EMG PATTERN CLASSIFICATION BASED ON AR MODELING

by

Zeynep Erim

B.S. in EE, Technical University of Istanbul

Submitted to the Biomedical Engineering Institute

in partial fulfillment of the requirements for

the degree of

Master of Science

in

Biomedical Engineering

Bogazici University

1986

ACKNOWLEDGEMENTS

I wish to express my gratitude to Dr. Bülent Sankur, my thesis supervisor, whose courage to explore a new topic with me made this study possible. I am greatful to Drs Yusuf P. Tan and Ömer Cerid whose support at various stages of my thesis work has been beyond all expectances. I would like to thank Prof. Dr. Necmi Tanyolaç for serving on the committee and his constant support during the three years I've spent at the Institute. Last but not least I wish to thank Semih Bingöl for helpful discussions throughout my studies.

MYOELECTRIC PATTERN CLASSIFICATION BASED ON AR MODELING

Myoelectric control of powered prostheses is a field of rehabilitation engineering that has received wide attention in the recent decades. In this thesis a historical perspective of the studies in the field is given. The physiological properties of muscles are reviewed. The linear models, algorithms for identifying model parameters, and basic pattern recognition considerations are outlined. A scheme to extract motion information from a single surface EMG channel is discussed. The results obtained in performance tests are given. Suggestions for future research topics are made. Major computer programs used are given in the appendix.

AR MODELİNE DAYALI MİYOELEKTRİK ÖRÜNTÜ SINIFLANDIRMA

Miyoelektrik protez denetimi rehabilitasyon mühendisliğinin son yıllarda yaygın ilgi toplamış bir dalıdır. Bu tezde konu üzerindeki çalışmaların zaman içindeki gelişimleri kısaca gözden geçirilmiştir. Kasların temel fizyolojik özellikleri incelenmiştir. Doğrusal modeller, model parametrelerini tanılama algoritmaları ve örüntü tanımadaki ana kavramlar ele alınmıştır. Tek bir yüzey EMG kanalından hareket bilgisi elde edecek bir düzenek tartışılmıştır. Başarım testlerinde elde edilen sonuçlar verilmiştir. Gelecekte söz konusu olabilecek araştırma konuları hakkında öneriler getirilmiştir. Geliştirilen belli başlı bilgisayar izlenceleri eklenmiştir.

TABLE OF CONTENTS

LIST OF FIGURES AND TABLES

I. INTRODUCTION

The physiological mechanisms that provide man with locomotion, postural control and manipulative skills are extremely complex. Coordinated muscoloskeletal behavior is based on a highly intgrated network of multilevel controllers, actuators, and information receptors connected by an enormous array of afferent and efferent information paths. The biological subsystems are also incredibly well suited to the performance of their individual functions under a wide variety of configurations and circumstances.

The major drawbacks in attempts to provide man-made duplicates for people who have lost their natural limbs are the lack of complete insight into the behavior of the neuromuscular system and the lack of implant material fully compatible with biological tissue. It would be desirable for an ideal artificial arm to directly connect to the skeleton, receive multichannel command information from higher centers as well as providing them with feedback information. With the current knowledge and technological capabilities the design of such a system is not possible. On one hand implant materials that the body would not reject for continual use is hard to find. On the other, interpreting the command information carried by the nerves and, furthermore, presenting feedback information to them requires much deeper

understanding of the neuromuscular system than is currently available.

1.1 Myoelectric Control of Powered Prosthesis

It is clear that man-made substitues cannot exactly replicate the behavior of the natural limb and will have to work on a simpler scheme. In studies to develop the artificial arms that are more complicated and advanced than the conventional prosthesis with straps, it is usual to use motors at points of the prosthesis where the natural limb has joints. The more chalenging point is the extraction of signals to control these motors.

In myoelectrically controlled prosthetic systems, the electrical systems that are intentionaly generated in the patient's muscles are employed as the source of the command information. This approach is based on the fact that an arm amputee can produce concious selective muscular tensions and flexions at the some sites although his physiological control system may be impaired [RNS1]. With training, the amputee can gain quite precise control over these muscles and, consequently, the electrical signals at certain electrode locations. With appropriate analysis of these signals it is possible to derive information regarding the arm motion the patient wishes to perform. Thus the motors of the prosthesis

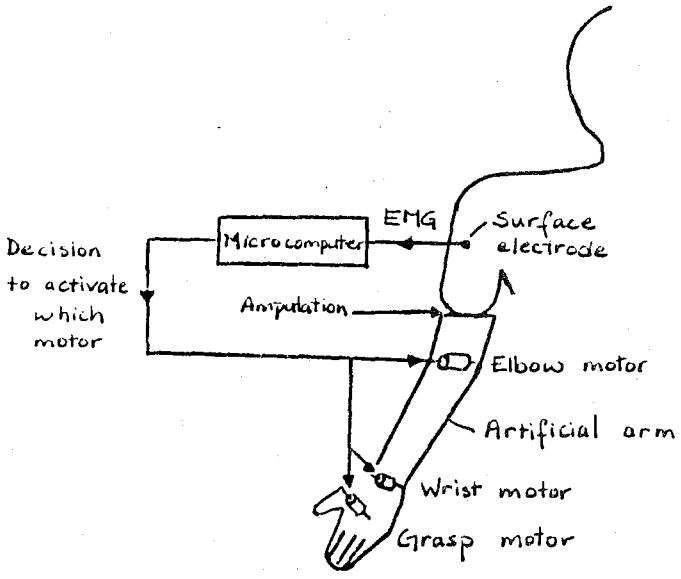

Figure 1.1 Block diagram of a myoelectric prosthesis [DG4]

can be activated in the necessary directions. A simple block diagram of a basic myoelectric prothesis system is given in Fig. 1.1

1.2 Historical Background

Since the arousal of interest in myoelectric control in 1940's, approaches with varying degrees of complexity have been proposed to tackle the problem of information processing. Most of the earlier research on the field has considered single motion control of the bang-bang type. In the experiments with myoelectrically controlled hand, Reiter used rectified signals from two muscle groups [UJ1]. The electromyogram (EMG) signals from finger flexors

and extansors were detected and compared to determine which one was stronger. A relay was activated to make the prosthesis move in the direction dictated by the stronger signal. The same principal is still used for single-functional prostheses. More sophisticated signal proceesing techniques have made it possible to control the rate of function as well.

In order to obtain multifunctional control, Dorcas and Scott divided the root mean square value of the EMG into discrete levels [WJ11]. The patient would produce different levels of contraction at the single site. The varying level of the generated EMG would correspond to differents motions. This method brings too much mental effort on the patient and the performance is restricted by the ability of the patient to consistently reproduce discrete levels of contraction.

An alternative to the preceeding approaches is to utilize the "phantom" limb sensation. Almost all amputees retain a cortical image of the lost limb [RNS11]. By performing imaginary movements of this limb they can cause the activation of the remaining parts of the muscles that would normally have been recruited if the limb had not been amputated. This activation is also reflected in the EMG which displays differences in pattern with varying limb functions. The use of this EMG as the information source requires the minimum mental effort to the patient.

Among the researchers using the phantom limb sensation, Wirta employs the mapping of many electrode sites [RW1]. The EMG data recorded at 10 electrodes sites at each of which the EMG is strongly correlated with a single limb function, are analyzed by a computer. This analysis gives the coefficients of the linear discriminant functions to be used in classifying unknown motions. Lyman also used the spatial distribution of the EMG signal power but applied the nearest neighbor decision rule in classifying unknown signals [UJ1]. Swedish researchers incorporated a microprocessor in their multi-electrode system in order to achieve proportional control [UJ1]. Like Wirta, they used linear discriminant functions in classifying motions. In a separate training session, the coefficients of the discriminant functions were calculated by analyzing the EMG recordings at various sites while performing different motions. Later the linear functions specified by these coefficients were used to classify a new set of EMG recordings as corresponding to a certain motion.

Saridis et.al. employed EMG signal variance and zero-crossings as parameters containing discriminatory information about the motion performed [GSi,SL1]. These features were chosen out of a possible feature space containing the variance, the second and fourth moments, the absolute value of the third and fifth moments, zero-crossings, autocorrelations, and power spectra of EMG signals measured at two different locations and their cross-correlation. The

feature selection criteria were that the tested variable should have large between-class scatter and small within-class scatter. Saridis et.al. used both linear discriminat functions and maximum likelihood decisions procedures at different studies. They also proposed a method to decompose combined motions into two or three simple motions and to assign a speed to them.

Graupe et. al. maintained that with appropriate signal processing, sufficient information could be extracted from a single EMG channel to control a multifunctional prosthesis [DG1,DG3,DG4]. In contrast to all the preceding approaches, they claimed that cross-talk between muscle groups was an important tool in motion discrimination. They tried to maximize this cross-talk, whereas other researhers tried to record the electrical activity of a single muscle group and treated the interference from others as noise. Their approach was based on the fact that different muscle groups have different temporal "signatures." The EMG signal recorded at a site with high muscle cross-talk represents the spatial integration of the electrical activities of the muscles in the vicinity. Since muscles have different signatures, the EMG recording will exhibit different characteristics according to the muscles that are activated. In order to model the EMG, Graupe first chose a (2,2) autoregressive moving average model and later, a third order autoregressive model. In motion classification they used two different methods. Both methods involved finding an average

representative coefficient set for each motion during the training session. In one method the coefficients of the test signal were estimated. The estimated coefficients were compared to reference coefficients obtained during the training. If the distance between the estimate and the reference closest to it was within a range formerly specified for that motion, the test signal was classified as belonging to the motion represented by the closest reference (see Chapter III concerning the concept of "closeness".) In the other method, instead of calculating the coefficients of the test signal, the test signal was filtered with the coefficient set of each motion. The filter giving the lowest prediction error was found. If the error was within a prespecified limit, the motion represented by the filter was chosen.

Doerschuk et.al. improved the system proposed by Graaupe by incorperating four EMG channels and their cross-correlations in their system [PD1]. They used the sequential version of the Bayes decision rule.

In this thesis, a scheme is proposed to classify EMG signals as having arisen from one of a finite number of motions. Some physiological background and the fundamentals of the information processing techniques are given before the explanation of the classification scheme. Explanation of the scheme and the results obtained in its evaluation are followed by conclusions and suggestions for future studies.

1.3 Thesis Outline

In Chapter II, fundamentals of the physiology of muscles are given as a background for the chapters to follow. Basic properties of muscles and their electrical activity are summarized.

Chapter III consists of the signal processing techniques commonly used in EMG analysis and pattern classification. Linear time series models, methods in estimating parameters of linear models, and basic concepts in pattern reconition are discussed.

In Chapter IV, a system to discriminate between three degrees of freedom is proposed. The experimental set-up, methodology, and signal processing techniques employed are discussed. The results obtained in evaluation tests are summarized.

In chapter 5 conclusions are drawn and suggestions for future studies are made.

The appendix contains major programs that have been developed.

II. PHYSILOLOGICAL BACKGROUND

The results obtained in studies on the behavior of the contracting muscle depend greatly on the type of the electrode used, electrode location within or on the muscle, electrode direction with respect to the muscle fiber, and the condition of the muscle at the specific time of investigation. This fact, as well as the varying properties of the different muscles studied, mostly accounts for the contradicting findings reached by different authors as those reviewed in [CD1] and [GA1]. However, certain conclusions about muscle phsiology and statistics are widely accepted by researchers of the field. Thase will be briefly summarized here to give an insight into the properties of muscles which are the main source of information in the motion classification scheme. Detailed information on muscle physiology and mathematics can be found in references [JB1,CD1,EK1,GA1,EK1,ES1].

2.1 Muscle Configuration

The human motor system is organized as a hierarchical control system from motor cortex to actuating muscles [GA1]. The most peripheral subunit of the motor

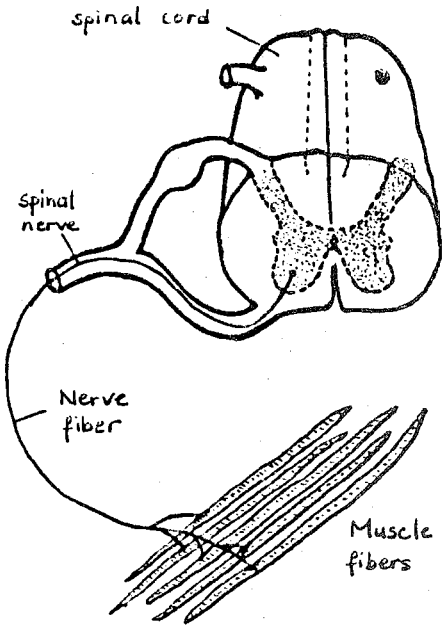

Figure 2.1 The motor unit [JB1)

control consists of the spinal motoneuron,the muscle fibers and various receptors providing feedback information to the central nervous system.

The muscle is made up of groups of muscle fibers which fire, or are depolarized, synchronously within the group but asynchronously with other groups. The simultaneously functioning group formed by the muscle fibers and the motoneuron innervating them is known as the motor unit. The motor unit is schematically depicted in Fig.2.1. It should be noted that the fibers belonging to a motor unit i.e., fibers that are supplied by the same nerve are not exclusively grouped together, as the illustration might suggest, but are intermingled with each other.

2.2 Muscle Fiber Action Potential

When the motoneuron passes down the input from the central nervous system,i.e, the action potential to the muscle fibers, the postsynaptic membrane of the muscle fiber is depolarized. The depolarization is propagated in both directions along the fiber, accompanied by an electromagnetic field in the vicinity of the muscle fibers. The waveform of the potential generated is known as the muscle fiber action potential.

The muscle fiber action potentials can be recorded with indwelling needle electrodes. Their amplitude depends on the diameter of the muscle fiber, the distance between the activated muscle fiber and the recording site, and the filtering properties of the electrode. The duration of the action potential is inversely proportional to the conduction velocity of the muscle fiber. The shape will depend on the filtering effect of the tissue between the muscle fiber and the recording electrode.

2.3 Motor Unit Action Potential

The action potentials of the muscle fibers in a motor unit overlap in time. The resultant signal recorded by the electrode usually comprises a spatial and temporal integration of the contributions of individual muscle fibers.

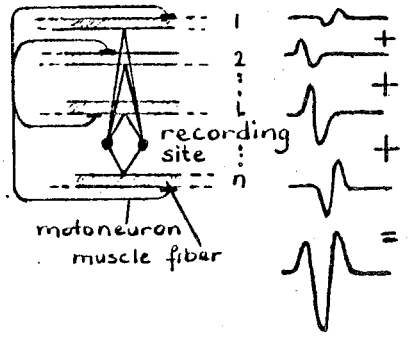

Figure 2.2 The generation of the MUAP [CD1]

This signal is known as the motor unit action potential (MUAP) and its generation is schematically shown in Fig.2.2.

The shape and amplitude of the MUAP are dependent on the geometric arrangement of the active muscle fibers with respect to the electrode site and the above mentioned factors which effect the individual muscle fiber action potentials.

