Multivariate Autoregressive Models for Classification of Spontaneous Electroencephalogram During Mental Tasks[1]

Charles W. Anderson
Department of Computer Science
Colorado State University
Fort Collins, CO 80523
anderson@cs.colostate.edu

Erik A. Stolz
Department of Electrical Engineering
Colorado State University
Fort Collins, CO 80523

Sanyogita Shamsunder
Department of Electrical Engineering
Colorado State University
Fort Collins, CO 80523

Abstract

This article explores the use of scalar and multivariate autoregressive (AR) models to extract features from the human electroencephalogram (EEG) with which mental tasks can be discriminated. This is part of a larger project to investigate the feasibility of using EEG to allow paralyzed persons to control a device like a wheelchair. EEG signals from four subjects were recorded while they performed two mental tasks. Quarter-second windows of 6-channel EEG were transformed into four different representations: scalar AR model coefficients, multivariate AR coefficients, eigenvalues of a correlation matrix, and the Karhunen-Loève transform of the multivariate AR coefficients. Feature vectors defined by these representations were classified with a standard, feedforward neural network trained via the error backpropagation algorithm. The four representations produced similar results, with the multivariate AR coefficients performing slightly better and more consistently with an average classification accuracy of 91.4% on novel, untrained, EEG signals.

[1]Please send all correspondence to Charles Anderson. This work was supported by the National Science Foundation through grant IRI-9202100.

I. Introduction

The purpose of the work described in this paper is to investigate the practicality of using a multivariate autoregressive (AR) model to extract classifiable features from human EEG. The research is aimed at disabled persons who are unable to communicate via normal physical methods, but who do have complete control over their mental faculties. Some success has been achieved in classifying EEG for this purpose, but either evoked responses [1] or biofeedback training [2] are involved. For this project it was proposed that several easily implemented mental tasks could be used as an alphabet that a person could use to control a device like a wheelchair. Our work is a continuation of the work of Aunon and Keirn [3], reviewed in the following section.

The success of this project depends on finding a signal representation that is as small as possible (for speed of processing and improved generalization) and yet contains the information needed to accurately classify different mental states. Here, scalar (univariate) and multivariate AR models were used to define representations. Various features based on these models were classified with a multilayer, feedforward, neural network using the error back-propagation training algorithm. Discrimination was performed between a single pair of tasks. A multivariate AR representation resulted in the best classification percentages when averaged over all subjects.

The remainder of this article is organized as follows. Section II summarizes pertinent literature related to the EEG recognition problem and the multivariate AR model. Techniques for estimating multivariate AR models are described in Section III. Classification experiments, the neural network, and results of several methods for selecting the order of the AR models are described in Section IV. Section V reports classification results, which are summarized in Figure 8. Conclusions are in Section VI.

II. Background

Rarely has a multivariate model extracted from multiple data channels been investigated in the study of EEG recognition problems. It is much more common for a single channel of data to be used. For example, Tsoi, So, and Sergejew [4] recorded one channel of EEG, estimated scalar AR models of the signals, and used the AR model coefficients to differentiate between psychiatric disorders. Data was recorded from normal subjects, subjects diagnosed as suffering from severe obsessive compulsive disorder, and subjects diagnosed as suffering from severe schizophrenia. Feature vectors were formed from a single channel by averaging AR model coefficients of order 8 over 250 consecutive seconds. The AR coefficients were calculated by solving the Yule-Walker equations, reviewed in Section III. The feature vectors were fed into a multilayer neural network with 8 input units, 15 hidden units, and 3 output units. The trained network was then used to classify 14 cases of novel test data. The network was able to correctly classify all normal cases, while missing one each from the obsessive compulsive and schizophrenia cases.

In another paper, Roberts and Tarassenko [5] recorded one channel of data to cluster sleep stages using AR model coefficients. The purpose was to quantitatively investigate the number of different human sleep states. The parameterization was performed using a 10^{th} order Kalman filter, and the coefficients were averaged over one second of time. These 10-dimensional feature vectors were then clustered with a Kohonen self-organizing neural network. It was found that there are eight different primary clusters and three types of transition trajectories among these eight clusters which correspond to states of wakefulness, dreaming sleep (REM), and deep sleep.

Single channel recordings are somewhat typical of sleep research, because there are noticeable changes in brainwave activity during the different stages of sleep. For other problems, however, the multichannel EEG must be recorded because single channel brainwaves do not provide enough information to be used alone. One must look at activity distributed over the entire scalp in order to detect brain state changes. For example, Ramadan, Ropella, Myklebust, Goldstein, Feng and Flynn [6] recorded from nine EEG channels to discriminate between normal and dyslexic readers. The relative power in four common frequency bands of the EEG spectral density—delta (0-3 Hz), theta (4-7 Hz), alpha (8-13 Hz), and beta (14-20 Hz)—and the root mean square and the probability density function of the EEG amplitude were computed for each channel and concatenated together to form the feature vector. The data was recorded from three children during a resting state and a reading task. One child was a normal reader, another type-1 dyslexic, and the third, type-2 dyslexic. A multilayer perceptron was trained with the backpropagation algorithm on data from all classes during both eyes closed and eyes open conditions. The neural net had 54 inputs, 61 hidden units, and 4 output units. After training, the neural net accurately predicted 87% of the novel data. This study shows that there may exist a quantitative difference between dyslexic and normal EEG. However, no control subjects were tested to see if the classification might be based on the basic EEG of the three children.

Papadourakis, Micheloyannis, Bebis, and Giachnakis [7] applied the same type of analysis to the detection of persons exposed to organic solvents. Data was recorded from a group of control subjects and a group of car painters. None of the car painters was diagnosed with a severe illness. EEG was recorded from 12 channels during a resting condition and while moving the fingers of the right hand. Power spectra differences between resting and finger movement conditions for the alpha band were considered meaningful parameters, along with the coherence spectra among 10 pairs of channels. A multilayer perceptron with the backpropagation learning rule was trained to discriminate between the two groups using three different sets of feature vectors: the power spectra values alone, the coherence spectra powers alone, and the combined powers. The size of the neural network was not given. It was found that the best classification of 69% occurred using the power spectra values alone. It was concluded that the training set size was not large enough for the network to be able to generalize well to novel data.

In an experiment similar to our own, Tumey, Morton, Ingle, Downey, and Schnurer [8] investigated the classification of EEG during three different mental tasks. The tasks included eyes open and alert, eyes closed and alert, and eyes closed while the subject performed a visualization task. Data was recorded from two channels. The data from each channel was fed into a phase-space algorithm where the signal was plotted against a lagged version (lag values were not given) to generate an attractor pattern over 10 seconds. A box counting algorithm was employed to quantize the attractor into bins and the counts of each bin were composed into a feature vector. A backpropagation neural network was trained on half of the feature vectors. The size of the network was not given. The trained network was then applied to the remaining half of the data. It was found that 100% of the test vectors were classified correctly. It was also shown that the same net could still classify the subject's EEG days after the original experiment.

Work reported in this article is a continuation of a project initiated by Aunon and Keirn [3]. In their study, they investigated the classification of five different mental tasks: a baseline task, mental multiplication, geometric figure rotation, mental letter composing, and visual counting. Data was recorded from seven subjects using six channels. Features were first extracted from spectral estimates, calculated from both the Fourier transform of the windowed autocorrelation function and a scalar AR model. Features extracted were the asymmetry ratios and power values for each channel from the four frequency bands mentioned earlier—delta, theta, alpha, and beta. Asymmetry ratios were taken across all right to left combinations of leads and are given by $(R-L)/(R+L)$, where R is a power value from a certain frequency band of a right hemisphere lead, and L is defined similarly for a left hemisphere lead. A second set of features was generated from the AR coefficients themselves concatenated together from all channels. Features were extracted from one quarter-second (and two-second) windows per trial, for ten trials of each of the tasks. The classifier in this case was a quadratic Bayesian classifier. By averaging results over five subjects, it was found that all task pairs could be reliably discriminated 84.6% of the time using the AR coefficients as features. The asymmetry ratio features using the windowed autocorrelation function or the AR method were classified just slightly worse.

