23/8/95.
16.55.

ANALYSIS OF THE CNV WAVEFORM IN THE TIME AND FREQUENCY DOMAINS

by

MICHAEL COELHO

A thesis submitted to the Council for National Academic Awards in partial fulfillment of the requirements for the degree of Master of Philosophy

Sponsoring Establishment : Department of Electrical and Electronic Engineering, Sheffield City Polytechnic

Collaborating Establishments : Department of Neurological Sciences, Freedom Fields Hospital, Plymouth

Depts of Neurophysiology & Psychiatry, Royal Hallamshire Hospital, Sheffield

April 1988

ANALYSIS OF THE CNV WAVEFORM IN THE TIME AND FREQUENCY DOMAINS

M. COELHO

ABSTRACT

The Contingent Negative Variation (CNV) response of the Electroencephalogram (EEG) is often obscured by the background EEG and/or ocular artefacts (OAs). These necessitate the application of a variety of signal processing techniques to reduce the influence of such phenomena on the measured signal. The methods discussed in this thesis are: ocular artefact removal (OAR) in the time domain and the use of data tapering and Fourier Transforms to give frequency domain amplitude and phase spectra.

Two OAR methods were compared: a non-recursive method using ordinary least squares parameter estimation developed by Nichols and a recursive least squares technique developed by Ifeachor. Recursive OAR was observed to distort the CNV response. This was discovered to be due to omission of the response from the algorithm being used, an omission ocurring in both techniques. To correct this the inclusion of a model of the response in the algorithm was proposed. A number of data sets were devised in order to investigate this effect and the successfulness of including the response model. It was shown that response modelling gave more efficient OAR and reduced response distortion. Similar investigations were performed on recorded response data which showed that modelling was essential for recursive OAR but that non-recursive OAR was relatively insensitive to the inclusion or omission of response modelling.

The use of data tapering was included in order to help improve the spectral analysis of short epochs of EEG data.

A comparison of statistical properties of the harmonics of the amplitude and phase spectra of the CNVs of normal, patient and at risk subjects was made using Predictive Statistical Diagnosis (PrSD) and Discriminant Analysis (DA). The investigation compared results between PrSD and DA and between 1 and 4 second inter-stimulus interval CNVs and suggested that the 1 second data gave better discrimination.

ACKNOWLEDGEMENTS

Any such work owes much to others for their help and support.

First I must thank my supervisors Barrie Jervis and Elaine Allen for their advice and interest in this project. I must also thank Barrie Jervis for his critical comments on this thesis. Thanks also go to Geoff Morgan for discussions on the use of the least squares method and also for allowing me to use his predictive statistical diagnosis programmes. Thanks are also due to Martin Nichols and Emmanuel Ifeachor for their helpful comments on running their software on a different machine. In this respect I would also like to thank the staff of the Computer Service sections of Sheffield City and Plymouth Polytechnics and the University of Sheffield. I would also like to express my gratitude to the staff of Sheffield City Polytechnic Library for their help in procuring all the material I studied from outside Sheffield.

For financial assistance I must thank Barry Plumb for arranging a Research Assistantship and the IEE for a postgraduate scholarship.

Finally I want to thank Detelina for all her encouragement and help.

COURSES AND SEMINARS ATTENDED

In accordance with regulations 3.8 and 3.9 I have attended and participated in the following:

Final year honours degree course in Signal Processing.
Jan 1985 and Oct-Dec 1985

EEG Society, Scientific Meeting, Birmingham.
29th. April 1987

CONTENTS

1 INTRODUCTION

The following points are discussed in this chapter : the origins of this thesis; the data processing methods to be applied; and the goals and aims of the thesis.

Section 1.1 is a desription of the thesis origins and a brief overview of the relevant areas in the field of EEG work, an outline of the Contingent Negative Variation (CNV) and Auditory Evoked Potential (AEP) responses of the human EEG, and Ocular Artefacts (OAs). The time and frequency domain methods to be applied are given in Section 1.2. These comprise Ocular Artefact Removal (OAR) in the time domain, signal processing methods of windowing (time domain) of data prior to transformation to the frequency domain phase and energy spectra. The statistical methods applied are also described in this section. These involved the generation of a number of test statistics, data reduction methods to determine a set of variables which gave best discrimination between a number of subject groups, and the use of Discriminant Analysis (DA) and Predictive Statistical Diagnosis (PrSD) to attempt to classify subjects which may fall into one or other of the known groups. Previous work which formed the starting point for this thesis is also described. Section 1.3 is a description of the experimental data to be used: 1 and 4 second CNVs. Section 1.4 gives the aims of this work and Section 1.5 is a brief outline of the thesis layout.

1.1 THE ELECTROENCEPHALOGRAM AND THE CONTINGENT NEGATIVE VARIATION AND AUDITORY EVOKED POTENTIAL RESPONSES

1.1.1 ORIGINS OF THE ELECTROENCEPHALOGRAM

The first reported observation of the electroencephalogram was by CATON (1875) who discovered that incessant current oscillations could be displayed by a galvanometer connected to two electrodes placed on the exposed cerebral cortex of an animal. BERGER (1929) was the first to report on the EEG of man, showing that it could be recorded through the skull. DAWSON (1947) reported a means of making scalp recordings of stimulated cerebral action potentials in man by displaying individual EEG traces on a cathode-ray oscilloscope and superimposing them on a single photograph. This was necessary due to the small magnitude of such evoked potentials when compared to the background EEG.

1.1.2 DESCRIPTION OF THE ELECTROENCEPHALOGRAM

An introduction to the physical structure and electrical activity of the brain is given in COOPER et al (1980). The ongoing background rhythmical EEG can be described in terms of four frequency ranges or bands (COOPER et al, op. cit.): delta (<4Hz), theta (4 to < 8Hz), alpha (8 to 13 Hz) and beta (>13 Hz). GEVINS (1984) considers any activity at > 30 Hz as a separate, gamma, band. In addition potentials due to endogenous or exogenous events can be observed in the EEG. In this thesis such phenomena will be referred to collectively as responses. This is to avoid any conflict of terminology arising out of the use of one of the commonly applied descriptions since a variety of names have been given to such EEG responses. Common references are to evoked potentials (EPs), event-related potentials (ERPs) and stimulus-related potentials (SRPs) even though the same specific response may be classed under any one of these three headings by different authors.

1.1.3 THE CONTINGENT NEGATIVE VARIATION

One of the numerous responses of the EEG is the Contingent Negative Variation (CNV) first described by WALTER et al (1964). The CNV is elicited by a constant foreperiod reaction time task. The standard experimental paradigm is to give a warning stimulus (S_1) which is followed after a time (known as the Inter-Stimulus Interval (ISI)) by a second (imperative) stimulus (S_2) at which some action is required on the subject's behalf (this is often a motor-response (MR) caused by pressing a button to terminate S_2). S_1 and S_2 are often either audio or visual stimuli. The CNV proper is the negative shift of the EEG baseline which occurs between the two stimuli.

The CNV is considered to comprise several components (e.g., see TECCE and CATTANACH, 1982 and ROHRBAUGH et al, 1976 and 1986). LOVELESS and SANFORD (1974) conclude that the CNV comprises two components which are an orientating response subsequent to S_1 (the "O" wave) and an expectancy response in anticipation of S_2 (the "E" wave). Using a 4 second ISI they found the "O" wave to peak no later than 850 msec after S_1 and that the "E" wave commenced about halfway through the foreperiod.

1.1.4 THE AUDITORY EVOKED POTENTIAL

Although this thesis is concerned with the CNV the Auditory Evoked Potential (AEP) is introduced here because of its presence in the experimental paradigm (described in the previous section) when S_1 and S_2 are auditory stimuli. It is also considered to be a multi-component response (see e.g., COOPER et al, 1980 or SHAGASS, 1972).

1.1.5 OCULAR ARTEFACTS

Ocular artefact (OA) is the collective reference to a number of eye-related artefacts: eye-movements (EMs) and eyelid and blink artefacts. OAs often obscure responses, especially the CNV. It has been noted that CNV changes at the vertex can be reproduced by voluntary downward eye movement (STRAUMANIS et al, 1969). HILLYARD and GALAMBOS (1970) concluded that on average 23% of the CNV was composed of Eye Artefact Potential (EAP). A discussion of the types and causes of OA is given in JERVIS et al (1988).

1.2 TIME AND FREQUENCY DOMAIN METHODS OF PROCESSING AND ANALYSING THE CONTINGENT NEGATIVE VARIATION

The use of both time and frequency domain methods was considered necessary since they provide complimentary views of a waveform. The time domain approach allows ease of visual interpretation of a waveform, indeed it was through study of the time domain representation of waveforms that the problems inherent in the Ocular Artefact Removal (OAR) method came to light (see Chapter 2 for details). The frequency domain is important because it is here that the statistical tests (Chapter 4) are applied. These are needed to examine important freqency components of the response and to quantify the properties of the harmonics.

1.2.1 OCULAR ARTEFACT REMOVAL

Section 1.1.5 gave the reasons why the presence of OAs in the EEG are undesirable. A variety of methods of OAR exist (details are given in JERVIS et al, 1988) which can be loosely grouped into three categories: rejection, fixation and subtraction.

OAR rejection methods discard records in which OAs are present. This has the drawback of being wasteful of data.

In eye fixation methods subjects are asked to fix their eyes on a target and refrain from blinking. This suffers from two drawbacks: not all subjects can co-operate and (especially for the CNV) the requested behaviour effectively assigns a 'secondary task' (GRATTON et al, 1983) or acts as a 'distracting stimulus' (WEERTS and LANG, 1973). Distraction processes are considered amongst the most disruptive of CNV development (TECCE and CATTANACH, 1982).

Methods of OAR using Electro-oculogram (EOG) subtraction have been used in both time and frequency domains. They have been applied on-line and off-line and have used both analogue and digital techniques (see JERVIS et al, 1988, for a review of various methods). The principle of subtraction methods is that the measured EEG is the linear sum of the true (background) EEG and OAs and hence if OAs in the EEG can be estimated the true EEG can also be estimated. In this thesis interest centres on time domain OAR using the least squares method to estimate how much OA in the measured EEG (derived from measurement of the EOG) should be subtracted to estimate the true EEG. This is achieved by computing a number of parameters known as transmission coefficients. Two such methods are considered in Chapter 2: the off-line technique of non-recursive OAR (NR-OAR) of QUILTER et al (1977) as extended by NICHOLS (1982), and the on-line recursive OAR (R-OAR) method of IFEACHOR (1984). The terms 'non-recursive' and 'recursive' refer to the manner of applying the least squares method. NR-OAR uses ordinary least squares (OLS) techniques, while R-OAR uses recursive least squares (RLS). OLS provides constant values for the transmission coefficients which depend upon the data of the whole record, while RLS provides values (based on weighted data) which are being updated point-by-point and hence use only some of the data (i.e., the latter part since the weighting discards the early data) and thus provide parameter estimates which can vary with time.

1.2.2 SPECTRAL ANALYSIS

The signal processing methods of NICHOLS (1982) applied a Tukey window (Chapter 3) in the time domain to data prior to transformation to frequency domain amplitude and phase spectra. The harmonics of these spectra were then subject to a number of statistical tests (see next section). This windowing also involved transformation of the windowed data due to the introduction of a spurious d.c. level even though the mean level of the data had already been removed.

1.2.3 STATISTICAL METHODS

Statistical tests were applied by NICHOLS (1982) to allow a quantitative comparison of the spectral properties of the CNVs of two subject groups (normal and Huntington's Chorea (HC)). An attempt was also made to classify subjects at risk (AR) of developing HC. This work was extended to a logic algorithm (LA) classification procedure (JERVIS et al, 1984). These tests are concerned with detecting additivity and phase ordering effects on responses in the EEG.

1.2.4 THE APPLICATION OF THE CONTINGENT NEGATIVE VARIATION TO HUNTINGTON'S CHOREA FOR DIAGNOSTIC PURPOSES

One possible application of the CNV in attempting to diagnose subjects at risk of developing HC was introduced in the preceding section. HC is commonly first detected between the ages of 30 and 50 (mean age 44 years), is invariably fatal and of duration of approximately 15 years from onset. NICHOLS (1982) and JERVIS et al (1984) considered that since HC affects the same regions of the brain responsible for generating the CNV, study of the latter could lead to an ability to discriminate between normal and HC subjects with application to pre-symptomatic diagnosis of HC in at risk subjects (the offsprings of HC sufferers).

1.3 EXPERIMENTAL DATA RECORDINGS

The data used in this thesis had been recorded in earlier work (NICHOLS, 1982). Both CNV and AEP responses are referred to. The CNV data had been recorded using plastic cup Ag/AgCl dc electrodes using vertex to linked earlobe electrodes. The EOG recordings used nasal, outer canthi, infra- and supra orbital electrodes. The acquisition system -3dB passband was 0.016 - 30 Hz. The warning stimulus S_1 was a 70 dB SPL click and the imperative stimulus S_2 was a 1 kHz tone of 90 dB SPL (A-weighting). At S_2 the subject was asked to press a handheld pushbutton to terminate it. Thirty-two such trials were obtained per subject. Two experimental conditions were used: the first had a 1 second ISI, the second had a 4 second ISI. The EEG and EOGs were recorded by sampling at 125Hz.

1.4 GOALS AND AIMS OF THESIS

The aims of the work can be seen as developing the methods described in Section 1.2. The first goal was the investigation of means of improving the time domain processing of the data. Initially a comparison of the NR-OAR and R-OAR methods was to have been made but in doing so it was observed that R-OAR produced response distortion. This was due to the omission, in the algorithm being used, of a model for the response. Thus it was decided to investigate, on both real and test data, the effects of the omission and the effectiveness (or otherwise) of including a model of the response. A second goal was the development of improved means of spectral analysis of short epochs of EEG data by improved data windowing.

In the frequency domain the main investigation concerned the application of PrSD (AITCHISON et. al., 1977) and DA alternatives to the LA classification procedure. In doing so it was hoped that improved classification would arise when applied to discriminating between various subject groups (as described in Section 1.2.4). The results from PrSD, DA and LA (1 second data) and PrSD and DA (4 second data) methods were to be compared within and between the two data sets.

In light of the findings of LOVELESS and SANFORD (1974) the 4 second CNV was to be investigated as two epochs of equal length, with a small amount of overlap, covering the first and second halves of the interval from the end of S_1 to the start of S_2.

1.5 OVERVIEW OF THE THESIS

The layout of this thesis follows the sequence of processing steps and analysis which were applied to the data. Chapter 2 deals with OAR. In it the following topics are discussed: the basic principles used in OAR, the omission of the response from the basic model and its effects and a proposed solution to introducing a response model and the results of this procedure. This investigation was applied to both real and test data. Chapter 3 is a discussion of various signal processing issues and includes a comparative study of two types of data window applied to both real and test data. Chapter 4 makes brief reference to the statistical tests of NICHOLS (1982), descibes DA and PrSD methods, shows how they were implemented and gives guidance on how to interpret the results. Chapter 5 gives the results when the processing steps were all applied and discusses and draws a number of conclusions based on the CNV data.

REFERENCES

AITCHISON J., HABBEMA J.D.F. and KAY J.W. (1977)
"A Critical Comparison of Two Methods of Statistical Discrimination"
Appl. Statist. 1977,26(1):15-25

BERGER H. (1929)
"Uber des Elektrenkephalogram des Menschen"
Arch. Psychiat. 1929,87:527-570

CATON R. (1875)
"The Electric Currents of the Brain"
Brit. Med. J. 1875,2:278

COOPER R, OSSELTON J.W. and SHAW J.C. (1980)
"EEG Technology" 3rd ed.
Butterworth, 1980

DAWSON G.D. (1947)
"Cerebral Responses to Electrical Stimulation of Peripheral Nerve in Man"
J. Neurol. Neurosurg. Psychiat. 1947,10:134-140

GEVINS A.S. (1984)
"Analysis of the Electromagnetic Signals of the Human Brain: Milestones, Obstacles and Goals"
IEEE Trans. Biomed. Eng. Oct. 1984,31(12):833-850

GRATTON G., COLES M.G.H. and DONCHIN E. (1983)
"A New Method for Off-Line Removal of Ocular Artefact"
Electroenceph. and Clin. Neurophysiol. 1983,55:468-484

HILLYARD S.A. and GALAMBOS R. (1970)
"Eye Movement Artifact in the CNV"
Electroenceph. and Clin. Neurophysiol. 1970,28:173-182

IFEACHOR E.C. (1984)
"Investigation of Ocular Artefacts in the Human EEG and Their Removal by a Microprocessor-Based Instrument" Ph.D. Thesis Plymouth Polytechnic, Dec., 1984

JERVIS B.W., ALLEN E. JOHNSON T.E., NICHOLS M.J. and HUDSON N.R. (1984)
"The Application of Pattern Recognition Techniques to the Contingent Negative Vatiation for the Differentiation of Subject Categories"
IEEE Trans. Biomed. Eng. April, 1984,31(4):342-348

JERVIS B.W., IFEACHOR E.C. and ALLEN E.M. (1988)
"The Removal of Ocular Artefacts from the Electroencephalogram: A Review"
Med. and Biol. Eng. and Comput. Jan.1988,26:2-12

LOVELESS N.E. and SANFORD A.J. (1974)
"Slow Potential Correlates of Preparatory Set"
Biol. Psychol. 1974,1:303-314

NICHOLS M.J. (1982)
"An Investigation of the Contingent Negative Variation Using Signal Processing Methods"
Ph.D. Thesis Plymouth Polytechnic, Aug., 1982

QUILTER P.M., MACGILLIVRAY B.B. and WADBROOK D.G. (1977)
"The Removal of Eye Movement Artefact from EEG Signals Using Correlation Techniques" in "Random Signal Analysis"
IEE Conf. Publ. 159:93-100, 1977

ROHRBAUGH J.W., SYNDULKO K. and LINDSLEY D.B. (1976)
"Brainwave Components of the Contingent Negative Variation in Humans"
Science, Mar. 1976,191:1055-1057

ROHRBAUGH J.W., McCALLUM W.C., GAILLARD A.W.K., SIMONS R.F., BIRBAUMER N. and PAPAKOSTOPOULOS D. (1986)
"ERPs Associated With Preparatory and Movement-Related Processes: a Review"
Electroenceph. and Clin. Neurophysiol. Suppl. (Event-Related Potentials) 1986,38:189-229

SHAGASS C. (1972)
"Evoked Brain Potentials in Psychiatry"
Plenum Press, 1972

STRAUMANIS J.J., SHAGASS C. and OVERTON D.A. (1969)
"Problems Associated with Application of the Contingent Negative Variation to Psychiatric Research"
J. Nerv. and Ment. Disease 1969,148(2):170-179

TECCE J.J. and CATTANACH L. (1982)
"Contingent Negative Variation" - Chapter 36 of "Electroencephalography: Basic Principles, Clinical Applications and Related Fields" eds. Niedermeyer E. and Lopes da Silva F.,Urban and Schwarzenberg, 1982

WALTER W.G., COOPER R., ALDRIDGE V.J., McCALLUM W.C. and WINTER A.L. (1964)
"Contingent Negative Variation: An Electric Sign of Sensorimotor Association and Expectancy in the Human Brain"
Nature, July, 1964, 203:380-384

WEERTS T.C. and LANG P.J. (1973)
"The Effects of Eye Fixation and Stimulus and Response Location on the Contingent Negative Variation (CNV)"
Biol. Psychol. 1973,1:1-19

2. OCULAR ARTEFACT REMOVAL

2.1 CHAPTER OUTLINE

The need to perform Ocular Artefact Removal (OAR) is discussed in this chapter. OAR by proportional subtraction of the electro-oculogram (EOG) from the electroencephalogram (EEG) by means of the least squares method of parameter estimation is discussed in section 2.2. Details of the two types of OAR investigated, viz. non-recursive OAR (NR-OAR) and recursive OAR (R-OAR) are also given. Section 2.3 is a discusion of OAR when a response is present and how such a situation causes erroneous parameter estimation and hence incorrect OAR. The means of rectifying this error by modelling the response in the least squares method is then described. Section 2.4 shows the need to investigate the effect on parameter estimation when d.c. (mean) level removal is and is not applied to the data. Section 2.5 is a description of a set of test data created to investigate the above points along with the results obtained, discussion and conclusions. Section 2.6 gives the results obtained when the methods of Section 2.5 were applied to two EEG response (a CNV and an AEP). Section 2.7 gives the recommended processing sequence to be used based on the results of sections 2.5 and 2.6.

2.2 THE BASIS OF THE ELECTRO-OCULOGRAM SUBTRACTION METHOD

OAR by proportional EOG subtraction assumes that the measured EEG is a linear combination of the true EEG and ocular artefact (OA) and that the OA is a linear combination of selected EOGs. The EOG is that electrical potential measured between two electrodes (which may be placed in a variety of configurations) adjacent to the eye. Then for the i th point, in discrete form, of an n point sequence, the general model is:

$$y(i) = \theta_1 x_1(i) + \theta_2 x_2(i) + \ldots + \theta_p x_p(i) + e(i)$$

$$y(i) = \sum_{j=1}^{p} \theta_j x_j(i) + e(i) \qquad i = 1,2,\ldots,n \quad - (2.1)$$

where the $y(i)$ are sampled values of the measured EEG containing OA, the $x_j(i)$ are the sampled values of the measured EOGs (or a function of them), $e(i)$ is the true (background) EEG which is regarded as an error term and the θ_j are proportionality values known as model parameters or transmission coefficients. Assuming that the θ_j can be estimated (by $\hat{\theta}_j$) then the true EEG can be estimated as $\hat{e}(i)$ (the circumflex denotes the estimate of a given value) where:

$$\hat{e}(i) = y(i) - \sum_{j=1}^{p} \hat{\theta}_j x_j(i) \qquad i = 1,2,\ldots,n \quad - (2.2)$$

Equation (2.1) can be written more compactly in matrix form for all n samples as:

$$\underline{Y}_n = \underline{X}_n \underline{\theta}_n + \underline{E}_n \qquad - (2.3)$$

where: $\underline{Y}_n = \begin{bmatrix} y(1) \\ y(2) \\ \cdot \\ \cdot \\ \cdot \\ y(n) \end{bmatrix}$ $\underline{X}_n = \begin{bmatrix} x^T(1) \\ x^T(2) \\ \cdot \\ \cdot \\ \cdot \\ x^T(n) \end{bmatrix}$ $\underline{\theta}_n = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \cdot \\ \cdot \\ \cdot \\ \theta_p \end{bmatrix}$ $\underline{E}_n = \begin{bmatrix} e(1) \\ e(2) \\ \cdot \\ \cdot \\ \cdot \\ e(n) \end{bmatrix}$

and $x^T(i) = [x_1(i), x_2(i), \ldots, x_p(i)]$

Then, similarly, equation (2.2) can be written:

$$\hat{\underline{E}}_n = \underline{Y}_n - \underline{X}_n \hat{\underline{\theta}}_n \qquad - (2.4)$$

2.2.1 THE LEAST SQUARES METHOD OF PARAMETER ESTIMATION

The basis of least squares estimation is to compute $\hat{\theta}_j$s which minimise the sum of squares of the estimated error term (the true, background EEG), $\hat{e}(i)$. This can be performed using the ordinary least squares (OLS) method (section 2.2.2) in which the $\hat{\theta}_j$s are constant or the recursive least squares (RLS) method can be used to give values of $\hat{\theta}_j$s which are updated from point to point and can thus vary with time.

Then the least squares estimate for $\underline{\theta}_n$ is $\hat{\underline{\theta}}_n$ given by (IFEACHOR et al 1986a and b).:

$$\hat{\underline{\theta}}_n = (\underline{X}_n^T \underline{X}_n)^{-1} \underline{X}_n^T \underline{Y}_n \qquad - (2.5)$$

A variety of EOG combinations can be used for the model of equation (2.1). A recent review (JERVIS et al, 1988) found an adequate model to be:

$$y(i) = \theta_1.VEOG_R(i) + \theta_2.HEOG_R(i) + \theta_3.HEOG_L(i) \quad -(2.6)$$
$$i = 1,2,\ldots,n$$

where the vertical EOG of the right eye is denoted $VEOG_R$ and the horizontal EOGs of left and right eyes are $HEOG_L$ and $HEOG_R$ respectively. A model which may be marginally better is (ibid):

$$y(i) = \theta_1.HEOG_R(i).HEOG_L(i) + \theta_2.VEOG_R(i) + \theta_3.HEOG_R(i) + \theta_4.HEOG_L(i) \quad -(2.7)$$

This was chosen as the model to be used in processing real data (section 2.6) and 'realistic' test data (section 2.5).

2.2.2 NON-RECURSIVE OCULAR ARTEFACT REMOVAL

NR-OAR estimates the θ_j values by using the whole duration of a record to compute a value of $\hat{\theta}_j$ which applies throughout the record (QUILTER et al, 1977; FORTGENS and DE BRUIN, 1983; JERVIS et al, 1985). It is to be noted that QUILTER et al (1977) used correlation techniques to obtain values for the $\hat{\theta}_j$s. However IFEACHOR (1984) has shown that correlation and OLS methods lead to identical expressions for the $\hat{\theta}_j$s and thus that the two methods give alternative ways of visualising the determination of $\hat{\theta}_j$.

Although the $\hat{\theta}_j$s are constant throughout a record they are computed for each record and thus allow for inter-trial variability and between subject variability.

In essence the NR-OAR method seeks to minimise:

$$J = \sum_{i=1}^{n} [\hat{e}(i)]^2 = [\sum_{i=1}^{n} \{y(i) - \sum_{j=1}^{p} \hat{\theta}_j x_j(i)\}^2] \quad - (2.8)$$

by computation of appropriate values of $\hat{\theta}_j$ (see Appendix A1 for details).

2.2.3 RECURSIVE OCULAR ARTEFACT REMOVAL

R-OAR computes updated values of the $\hat{\theta}_j$s by using the more recent data and discarding the earlier data. Thus equation (2.8) is modified to:

$$J = \sum_{i=1}^{n} \gamma^{n-i} [\hat{e}(i)]^2 = [\sum_{i=1}^{n} \gamma^{n-i} \{y(i) - \sum_{j=1}^{p} \hat{\theta}_j x_j(i)\}^2] \quad - (2.9)$$

(IFEACHOR, 1984; IFEACHOR et al 1986a) where γ is known as the 'forgetting factor' and allows the tracking of a slowly varying parameter. Typically γ is between 0.98 and 1 since smaller values assign too much weight to the more recent values giving wildly fluctuating results (IFEACHOR, 1984; IFEACHOR et al, 1986a).

Two advantages of R-OAR as compared to NR-OAR are that it is adaptive with the $\hat{\theta}_j$ being continuously updated, and that it makes possible automatic on-line OAR.

2.3 OCULAR ARTEFACT REMOVAL WITH A RESPONSE PRESENT

2.3.1 ERRONEOUS PARAMETER ESTIMATION DUE TO PRESENCE OF NON-RANDOM RESPONSE

It was decided to compare the two correction methods described in the previous section by applying them to EEGs containing the CNV and AEP responses. NR-OAR was implemented using the methods of QUILTER et al (1977) but extended to four model parameters (JERVIS et al, 1980; NICHOLS, 1982; JERVIS et al, 1985). This used subroutine NROARM (Appendix A2) which was adapted from subroutine EYECOR of NICHOLS (1982). R-OAR was implemented using the methods described in IFEACHOR et al (1986a and b). This used subroutine RCOARM (Appendix A2) based on the software in IFEACHOR (1984). The model used was that given in equation (2.7) and was selected because it allows for the effects of vertical ($VEOG_R$) and horizontal ($HEOG_L$ and $HEOG_R$) eye movements and attempts to compensate for possible non-linear interaction between the horizontal EOGs ($HEOG_L.HEOG_R$).

Plots of the $\hat{\theta}_j$s against time showed the R-OAR algorithm took typically 2s and sometimes as long as 5s to converge. Since this is an appreciable length of the 8s record R-OAR was applied twice. The $\hat{\theta}_j$s obtained at the end of the first computation were used as the starting values for the second. This gave much shorter convergence times and gave confidence that correct parameter estimates were used for those parts of the record of interest.

Figures 2.1, 2.2 and 2.3 show the no-OAR, NR-OAR and R-OAR corrected EEGs from the same subject. Each waveform is a 32 trial average. For Figures 2.2 and 2.3 OAR was applied to each trial prior to inclusion in the averaging process.