The MUAP is followed by a twitch of the muscle fibers of the motor unit. In order to sustain the muscle contraction,the muscle fibers must be repeatedly activated. The resultant signal is known as the MUAP train (MUAPT). The MUAPT will remain constant as long as there are no changes in the factors effecting the MUAP's.

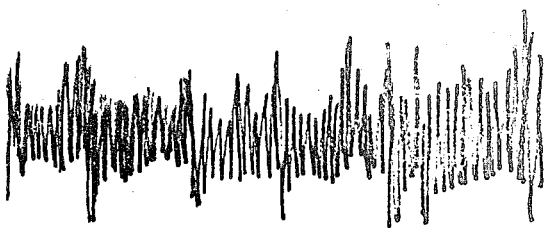

Figure 2.3 A typical EMG recording

2.4 The Electromyogram

During a muscle contraction it is very seldom observed that only the muscle fibers of a single motor unit be activated. Normally many motor units fire at the same time to make up a muscle contraction. Since the muscle fibers belonging to different motor units are intermingled, it is very difficult to discriminate individual MUAPs from a signal recorded during a muscle contraction. Instead, the recording will represent the overall activity of the motor units within the vicinity of the electrode. This superposed signal is called the myoelectric signal,or more commonly the electromyogram (EMG). When skin surface electrodes are used, the filtering effect of the skin tissue will accompany the superposition of MUAPs in determining the gross signal known as the surface EMG. The amplitude and spectrum of the EMG will depend on the motor units innervated, their spatial arrangement with respect to the recording site, and the properties of the tissue between the muscle and the electrodes, as well as the properties of the electrodes and even the cables to the amplifiers and the amplifiers themselves. Fig.2.3 shows a typical EMG recording while Fig.2.4 schematizes the generation of the EMG.

It has been shown that different muscle groups have intrinsicly different spectral properties. There is also evidence that these properties are further modified by the conduction media and the distance form the muscles to the

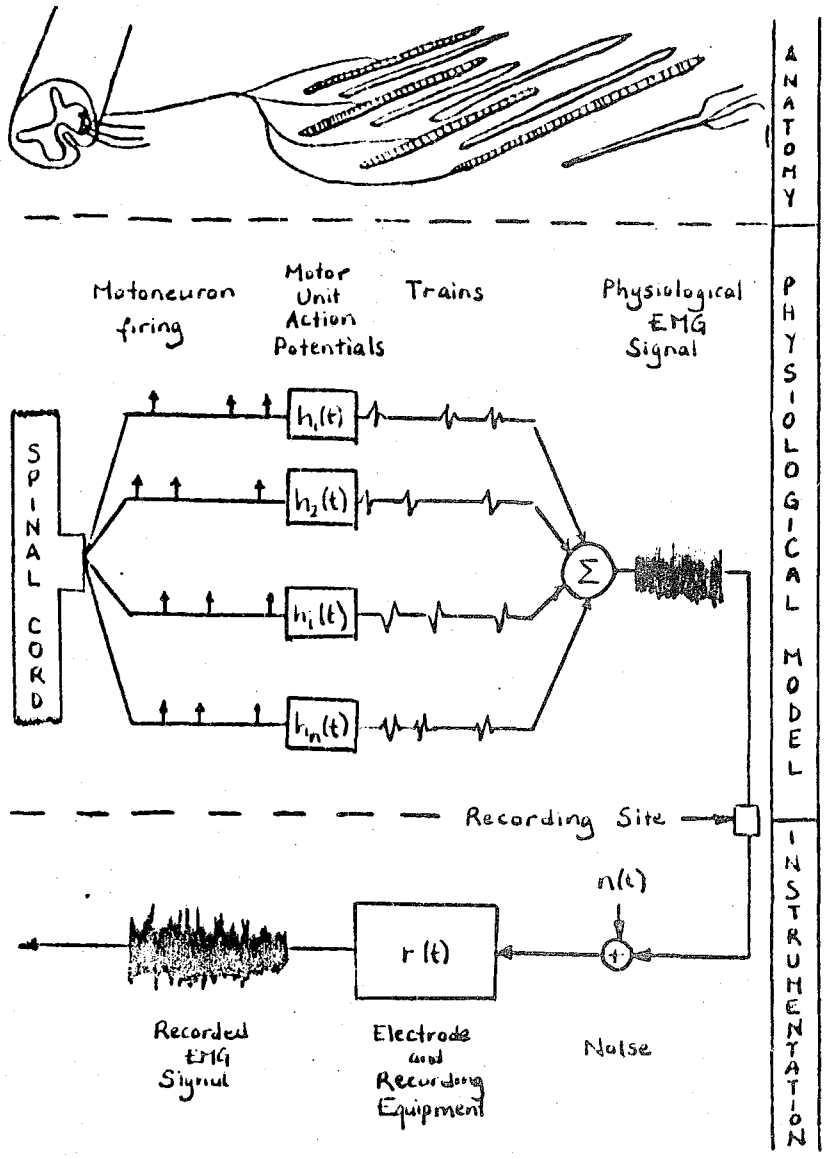

Figure 2.4 Generation of the EMG [CD1]

electrodes. These facts imply that the EMG contains information about the muscles that contracted to generate the specific signal recorded.

2.5 General Properties of Muscle Contraction

There are three major motivations for analyzing the EMG: 1) to conduct basic research on the electrical properties of muscles, 2) to extract informaton about the nature of the muscle and the innervation signal to it for diagnostic and therapeutic purposes,and 3) to obtain discriminatory information about the type of contraction for developing aids for the handicapped. Though our main interest lies in the last aspect, the findings about the mammalian EMG which we will list below are mostly agreed upon by workers in all three fields.

1) stochastic nature: the EMG can be considered to be a zero-mean Gaussian process

2) spectrum: the spectrum of the EMG lies mainly within the 20-1000 Hz band

3) amplitude: typical EMG peak-to-peak values are 10 mV when measured with needle electrodes and 1 mV when measured with surface eletrodes

4) interspike interval: the interval between two consecutive motor unit discharges is irregular and can be described as a random variable

5) recruitment: during a constant force isometric contraction, a motor unit which is active at the beginning usually remains active throughout the contraction; the order of recruitment is generally from smallest to largest diameter motoneurons

6) firing rate: during constant force isometric contractions, the average firing rate, i.e., the reciprocal of the average interspike interval, decreases monotonically, suggesting that the sequential discharge of a motor unit is a time-dependent process

7) posttetanic potentiation: during repetitive excitation the twitch tension of motor unit increases with time,though no correlation has been found between the twitch tension and the amplitude of the MUAP

8)synchronization: motor units tend to synchronize during fatigue, during physiological tremor, or after exercise.

9) effect of load: load tends to cause a shift in the spectrum of the EMG towards the lower frequencies

III. INFORMATION AND SIGNAL PROCESSING

Using a simpler model to investigate the nature of a complex system is a technique commonly applied in science. In this study we establish a mathematical model for the surface EMG whose behavior we want to explore. Based on the parameters of this model, we employ pattern recognition techniques to extract information about the EMG and the conditions that gave rise to it i.e., the desire to perform a certain motion.

In this chapter we introduce the information and signal processing techniques to be used in the final myoelectric prosthesis control system. Linear modeling in time series analysis is discussed first. After a brief exposition to the fundamentals of autoregressive moving average (ARMA), autoregressive (AR) and moving average (MA) models, least-squares identification of these models are reviewed. Basic pattern recognition concepts are introduced to form a background for the second step in determining the limb function to be performed by the prosthesis, i.e., the classification of the pattern represented by the identified model parameters into distinct function classes. Finally, the nearest neighbor algorithm is discussed.

3.1 Linear Time-Series Modeling

Most processes can be modeled with linear combinations of past outputs and present and past inputs to the system. A relation of the form

$$y_k = -\sum_{i=1}^{N} a_i y_{k-i} + \sum_{l=0}^{Q} b_l w_{k-l} \qquad (1)$$

where y is the output and w is the input series, holds for such systems. In cases where there is no access to the input, as is the case with the EMG, it is common practice to consider the input to be white noise. Since there is no access to and therefore no information about the input, it is reasonable to choose a general signal with no distribution restrictions. The choice of a white noise driving signal is justified by the fact that it has a flat spectrum without bias toward any frequency.

Once model parameters a_i , $1<i<N$ and b_l, $1<l<Q$ are known, one can predict the output of the system from past outputs and inputs. For this reason linear modeling is widely referred to as linear prediction.

The model can be represented in the frequency domain with

$$H(z) = \frac{Y(z)}{W(z)} = \frac{\sum_{l=0}^{Q} b_l z^{-l}}{1+\sum_{i=0}^{N} a_i z^{-i}} \qquad (2)$$

where Y(z) and W(z) are the z-transforms of y_k and w_k , respectively. The general model of (2) containing both zeros and poles is known in literature as the ARMA model. Two

special cases of this model namely 1) where $a_i = 0 \quad 1<i<N$ and 2) where $b_l = 0 \; 1<l<Q$ give rise to the all-zero model known as the MA model and the all-pole model known as the AR model, respectively.

The time domain representation of the MA model is

$$y_k = \sum_{l=0}^{Q} b_l w_{k-l} \quad . \tag{3}$$

The AR relation can be represented with

$$y_k = - \sum_{i=1}^{N} a_i y_{k-i} + b_0 w_k \quad . \tag{4}$$

Among the three models, AR is the most thoroughly studied and most frequently used. Although ARMA is the most parsimonious model in terms of model order, and MA is the most efficient in modeling nonstationary systems, AR models can be used instead of them [DG5,DG2]. This is possible because the zeros in the two models can be represented with poles in AR, though this causes the model order to rise considerably and the modeling efficiency to degrade somewhat. Graupe gives proofs that ARMA and MA models can be obtained from identified pure AR models by means of polynomial division for various cases of stability / unstability , invertibility/noninvertibility [DG5,DG2]. The reason for preferring to work with AR models is the extensive literature available about its properties and the computational efficiency in its identification. Other attractive features are its lesser sensitivity to model order variations and involving only one inaccessible term.

3.2 Least Squares Identification of AR Coefficients

Obtaining the parameters of an unknown model, or in other words parameter estimation, is an important topic in time series analysis. There are various approaches to parameter estimation, but only least-squares (LS) estimation will be dealt with here. LS is a common and well-documented technique. Major reasons for the popularity of LS algorithms are their convergence properties and robustness to round-off errors [DG5,JS1].

Since the signal which is of main interest in this study, namely the EMG is a stochastic process, the discussions in this section will be in terms of stochastic processes. However, the same reasoning is also valid in the case of deterministic signals.

In the LS technique, the parameters of the system under observation are obtained by minimizing the square of the prediction error with respect to each of the parameters. Because the input to the system is unknown, the only means to predict the output of the system is to use the linear combination of the past outputs. If $\hat{y}_k$ represents the predicted value of the signal y such that

$$\hat{y}_k = -\sum_{i=1}^{N} a_i y_{k-i} \quad , \tag{5}$$

then the prediction error and the expected value of its square, E_T is given by

$$e_k = y_k - \hat{y}_k = y_k + \sum_{i=1}^{N} a_i y_{k-i} \tag{6}$$

$$E_T = E[e^2] = E[(y_K + \sum_{i=1}^{N} a_i y_{K-i})^2] \qquad (7)$$

In order to minimize the squared error we set

$$\frac{\partial E_T}{\partial a_m} = 0 \qquad , 1 \leq m \leq N \qquad (8)$$

Substituting (7) in (8) gives the equation

$$\sum_{i=1}^{N} a_i E[y_{K-i} y_{K-m}] = -E[y_K y_{K-m}] \qquad 1 \leq m \leq N \qquad (9)$$

which is known as the normal equation. The minimum error is given by

$$E_T = E[y_K^2] + \sum_{i=1}^{N} a_i E[y_K y_{K-i}] \qquad (10)$$

The cases of y_K being stationary and nonstationary have to be addressed separately in taking the expectations in (9) and (10).

a) Stationary case : When y is a stationary signal,

$$E[y_{K-i} y_{K-m}] = R(m-k) \qquad (11)$$

where R(m) is the autocorrelation function. Thus we have from (9) and (10)

$$\sum_{i=1}^{N} a_i R(m-i) = -R(m) \qquad 1 \leq m \leq p \qquad (12)$$

$$E_T = R(0) + \sum_{i=1}^{N} a_i R(i)$$

b) Nonstationary case : When y_K is a nonstationary signal,

$$E[y_{K-i} y_{K-m}] = R(k-i, k-m) \qquad (13)$$

where R(t,t') is the nonstationary autocorrelation function between times t and t'. At time k=0 the normal equations become

$$\sum_{i=1}^{N} a_i R(-i, -m) = -R(0, -m) \qquad 1 \leq m \leq N \qquad (14)$$

$$E_T = R(0,0) + \sum_{i=1}^{N} a_i R(0,i) \qquad (15)$$

Many LS algorithms have been proposed which arrive at the parameters to be estimated by different means though they all start out by minimizing the squared error. The most popular ones among these will be briefly outlined here with a comparison based on convergence, convergence rate, computational effort per iteration, and robustness to round-off errors arising from limited word lengths of computers and microprocessors.

3.2.1 Batch LS Methods

In this section three batch LS Methods will be briefly explained.