The common denominator in all of the above experiments is a reliance on feature vectors constructed from single channel features alone or multiple channel features concatenated together. To date, very little work has been performed regarding multivariate models and the EEG recognition problem. However, a few researchers have studied multivariate AR models applied to other areas of EEG research.

Franaszczuk, Blinowska, and Kowalczyk [9] constructed a multivariate AR model to study the synchronization of brain structures. The AR parameters were estimated from the multivariate analog of the Yule-Walker equations, described in Section IV. An AR model of order 6 was found to be acceptable for parameterizing four channels of data recorded at 205 Hz. Auto- and cross-spectra were estimated from the multivariate AR model, and from these, coherence values were calculated. Partial coherences were found to be the proper measure of synchronization, because they remove the effects of other channels. Analysis of the multichannel EEG with this method could be used in researching the synchronization of brain structures, the degree of coupling between channels, the estimation of phase delays, and eventually the direction of spreading of brain activity.

Similarly, Rappelsberger and Petsche [10] constructed a multivariate AR model from rabbit EEG during drug induced epileptic seizures. Models were estimated from 16 channels of data sampled at 64 Hz by solving the Yule-Walker equations. The model orders were arbitrarily selected to be 11 or 13. Spectra and coherences were calculated from the AR coefficients and plotted versus time. From the phase delays they concluded that 8.8 Hz activity is generated in one corner of the electrode grid and gradually spreads to other areas during the onset of a seizure.

Spatial smoothing of the EEG from a multivariate AR model was the topic of another paper by Jimenez, Biscay, and Montoto [11]. Data was recorded from a normal subject and a subject with a tumor. Models were estimated from 16 channels using orders from 6-12 via the Yule-Walker equations. Cross-spectra were estimated from the model, and then a spherical splines technique was used to smooth the spectral estimates to give a detailed frequency map of the scalp. From the maps, one can see a decreased area of energy in the region of the tumor in the abnormal subject.

In another sequence of papers, Gersch and Yonemoto [12, 13], constructed a multivariate AR model from multiple channels of EEG in order to cluster sleep stages. Multivariate AR models were estimated from two channels of data sampled at 33 Hz using Whittle's recursive procedure [14]. It was found that a maximum order of 6 was necessary to parameterize the data. Eigenvalues were calculated from the state-space system matrix built with the AR coefficient matrices. They showed that the eigenvalues could separate different sleep stages. They did not attempt any classification experiments.

Other than [12, 13], there is a conspicuous lack of research into the extraction and use of multivariate AR features for EEG pattern recognition. The possibility that multivariate AR models would capture cross-channel information relevant to our classification problem motivated the experiments reported here.

III. The Multivariate AR Model

In this section we review multivariate AR models and present two methods for estimating the model coefficients. Multivariate AR models have been used for applications in control, communications, and sensor-array processing [15, 16, 17, 18].

A. Yule-Walker Estimates

To apply the AR framework to EEG signals, we assume that a linear filter describes the process of EEG generation and that this filter is fed with a white noise signal. We also assume that the characteristics of the output can be determined by the current and past time values of the output. A common form of a multivariate linear parametric model is given by the autoregressive moving-average (ARMA) model which has the following form (see Chapter 9, Section 4 in [19]):

$$\begin{aligned}&\mathbf{A}(0)\mathbf{x}(k) + \mathbf{A}(1)\mathbf{x}(k-1) + \mathbf{A}(2)\mathbf{x}(k-2) + \cdots + \mathbf{A}(p)\mathbf{x}(k-p) = \\ &\mathbf{B}(0)\mathbf{e}(k) + \mathbf{B}(1)\mathbf{e}(k-1) + \cdots + \mathbf{B}(q)\mathbf{e}(k-q) \qquad (1)\end{aligned}$$

where $q \leq p$. If we simplify this by setting $\mathbf{B}(i) = \mathbf{0}$ for $i = 1, \ldots, q$ and letting $\mathbf{A}(0), \mathbf{B}(0) = \mathbf{I}$), we are left with the so-called multivariate autoregressive, or AR model. In the multivariate AR model, the current observation is represented as a weighted linear combination of past observations plus a random, uncorrelated input:

$$\mathbf{x}(k) = -\mathbf{A}(1)\mathbf{x}(k-1) - \mathbf{A}(2)\mathbf{x}(k-2) + \cdots - \mathbf{A}(p)\mathbf{x}(k-p) + \mathbf{e}(k) \tag{2}$$

where $\mathbf{x}(k)$ is a d-dimensional vector of data at time k, and $\mathbf{e}(k)$ is a d-dimensional vector of unobserved random input. The $\mathbf{A}(i),\ i = 1, \ldots, p$, are the $d \times d$ matrices of autoregression coefficients to be estimated from $\mathbf{x}(k),\ k = 1, \ldots, N$, and p is the AR model order.

The linear prediction of the current sample vector can be given by

$$\hat{\mathbf{x}}(k) = -\sum_{i=1}^{p} \mathbf{A}(i)\mathbf{x}(k-i). \tag{3}$$

The prediction error or residue of this estimated value is given by

$$\mathbf{e}(k) = \mathbf{x}(k) - \hat{\mathbf{x}}(k). \tag{4}$$

Our problem is to determine the coefficients of the filter such that a function of the prediction error is minimized.

We can estimate the parameter matrices by way of the least squares procedure. This method results in parameter matrices that minimize the sum of the squared residue samples, given by:

$$J(N) = \sum_{k=p+1}^{N} \mathbf{e}^T(k)\mathbf{e}(k) \tag{5}$$

This is minimized by equating the partial derivatives of $J(N)$ with respect to $\mathbf{A}(i)$ to zero and solving for $\mathbf{A}(i)$. To do this we need to perform some matrix derivatives. Using the procedures given in [20] we find:

$$\frac{\partial(\mathbf{e}^T(k)\mathbf{e}(k))}{\partial\langle\mathbf{A}(i)\rangle} = \frac{\partial(\mathbf{e}^T(k))}{\partial\langle\mathbf{A}(i)\rangle}\mathbf{e}(k) + \mathbf{e}^T(k)\frac{\partial(\mathbf{e}(k))}{\partial\langle\mathbf{A}(i)\rangle} \tag{6}$$

$$= \mathbf{x}^T(k-i)\mathbf{J}^T\mathbf{e}(k) + \mathbf{e}^T(k)\mathbf{J}\,\mathbf{x}(k-i), \tag{7}$$

where $\langle\rangle$ indicates that a fixed element from the matrix is chosen, and $\mathbf{J}$ is a matrix associated with the denominator that has the size of $\mathbf{A}(i)$ and is all zeros except for a 1 at the m^{th} row and n^{th} column. We can rewrite this equation in terms of a matrix $\mathbf{K}$ associated with the numerator by using the following rules:

