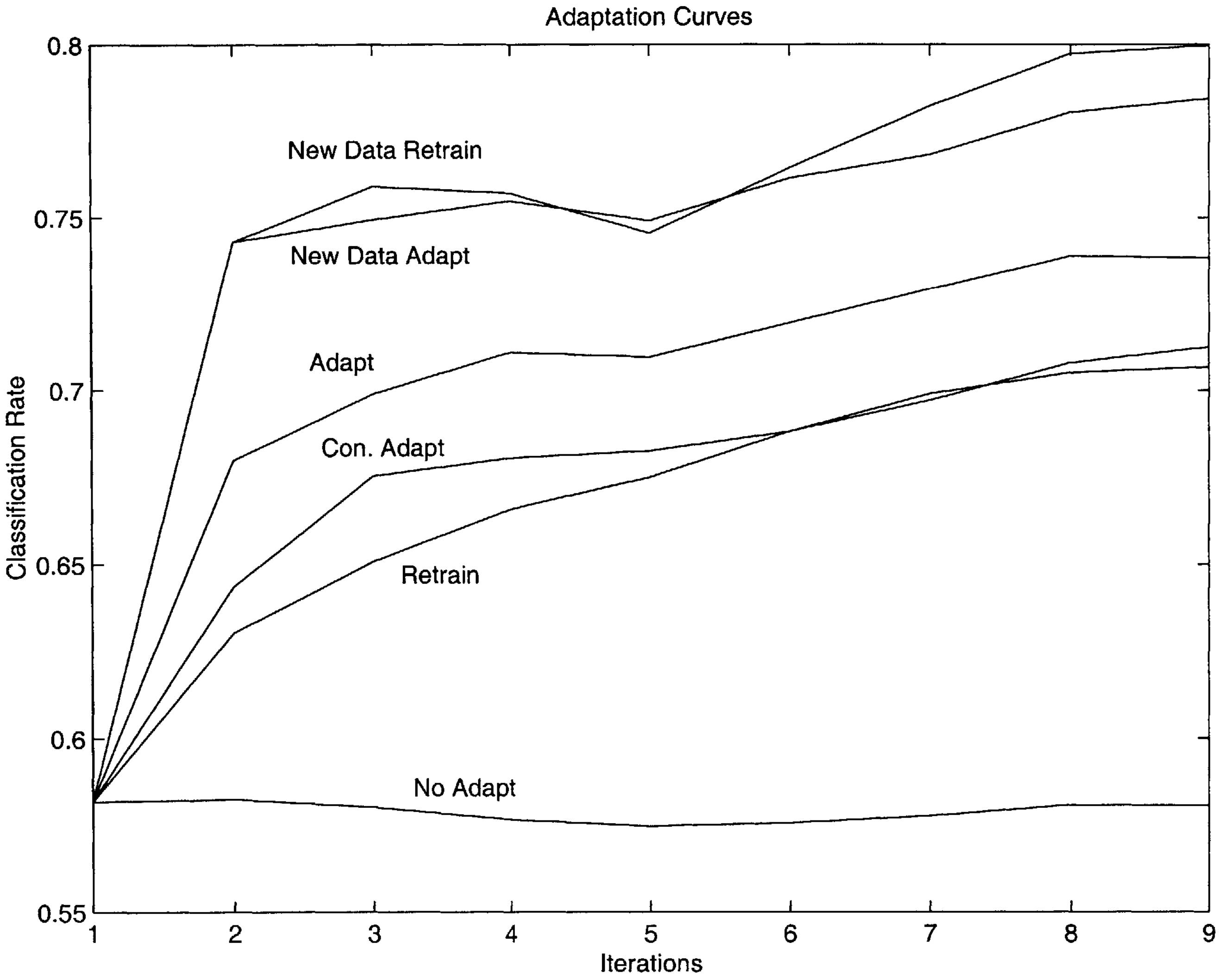

Figure 7-3: Performance of Various Adaptation Schemes

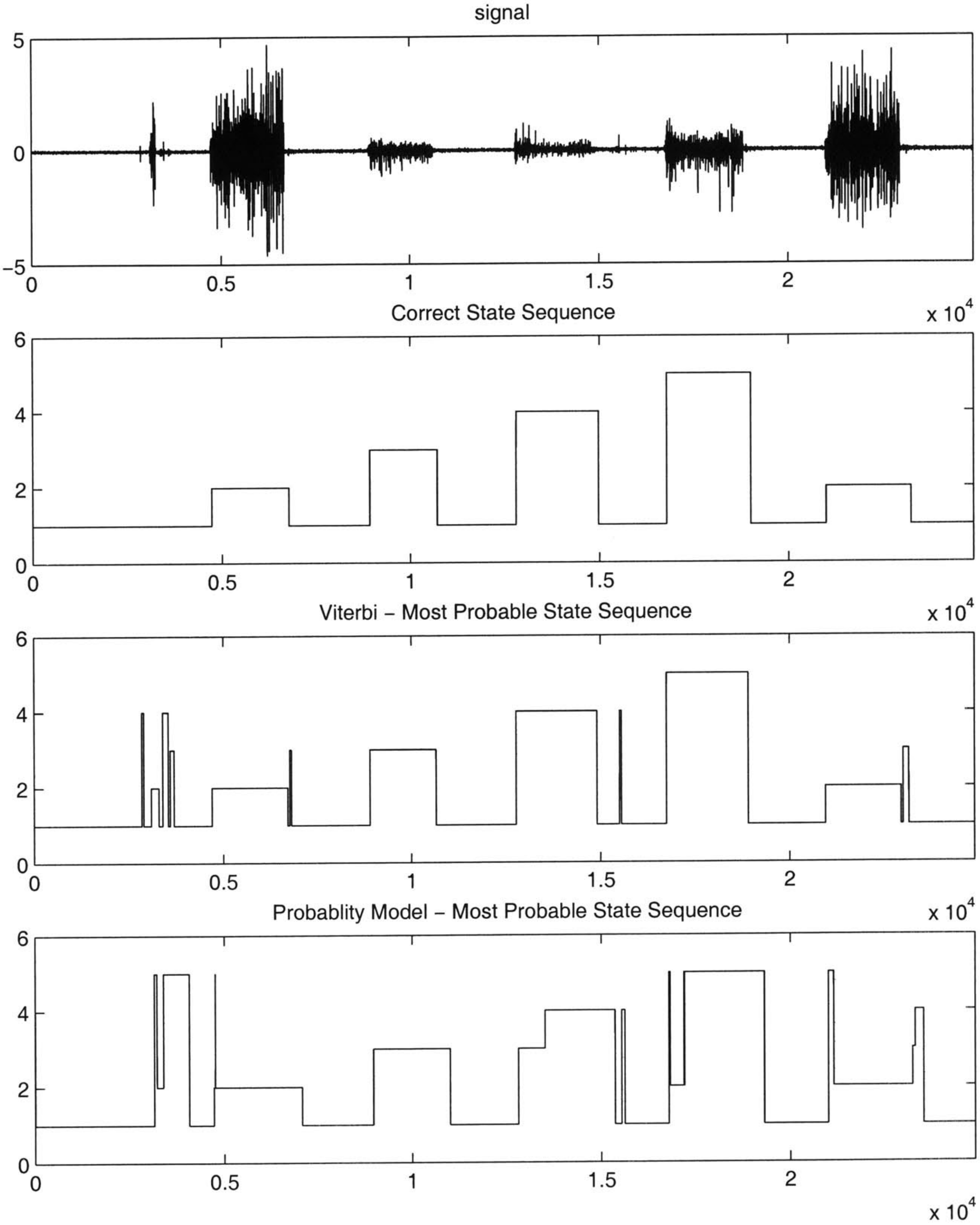

Figure 7-4: An example of sequence decoding with an HMM and a stateless model

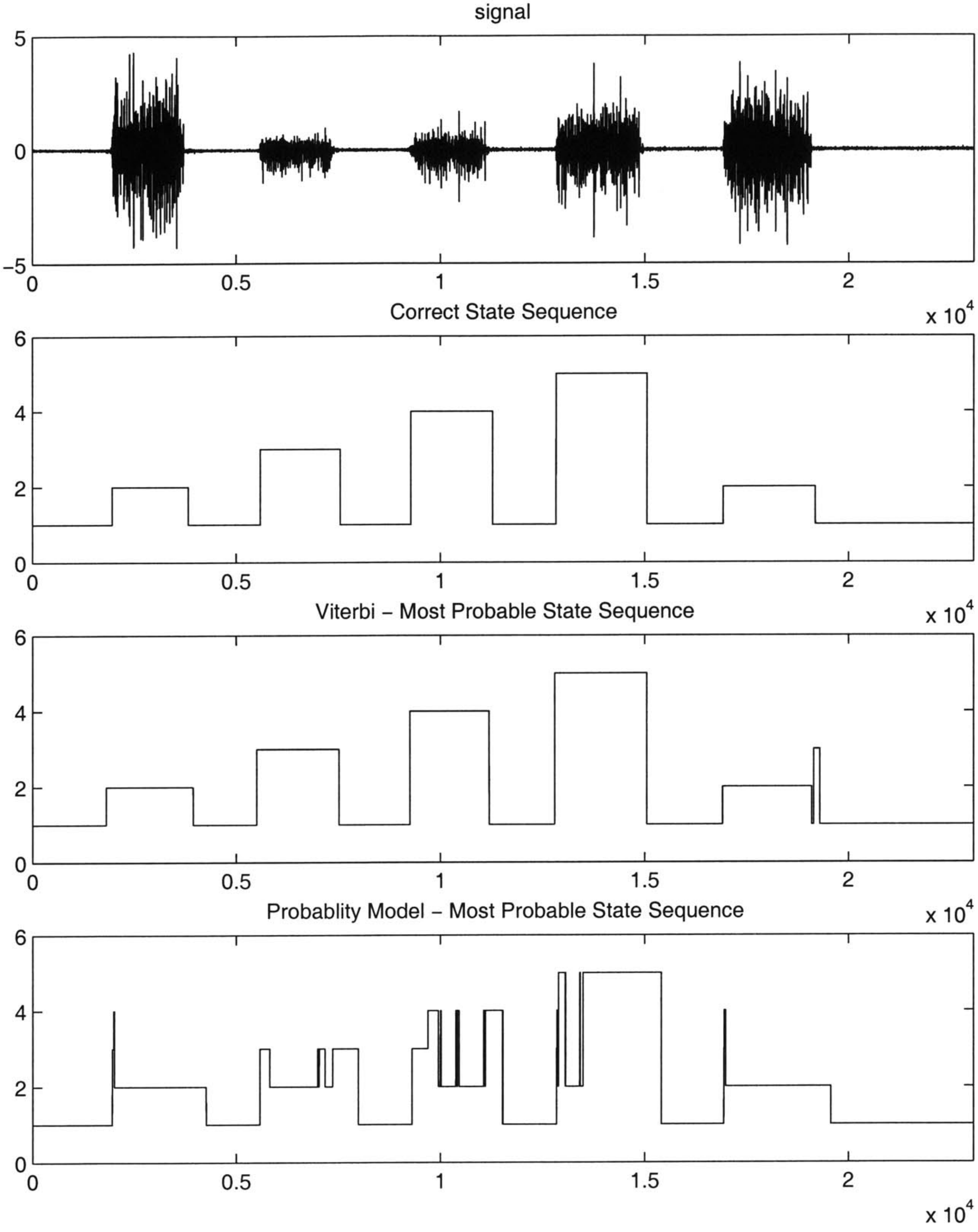

Figure 7-5: An example of sequence decoding with an HMM and a stateless model

Chapter 8

Discussion

8.1 Classification

The best classification results on the data were around 80%. This is significantly lower than the classification rates stated in other papers relating to EMG classification. The reason for this disparity is most likely due to a few factors. First, in all the previous attempts at EMG classification, large gestures and muscle groups were used for the experiments. This is sensible for the types of applications previous researchers were trying to implement, namely, control systems for amputees. In the control system case, you are dealing with the remaining muscles in a limb which are generally large muscles controlling large gestures. Our set of gestures are very fine in relation to these gestures, and therefore more difficult to distinguish. In addition, the EMG recorders are not in the optimal position to record muscle activity. The position of the electrodes is a good compromise between utility and signal strength, and as a result some signal information is lost. As stated in the experiments chapter, this thesis implemented and tested many of methods used in previous papers, and found them to do very poorly on the data. Therefore, the data and situation for these experiments are different than previous experiments.

Though an 80% classification rate is a good result, it is not good enough for the intended application. In order for the input device to be a truly successful product, it must have a near perfect classification rate. The future work section will discuss some possibilities for improving the classification rate.

8.2 Adaptation

The adaptation experiments show that adapting using the Widrow-Hoff algorithm performed as well as batch performance. Another interesting observation was that training on old and new data actually performs worse than training only on new data. Therefore, the optimal retraining system would be to throw away all previous data, calculate AR coefficients for each class using the first set of new data, and then adapt using the Widrow-Hoff algorithm and the remaining new data. The current scenerio for retraining the system from day to day would be for a user to perform a quick sequence of gesture when they put on the device in the morning, and then the device would use the gradient descent adaptation scheme to weight the coefficients

8.3 Sequence Decoding

The sequence decoding tests clearly show that an HMM model is better than a stateless model. The problem with a stateless model is that it believes that the entire sequence is one gesture, and therefore in order for the stateless model to change from one gesture to another, it needs enough evidence to overcome the evidence for the previous gesture. For example, notice in the example graphs the stateless model takes some extra time to transition from a gesture back to the rest state. In the HMM model, this does not occur, because it is not looking