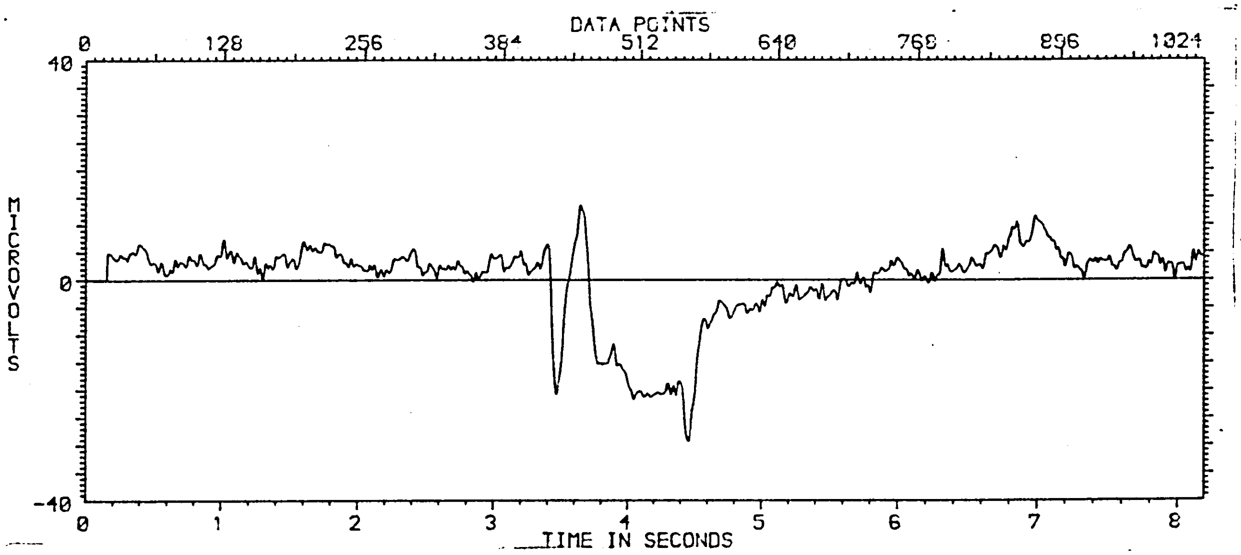

Figure 2.1

The averaged 1 s ISI CNV of a co-operative subject

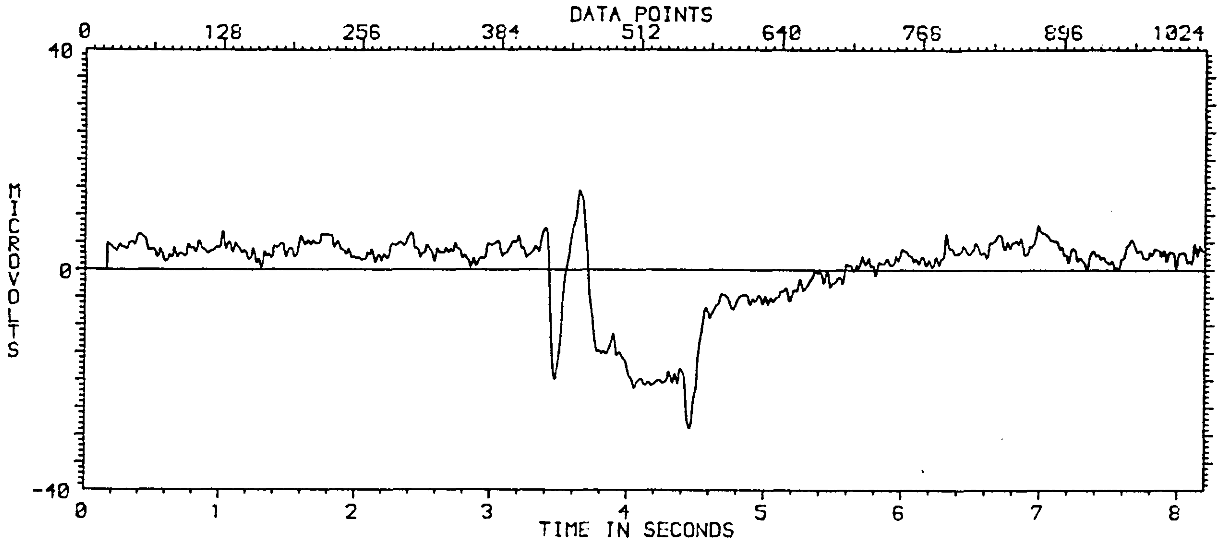

Figure 2.2

The averaged 1s ISI CNV of the same subject as Fig. 2.1 subsequent to implementation of NR-OAR (dc = 0)

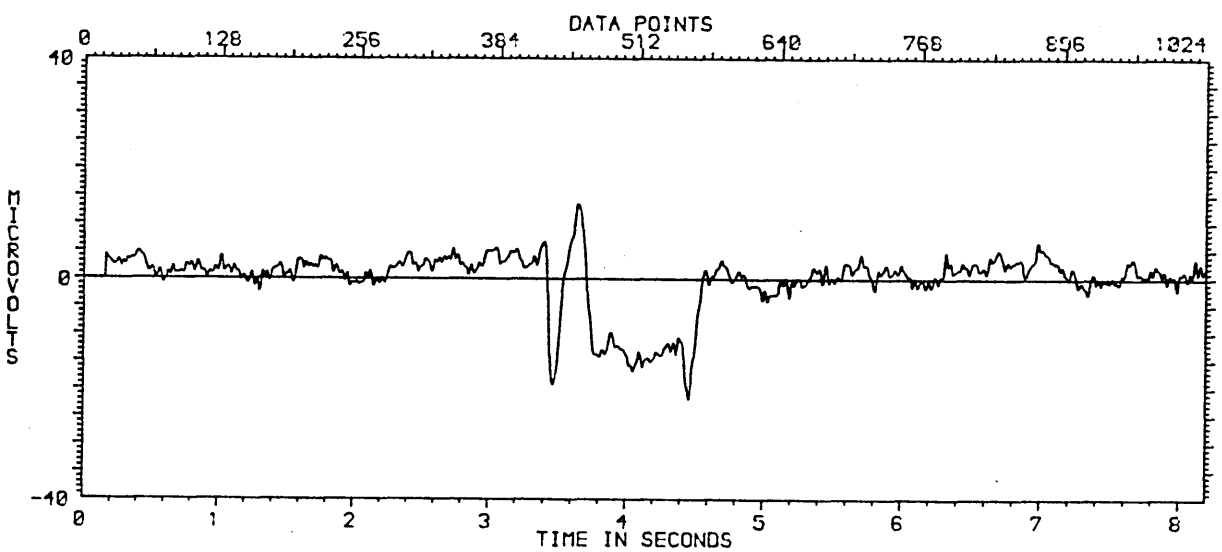

Figure 2.3

The averaged 1s CNV of the same subject as Fig. 2.1 subsequent to implementation of R-OAR (dc = 0)

The CNVs shown were obtained from a normal subject who had been asked to refrain from moving their eyes or blinking until after completion of the CNV paradigm. Inspection of the EOGs of each trial for this subject showed that the CNV epoch did not contain OA and that on numerous occasions (24 out of 32 trials) OA was present in the post-CNV epoch (this is manifest as the hump centered at 7s in Figure 2.1 which is smeared in this fashion because of the averaging process). Inspection of the three waveforms shows that the R-OAR corrected CNV response is distorted when compared to the no-OAR and NR-OAR waveforms (between which no difference can be seen for the CNV or AEPs).

This shape modification is the result of a deficency in the two OAR methods as previously implemented. The least squares method computes the $\hat{\theta}_j$s by minimising the sum of squares of the error terms (the e(i) or $EEG_t(i)$) on the assumption that they are random. However the CNV is not a random signal and its presence in the measured EEG will cause incorrect estimates of the θ_js and hence of the true EEG and response itself. This follows since equation (2.1) must now include the response R(i):

$$y'(i) = \sum_{j=1}^{p} \theta_j x_j(i) + R(i) + e(i) \qquad i = 1,2,\ldots,n \quad - (2.10)$$

where y'(i) is the measured EEG including the response. Equation (2.2) is now:

$$\hat{e}(i) = y'(i) - \sum_{j=1}^{p} \hat{\theta}_j x_j(i) - R(i) \qquad i = 1,2,\ldots,n \quad - (2.11)$$

However both existing OAR methods do not include the R(i) term in equation (2.11) which thus becomes:

$$\hat{\hat{e}}(i) = y'(i) - \sum_{j=1}^{p} \hat{\hat{\theta}}_j x_j(i) \qquad i = 1,2,\ldots,n \quad - (2.12)$$

where $\hat{\hat{\theta}}_j$ are now the incorrect estimates of the θ_j and $\hat{\hat{e}}(i)$ is an incorrect estimate for the background EEG.

This erroneous estimation will occur for both methods but its effects can be anticipated to differ in form and magnitude between NR-OAR and R-OAR. NR-OAR has fixed values of $\hat{\theta}_j$ which are computed from the data of the whole record. However R-OAR attaches more weight to the most recent data, discarding earlier data, and can have values of $\hat{\theta}_j$ which vary with time due to updating of the estimates. Thus it can be anticipated that incorrect θ_j estimation due to NR-OAR could lead to magnitude and/or level differences but not shape distortion, whereas such a problem could occur in R-OAR in which the $\hat{\theta}_j$ may be varying with time.

2.3.2 RESPONSE MODELLING IN THE APPLICATION OF OCULAR ARTEFACT REMOVAL

The previous section showed how the presence of a response can cause incorrect estimates of the transmission coefficients and hence of the true EEG and response itself when OAR is carried out. Here a possible remedy is given.

It has been seen that inclusion of the response, R(i), in the general model of equation (2.1) to give equation (2.10) represents the correct model to be used in OAR. However there are cases when the true shape of the response is unknown. One such case is when the response is obscured by OA. This is especially true of the CNV and was introduced in Section 1.1.5. Here it is essential to remove the OA in order to elicit the true shape of the response. However, as has been seen, the response presence corrupts the OAR process leading to incorrect estimates of the background EEG and the response. One solution to this impasse is to model the response.

In the rest of this text to avoid confusion equation (2.1) shall be referred to as the 'general model' while the response in equation (2.10) shall be replaced by a 'response model' or 'model of the response'. Denoting the response model as RM(i) for the ith point, equation (2.10) becomes:

$$y'(i) = \sum_{j=1}^{p} \theta_j x_j(i) + RM(i) + e(i) \quad i = 1,2,\ldots,n \qquad -(2.13)$$

where $y'(i) = y(i) + R(i)$ $\qquad -(2.14)$

The response model can comprise a number of components, say q and so equation (2.13) can be written:

$$y'(i) = \sum_{j=1}^{p+q} \theta_j x_j(i) + e(i) \qquad -(2.15)$$

where:

$$RM(i) = \sum_{j=p+1}^{p+q} \theta_j x_j(i) \qquad -(2.16)$$

The $x_j(i)$, for $j \geqslant p+1$, now represent the components of the response model. For $j \leqslant p$ the $x_j(i)$ still represent the measured EOGs.

The form of equation (2.15) means that the existing OAR software (NICHOLS, 1982 and IFEACHOR, 1984) can be utilised by suitable modifications to extend the number of terms which can be corrected for (the existing versions were both written to handle up to five terms, the four EOG channels plus the measured EEG).

It is to be noted that the estimates of the transmission coefficients associated with the response model components are used only in so far as giving improved estimates of the EOG channel transmission coefficients is concerned. This is because an estimate of both background EEG and response is required. Thus equation (2.2) becomes:

$$\hat{e}'(i) = y'(i) - \sum_{j=1}^{p} \hat{\theta}_j x_j(i) \qquad i = 1,2,\ldots,n \quad -(2.17)$$

where: $$\hat{e}'(i) = \hat{e}(i) + \hat{R}(i) \quad -(2.18)$$

2.4 THE EFFECT OF D.C. LEVEL REMOVAL

This section shows that the data pre-processing procedure of subtracting the mean (or d.c. level) of the data from the data may affect the parameter estimates.

Consider NR-OAR without modelling (NR-OAR-NM) in which there is one EEG channel and one EOG channel ($EEG_m(i)$ and $EOG_m(i)$ respectively) and in which d.c. levels remain. Denote the corresponding means for each channel as μ_{EEG} and μ_{EOG}. Thus equation (2.1) can be written for this case as:

$$EEG_m(i) = \theta . EOG_m(i) + EEG_t(i) \qquad i = 1,2,\ldots,n \quad -(2.19)$$

where $EEG_t(i)$ is the true or background EEG.

Equation (2.2) can be written for this case as:

$$\widehat{EEG}_t(i) = EEG_m(i) - \hat{\theta} . EOG_m(i) \quad i = 1,2,\ldots,n \quad -(2.20)$$

and hence equation (2.8) becomes:

$$J = \sum_{i=1}^{n} [\widehat{EEG}_t(i)]^2 = \sum_{i=1}^{n} [EEG_m(i) - \hat{\theta}.EOG_m(i)]^2 \qquad -(2.21)$$

from which the expression for the estimated value of θ, $\hat{\theta}$ is:

$$\hat{\theta} = \frac{\sum_{i=1}^{n} EOG_m(i).EEG_m(i)}{\sum_{i=1}^{n} [EOG_m(i)]^2} \qquad -(2.22)$$

If now d.c. levels are removed from both $EEG_m(i)$ and $EOG_m(i)$, i.e. the means of the data are removed from the data, equation (2.22) becomes:

$$\hat{\theta}\Big|_{dc=0} = \frac{\sum_{i=1}^{n} [EOG_m(i)-\mu_{EOG}][EEG_m(i)-\mu_{EEG}]}{\sum_{i=1}^{n} [EOG_m(i)-\mu_{EOG}]^2} \qquad -(2.23)$$

$$= \frac{\sum_{i=1}^{n} EOG_m(i).EEG_m(i)-n\mu_{EOG}.\mu_{EEG}}{\sum_{i=1}^{n} [EOG_m(i)]^2 - n\mu_{EOG}^2} \qquad -(2.24)$$

Thus, in general $\hat{\theta} \neq \hat{\theta}\big|_{dc=0}$ and hence the need to investigate the effects of d.c. level removal arises.

2.5 TEST DATA

In order to test the truthfulness (or otherwise) of the ideas concerning the effect on parameter estimation when OAR is performed with a response present a number of test data sets involving simulated CNV and OAs were created and processed. The degree of effectiveness of response modelling was also to be assessed and a comparison of results was planned when d.c. level removal from the data is and is not carried out. The removal of d.c. levels from the data will be indicated by ($dc = 0$) and its inclusion will be denoted ($dc \neq 0$). The use of known test data makes such investigations easier and more accurate since the response shapes are known.

2.5.1 DESCRIPTION OF TEST DATA

The simulated data comprised four sets: three involved simplified data consisting of one EEG channel, one EOG channel and one CNV model component, the fourth case used more realistic test data comprising one EEG channel, four EOG channels and two CNV model components. Further details are given below (Sections 2.5.1.1-2.5.1.3 use the simplified data while Section 2.5.1.4 uses the realistic test data). For the simplified data the general model is:

$$y(i) = \sum_{j=1}^{2} \theta_j x_j(i) + e(i) \qquad i = 1,2,\ldots,1024 \qquad -(2.25)$$

where: θ_1 is the transmission coefficient for the EOG channel and θ_2 is the transmission coefficient for the response model component. In Sections 2.5.1.1 - 2.5.1.3 the 'true' value of θ_1 was set to 0.2.

In the case of the more realistic test data the general model is:

$$y(i) = \sum_{j=1}^{6} \theta_j x_j(i) + e(i) \qquad i = 1,2,\ldots,1024 \qquad -(2.26)$$

where θ_1 - θ_4 are the transmission coefficients for the EOG channels and θ_5 and θ_6 are the transmission coefficients for the response model components.

2.5.1.1 OA AND A SEPARATE RESPONSE

As a start the simplest experimental condition possible was simulated in which a CNV response was present along with a temporally distinct OA. The simulated measured EEG representing this (after dc level removal (d.c. = 0)) is given in Figure 2.4 while Figure 2.5 shows the simulated measured EOG without d.c. level removal (dc $\neq$ 0).

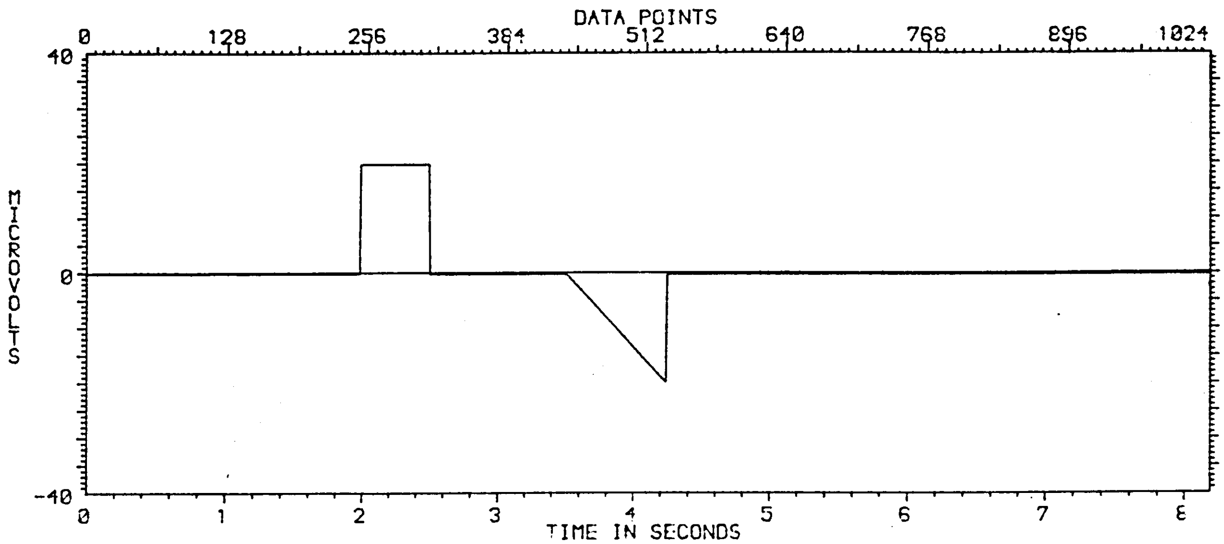

Figure 2.4

Simulated measured EEG which contains an OA and a response

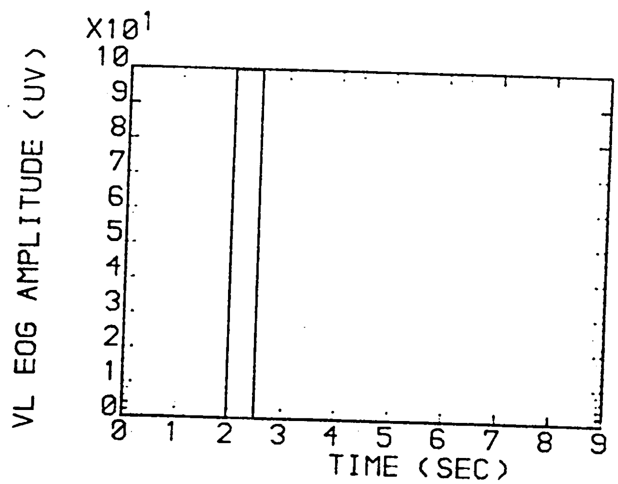

Figure 2.5

The simulated EOG corresponding to the OA of Figure 2.4

The single component response model (dc $\neq$ 0) is shown in Figure 2.6.

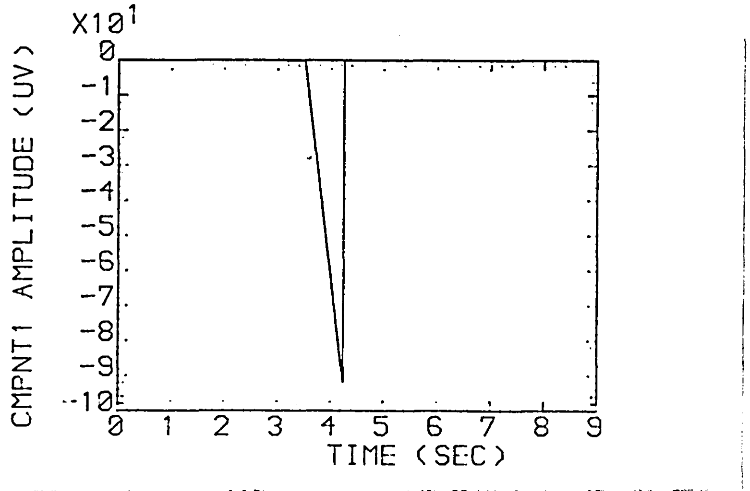

Figure 2.6

Linearly modelled response

Figure 2.7 (dc = 0) shows the EEG corrected for OA non-recursively and without modelling (NR-OAR-NM).

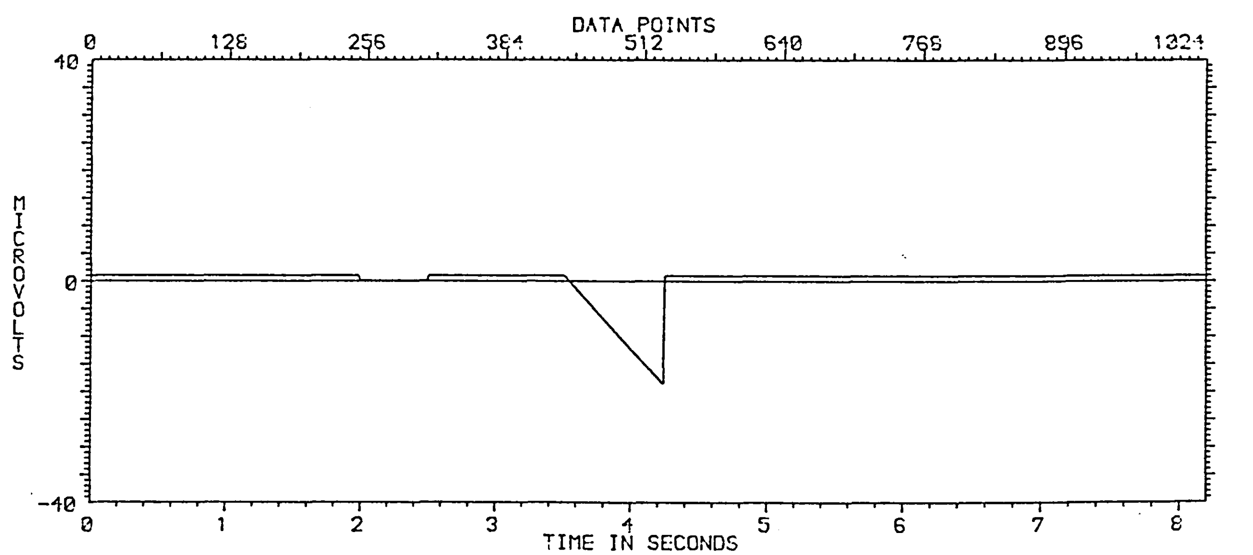

Figure 2.7

The NR-OAR-NM (dc = 0) corrected EEG

The estimated CNV differs from the known one only in that there is a small (1μV) but constant level shift. The shape, slope and maximum amplitude (relative to the baseline) are unchanged and there is an OA remnant (in the form of overcorrection of the measured EEG) of 1μV (5% of the original OA). The correction, with d.c. level removal, gave rise to the $\hat{\theta}$ (the estimated value) of 0.20984 , compared with the known value of 0.2, i.e., 4.92% in error.

Figure 2.8 (dc = 0) shows the NR-OAR corrected EEG when modelling of the CNV was included (NR-OAR-WM).

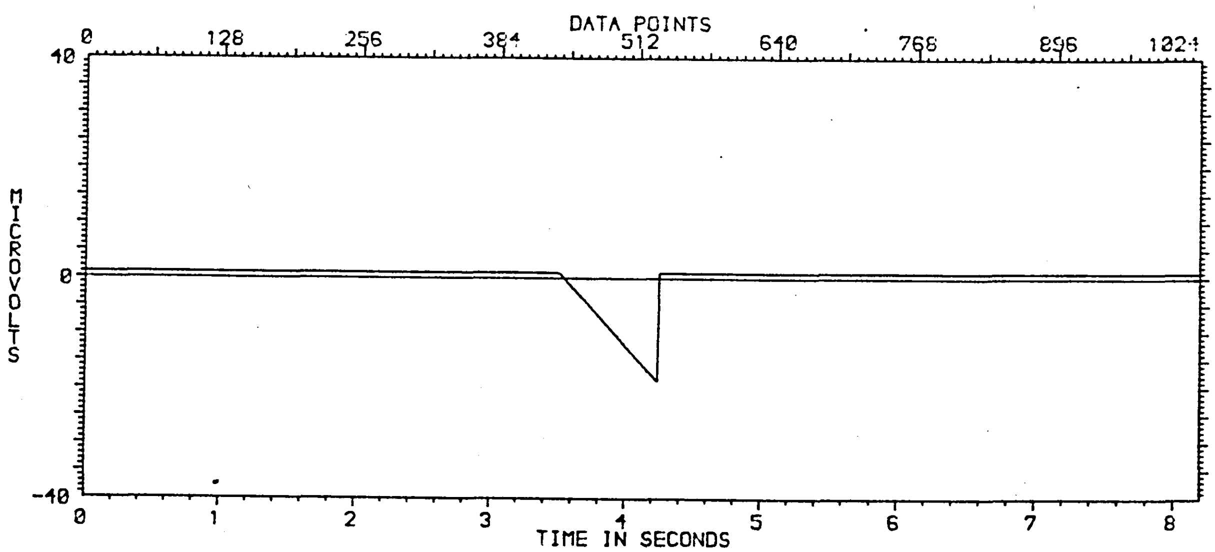

Figure 2.8

The NR-OAR-WM (dc = 0) corrected EEG

There are no visual traces of the OA and the CNV shape is unaltered. The value of $\hat{\theta}$ with d.c. level removal is 0.20047 differing from the known value of 0.2 by 0.24%. Hence it has been established for these simplified data that successful OAR can be achieved non-recursively provided the response is modelled.

Figure 2.9 (dc = 0) shows the corrected EEG obtained using the recursive method, but with no modelling (R-OAR-NM).

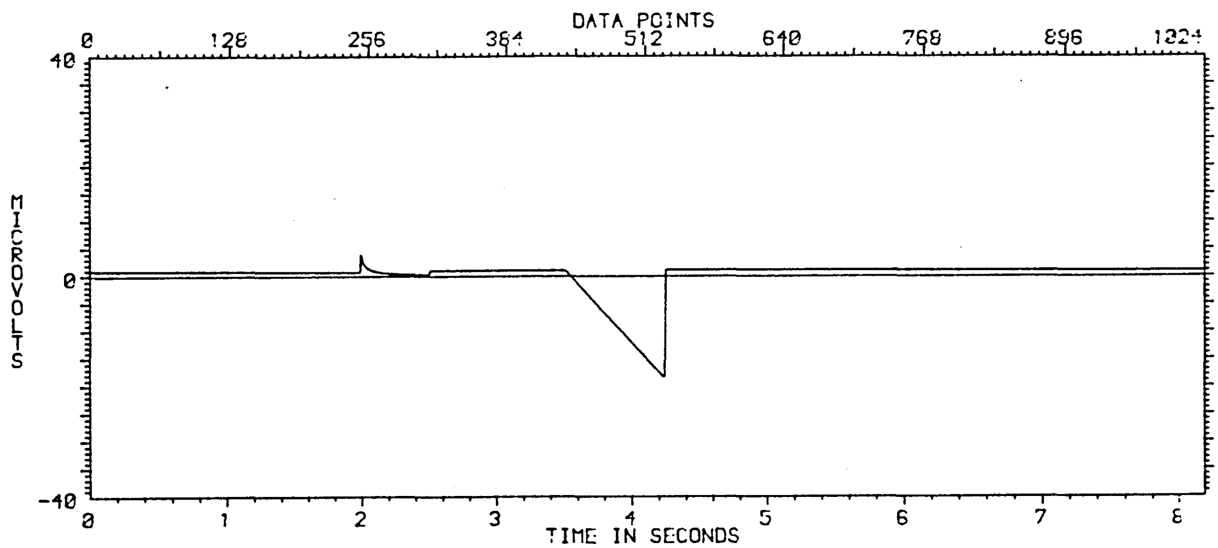

Figure 2.9

The R-OAR-NM (dc = 0) corrected EEG

A vestige of OA remains varying from +2.8μV to -0.7μV, relative to the EEG baseline. It can also be observed that there exists a small difference in pre- and post-OA EEG baseline of ~0.5μV. It appears that the CNV is unaltered. Scrutiny of the recursively corrected EEG when modelling is applied (R-OAR-WM) and the d.c. level is removed (dc = 0) show that the CNV shape is unaltered and no OA remnant is discernible, i.e., the EEG is the same as in Figure 2.8. In addition the pre- and post-OA baseline difference of Figure 2.9 has been eliminated.

Figure 2.10 (dc = 0) is a plot of $\hat{\theta}$ against sample number for the R-OAR-NM case showing a variation of $\hat{\theta}$ value during the record which accounts for the pre- and post-OA EEG baseline difference observed in Figure 2.9.

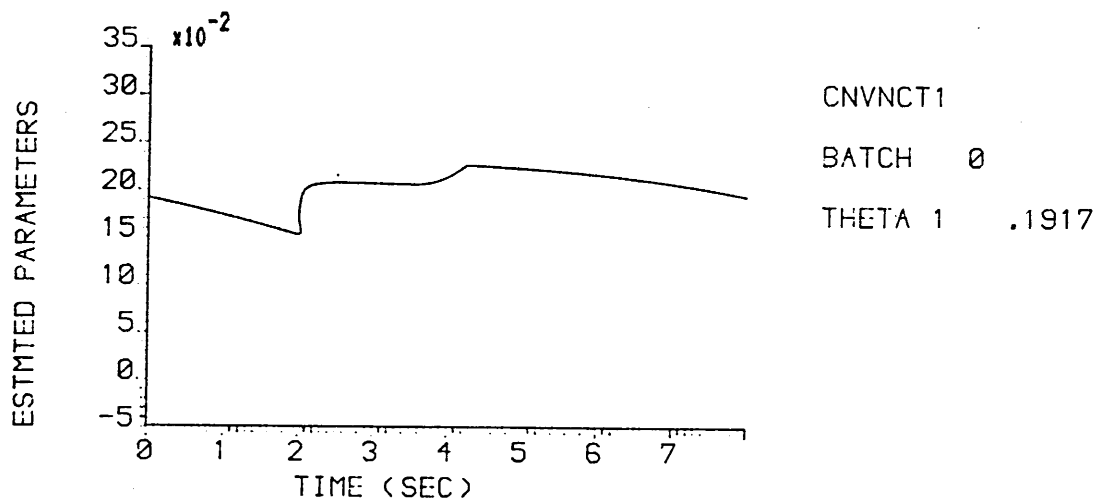

Figure 2.10

Variation of $\hat{\theta}$ with sample number for R-OAR-NM (dc = 0)

The following observations can be made: the abrupt change in $\hat{\theta}$ at t = 2 corresponds to the onset of OA; the more gradual increase in $\hat{\theta}$ commences in the vicinity of the start of the CNV (there being a slight delay ~0.2s); the change in sign of the slope of $\hat{\theta}$ occurs at CNV termination. Figure 2.11 (dc = 0) shows the R-OAR-WM case in which the second trace (labelled "THETA 2") is the estimate for the CNV model component parameter, the $\hat{\theta}_2$ of equation (2.25).

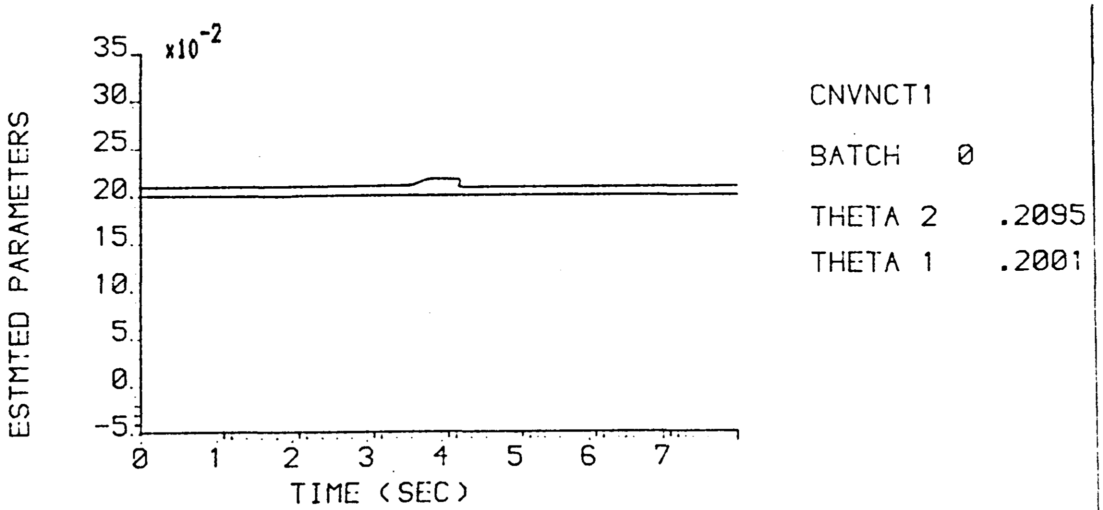

Figure 2.11

Variation of $\hat{\theta}$ with sample number for R-OAR-WM (dc = 0)

This figure shows immediately the very good estimate of θ obtained when modelling is introduced and the elimination of the fluctuating $\hat{\theta}$ values noted in Figure 2.10. The only discernible variation is for $\hat{\theta}_2$ during the CNV presence.