1. Replace $\mathbf{J}$ by $\mathbf{K}$ and $\mathbf{J}^T$ by $\mathbf{K}^T$;

2. The pre- (or post-) multiplier of $\mathbf{J}$ becomes its transpose;

3. The pre- (or post-) multiplier of $\mathbf{J}^T$ becomes the post- (or pre-) multiplier of $\mathbf{K}^T$.

Thus,

$$\frac{\partial\langle \mathbf{e}^T(k)\mathbf{e}(k)\rangle}{\partial(\mathbf{A}(i))} = \mathbf{e}(k)\mathbf{K}^T\mathbf{x}^T(k-i) + \mathbf{e}(k)\mathbf{K}\mathbf{x}^T(k-i), \tag{8}$$

but in this case $\mathbf{K}$ is the size of $\mathbf{e}^T(k)\mathbf{e}(k)$ which is a scalar. Therefore, $\mathbf{K} = \mathbf{K}^T = 1$ and this equation can be rewritten as

$$\frac{\partial\langle \mathbf{e}^T(k)\mathbf{e}(k)\rangle}{\partial(\mathbf{A}(i))} = \frac{\partial(\mathbf{e}^T(k)\mathbf{e}(k))}{\partial(\mathbf{A}(i))} = 2\mathbf{e}(k)\mathbf{x}^T(k-i). \tag{9}$$

Rewriting equation (5) in terms of the derivative,

$$\frac{\partial J(N)}{\partial(\mathbf{A}(i))} = 2\sum_{k=p+1}^{N} \mathbf{e}(k)\mathbf{x}^T(k-i) = \mathbf{0}. \tag{10}$$

Since $E(\mathbf{e}(k)\mathbf{x}^T(k-i)) = \mathbf{0}$, for $i > 0$, substituting $\mathbf{e}(k)$ given by equation (2) into equation (10), and taking expected values, we get

$$\mathbf{R}(i) = -\mathbf{A}(1)\mathbf{R}(i-1) - \mathbf{A}(2)\mathbf{R}(i-2) + \cdots - \mathbf{A}(p)\mathbf{R}(i-p), \text{ for } i > 0 \tag{11}$$

where $\mathbf{R}(i) = E(\mathbf{x}(k)\mathbf{x}^T(k-i))$ is a correlation matrix of lag i and dimension $d \times d$ for the vector $\mathbf{x}(k)$. Substituting $i = 1, 2, \ldots, p$ in equation (11) we get the set of equations referred to as the Yule-Walker equations [19, 21]:

$$\begin{array}{ccccccccccc}
\mathbf{R}(1) & + & \mathbf{A}(1)\mathbf{R}(0) & + & \mathbf{A}(2)\mathbf{R}^T(1) & + & \cdots & + & \mathbf{A}(p)\mathbf{R}^T(p-1) & = & \mathbf{0} \\
\mathbf{R}(2) & + & \mathbf{A}(1)\mathbf{R}(1) & + & \mathbf{A}(2)\mathbf{R}(0) & + & \cdots & + & \mathbf{A}(p)\mathbf{R}^T(p-2) & = & \mathbf{0} \\
\vdots & & \vdots & & & & & & & & \\
\mathbf{R}(p) & + & \mathbf{A}(1)\mathbf{R}(p-1) & + & \mathbf{A}(2)\mathbf{R}(p-2) & + & \cdots & + & \mathbf{A}(p)\mathbf{R}(0) & = & \mathbf{0}
\end{array}$$

where we have taken into account that $\mathbf{R}^T(i) = \mathbf{R}(-i)$. These equations can be written as

$$-\left[\ \mathbf{R}(1)\mathbf{R}(2)\cdots\mathbf{R}(p)\ \right] = \left[\ \mathbf{A}(1)\mathbf{A}(2)\cdots\mathbf{A}(p)\ \right]\tilde{\mathbf{R}}, \tag{12}$$

where

$$\tilde{\mathbf{R}} = \begin{bmatrix} \mathbf{R}(0) & \mathbf{R}(1) & \cdots & \mathbf{R}(p-1) \\ \mathbf{R}^T(1) & \mathbf{R}(0) & \cdots & \mathbf{R}(p-2) \\ \vdots & \vdots & & \vdots \\ \mathbf{R}^T(p-1) & \mathbf{R}^T(p-2) & \cdots & \mathbf{R}(0) \end{bmatrix}, \tag{13}$$

which must be solved for the $\mathbf{A}(i),\ i = 1, \ldots, p$. We can estimate these parameter matrices by first calculating the cross-correlations using

$$\mathbf{R}(i) = \frac{1}{N} \sum_{k=i}^{N-1-i} [\mathbf{x}(k) - \mathbf{m}][\mathbf{x}(k-i) - \mathbf{m}]^T, \tag{14}$$

where $\mathbf{m}$ is the mean vector of the observed data

$$\mathbf{m} = \frac{1}{N} \sum_{k=1}^{N} \mathbf{x}(k). \tag{15}$$

Solution of equation (12) for the unknown AR parameter matrices requires the inversion of a $dp \times dp$ matrix. In the next section we show how to solve for the parameters recursively to avoid this inversion, which can be computationally expensive for large orders, p.

B. Generalization of the Levinson Recursions

Whittle [14] generalized Levinson's recursions [21] to the multivariate case and showed that equation (12) can be solved recursively so that only a total of $2p$ inversions of $d \times d$ matrices are required. In order to do this for the multivariate case, one must fit both a forward and a backward autoregression.

Let the $\mathbf{A}(i)$ now be represented by $\mathbf{A}_p(i)$ so that we may reference the parameters at a given order, p. Thus, the parameter matrices are solutions of the Yule-Walker equations

$$\sum_{i=0}^{p} \mathbf{A}_p(i)\mathbf{R}(j-i) = \mathbf{0},\ j = 1, \ldots, p, \tag{16}$$

where $\mathbf{A}_p(0) = \mathbf{I}$. Consider also the equations

$$\sum_{i=0}^{p} \overline{\mathbf{A}}_p(i)\mathbf{R}(i-j) = \mathbf{0},\ j = 1, \ldots, p, \tag{17}$$

where $\overline{\mathbf{A}}_p(0) = \mathbf{I}$. Define the following prediction error covariance matrices

$$\left.\begin{aligned}
\mathbf{V}_p &= \sum_{i=0}^{p} \mathbf{A}_p(i)\mathbf{R}(-i), \\
\boldsymbol{\Delta}_p &= \sum_{i=0}^{p} \mathbf{A}_p(i)\mathbf{R}(p-i+1), \\
\overline{\mathbf{V}}_p &= \sum_{i=0}^{p} \overline{\mathbf{A}}_p(i)\mathbf{R}(i), \\
\overline{\boldsymbol{\Delta}}_p &= \sum_{i=0}^{p} \overline{\mathbf{A}}_p(i)\mathbf{R}(-p+i-1).
\end{aligned}\right\} \tag{18}$$

The recursions are then given by

$$\left.\begin{aligned}
\mathbf{A}_{p+1}(i) &= \mathbf{A}_p(i) + \mathbf{A}_{p+1}(p+1)\overline{\mathbf{A}}_p(p-i+1), \\
\overline{\mathbf{A}}_{p+1}(i) &= \overline{\mathbf{A}}_p(i) + \overline{\mathbf{A}}_{p+1}(p+1)\mathbf{A}_p(p-i+1),\ i = 1, \ldots, p,
\end{aligned}\right\} \tag{19}$$

$$\left.\begin{aligned}
\mathbf{A}_{p+1}(p+1) &= -\boldsymbol{\Delta}_p\overline{\mathbf{V}}_p^{-1}, \\
\overline{\mathbf{A}}_{p+1}(p+1) &= -\overline{\boldsymbol{\Delta}}_p\mathbf{V}_p^{-1}.
\end{aligned}\right\} \tag{20}$$

These equations can be used to recursively solve for the AR coefficient matrices.