for the best state at a given instant, it is instead looking for the best state sequence. In addition, the majority of the mistakes made by the HMM are due to noise in the data as in figure 7-4 where the subject twitched and the HMM believed it to be a gesture. The only real issue is that the HMM model is more expensive to implement than the stateless probability model, and it is not known if the Viterbi algorithm will work in real time, especially with the computing power available on a portable system.

8.4 Future Work

There is a lot to be done in many area of the this project. First, in terms of classification, more can be explored with the IS algorithm. The feature set given to the algorithm was relatively simple, just the Fourier coefficients. It might be interesting to see what happens when different feature sets are given to the learning algorithm. In preliminary tests wavelet, wavelet packet and Gabor coefficients were used as features and they did not perform as well as the Fourier coefficients. However, other feature sets are still possible. Another improvement is to add cross-correlation information between the two channels to the IS method. For the two-channel IS implementation, the two channels were considered independent of each other, and therefore discarded some possibly important information. If inter-channel relationship information had been incorporated into the feature set, there might have been an increase in performance. A final improvement on the IS method would be to incorporate it into the HMM framework. One possible way of doing this would be to compute the Fourier coefficients on a sliding window of the data. Then at each step use the $P(y|x)$ probabilities from IS as "observations" at the HMM nodes.

Another method of improving classification rate is to use Error Correcting Codes (ECC). If errors consistently occur between pairs of gestures, they can be analyzed. Then, a code can be created based on distinguishable sequences of gestures. For example, a pinky tap and a ring tap might not be distinguishable from each other, and a index tap and a middle tap might not be distinguishable from each other, but a middle tap followed by a pinky tap could be distinguishable from a ring tap followed by an index tap. These error correcting gesture sequences maybe able to boost the classification rate to an acceptable level for the device.

Once the classification rate is up to the level needed for the device, the next logical step is to create a prototype. The purpose of the prototype would be to see how these methods work in a real world setting, and more importantly if they can be implemented in real-time with the computing power available in a portable device.

Appendix A

Transforms and Bases

Various transforms were used in conjunction with nearest neighbor and the IS machine learner. Transforming an EMG signal from the time domain to different basis can create new feature sets and tilings of the time frequency plane (See Figure A-1) that allow for better classification. The ideal transformation for classification would be a mapping of each gesture class to an orthogonal delta function. Even though this transform may not be found, other basis may still be helpful for classification.