Figure 2.12 (dc $\neq$ 0) shows the NR-OAR-NM corrected EEG, derived from data from which the d.c. level has not been removed.

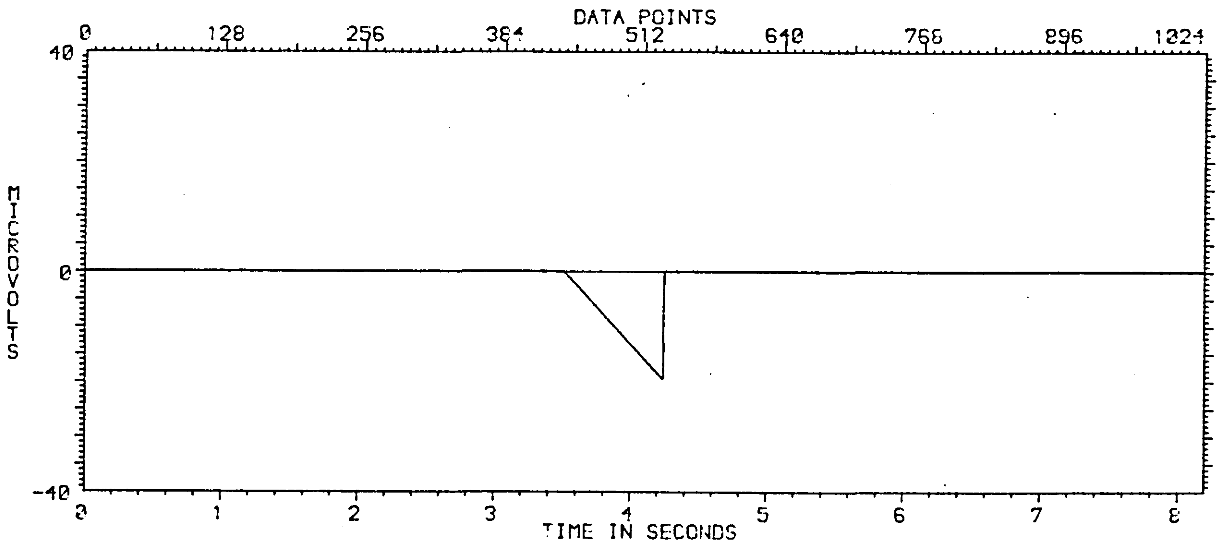

Figure 2.12

The NR-OAR-NM (dc $\neq$ 0) corrected EEG

This is seen to exhibit neither OA remnant nor CNV shape modification. This was found to be the case for all processing options (NR-OAR-NM, NR-OAR-WM, R-OAR-NM and R-OAR-WM). The value of $\hat{\theta}$ obtained by NR-OAR-NM and NR-OAR-WM was identical and equalled 0.20034, i.e., 0.17% in error. Figure 2.13 (dc $\neq$ 0) shows the plot of $\hat{\theta}$ variation when R-OAR-NM was performed. An identical plot was obtained when modelling was applied (except, of course, for the addition of the estimate of θ_2, the model component).

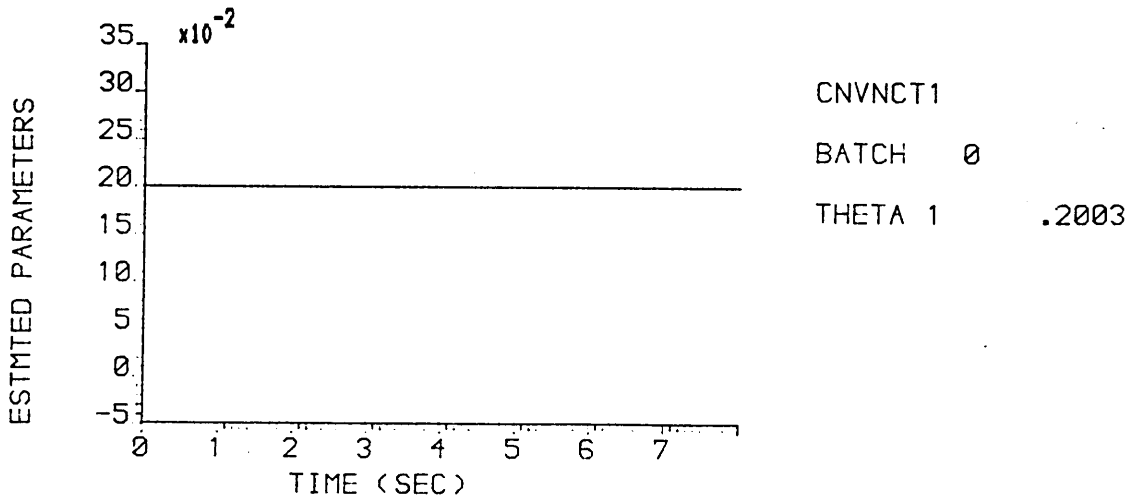

Figure 2.13

Variation of $\hat{\Theta}$ with sample number for R-OAR-NM (dc $\neq$ 0)

For these simple data, the conclusions are that the response must be modelled if the d.c. levels of the data have been removed, and that response modelling gives good results in all cases.

For simplicity no EEG or EOG background noise was simulated. This condition can be justified by noting that the worst case error in estimating the Θ values will be when there is no noise. Consider an N point sequence, where for simplicity N is an odd integer next in sequence after a multiple of 4. Denote the background EEG as s(i) and let this be constant at a voltage v_s. Furthermore let the 'noise' be a voltage waveform of value $\pm v_n$ relative to v_s. Then:

$$s(i) = \begin{cases} v_s - v_n & i = 2,6,10,\ldots,N-7,N-3 \quad \text{i.e. } \dfrac{N-1}{4} \text{ terms} \\ v_s & i = 1,3,5,\ldots,N-2,N \quad \text{i.e. } \dfrac{N+1}{2} \text{ terms} \\ v_s + v_n & i = 4,8,12,\ldots,N-5,N-1 \quad \text{i.e. } \dfrac{N-1}{4} \text{ terms} \end{cases} \qquad -(2.27)$$

or for no 'noise':

$$s(i) = v_s \qquad i = 1,2,3,\ldots,N \qquad -(2.28)$$

The above are depicted in Figure 2.14.

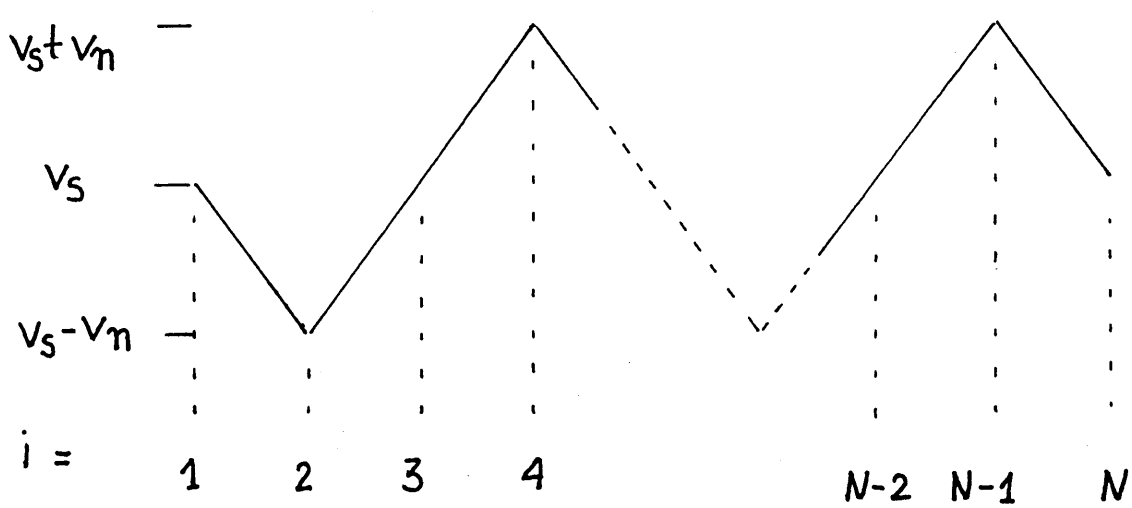

Figure 2.14

Illustration of effect of noise on sum of squares, J

Then for no 'noise':

$$J = \sum_{i=1}^{N} [s(i)]^2 = Nv_s^2 \qquad -(2.29)$$

And when 'noise' is present:

$$J_n = \left(\frac{N+1}{2}\right)v_s^2 + \left(\frac{N-1}{4}\right)(v_s - v_n)^2 + \left(\frac{N-1}{4}\right)(v_s + v_n)^2$$

$$= Nv_s^2 + \left(\frac{N-1}{2}\right)v_n^2 \qquad -(2.30)$$

Then comparing (2.29) and (2.30) J_n must always be greater than J. Hence any given non-random response in the EEG will have a greater effect on J than J_n.

2.5.1.2 SEPARATE OA AND A RESPONSE HAVING AN ADDITIONAL OA

In the following d.c. levels have been removed from the data yielding Figures 2.15, 2.17 - 2.21.

The remaining figures were obtained from data possessing d.c. levels.

Since OAs may be superimposed on the responses this situation was investigated using simple data. The simulated measured EEG (dc = 0) is shown in Figure 2.15.

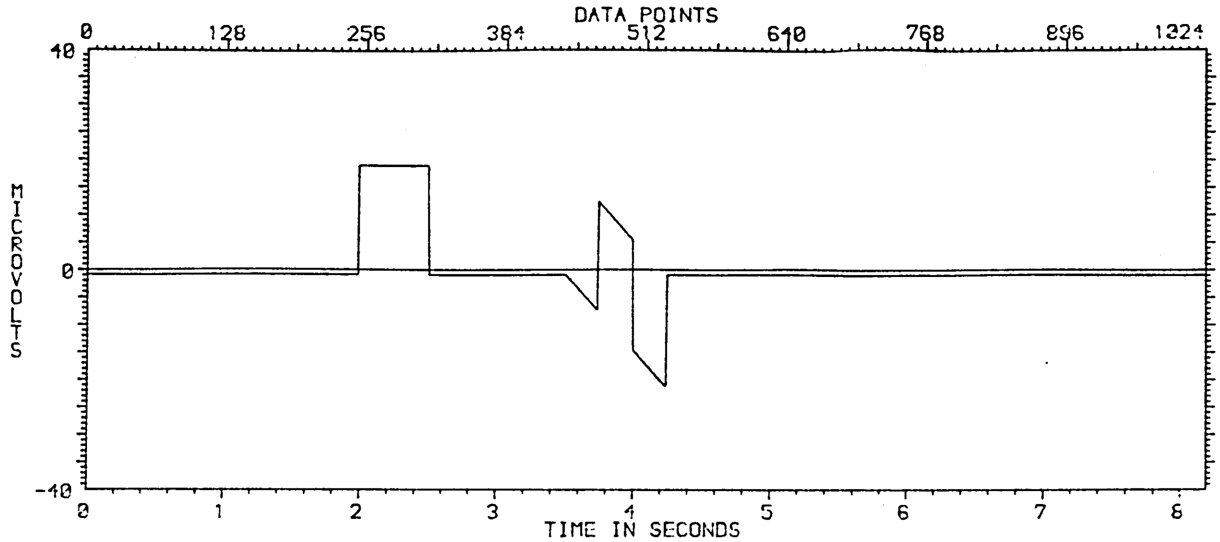

Figure 2.15

Simulated measured EEG which contains an OA and an OA superimposed on the response (dc = 0)

The EOG causing the OAs is given in Figure 2.16 (dc ≠0).

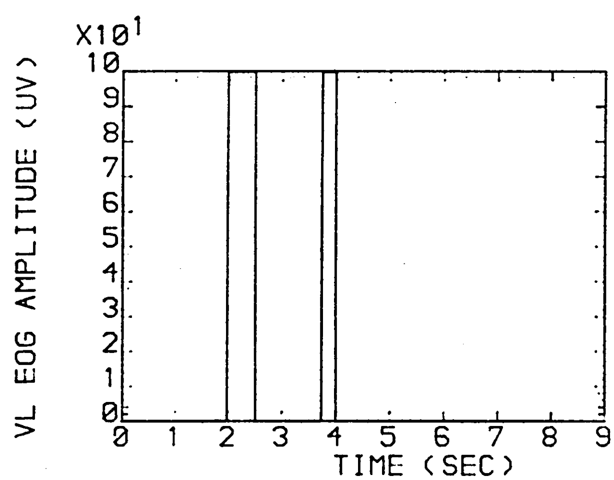

Figure 2.16

The simulated EOG (dc ≠ 0) corresponding to the OAs of Figure 2.15

Using NR-OAR-NM gives the corrected EEG of Figure 2.17 (dc = 0).

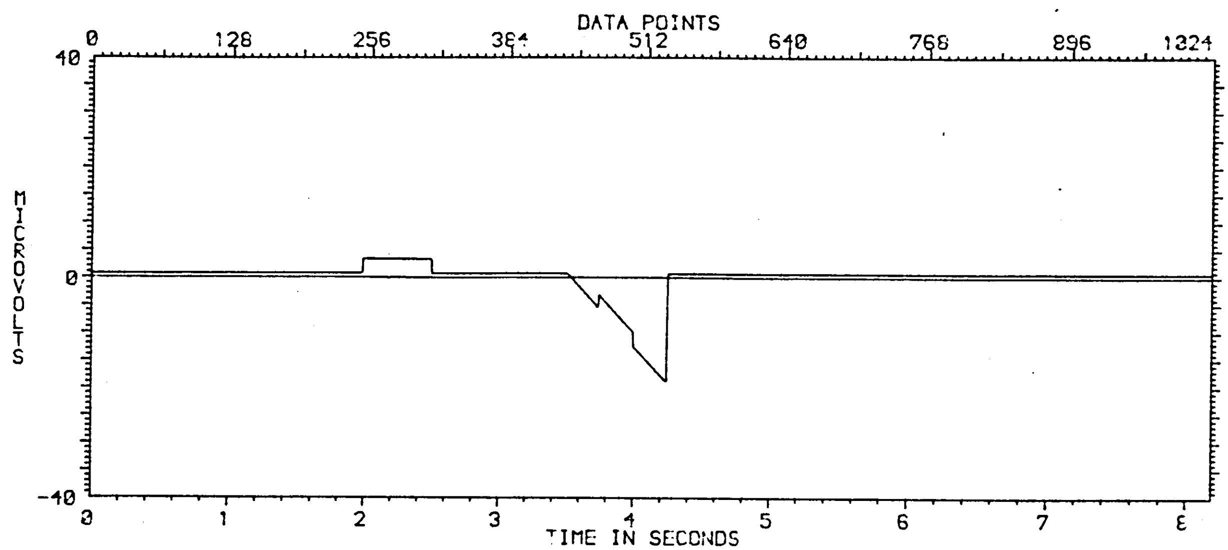

Figure 2.17

The NR-OAR-NM (dc = 0) corrected EEG

Incomplete removal of both OAs is observed, there remaining 13% of the original OA in each case. With NR-OAR-WM (dc = 0) the corrected EEG showed no trace of either OA, i.e., it was as for Figure 2.8. The values of $\hat{\theta}$ in these cases were 0.17389 (13.1% in error) and 0.19863 (0.7% in error) respectively. R-OAR-NM (dc = 0) gave the corrected EEG of Figure 2.18 which shows incomplete OA removal.

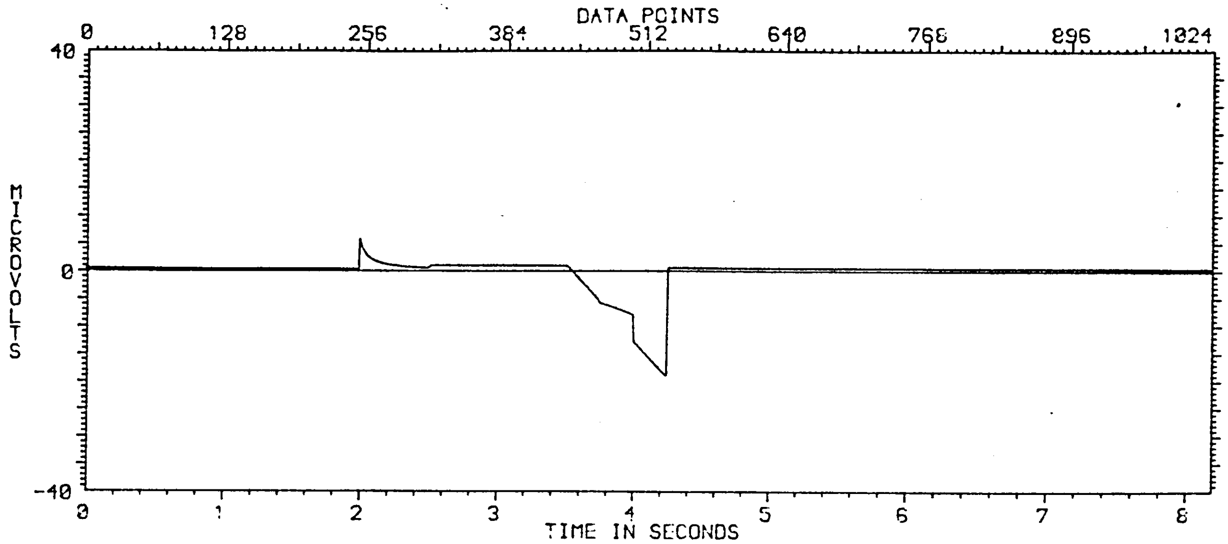

Figure 2.18

The R-OAR-NM (dc = 0) corrected EEG

The first OA is initially undercorrected by 6μV but is reduced to ~0.5μV of overcorrection. The second OA becomes worse with CNV development being ~5.5μV undercorrected at OA termination. There is also a small distortion of the CNV and EEG baseline differences are noted in the different regions of the corrected EEG. Figure 2.19 (dc = 0) shows the R-OAR-WM corrected EEG. The OAs are now almost completely removed (scrutiny of the regions in which the OAs were present reveals slight corrected EEG perturbations). The EEG baseline differences have been eliminated and the CNV appears undistorted.

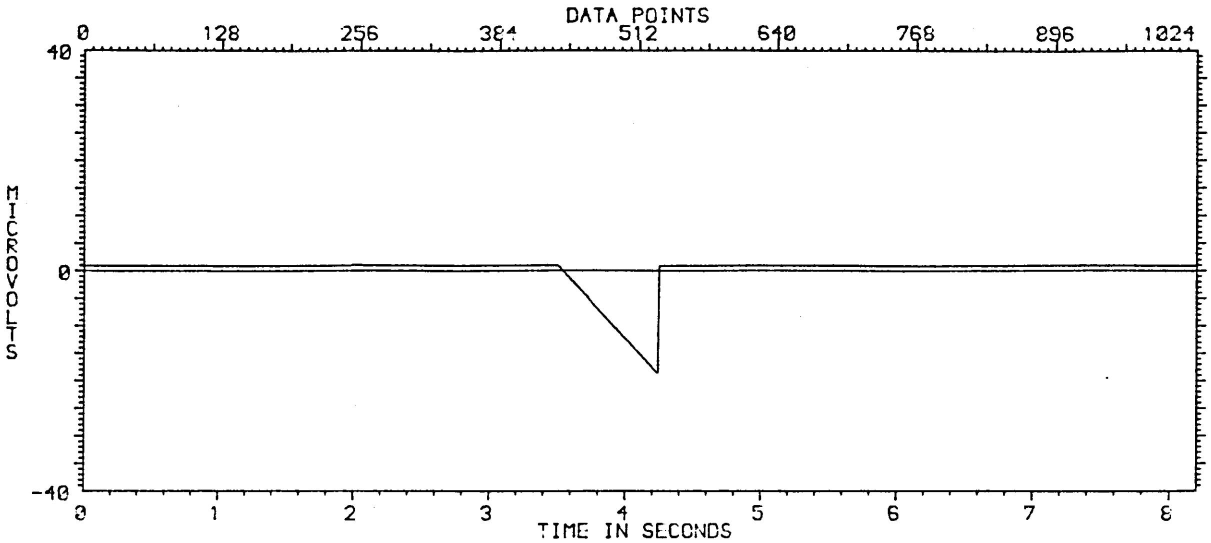

Figure 2.19

The R-OAR-WM (dc = 0) corrected EEG

The variations of $\hat{\theta}$ with sample number for R-OAR-NM (dc = 0) and R-OAR-WM (dc = 0) are shown in Figures 2.20 and 2.21 respectively.

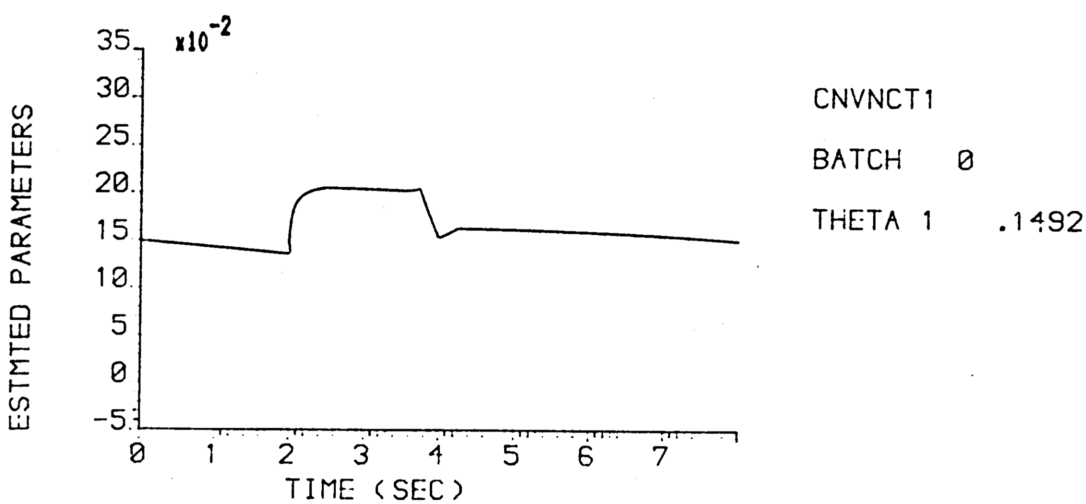

Figure 2.20

Variation of $\hat{\theta}$ with sample number for R-OAR-NM (dc = 0)

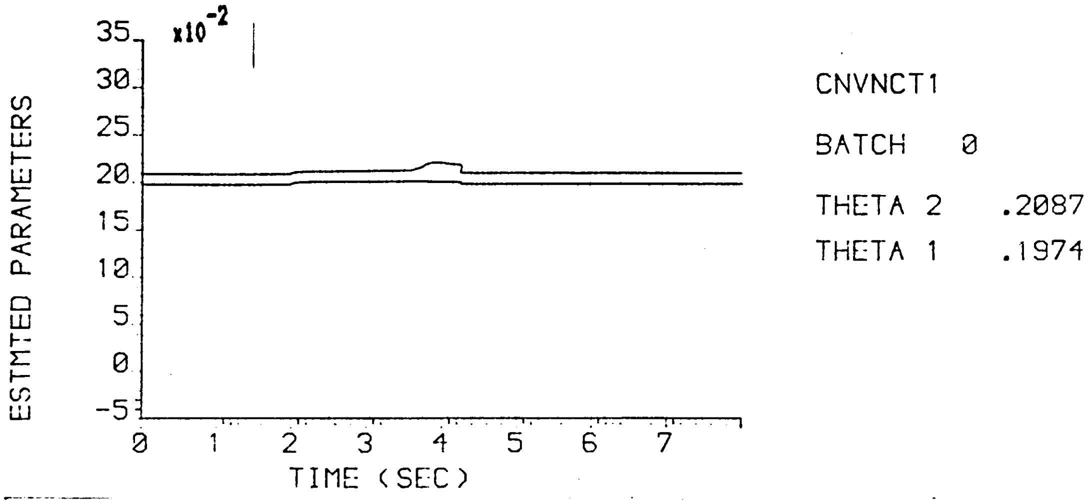

Figure 2.21

Variation of $\hat{\underline{\theta}}$ with sample number for R-OAR-WM (dc = 0)

In the former $\hat{\theta}$ suffers a step change at the rising edge of the first OA with another rapid change starting at the rising edge of the second OA. The final values of $\hat{\theta}$ of 0.1492 was in error by 25.4% but between the rising edges of the two OAs $\hat{\theta}$ was much closer to the true value of 0.2. With response modelling the OAs cause much smaller change in $\hat{\theta}$ which, as expected for this one type of artefact, is more nearly constant, the final value of 0.1974 being within 1.3% of the true value. It is to be concluded once more that efficient OAR requires response modelling which also overcomes the response distortion introduced by R-OAR (Figures 2.18 and 2.19).

Study of the NR-OAR-NM, NR-OAR-WM, R-OAR-NM and R-OAR-WM corrected EEGs, in which d.c. levels remain, were similar to their d.c. level-removed counterparts, excepting that their baselines remained at zero. The non-recursive estimates of θ with and without response modelling were 0.19863 (0.7% in error) and 0.16379 (22.1% in error) respectively. Figure 2.22 shows the $\hat{\theta}$ variation with sample number for R-OAR-NM (dc $\neq$ 0).

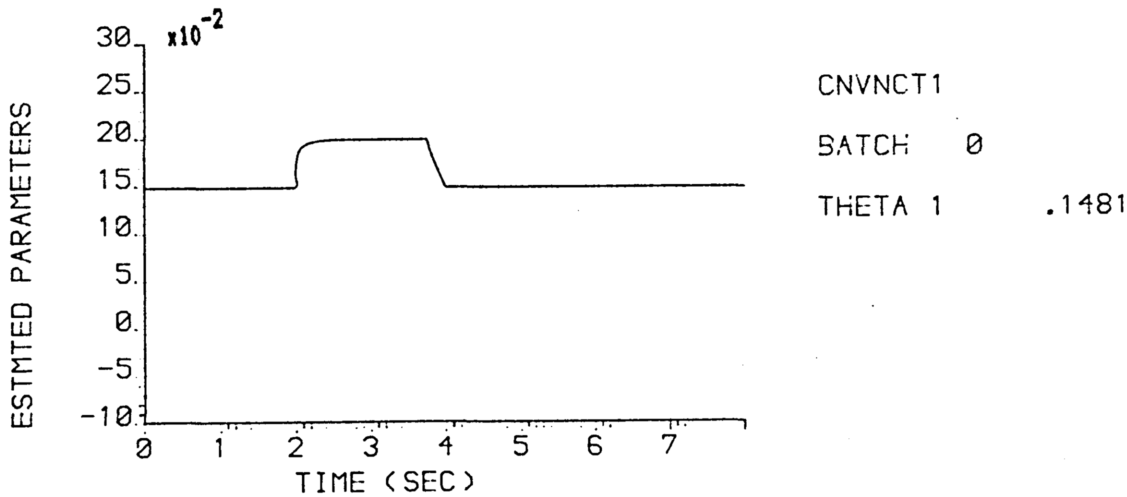

Figure 2.22

Variation of $\hat{\theta}$ with sample number for R-OAR-NM (dc $\neq$ 0)

It can be seen that it has a similar form to its d.c. level-removed counterpart (Figure 2.20) but that, apart from the regions of change, it remains at one or other of two constant values (either 0.2 or 0.1481). The plot of $\hat{\theta}$ variation for R-OAR-WM (dc $\neq$ 0) was the same as for the d.c. = 0 case of Figure 2.21.

Thus it is to be concluded that satisfactory OAR requires response modelling whether or not d.c. levels are removed.

2.5.1.3 CONTAMINATION OF THE EOG BY THE RESPONSE

Contamination of the EOGs by background EEG or by the response (IFEACHOR et al, 1986a; JERVIS et al, 1988) is another problem found in practice. This causes partial correlation between the EOGs and the measured EEG which leads to incorrect estimates of the θ_js and hence of the true (background) EEG and the responses. Thus the term $x_j(i)$ in equation (2.13) is replaced by $x_j'(i)$ where:

$$x_j'(i) = x_j(i) + K_{j1}EEG_t(i) + K_{j2}CNV(i) \quad -(2.31)$$

where K_{j1} and K_{j2} are transmission coefficients indicating contamination of the EOG by the true EEG and response respectively. This situation was investigated by simulation in which, as a simplification, K_{j1} was set to zero. A value of $K_{j2} = 0.2$ was introduced. To investigate the conflicting results of Sections 2.5.1.1 and 2.5.1.2 regarding d.c. level removal offsets of +10μV and -20μV were introduced into the simulated measured EEG and EOG respectively.

The simulated measured EEG and EOG are given in Figures 2.23 and 2.24 respectively.