IV. Experiments

In this section, we describe the recording of the EEG signals, the mental tasks performed by the subjects, the neural network classifier, and the results of several methods for selecting an appropriate AR model order.

A. EEG Data Collection

The subjects were seated in an Industrial Acoustics Company sound controlled booth with dim lighting and noiseless fans for ventilation. An Electro-Cap elastic electrode cap was used to record from positions C3, C4, O1, O2, P3, and P4, shown in Figure 1 and defined by the 10-20 system of electrode placement [22]. Another channel was included to record eye blinks. The impedance of all electrodes was kept below 5 Kohms. Data was recorded at a sampling rate of 250 Hz with a Lab Master 12 bit A/D converter mounted in an IBM-AT computer. The electrodes were connected through a bank of Grass 7P511 amplifiers with analog bandpass filters. The passband was from 0.1 - 100 Hz. Figure 2 shows one second of data from a subject doing the two tasks described below.

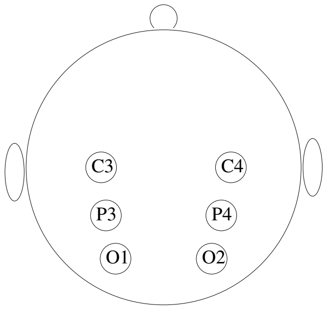

Figure 1: Placement of the electrodes according to the 10-20 system.

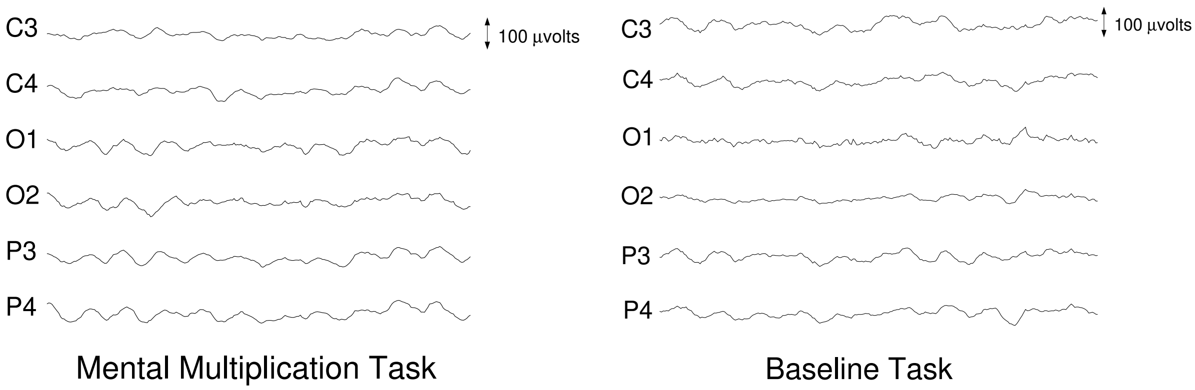

Figure 2: One second of data from Subject 3 performing each task.

Before each recording session the system was calibrated with a known voltage. Data was recorded for 10 seconds during each task and each task was repeated five times per session. Most subjects attended two such sessions recorded on separate weeks. For this paper we analyzed the data from four subjects, with the second subject completing only one session. All data was obtained previously by Keirn and Aunon [23, 3].

Data recorded during an eye blink was removed in all of our experiments. Eye blinks were detected by means of a separate channel of data recorded from two electrodes placed above and below the subject's left eye. An eye blink was said to have occurred if a change in magnitude greater than 100 μVolts occurred within a 10 milliseconds period.

B. Tasks

The mental tasks described here are two from Keirn and Aunon [3] and were chosen to invoke hemispheric brainwave asymmetry; it was shown in [24] that the peak of the power spectrum in the alpha frequency range (8-13 Hz) increased in the left hemisphere rather than the right during arithmetic tasks, whereas it increased in the right hemisphere rather than the left for visual tasks. The two tasks studied for this paper include:

- **Baseline Measurement**: The subjects were not asked to perform a specific mental task, but to relax as much as possible and think of nothing in particular. This task is considered the baseline task for alpha wave production and used as a control measure of the EEG.

- **Mental Multiplication**: The subjects were given nontrivial multiplication problems and were asked to solve them without vocalizing or making any other physical movements. An example is 49 times 78. The problems were not repeated and were designed so that an immediate answer was not attainable. Subjects were asked after each trial whether or not they found the answer, and no subject completed the problem before the end of the 10 second recording trial.

C. Neural Network Classifier

The classifier implemented for this work is a standard, feedforward, neural network with one hidden layer and one output layer, trained with the error backpropagation algorithm. We will briefly summarize the algorithm here; details can be found in [25] and any recent textbook, such as [26].

An input vector is applied to the input layer where all of the inputs are distributed to each unit in the first hidden layer. All of the units have weight vectors which are multiplied by these input vectors. Each unit sums these inputs and produces a value that is transformed by a nonlinear activation function, for which we used the common asymmetric sigmoid function. The output of the final layer is then computed by multiplying the output vector from the hidden layer by the weights into the final layer. More summations and activations at these units then gives the actual output of the network. We used a network with a variable number of hidden units and one output unit. One output unit is all that is needed because we are only classifying a two-task problem and the output sigmoid function can be trained to give a high or low output depending on the class of the input vector.

Training the network is accomplished by initializing all weights to small, random values and then performing a gradient-descent search in the network's weight space for a minimum of a squared error function of the network's output. The error is between the network's output and the target value for each input vector. We used target values of 0.1 for the baseline task and 0.9 for the mental multiplication task. The error backpropagation algorithm is derived by decomposing the gradient calculation into computations performed in each layer, starting with the final layer and passing results backwards through the network.

The amount by which the weights are adjusted on each step is parameterized by learning rate constants. We used one learning rate for the hidden layer and a different rate for the output layer. After trying a large number of different values, we found that a learning rate of 0.1 for the hidden layer and 0.01 for the output layer produced the best performance.

We performed the following cross-validation procedure for training the network as a way to control the over-fitting of training data. We randomly select 80% of the data set for training the network and 10% of the data for validation after each training epoch. The error of the network on the validation data is calculated after every pass, or epoch, through the training data. After a 1,000 epochs, the network state (it's weight values) at the epoch for which the validation error is smallest is chosen as the network that will most likely perform the best on novel data. This best network is then applied to the remaining 10% of the data, referred to as the test set. All representations were classified 30 times using different random selections of train, validation, and test sets and initial weight values.

D. Multivariate AR Model Algorithm Testing

Programs were written in MATLAB to estimate a multivariate AR model given data from six channels. Two approaches were implemented: the direct solution and the recursive solution. In the direct method the program was written to directly solve the system of matrix equations given by equation (12). In the recursive method the program was written to solve for the AR coefficients recursively using the generalization of the Levinson recursions given in equation (19). Implementing the direct method required about 85 lines of code, whereas the recursive method required about 125. Details of our implementation, including MATLAB code, appear in [27].

We tested our implementation of the AR model estimation algorithms by applying them to data generated from a known, second order, bivariate AR model. White noise was added to the model and 256 samples from the resulting process were collected. This was repeated 25 times. The AR algorithms provided good estimations of the AR coefficients for all 25 repetitions. To test our implementation for a case more similar to our EEG data, an order 6 AR process was generated for a 6 input, 6 output model and 25 sets of samples were generated. We found that 256 or more points of the process resulting in an acceptable estimate of the AR coefficients.

E. Multivariate AR Model Order Determination

When constructing an AR model it is important to determine the order of the model which best fits the data. The model order can be thought of as the number of past data samples needed to predict the present value of the data and, therefore, is dependent on the data sampling rate. Our sampling rate of 250 Hz is comparable to the 205 Hz used by Franaszczuk [9] to obtain coherence measures between electrodes. However, 250 Hz is much greater than the sampling rate used in other studies [10, 11, 12].