The Fourier transform is probably the most well known transform. It projects the time domain onto a space spanned by sinusoids, also known as the frequency domain. The formal equation for the Fourier transform is:

$$X(j\omega) = \int_{-\infty}^{+\infty} x(t)e^{-j\omega t}dt$$

where $x(t)$ is the time domain signal and $X(j\omega)$ is the Fourier transform of $x(t)$

The benefit of looking at signals in the Fourier domain is that it allows for analysis of the frequency content of the signal. In addition, the Fourier transform is shift invariant, meaning that the transform of a time-shifted signal and the transform of the original signal look similar.

The wavelet transform projects a signal on to a basis whose functions are scaled and shifted versions of a prototype wavelet function. The formal equation for the wavelet transform is:

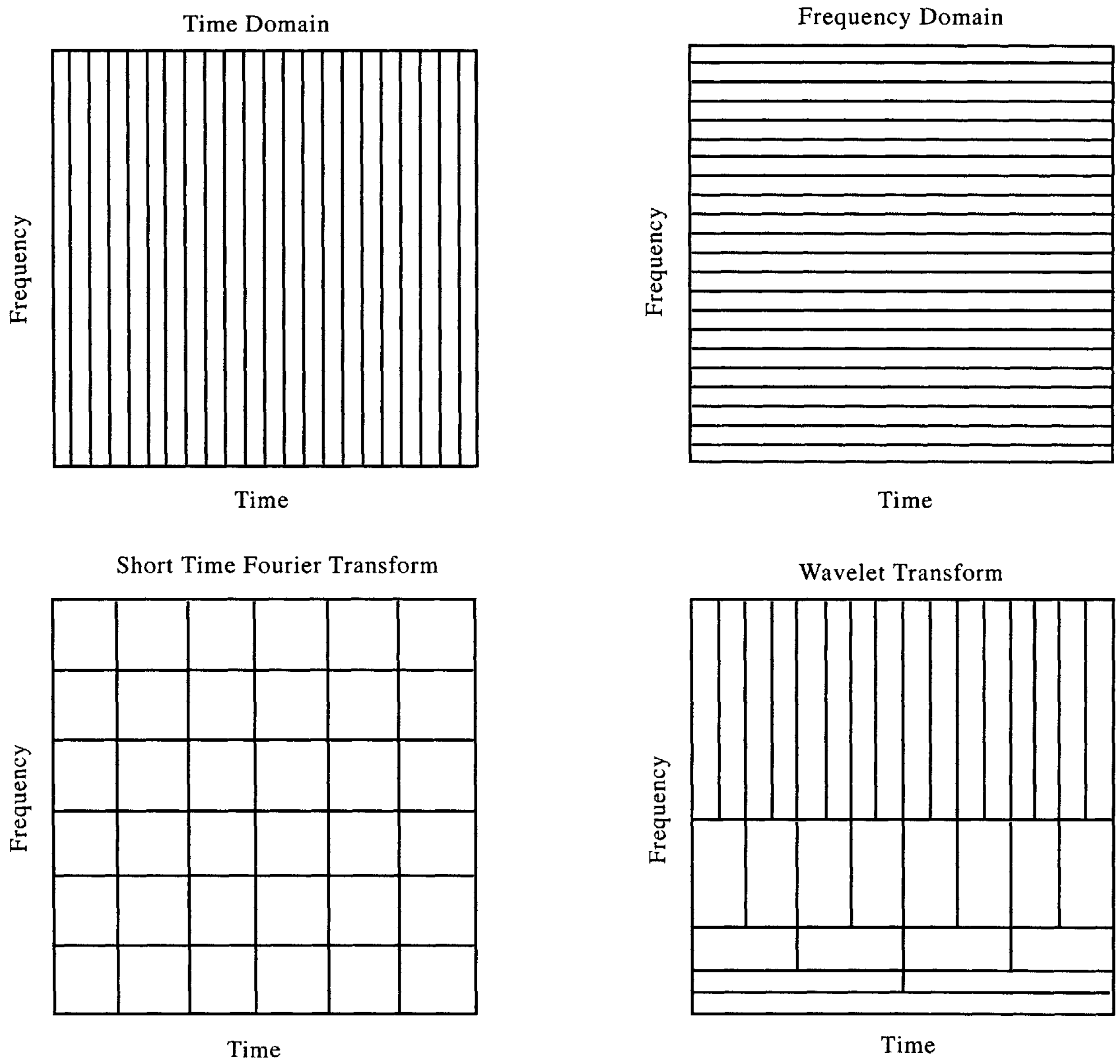

Figure A-1: How the different transforms divide the time-frequency plane

$$\gamma(s,\tau) = \int f(t)\Psi^*_{s,\tau}(t)dt,$$

where

* denotes complex conjugation.

This equation describes how $f(t)$ is decomposed into a set of basis functions $\Psi^*_{s,\tau}(t)$. The variables s and τ are new dimensions, scale and translation, after the wavelet transform. The basis functions of the wavelet transform are generated from a basic mother wavelet, by scaling and translation:

$$\Psi^*_{s,\tau}(t) = \frac{1}{\sqrt{s}}\Psi(\frac{t-\tau}{s}),$$

where

s is the scale factor,

τ is the translation factor and

$\frac{1}{\sqrt{s}}$ is for energy normalization.

Note that the wavelet basis functions are not specified. Wavelet theory gives a general framework where one can design one's own wavelets. See Figure A-2 for an example of the Haar wavelet.

The two benefits of the wavelet transform are its compact support and multiresolution capabilities. One of the major problems with the Fourier transform is that its basis functions are sinusoids and therefore infinite in length, which can cause complications when dealing with finite length signals. Wavelets, in general, have compact support, meaning they are finite in length, and therefore they have some advantages when modeling finite signals. The other benefit of a wavelet decomposition is that it analyzes a signal on varying resolutions in frequency and time. The Fourier transform has infinite resolution in frequency, and zero resolution in time, and the time domain has infinite resolution in time and zero in frequency, while the wavelet transform has a good mix. In fact, using a wavelet packet [15] allows you to specify where in the frequency domain you want more resolution and where in the time domain you want more resolution. A problem with the wavelet

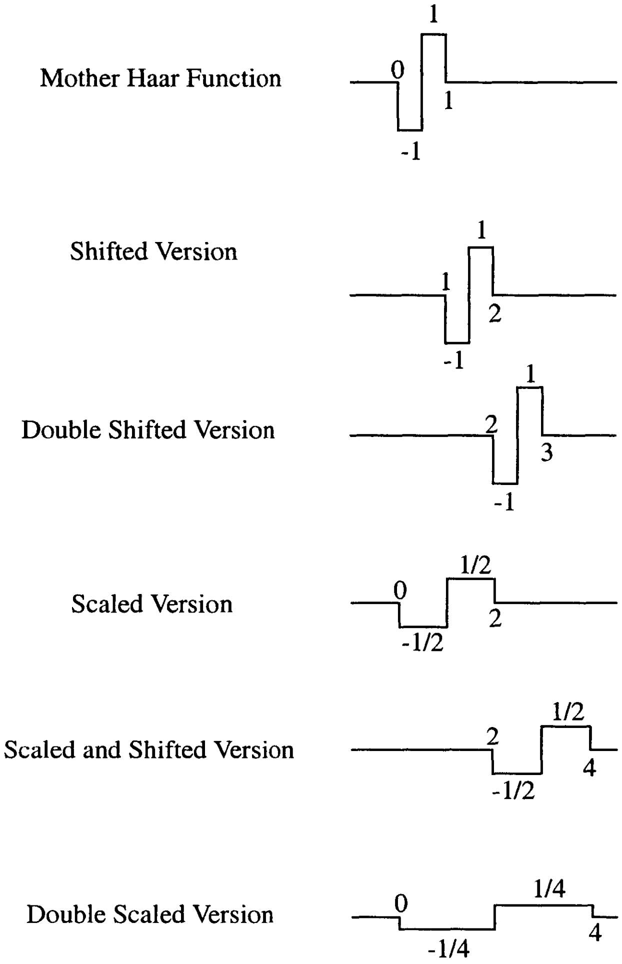

Figure A-2: The Haar mother wavelet and how the other wavelet functions are produced transform is that it is not time-invariant. The wavelet transform of a signal shifted one time-step could look completely different than the wavelet transform of the original signal.