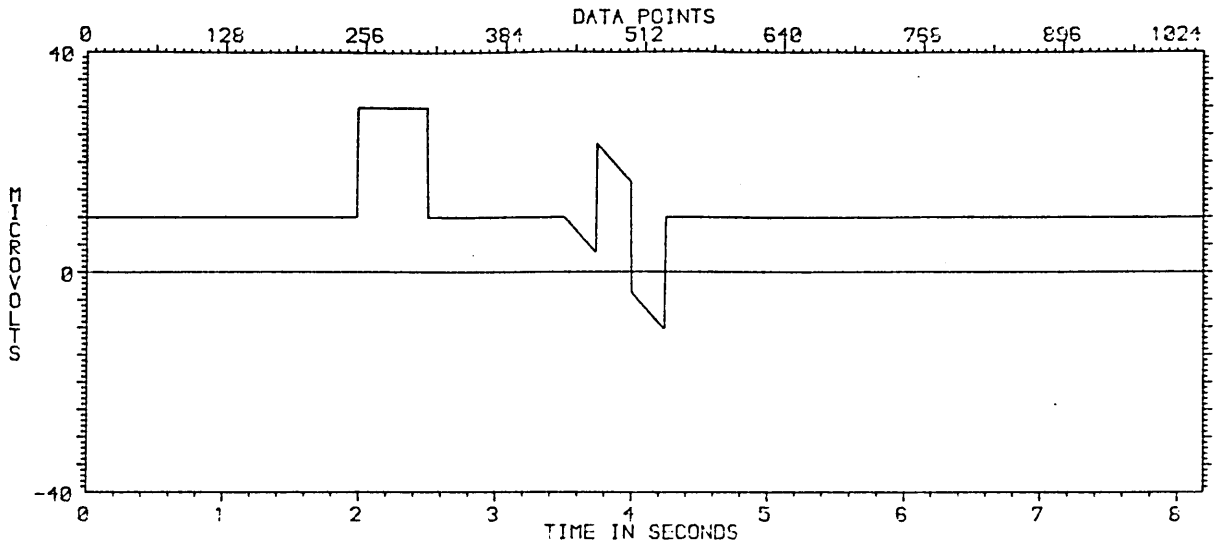

Figure 2.23

Simulated EEG which contains an OA and an OA superimposed on a response, possesses an offset and in which dc ≠ 0

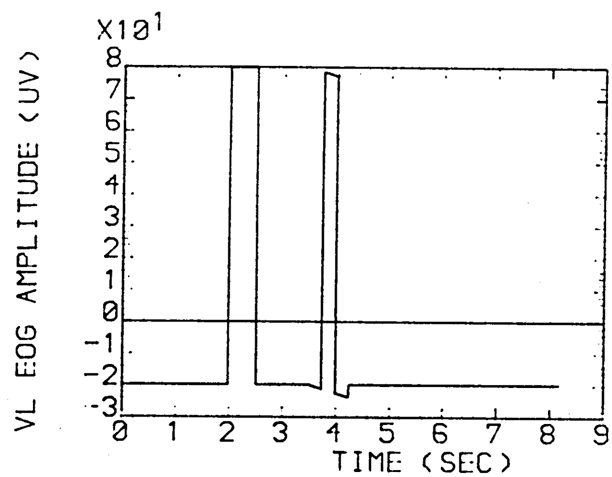

Figure 2.24

Simulated EOG corresponding to the OAs of Figure 2.23, which is contaminated by the responses, possesses an offset and in which dc ≠ 0

They both include d.c. levels and show the offsets introduced. The EEG was corrected for all four OAR methods, i.e., non-recursive/recursive and with/without response modelling, in which d.c. levels were removed. The resultant corrected EEGs were compared with the corresponding waveforms of Section 2.5.1.2 (i.e., in which no contamination of the EOGs occurred). Each pair of waveforms were very similar, i.e., the sequence of correction methods NR-OAR-NM, NR-OAR-WM, R-OAR-NM and R-OAR-WM yielded the same waveforms as these shown in Figures 2.17, 2.8, 2.18 and 2.19 respectively. The values of $\hat{\theta}$ obtained for NR-OAR-NM and NR-OAR-WM were 0.17665 (11.7% in error) and 0.19869 (0.7% in error). The corresponding estimates of θ for no contamination of the EOG (Section 2.5.1.2) were 0.17389 and 0.19863, i.e., differences of $<$ 1.6% (for NR-OAR-NM) and $<$ 0.04% (for NR-OAR-WM) between contaminated and uncontaminated EOGs. The $\hat{\theta}$ variations for R-OAR-NM and R-OAR-WM are shown in Figures 2.25 and 2.26.

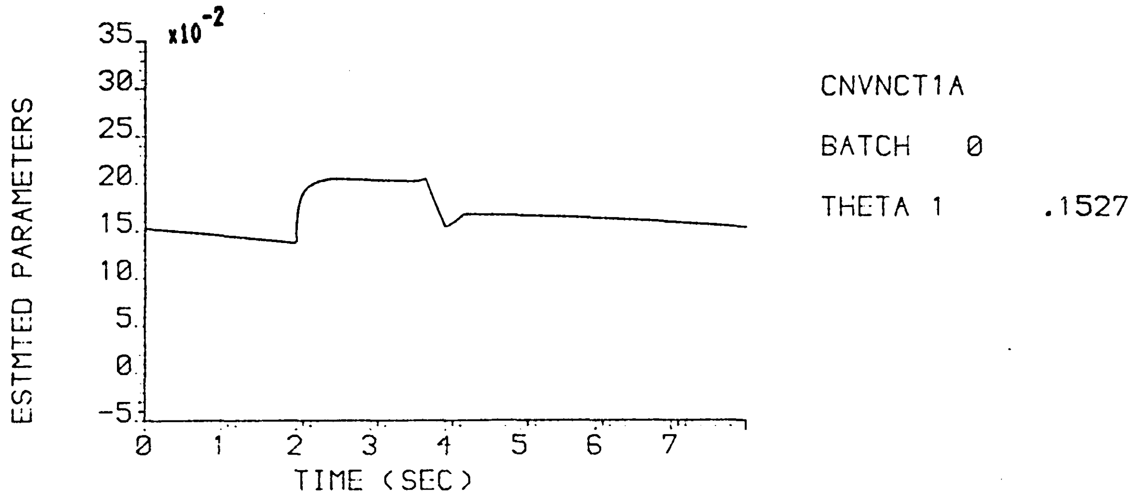

Figure 2.25

Variations of $\hat{\theta}$ with sample number for R-OAR-NM (dc = 0)

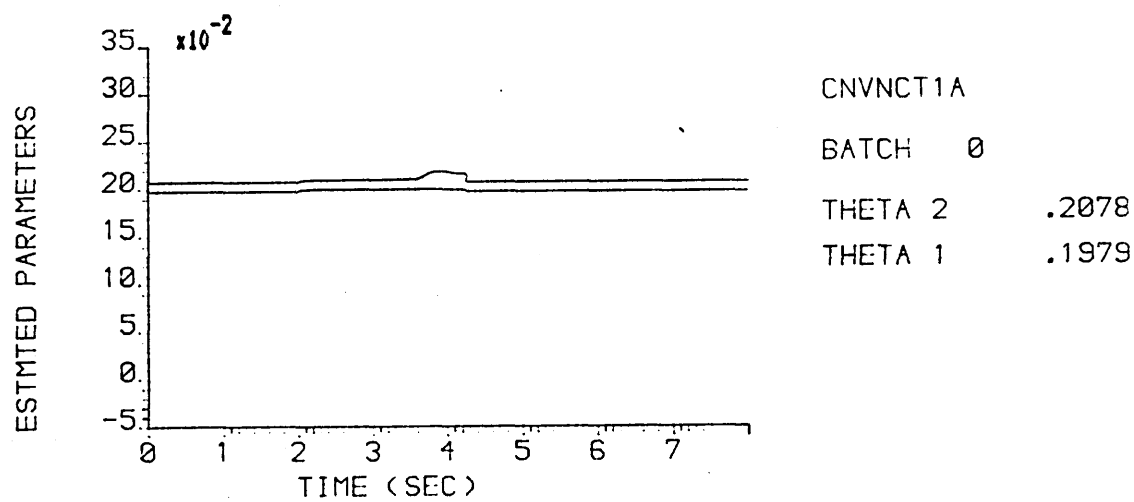

Figure 2.26

Variation of $\hat{\theta}$ with sample number for R-OAR-WM (dc = 0)

Apart from small differences in values the general form of both figures is as for the no EOG contamination case, i.e., Figures 2.20 and 2.21 respectively.

The conclusion for d.c. level removed data is that modelling is necessary for both NR-OAR and R-OAR cases. Furthermore, for the test data used here, the effect of contamination of the EOG by the response is very small.

When d.c. levels remain, use of NR-OAR-NM and NR-OAR-WM result in the corrected EEGs in Figures 2.27 and 2.28.

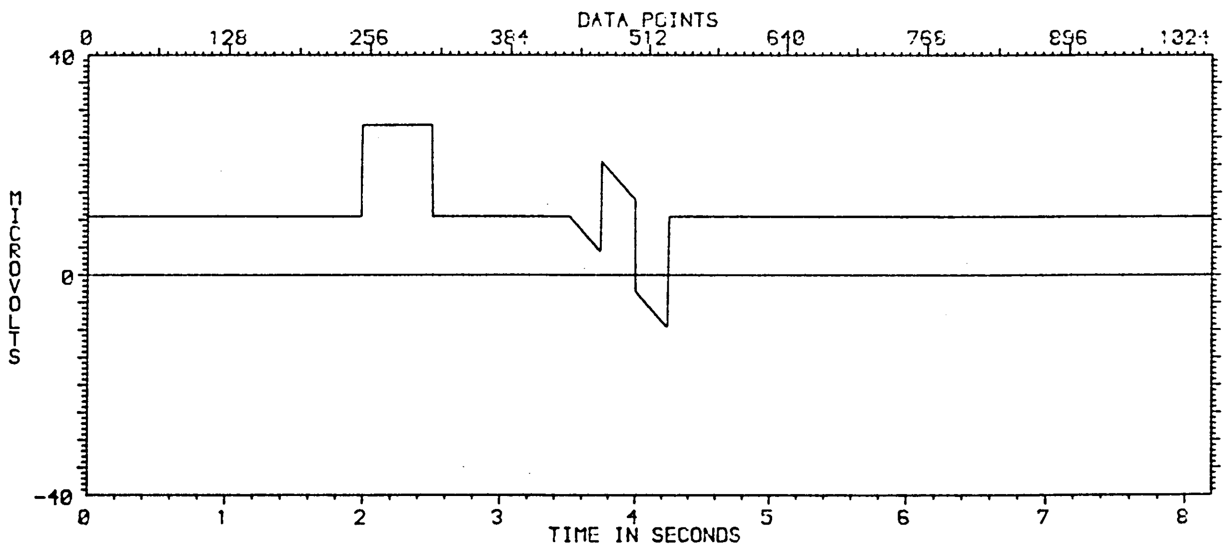

Figure 2.27

The NR-OAR-NM (dc $\neq$ 0) corrected EEG

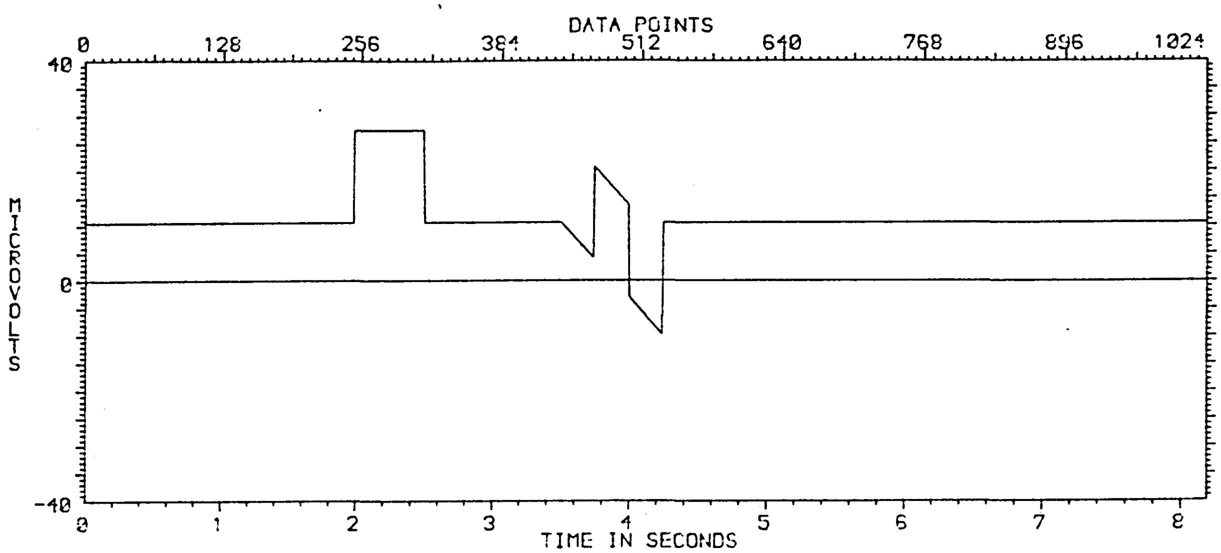

Figure 2.28

The NR-OAR-WM (dc $\neq$ 0) corrected EEG

It is immediately apparent that little OA has been removed. The $\hat{\theta}$ values obtained with and without response modelling were 0.03180 and 0.03391 respectively, i.e., errors of 84% and 83%. Artefacts of 16.6μV and 16.8μV remain (out of original OAs of 20 μV) for NR-OAR-NM and NR-OAR-WM cases.

Figure 2.29 shows the R-OAR-NM (dc $\neq$0) corrected EEG.

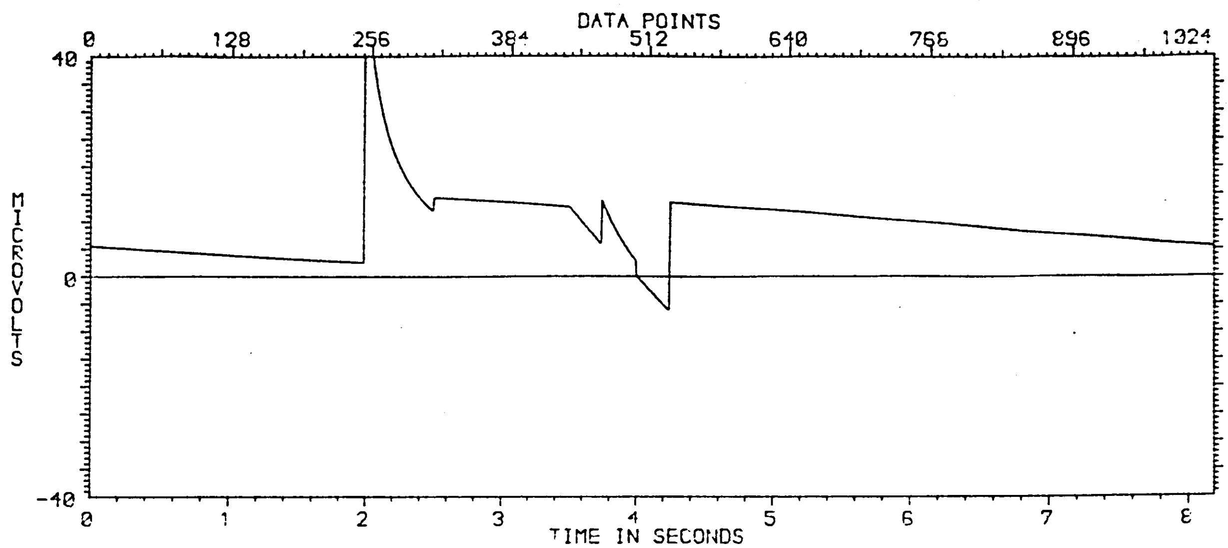

Figure 2.29

The R-OAR-NM (dc $\neq$ 0) corrected EEG

Substantial EEG baseline distortion has occurred, considerable OA remnants are evident and the CNV has been distorted. Figure 2.30 is the corrected EEG when R-OAR-WM (dc $\neq$0) is used.

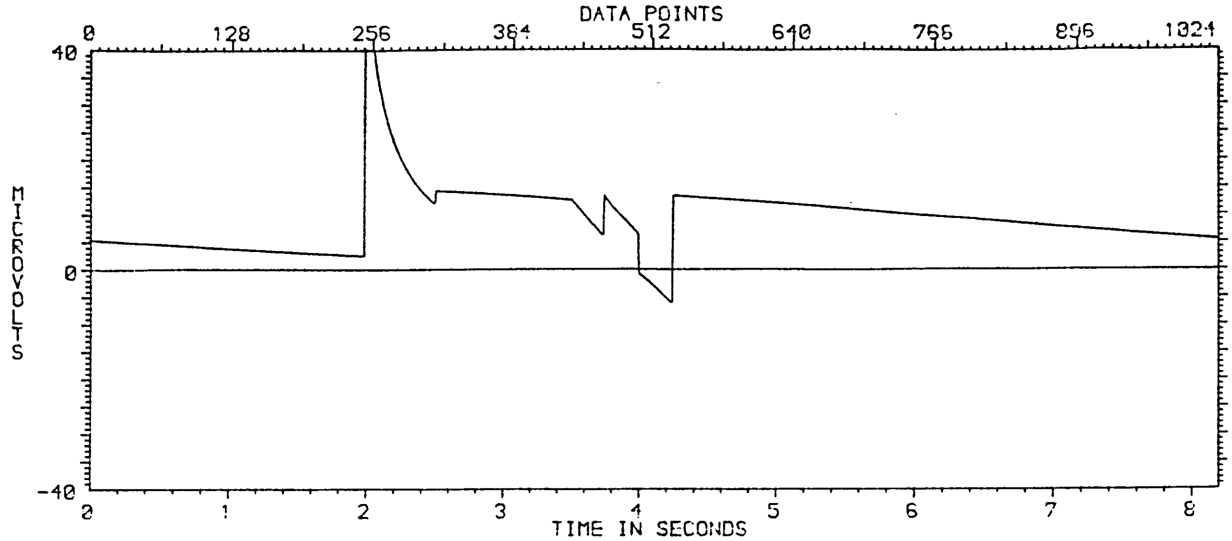

Figure 2.30

The R-OAR-WM (dc $\neq$ 0) corrected EEG

This is marginally worse than the case without response modelling in that the response OA remnant is ~4μV greater (at OA termination) than when R-OAR-NM (dc $\neq$0) is used. Figures 2.31 and 2.32 show wildly varying $\hat{\theta}$s for both R-OAR-NM (dc $\neq$0) and R-OAR-WM (dc $\neq$0).

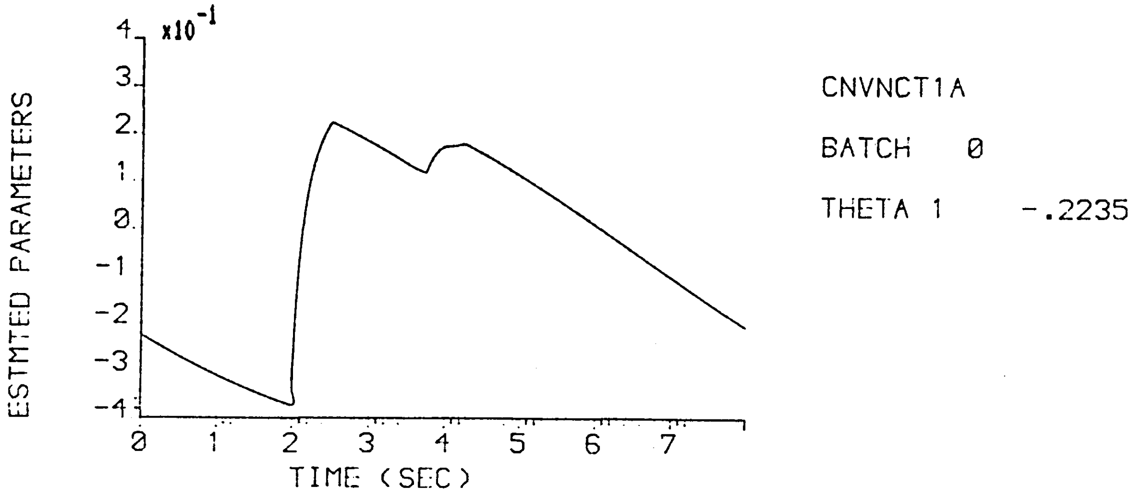

Figure 2.31

Variation of $\hat{\theta}$ with sample number for R-OAR-NM (dc $\neq$ 0)

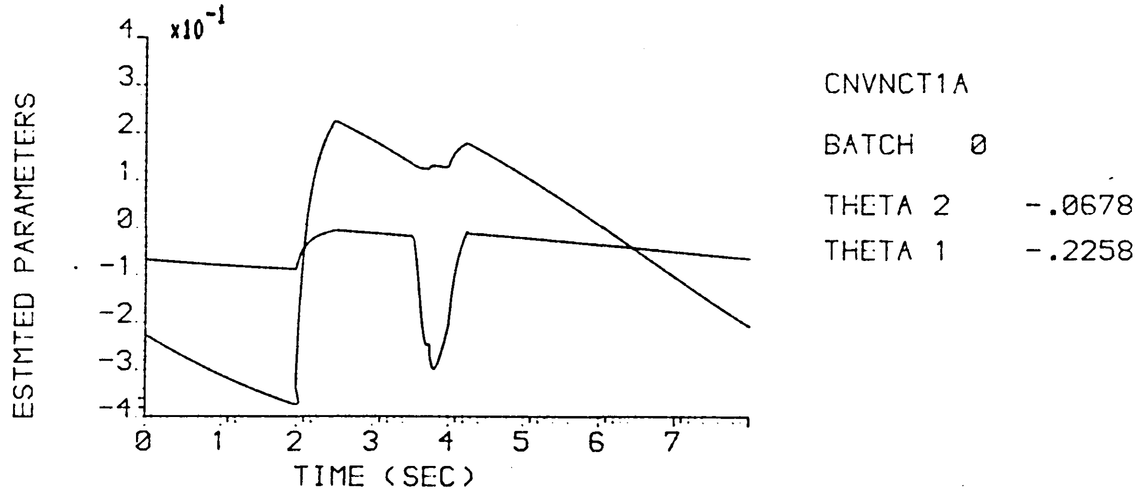

Figure 2.32

Variation of $\hat{\theta}$ with sample number for R-OAR-WM (dc $\neq$ 0)

On the basis of the above observations on this test data it is to be concluded that when offsets are present d.c. level removal is essential. In addition it is impossible to discern any effects due to response contamination of the EOG. It is assumed that any such effect is of the same (small) magnitude as for the above cases in which the d.c. level was removed and so it is obvious that effects on the corrected waveforms due to d.c. offsets will be far greater. Thus it is to be concluded, for the given test data, that EOG contamination by the response can be neglected.

2.5.1.4 REALISTIC TEST DATA

Here the more realistic case in which the OA has been modelled in terms of several EOGs is considered. It was also decided to simulate the experimental paradigm used in obtaining the CNVs investigated in this work (see Section 1.3 for recording details) by introducing simulated AEPs at each end of the CNV. The effect of 'mismodelling' the CNV by making the model and actual CNV shapes somewhat different was also to be investigated. This was done by assuming a two component CNV model. The 'true' CNV, however, had the form of Figure 2.33, i.e., a very short negative going

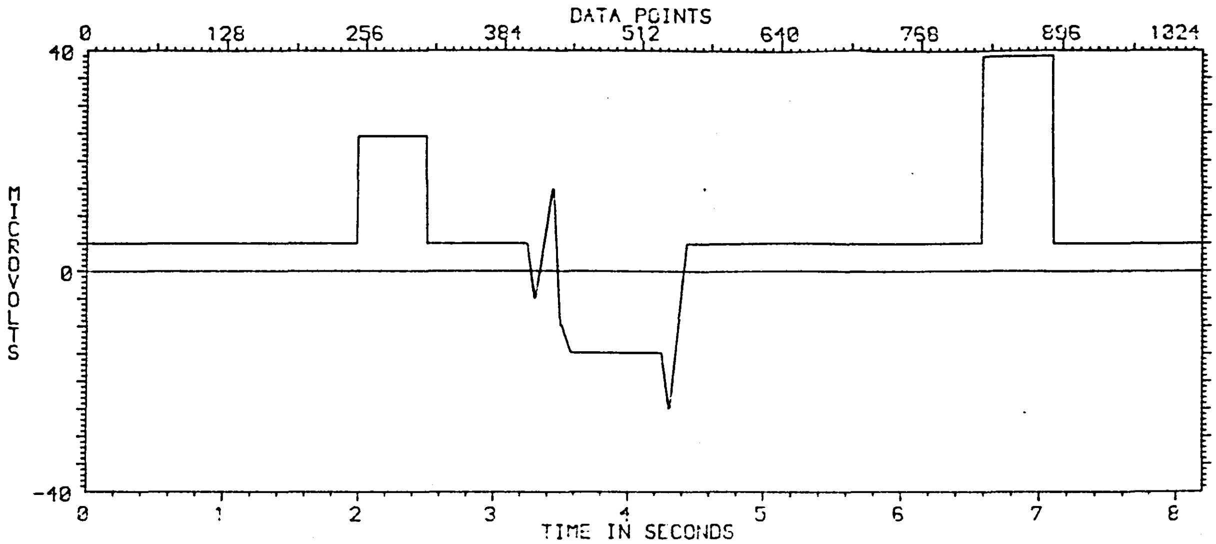

Figure 2.33

Simulated measured 'realistic' EEG, with offset and d.c. level

segment after the first AEP followed by a constant level until the second AEP, and as such was intended to bear a degree of resemblance to the real CNV of Figure 2.1. In addition d.c. offsets of +5μV for the EEG channel and -10μV for each EOG channel were included. It was decided to vary the known θ values between EOG channels and between OAs. This was to allow for differing θs due to different transmission paths and also to allow for variation of θ within a channel due to, say, electrode displacement during recording and the fact that differing OAs (e.g., blinks and eye-movements (EMs)) require different amounts of compensation (WEERTS and LANG, 1973 and IFEACHOR et al, 1986a). This assumed data is summarised in Table 2.1.

TABLE 2.1

EOG AND EEG DATA USED IN SECTION 2.5.1.4

EOG channel	θ value		Measured EOG (uV)		Contribution to EEG (μV)	
	OA_1	OA_2	OA_1	OA_2	OA_1	OA_2
VL	0.10	0.05	350	500	35	25
VR	-0.01	0.10	280	400	-2.8	40
HL	-0.10	-0.20	56	80	-5.6	-16
HR	-0.10	-0.15	70	100	-7	-15

34μV. The two OAs in the EEG have magnitudes of 19.6μV and 34μV. The negative values of the θs are included in Table 2.1 since the simulated EMs in the EOGs of Figure 2.34 are all positive going whereas negative going EMs also occur in real data (a positive EM with a negative value of θ transmission coefficient has the same contribution to OA in the EEG as a negative EM with a positive value θ).

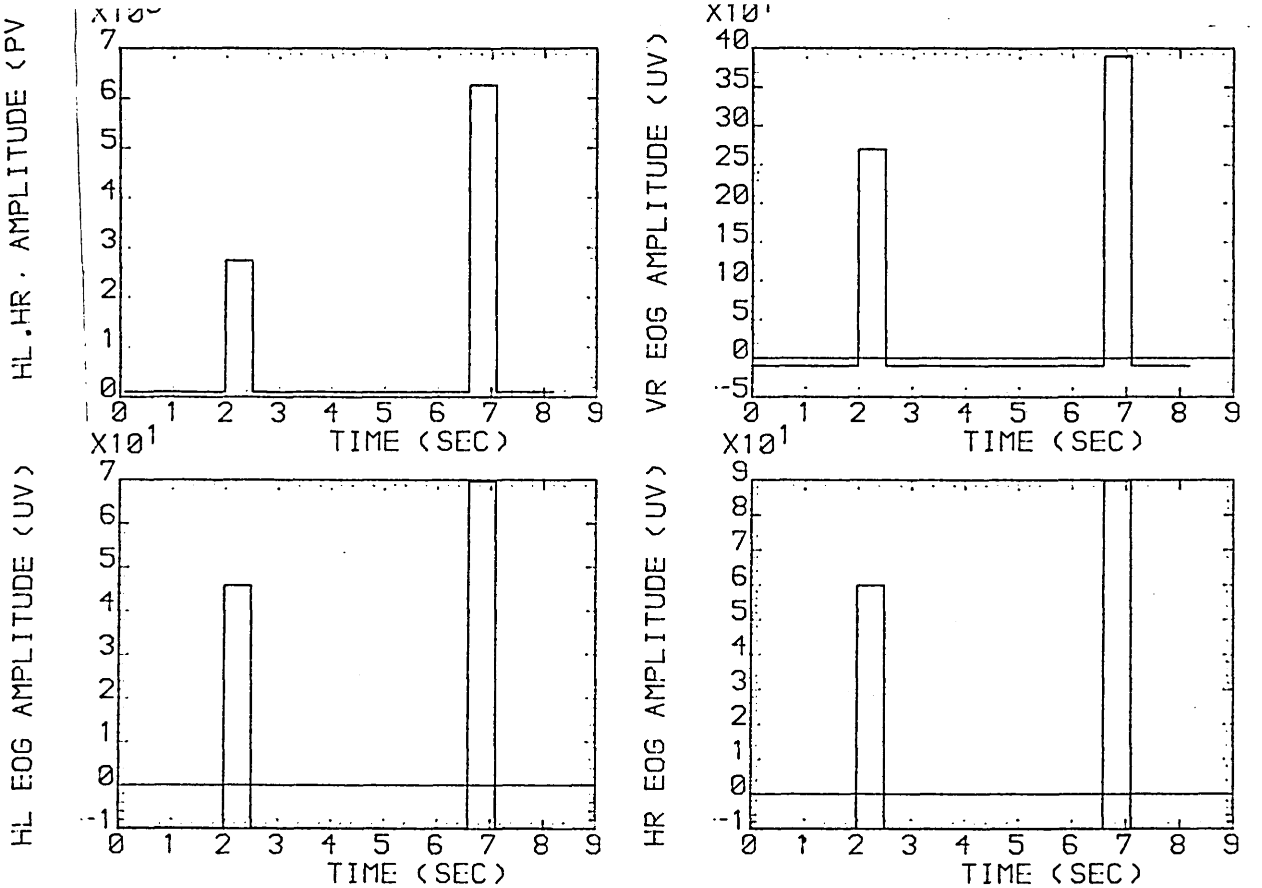

Figure 2.34

Simulated EOGs corresponding to OAs of Figure 2.33 with offset and d.c. level

The two AEPs were not modelled since the major part of the error in estimating the θ values is due to the presence of the CNV (insofar as the test data used here is concerned). This arises since the calculation of $\hat{\theta}$ will be affected more adversely for non-random waveforms possessing larger amplitudes and/or longer duration (the CNV) than for smaller amplitude and/or shorter duration responses (the AEPs). This is a consequence of the least squares method of estimating the θs. Thus even though the second AEP in Figure 2.33 has a larger maximum amplitude than the CNV the latter's greater duration will outweigh the AEP's influence.

The first part of the following discussion concerns data which have had their d.c. levels removed and non-removal of d.c. levels is discussed after that.

The simulated measured EEG and EOGs are given in Figures 2.33 and 2.34 and show the respective offsets. The values of $\hat{\theta}$ obtained in using NR-OAR are shown in Table 2.2.

TABLE 2.2

VALUES OF $\hat{\theta}$s OBTAINED IN SECTION 2.5.1.4

	NR-OAR-NM		NR-OAR-WM	
	dc = 0	dc = 0	dc = 0	dc = 0
$\hat{\theta}_1$	466.78442	1968.21118	533.05396	2298.06421
$\hat{\theta}_2$	- 0.16260	- 0.01134	- 0.17977	0.05383
$\hat{\theta}_3$	1.32366	- 2.46946	1.42793	- 2.13716
$\hat{\theta}_4$	- 0.08000	2.26075	- 0.11716	1.69729
$\hat{\theta}_5$	--	--	0.53200	0.53371
$\hat{\theta}_6$	--	--	- 0.16386	- 0.16441

The estimated values of $\hat{\theta}$ of this table are related to the EOGs of Table 2.1 by the form of equation (2.7), i.e., θ_1 is associated with the term $HEOG_L.HEOG_R$, $\hat{\theta}_2$ with $VEOG_R$ etc. Each of these θ estimates depends upon a corresponding pair of known θs in Table 2.1 (one for each OA). This is because the NR-OAR method uses all the EOG (and EEG) data in a record to compute a single estimate of each θ. It is seen that the estimated values of θ (Table 2.2) bear no obvious relation to the known values in Table 2.1. The values of $\hat{\theta}_1$ are estimates of the transmission coefficient associated with the $HEOG_L.HEOG_R$ term and are relatively large. However the product $\Theta_1 HEOG_L . HEOG_R$ is relatively small because $HEOG_L . HEOG_R$ is small.