3.2.1.1 Direct LS estimation

We can express the AR model

$$y_k = -\sum_{i=1}^{N} a_i y_{k-i} + w_k \qquad (16)$$

in vector form as

$$y_k = \underline{a}^T \underline{y}_k + w_k \qquad (17)$$

where

$$\underline{a}^T = [\, a_1 \;\; a_2 \;\; \cdots \;\; a_N \,] \qquad (18)$$

$$\underline{y}_k^T = [\, y_{k-1} \;\; y_{k-2} \;\; \cdots \;\; y_{k-N} \,] \qquad (19)$$

If we define $\underline{y}^*$, $\underline{e}$, and $\underline{\underline{U}}$ as

$$\underline{y}^{*T} = [\, y_k \;\; y_{k-1} \;\; \cdots \;\; y_{k-r+1} \,] \qquad (20)$$

$$\underline{e}^T = [\, e_k \;\; e_{k-1} \;\; \cdots \;\; e_{k-r+1} \,] \qquad (21)$$

and

$$\underline{\underline{U}} = \begin{bmatrix} y_{k-1} & y_{k-2} & \cdots & y_{k-N} \\ y_{k-2} & y_{k-3} & \cdots & y_{k-N-1} \\ \vdots & \vdots & & \vdots \\ y_{k-r} & y_{k-r-1} & \cdots & y_{k-N-r+1} \end{bmatrix} \qquad (22)$$

where r is the number of observations, we can express the prediction error vector $\underline{e}$ in terms of the obseervation vector y_k^*, the observation matrix $\underline{\underline{U}}$ and the estimated vector $\hat{a}$ for $\underline{a}$ as

$$\underline{e} = \underline{y}^* - \underline{\underline{U}}\,\hat{\underline{a}} \qquad (23)$$

The LS cost of the error in predicting y_k with the estimate $\hat{\underline{a}}$ based on r observations can be defined by

$$J_r = (\underline{y}^* - \underline{\underline{U}}\hat{\underline{a}})^T (\underline{y}^* - \underline{\underline{U}}\hat{\underline{a}}) = \underline{e}^T \underline{e} \qquad (24)$$

In order to minimize the cost J_r with respect to $\hat{\underline{a}}$ we set

$$\frac{\partial J}{\partial \hat{\underline{a}}} = 0 \qquad (25)$$

Solving for $\hat{\underline{a}}$, we get

$$\hat{\underline{a}} = (\underline{\underline{U}}^T \underline{\underline{U}})^{-1} \underline{\underline{U}}^T \underline{y}^* \qquad (26)$$

3.2.1.2 Covariance Method

Noting that the terms $(\underline{\underline{U}}^T \underline{\underline{U}})$ and $\underline{\underline{U}}^T \underline{y}^*$ in (26) approximate

$$\underline{\underline{W}} = \begin{bmatrix} R(1,1) & R(2,1) & \cdots & R(N,1) \\ R(1,2) & R(2,2) & \cdots & R(N,2) \\ \vdots & \vdots & & \vdots \\ R(1,N) & R(2,N) & & R(N,N) \end{bmatrix} \qquad (27)$$

and

$$R^T(0) = [\, R(0,1) \quad R(0,2) \quad \cdots \quad R(0,N) \,] \qquad (28)$$

for a nonstationary random process y_k, we obtain the parameter estimate vector with the equation

$$\hat{\underline{a}} = -\underline{W}^{-1} R(0) \qquad (29)$$

3.2.1.3 Autocorrelation Method

If the process y_k is wide sense stationary, the covariances of the covariance method reduce to autocorrelations. Defining $\underline{Z}$ and $\underline{S}(0)$ as

$$\underline{Z} = \begin{bmatrix} R(0) & R(1) & . & . & . & R(N-1) \\ R(1) & R(0) & . & . & . & R(N-2) \\ . & . & & & & . \\ . & . & & & & . \\ R(N-1) & R(N-2) & & & & R(0) \end{bmatrix} \qquad (30)$$

and

$$\underline{S}^T(0) = [\, R(1)\ R(2)\ .\ .\ .\ R(N)\,] \ ,$$

we obtain the formulation of the autocorrelation method:

$$\hat{\underline{a}} = -\underline{Z}^{-1} \underline{S}(0) \qquad (31)$$

Here $\underline{Z}$ is a symmetric Toeplitz matrix in which all elements along a diagonal are equal.

3.2.1.4 PARCOR Method

The major shortcoming of the above methods is the matrix inversion operation that is required to obtain the parameter estimates. Noting the Toeplitz matrix feature of the autocorrelation matrix, Levinson proposed a recursion method, which was further improved by Durbin, in order to overcome the problem of matrix inversion. This method reduces

the N dimensional problem into N one-dimensional problems. The method, known as the PARCOR method because of the intermediate parameters called partial correlations it uses, can be specified as

$$E_0 = R(0) \tag{32}$$

$$k_i = - [R(i) + \sum_{j=1}^{i-1} a_j^{(i-1)} R(i-j)] / E_{i-1} \tag{33}$$

$$a_i^{(i)} = k_i \tag{34}$$

$$a_i^{(j)} = a_i^{(j-1)} + k_i a_{i-j}^{(i-1)} \quad , \quad 1<j<i-1 \tag{35}$$

$$E_i = (1-k_i^2) E_{i-1} \tag{36}$$

where $a_i^{(j)}$ is the i'th AR coefficient of a model of order j. Equations (33)-(36) are solved recursively for i=1,2,...,N. The final solution is given by

$$a_j = a_j^{(N)} \tag{37}$$

The parameters k_i are known as the reflection coefficients or partial correlation coefficients.

3.2.2 Sequential (Iterative) Methods

In some applications where the parameters are to be estimated continually, it might not be convenient to wait till all the data are available. In these cases it is desired to update the estimates as new data come in without going through the whole estimation procedure again. Sequential algorithms start with an initial guess and "correct" this as new samples become available without solving any of the equations of the batch algorithms afresh.

The sequential LS (SLS) algorithm uses the matrix inversion lemma in order to avoid the computational difficulties of matrix inversion as well as supplying estimates continually. It can be specified with the iterative equations

$$\underline{a}_{k+1} = \underline{a}_k + \underline{\underline{P}}_{k+1}\, \underline{y}_{k+1}\, (y_{k+1} - \underline{y}_k^T \underline{a}_k) \qquad (38)$$

$$\underline{\underline{P}}_k^{-1} = \underline{\underline{P}}_k - (\underline{\underline{P}}_k \underline{y}_{k+1}\, \underline{y}_{k+1}^T\, \underline{\underline{P}}_k) \,/\, (1+\underline{y}_{k+1}\, \underline{\underline{P}}_k\, \underline{y}_{k+1}) \qquad (39)$$

where

$$\underline{\underline{P}}_k^{-1} = \sum_{i=1}^{k} \underline{y}_i\, \underline{y}_i^T \qquad (40)$$

and

$$\underline{\underline{P}}_o = \frac{1}{\varepsilon}\, I \qquad (41)$$

Since the denominator of the second term in the right hand side of (39) is a scalar no matrix inversion is needed to reach the solutions.

With the recursion formula for $\underline{R}$

$$R^{j+1}(l) = R^j(l) + \frac{1}{j+1}\, [y_{j+1}\; y_{j+1-l} - R^j(l)] \qquad (42)$$

one can obtain sequential solutions for the covariance and PARCOR methods [JS11. Using equation (42) with the original equations of the methods provides the sequential version.

3.2.3 Lattice Algorithms

For AR models of order higher than 6 lattice algorithms provide fast and convergent solutions [DG5,JS11. The lattice formulation is based on the minimization of the

forward and backward errors defined as:

$$e_k^N = y_k - \sum_{i=1}^{N} a_i y_{k-i} \qquad (43)$$

$$r_k^N = y_{k-N} - \sum_{i=1}^{N} b_i y_{k-N+i} \qquad (44)$$

where a_i and b_i denote the forward and backward predictor coefficients and e_k^N and r_k^N denote the forward and backward prediction errors for a model of order N, given observations $y_k, \ldots, y_{k-N}$. For an AR model of order j+1, j=1,...,N the forward and backward reflection coefficients K_e and K_r are given by

$$K_e^{(j+1)} = \frac{\Delta^{(j+1)}}{R_e^{(j+1)}} \qquad (45)$$

$$K_r = \frac{\Delta^{(j+1)}}{R_r^{j+1}} \qquad (46)$$

where

$$R_e^{(j+1)} = E[e_k^{(j)2}] \qquad (47)$$

$$R_r^{(j+1)} = E[r_k^{(j)2}] \qquad (48)$$

$$\Delta^{(j+1)} = E[e_k^{(j)} r_k^{(j)}] \qquad (49)$$

The recursion formula for the forward and backward errors can be expressed as

$$e_k^{(j+1)} = e_k^{(j)} - K_r^{(j+1)} r_{k-1}^{(j)} \qquad (50)$$

$$r_k^{(j+1)} = r_{k-1}^{(j)} - K_e^{(j+1)} e_k^{(j)} \qquad (51)$$

The forward predictor coefficients a_i, which are actually equivalent to b_i for stationary signals can be obtained from [DG5] :

$$a_i^{(j+1)} = a_i^{(j)} + K_e^{(j+1)} a_{j+1-i} \qquad i=1,\ldots,j,\ j=0,\ldots,N \qquad (52)$$

$$K_e^{(j+1)} = a_{j+1}^{(j+1)}, \quad a_0^{(j)} = 1 \qquad (53)$$

A sequential formulation of the lattice algorithm

also exists which retains the convergence and convergence rate properties of the original batch algorithm [JS11].

3.2.3 Comparison of Various LS Algorithms

The various LS algorithms outlined exhibit different computation time, convergence and robustness to roundoff errors. These varying properties should be considered in light of the requirements of the specific application in choosing the algorithm to be used.

The computational time for PARCOR and lattice algorithms is shorter than for direct LS algorithms, especially for high model orders. Obviously the computational time required for each technique is related to the number of operations to be performed. The approximate number of operations for various algorithms are listed below, n being the number of samples and N the model order:

Algorithm	Operations
Direct LS	$n(N^2/2)+N^3/6$
PARCOR	$nN+5N^2/2$
Lattice	$nN+5N^2/2$

It is reported that for long word lengths the convergence and convergence rate properties of all LS algorithms are similar with PARCOR exhibiting somewhat large oscillations about the true parameters [JS11]. Hovewer with short word-length machines, there is no guarantee that the algorithms except sequential LS will converge to the true

parameters. Sequential LS provides convergence regardless of word-length limitations. The direct LS, both sequential and batch are the only algorithms found to converge for unstable models [DGS,JS11.

3.3 Basic Concepts in Pattern Recognition

The main aim of pattern recognition studies is to classify an unknown sample as belonging to one of a finite number of classes. The classification is based on knowledge obtained from "labeled" samples whose class memberships are known.

The basic steps in a pattern recognition problem are schematized in Fig.3.1. From the physical system which generates the patterns to be classified, we get the raw data describing the system by means of measurements. At this stage the pattern is represented by the values of the measurements. With preprocessing, features that contain discriminatory information about the patterns are selected. Points of the measurement space are transformed into the pattern space where they are represented by the values of the selected features. The main concerns in feature selection are that the pattern space be of low dimension, that the features carry sufficient information about the patterns and that the features allow consistent comparison between patterns. We might later decide that only a subset of the pattern space is adequate in reaching the decision rule. Finally we must

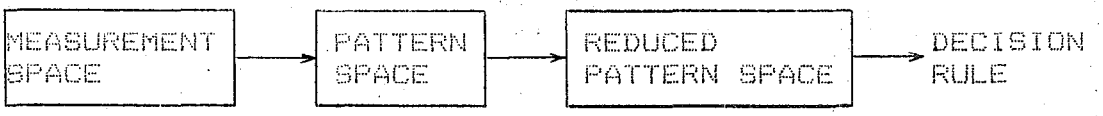

Figure 3.1 Stages in pattern recognition

develop, based on the labeled samples, a decision rule to classify an unknown sample into the correct pattern class.

In most pattern recognition approaches it is assumed that the closeness of two samples in pattern space corresponds to the similarity of their patterns [WM1]. The closeness measure between two samples x and y need not be the Euclidian distance

$$d_E(x,y) = \left[\sum_{i=1}^{n} (x_i - y_i)^2 \right]^{1/2} \qquad (54)$$

where

$$\underline{x}^T = [\, x_1 \;\; x_2 \;\; \ldots \;\; x_n \,] \qquad (55)$$

$$\underline{y}^T = [\, y_1 \;\; y_2 \;\; \ldots \;\; y_n \,] \qquad (56)$$

For a distance function d(x,y) the following properties are usually desired:

i) $d(x,x) = 0$
ii) $d(x,y) > 0 \qquad x \neq y$
iii) $d(x,y) = d(y,x)$
iv) $d(x,y) + d(y,z) \geq d(x,z)$.

3.3.1 Nearest Neighbor Algorithm

Among the many pattern recognition algorithms, in literature, the nearest neighbor (NN) algorithm is a simple one capable of generating complex boundaries between classes nevertheless. With this method, the class membership of an unknown sample is decided based upon its nearest neighbor.

The sample is classified as a member of the class to which its nearest neighbor belongs. Different versions of the NN algorithm are the K-nearest neighbor (KNN) algorithm where the majority vote of K nearest neighbors of the unknown sample are considered and the generalized K-nearest neighbor algorithm where the total distances of the K nearest neighbors of the sample in each class are taken into account.

Since the NN algorithms assume that the distance between samples is a legitimate measure of the similarity between the patterns they represent, special care must be taken in choosing the appropriate distance measure for the specific application considered.

In implementation NN pattern classification requires the computation of the distance from all labeled samples to the point in question and a large storage space to store all the labeled samples. In order to reduce the necessary storage space and number of computations, Hart proposed a technique which came to be known as the condensed nearest neighbor rule. Later, modifications to this technique were also suggested. However the common idea is to go through the labeled samples one by one and try to make a classification based on the samples available until then. If the classification is correct, the sample is discarded. If not, the sample is added to the set of samples which is to be used as a reference library for later classifications. The procedure is repeated until either all the samples are placed

in the reference library or a complete run over all the discarded samples is done without any incorrect classifications. The method is reported to pick samples on class boundaries as references and discard the ones that are deeply imbedded in a sample class [PH11].

IV. MOTION CLASSIFICATION BASED ON AR COEFFICIENTS OF EMG SIGNAL

In a myoeletrically controlled prosthesis system, the EMG signal recorded with surface electrodes is processed to decide on the motion intended by the amputee. Based on this decision, the motors of the prosthesis are activated in the appropriate fashion. Observing the actual performance of the prosthesis, the amputee can alter the contraction in his muscles to create the change he wishes in the operation of his prosthesis. The operation of the myoelectric prosthesis in conjunction with the amputee's neuromuscular system is schematized in Fig.4.1.

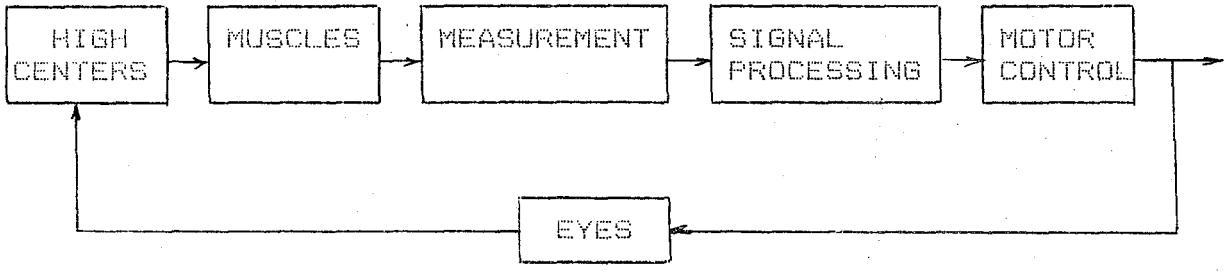

Figure 4.1 Operation of the myoelectric prosthesis in conjunction with the amputee's neuromuscular system

A myoelectrically controlled prosthesis basically consists of a signal processing unit and a motor control unit. A block diagram of the signal processing unit which is of main interest in this thesis is given in Fig.4.2. The

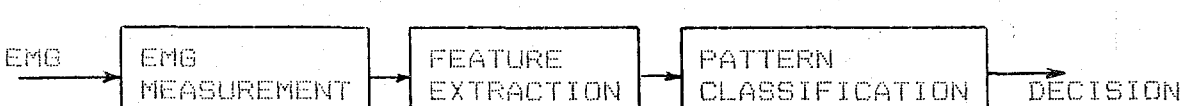

Figure 4.2 Block diagram of signal processing for a myoelectric prosthesis

acquired EMG is first analyzed in order to find features characterizing the signal most efficiently. Based on these features, EMG signals are classified as having resulted from certain motions. The techniques employed in feature extraction and pattern classification vary widely among different approaches.