Another transform used in our experiments was the short-time Fourier transform (STFT), also known as the Gabor transform. The STFT slides a windowing function along a signal, windowing portions of it, and then Fourier transforms each portion (See Figure A-3). The formal definition is:

$$X_n(j\omega) = \int_{-\infty}^{+\infty} w(nk - t)x(t)e^{-j\omega}$$

where

$X_n(j\omega)$ is the STFT of x(t),

$n = 1...length(x(t))/k$,

k is the overlap of the window,

and

$w(t)$ is the windowing function.

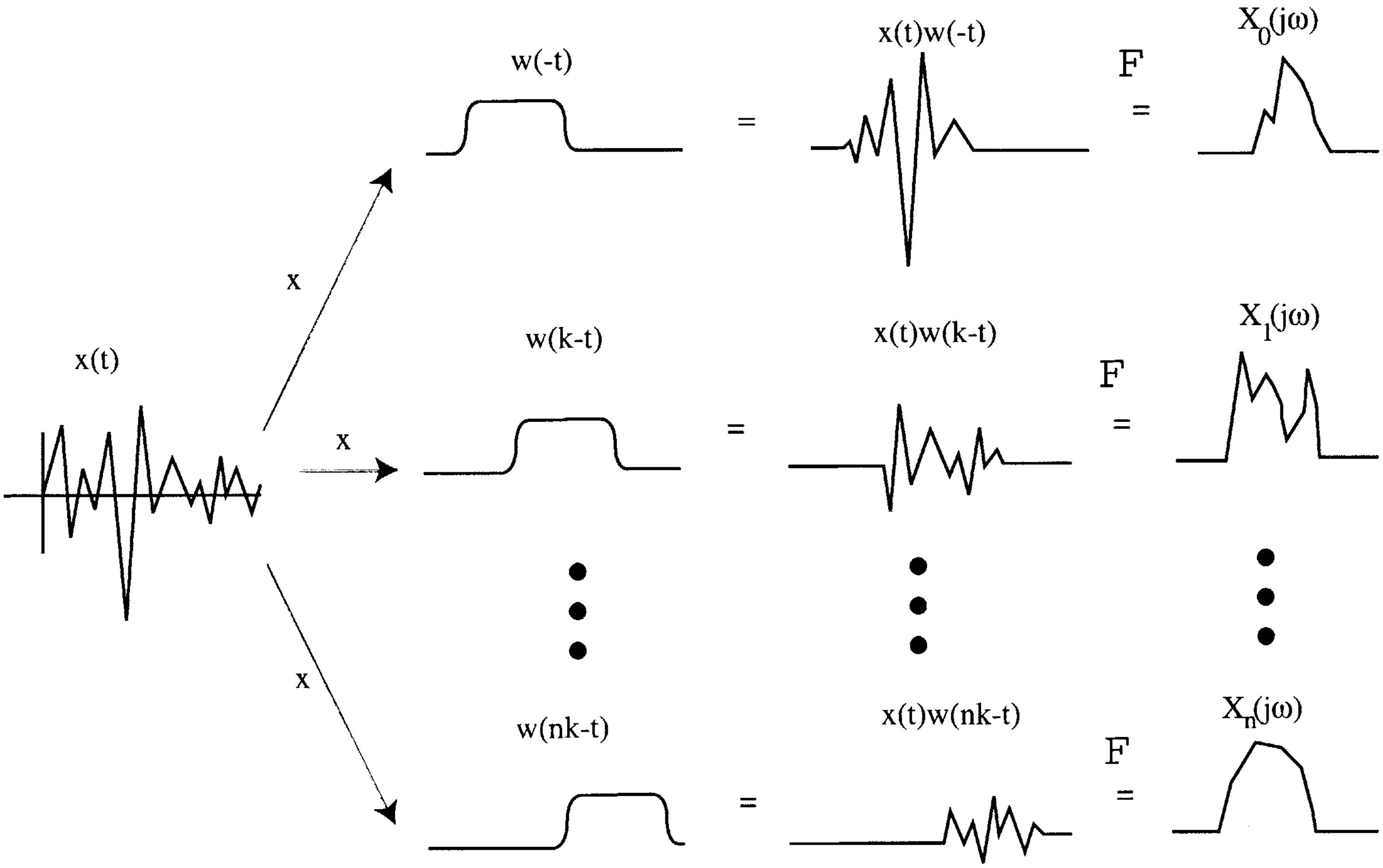

Figure A-3: The Steps of the Short Time Fourier Transform

The benefit of the STFT is that, like the wavelet transform, it allows for time-frequency analysis. In addition it is relatively shift-invariant. The STFT is also an overcomplete basis meaning that certain signals can be represented by two or more different combinations of the basis functions for the STFT.

The Cepstral transform is another transform used in our classification experiments. The real Cepstral transform is defined as:

$$\hat{x}[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} \log |X(j\omega)| e^{j\omega} d\omega,$$

where $\hat{x}[n]$ is the real Cepstrum transform of signal $x[n]$ and $X(j\omega)$ is the Fourier transform of signal $x[n]$.

In other words, the Cepstrum of a signal is the inverse Fourier transform of the natural log of the magnitude of the Fourier transform of a signal. The intuition behind why the Cepstral transform is useful is that it separates a source signal from a noise channel. A source signal going through a noise channel is a convolution of the signal and the channel. When you take the Fourier transform of the noisy signal you change the convolution to a multiplication. When you take the log, the multiplication becomes an addition, and when you inverse transform the signal, you get a signal that is a sum of the noisy channel and the original source signal instead of a convolution.

Appendix B

Experiment Data

Subject \ Classifier	Time	AR	GIS
S1	35.71	60.71	73.57
S2	53.57	85.71	88.57
S3	39.29	75.00	77.86
S4	46.43	78.57	54.29
S5	35.71	64.29	80.71
S6	42.86	96.43	90.00
S7	28.57	64.29	85.00
S8	53.57	92.86	71.43
S9	28.57	89.29	82.14
S10	14.29	60.71	77.14
S11	42.86	64.29	65.71
S12	39.29	75.00	75.00
S13	39.29	57.14	67.86
S14	46.43	100.0	91.43
S15	39.29	67.86	54.29
S16	57.14	89.29	90.00
S17	39.29	89.29	92.14
S18	53.57	75.00	71.43
S19	28.57	89.29	85.71
S20	32.14	78.57	72.86
S21	46.43	85.71	85.71
mean	**40.1367**	**73.3476**	**77.7548**

Table B.1: Base Case Classification Rates

Subject \ Number of training sessions	1 session	2 sessions	3 sessions	4 sessions	5 sessions
S1	16.07	8.93	8.93	4.91	8.21
S2	48.21	9.82	7.14	4.02	7.86
S3	16.07	7.14	8.33	7.59	8.21
S4	37.50	15.18	12.50	5.36	12.14
S5	17.86	11.61	8.93	6.25	8.57
S6	25.00	12.50	7.14	7.59	10.36
S7	26.79	16.07	8.33	2.68	6.79
S8	28.57	12.50	14.88	13.84	10.00
S9	16.07	25.00	11.90	4.46	10.00
S10	25.00	13.39	8.93	7.14	9.64
S11	25.00	17.86	5.36	6.25	7.50
S12	23.21	8.93	10.71	5.80	10.71
S13	17.86	14.29	5.36	7.14	8.57
S14	25.00	15.18	13.69	7.59	10.36
S15	32.14	16.96	19.05	8.48	12.50
S16	16.07	8.93	12.50	5.80	11.07
S17	46.43	8.93	17.26	7.59	9.29
S18	23.21	13.39	5.95	9.82	10.36