In the remainder of this section, we discuss three methods for estimating the best model order.

A first attempt at model order determination was made by minimizing the error in the predictor equation. An AR model was estimated from 256 points of data from all channels, and this model was tested on the next 256 points. To do this we used equation (3) to estimate the present sample vector based on past sample vectors. This estimate was compared with the true value and the squared error was computed. This squared error (SSE) was summed over all the estimates in the 256 point window. The process continued by estimating a new AR model from the next 256 points, and then testing the predictor on the next

256 points after that, and so on through all of the data in a trial.

This procedure was performed for two trials each of the baseline and multiplication tasks. Model orders from 1 to 10 were investigated. Results are shown in Figure 3. For all cases the minimum predictor error occurs at order 2. The minimum value of the SSE should indicate which order best fits the data.

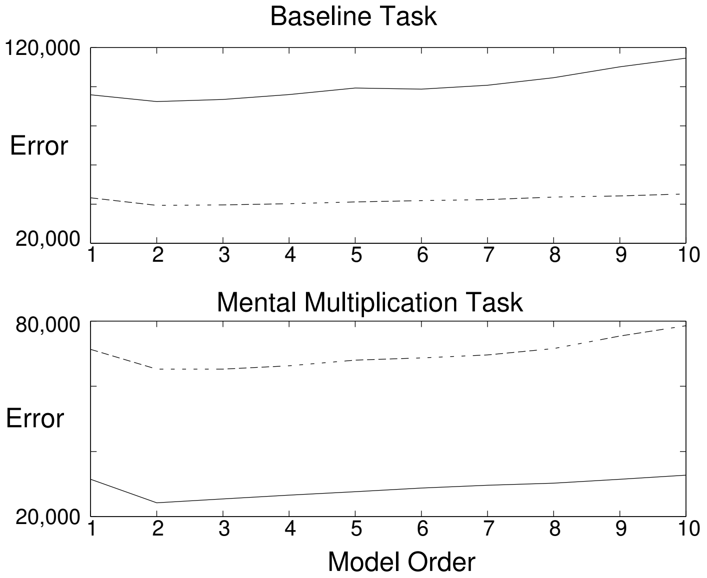

Figure 3: Sum-squared error of AR model versus order for two trials from each task.

Another attempt was made to determine the proper order by minimizing the Akaike information criterion (AIC) (see Chapter 9, Section 4 in [19], or Chapter 14, Section 7 in [21]). This criterion represents a trade-off between the estimated prediction error and the size of the model:

$$AIC(p) = N \ln \det(\hat{\Sigma}_p) + 2d^2 p, \tag{21}$$

where d is the number of inputs (or channels in this case) and p is the model order. $\hat{\Sigma}_p$ is the estimate of the prediction error covariance matrix assuming a p^{th} order predictor and is given by

$$\hat{\Sigma}_p = \sum_{i=0}^{p} \mathbf{A}(i)\mathbf{R}(-i). \tag{22}$$

Note that this summation is a by-product of the Levinson recursions and so is calculated recursively within the algorithm.

The AIC was calculated for two trials each of the baseline and multiplication tasks. An AIC value was computed for each 256 point window in each trial, and this was done for model orders from 1 to 10.

Results are shown in Figure 4. For both tasks there is one case each of orders 2 and 3 being the minimum AIC. Once again, the minimum value of the AIC should indicate which order best fits the data.

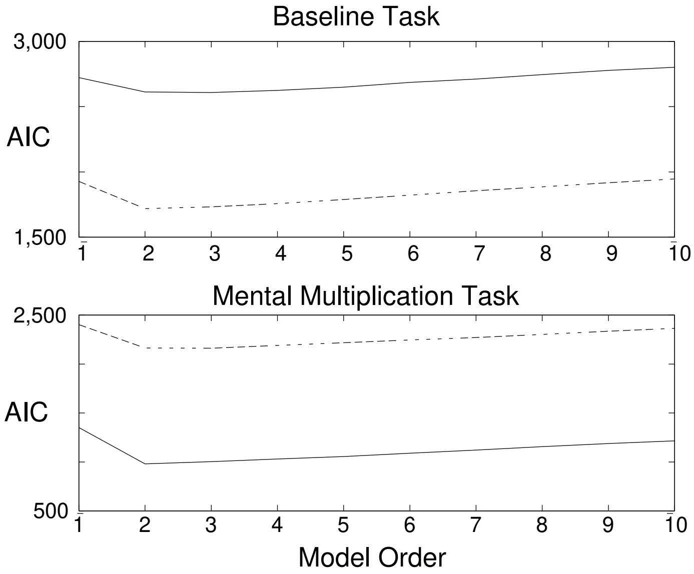

Figure 4: AIC criterion of AR model versus order for two trials from each task.

As a final method for estimating the proper model order, we calculated the matrix $\tilde{\mathbf{R}}$ given by equation (12) and then calculated its eigenvalues. The eigenvalues are plotted in Figure 5 for a 10^{th} order model for one trial of both the baseline and the multiplication task. One can see that the eigenvalues drop off very quickly and are not discernible after about 10 to 20 points. Since every six points represents one model order, these plots also indicate that a low model order of 1 to 3 is the best.

For the following classification experiments, we decided to implement a model of order 6, even though the maximum optimal value was found to be 3. This is because we wanted to incorporate more detail into the model that might be missing from a lower order, and also to enable us to compare this model to the scalar AR model which was found to have good performance at order 6 in Keirn and Aunon's study [3].

V. Results of Classification Experiments

In this section we present the results of classification experiments using data from four subjects performing the baseline and mental multiplication tasks. We compare the classification performance using four representations of the EEG signals: scalar AR coefficients, eigenvalues of matrix $\tilde{\mathbf{R}}$ given by

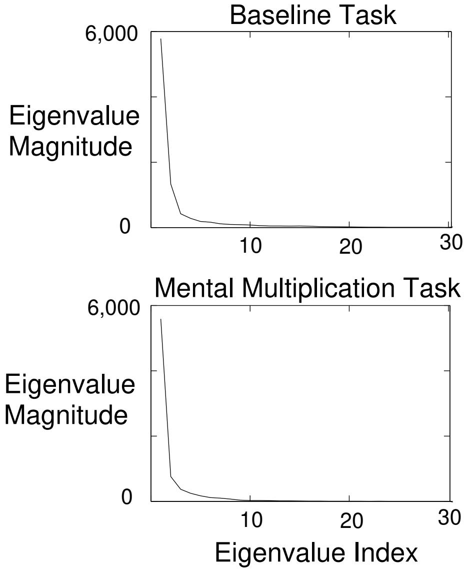

Figure 5: Magnitude of eigenvalues of matrix $\tilde{\mathbf{R}}$ versus order for two trials from each task.

equation (12), multivariate AR coefficients, and the Karhunen-Loève transform of the multivariate AR coefficients.

A. Scalar AR Coefficients

As a basis for comparison with Keirn and Aunon's results [3], we classified the two tasks based on the scalar AR coefficients estimated from our data. Coefficients were estimated from each channel using the Burg method(see Chapter 7, Section 6 in [21]) and concatenated together to form the feature vector.