4.1 Method of Approach

In the present study we will approach the problem of motion classification by estimating the intended limb function directly from the single channel of surface EMG signal. We will not try to decide on the limb function by means of muscle force estimates drawn from the average power of the EMG signal. Nor will we be concerned with how each muscle is functioning. We maintain that there is sufficient information in the spectral shape of EMG recorded with high muscle cross-talk to enable discrimination between at least three degrees of freedom. Using a single channel is important in the case of amputees with very short stumps or high nerve damage.

We assume that the EMG signals can be modeled as stochastic autoregresive processes. This assumption is based on the properties of the EMG discussed in Chapter II, basically its being a nearly Gaussian and nearly stationary process for intervals of up to 0.3 s. A suitable model for the EMG can be specified by

$$y_k = \sum_{i=1}^{N} a_{i,m}\, y_{k-i} + e_k \qquad m = 1,\ldots,M \qquad (1)$$

where y_k is the EMG signal observed at time k, $a_{i,m}$ is the i'th AR coefficient of the m'th limb function model, M is the number of limb functions, N is the model order and e_k is the one step ahead prediction error.

The AR coefficients for each limb function were determined from a set of data recorded while the specific function was being performed. In the identification of these parameters the PARCOR algorithm of Chapter III was employed. The choice of this algorithm was justified by its computational speed and convergence for the model order, sample size and word-length of interest.

In the limb function discrimination stage, the model parameters of the recorded signal were identified, again with the PARCOR algorithm, and compared to each of the parameters formerly obtained during the execution of different limb functions. The comparison was based on the K-nearest neighbor algorithm employing the Itakura-Saito distance measure. The reference library was condensed with the K-nearest neighbor rule in order to reduce the number of reference coefficient sets to be stored and to increase

computation speed by reducing the number of comparisons required for each sample.

Fig.4.3 shows the basic steps in classifying an unknown EMG signal as corresponding to the desire to execute one of M limb functions.

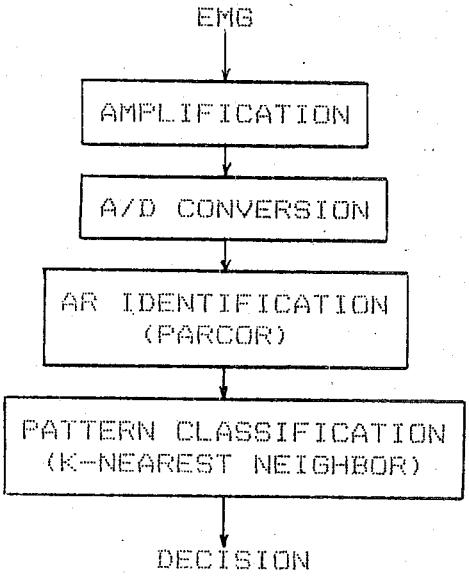

Figure 4.3 Block diagram of EMG classification scheme

4.2 Experimental Procedures

In the implementation and evaluation of a system to classify EMG signals with the above approach, the following set-up was used.

The EMG signals were acquired with Beckman type silver-silver chloride electrodes. These signals were fed into the differential amplifier of the Physiograph MK-IV-P manufactured by Narco Bio-Systems. The amplifier, the

universal coupler as it is called, has adjustable gain up to 100 and four high-pass filter settings of which 10 kHz was used. The output of the amplifier was fed to the computer PDP 11/23 of Digital Equipment Corp. after being digitized by the 12-bit Analog/Digital converter.

The EMG was recorded from a 25-years-old healthy male subject. The electrodes were placed at the midpoints of the lines connecting the bulges of the biceps and triceps muscles in order to maximize cross-talk.

A series of 15 experiments were conducted for each limb function to be classified. In five of these experiments the subject started from the rest position, performed a function and sustained the contraction in his muscles for some time. In five other experiments, he started from rest, performed a function and returned his arm to rest. The last five experiments started from a muscle contraction and returned to rest. In all the experiments the subject tried to apply the same force in his muscle contractions. During each experiment 4096 data points were colected over a period of 2 sec.

The EMG signals were sampled at a rate of 2048 Hz. There were two major concerns in the selection of the sampling rate. The first one which concerned the higher frequencies was Nyquist's theorem stating that the sampling frequency of a signal should be at least twice its highest frequency component. Lower frequency considerations can be

explained with the help of equations (30) and (31) of Chapter III. From these equations it is evident that the AR coefficients are a function of signal correlations with delays of up to N sampling intervals, N being the model order. For the same model order increasing the sampling rate will cause only those samples that are observed closer to each other in time to be taken into account. Thus only fast changes in the signal will be considered and components with lower frequencies will be disregarded [JS1,DG1,DG5]. Considering the low and high frequency constraints, it was decided that the lowest sampling frequency permitted by Nyquist would be the optimum one. Since the EMG signal was considered to have significant components up to 1000 Hz, the sampling rate of 2048 Hz seemed adequate.

The major concern in selecting the model order was class separability. Theoretically the model order should be infinity in order to model a process efficiently [DG5]. However it is obvious that discrimination is not possible with a model of infinite order and some truncation is necessary. A test for the validity of the truncation is to check if the resulting prediction error e_k is white [DG4,DG5]. Another practical test is to check a_i for N and N+1 . If a_i $i<N$ don't change considerably with increased model order, N can be considered to be adequate. There is a formal criterion given by Akaike for finding the optimum model order [DG5]. This criterion is reported in [PD1] to have no minimum for model orders less than 20. Since our aim in modeling the EMG is motion discrimination and not the

reproduction of the signal, the model order of four which is reported in various studies [DG4,JS1,PD1] to provide adequate class separability was chosen.

The mean of each record of 4096 points were equated to zero in order to eliminate any offset arising from the amplifier or the ADC. The EMG signals were analyzed in frames of 256 samples. These frames were designed to slide 192 samples each time so that consecutive frames had 64 samples in common, instead of being completely disjoint. This produced a smoothing effect and prevented spurious leaps in the estimated parameters. The first and last 128 points of each data set were discarded.

One major problem encountered in the analysis of the EMG signals was the detection of the starting point of the motion. Although it is reported that the signal variance and the zero-crossings are features carrying information in this regard, and even in determining the type of motion [GS1,DG4,SL1], we couldn't establish a reliable rule based on these parameters. However noting that the AR coefficients of EMG data recorded while the arm was in a resting position were much farther apart from the coefficients in the AR-coefficient space than those identified for other motions, we devised another scheme to detect the starting of motion. We identified second-order models for resting position recordings. Among these models, we selected average values for a_1 and a_2 coefficients. When we filtered all signals with the filter described by these coefficients, we found that the

maximum value of the total squared prediction error in the case of a rest signal was exceeded by far when the signal corresponded to the execution of any of the motions. Thus we had the rule to locate the beginning of the motion : motion starts in the frame which, when filtered with the rest coefficients, gives a prediction error above the predetermined threshold. Our experiments showed that this rule worked accurately.

Five motion classes were considered at first: grasp, wrist flexion , elbow flexion , wrist supination , and wrist pronation. In order to get an idea of the distribution of the coefficients of each motion in the AR-space, 2-dimensional plots were made for the identified coefficients. Since the coefficients identified for wrist supination and wrist pronation were too dispersed in the coefficient space to allow reasonable class separability, they were not included in the discrimination scheme. The motion classification procedures were based on the motions hand grasp, wrist flexion, and elbow extension. The analysis of the EMG data recorded during the experiments revealed that the reverse motions of these three limb functions could be modeled with the same set of coefficients, namely the relaxation coefficients.

Tables 4.1-4.3 list 15 coefficient sets for each motion chosen randomly. Figures 4.4-4.6 show plots of a_1 vs. a_2 , a_1 vs. a_3 and a_1 vs. a_4 . Each point on the plot is the

-0.9868973	0.4135441	-0.1781272	4.9872585E-02
-0.9238889	0.3700785	-0.1411104	2.2664343E-03
-0.9662027	0.3497936	-0.1850980	8.2899317E-02
-1.036111	0.4310451	-0.1640306	5.4973766E-02
-0.9736309	0.3203971	-7.5177759E-02	3.7194159E-02
-0.9901525	0.2519591	5.4375257E-02	-8.6420745E-02
-1.033489	0.3749804	-0.1015270	-0.1097595
-1.205060	0.5341441	-0.2260433	7.1972273E-03
-1.004602	0.3640381	-0.2035818	1.8573069E-03
-0.9606115	0.2372181	-7.8787826E-02	-5.9341472E-02
-1.055807	0.3719173	-0.1407399	7.2053154E-03
-0.9460234	0.3372313	-4.9495578E-02	-0.1876967
-1.049916	0.3994274	-0.1250756	-0.1068040
-1.125023	0.4724110	-0.2027663	-8.1809446E-02
-1.098674	0.3520922	-0.1640715	-2.4301369E-02
-1.013338	0.3620485	-0.1216763	-9.5656916E-02

Table 4.1 Some AR coefficients for grasp motion

-1.100764	0.2540908	9.1454424E-03	3.1072866E-02
-1.150224	0.4006451	-4.4933207E-02	-2.2423910E-02
-1.039038	0.3949793	-0.1015580	-0.1173470
-0.9453627	0.2893032	-0.1636174	-9.2548765E-02
-1.279399	0.5382760	-5.2845418E-02	-8.2068607E-02
-1.227702	0.4852079	-6.9364898E-02	-8.3742306E-02
-1.230392	0.4987342	-0.1832297	2.8145600E-02
-1.392853	0.7240800	-0.1418037	-6.1277784E-02
-1.204865	0.5662515	-0.2002968	7.2724566E-02
-1.214135	0.6623532	-0.4676208	0.2611093
-1.115842	0.4289446	-0.2591258	0.1724766
-1.089047	0.3361984	-2.9212534E-02	5.7100587E-02
-1.163365	0.4853347	-0.1694197	0.1030014
-1.239348	0.6224927	-0.2541445	0.1169202
-1.264089	0.6329623	-0.2091307	7.6514594E-02

Table 4.2 Some AR coefficients for wrist flexion

-1.444376	0.6926787	-0.1272859	6.2175255E-02
-1.518281	0.8658500	-0.2254037	0.1062493
-1.452807	0.8047669	-0.2137695	0.1314627
-1.400902	0.8123088	-0.3369631	0.2511925
-1.337743	0.3057746	-3.4845658E-02	7.5522639E-02
-1.335898	0.1868511	0.2318017	-6.3675337E-02
-1.398910	0.2847454	0.2418235	-0.1019114
-1.420795	0.4612981	4.1054849E-02	-4.0890023E-02
-1.557871	0.7129784	3.8592875E-02	-0.1383368
-1.477771	0.7665429	-0.1933352	5.3932574E-02
-1.393965	0.6151355	-6.7743503E-02	5.9187304E-02
-1.418021	0.5595306	4.7464103E-02	-8.5382022E-02
-1.662964	0.9286942	-0.1865969	-1.4500367E-02
-1.617856	0.7475978	6.9310173E-02	-0.1300564
-1.557404	0.8026699	-0.1381874	-3.1776302E-02