Table B.2: Across People Classification Rates Using Template Matching

Subject \ Number of training sessions	1 session	2 sessions	3 sessions	4 sessions	5 sessions
S1	12.50	7.14	5.36	5.36	5.36
S2	51.79	67.86	75.00	75.00	64.29
S3	25.00	25.00	30.36	35.71	26.79
S4	41.07	42.86	42.86	32.14	35.71
S5	28.57	30.36	25.00	25.00	25.00
S6	14.29	19.64	17.86	23.21	21.43
S7	33.93	17.86	28.57	26.79	28.57
S8	42.86	48.21	48.21	48.21	48.21
S9	48.21	57.14	57.14	35.71	37.50
S10	26.79	26.79	23.21	23.21	17.86
S11	30.36	32.14	46.43	46.43	48.21
S12	23.21	26.79	19.64	14.29	8.93
S13	37.50	33.93	25.00	41.07	33.93
S14	25.00	50.00	48.21	25.00	25.00
S15	39.29	35.71	26.79	23.21	16.07
S16	32.14	26.79	21.43	26.79	30.36
S17	07.14	48.21	25.00	7.14	25.00
S18	41.07	48.21	83.93	83.93	75.00

Table B.3: Across People Classification Rates Using AR Classification

Subject \ Number of training sessions	1 session	2 sessions	3 sessions	4 sessions	5 sessions
S1	26.07	25.00	30.36	32.50	28.93
S2	15.71	30.71	31.07	8.57	35.71
S3	43.57	17.50	30.36	39.29	29.29
S4	33.21	24.64	34.29	48.93	49.29
S5	35.36	14.64	45.36	46.43	17.86
S6	39.29	6.07	34.29	44.29	12.86
S7	36.79	15.00	46.07	32.14	5.00
S8	41.79	6.07	54.64	36.07	46.79
S9	39.64	24.64	54.29	43.21	36.07
S10	38.21	21.43	47.50	27.50	55.71
S11	26.43	12.14	40.00	31.07	61.43
S12	35.00	16.07	47.14	51.07	56.07
S13	35.00	23.21	52.86	46.43	56.79
S14	38.21	39.29	47.14	49.64	58.21
S15	35.36	51.07	10.36	50.36	49.29
S16	41.43	48.57	12.50	46.07	43.57
S17	33.21	50.71	29.29	29.64	50.71
S18	35.00	29.64	24.64	26.43	51.43

Table B.4: Across People Classification Rates Using IS Classification

	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S1	S15	S16	S17	S18
S1	0	30.77	17.95	12.50	29.17	29.79	28.12	30.67	29.23	22.34	27.03	09.52	32.35	17.95	24.24	17.14	25.81	16.07
S2	28.77	0	21.62	16.67	18.18	18.97	33.33	35.37	35.38	17.33	24.39	25.00	24.32	13.33	27.78	4074	1579	3269
S3	25.93	38.46	0	26.56	23.91	29.17	23.94	27.59	21.43	27.06	22.50	25.00	23.40	25.00	17.24	29.41	40.82	38.89
S4	19.05	18.75	10.00	0	17.50	20.00	28.12	27.06	38.60	27.59	40.00	5.26	29.55	30.30	28.57	28.57	26.76	22.64
S5	1905	1739	2292	19.74	0	17.24	34.29	25.88	33.33	28.05	28.57	27.27	25.00	18.00	26.67	26.83	32.00	20.00
S6	36.76	34.62	36.00	22.06	25.45	0	30.16	26.58	27.78	33.77	29.63	30.00	28.07	23.91	30.77	29.63	15.22	34.78
S7	24.32	17.86	25.00	28.00	25.49	20.93	0	23.17	15.58	21.52	29.63	20.00	16.00	21.57	36.00	38.30	40.48	29.63
S8	24.56	25.00	22.73	23.61	27.27	31.25	24.59	0	18.97	28.17	19.67	32.14	2.667	34.62	25.71	18.42	18.52	17.65
S9	31.94	35.56	20.83	16.95	29.79	28.33	25.45	35.37	0	22.54	27.45	16.00	35.19	20.69	35.29	23.64	28.57	31.82
S10	21.67	44.44	32.73	27.12	25.71	27.27	22.89	29.58	21.18	0	12.50	35.90	32.69	20.45	28.81	23.08	31.37	13.64
S11	32.79	24.24	26.19	16.67	18.18	21.43	30.43	26.83	12.68	20.00	0	28.57	16.07	22.58	37.50	25.00	20.59	13.73
S12	23.21	28.00	31.58	23.81	23.53	25.00	31.08	34.78	28.17	22.09	29.03	0	32.76	24.32	23.26	20.83	26.79	31.37
S13	13.51	20.83	34.48	31.67	22.92	22.22	23.73	33.33	24.19	27.17	36.84	35.00	0	37.21	24.00	25.00	12.07	25.86
S14	36.49	17.65	28.21	31.58	25.58	27.27	25.76	23.94	17.95	25.61	28.57	29.17	12.00	0	19.51	25.81	21.05	34.43
S15	22.89	24.00	8.82	22.39	30.00	30.65	18.84	28.57	29.09	30.49	23.08	7.14	19.05	20.83	0	28.12	29.55	25.00
S16	37.31	30.30	24.39	35.48	16.00	19.51	24.62	19.48	23.19	28.57	27.91	38.71	25.00	14.29	20.59	0	15.87	8.77
S17	26.98	20.51	23.53	32.81	18.37	37.21	26.03	31.40	20.51	32.00	23.81	29.27	33.33	39.13	20.00	29.03	0	29.17
S18	27.69	39.02	20.83	18.64	18.92	32.61	23.38	24.24	19.44	22.22	37.50	23.68	24.00	23.53	37.50	26.92	29.82	0

Table B.5: Pairwise Across People Classification Rates Using Template Matching: Test Subjects on the Vertical, Training Subject on the Horizontal

	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15	S16	S17	S18
S1	0	26.79	0	19.64	30.36	44.64	32.14	1.79	21.43	21.43	25.00	26.79	28.57	46.43	17.86	30.36	7.14	12.50
S2	30.36	0	51.79	25.00	0	44.64	37.50	48.21	55.36	23.21	44.64	25.00	48.21	51.79	35.71	46.43	55.36	35.71
S3	12.50	42.86	0	35.71	23.21	30.36	25.00	32.14	50.00	46.43	35.71	25.00	50.00	5.36	25.00	58.93	51.79	39.29
S4	21.43	26.79	44.64	0	32.14	5.36	32.14	41.07	30.36	39.29	14.29	21.43	33.93	28.57	39.29	37.50	26.79	39.29
S5	28.57	21.43	25.00	23.21	0	25.00	25.00	2500	2679	2143	2500	1250	4286	2500	32.14	25.00	25.00	25.00
S6	10.71	26.79	30.36	14.29	25.00	0	39.29	41.07	44.64	26.79	53.57	23.21	44.64	35.71	39.29	50.00	48.21	42.86
S7	33.93	44.64	26.79	25.00	21.43	44.64	0	32.14	50.00	25.00	28.57	25.00	44.64	14.29	58.93	46.43	50.00	37.50
S8	5.36	3.57	28.57	25.00	25.00	53.57	23.21	0	42.86	39.29	55.36	25.00	48.21	16.07	48.21	50.00	48.21	62.50
S9	1.79	66.07	30.36	25.00	28.57	32.14	25.00	30.36	0	25.00	51.79	25.00	57.14	48.21	42.86	48.21	32.14	55.36
S10	48.21	26.79	25.00	41.07	25.00	25.00	25.00	25.00	26.79	0	25.00	28.57	42.86	30.36	19.64	41.07	41.07	23.21
S11	26.79	37.50	48.21	30.36	39.29	57.14	23.21	37.50	44.64	42.86	0	21.43	44.64	53.57	28.57	62.50	44.64	39.29
S12	14.29	44.64	12.50	16.07	5.36	46.43	50.00	7.14	41.07	37.50	44.64	0	23.21	57.14	3.57	28.57	37.50	12.50
S13	44.64	357.1	28.57	46.43	23.21	25.00	25.00	25.00	37.50	50.00	25.00	21.43	0	44.64	33.93	25.00	26.79	37.50
S14	41.07	32.14	26.79	48.21	7.14	42.86	35.71	26.79	37.50	33.93	62.50	25.00	48.21	0	26.79	48.21	25.00	35.71
S15	33.93	23.21	16.07	35.71	14.29	17.86	44.64	33.93	30.36	23.21	3.57	10.71	39.29	10.71	0	25.00	37.50	60.71
S16	39.29	50.00	39.29	32.14	32.14	33.93	23.21	39.29	60.71	30.36	46.43	32.14	50.00	48.21	35.71	0	39.29	60.71
S17	14.29	53.57	48.21	1.79	12.50	48.21	44.64	25.00	50.00	26.79	39.29	26.79	42.86	7.14	33.93	28.57	0	28.57
S18	10.71	50.00	33.93	33.93	0	26.79	30.36	39.29	71.43	25.00	26.79	25.00	50.00	16.07	75.00	57.14	41.07	0