Figure 2.35 gives the NR-OAR-NM (dc = 0) corrected EEG which shows overcorrection of both OAs (~4.5μV for OA_1 and ~1.5μV for OA_2).

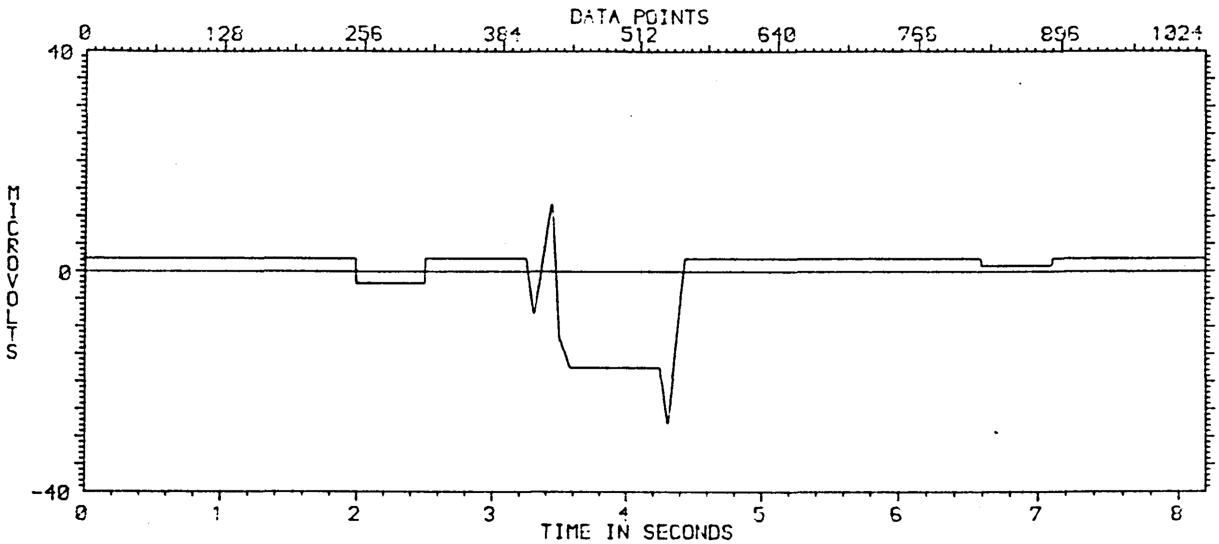

Figure 2.35

The NR-OAR-NM (dc = 0) corrected EEG

Figure 2.36 gives the NR-OAR-WM (dc = 0) corrected EEG showing improved OA correction, there being remnants of ~3μV for OA_1 (overcorrected) and ~0.5μV for OA_2 (undercorrected).

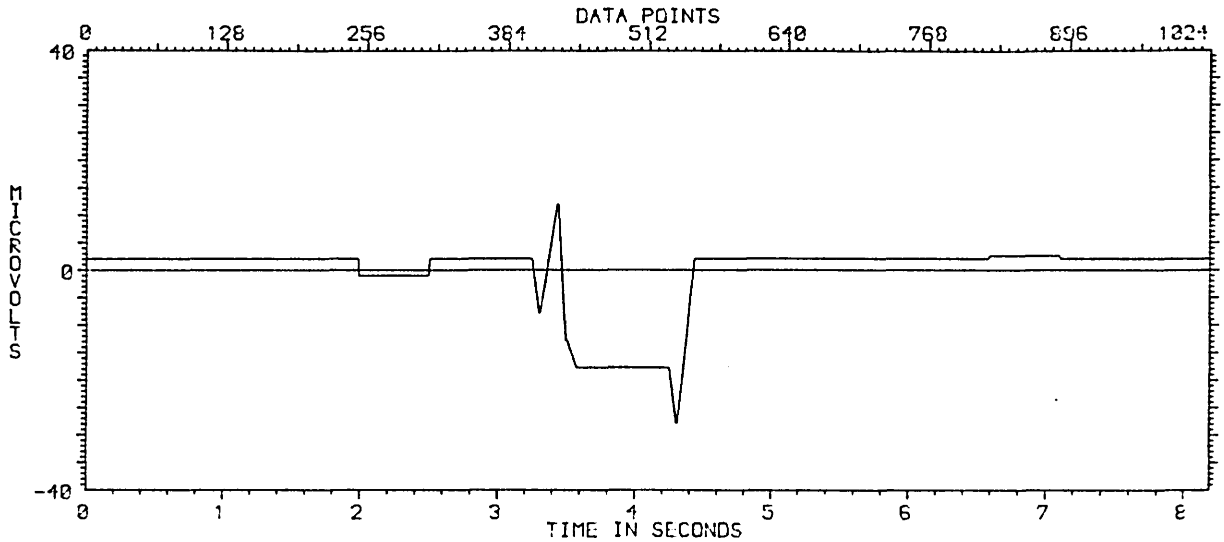

Figure 2.36

The NR-OAR-WM (dc = 0) corrected EEG

Figure 2.37 gives the R-OAR-NM (dc = 0) corrected EEG which shows large CNV and post-CNV EEG baseline distortion.

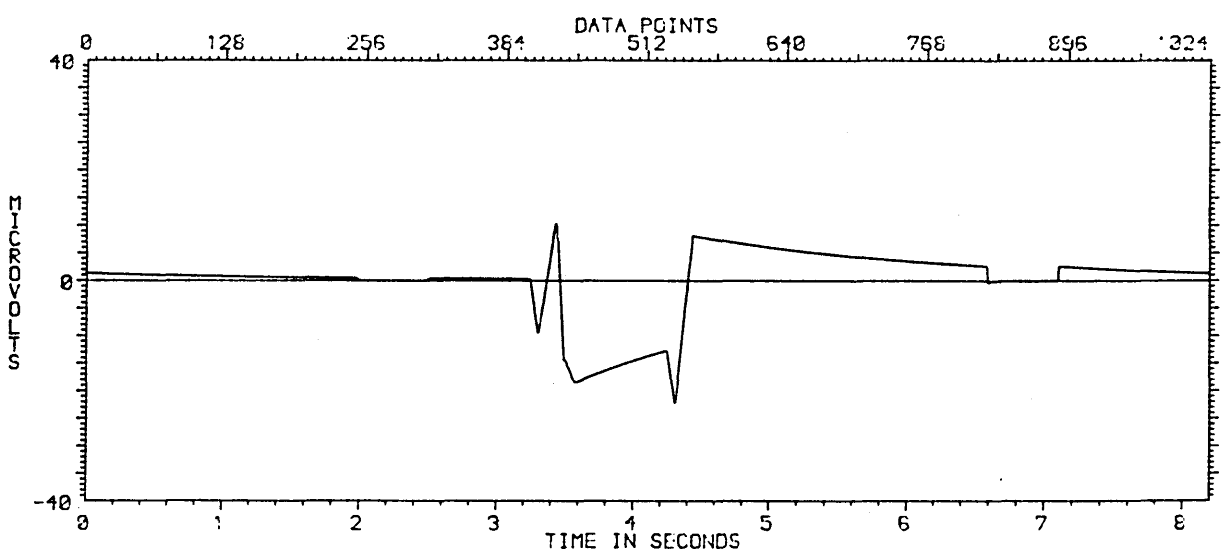

Figure 2.37

The R-OAR-NM (dc = 0) corrected EEG

The CNV has a $+6\mu V$ shift just before AEP_2, and the EEG baseline suffers a $+7\mu V$ shift just after AEP_2 (both shifts are relative to the corresponding known value of CNV or EEG). The OA remnants are $\tilde{}0.5\mu V$ for OA_1 and $\tilde{}3\mu V$ for OA_2 (both overcorrected). It is to be noted that the above values of distortion of CNV and EEG are comparable to those observed in the real data when Figures 2.1 and 2.3 are compared. Indeed the overall evolution of both R-OAR-NM waveforms when compared to their respective no-OAR EEGs are similar; an increasingly positive shift of the CNV during its development and a positive (but decreasing with time) shift of the post AEP_2 EEG baseline. The R-OAR-WM (dc = 0) corrected EEG is shown in Figure 2.38.

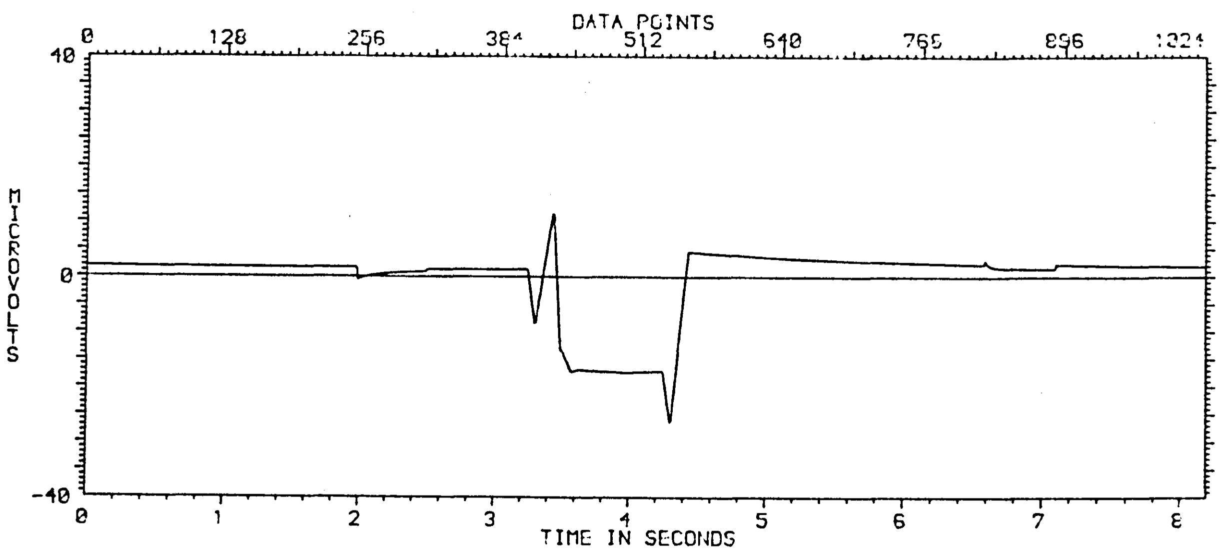

Figure 2.38

The R-OAR-WM (dc = 0) corrected EEG

The post AEP_2 EEG baseline distortion is reduced to a $+3.5\mu V$ maximum shift. Much of the CNV distortion has been eliminated but there still exists $+1\mu V$ shift prior to AEP_2.

The variations of $\hat{\theta}$ for R-OAR-NM (dc = 0) and R-OAR-WM (dc =0) are shown in Figures 2.39 and 2.40.

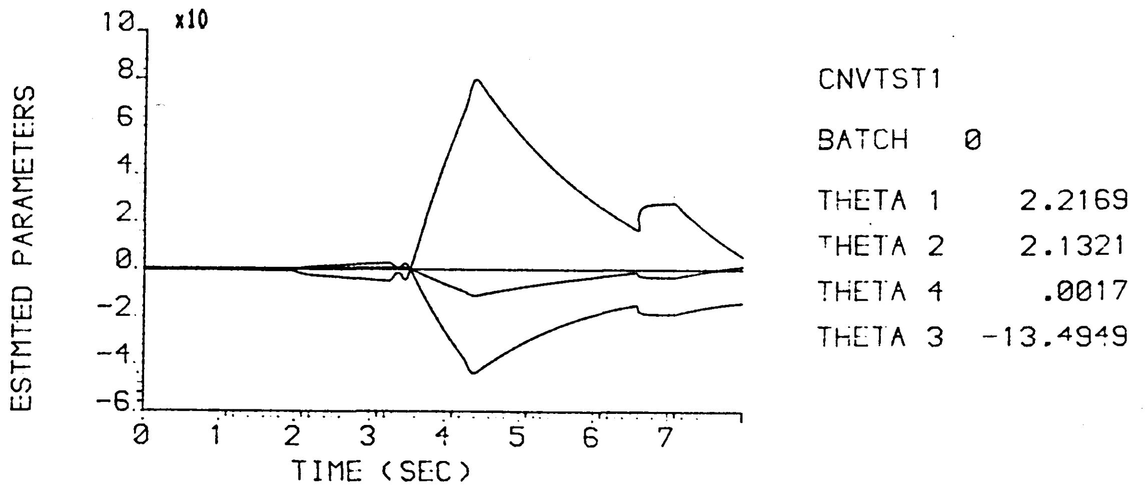

Figure 2.39

Variation of $\hat{\theta}$ with sample number for R-OAR-NM (dc = 0)

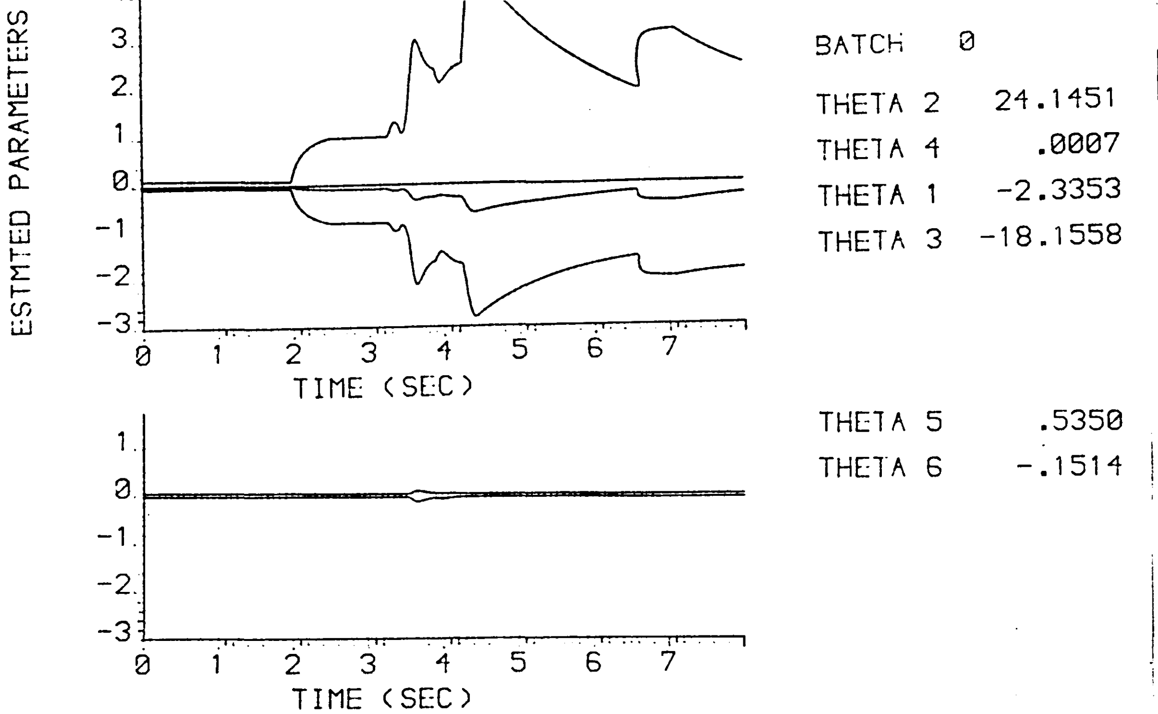

Figure 2.40

Variation of $\hat{\theta}$ with sample number for R-OAR-WM (dc = 0)

It is seen that rapid changes in $\hat{\theta}$ value start during AEP_1 and AEP_2 in both cases. However, the range of $\hat{\theta}$ values ((maximum $\hat{\theta}$ value) - (minimum $\hat{\theta}$ value)) is smaller when the reponse is modelled, i.e., ~78 as compared to ~123 without modelling.

Thus for the d.c. level removed results given above it is concluded that for both NR-OAR and R-OAR modelling of the response gives better corrected waveforms than when modelling is not performed.

In the following paragraphs the effects of non-removal of d.c. levels are considered. Figure 2.41 shows the NR-OAR-NM (dc $\neq 0$) corrected EEG.

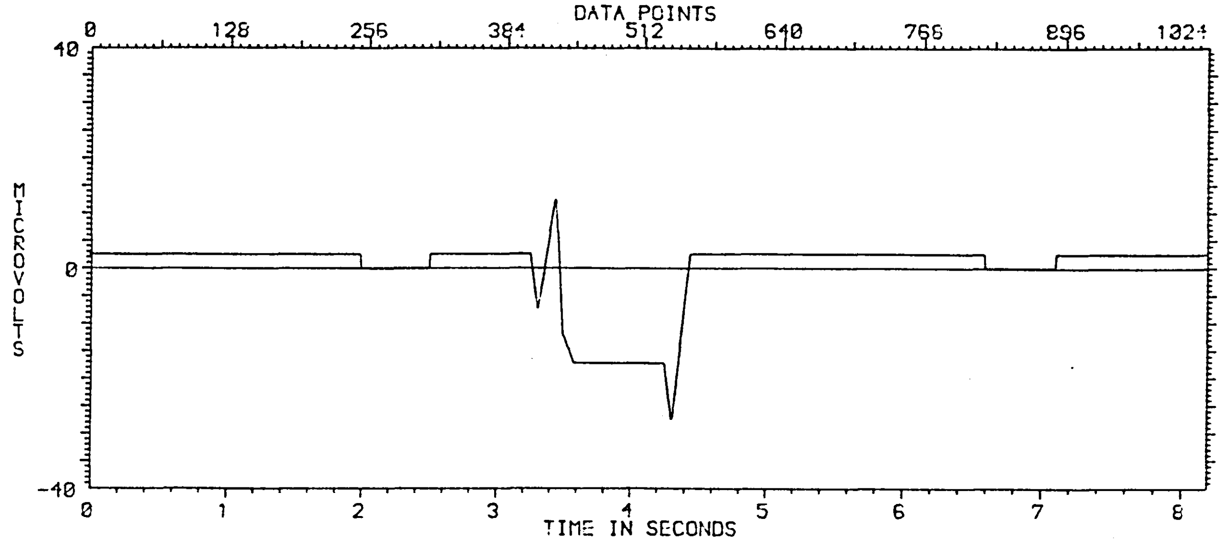

Figure 2.41

The NR-OAR-NM (dc $\neq$ 0) corrected EEG

The remnant OA (for both OAs) is ~2.5µV (overcorrected). This value is reduced to ~1µV (overcorrected) when NR-OAR-WM (dc $\neq$0) is applied as shown in Figure 2.42.

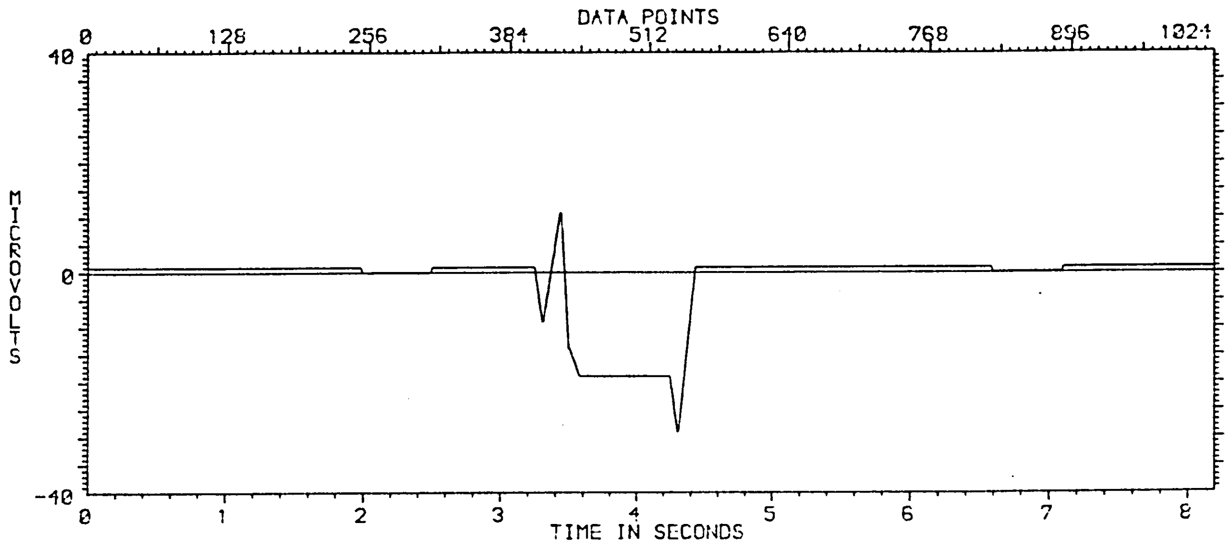

Figure 2.42

The NR-OAR-WM (dc $\neq$ 0) corrected EEG

Comparing these two figures with those in which d.c. levels have been removed (Figures 2.35 and 2.36) it appears that not performing d.c. level removal gives marginally better correction. However, in light of the previous results in which the non-removal of d.c. levels led to much greater differences it is felt to be an insignificant result. This assertion can be supported by noting that the offsets introduced in this section are smaller in relation to the data than these of Section 2.5.1.3 and hence will have less effect on the corrected waveforms.

Figures 2.43 and 2.44 show the R-OAR-NM (dc $\neq$ 0) and R-OAR-WM (dc $\neq$ 0) corrected EEGs which are very similar to the corresponding d.c. level removed cases.

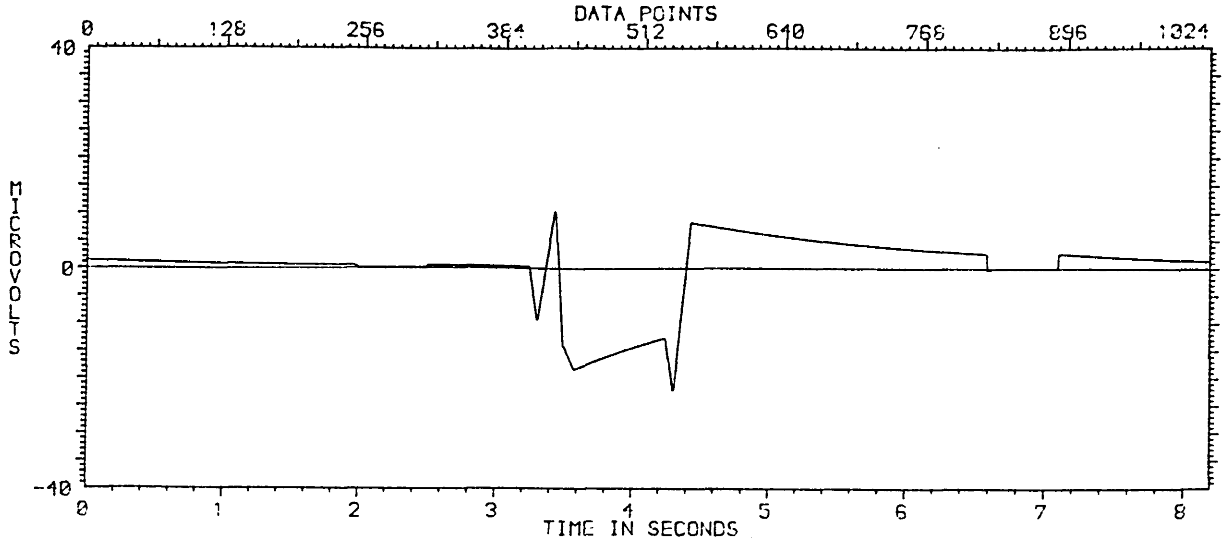

Figure 2.43

The R-OAR-NM (dc ≠ 0) corrected EEG

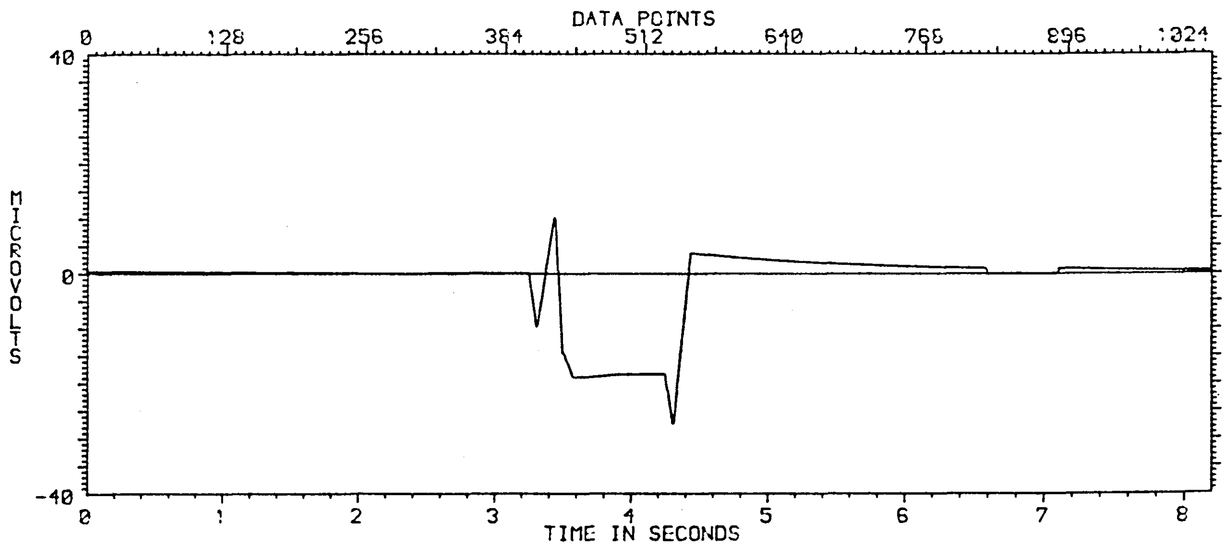

Figure 2.44

The R-OAR-WM (dc ≠ 0) corrected EEG

In the latter the CNV distortion is of a slightly different form but is the same in magnitude.

The plots of $\hat{\theta}$ variation for R-OAR-NM (dc $\neq$ 0) and R-OAR-WM (dc $\neq$0) are shown in Figures 2.45 and 2.46.

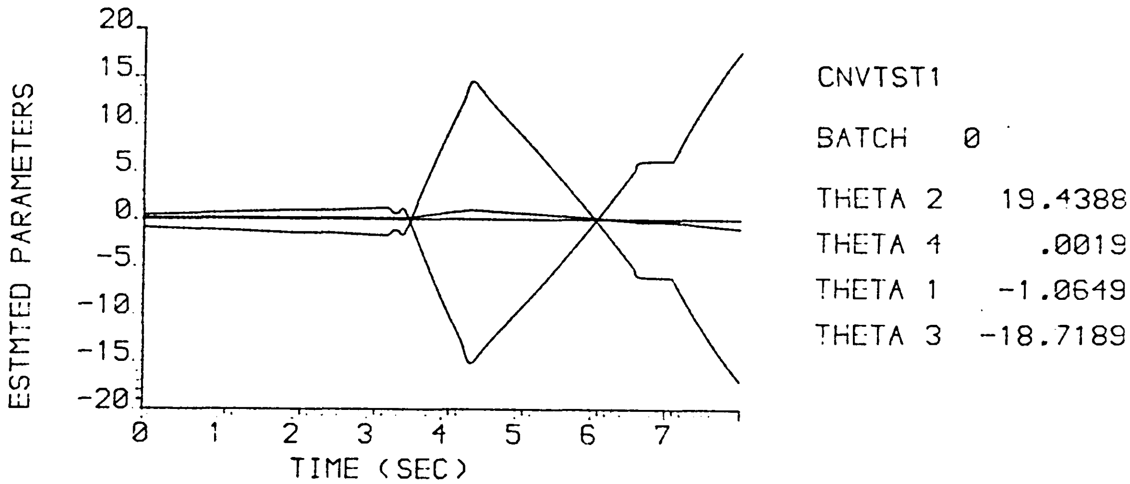

Figure 2.45

Variation of $\hat{\theta}$s with sample number for R-OAR-NM (dc $\neq$ 0)

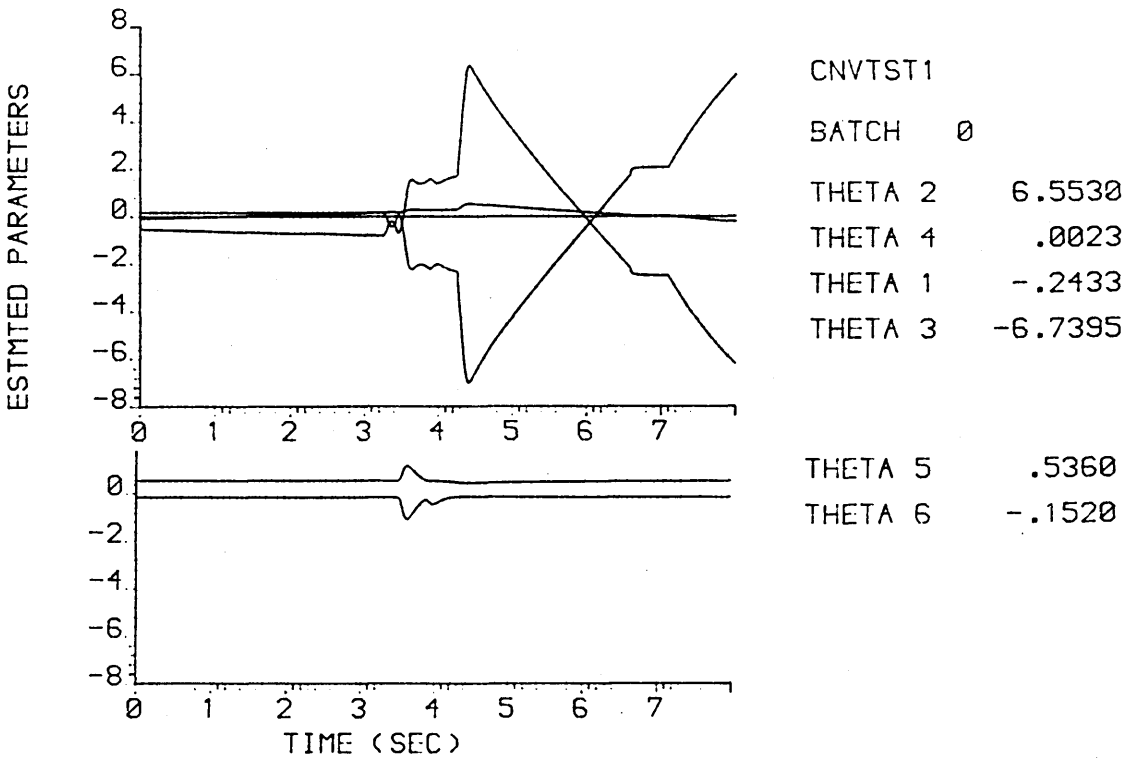

Figure 2.46

Variation of $\hat{\theta}$s with sample number for R-OAR-WM (dc $\neq$ 0)

Both of them show reduced value ranges as compared with their d.c. level removed counterparts and once more the range of values is smaller in the response modelled case.