Table 4.3 Some AR coefficients for elbow flexion

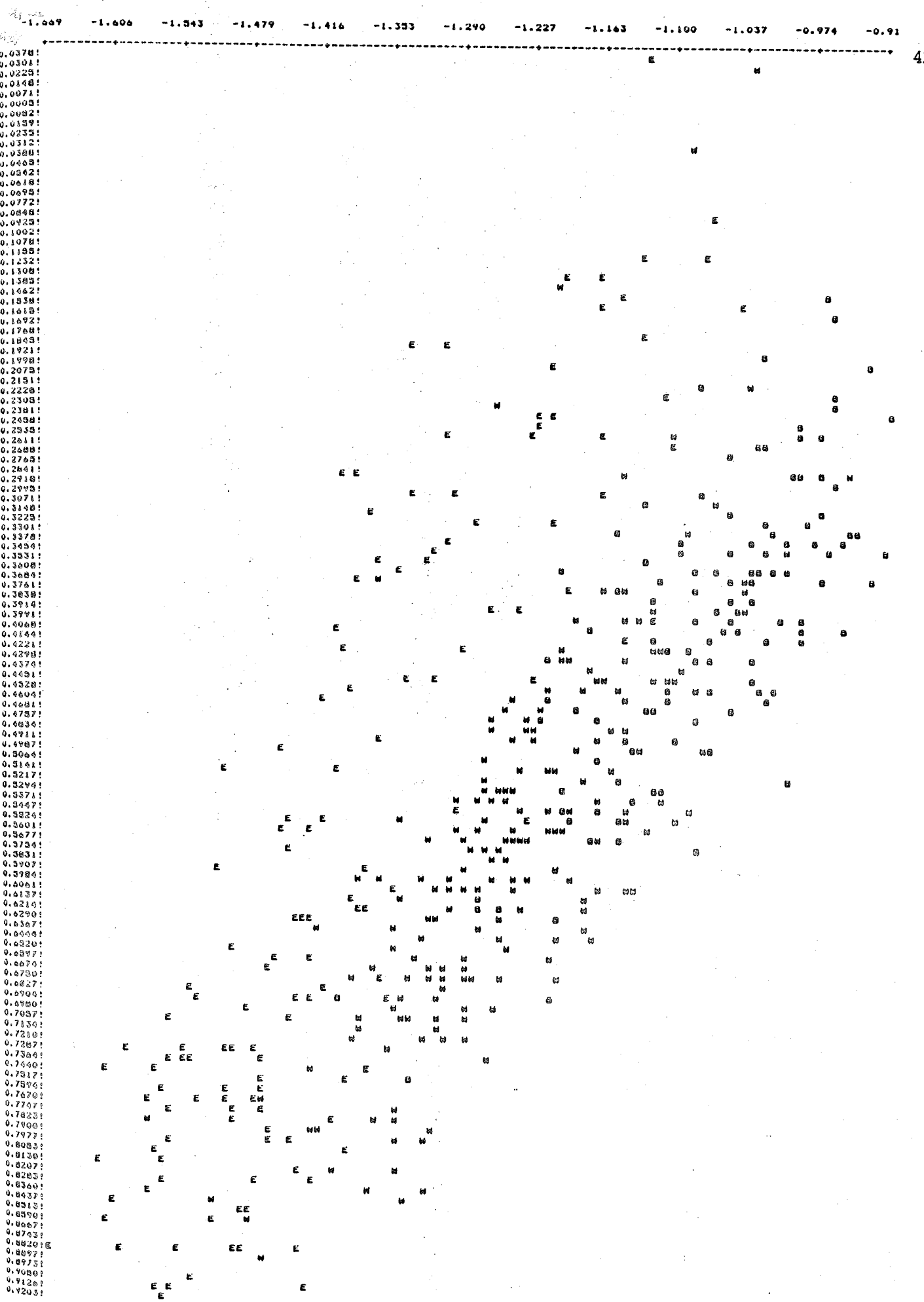

Figure 4.4 Plot of a_1 vs. a_2 for three motions

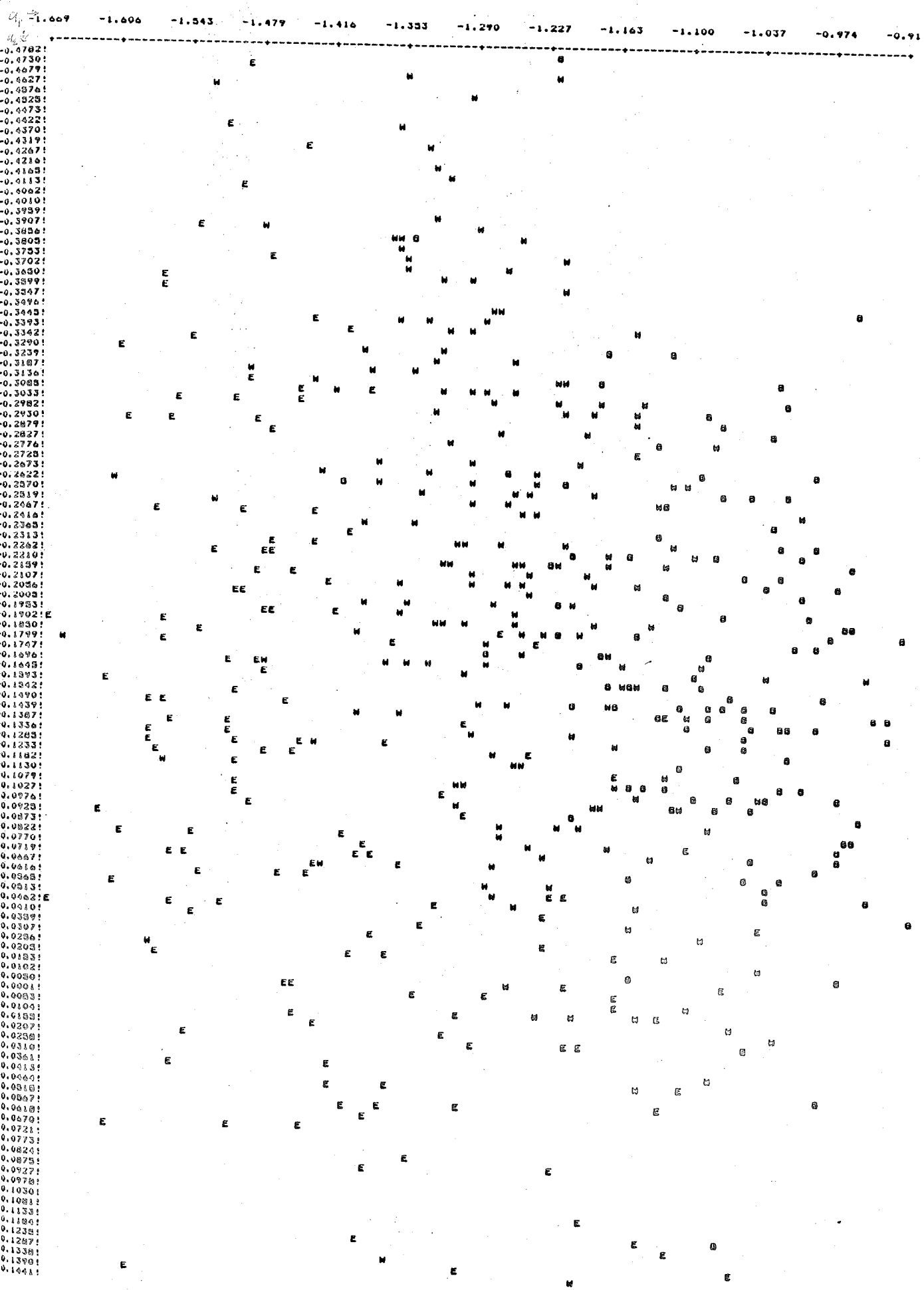

Figure 4.5 Plot of a_1 vs. a_3 for three motions

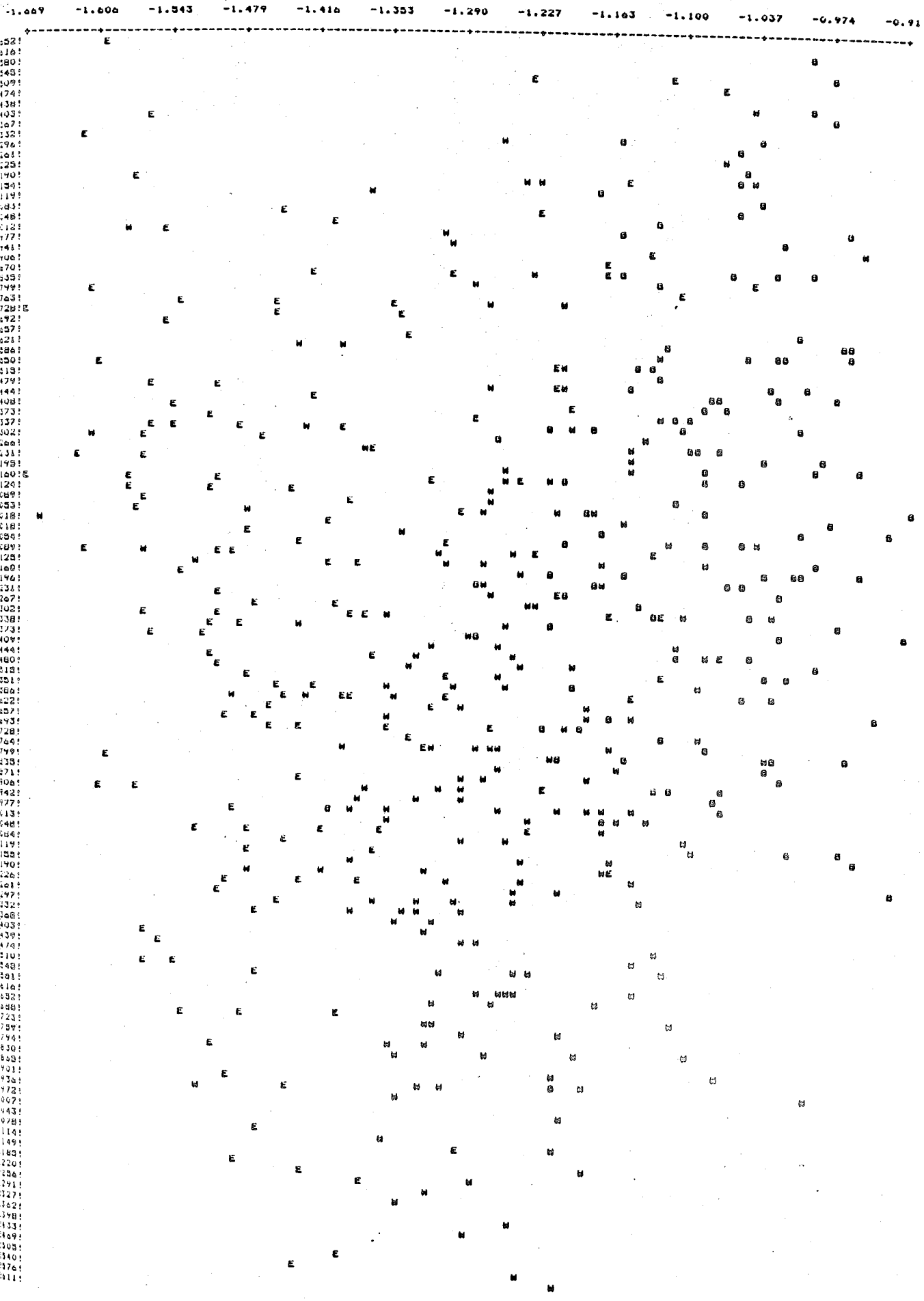

Figure 4.6 Plot of a_1 vs. a_4 for three motions

result of the identification run for a frame of 256 points recorded during the execution of the motion to which the point corresponds. We comment that our results show much less clustering of class members than reported in [JS1,DG4]. The most important factor in this is the absence of subject training. In subject training the subject is shown visually the model parameters that result from the limb functions and is asked to perform the function so as to generate repeatable and separable coefficient sets for each motion class. The significance of subject training is discussed in [RNS1, DG4,JS1]. Other reasons for the high degree of scatter we observed may be the absence of preamplification and the inadequate selection of the electrode locations. It is also claimed by Graupe et.al. that sampling frequencies about one fourth the Nyquist rate provide much greater class separability.

In order to evaluate our approach we divided our data into two sets: the design and test sets. The design set consisted of three experiments for each of the three motions considered. Two of the experiments were chosen, randomly, among the five experiments where the subjects started from rest, performed a limb function, and sustained the muscle contraction. One was chosen, again randomly, among the experiments where the subject started from rest, performed a function, and relaxed his muscles. The design set was used as the reference library containing the labeled samples. Each experiment where 4096 data points were collected comprised

20 frames. Among these 20, the number of frames that contained motion depended on the way the subject performed the limb function.

The classification of the limb functions were to be realized according to the K-nearest neighbor algorithm. Since this algorithm requires the comparison of the test sample to each of the reference points, reducing the size of the reference library increases the computational speed considerably as well as relaxing the storage space requirements. In order to reduce the library size without compromising the information content, the condensed K-nearest neighbor algorithm was employed. This algorithm works on the principle of classifying a sample with the current reference library and adding the sample to the library if the classification turns out to be incorrect and discarding the sample otherwise. Our condensing routine started with a minimal reference library containing K+1 reference AR coefficient sets, chosen randomly, for each motion class. For each experiment in the design set the following procedure was repeated. The experiment recordings were analyzed in frames described above. Each frame was first filtered with the rest filter. If the resulting squared prediction error summed over the frame was above the threshold value, meaning that the frame corresponded to motion of some kind, an attempt was made at classifying the frame with the present reference library. As in the final recognition scheme, classification was based on the K-nearest neighbor algorithm employing the

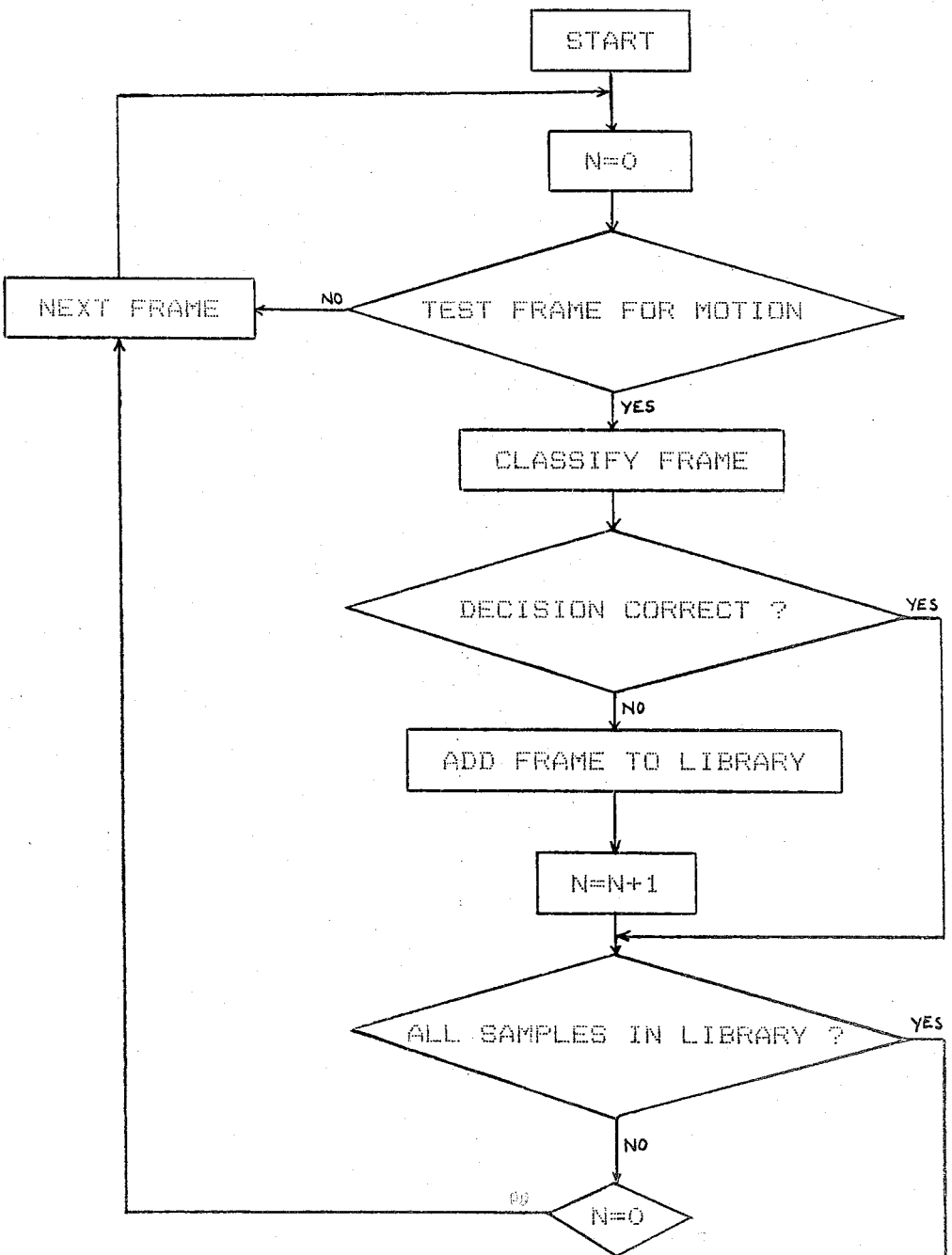

Fig.4.7 Steps in condensing reference library

AR coefficients as features and the Itakura-Saito distance measure. If the decision based on this algorithm, which will be explained in detail below, is correct the signal is discarded. If not, the AR vector identified for the test signal is added to the reference library. This procedure is repeated for all the frames that are not already in the library until either all frames are placed in the library or a complete run is made over the frames without any transfers to the library. Fig.4.7 outlines the basic operation of the condensing algorithm. Table 4.4 gives the reference library built by the condensed K-nearest neighbor algorithm. Fig.4.8 gives the plot a_1 vs. a_2 of elements of the library.