Table B.6: Pairwise Across People Classification Rates Using AR Classification: Test Subjects on the Vertical, Training Subject on the Horizontal

	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S1	S15	S16	S17	S18
S1	0	30.36	23.93	25.36	18.57	13.21	13.21	36.07	41.43	43.57	21.43	53.93	10.00	34.64	29.29	42.86	41.07	52.86
S2	18.21	0	18.57	16.79	23.93	13.93	23.93	44.29	43.21	45.71	26.79	33.93	3143	3143	5500	4357	4250	2964
S3	11.43	0	0	31.07	25.00	27.14	13.93	30.71	25.00	38.93	41.79	48.21	50.71	30.36	46.79	43.21	13.93	38.21
S4	30.36	15.36	44.64	0	36.07	42.50	16.79	43.57	22.50	53.21	33.93	26.79	48.93	32.50	43.93	31.43	24.64	32.86
S5	36.07	53.21	35.71	38.57	0	37.50	31.79	42.50	43.21	55.36	25.00	56.43	52.50	32.86	15.00	25.00	51.79	27.14
S6	28.21	3.93	41.43	27.14	13.57	0	53.21	30.71	48.21	47.50	21.79	38.21	4750	30.00	53.93	34.64	9.64	17.86
S7	1.79	28.21	43.57	43.57	4.29	21.07	0	52.86	27.14	28.21	27.86	42.50	25.00	32.14	30.36	1.43	30.71	27.14
S8	16.07	24.64	28.93	26.07	40.00	19.64	34.64	0	18.21	31.79	26.43	26.43	4.29	27.86	30.71	5.00	52.86	38.57
S9	37.86	50.00	32.50	31.43	22.14	12.14	28.21	26.07	0	43.21	33.21	28.21	27.50	8.21	17.86	52.86	29.29	25.71
S10	25.00	50.00	27.50	18.21	8.57	45.36	36.07	26.43	30.00	0	25.00	41.07	22.86	2.50	20.71	25.71	25.00	52.50
S11	38.21	15.00	37.86	42.86	11.43	46.07	50.36	20.36	26.43	28.93	0	8.57	32.86	44.29	40.36	19.64	51.43	38.57
S12	20.71	38.57	32.14	25.00	31.79	31.07	11.79	23.93	32.50	12.50	9.29	0	25.00	46.79	33.93	10.36	24.64	41.79
S13	33.57	35.71	46.07	28.93	5.36	19.64	2.86	22.14	17.50	14.29	45.36	27.86	0	14.64	53.57	13.93	42.50	44.64
S14	8.21	35.71	43.93	33.93	31.43	44.29	25.71	29.29	30.36	45.71	31.79	43.21	25.00	0	29.64	45.36	53.21	35.71
S15	25.36	34.29	39.29	22.14	7.14	35.71	25.00	36.79	36.79	27.86	2.14	41.07	49.64	44.29	0	48.21	30.00	43.57
S16	33.93	35.00	4.29	05.71	40.36	23.57	14.29	10.36	32.14	28.93	30.71	45.36	42.50	46.43	25.00	0	40.00	35.71
S17	28.93	2.86	39.29	21.43	9.64	5.71	21.07	54.64	35.00	27.50	27.50	25.00	55.71	29.64	48.93	28.57	0	47.50
S18	27.86	8.93	28.93	36.43	26.79	14.29	40.36	37.14	36.07	28.93	43.57	26.43	24.29	42.14	23.21	50.00	50.36	0

Table B.7: Pairwise Across People Classification Rates Using IS Classification: Test Subjects on the Vertical, Training Subject on the Horizontal

Test Session \ Number of Training Sessions	1 session	2 sessions	3 sessions	4 sessions	5 sessions
1	57.14	16.96	18.45	17.41	19.29
2	46.43	16.07	18.45	18.75	16.07
3	50.00	14.29	11.31	13.39	12.50
4	32.14	10.71	10.71	10.71	15.00
5	33.93	14.29	10.12	11.16	13.57
6	30.36	9.82	12.50	17.41	15.36
7	41.07	13.39	10.12	15.62	14.64
8	48.21	10.71	11.90	13.84	15.71
mean	**42.41**	**13.28**	**12.95**	**147.9**	**15.27**

Table B.8: Across Session Classification Using Template Matching (Brian's Data)

Test Session \ Number of Training Sessions	1 session	2 sessions	3 sessions	4 sessions	5 sessions
1	51.79	30.36	53.57	53.57	46.43
2	60.71	66.07	64.29	67.86	69.64
3	73.21	75.00	82.14	80.36	83.93
4	51.79	44.64	51.79	48.21	50.00
5	53.57	57.14	51.79	48.21	51.79
6	67.86	76.79	78.57	75.00	62.50
7	50.00	44.64	39.29	44.64	37.50
8	42.86	46.43	50.00	50.00	42.86
bf mean	**56.47**	**55.13**	**58.93**	**58.48**	**55.58**

Table B.9: Across Session Classification Using AR Classification (Brian's Data)

Test Session \ Number of Training Sessions	1 session	2 sessions	3 sessions	4 sessions	5 sessions
1	65.71	55.00	52.86	57.50	67.50
2	70.71	54.64	65.36	60.00	69.29
3	73.21	49.29	64.29	51.07	75.00
4	77.86	64.64	57.14	58.57	53.57
5	78.21	68.21	22.14	51.07	45.71
6	36.79	71.79	41.43	38.57	57.14
7	47.50	76.43	50.71	74.64	56.07
8	52.86	57.50	60.36	67.86	56.79
mean	**62.86**	**62.19**	**51.79**	**57.41**	**60.13**

Table B.10: Across Session Classification Using IS Classification (Brian's Data)

Test Session \ Number of Training Sessions	1 session	2 sessions	3 sessions	4 sessions	5 sessions
1	33.93	6.25	2.98	3.57	7.14
2	23.21	6.25	3.57	4.02	6.79
3	26.79	8.04	2.98	2.68	7.14
4	28.57	9.82	4.17	2.23	6.07
5	21.43	15.18	2.38	1.34	6.43
6	30.36	8.04	4.76	2.68	7.14
7	32.14	10.71	2.98	2.68	6.43
8	26.79	7.14	2.98	2.23	6.07
9	17.86	8.93	2.98	0	6.43
mean	**26.79**	**8.93**	**3.31**	**2.38**	**6.63**

Table B.11: Across Session Classification Using Template Matching (Bob's Data)

Test Session \ Number of Training Sessions	1 session	2 sessions	3 sessions	4 sessions	5 sessions
1	46.43	67.86	57.14	55.36	58.93
2	83.93	71.43	75.00	76.79	76.79
3	83.93	69.64	69.64	67.86	66.07
4	46.43	50.00	48.21	42.86	41.07
5	87.50	71.43	75.00	80.36	75.00
6	64.29	64.29	62.50	66.07	64.29
7	53.57	58.93	76.79	75.00	75.00
8	75.00	71.43	67.86	64.29	71.43
9	71.43	69.64	69.64	76.79	75.00
mean	**68.06**	**66.07**	**66.87**	**67.26**	**67.06**

Table B.12: Across Session Classification Using AR Classification (Bob's Data)