To compare with Keirn and Aunon's [3, 23] results, we repeated one of their experiments using all possible data windows. Keirn and Aunon achieved an 87.5% classification accuracy using a quadratic Bayes classifier applied to single quarter-second windows extracted from each of 20 sessions for each task. The signals in each window were represented by the coefficients of order 6, AR models estimated for each of the 6 channels. The coefficients for each channel are concatenated into a 36-component feature vector. We used a much larger data set consisting of all quarter-second windows overlapped by one-eighth second, making a total of 79 windows for each 10 second session. Windows containing eye blinks were removed from the data set. This resulted in approximately 600 windows for each task. Each window was represented by AR models of order 6. We achieved a classification rate of 81.3% by using a neural network with a single hidden unit as a classifier. Thus, our results for this particular subject and task pair are slightly worse than those in the previous study. As expected, training and testing on more feature vectors lowers the classification rate.

We broadened this experiment by including all 1-second windows, overlapped by 1/2 second, and excluding all windows that overlap an eye blink. We again modeled the data in each window with an order 6 AR model. The classification results on test data are given in Table 1 for 1, 2, and 5 hidden units. The results are averages over 30 runs, each run with different combinations of train, test, and validate sets and different initial weight values. The total number of feature vectors for each subject was about 300, except subject 2 which had about half this amount. The validate and test sets each contained 10% of the total number of feature vectors, with the rest in the training set. Shown with the classification rates are the 90% confidence intervals. Although the error on the training data (not shown) decreases as the number of hidden units increases, the error on the test data, shown in the table, doesn't change significantly with the number of hidden units. This suggests that we can't do much better than a linear classifier with this representation. For all subjects except subject 3, the classification percentages on test data are above 90%. Subject 3 features classify at least 10% worse than the other subjects. Overall, an average of 90.6% classification is achieved across all subjects using five hidden units in the neural net.

	Number of Hidden Units		
	1	2	5
Subject 1	93.4 ± 1.5	93.6 ± 1.5	93.6 ± 1.5
Subject 2	96.7 ± 1.7	96.7 ± 1.6	96.1 ± 1.7
Subject 3	80.9 ± 2.5	80.7 ± 2.6	82.5 ± 2.6
Subject 4	91.0 ± 2.0	91.0 ± 2.0	90.0 ± 2.1
Average	90.5 ± 1.9	90.5 ± 1.9	90.6 ± 2.0

Table 1: Classification based on scalar AR coefficients (in %).

As a preliminary look at the ability of the neural network to generalize between trials, we classified the data for subject 3 once again, but this time randomizing the train, validate, and test sets separately, producing train, validate and test sets that contain data from distinct trials. The results are given in Table 2, averaged over 30 runs. The classification accuracies are lower for this case—63.7% compared to 82.5% for Subject 3 and 5 hidden units. This result might be improved in a number of ways. We are currently investigating the effect of more hidden units and more training data from additional sessions.

	Number of Hidden Units		
	1	2	5
Subject 3	54.8 ± 6.1	60.2 ± 7.0	63.7 ± 7.2

Table 2: Classification based on scalar AR coefficients, subject 3, randomizing by trial

B. Eigenvalues of Matrix $\tilde{\mathbf{R}}$

As a preliminary *multivariate* classification experiment, we classified the two tasks based on the eigenvalues of the matrix $\tilde{\mathbf{R}}$ given in equation (12). The eigenvalues were generated for a multivariate AR model of order 6, resulting in feature vectors of size 36 once again. The results are given in Table 3.

The network is able to classify very well with these features for three subjects. The classification percentages are all above 95%, except for subject 3, whose results were worse again, this time over 20% lower than the others. Averaging over all subjects the classification accuracy is approximately 90%.

	Number of Hidden Units		
	1	2	5
Subject 1	95.9 ± 1.1	95.9 ± 1.2	96.1 ± 1.1
Subject 2	97.8 ± 1.3	97.8 ± 1.3	98.1 ± 1.1
Subject 3	69.8 ± 2.7	70.2 ± 2.8	70.1 ± 2.4
Subject 4	98.3 ± 0.7	97.3 ± 0.8	97.2 ± 0.7
Average	90.5 ± 1.5	90.3 ± 1.5	90.4 ± 1.3

Table 3: Classification based on eigenvalues of matrix $\tilde{\mathbf{R}}$ for order 6 model.

C. Multivariate AR Coefficients

The next classification experiment involved training the neural net on the multivariate AR coefficients. For a 6^{th} order model and 6 channels, this means a feature vector of size 216. As in our previous experiments, these coefficients were generated from 256 point windows of data which overlapped each other by 1/2 second. No windows which overlapped an eye blink were included. Results are given in Table 4. The results are more consistent here with subject 3 having the worst classification percentage at 87.0% and subject 4 having the best at 96.2% using five hidden units. Averaging over all subjects we find the best classification rate of 91.4% with five hidden units, suggesting that there is some non-linearity in the decision function. However, the differences are still very small and not statistically significant.

To get an idea of what information the neural network is capturing, we have plotted the magnitudes of the weights in the hidden unit of a one hidden unit network as matrices corresponding to the six AR

	Number of Hidden Units		
	1	2	5
Subject 1	87.4 ± 1.8	87.7 ± 1.9	88.3 ± 1.9
Subject 2	93.9 ± 2.7	94.2 ± 2.4	94.2 ± 2.3
Subject 3	85.4 ± 1.3	86.0 ± 1.4	87.0 ± 1.7
Subject 4	95.8 ± 1.5	96.3 ± 1.3	96.2 ± 1.3
Average	90.6 ± 1.8	91.1 ± 1.8	91.4 ± 1.8

Table 4: Classification based on multivariate AR coefficients.

coefficient matrices. The plots are given in Figure 6, ordered by lag. In the first four plots we see that weight magnitudes are relatively larger along the diagonal. These weights correspond to the scalar AR coefficients from each channel. However, in each plot there are also weights off the diagonal that have large magnitudes, so the neural net is using information from the multivariate model. The magnitude of the weights corresponding to the last two AR coefficient matrices decrease slightly compared with the other four. This agrees with our earlier finding that models of order 2 and 3 fit the data quite well.