The results depicted in Figures 2.41 - 2.46 reinforce the view that modelling the response leads to better corrected waveforms than when modelling is not performed.

Finally the question of what the effects are (if any) of mismodelling of the CNV and/or non-modelling of the AEP are examined. Scrutiny of the CNVs of both NR-OAR-WM cases (i.e., dc = 0 and dc ≠ 0 in Figures 2.36 and 2.42 respectively) revealed no noteable differences when compared to the no-OAR case (Figure 2.33). However comparison of Figures 2.38 and 2.44 (R-OAR-WM (dc = 0) and R-OAR-WM (dc ≠ 0)) both showed some differences in CNV shape when compared to Figure 2.33. The source of this could be either the CNV mismodelling or the lack of modelling of the AEPs. Comparison of the AEPs in Figures 2.36, 2.38, 2.42 and 2.44 with those of 2.33 showed no discernible differences and thus it is felt that the CNV shape modification is due to the response mismodelling.

The results of this section dealing with realistic data lead to the following conclusions:

(i) In contrast to the results of the previous test data set (section 2.5.2.3) non-removal of d.c. level does not cause severe errors. However neither does it bring about any significant improvement in results.

(ii) NR-OAR is improved, in terms of OA correction, by the use of modelling.

(iii) R-OAR must be done with modelling to reduce both CNV and EEG baseline distortion and to improve OA correction.

(iv) Mismodelling of the CNV can lead to small amounts of distortion of the CNV in the corrected EEG for R-OAR .

2.6 RESULTS USING REAL DATA

The effectiveness of incorporating modelling in the OAR procedure applied to experimental response data was investigated. Both CNV and AEP responses were used, which had been recorded in earlier work (NICHOLS, 1982). In light of the previous conclusions concerning d.c. level removal, it was decided to carry out the processing with and without d.c. level removal.

2.6.1 CNV RESPONSES

The CNV was modelled by two components (as described in Section 2.5.1.4. An ISI of 1 second was used in this investigation. The AEPs were not modelled for the reasons given in Section 2.5.1.4. Figure 2.47 shows the averaged 1 second ISI CNV (with d.c. level removed) of the same co-operative subject for whom results were given in Figures 2.1 - 2.3.

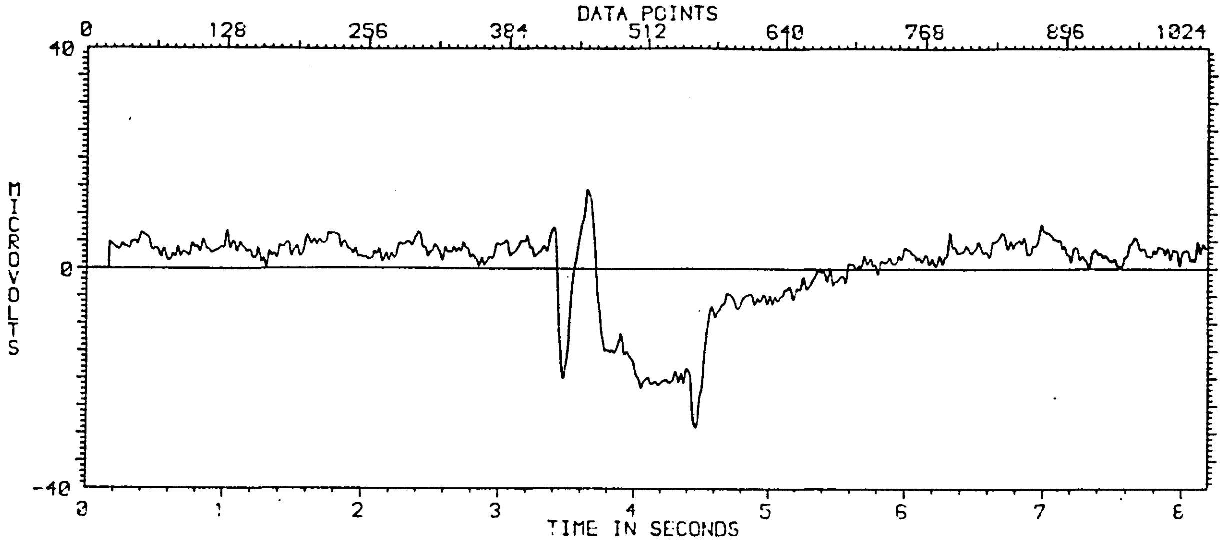

Figure 2.47

The averaged 1s ISI CNV of the subject of Fig. 2.1 subsequent to implementation of NR-OAR-WM (dc = 0)

The results of Figure 2.47 were obtained using the NR-OAR-WM method. Comparison of this averaged CNV with that of Figure 2.1 shows that NR-OAR-WM has had little effect on the CNV shape. The R-OAR-WM (dc = 0) corrected EEG is shown in Figure 2.48.

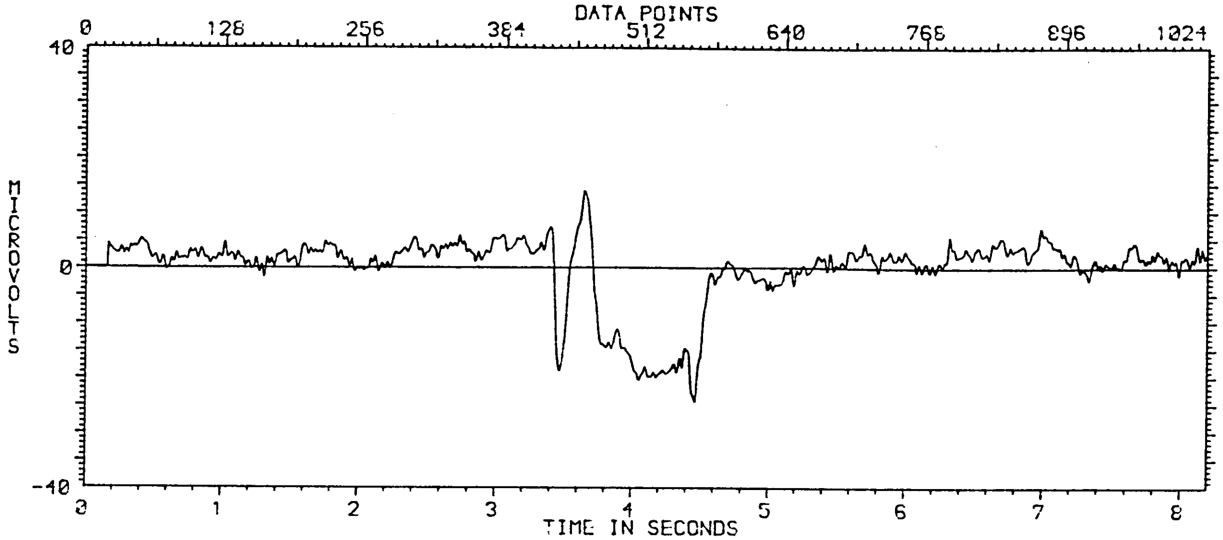

Figure 2.48

The averaged 1s ISI CNV of the subject of Fig. 2.1 subsequent to implementation of R-OAR-WM (dc = 0)

It is seen that the CNV shape modification is now reduced to a maximum of ~4μV as compared to ~7.5μV without modelling.

Figures 2.49 - 2.52 refer to OAR processing in which no d.c. level removal was performed. However, the plots of the corrected EEGs shown have had their d.c. levels removed subsequent to processing in order to bring them within the display area for plotting. Figures 2.49 and 2.50 show EEGs corrected with NR-OAR-NM (dc $\neq$ 0) and NR-OAR-WM (dc $\neq$ 0) between which no differences are observed.

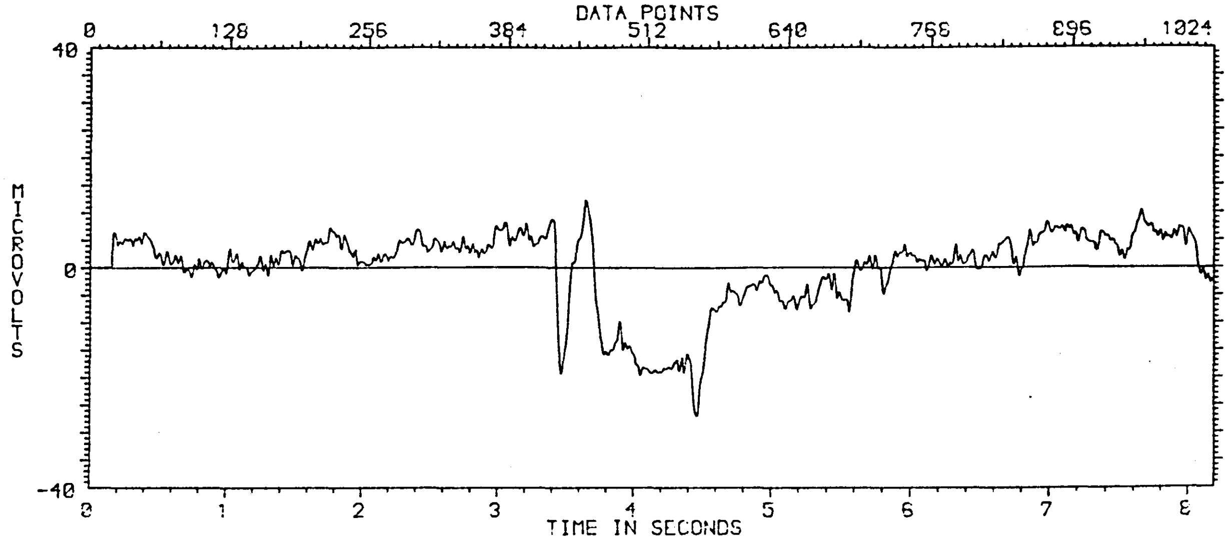

Figure 2.49

The averaged 1s ISI CNV of the subject of Fig. 2.1 subsequent to implementation of NR-OAR-NM (dc $\neq$ 0)

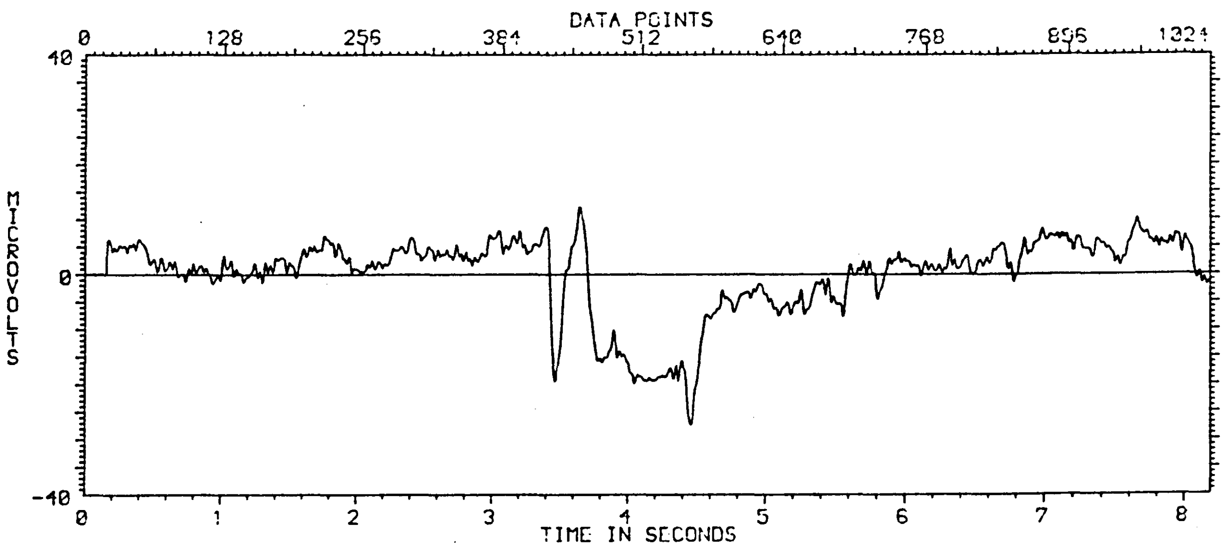

Figure 2.50

The averaged 1s ISI CNV of the subject of Fig. 2.1 subsequent to implementation of NR-OAR-WM (dc $\neq$ 0)

Comparing these figures with Figure 2.1 (the no-OAR case) there is an attenuation of the first AEP (by ~1-2μV) and the CNV shape has been modified, this being manifest by up to a 2μV positive shift of the mid- to latter part of the CNV. There are also various differences in the background EEG (between the NR-OAR and no-OAR cases) outside the CNV and AEP region. Figure 2.51 shows the results from R-OAR-NM.

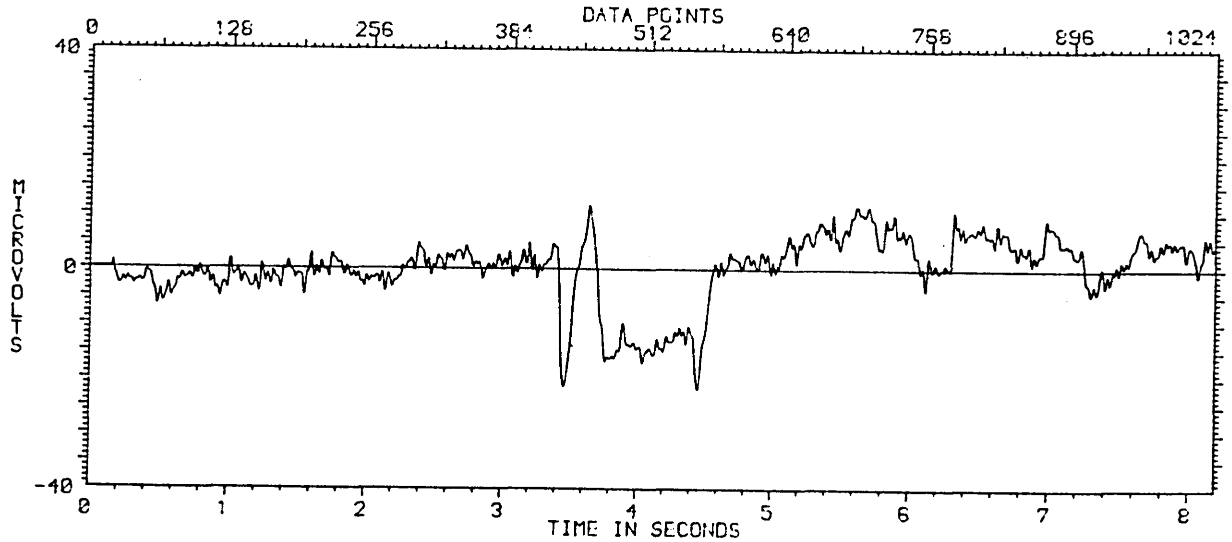

Figure 2.51

The averaged 1s ISI CNV of the subject of Fig. 2.1 subsequent to implementation of R-OAR-NM (dc $\neq$ 0)

Apart from miscellaneous background EEG differences there is an obvious CNV distortion in the form of a maximum +7.5μV shift in the latter part of the CNV with respect to the no-OAR case. Figure 2.52 was obtained with R-OAR-WM.

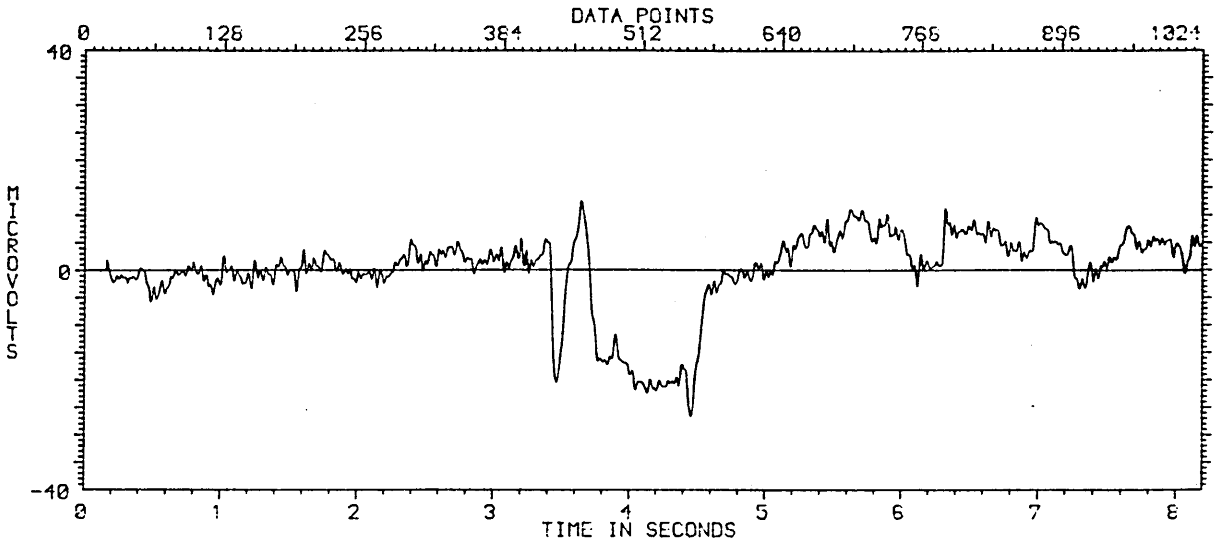

Figure 2.52

The averaged 1s ISI CNV of the subject of Fig 2.1 subsequent to implementation of R-OAR-WM (dc $\neq$ 0)

Although the previous CNV shape distortion is much reduced it is still ~2μV. Furthermore there still exist noticeable background EEG differences when comparing R-OAR cases with the no-OAR case.

The above results lead to the conclusion that whether or not d.c. levels are removed response modelling in NR-OAR produces no difference in the corrected EEG from that for the NR-OAR-NM case and can be omitted whereas response modelling should be used when R-OAR is performed in order to reduce CNV distortion.

2.6.2 AEP RESPONSES

A pair of 32 trial averaged AEPs are shown in Figure 2.53. These were obtained without any OAR.

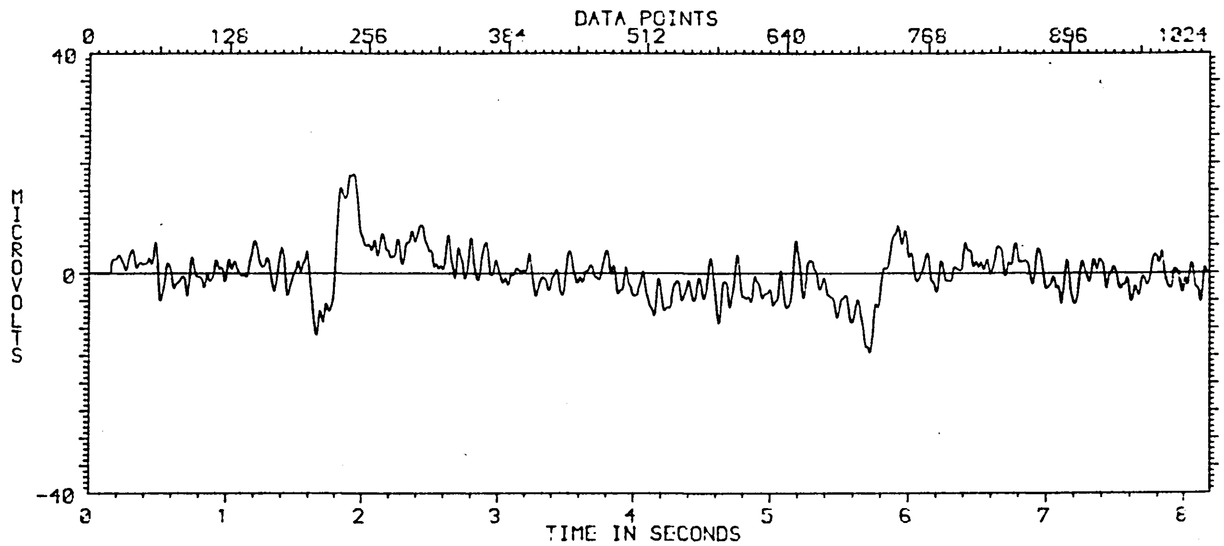

Figure 2.53

A pair of 32 trial averaged AEPs without OAR

The AEP was modelled piecewise linearly as shown in Figure 2.54.

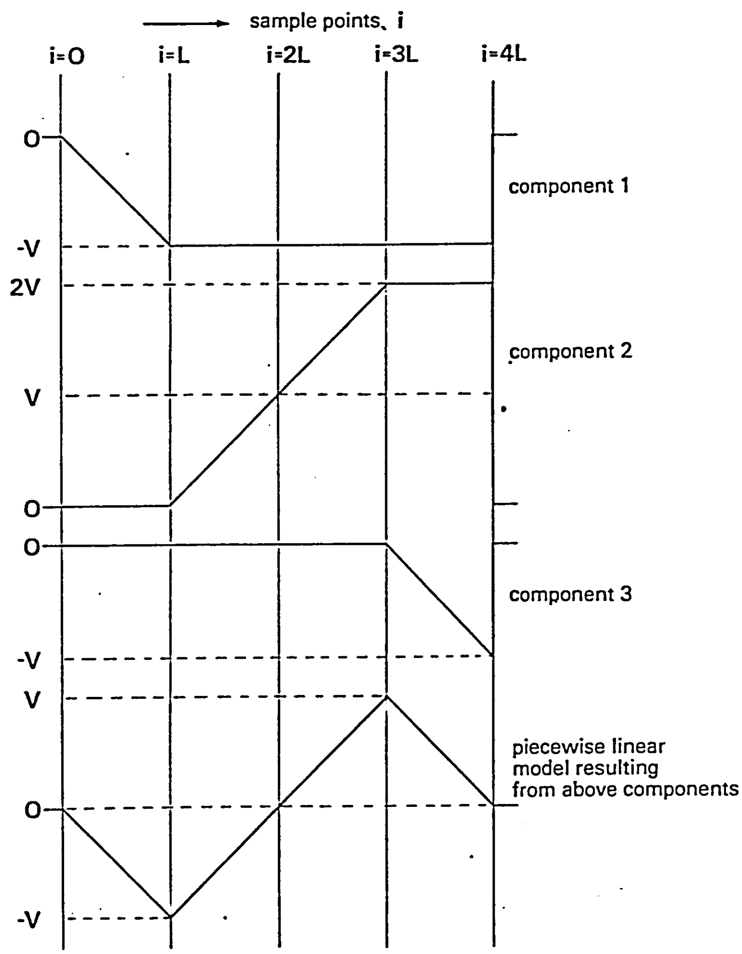

Figure 2.54

The piecewise linear model for the AEP

Figures 2.55 - 2.58 show, respectively, the averaged AEPs obtained by NR-OAR-NM, NR-OAR-WM, R-OAR-NM and R-OAR-WM and have all had d.c. levels removed.

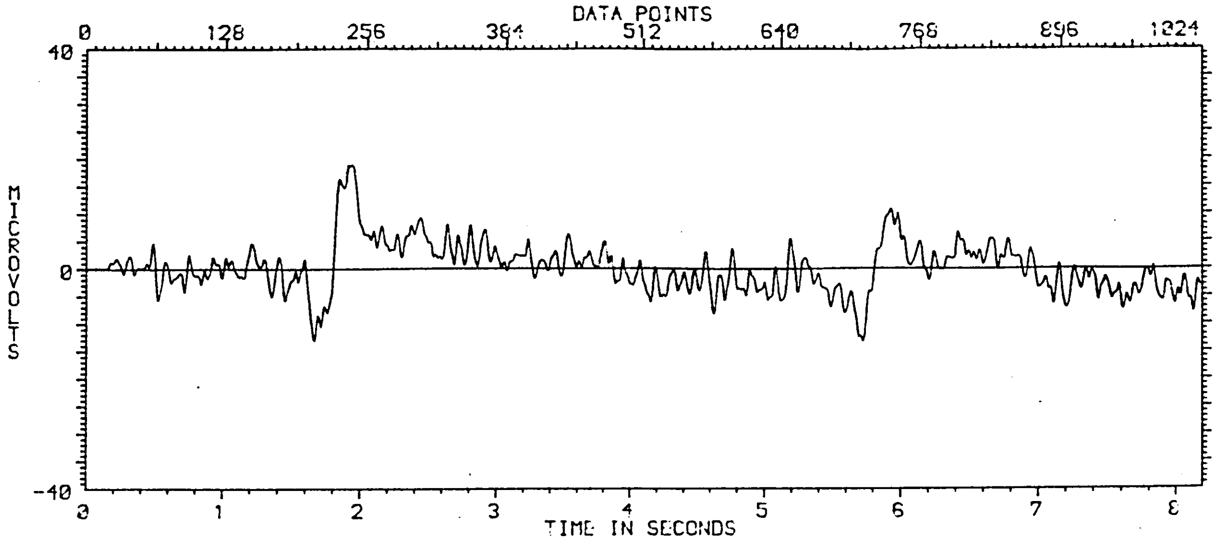

Figure 2.55

The averaged AEP subsequent to NR-OAR-NM (dc = 0)

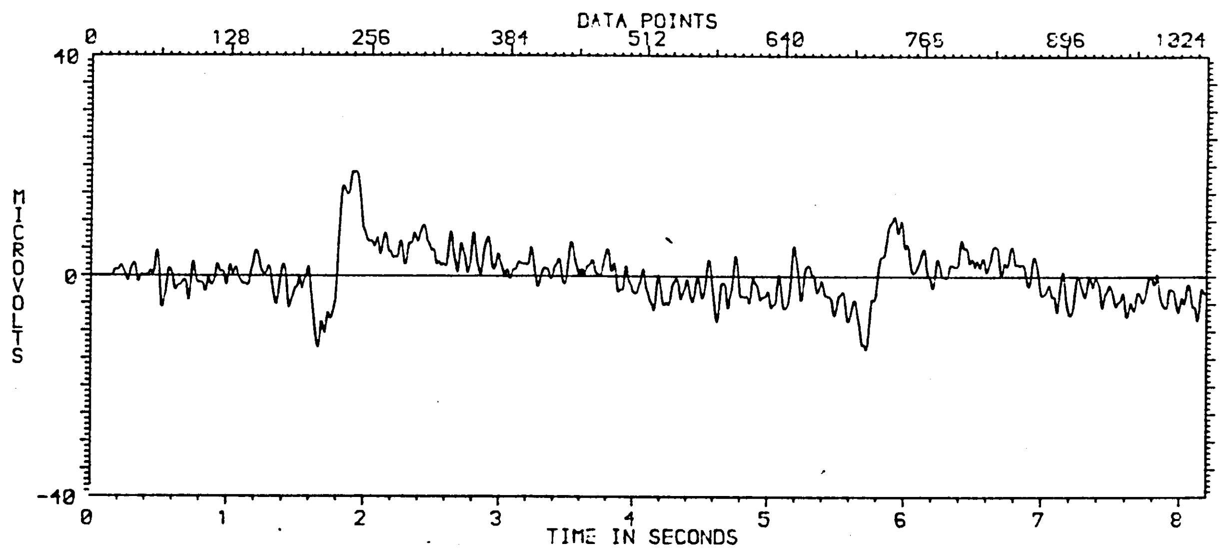

Figure 2.56

The averaged AEP subsequent to NR-OAR-WM (dc = 0)

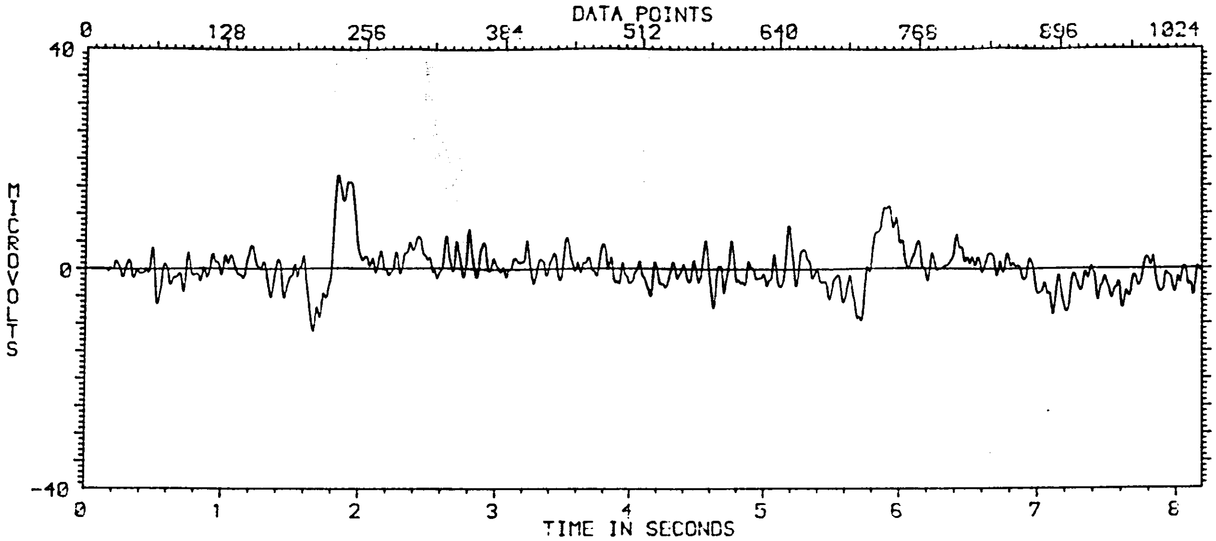

Figure 2.57

The averaged AEP subsequent to R-OAR-NM (dc = 0)

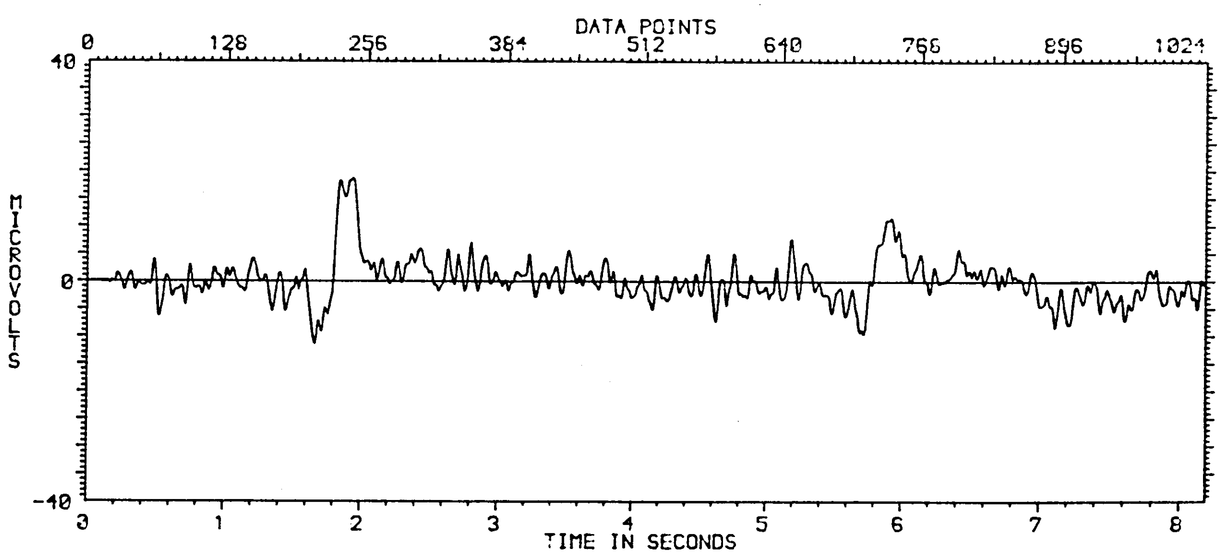

Figure 2.58

The averaged AEP subsequent to R-OAR-WM (dc = 0)

For both NR-OAR cases the only noticeable AEP difference (compared to the no-OAR case of Figure 2.53) is in

AEP_1 where an attenuation of ~1μV occurs at either extremity, i.e., the apparent shape and development of each AEP is unchanged, although changes in level can be discerned (~2.5μV for AEP_2).