Test Session \ Number of Training Sessions	1 session	2 sessions	3 sessions	4 sessions	5 sessions
1	48.93	87.14	43.57	43.21	52.86
2	58.57	62.14	46.79	43.21	79.29
3	61.79	46.79	79.29	34.29	83.93
4	51.79	50.71	75.00	56.79	87.14
5	55.00	52.86	74.29	69.64	82.86
6	68.21	51.79	75.36	62.86	79.29
7	87.14	32.50	83.93	62.86	80.71
8	89.64	45.36	46.07	67.86	74.64
9	85.71	41.79	45.71	60.00	80.36
mean	**67.42**	**52.34**	**63.33**	**55.63**	**77.90**

Table B.13: Across Session Classification Using IS Classification (Bob's Data)

Test Session \ Number of Training Sessions	1 session	2 sessions	3 sessions	4 sessions	5 sessions
1	42.86	3.57	9.52	4.46	8.21
2	33.93	9.82	8.93	5.36	12.50
3	35.71	11.61	7.14	6.25	10.71
4	41.07	6.25	4.17	8.04	9.29
5	37.50	10.71	11.90	7.14	9.29
6	39.29	12.50	12.50	7.59	12.50
7	41.07	5.36	7.74	7.14	7.14
8	33.93	16.96	16.07	8.93	11.79
9	30.36	10.71	10.12	8.93	15.71
10	37.50	8.04	10.12	6.25	11.79
mean	**37.32**	**9.55**	**9.82**	**7.01**	**10.89**

Table B.14: Across Session Classification Using Template Matching (Peter's Data)

Test Session \ Number of Training Sessions	1 session	2 sessions	3 sessions	4 sessions	5 sessions
1	51.79	64.29	62.50	67.86	67.86
2	50.00	76.79	69.64	66.07	67.86
3	78.57	60.71	60.71	78.57	64.29
4	80.36	85.71	82.14	83.93	83.93
5	73.21	66.07	75.00	75.00	73.21
6	60.71	60.71	62.50	64.29	66.07
7	67.86	64.29	67.86	69.64	71.43
8	28.57	48.21	42.86	37.50	37.50
9	41.07	41.07	44.64	44.64	44.64
10	60.71	21.43	30.36	23.21	30.36
mean	**59.29**	**58.93**	**59.82**	**61.07**	**60.71**

Table B.15: Across Session Classification Using AR Classification (Peter's Data)

Test Session \ Number of Training Sessions	1 session	2 sessions	3 sessions	4 sessions	5 sessions
1	57.86	37.14	41.43	56.43	45.36
2	71.07	62.86	69.64	34.29	32.86
3	69.64	68.93	74.29	32.14	43.21
4	80.36	65.36	84.64	28.93	38.21
5	78.57	69.29	78.93	30.71	33.21
6	62.86	72.50	28.57	23.93	53.21
7	57.50	81.07	38.57	27.50	52.50
8	65.00	77.14	39.29	42.14	53.57
9	53.57	83.57	33.93	37.86	50.00
10	56.43	79.29	37.86	63.93	43.57
mean	**65.29**	**69.71**	**52.71**	**37.79**	**44.57**

Table B.16: Across Session Classification Using IS Classification (Peter's Data)

Test Session \ Training Session	1	2	3	4	5	6	7	8
1	0	58.93	57.14	39.29	41.07	33.93	42.86	55.36
2	51.79	0	46.43	46.43	35.71	25.00	50.00	44.64
3	64.29	50.00	0	42.86	4.107	42.86	48.21	44.64
4	46.43	60.71	42.86	0	32.14	41.07	48.21	32.14
5	48.21	50.00	44.64	32.14	0	33.93	32.14	44.64
6	35.71	42.86	35.71	39.29	21.43	0	32.14	30.36
7	42.86	51.79	44.64	41.07	41.07	32.14	0	42.86
8	48.21	39.29	48.21	35.71	41.07	30.36	35.71	0

Table B.17: Pairwise Across Session Classification Using Template Matching (Brian's Data)

Test Session \ Training Session	1	2	3	4	5	6	7	8
1	0	75.00	73.21	51.79	48.21	64.29	35.71	42.86
2	73.21	0	78.57	37.50	51.79	60.71	57.14	57.14
3	76.79	73.21	0	46.43	64.29	66.07	42.86	62.50
4	58.93	51.79	53.57	0	51.79	48.21	62.50	55.36
5	58.93	41.07	53.57	41.07	0	44.64	42.86	64.29
6	67.86	53.57	76.79	64.29	67.86	0	37.50	64.29
7	50.00	46.43	51.79	51.79	42.86	35.71	0	87.50
8	42.86	39.29	37.50	53.57	58.93	48.21	76.79	0

Table B.18: Pairwise Across Session Classification Using AR Classification (Brian's Data)

Test Session \ Training Session	1	2	3	4	5	6	7	8
1	0	56.43	63.21	50.71	54.29	22.86	24.64	75.00
2	73.21	0	38.57	54.64	51.07	65.71	32.14	55.00
3	60.71	33.57	0	51.43	37.14	63.93	60.36	52.14
4	60.36	39.64	65.71	0	44.64	60.36	53.21	62.14
5	54.29	47.50	65.36	66.43	0	44.29	68.93	45.36
6	67.14	52.50	59.29	50.00	55.71	0	47.50	58.21
7	67.14	71.43	38.57	66.07	51.79	63.57	0	31.43
8	51.07	65.00	53.21	50.36	21.79	33.21	65.00	0

Table B.19: Pairwise Across Session Classification Using IS Classification (Brian's Data)

Test Session \ Training Session	1	2	3	4	5	6	7	8	9
1	0	41.07	21.43	19.64	39.29	37.50	33.93	39.29	17.86
2	33.93	0	23.21	30.36	23.21	23.21	37.50	35.71	26.79
3	21.43	32.14	0	28.57	26.79	16.07	23.21	32.14	32.14
4	28.57	23.21	28.57	0	28.57	19.64	25.00	26.79	16.07
5	23.21	26.79	26.79	30.36	0	21.43	41.07	48.21	37.50
6	30.36	32.14	26.79	25.00	19.64	0	30.36	32.14	26.79
7	32.14	21.43	16.07	23.21	33.93	28.57	0	33.93	35.71
8	32.14	32.14	37.50	25.00	32.14	26.79	35.71	0	32.14
9	17.86	25.00	28.57	17.86	33.93	28.57	33.93	28.57	0

Table B.20: Pairwise Across Session Classification Using Template Matching (Bob's Data)

Test Session \ Training Session	1	2	3	4	5	6	7	8	9
1	0	75.00	46.43	57.14	76.79	51.79	64.29	75.00	50.00
2	75.00	0	5357	39.29	80.36	57.14	83.93	82.14	73.21
3	42.86	50.00	0	73.21	58.93	83.93	85.71	50.00	58.93
4	28.57	46.43	53.57	0	39.29	41.07	50.00	42.86	50.00
5	66.07	87.50	44.64	42.86	0	55.36	94.64	87.50	83.93
6	46.43	51.79	71.43	64.29	64.29	0	82.14	50.00	71.43
7	48.21	60.71	62.50	53.57	73.21	55.36	0	71.43	58.93
8	73.21	75.00	48.21	62.50	75.00	53.57	73.21	0	75.00
9	50.00	71.43	42.86	51.79	75.00	53.57	73.21	82.14	0

Table B.21: Pairwise Across Session Classification Using AR Classification (Bob's Data)

Test Session \ Training Session	1	2	3	4	5	6	7	8	9
1	0	62.50	54.64	26.79	26.43	21.43	39.64	56.43	82.14
2	35.00	0	58.57	48.57	38.21	63.57	47.14	48.21	53.57
3	44.29	80.36	0	42.50	35.71	80.71	41.43	58.93	72.86
4	62.14	18.57	59.64	0	45.36	62.86	41.79	75.36	45.36
5	7.50	68.21	50.36	46.07	0	24.29	56.79	55.00	61.07
6	50.00	83.93	32.86	78.57	26.07	0	66.79	60.71	64.64
7	51.79	77.14	31.79	88.57	45.36	45.00	0	68.57	43.57
8	48.21	33.57	41.43	57.86	30.71	57.50	40.00	0	45.36
9	70.36	39.64	21.43	46.43	30.36	41.79	68.93	84.29	0

Table B.22: Pairwise Across Session Classification Using IS Classification (Bob's Data)