The test set consisted of the experiments that were not included in the design set, i.e., the design and test sets were completely disjoint. In classifying the members of the test set, the coefficient sets chosen by the condensing routine was used as the reference library. The distance measure was a form of the Itakura-Saito distance and can be specified as

$$d_{\ell,m} = \log \frac{\sum_{FRAME} \underline{a}^T_{\ell,m} \underline{y}_k \underline{y}_k^T \underline{a}_{\ell,m}}{\sum_{FRAME} \underline{a}^T \underline{y}_k \underline{y}_k^T \underline{a}}$$

where $\underline{a}_{\ell,m}$ is the l'th reference AR coefficient vector of m'th motion, $\underline{a}$ is the AR vector identified for test sample, y_k is the observation vector, and $d_{\ell,m}$ is the distance of l'th reference set of m'th motion to test signal. This distance measure is equivalent to feeding the test signal into filters whose coefficients are looked up from the reference library and calculating the squared prediction errors over the

MOTION 1 IS REPRESENTED WITH 8 REFERENCES

A1	A2	A3	A4
-1.070000	0.4200000	-0.2900000	9.9999994E-02
-1.157000	0.5260000	-0.1590000	-9.9199995E-02
-0.9700000	0.2610000	-7.0599996E-02	-0.1540000
-0.9540000	0.3430000	-9.1399997E-02	1.8199999E-02
-1.051927	0.2749816	2.7908280E-02	-0.2361932
-1.187739	0.5289417	-0.1856766	-3.2844741E-03
-1.170494	0.4758236	-0.3239900	0.1024713
-1.151139	0.3301669	-0.1072704	1.7655095E-02

MOTION 2 IS REPRESENTED WITH 17 REFERENCES

A1	A2	A3	A4
-1.202000	0.4290000	-0.1310000	-5.2699998E-02
-1.608000	0.9840000	-0.2580000	-2.8999999E-02
-1.425000	0.6380000	-6.4700000E-02	-3.0699998E-02
-1.150000	0.4620000	-9.8899998E-02	-1.7299999E-02
-1.389557	0.7208093	-0.2412199	9.9562116E-02
-1.219864	0.5515651	-0.2990901	0.1922530
-1.251618	0.5323874	-0.1719929	0.1565617
-1.352454	0.5591766	-0.1971812	0.1836835
-1.254586	0.4723584	-4.4217452E-02	-0.1327962
-1.382802	0.8412220	-0.3178419	9.4349205E-02
-1.094433	0.4445536	-0.2253723	0.1153982
-1.281014	0.5383275	-0.1765412	7.7161871E-02
-1.217827	0.5672443	-0.3132000	8.3143793E-02
-1.065644	0.3112926	2.0622581E-02	-0.1228178
-1.033525	0.2222362	-0.1595654	8.3201647E-02
-1.039038	0.3949793	-0.1015580	-0.1173470
-0.9453627	0.2893032	-0.1636174	-9.2548765E-02

MOTION 3 IS REPRESENTED WITH 11 REFERENCES

A1	A2	A3	A4
-1.145000	0.3040000	1.6700000E-02	8.4399998E-02
-1.373000	0.7230000	-347.0000	0.1870000
-1.430000	0.6849999	-0.2100000	9.5199995E-02
-1.262000	0.3750000	2.1100000E-02	2.6299998E-02
-1.215005	0.1209750	6.4740352E-02	3.5458378E-02
-1.360709	0.2923837	8.5502550E-02	-2.3746984E-03
-1.620008	0.8551712	-0.1811537	2.8868760E-03
-1.538374	0.8431432	-0.1814122	-8.8311825E-03
-1.385982	0.7502618	-0.2609550	0.1050746
-1.481586	0.8343262	-0.2896611	0.1706185
-1.068778	0.1220189	-1.5853643E-03	4.4752307E-02

Table 4.4 The condensed reference library

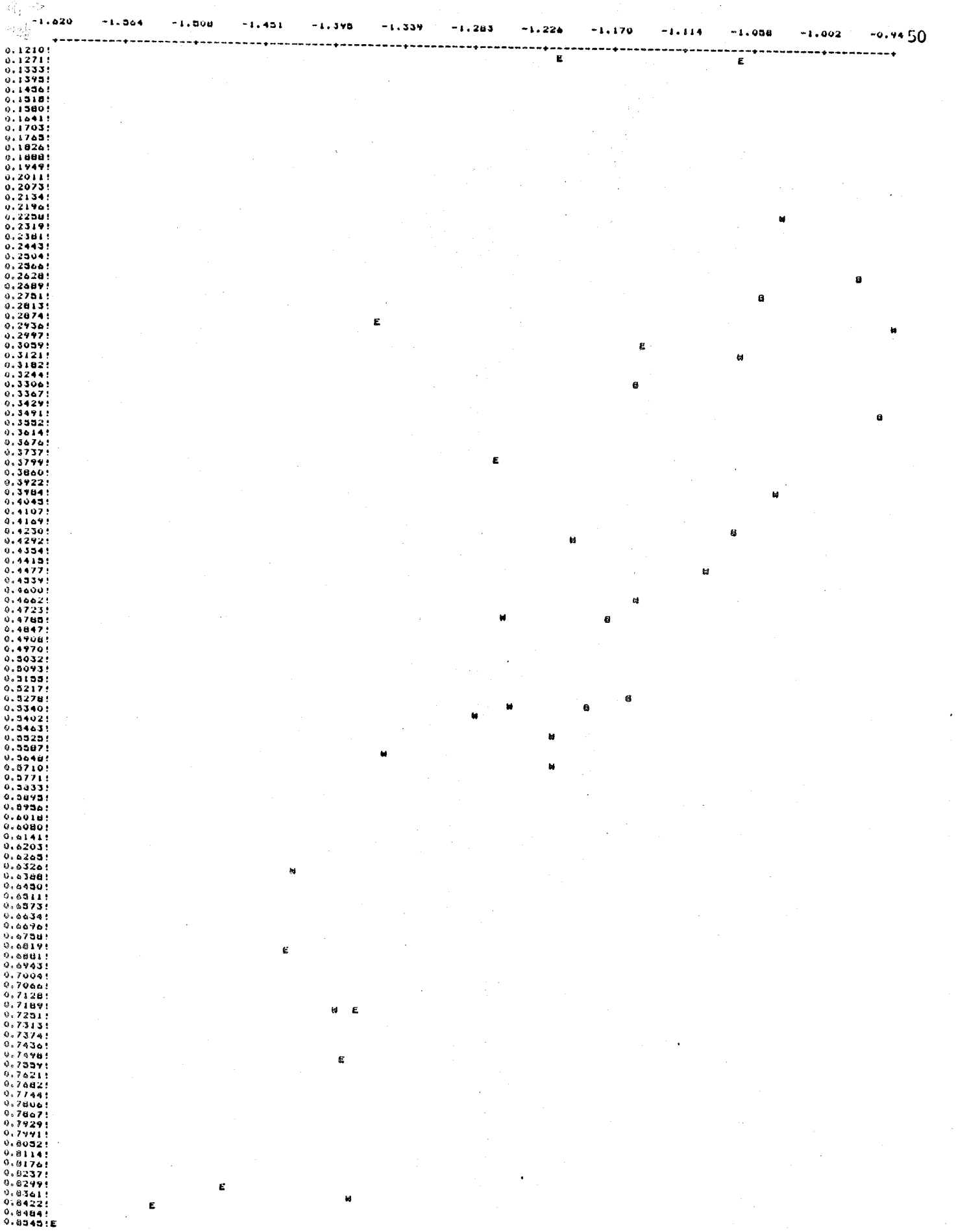

Figure 4.8 Plot of the condensed reference library

frames. This value is divided by the total error resulting from the identified coefficients of the test signal. Taking the logarithm amounts to taking a geometric average of the distances.

The distances $d_{l,m}$ were computed for $l=1,\ldots,R$, $m=1,\ldots,M$, R being the number of references for motion and M being the number of motions. For each motion class the lowest K distances were determined. The sum of these K distances represented the overall distance of the test sample to the specific motion class. Needless to say, "distance" does not refer to Euclidian distance here, but is a measure of the similarity between the test sample and the reference samples. These overall distances computed for each class are again compared and the smallest distance is determined. The class which generated the smallest distance is decided upon as the motion that was being performed while the test signal was recorded. The basic steps in motion classification are schematized in Fig.4.9. Note that the program for implementing this flow chart given in the appendix comprises some modifications introduced for the sake of computational ease and speed. For example, instead of taking the logarithm of K terms with the same denominator and adding these up, the logarithm of the product of the K numerators were computed and K times the logarithm of the denominator was subtracted from this quantity.These modifications bring only computational advantages and don't change the final results.

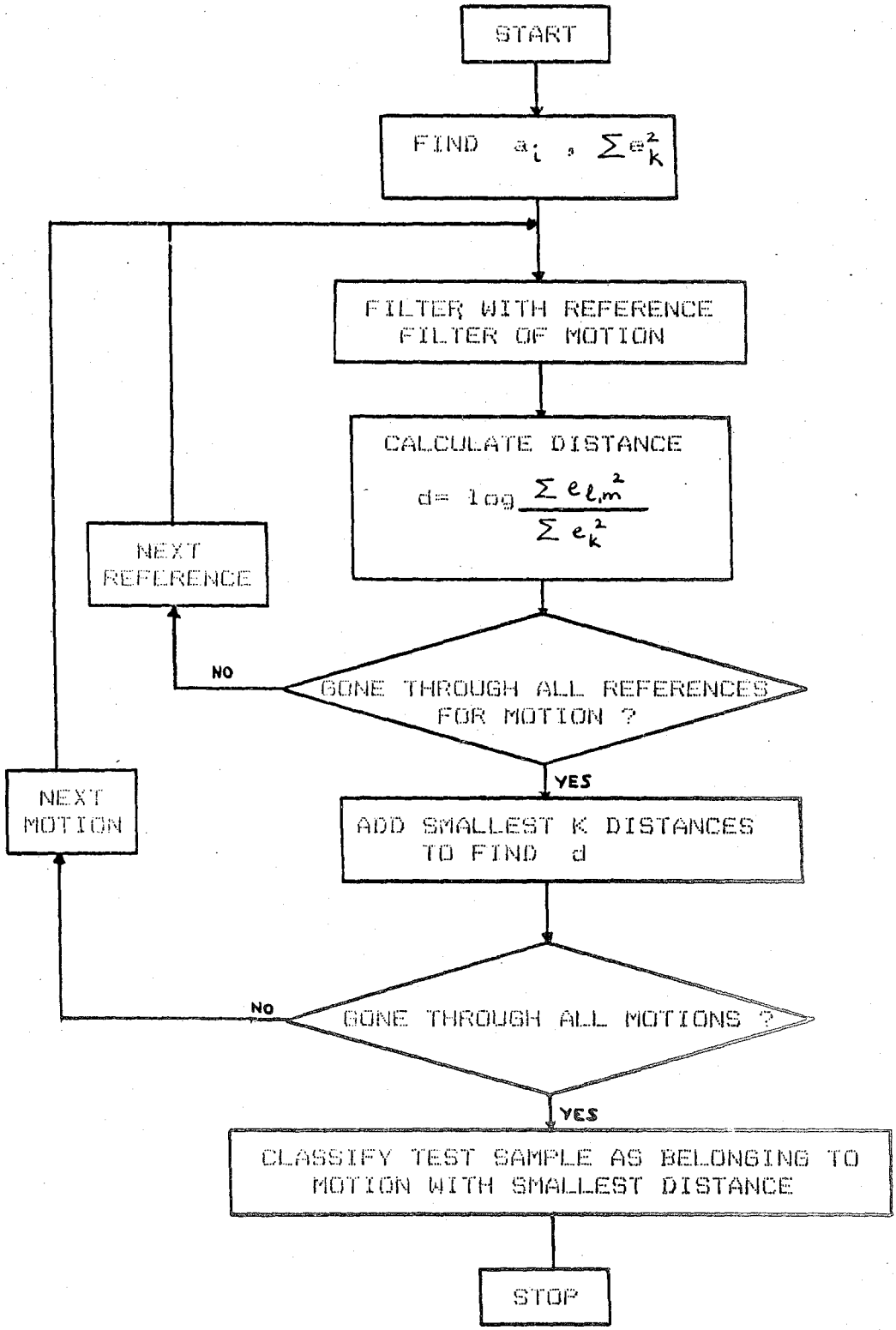

Fig.4.9 Steps in the classification of one frame

The results obtained in the classification tests run with 3-NN algorithm are summarized in the confusion matrix of Fig.4.10. The elements in each row of this matrix show, as percentages, the rate of deciding each function when the true function was what is written in the left hand side of each row. As would be expected from the scattered distribution of our AR data, our classification didn't give as good results as claimed by various authors [JS1,DG4,DD1]. We, again, want to emphasize the role of training in the clustering of class members and thus the performance of classification procedures.

	DECISION:		
TRUE MOTION	GRASP	WRIST FLEXION	ELBOW FLEXION
GRASP	43.5	56.5	0
WRIST FLEXION	2.9	95.7	1.4
ELBOW FLEXION	0	11.0	89.0

Figure 4.10 Confusion matrix for 3-NN algorithm

In the search for improvement of classification accuracy, we applied a form of smoothing in the classification scheme. The block diagram of this process which resembles median filtering is given in Fig 4.11. After two decisions had been made, each decision was altered by considering the decisions made for the two previous frames also. The decision was made on a majority vote of the three most recent classifications. The main aim in applying this filtering was to reduce the effect of momentary wrong