When R-OAR-NM is applied, not only do the level and magnitude differences, described above , occur but there are also small changes in the shape of the first AEP. When modelling is applied recursively it is not clear that these changes are overcome. These differences can be seen in the upper region of the AEP by noting that it possesses two peaks and further noting their relative magnitudes.

Attention is now turned to the case when d.c. level removal is performed only for display purposes subsequent to OAR processing. Figures 2.59 - 2.62 show EEGs corrected by NR-OAR-NM, NR-OAR-WM, R-OAR-NM and R-OAR-WM respectively. Both NR-OAR methods give very similar results to each other. The R-OAR methods show a small difference for the first AEP but otherwise are similar. Comparing no-OAR (Figure 2.53) with NR-OAR (both with and without modelling) there can be seen a difference in peak to peak values for both AEPs. For no-OAR the peak to peak values for first and second AEPs were ~29.5μV and ~23μV respectively. When NR-OAR was applied these values became ~36μV and ~28μV respectively. However, the general features of the AEP shape are similar. Application of R-OAR-NM shows peak to peak values of ~30μV and ~26μV for the first and second AEPs (Figure 2.61). For R-OAR-WM (Figure 2.62)) these values are ~29μV and ~26μV respectively. AEP shape modification can be discerned in a similar fashion to that described above for R-OAR-NM (dc = 0) and R-OAR-WM (dc = 0), i.e., Figures 2.57 and 2.58.

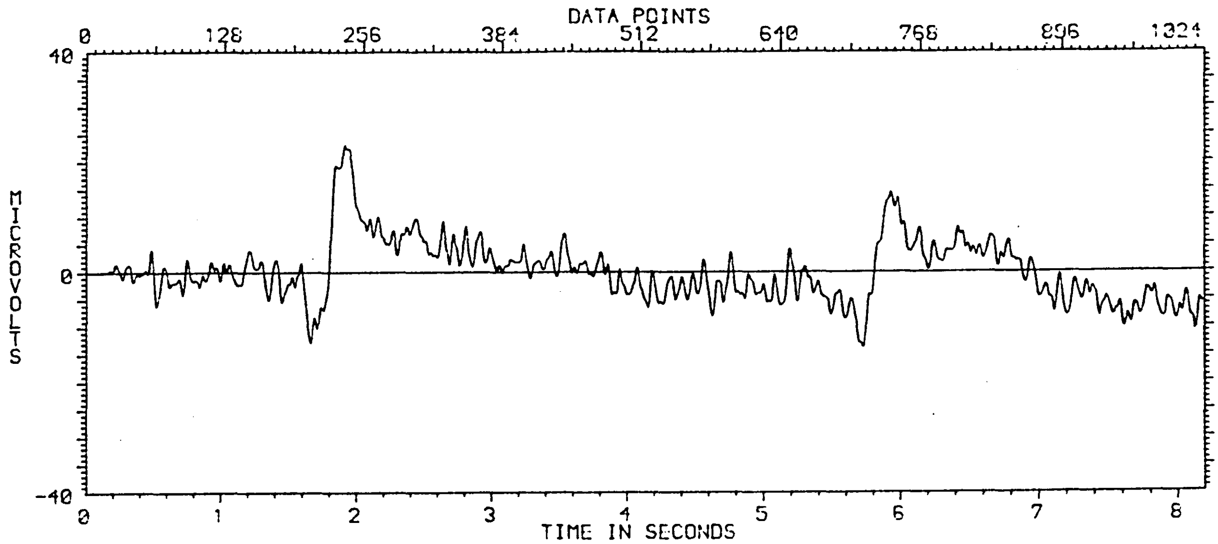

Figure 2.59

The averaged AEP subsequent to NR-OAR-NM (dc $\neq$ 0)

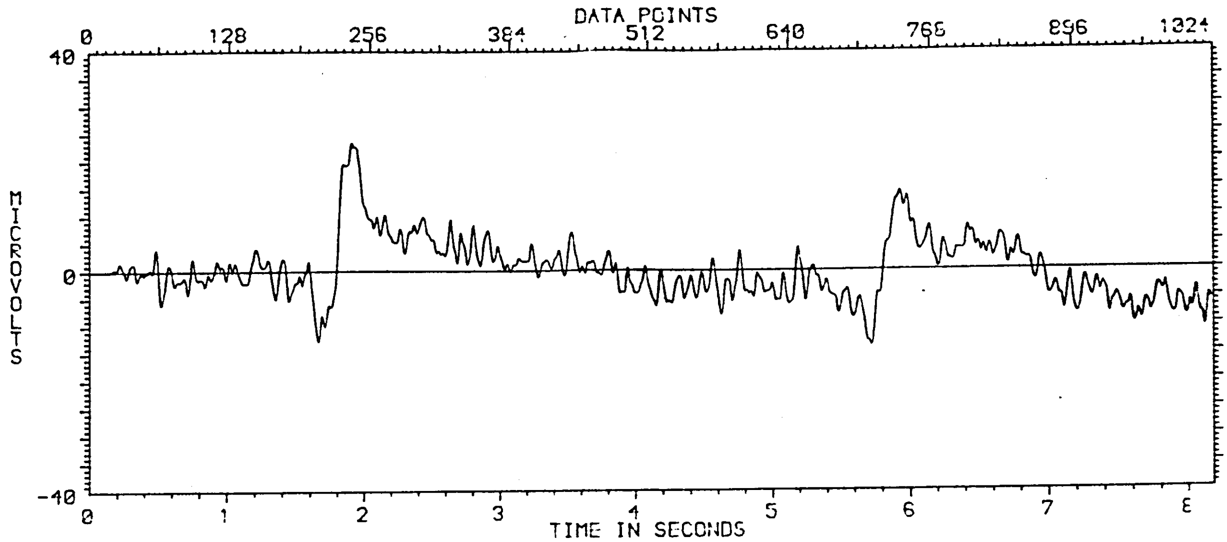

Figure 2.60

The averaged AEP subsequent to NR-OAR-WM (dc ≠ 0)

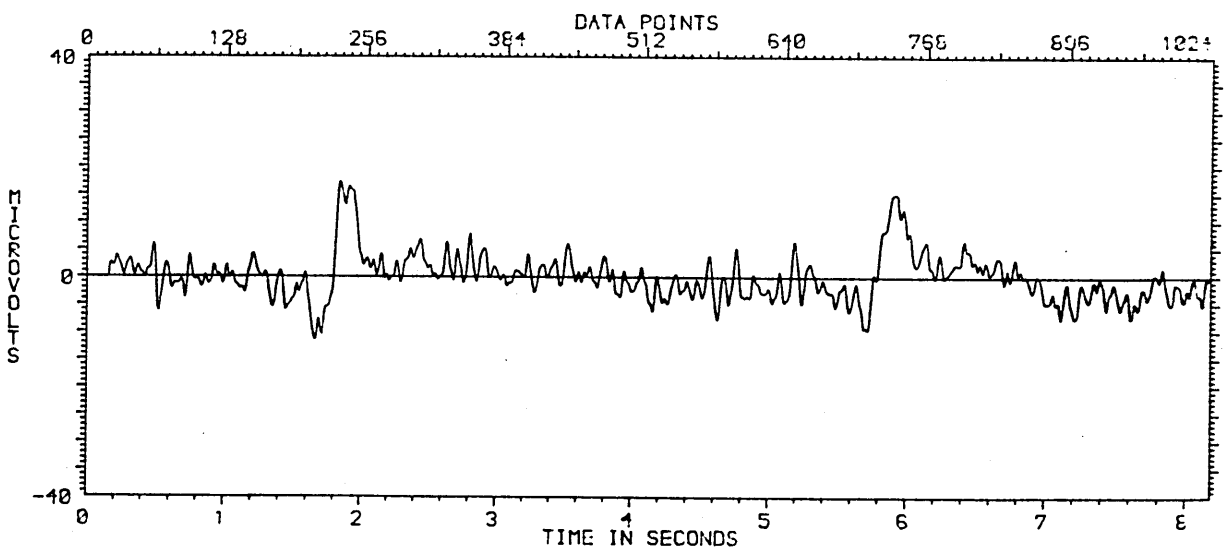

Figure 2.61

The averaged AEP subsequent to R-OAR-NM (dc ≠ 0)

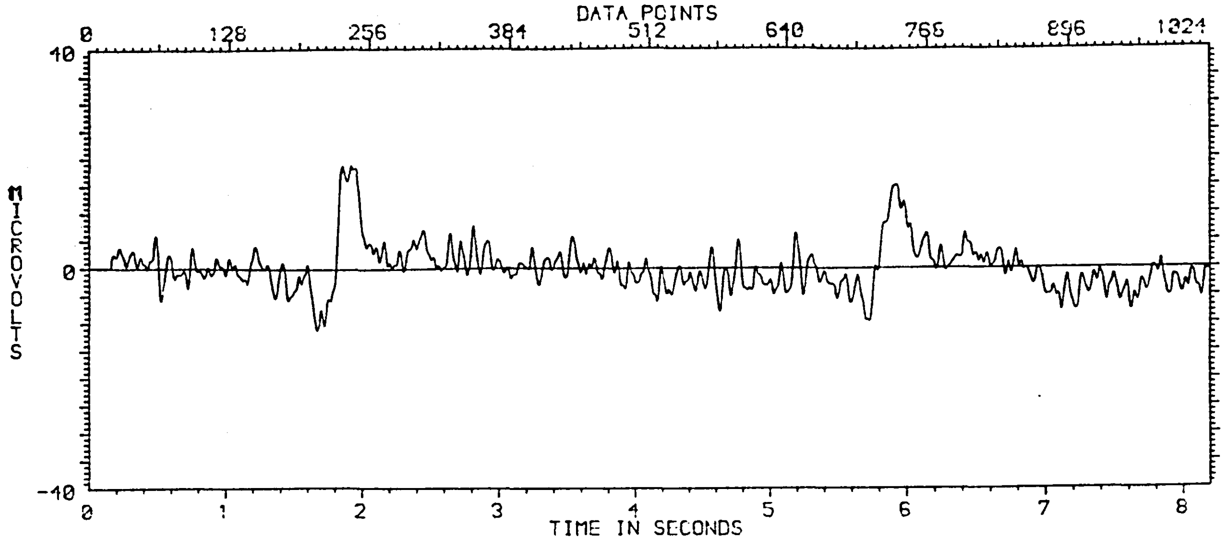

Figure 2.62

The averaged AEP subsequent to R-OAR-WM (dc $\neq$ 0)

From the above results it is concluded that NR-OAR introduces amplitude changes but causes no distortion, that omission of response modelling (i.e., NR-OAR-NM) produces no discernible differences when compared to NR-OAR-WM and that the use of R-OAR causes response distortion and even when response modelling is applied this distortion remains.

2.7 DISCUSSION

The results presented in this chapter were obtained from both simulated and real data. For the simulated data observations it was shown that modelling of the response is necessary for efficient OAR (for both non-recursive and recursive methods) and to avoid response distortion (recursive method). When the real data results were studied the response distortion of the CNV was evident when R-OAR-NM was used and it was demonstrated that response modelling reduced this distortion. By contrast the NR-OAR method was found to be relatively insensitive to the presence or absence of a response model. Since the opportunity to devise accurate CNV models did not exist, it was decided to use the NR-OAR-NM method.

The consequences of removing the d.c. level from all the data channels or of leaving it in have been investigated and showed some contradictory results. It appeared (from Section 2.5.1.1) that non-removal of the d.c. level obviated the need for modelling. However, the results of Section 2.5.1.2 indicated that modelling could not be avoided by non d.c. level removal. This led to consideration of d.c. offsets in the data which, in Section 2.5.1.3, produced results showing the necessity of d.c. level removal. Section 2.5.2 gave results which were neutral. In considering these findings discussion is restricted to the two sections in which offsets were present. This can be justified by noting that for the real data, when no OAR was applied, and when d.c. levels were removed, EEG offsets of -55.3μV and 10.1μV for the CNV and AEP data were observed (for the 32 trial averaged waveforms). The differences between the observations of Section 2.5.1.3 and 2.5.2 might then be accounted for by the fact that the offsets for the latter were proportionally smaller than those for the former. Thus, it is felt that the best course of action was to perform d.c. level removal on the data.

In Section 2.5.2 the effect of CNV and response model mismatching was considered. While for NR-OAR-WM there appeared to be no effect, there was a small observable effect for R-OAR-WM. This is not surprising because of the previous discussion of the RLS and OLS methods. Another cause of mismatching between the response and its model which was not investigated but could well affect results, is the temporal misalignment between them.

In light of the results of this chapter the following OAR processing was selected for the remainder of this work:

(i) remove d.c. levels from all data channels;

(ii) use non-recursive OAR and

(iii) omit a response model from equation (2.1).

REFERENCES

FORTGENS C. and DE BRUIN M.P. (1983)
"Removal of Eye Movement and ECG Arefacts from the Non-cephalic Reference EEG"
Electroeneceph. and Clin. Neurophysiol. 1983,56:90-96

IFEACHOR E.C. (1984)
"Investigation of Ocular Artefacts in the Human EEG and Their Removal by a Microprocessor-based Instrument"
Ph.D. Thesis, Plymouth Polytechnic, Dec. 1984

IFEACHOR E.C., JERVIS, B.W., MORRIS, E.L., ALLEN E.M. and HUDSON N.R. (1986a)
"New Online Method for Removing Ocular Artefacts from EEG Signals"
Med. and Biol. Eng. and Comp. July 1986:356-364

IFEACHOR E.C., JERVIS B.W., MORRIS E.L., ALLEN E.M. and HUDSON N.R. (1986b)
"A New Microcomputer-based Online Ocular Artefact Removal (OAR) System"
Proc. IEE, pt.A, July 1986,133(5):291-300

JERVIS B.W., NICHOLS M.J., ALLEN E.M., HUDSON N.R. and JOHNSON T.E. (1980)
"The Quantitative Assessment of Electroencephalograms Corrected for Eye Movement Artefact"
1st Europ. Signal Proc. Conf. Lausanne, Switzerland, Sept. 1980

JERVIS B.W., NICHOLS M.J., ALLEN E.M., HUDSON, N.R. and JOHNSON T.E. (1985)
"The Assessment of Two Methods for Removing Eye Movement Artefacts from the EEG"
Electroeneceph. and Clin. Neurophysiol. 1985,61:444-452

JERVIS B.W., IFEACHOR E.C. and ALLEN E.M. (1988)
"The Removal of Ocular Artefacts from the Electroencephalogram: a Review"
Med. and Biol. Eng. and Comput. Jan. 1988,26:2-12

NICHOLS M.J. (1982)
"An Investigation of the Contingent Negative Variation Using Signal Processing Methods"
Ph.D. Thesis, Plymouth Polytechnic, Aug. 1982

QUILTER P.M., MACGILLIVRAY B.B. and WADBROOK D.G. (1977)
"The Removal of the Eye Movement Artefact from EEG Signals Using Correlation Techniques"
Random Signal Analysis, IEE Conference Publication 1977,159:93-100

WEERTS T.C. and LANG P.J. (1973)
"The Effects of Eye Fixation and Stimulus and Response Location on the Contingent Negative Variation (CNV)"
Biol. Psychol. 1973,1:1-19

3 FREQUENCY DOMAIN METHODS

3.1 CHAPTER OUTLINE

The frequency domain signal processing techniques to be applied to the data are discussed in this chapter, together with demonstrations and results which illustrate some of these points and which indicate the preferred processing steps to be followed.

Section 3.2 is a brief introduction to the Discrete, Fast and Continuous Fourier Transforms along with power and energy spectra. The use of the FFT to obtain amplitude and phase spectra and the energy spectrum will be given together with the use of each of these representations. Section 3.3 covers a range of signal processing topics: sampling, aliasing, picket-fencing, augmenting zeroes and the fundamental limitations of the data epoch length. Section 3.4 describes windowing and its drawbacks, spectral leakage, biasing, and two examples of windows (Tukey and Kaiser-Bessel). Section 3.5 describes an investigation to compare (and determine the best choice of) Tukey and Kaiser-Bessel windows. Section 3.6 is a review of the results with recommendations as to the processing regime to be used in the rest of this work.

3.2 THE FOURIER TRANSFORM AND ENERGY SPECTRUM

3.2.1 BACKGROUND

The Fourier Transform (FT) is a means whereby one representation of a signal (in the time domain) can be translated into another representation (in the frequency domain). The reverse process is equally possible but is not required in this work. The time domain signal (e.g. EEG recording) is Fourier Transformed to yield a pair of spectra (either amplitude and phase or real and imaginary components of a complex function). All these spectra show the variation of some quantity (depending on which spectrum is considered) against frequency. From these it is possilbe to obtain information showing the variation of power (for an infinite signal) or energy (for finite signals) with frequency. The former is the power spactrum, the latter the energy spectrum.

Signals may be continuous (in time) or discrete. Because of the nature of digital computers (on which much signal processing work is performed) even a continuous signal will be rendered discrete by sampling.

The FT $F(\omega)$ of a time signal $f(t)$ is given by:

$$F(\omega) = \int_{-\infty}^{\infty} f(t)e^{-j\omega t}dt \qquad -(3.1)$$

Where ω is angular frequency = $2\pi f$ and f is frequency.

$F(\omega)$ is complex and may be written as the sum of two other functions:

$$F(\omega) = R(\omega) + jI(\omega) \qquad -(3.2)$$

where $R(\omega)$ is an even function and $I(\omega)$ is an odd function.

It is also possible to consider the amplitude, ($A(\omega)$), and phase,($\phi(\omega)$), spectra in which:

$$A(\omega) = |F(\omega)| = \sqrt{R^2(\omega) + I^2(\omega)} \qquad -(3.3)$$

$$\text{and } \phi(\omega) = \tan^{-1}\frac{I(\omega)}{R(\omega)} \qquad -(3.4)$$

Equations (3.1) - (3.4) are for continuous aperiodic signals, $f(t)$. For sampled data signals the translation from time to frequency domain is made by the Discrete Fourier Transform (DFT).

Let a sequence of N samples be spaced T seconds apart in an interval (0,(N-1)T), i.e.,

$$f(kT) = f(0), f(T), f(2T), \ldots, f([N-1]T) \qquad -(3.5)$$

The DFT is defined as a sequence of N complex-valued samples in the frequency domain by (STREMLER, 1977):

$$F_D(n\Omega) = \sum_{k=0}^{N-1} f(kT)e^{-j\Omega Tnk} \qquad n = 0,1,2,\ldots,N-1 \qquad -(3.6)$$

where $\Omega = \dfrac{2\pi}{(N-1)T}$

In using the DFT the FT is being approximated numerically. To see the relationship between the FT and DFT consider a truncated time signal $\tilde{f}(t)$ with an FT of $\tilde{F}(\omega)$. If:

$$\tilde{f}(t) = \begin{cases} f(t) & 0 \leqslant t \leqslant (N-1)T \\ 0 & \text{elsewhere} \end{cases} \qquad -(3.7)$$

then:

$$\tilde{F}(\omega) = \int_0^{(N-1)T} f(t)e^{-j\omega t}dt \qquad -(3.8)$$

With variable changes ω->$n\Omega$, t->kT and dt ->T equation (3.8) can be approximated by (STREMLER, 1977):

$$\tilde{F}(n\Omega) = \sum_{k=0}^{N-1} f(kT)e^{-jn\Omega kT}.T \qquad -(3.9)$$

then:

$$\tilde{F}(\omega)\Big|_{\omega=n\Omega} = TF_D(n\Omega) \qquad -(3.10)$$

The complex spectrum $F_D(n\Omega)$ can be written in the form of equation (3.2) as:

$$F_D(n\Omega) = R_D(n\Omega) + jI_D(n\Omega) \qquad -(3.11)$$

and hence the phase spectrum of equation (3.4) can be written:

$$\text{and } \phi_D(n\Omega) = \tan^{-1}\frac{I_D(n\Omega)}{R_D(n\Omega)} \qquad -(3.12)$$

Although direct application of equation (3.6) will give the required spectrum this method is seldom applied because of the large number (N^2) of complex multiplications required with increasing N. Instead Fast Fourier Transform (FFT) algorithms have been developed (COOLEY and TUKEY, 1965) which offer drastic reductions by using the fact that many multiplications are repetitive. This can result in as few as $(N/2)\log_2 N$ multiplications being required (BERGLAND, 1969). For N = 1024 this means a reduction in complex multiplications of 204.8 : 1. A large number of computer programs to perform the FFT are available. The one used here is due to ROBINSON (1978) and is given in Appendix 2 as subroutine NLOGN.

3.2.2 ENERGY SPECTRUM

The energy spectrum denoted $G(n\Omega)$ of a signal is given by (OTNES and ENOCHSON, 1972):

$$G(n\Omega) = \frac{2T}{N}\left|F_D(n\Omega)\right|^2 \qquad n = 0, 1, 2, \ldots, N/2 \qquad -(3.13)$$

The factor of 2 is present in order to render the double-sided spectrum single-sided.

3.2.3 THE USE OF THE SPECTRA IN SIGNAL PROCESSING

The energy and phase spectra were computed for the averaged waveforms obtained from the individual trials subsequent to OAR. These were then studied in order to determine which DFT harmonics were to be subjected to the statistical tests described in Chapter 4. In practice, this selection was based on the energy spectrum alone (a typical example is given in Figure 3.1) since the phase spectrum (Figure 3.2 shows an example) gave no visual clues as to 'interesting' harmonics. The energy spectrum of the averaged waveform, however, showed frequencies of maximum energy content.

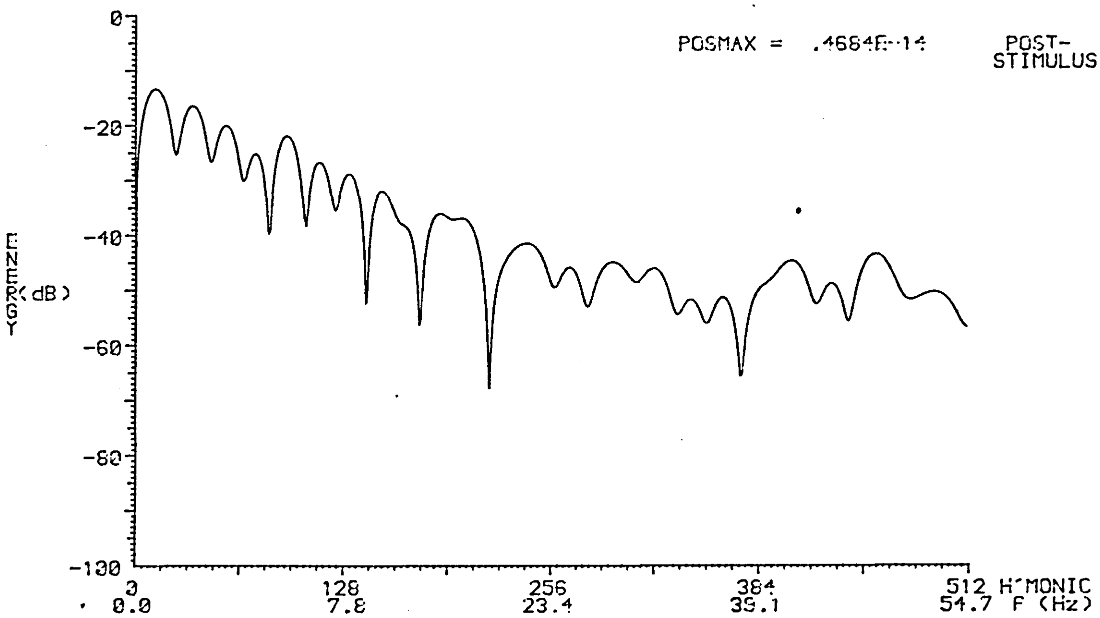

Figure 3.1

The energy spectrum of a 32 trial averaged CNV

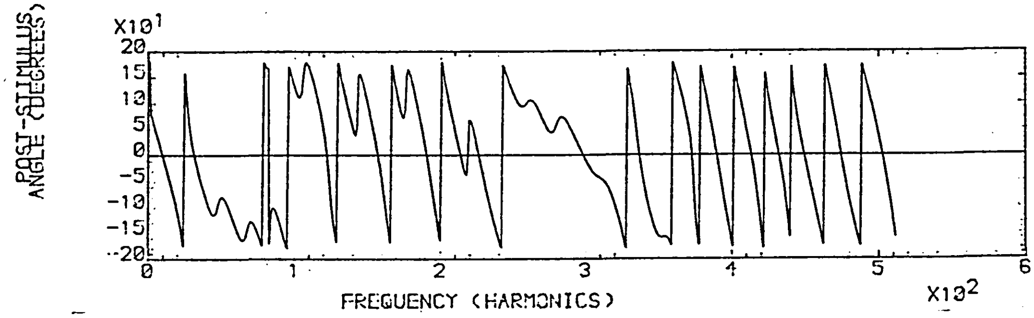

Figure 3.2

The phase spectrum of a 32 trial averaged CNV

After selection of the required harmonics the phase and amplitude spectra for each trial for the pre-stimulus (pre-S_1) and post-stimulus (between end of S_1 and start of S_2) eras were computed and subjected to the statistical tests of Chapter 4. The pre-stimulus era comprised one epoch for both 1 and 4 second CNVs. The post-stimulus era had one epoch for the 1 second CNV, but consisted of two epochs for the 4 second data.

3.3 FURTHER SIGNAL PROCESSING TOPICS

3.3.1 SAMPLING AND ALIASING

Sampling is the process of measuring the value of a continuous waveform at (usually) equally spaced time intervals. This renders a discrete (sampled) representation of the underlying continuous signal. The spectrum of a sampled signal is a repeated version of that of the underlying continuous waveform repeated every $2\pi/T$ rads/sec (T is the interval between samples), (LYNN, 1982), as shown in Figure 3.3.

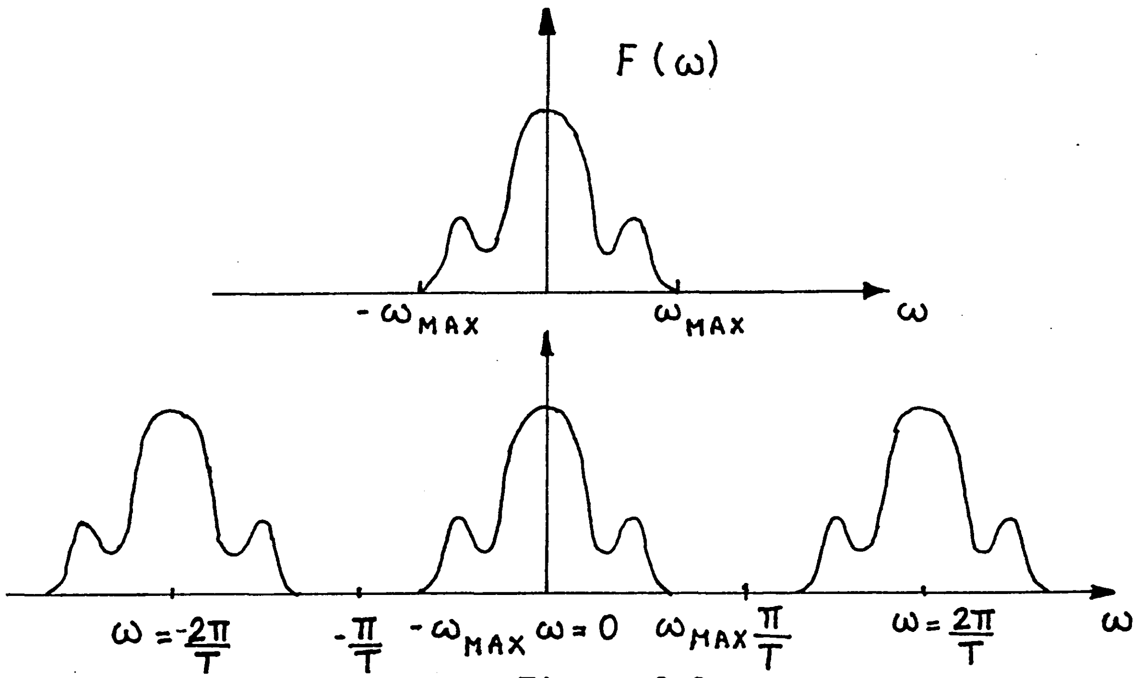

Figure 3.3

The repeated spectra of a sampled signal

If ω_{MAX} (the highest frequency component present in the signal) exceeds π/T overlap of the repeated spectrum, a phenomena known as aliasing or spectral folding, occurs. This is shown in Figure 3.4. As can be seen this is an undesirable process (the true spectrum is distorted thus giving an erroneous representation). To prevent this problem use of an anti-aliasing filter is recommended (BELLANGER, 1984), the purpose of which is to ensure that the frequency components of the signal at/or greater than a threshold value (the folding frequency) are negligible.