Test Session \ Training Session	1	2	3	4	5	6	7	8	9	10
1	0	44.64	42.86	57.14	51.79	42.86	37.50	35.71	26.79	30.36
2	32.14	0	46.43	35.71	35.71	23.21	33.93	26.79	28.57	42.86
3	37.50	44.64	0	41.07	41.07	28.57	35.71	41.07	33.93	39.29
4	50.00	48.21	41.07	0	60.71	42.86	41.07	42.86	46.43	44.64
5	39.29	37.50	33.93	46.43	0	35.71	41.07	35.71	30.36	30.36
6	37.50	26.79	33.93	37.50	30.36	0	39.29	33.93	26.79	46.43
7	35.71	41.07	37.50	41.07	35.71	32.14	0	32.14	39.29	32.14
8	30.36	37.50	39.29	51.79	33.93	33.93	39.29	0	37.50	33.93
9	25.00	30.36	35.71	37.50	50.00	33.93	33.93	32.14	0	33.93
10	37.50	28.57	39.29	37.50	33.93	48.21	33.93	37.50	32.14	0

Table B.23: Pairwise Across Session Classification Using Template Matching (Peter's Data)

Test Session \ Training Session	1	2	3	4	5	6	7	8	9	10
1	0	76.79	57.14	64.29	71.43	64.29	58.93	51.79	64.29	69.64
2	83.93	0	50.00	64.29	85.71	87.50	69.64	44.64	64.29	75.00
3	50.00	53.57	0	78.57	73.21	48.21	37.50	67.86	33.93	44.64
4	58.93	67.86	80.36	0	83.93	76.79	64.29	75.00	55.36	51.79
5	62.50	73.21	66.07	69.64	0	71.43	57.14	64.29	53.57	57.14
6	57.14	66.07	55.36	60.71	67.86	0	62.50	46.43	46.43	50.00
7	62.50	73.21	46.43	60.71	71.43	67.86	0	17.86	64.29	60.71
8	37.50	30.36	62.50	58.93	55.36	46.43	25.00	0	25.00	28.57
9	42.86	42.86	46.43	44.64	41.07	41.07	48.21	50.00	0	41.07
10	67.86	60.71	12.50	39.29	44.64	60.71	67.86	10.71	37.50	0

Table B.24: Pairwise Across Session Classification Using AR Classification (Peter's Data)

Test Session \ Training Session	1	2	3	4	5	6	7	8	9	10
1	0	76.43	34.64	74.29	82.50	41.79	42.86	32.14	39.29	37.14
2	61.07	0	53.93	73.57	39.64	37.86	48.57	23.93	39.64	51.43
3	70.00	63.93	0	37.14	37.86	63.57	28.93	55.00	37.14	55.00
4	73.93	59.29	22.50	0	58.57	43.21	52.14	71.43	42.86	30.36
5	50.36	72.14	62.14	57.14	0	45.71	43.93	25.36	48.21	35.00
6	39.29	40.00	58.21	42.14	43.21	0	68.21	72.86	52.14	39.29
7	59.64	18.21	37.14	80.00	52.50	23.93	0	66.79	44.29	51.79
8	28.21	67.14	75.00	66.79	34.64	32.50	74.64	0	38.57	42.86
9	53.93	66.43	57.14	75.36	30.36	29.29	45.00	40.71	0	33.57
10	79.29	40.00	74.29	80.71	35.71	51.07	29.64	46.79	26.07	0

Table B.25: Pairwise Across Session Classification Using IS Classification (Peter's Data)

Trial Number \ Decoding Method	HMM	Classical
1	94.77	84.27
2	78.37	67.00
3	96.76	77.15
4	74.73	59.03
5	88.13	79.13
6	70.45	70.06
7	78.40	67.53
8	87.79	82.23
9	90.32	75.74
10	86.57	63.95
11	83.09	75.28

Table B.26: Sequence Decoding Classification Rates

Bibliography

[1] Bishop, C. (1998). *Neural Networks for Pattern Recognition.* Clarendon Press.

[2] Berger, A., Pietra, S., & Pietra, V. (1996). A maximum entropy approach to natrual language processing. *Computational Linguistics*, 22(1), 39-71.

[3] Chen, S., Donoho, D., & Saunders, M. (1998). Atomic decomposition by basis pursuit. *Siam J Sci Comput*, 20(1), 33-61.

[4] Mallat, S., & Zhang, Z. (1993). Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41(12), 3397-3415.

[5] Coifman, R., & Wickerhauser, M. (1992). Entropy-based algorithms for best basis selection. *IEEE Transactions on Information Theory*, 39(2), 713-718.

[6] Englehart, K., Hudgins, B., Parker, P., & Stevenson, M. (1999). Classification of the myoelectric signal using time-frequency based representations. In *Med. Eng. and Phys. on Intelligent Data Analysis in Electromyography and Electroneurography.*

[7] Graupe, D., & Cline, W. (1975). Functional separation of EMG signals via ARMA identification methods for prosthesis control purposes. *IEEE Trans. Syst. Man Cybern.*, vol. 5, 252-259.

[8] Doerschuk, P., Gustafson, D., & Willsky, A. (1983). Upper extremity limb function discrimination using EMG signal analysis. *IEEE Trans. Bio. Eng.*, 30, 18-29.

[9] Hannaford, B., & Lehman, S. (1986). Short time fourier analysis of the electromyogram: fast movements and constant contraction. *IEEE Trans. Bio. Eng.*, 33, 1173-1181.

[10] Kelly, M., Parker, P., & Scott, R. (1990). The application of neural networks to myoelectric signal analysis: a preliminary study. *IEEE Trans. Bio. Eng.*, 37, 221-230.

[11] Kang, W., Shiu, J., Cheng, C., Lai, J., Tsao, H., & Kuo, T.(1995). The Application of Cepstral Coefficients and Maximum Likelihood Method in EMG Pattern Recognition. *IEEE Trans. Bio. Eng.*, 42, 777-785.

[12] Karlsson, S., Yu, J., & Akay, M. (1999). Enhancement of spectral analysis of myoelectric signals during static contractions using wavelet methods. *IEEE Trans. Bio. Eng.*, 46, 670-684.

[13] Fang, J., Agarwal, G., & Shahani, B. (1999). Decomposition of multiunit electromyographic signals. *IEEE Trans. Bio. Eng.*, 46, 685-697.

[14] Crouse, M., Nowak, R., & Baraniuk, R. (1998). Wavelet-based statistical signal processing using hidden markov models. *IEEE Transactions on Signal Processing*, 46, 886-902.

[15] Strang, G., & Nguyen, T. (1997). *Wavelets and Filter Banks*, Wellesley-Cambridge Press.

[16] Oppenheim, A., & Schafer, R. (1989). *Discrete-Time Signal Processing*, Prentice Hall.

[17] Csiszar, I., & Tusnady, G. (1984). Information geometry and alternating minimization procedures. *Statistics and Decisions, Supplement Issue*, 1, 205-237.

[18] Della Pietra, S., Della Pietra, V., & Lafferty, J. (1997) Inducing features of random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5, 179-190.

[19] Blum, A. (1997). Empirical support for winnow and weighted-majority based algorithms: results on a calendar scheduling domain. *Machine Learning*, 26, 5-23.

[20] Schapire,R .E., & Singer, Y. (1999). Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 37(3),1-40.

[21] Freund,Y., Schapire, R. E., Singer, Y., & Warmuth, M. (1997). Using and conbining predictors that specialize. In *Proceedings of the Twenty-Ninth Annual ACM Symposium on the THeory of Computing*, 334-343.

[22] Widrow, B., & Hoff, M. E. (1960). Adaptive switching circuits. *1960 IRE WESCON Convention Record*, 96-104. Reprinted in *Neurocomputing* (MIT Press, 1988).

[23] Porat, B. (1997). *A course in Digital Signal Processing.* Wiley.

[24] Baum,L. E., & Eagon, J. A. (1967). An inequality with applications to statistical estimation for probabilistic functions of markov processes and to a model for ecology. *Bulletin of the American Mathematical Society*, 73, 360-363.