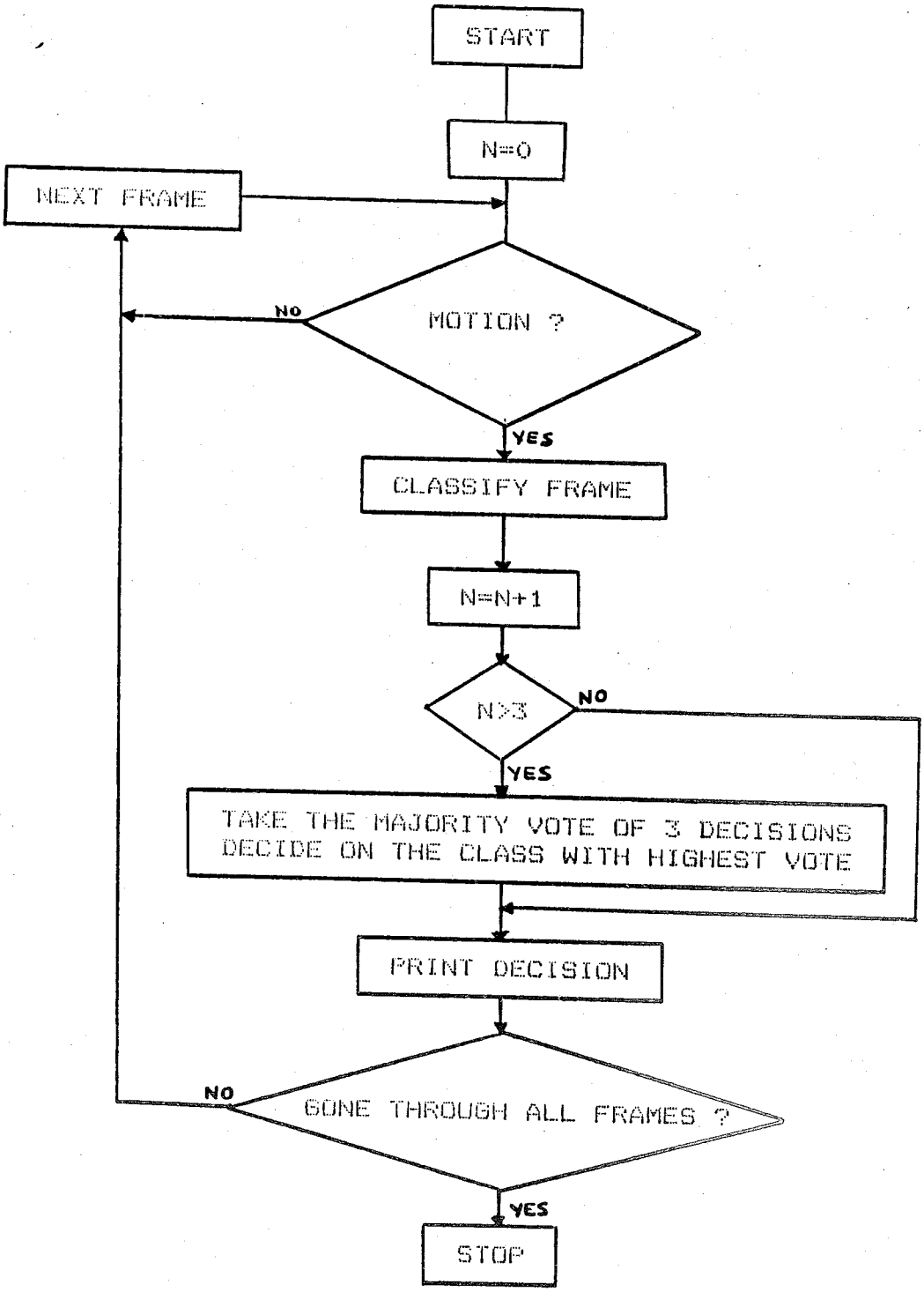

Fig 4.11 Steps in classification with median filtering

decisions. Note that in cases with very low success rates this procedure can end up in eliminating occasinal correct decisions among more frequent wrong ones. This method also introduces a delay of one frame in the case of insistent wrong decisions or a true change in the limb function executed. However a delay of one frame which amounts to 0.09 seconds is not crucial for our purposes especially when weighed against the improvement obtained. Also such a delay is insignificant when compared to the time constants of the mechanical drive. The confusion matrix obtained for the same reference library and the same test set evaluated with a 3-NN algorithm incorporating median filtering is given in Fig.4.12. The results reflect an average improvement of 5.1%.

TRUE MOTION	DECISION GRASP	WRIST FLEXION	ELBOW FLEXION
GRASP	47.1	52.9	0
WRIST FLEXION	2.1	97.9	0
ELBOW FLEXION	0	5.2	94.9

Figure 4.12 Confusion matrix for 3-NN algorithm with median filtering

Classification was also run with K=1 in the KNN algorithm. This value of K reduces the algorithm to the NN algorithm. The results obtained are summarized in the confusion matrix of Fig.4.13. The same condensed reference library and the same test set as for the 3-NN algorithm was

TRUE MOTION	DECISION GRASP	WRIST FLEXION	ELBOW FLEXION
GRASP	48.7	57.3	0
WRIST FLEXION	0	88.1	11.9
ELBOW FLEXION	0	5.4	94.6

Figure 4.13 Confusion matrix for NN algorithm

used. Though performance is somewhat impaired for wrist flexion, success rate for grasp is improved. This can be a desirable change as it brings more symmetry on the confusion matrix and decreases the bias towards wrist flexion in case of grasp motion. The NN might also be preferred to KNN as it requires less operations per sample and, therefore is somewhat faster.

V. CONCLUSIONS

In this study we have shown that there exists in the single channel of EMG sufficient information to discriminate between three degrees of freedom. This information can be extracted from the AR coefficients of the EMG signal with the K-nearest neighbor ornearest neighbor decision rules. Using a condensed reference library we obtained an overall success rate of 80.4% for an untrained subject. Employing median filtering in the classification routine introduced an improvement of 5.1%.

Performance in our study has been limited by the absence of subject training. With subject training and the use of more professional equipment in data acquisition, we expect considerable improvement in the final classification. As remarked in Chapter 4, electrode locations, sampling frequency, and model order are aspects to be investigated for the enhancement of success rate. Different identification algorithms can be tested to see if they improve discrimination success or speed. Various values of k in the KNN algorithm, frame size,and overlapping ratio are other points that can be explored. Considering the total distance of n-consecutive frames in the classification algorithm can also be tried.

Safety threshods can be introduced to prevent a decision to be made unless the similarity between the test sample and the reference is below a predetermined level. Note that performing a wrong limb function may be more frustrating for the amputee than performing no function at all. Prior statistics about the occurence of limb functions may be useful in enhancing speed and accuracy by causing only the functions with high probability of occuring after a certain function to be considered.

5.1 Future Studies

Since the main application of the EMG classification scheme proposedis the control of powered prostheses or orthoses, in future studies emphasis should be placed on practical constraints. The most important consideration is obviously the on-line operation of the system. In the design of a anthropomorphically operating artificial limb, it would be desired to increase the degrees of freedom that can be controlled. Information about the force and speed of the motion to be executed would also important in a practical system. Combined motions, i.e., motions where two or more degrees of freedom are activated are another problem the classification algorithm has to be able to tackle. Transitions between motions where directly after the execution of one motion another is intended without returning to rest position in between have to be dealt with.

Power consumption, physical size, and weight of the electronics are other considerations the designers will be concerned with.

The application of myoelectric motion classification can also be extended to the stimulation of paralyzed muscles. In such a system, according to the motion information extracted from muscles, the paralyzed muscle is stimulated in a certain manner. If atrophy has not occurred, the external stimulation causes the contraction of the muscle in accordance with the intended motion. Obviously, this application calls for a deep understanding of the behavior of muscles.