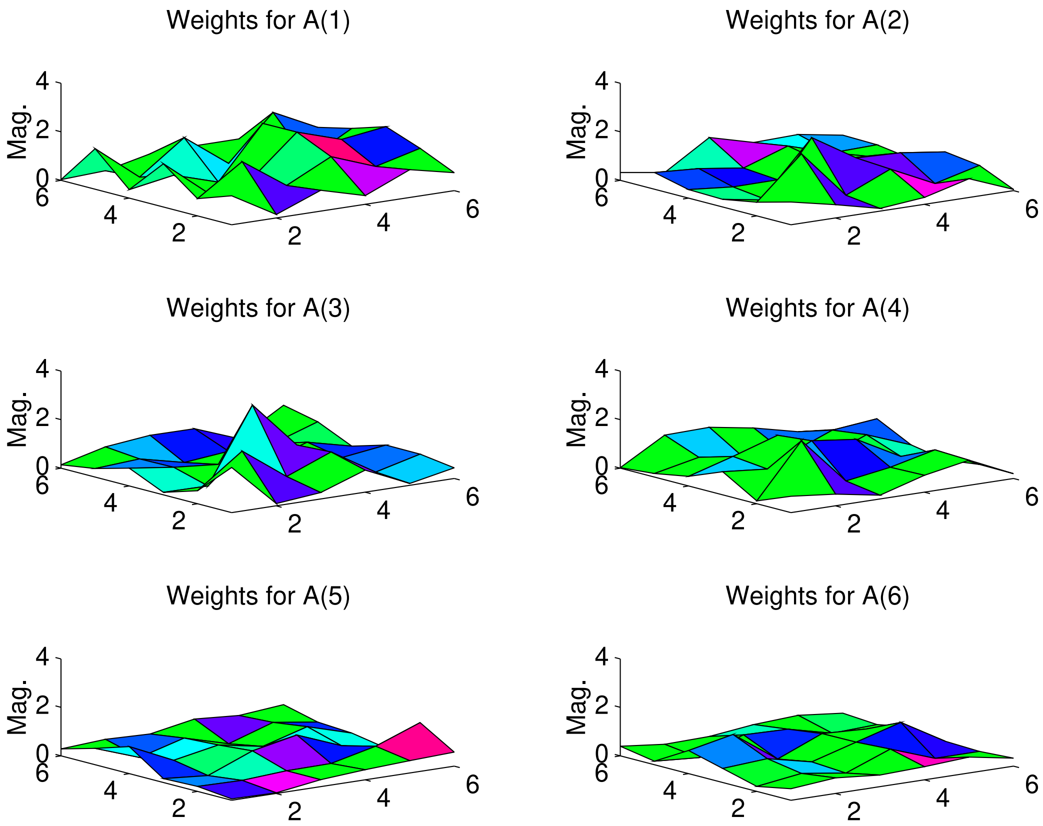

Figure 6: Magnitudes of the weights of the input to the neural net corresponding to the multivariate AR coefficient matrices, $\mathbf{A}(1) - \mathbf{A}(6)$

D. Karhunen-Loève Transform of Multivariate AR Coefficients

In order to reduce the number of features needed to train the network, we calculated the Karhunen-Loève (K-L) transform of the multivariate AR feature vectors. It was thought these high-dimensional feature vectors might be larger than what is actually needed. The K-L transform can reduce the feature space by projecting the original feature vectors onto a small number of eigenvectors.

To perform the K-L decomposition, the covariance matrix of the set of 216-dimensional original feature vectors (with means subtracted) is calculated. The eigenvectors and eigenvalues of this covariance matrix are then calculated. The eigenvectors corresponding to the largest eigenvalues are chosen to project onto the data. The choice of the number of eigenvectors to use can be estimated with the K-L estimate, given by the index i for which $\lambda_i/\lambda_{max} \leq 0.01$, where the $\lambda_i,\ i = 1, 2, ...$ are the eigenvalues in decreasing order.

Figure 7 shows a plot of the first 100 eigenvalues estimated from the covariance matrix of our multivariate AR feature vectors for subject 3. The eigenvalues are similar for all subjects. The number of eigenvectors was chosen to be 50 according to the above criteria. The K-L transform was generated by projecting each 216-dimensional pattern onto these 50 eigenvectors.

The 50-dimensional feature vectors were trained and tested the same way as in the previous experiments. The results are given in Table 5. There is a large spread in classification percentages among subjects. Subject 3 performs the worst with 72.6% accuracy, and subject 4 the best with 96.4% accuracy with one hidden unit. Averaging over all subjects we find a classification rate of 86.1% with five hidden units, which is slightly better than with fewer hidden units.

	Number of Hidden Units		
	1	2	5
Subject 1	80.8 ± 1.9	80.9 ± 1.8	81.1 ± 1.9
Subject 2	92.5 ± 2.5	92.5 ± 2.4	92.8 ± 2.4
Subject 3	72.6 ± 2.3	73.2 ± 2.2	73.9 ± 2.2
Subject 4	96.4 ± 1.0	96.5 ± 0.9	96.5 ± 0.9
Average	85.6 ± 1.9	85.8 ± 1.8	86.1 ± 1.9

Table 5: Classification based on K-L transform of multivariate AR coefficients.

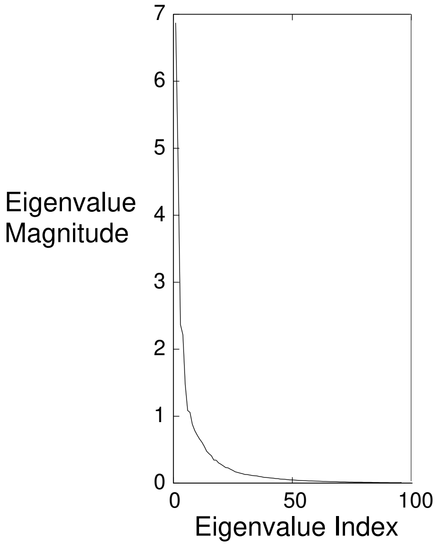

Figure 7: Eigenvalues of covariance matrix of multivariate AR coefficients.

E. Comparison

Table 6 repeats the average classification accuracy for all representations and all subjects using five hidden units. These values are also plotted in Figure 8. Looking at the subjects individually, subject 3 has the worst results over all representations. Subjects 2 and 4 perform consistently over 90% for all representations. The eigenvalues of matrix $\tilde{\mathbf{R}}$ give the best results for subjects 1, 2, and 4, whereas the multivariate AR coefficients are superior for subject 3. The K-L transform of the multivariate AR coefficients classify the worst for subjects 1 and 2, and the eigenvalues of matrix $\tilde{\mathbf{R}}$ and the scalar AR coefficients classify the worst for subjects 3 and 4, respectively.

Representation	Subject				Average
	1	2	3	4	
Scalar AR	93.6	96.1	82.5	90.0	90.6 $\pm$ 2.0
MV AR	88.3	94.2	87.0	96.2	91.4 $\pm$ 1.8
$\tilde{\mathbf{R}}$ Eig.	96.1	98.1	70.1	97.2	90.4 $\pm$ 1.3
K-L Trans.	81.1	92.8	73.9	96.5	86.1 $\pm$ 1.9

Table 6: Results for each subject and each representation using five hidden units.

One can see when looking at the averages across subjects that the multivariate AR coefficients give the best classification accuracies at 91.4%, but not by much. The scalar AR coefficients and the eigenvalues of matrix $\tilde{\mathbf{R}}$ perform slightly worse at 90.6% and 90.4%, respectively. The K-L transform of the multivariate AR coefficients perform about 4% worse than the previous two.

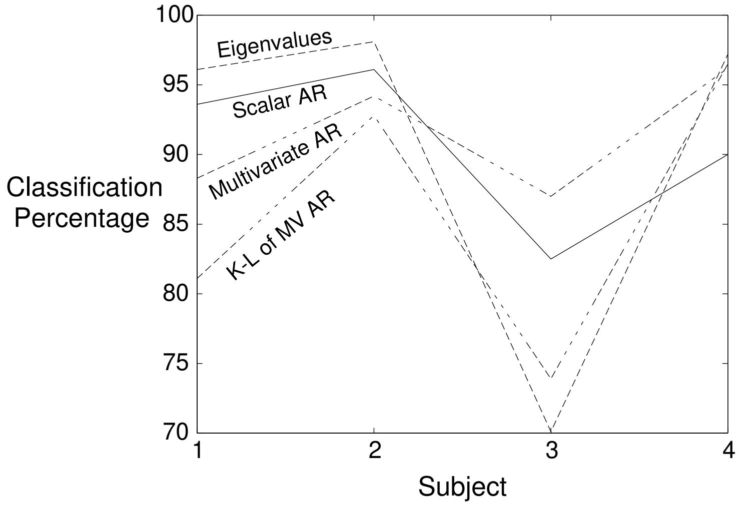

Figure 8: Results across all subjects and all representations with five hidden units.

These results indicate that the difference in performance among the various feature extractors is not significant at the 90% confidence level, except that the K-L Transform of the multivariate AR coefficients performs significantly worse than the other representations. Our results show that the multivariate AR coefficients are the most consistent feature vector. However, if in the future many subjects are to be tested and computation time is an issue, the scalar AR coefficients appear to be the best choice. If a single subject is to be studied, all of the representations should be implemented to find the one which gives the best results.