APPENDIX

```
C-----------------------------------------------------------------------
C
C       A ROUTINE TO CONDENSE THE REFERENCE LIBRARY OF
C                                  K-NEAREST NEIGHBOR ALGORITHM
C
C       This routine gets the data from a disk by calling the subroutine
C       READ which provides a 4096-point zero-mean data series by means
C       of the common area SAMPLE.The data is analyzed in sliding frames
C       of 256 data points.Each frame shares 64 points with the previous
C       one. 128 points are discarded from beginning and end of file.
C
C       Each frame is first tested for motion by filtering with the rest
C       filter. If the rest filter gives a total squared prediction error
C       above the preset threshold value THRESH, it is concluded that the
C       frame corresponds to some motion.
C
C       For frames corresponding to motion, a classification is made with
C       the currently available reference library which is stored in the
C       virtual memory under the name VM:REF.LIB. The classification is
C       based on the k-nearest neighbor algorithm employing the Itakura-
C       Saito distance measure. The AR coefficients of the frame and the
C       residual energy is computed by the subroutine AUTO implementing
C       the PARCOR algorithm. Subroutine MOTION computes the sum of the
C       distances of the k nearest neighbors of a class to the test
C       sample. Among the total distances computed for each class, the
C       smallest is determined. The frame is classified as belonging to
C       the motion class that has generated the smallest distance. The
C       the frame elements are transfered to the subprograms by the
C       common area FRAME. The distances and the number of references of
C       each class is exchanged through the common area DSTNC.
C
C       If the classification is correct the frame is temporarily
C       discarded. If the classification is incorrect, the sample is
C       added to the reference library. This procedure is repeated
C       until either AR vectors of all the frames are transfered to the
C       reference library or a complete run
C       through the frames that are not in the library.
C
C-----------------------------------------------------------------------

        COMMON/SAMPLE/SAM(4096)
        COMMON/DSTNC/DISTG(10),N(10)
        COMMON/FRAME/X(256)
        DIMENSION AT(5),AR(5),N2(10),MIST(10)
        THRESH=1.0E-3           !motion threshold
        KNEAR=3                 !# of nearest neighbors considered
        NCLASS=3                !# of motions to be classified
        DO 48 I=1,NCLASS
        MIST(I)=0               !times motion I is decided upon
48      CONTINUE
        TYPE *,'ENTER MOTION # AND FILE #'
        ACCEPT *,MSAM,IFILE
        NREC=MSAM*2000          !record showing lib size for motion
```

```
C
C       GET SAMPLES FROM DISK
C
        CALL READ
        OPEN (UNIT=3,NAME='VM:REF.LIB',TYPE='OLD',ACCESS='DIRECT',
        1RECORDSIZE=1)
C
C       REPEAT THE CONDENSING PROCEDURE FOR EACH FRAME
C
        DO 1010 NFRAME=1,20
        TYPE *
        TYPE *
        TYPE *,'FRAME=',NFRAME
        NBEG=(NFRAME-1)*192+129 !frame begins
        NEND=NBEG+255           !frame ends
C
C       TEST IF THE FRAME REPRESENTS MOTION
C
        TOTER=0.
        DO 20 I=1,256
        EST=.8*SAM(NBEG+I-2)+.2*SAM(NBEG+I-3)   !rest estimate
        X(I)=SAM(NBEG+I-1)      !fills frame
        ERR=X(I)-EST            !prediction error
        TOTER=TOTER+ERR*ERR     !total prediction error over the frame
20      CONTINUE
        TYPE *,'FOR REST TOTAL PREDICTION ERROR=',TOTER
        IF (TOTER.LT.THRESH) GOTO 1005  !checks motion threshold
C
C       IN CASE OF MOTION TRY TO MAKE A CLASSIFICATON
C       FIRST CHECK IF THE FRAME IS ALREADY IN THE LIBRARY
C
        NKEEP=(NSAM+4)*2000     !rec showing last rec in area where
C                               file # and frame # are recorded
C                               for frames transfered to ref lib
C                               for each motion
        READ(3'NKEEP)N2(NSAM)   !N2(NSAM) is the last rec of area
        NOFF=N2(NSAM)-(NSAM+3)*2000  !end of area for previous motion
C
C       READ FILE AND FRAME NUMBERS OF REFERENCES IN THE LIBRARY
C       AND CHECK IF THEY ARE THE SAME AS THE ONES FOR THE CURRENT
C       FRAME
C
        DO 4 I=NOFF+1,N2(NSAM),2
        READ(3'I)IFI
        IF (IFI.NE.IFILE) GOTO 4
        READ(3'I+1)NF
        IF (NF.NE.NFRAME) GOTO 4
        TYPE *,'FRAME ALREADY IN LIBRARY'
        GOTO 1010
4       CONTINUE
C
C       COMPUTE THE AR COEFFICIENTS OF THE FRAME
C
        CALL AUTO(256,X,4,AT,ALPHA)
        TYPE *,'AR''s:',(AT(I),I=1,5)
```

```
C       FIND THE TOTAL DISTANCES OF
C                   THE K-NEAREST NEIGHBORS OF EACH CLASS
C
        DO 25 MOT=1,NCLASS
        CALL MOTION(ALPHA,KNEAR,MOT)
25      CONTINUE
        TYPE *,'DISTANCE OF EACH MOTION FROM TEST SAMPLE:'
        TYPE *,(DISTG(I),I=1,NCLASS)
C
C       FIND THE NEAREST CLASS
C             BY COMPARING THE TOTAL DISTANCES OF ALL CLASSES
C
        GMIN=DISTG(1)
        MOT=1
        DO 30 I=2,NCLASS
        IF (DISTG(I).GE.GMIN) GOTO 30
        GMIN=DISTG(I)
        MOT=I
30      CONTINUE
        MIST(MOT)=MIST(MOT)+1
        TYPE *
        TYPE *,'I CLASSIFY SAMPLE AS BELONGING TO MOTION',MOT
        TYPE *
C
C       TEST IF THE CLASSIFICATION IS CORRECT
C
        IF (MOT.EQ.MSAM) GOTO 2020
C
C       IF THE CLASSIFICATION IS INCORRECT,ADD THE TEST SAMPLE TO
C       REFERENCE LIBRARY
C
        TYPE *,'CLASSIFICATION WAS INCORRECT SO I''M ADDING SAMPLE
        1TO REFERENCE LIB OF MOTION',MSAM
        DO 35 I=1,4
        N(MSAM)=N(MSAM)+1
        WRITE(3'N(MSAM))AT(I+1) !record AR coefficients
35      CONTINUE
        WRITE(3'NREC)N(MSAM)    !record last record
C
C       RECORD FILE AND FRAME # OF SAMPLE PLACED IN LIBRARY
C
        N2(MSAM)=N2(MSAM)+1
        WRITE(3'N2(MSAM))IFILE  !record file #
        N2(MSAM)=N2(MSAM)+1
        WRITE(3'N2(MSAM))NFRAME !record frame #
        WRITE(3'NKEEP)N2(MSAM)  !record last rec
        GOTO 1010
2020    TYPE *,'CLASSIFICATION WAS CORRECT SO I''M TEMPORARILY
        1DISCARDING SAMPLE'
        GOTO 1010
1005    TYPE *,'FRAME DOESN''T CONTAIN MOTION'
1010    CONTINUE
        CLOSE(UNIT=3)
        TYPE *,'DECISIONS FOR EACH MOTION CLASS:'
        TYPE *,(I,MIST(I),I=1,NCLASS)
        STOP
        END
```

```
C-------------------------------------------------------------------------
      SUBROUTINE READ
C
C     A SUBROUTINE TO TRANSFER 4096 ZERO MEAN SAMPLES
C
C     This subroutine reads data from the file VM:DATA.DAT in the
C     virtual memory. It calculates the dc value and subtracts it from
C     each data point. It also calculates the variance and the zero-
C     crossings. The data is transfered through the common area SAMPLE
C     to the calling program.
      INTEGER*2 IA(4096)
      COMMON/SAMPLE/SAM(4096)
      OPEN(UNIT=2,NAME='VM:DATA.DAT',TYPE='OLD',RECORDSIZE=1,
     1ACCESS='DIRECT')
      DO 10 I=1,2048
      J=2*I-1
      READ(2'I)IA(J),IA(J+1)
      SAM(J)=FLOAT(IA(J)-2048)/2048.
      SAM(J+1)=FLOAT(IA(J+1)-2048)/2048.
10    CONTINUE
      CLOSE(UNIT=2)
      DO 20 I=1,4096
      DC=DC+SAM(I)            !calculate the dc
20    CONTINUE
      DC=DC/4096.
      VAR=0.
      IZC=0
      DO 30 I=1,4096
      SAM(I)=SAM(I)-DC        !subtract the dc
      VAR=VAR+SAM(I)*SAM(I)   !calculate the variance
      IF (SAM(I)*SAM(I-1).LE.0.) IZC=IZC+1 !calculate the zero-crossings
30    CONTINUE
      VAR=VAR/4096.
      WRITE(7,*)' DC=',DC,' VAR=',VAR,' ZC=',IZC
      RETURN
      END
```

```
      SUBROUTINE AUTO(N,X,M,A,ALPHA)
C     Inputs: N-# of data points
C             X(N)-Input data sequence
C             M-Order of filter(M<21)
C     Output: A-Filter coefficients
C             ALPHA-Residual energy
C
C     Written by: A.H. GRAY,Jr. and J.D. MARKEL
C
C     Program limitted to M>21 because of dimensions of R(.)
C
      DIMENSION X(1),A(1)
      DIMENSION R(31),RC(5)
      MP=M+1
      DO 20 K=1,MP
      R(K)=0.
      NK=N-K+1
      DO 10 NP=1,NK
      N1=NP+K-1
      R(K)=R(K)+X(NP)*X(N1)
10    CONTINUE
20    CONTINUE
      RC(1)=-R(2)/R(1)
      A(1)=1.
      A(2)=RC(1)
      ALPHA=R(1)+R(2)*RC(1)
      DO 50 MINC=2,M
      S=0.
      DO 30 IP=1,MINC
      N1=MINC-IP+2
      S=S+R(N1)*A(IP)
30    CONTINUE
      RC(MINC)=-S/ALPHA
      MH=MINC/2+1
      DO 40 IP=2,MH
      IB=MINC-IP+2
      AT=A(IP)+RC(MINC)*A(IB)
      A(IB)=A(IB)+RC(MINC)*A(IP)
      A(IP)=AT
40    CONTINUE
      A(MINC+1)=RC(MINC)
      ALPHA=ALPHA+RC(MINC)*S
      IF (ALPHA)70,70,50
50    CONTINUE
60    RETURN
70    CONTINUE
      WRITE(7,9999)
9999  FORMAT(1X,' WARNING - SINGULAR MATRIX - AUTO')
      TYPE *,'ALPHA=',ALPHA
      GOTO 60
      END
```

```
C-----------------------------------------------------------------------
      SUBROUTINE MOTION(ALPHA,KNEAR,MOT)
C
C     A SUBROUTINE TO CALCULATE THE TOTAL DISTANCE OF THE K-NEAREST
C     NEIGHBORS OF THE MOTION CONSIDERED
C
C     This subroutine receives the elements of the frame through the
C     common area FRAME. A set of reference coefficients are read from
C     the reference library. The total squared prediction error
C     that results when the frame is fed to the filter with these
C     coefficients is calculated. The procedure is repeated for all
C     reference coefficient sets for the motion in question. Among
C     the errors calculated for all  references, the K smallest ones
C     are determined. These are multiplied out and the logarithm is
C     taken. From this quantity K times the logarithm of the prediction
C     error, calculated in the identification of the frame, is
C     subtracted. This yields the overall distance of the frame to the
C     particular motion class. The overall distance of the class and
C     the last record for the motion in the reference library is
C     transfered to the main program by the common area DSTNC.
C
C-----------------------------------------------------------------------
      COMMON/FRAME/X(256)
      COMMON/DSTNC/DISTG(10),N(10)
      DIMENSION TOTER(200),AR(5),RMIN(10)
      IREC=MOT*2000           !rec showing last rec for motion
      READ(3'IREC)N(MOT)      !read last rec #
      ILAST=(N(MOT)-(MOT-1)*2000)/4   !# of reference sets
      AR(1)=1.
      DO 1000 IREF=1,ILAST
C
C     READ A SET OF REFERENCES
C
      NOFF=(MOT-1)*2000+(IREF-1)*4
      DO 5 K=1,4
      IREAD=NOFF+K
      READ(3'IREAD)AR(K+1)
5     CONTINUE
C
C
C     CALCULATE TOTAL PREDICTION ERROR FOR EACH REFERENCE SET
C
88    TOTER(IREF)=0.
      DO 20 I=5,256
      ERR=0.
      DO 30 K=1,5
      ERR=ERR+AR(K)*X(I-K+1)          !prediction error
30    CONTINUE
      TOTER(IREF)=TOTER(IREF)+ERR*ERR !total squared prediction error
20    CONTINUE
1000  CONTINUE
```

```
C
C       FIND K NEAREST NEIGHBORS
C       AND CALCULATE TOTAL DISTANCE OF THE MOTION CLASS TO TEST FRAME
C
        GDIST=1.
        DO 1 J=1,KNEAR
        RMIN(J)=TOTER(J)        !RMIN will be the ordered array
        INDEX=J
        DO 2 I=J+1,ILAST
        IF (TOTER(I).GE.RMIN(J)) GOTO 2
        RMIN(J)=TOTER(I)
        INDEX=I
2       CONTINUE
        TOTER(INDEX)=TOTER(J)
        GDIST=GDIST*RMIN(J)     !product of k nearest neighbors
1       CONTINUE
        GDIST=ALOG10(GDIST)-(KNEAR*ALOG10(ALPHA))  !overall distance
        DISTG(NOT)=GDIST        !overall distance of motion to frame
        RETURN
        END
```

```
C-----------------------------------------------------------------------
C       A ROUTINE TO CLASSIFY A SIGNAL
C                              WITH K-NEAREST NEIGHBOR ALGORITHM
C
C       This program classifies test frames as belonging to certain
C       motion classes with the K-Nearest Neighbor algorithm. The data
C       is read from a disk by calling subroutine READ which transfers
C       4096 zero-mean data points through common area SAMPLE.
C       The data is analyzed in 256-point frames which overlap by 64
C       points. 128 samples are discarded from beginning and end.
C       Each frame is first tested for motion by feeding to the rest
C       filter. If the resulting prediction error is above the
C       threshold value THRESH, it is conluded that the frame contains
C       motion.
C       For each frame containing motion, the AR coefficients and the
C       residual energy are computed with the subroutine AUTO. MOTION
C       subroutine computes the overall distance of one motion to the
C       test frame. First the frame is fed into filters described by
C       each of the reference coefficient sets. The total errors are
C       computed. The k-smallest ones are determined and multiplied out.
C       The logarithms are taken and K times the logarithm of the
C       residual energy is subtracted. This gives the overall distance
C       of the particular motion class to test frame. This distance and
C       the reference numbers for each motion  transfered through the
C       common area DSTNC.
C       Among all distances calculated for all motion classes, the
C       the class yielding the smallest distance is determined and the
C       test frame is classified as belonging to that class.
C       The classification is tested for correctness and the results
C       for the total file is displayed.
C-----------------------------------------------------------------------
        COMMON/SAMPLE/SAM(4096)
        COMMON/DSTNC/DIST6(10),N(10)
        COMMON/FRAME/X(256)
        DIMENSION AT(5),AR(5),N2(10),MIST(10)
        THRESH=1.0E-3           !motion threshold
        KNEAR=3                 !# of nearest neighbors considered
        NCLASS=3                !# of motions to be classified
        DO 46 I=1,NCLASS
        MIST(I)=0               !times each class is decided upon
46      CONTINUE
        NCOR=0
        INCOR=0
        TYPE *,'ENTER MOTION #'
        ACCEPT *,NSAM
        NREC=NSAM*200           !record showing lib size for motion
C
C       GET SAMPLES FROM DISCETTE
C
        CALL READ
        OPEN (UNIT=3,NAME='VM:REF.LIB',TYPE='OLD',ACCESS='DIRECT',
        1RECORDSIZE=1)
```

```
C
C       REPEAT PROCEDURE FOR EACH FRAME
C
        DO 1010 NFRAME=1,20
        TYPE *
        TYPE *
        TYPE *,'FRAME=',NFRAME
        NBEG=(NFRAME-1)*192+129 !frame begins
        NEND=NBEG+255           !frame ends
C
C       TEST IF THE FRAME REPRESENTS MOTION
C
        TOTER=0.
        DO 20 I=1,256
        EST=.8*SAM(NBEG+I-2)+.2*SAM(NBEG+I-3)   !rest estimate
        X(I)=SAM(NBEG+I-1)      !fills frame
        ERR=X(I)-EST            !prediction error
        TOTER=TOTER+ERR*ERR     !total prediction error over the frame
20      CONTINUE
        TYPE *,'FOR REST TOTAL PREDICTION ERROR=',TOTER
        IF (TOTER.LT.THRESH) GOTO 1005  !checks motion threshold
C
C       IN CASE OF MOTION FIND AR COEFFICIENTS
C
        CALL AUTO(256,X,4,AT,ALPHA)     !find AR coefficients
        TYPE *,'AR''s:',(AT(I),I=1,5)
C
C       FIND THE TOTAL DISTANCES OF
C                   THE K-NEAREST NEIGHBORS OF EACH CLASS
C
        DO 25 MOT=1,NCLASS
        CALL MOTION(TESTER,KNEAR,MOT)
25      CONTINUE
        TYPE *,'DISTANCE OF EACH MOTION FROM TEST SAMPLE:'
        TYPE *,(DISTG(I),I=1,NCLASS)
C
C       FIND THE NEAREST CLASS
C
        GMIN=DISTG(1)
        MOT=1
        DO 30 I=2,NCLASS
        IF (DISTG(I).GE.GMIN) GOTO 30
        GMIN=DISTG(I)
        MOT=I
30      CONTINUE
        MIST(MOT)=MIST(MOT)+1
        TYPE *
        TYPE *,'I CLASSIFY SAMPLE AS BELONGING TO MOTION',MOT
        TYPE *
```

```
C
C       TEST IF THE CLASSIFICATION IS CORRECT
C
        IF (NOT.EQ.MSAM) GOTO 2020
        TYPE *,'CLASSIFICATION WAS INCORRECT'
        INCOR=INCOR+1
        GOTO 1010
2020    TYPE *,'CLASSIFICATION WAS CORRECT'
        NCOR=NCOR+1
        GOTO 1010
1005    TYPE *,'FRAME DOESN''T CONTAIN MOTION'
1010    CONTINUE
        TYPE *,'#OF DECISIONS FOR EACH CLASS:'
        TYPE *,(I,NIST(I),I=1,NCLASS)
        TYPE *,'# OF CORRECT DECISIONS=',NCOR
        TYPE *,'# OF INCORRECT DECISIONS=',INCOR
        CLOSE(UNIT=3)
        STOP
        END
```

REFERENCES

[BJ11] B. H. Jansen, "Analysis of biomedical signals by means of linear modeling, " CRC Crit. Rev. Biomed. Eng. , vol. 12, pp. 343-392, 1985.

[CD11] C. J. De Luca, " Physiology and mathematics of myoelectric signals, " Trans. Biomed. Eng., vol. BME-26, pp. 313-325, 1979.

[DC11] D. S. Childress, " Myoelectric control of powered prosthesis, " Eng. Med. Bio. Mag. pp. 23-25, 1982.

[DD11] D. C. Dening, F. G. Gray, and R. M. Haralick, " Prosthesisis control using a nearest neighbour electromyogrphic pattern classifier, " IEEE Trans. Biomed.Eng. vol. BME-30, pp.356-360,1983.

[DG11] D. Graupe and W. K. Cline, " Functional separation of EMG signals via ARMA identification methods for prosthesis control purposes, " IEEE Trans. Syst, Man & Cyb. , vol. SMC-5 pp. 252-259, 1975.

[DG21] D. Graupe, D. J. Krause, and J. B. Moore, "Identification of Autoregressive Moving-Average Parameters of Time-Series," IEEE Trans. Automat. Contr. , vol. AC-20, pp. 104-107, 1975.

[DG83] D. Graupe, J. Magnussen, and A. M. Beex, " A microprocessor system for multifunctional control of upper-limb prosthesis via myoelectric signal identification," IEEE Trans. Automat. Contr. , vol. AC-23, pp.538-544, 1978.

[DG84] D. Graupe, J. Salahi, and D. Zhang, " Stochastic analysis of myoelectric temporal signatures for multifunctional single-site activation of prosthesis and orthoses," J. Biomed. Eng. , vol. 7, 1985.

[DG85] D. Graupe, Time Series Analysis, Identification and Adaptive Filtering. Florida: Krieger, 1984.

[EK1] E. Kwatny, D. H. Kwatny, and H. G. Kwatny, " An application of signal processing techniques to the study of myoelectric signals, " IEEE Trans. Biomed. Eng. , vol. BME-17,pp. 303-313,1970.

[ES1] E.Shwedyk, R. Balasubramanian, and R. M. Scott, "A nonstationary model for the electromyogram, " IEEE Trans. Biomed. Eng. vol. BME-24, pp. 417-424, 1977.

[GA1] G. C. Agarwal and G. L. Gottlieb, " Mathematical modeling and simulation of the postural control loop," CRC Crit. Rev. Biomed. Eng. , vol.8, pp. 93-134, 1982.

[GS1] G. N. Saridis and T. P. Gootee, " EMG pattern analysis and classification for a prosthetic arm, " IEEE Trans. Biomed. Eng. , vol. BME-29, pp. 403-412,1982.

[GS2] G. N. Saridis and H. E. Stephanou, " A hierarchical approach to the control of a prosthetic arm, " IEEE Trans. Syst, Man, Cybern. , vol. SMC-7, pp.407-420,1977.

[JB1] J. V. Basmajian, Muscles Alive. Their Function Revealed by Electromyography, 4th ed. Baltimore: Williams & Wilkins, 1978.

[JM1] J. Makhoul, " Linear prediction: A tutorial review,"Proc. IEEE, vol 63 pp. 561-580, 1975.

[JS1] J. Salahi, Identification and Recognition Applied to Biomedical Signals, Ph. D. Thesis, Illinois Institute of Technology, Illinois, 1981.

[KF1] K. Fukunaga, Introduction to Statistical Pattern Recognition. New York: Academic Press, 1972.

[PD1] P. C. Doerschuk, D. E. Gustafson, and A. S. Willsky, " Upper extremity limb function discrimination using EMG signal analysis, " IEEE Trans. Biomed Eng. vol. BME-30, pp. 8-29, 1983.

[PH1] P. E. Hart, " The condensed nearest neighbor rule," IEEE Trans. Inform. Theory, vol. IT-14,

pp. 512-516, 1968.

[RBS1] R. B. Stein and M. Walley, " Functional comparison of upper extremity amputees using myoelectric and conventional prosthesis, " Arch. Phys. Med. Rehabit. , vol 64, 243-247, 1983.

[RNS1] R. N. Scott, P. A. Parker, and V. A. Dunfield, " Myoelectric control, " IEE Medical Electronics Monographs, Ed. D. W. Hill, B. W. Watson. Exeter: Peter Peregrinus Ltd. 1977.

[RW1] R. W. Wirta, D. R. Taylor, and F. R. Finley, " Pattern recognition arm prosthesis: A historical perspective - a final report, " Bulletin of Prosthetics Research, Fall 1978.

[SL1] S. Lee and G. N. Saridis, " The control of a prosthetic arm by EMG pattern recognition, " IEEE Trans. Automat. Contr. , vol AC-29, 1984

[UJ1] U. Johnsson, P. Herberts, L. Korner, " Advances in myoelectric control of multifunctional prosthesis" Computer - Aided Electromyography, Prog. Clin. Neurophysiol., vol. 10, ED. J. E. Desmedt, pp. 297-307: Basel: Karger, 1983.

[WM1] W. S. Meisel, Computer - Oriented Approaches to Pattern Recognition. New York: Academic Press, 1972.