VI. Conclusions

The purpose of this project was to investigate a multivariate AR model for the extraction of classifiable features from spontaneous EEG during two mental tasks. As stated earlier, this work is part of a larger project, initiated by Aunon and Keirn [3, 23], to study the feasibility of using various mental tasks as an alphabet for controlling a device such as a wheelchair for those who are physically handicapped. To form a bridge between this study and the previous one, we repeated an experiment using many more feature vectors in the training set than in the previous work. We found that the classification percentages between the two tasks go down slightly when using this larger data set. These results are based on data from one session only, all other experiments were performed with data from all recorded sessions.

Next we classified one scalar AR feature set and three feature sets based on a multivariate AR model:

the AR coefficient matrices themselves, the eigenvalues of an intermediate matrix of cross-correlation matrices, and the K-L Transform of the multivariate AR coefficient matrices. The multivariate AR model had been rarely used in EEG research for this purpose.

Tests on the data from four subjects showed that no representation is superior for all subjects. It appears that when studying one subject alone, many representations should be investigated to find the one which best fits the subject. The multivariate AR coefficients classify most consistently over all subjects, and also have the best performance of 91.4% when results are averaged over all subjects. However, the scalar AR coefficients and the eigenvalues of the cross-correlation matrices perform almost as well at 90.6% and 90.4%, respectively. So it appears that if in the future computation time needs to be conserved, the scalar AR coefficients would be the most efficient representation. It also appears that due to the extra calculation time involved in calculating the K-L transform of the multivariate AR coefficients and its lower classification percentages, this representation may not be worth considering for a real-time system.

Overall, it was found that for all but one subject there exists one representation that could be used to discriminate between the two tasks with greater than 96% accuracy. This type of result might be marginally acceptable for a real-time system based on two commands for some subjects. In order to improve the above results, it might be desirable to average the classification of feature vectors over time, which is part of the current research in this project. Other future work could involve down-sampling the data prior to estimating the AR models in order to capture longer time dependencies. Increasing the number of channels to the 19 given by the 10/20 system might also improve the results, because areas other than the central, parietal, and occipital regions of the brain are probably involved in the execution of these tasks.

An important limitation of the present study is the little attention given to generalization between trials. Our preliminary results (see Table 2) on the data from one subject suggest that a much larger data set consisting of many sessions might be needed to generalize well from one trial to another. It also appears that a greater number of hidden units might produce better results.

References

[1] L. A. Farwell and E. Donchin. Talking off the top of your head: Toward a mental prosthesis utilizing event-related brain potentials. *Electroencephalography and Clinical Neurophysiology*, 70:510–

523, 1988.

[2] D. J. McFarland, G. W. Neat, R. F. Read, and J. R. Wolpaw. An EEG-based method for graded cursor control. *Psychobiology*, 21(1):77–81, 1993.

[3] Zachary A. Keirn and Jorge I. Aunon. A new mode of communication between man and his surroundings. *IEEE Transactions on Biomedical Engineering*, 37(12):1209–1214, December 1990.

[4] A. C. Tsoi, D. S. C. So, and A. Sergejew. Classification of electroencephalogram using artificial neural networks. In Jack D. Cowan, Gerald Tesauro, and Joshua Alspector, editors, *Advances in Neural Information Processing Systems 6*, pages 1151–1158. Morgan Kaufmann, 1994.

[5] S. Roberts and L. Tarassenko. EEG analysis using self-organisation. In *Proc. Second Int. Conf. on Artificial Neural Networks*, pages 210–213, London, UK, 1991. IEE.

[6] Z. Ramadan, K. Ropella, J. Myklebust, M. Goldstein, X. Feng, and J. Flynn. A neural network to discriminate between dyslexic subtypes. In *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, volume 13, pages 1405–1406. 1991.

[7] G. M. Papadourakis, S. M. Micheloyannis, G. Bebis, and M. Giachnakis. EEG signal classification using neural networks. In *Engineering Systems with Intelligence: Concepts, Tools, and Applications*, pages 221–228. Kluwer Academic Publishers, The Netherlands, 1991.

[8] D. M. Tumey, P. E. Morton, D. F. Ingle, C. W. Downey, and J. H. Schnurer. Neural network classification of EEG using chaotic preprocessing and phase space reconstruction. In *Proceedings of the 1991 IEEE Seventeenth Annual Northeast Bioengineering Conference*, pages 51–52. 1991.

[9] P. J. Franaszczuk, K. J. Blinowska, and M. Kowalczyk. The application of parametric multichannel spectral estimates in the study of electrical brain activity. *Biological Cybernetics*, 51:239–247, 1985.

[10] P. Rappelsberger and H. Petsche. Spectral analysis of the EEG by means of autoregression. In G. Dolce and H. Kunkel, editors, *CEAN-Computerized EEG Analysis*, pages 27–40. Gustav Fischer, 1975.

[11] J. C. Jimenez, R. Biscay, and O. Montoto. Modeling the electroencephalogram by means of spatial spline smoothing and temporal autoregression. *Biological Cybernetics*, 72:249–259, 1995.

[12] W. Gersch and J. Yonemoto. Parametric time series models for multivariate eeg analysis. *Computers and Biomedical Research*, 10:113–125, 1977.

[13] W. Gersch and J. Yonemoto. Automatic classification of multivariate EEGs using an amount of information measure and the eigenvalues of parametric time series model features. *Computers and Biomedical Research*, 10:297–318, 1977.

[14] P. Whittle. On the fitting of multivariate autoregressions, and the approximate canonical factorization of a spectral density matrix. *Biometrika*, 50:129–134, 1963.

[15] A. Swami, G. Giannakis, and S. Shamsunder. Multichannel arma processes. *IEEE Transactions on Signal Processing*, 42(3):898–913, April 1994.

[16] E. J. Hannan. The identification and parameterization of ARMAX and state space forms. *Econometrica*, 44(4):713–723, 1976.

[17] O. N. Strand. Multichannel complex maximum entropy (autoregressive) spectral analysis. *IEEE Transactions on Automatic Control*, AC-22:634–640, 1977.

[18] E. A. Robinson. *Multichannel Time Series Analysis with Digital Computer Programs*. Holden-Day, San Fransciso, CA, 1967.

[19] M. B. Prestley. *Spectral Analysis and Time Series*. Academic Press, 1981.

[20] P. S. Dwyer and M. S. MacPhail. Symbolic matrix derivatives. In *Annals of Mathematical Statistics*, volume 19, pages 517–534. 1948.

[21] S. M. Kay. *Modern Spectral Estimation: Theory and Application*. Prentice-Hall, 1988.

[22] H. Jasper. The ten twenty electrode system of the international federation. *Electroencephalographic Clinical Neurophysiology*, 10:371–375, 1958.

[23] Zachary A. Keirn. Alternative modes of communication between man and machine. Master's thesis, Purdue University, 1988.

[24] M. Osaka. Peak alpha frequency of EEG during a mental task: Task difficulty and hemispheric differences. *Psychophysiology*, 21:101–105, 1984.

[25] D. E. Rumelhart, G. E. Hinton, and R. W. Williams. Learning internal representations by error propagation. In D. E. Rumelhart, J. L. McClelland, and The PDP Research Group, editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, volume 1. Bradford, Cambridge, MA, 1986.

[26] Simon Haykin. *Neural Networks: A Comprehensive Foundation*. Macmillan College Publishing Company, Inc., New York, New York, 1994.

[27] Charles W. Anderson, Erik Stolz, Sanyogita Shamsunder, and Charles Martin. Implementation of multivariate AR algorithms in MATLAB and applied to EEG pattern classification. *Scientific Programming*, submitted.