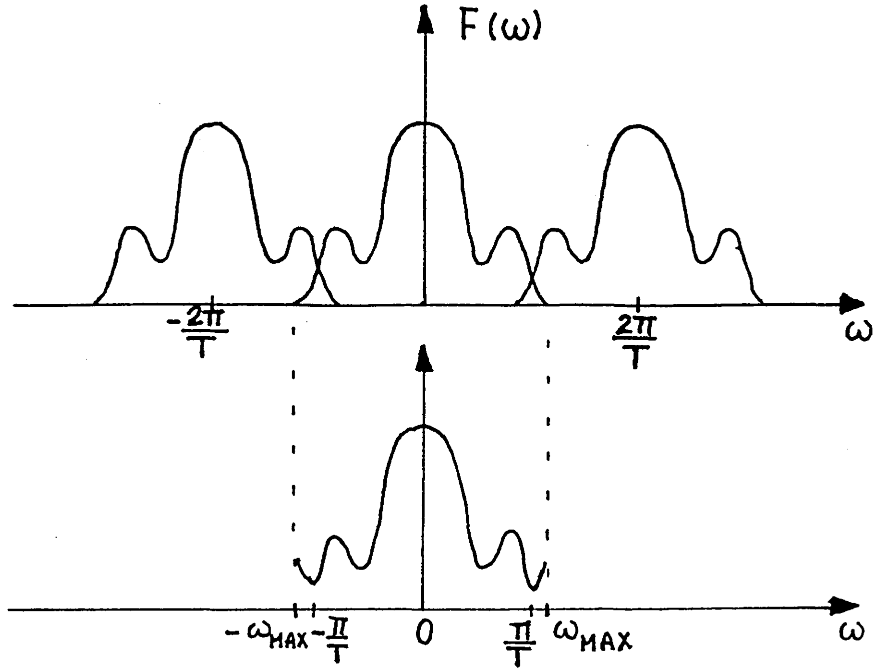

Figure 3.4

Aliasing when spectra of Figure 3.3 overlap.

In practice the anti-aliasing filter is designed to have a cut-off frequency below that of the folding frequency. Such a filter is given in subroutine FILTER (see Appendix 2).

3.3.2 PICKET FENCE (SCALLOPING) EFFECT

Since the DFT spectrum is also discrete any signal component which occurs at a frequency between two adjacent DFT harmonics will have its energy shared between these harmonics thus distorting them. This phenomenon is known as picket-fence or scalloping effect. From this description it can be seen that reducing the DFT harmonic separation (i.e., incresing the number of harmonics for a given frequency range) will reduce the chance that a signal component falls between adjacent DFT harmonics. This reduction in harmonic separation is achieved by using augmenting zeroes which are appended to the end of the data sequence (comprising N points). The number of augmenting zeroes N' must satisfy two conditions:

(i) $N + N' = 2^M$ where M is an integer (since the FFT algorithm usually needs an integer power of 2 number of data points);

(ii) sufficient resolution is achieved, given by $1/[(N+N'-1)T]$

Those frequency components of the signal which do not coincide with the DFT harmonics suffer a loss in gain known as scalloping loss (HARRIS, 1978).

3.3.3 LIMITATIONS DUE TO DATA DURATION

Although in principle the previous section indicates that any required resolution can be achieved, another limit exists which is that of the signal duration, NT. This places a fundamental limit on the resolution that can be obtained despite the use of augmenting zeroes. This phenomena is related to time-bandwidth product. This is given by HARRIS (1978) as $NTB \geqslant 1/(4\pi)$ where B is the bandwidth. Thus the shorter (in time) a signal is the wider its bandwidth will be and the greater the likelihood of overlapping of adjacent signal components.

3.4 THE USE OF DATA WINDOWS

3.4.1 SPECTRAL LEAKAGE AND DATA WINDOWS

As has been noted any real signal is of finite duration. This is, in effect, the truncation of an infinite signal which can be considered equivalent to multiplying, or windowing, the infinite signal by a rectangular pulse, or window, of width NT and height unity (Figure 3.5).

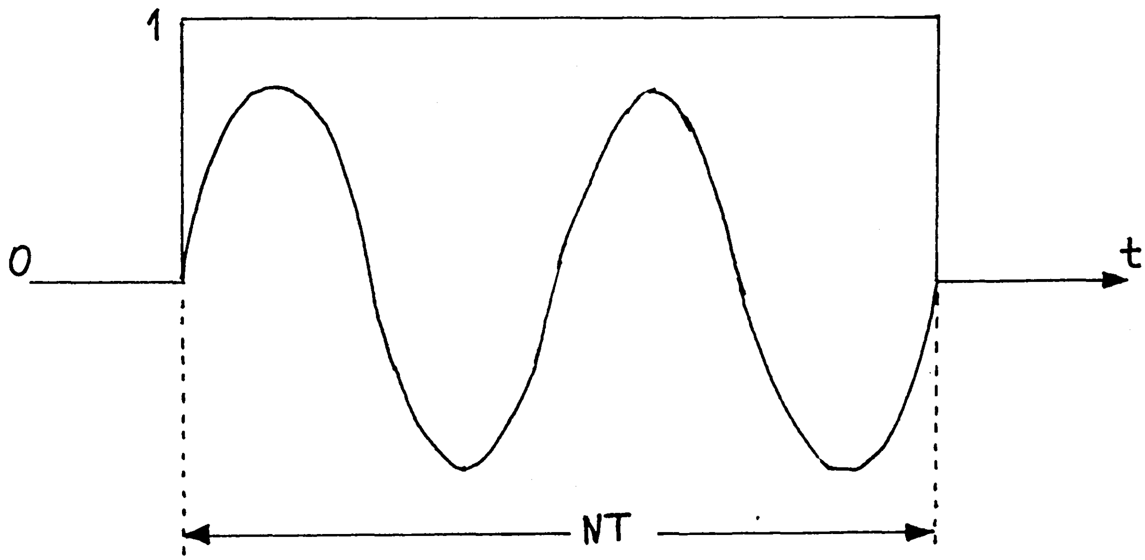

Figure 3.5

Sinewave signal whose frequency is a harmonic of 1/(NT)

Since the signal shown there is a simple sinewave possessing an integer number of cycles in the interval NT this effectively means that no signal truncation has occurred (i.e., that the signal is periodic and exists throughout all time) and will possess the amplitude spectrum of Figure 3.6.

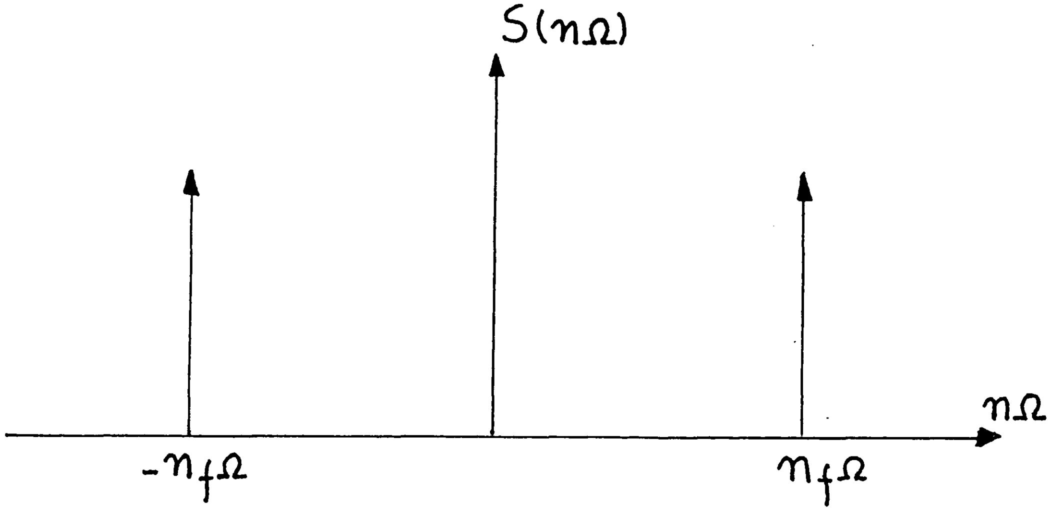

Figure 3.6

The amplitude spectrum of the signal of Figure 3.5

Now in practice a number of sinusoids of different frequency, amplitude and phase will make up the signal of interest, and so it is unlikely that all the components will have frequencies such that every one has an integer number of cycles within the window. To see the effect of this consider a sinewave in which there are a non-integer number of cycles in the window, as shown in Figure 3.7.

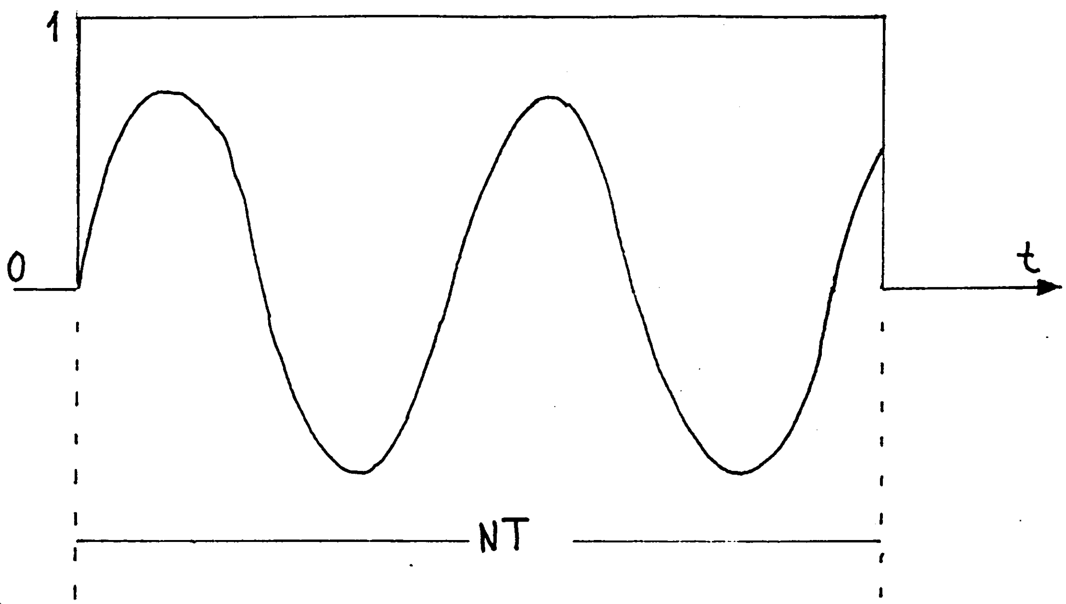

Figure 3.7

Sinewave signal whose frequency is not a harmonic of 1/(NT)

In the time domain this situation can be described by:

$$s(i) = w(i)x(i) \quad i = 1, 2, \ldots, N \qquad -(3.14)$$

in which s(i) = the windowed signal

w(i) = the window function

x(i) = the true signal

Now, time domain multiplication of two signals is equivalent to convolving the spectra of the two signals in the frequency domain and is expressed as (COOLEY et al, 1969):

$$S(n\Omega) = \sum_{k=-N}^{N} W(n\Omega - k\Omega)X(k\Omega) \qquad -(3.15)$$

where: $n\Omega$ = angular frequency of nth harmonic;

$S(n\Omega)$ = complex DFT component at frequency $n\Omega$;

$W(n\Omega)$ = amplitude spectrum of window at $n\Omega$;

$X(k\Omega)$ = amplitude spectrum of true signal at $k\Omega$.

For the case of Figure 3.7 the amplitude spectrum of X is given by:

$$X(k\Omega) = \begin{cases} A_s \text{ at } k = \pm n_f \\ 0 \text{ for all other } k \end{cases} \qquad k = 0, \pm 1, \pm 2, \ldots, \pm N \qquad -(3.16)$$

where A_s is the signal amplitude at frequency $\pm n_f\Omega$.

The amplitude spectrum of the rectangular window is given by:

$$W(n\Omega) = \frac{NT.\sin\left(\frac{n\Omega NT}{2}\right)}{\left(\frac{n\Omega NT}{2}\right)} = NT.Sa\left(\frac{n\Omega NT}{2}\right) \qquad -(3.17)$$

Where Sa denotes the sampling function. Substitution of equations (3.16) and (3.17) into (3.15) gives the amplitude spectrum of Figure 3.8.

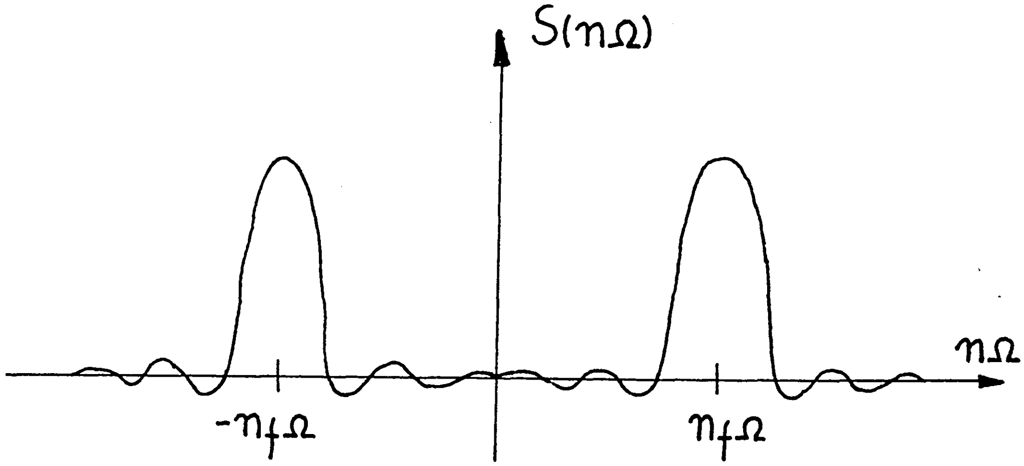

Figure 3.8

The amplitude spectrum of the signal of Figure 3.7

This shows two overlapping sampling functions centred on the two frequency components due to the signal. Each sampling function comprises a mainlobe and an infinite number of sidelobes of decreasing amplitude. Thus instead of having two impulses at $\pm n_f \Omega$ spurious peaks have been introduced into the spectrum. This is known as spectral leakage and refers to the fact that energy in the original spectral components at $k = \pm n_f$ leaks to other frequencies after truncation in time (STARK and TUTEUR, 1979). When a large number of varying signal components are present it can be seen that the true spectrum can become distorted with spurious peaks being introduced or true ones being cancelled out.

In order to reduce this distortion it is desirable to suppress the large sidelobes and hence reduce leakage. From the time domain point of view this is equivalent to reducing the discontinuities at each end of the finite time segment (OTNES and ENOCHSON, 1972). A variety of windows have been developed each of which aim to satisfy a number of criteria as to what constitutes a 'good' window. The general method is to multiply the data by a window function which has a value of 1 at the mid-data point and which is tapered (hence windowing in the time domain is also known as tapering) smoothly to zero at each end of the data. Studies of data windows have been made (DURRANI and NIGHTINGALE, 1972 and HARRIS, 1978). The former studied 3 families of data windows and the latter investigated 23 such families.

HARRIS op. cit. ,defined and evaluated a number of measures on each window: equivalent noise bandwidth (ENBW); processing gain (PG); scalloping loss (SL); worst case processing loss (WCPL) and minimum resolution bandwidth (MRB). ENBW is the width (in the frequency domain) of a rectangular filter with the same peak power gain and which transmits the same amount of noise power as the window. To improve harmonic detection the noise signal should be kept as small as possible by minimising ENBW. PG is the ratio of output signal-to-noise ratio to input signal-to-noise ratio. SL is a measure of the loss in gain of a signal frequency component falling midway between adjacent DFT frequency spectrum harmonics. WCPL (in dB) is the sum of maximum SL of a window and processing loss (PL) of a window.

PL is due to the window having reduced the data to zero at or near the window boundaries. MRB is a measure of frequency resolution for two adjacent mainlobes of equal amplitude. It is defined as the bandwidth when mainlobe energy is 6 dB below its peak value (i.e., 0.25 of the maximum value).

HARRIS (1978) found the ratio ENBW to 3dB bandwidth was a sensitive indicator of overall window performance with a value in the range 1.04 to 1.055 indicating good performance. He also concluded that maximum dynamic range of multitone detection requires the window transform (i.e., its frequency domain spectrum) to possess a highly concentrated central lobe with very low sidelobe structure. Four window families met this requirement of which two had minor performance advantages over the others: the Blackman-Harris and Kaiser-Bessel windows. Of these the Kaiser-Bessel window was recommended on the grounds of ease of computation of coefficients and simple trade-off between mainlobe width and sidelobe levels by varying a single parameter, α.

In light of the work of NICHOLS (1982) where a 12.5% taper was applied to each end of a Tukey window prior to signal processing it was decided to compare this window with the Kaiser-Bessel window.

3.4.2 THE TUKEY WINDOW

This is in fact a family of windows in which the amount of taper varies with a parameter α. This is the ratio of total tapered length (i.e., the tapers at both ends of the data) to the data length.

For a data sequence of N points, with total taper α, the window function in discrete time, is given by:

$$w(i) = \begin{cases} 0.5\left[1+\cos\left\{\dfrac{\pi\left(i - \alpha\frac{N}{2}\right)}{\left(\alpha\frac{N}{2}\right)}\right\}\right] & 0 \leqslant i \leqslant \alpha\frac{N}{2} \\ 1 & \alpha\frac{N}{2} \leqslant i \leqslant \left(1 - \frac{\alpha}{2}\right)N \\ 0.5\left[1+\cos\left\{\dfrac{\pi\left(i-\left[1 - \frac{\alpha}{2}\right]N\right)}{\left(\alpha\frac{N}{2}\right)}\right\}\right] & \left(1-\frac{\alpha}{2}\right)N \leqslant i \leqslant N \end{cases} \quad \text{-(3.18)}$$

This was implemented in subroutine TAPER2 (see Appendix 2) taken from NICHOLS, 1982) with a value of $\alpha = 0.25$ (i.e., 12.5% taper at each end of the data).

3.4.3 THE KAISER-BESSEL WINDOW

The Kaiser-Bessel family of windows are given by (HARRIS, 1978):

$$w(i) = \frac{I_o[\pi\alpha\sqrt{1 - (\frac{i}{N/2})^2}]}{I_o[\pi\alpha]} \qquad 0 \leqslant |i| \leqslant \frac{N}{2} \qquad -(3.19)$$

This expression is for a window even about the origin (i.e., an odd number of points). To convert to an even number of points sequence the right end point is discarded and the sequence right shifted so that the left most point coincides with the origin. I_o is the zero-order modified Bessel function of the first kind given by:

$$I_o(x) = \sum_{k=0}^{\infty} [\frac{(x/2)^k}{k!}]^2 \qquad -(3.20)$$

The above was implemented using the FORTRAN subroutines TAPKAI, KAIGEN and SUFACT and function program unit BSSL.* These have been modified to improve computational efficiency. In practice an upper limit of k = 32 in the summation of equation (3.20) is adequate since values of k>33 produce very small changes in I_o.

---

* Program units KAIGEN and BSSL are based on a program written by Paul Bassingdale.

3.4.4 PROBLEMS WITH WINDOWING

The use of the above windows suffers from a disadvantage: the introduction of a spurious d.c. level due to the windowing process itself. This problem can be corrected for by the use of a data transformation which accompanies the windowing.

To see how this spurious d.c. level arises consider a data sequence x_i and a window function given by the sequence w_i. Denote the means of the respective sequences by μ_x and μ_w and define μ_{xw} as the mean of the windowed data. Further denote x_i' and w_i' as the sequences obtained by subtracting the means μ_x and μ_w from x_i and w_i.

$$\mu_{xw} = \frac{1}{N}\sum_{i=1}^{N} x_i w_i = \frac{1}{N}\sum_{i=1}^{N} [x_i'+\mu_x][w_i'+\mu_w] \qquad -(3.21)$$

$$= \frac{1}{N}\sum_{i=1}^{N} x_i' w_i' + \mu_x \mu_w \qquad -(3.22)$$

Now consider a general window comprising three sections: two symmetrical tapers (at either end of the data) and a central constant region, i.e.,

$$w_i = \begin{cases} y_{iL} & i=1,\ldots,N_T \\ 1 & i=N_T+1,\ldots,N-N_T \\ y_{iU} & i=N-N_T+1,\ldots,N \end{cases} \qquad -(3.23)$$

where N_T is the length of the taper, Y_{iL} is the lower end of the window taper and Y_{iU} the upper end of the taper. However this is required in terms of w'_i, i.e.,

$$\begin{aligned} y_{iL} &= w'_i + \mu_w && i=1,\ldots,N_T \\ 1 &= w'_i + \mu_w && i=N_T+1,\ldots,N-N_T \qquad -(3.24) \\ y_{iU} &= w'_i + \mu_w && i=N-N_T+1,\ldots,N \end{aligned}$$

which can be rearranged to:

$$w'_i = \begin{cases} y_{iL} - \mu_w & i=1,\ldots,N_T \\ 1 - \mu_w & i=N_T+1,\ldots,N-N_T \\ y_{iU} - \mu_w & i=N-N_T+1,\ldots,N \end{cases} \qquad -(3.25)$$

Then:

$$\mu_{xw} = \frac{1}{N}\sum_{i=1}^{N_T} x'_i y_{iL} + \frac{1}{N}\sum_{i=N_T+1}^{N-N_T} x'_i + \frac{1}{N}\sum_{N-N_T+1}^{N} x'_i y_{iU} + \mu_w \frac{1}{N}\sum_{i=1}^{N} x'_i + \mu_x \mu_w$$

Since OAR requires that the d.c. levels be removed (Chapter 2) $\mu_x = 0$ and so the last term of this equation vanishes as does the fourth term. This gives:

$$\mu_{xw} = \frac{1}{N}\{\sum_{i=1}^{N_T} x'_i y_{iL} + \sum_{i=N_T+1}^{N-N_T} x'_i + \sum_{i=N-N_T+1}^{N} x'_i y_{iU}\} \qquad -(3.26)$$

Now the expression in brackets in equation (3.26) will no longer be zero since each end of the data has been tapered. Thus, in general, $\mu_{xw} \neq 0$.

This effect is shown in Figure 3.9.

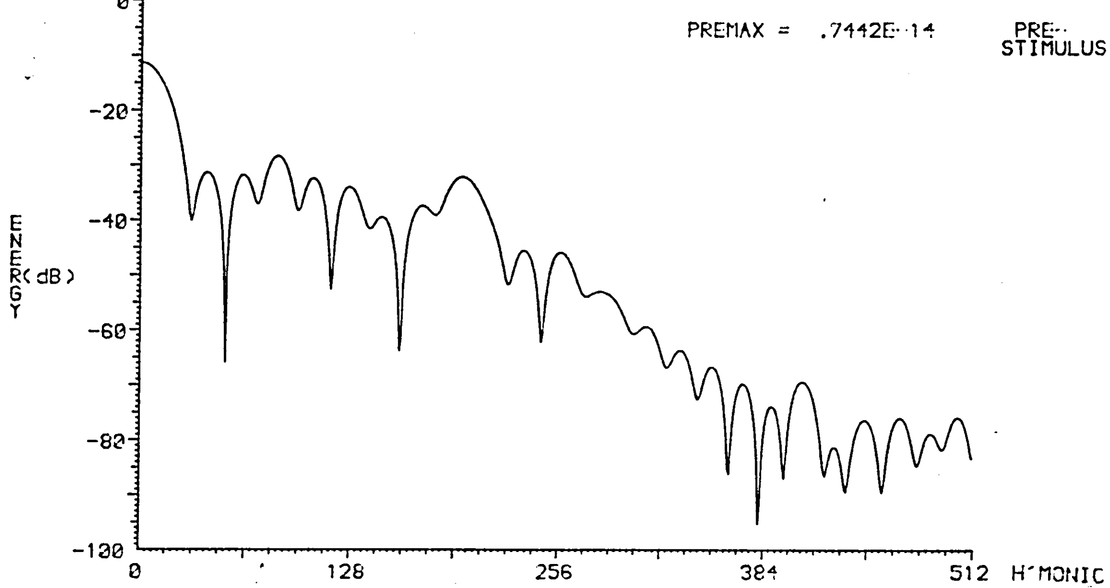

Figure 3.9

Energy spectrum showing the introduction of a dc level due to windowing

The data used to give this spectrum had its mean level removed prior to windowing but, as can be seen, a new d.c. level has been introduced. Figure 3.10 shows the energy spectrum when the d.c. level present, due to windowing, is removed. Although the 0 Hz component is removed a substantial (and undesirable) increase in sidelobe structure occurs.

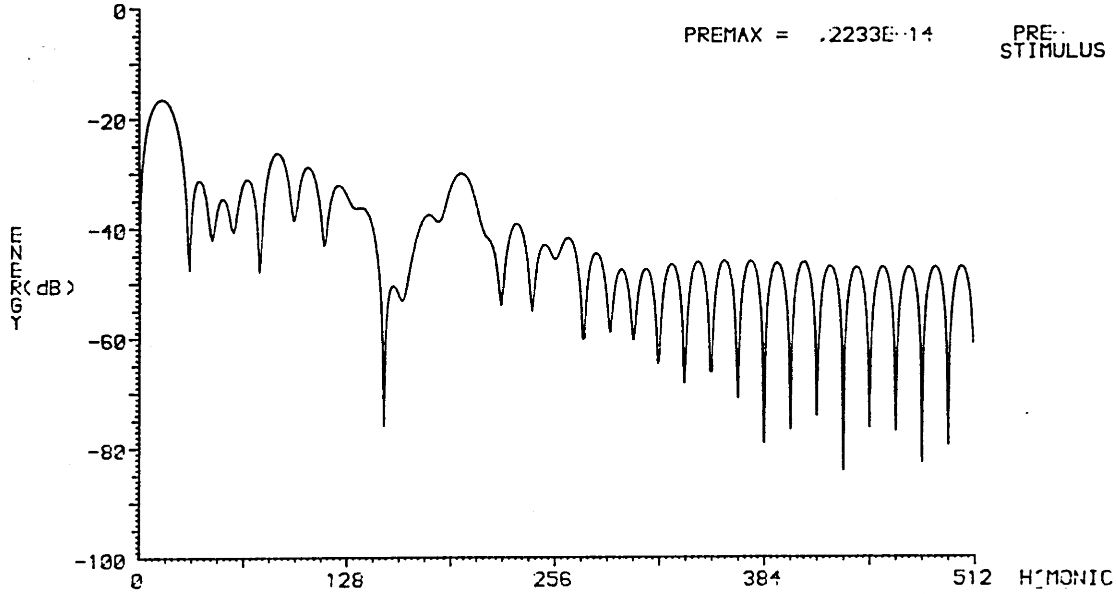

Figure 3.10

Energy spectrum when the dc level due to windowing is subtracted from the windowed data

To overcome this JERVIS et al (Submitted for publication) proposed the following mean level correction:

$$s_i = w_i(x_i' - k_1)k_2 \qquad -(3.27)$$

where: s_i = the transformed data sequence

w_i = the window sequence

x_i' = the original data sequence with its mean removed

k_1 = a correction constant to compensate for the d.c. level introduced due to windowing

k_2 = a correction constant to restore the energy content of s_i to the value of the pre-windowed data (it will have been reduced due to tapering)

and:

$$k_1 = \frac{\sum_{i=1}^{N} x_i' w_i}{\sum_{i=1}^{N} w_i} \qquad -(3.28)$$

$$k_2 = \sqrt{\frac{N}{\sum_{i=1}^{N} w_i^2}} \qquad -(3.29)$$

Use of such a correction gave Figure 3.11 where the d.c. level and the sidelobe effects have been removed.

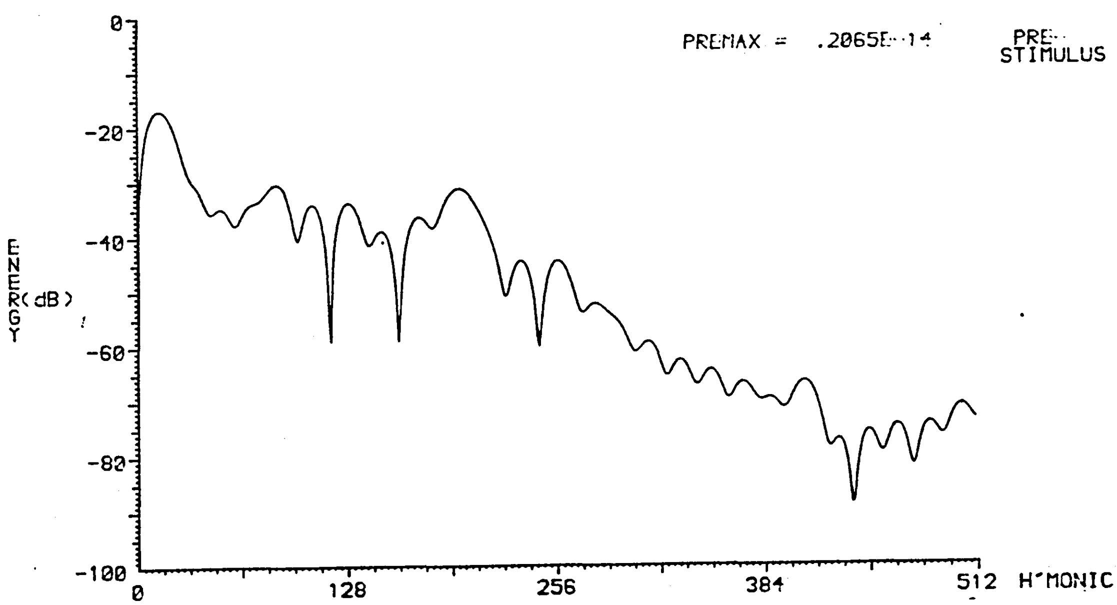

Figure 3.11

Energy spectrum when the transformation of equation (3.27) is used

3.5 INVESTIGATION OF SIGNAL PROCESSING CONCEPTS

To test a number of the above ideas an investigation of both real and simulated (test) data was carried out. The aims were to: compare the Tukey and Kaiser-Bessel windows and to observe the practical effects of spectral leakage.

3.5.1 TEST DATA

3.5.1.1 DESCRIPTION OF THE TEST DATA

The test data comprised two superimposed sinusoidal signals close in frequency and of large amplitude difference. Two simulated data sequences were generated: one a short duration signal comprising 64 points (to which 960 augmenting zeroes were added) the other a 1024 point data sequence. Each sequence had two representations: one where both signals contained an integer number of cycles (the 'transparent' sequence), the second where the larger amplitude sinewave was adjusted in frequency so that it no longer contained an integer number of cycles (the 'opaque' sequence). Thus in what follows reference is made to 64 and 1024 point transparent and opaque data.

3.5.1.2 APPLICATION AND RESULTS

The test data was studied in two stages: the first used 64 point transparent and opaque data (with 960 augmenting zeroes) while the second used 1024 point data (with no augmenting zeroes).

The 64 point (representing 0.512 msec) transparent data comprised two sinusoids of frequencies 7.8125 Hz (for the larger amplitude signal) and 9.765625 Hz (for the smaller amplitude signal). These frequencies were integer multiples of 1/(NT) = 1.953125 Hz. The two signals had an energy difference of 20dB (a factor of x10). Figure 3.12 shows the spectrum of the transparent sequence after windowing with a rectangular window followed by the addition of 960 augmenting zeroes.

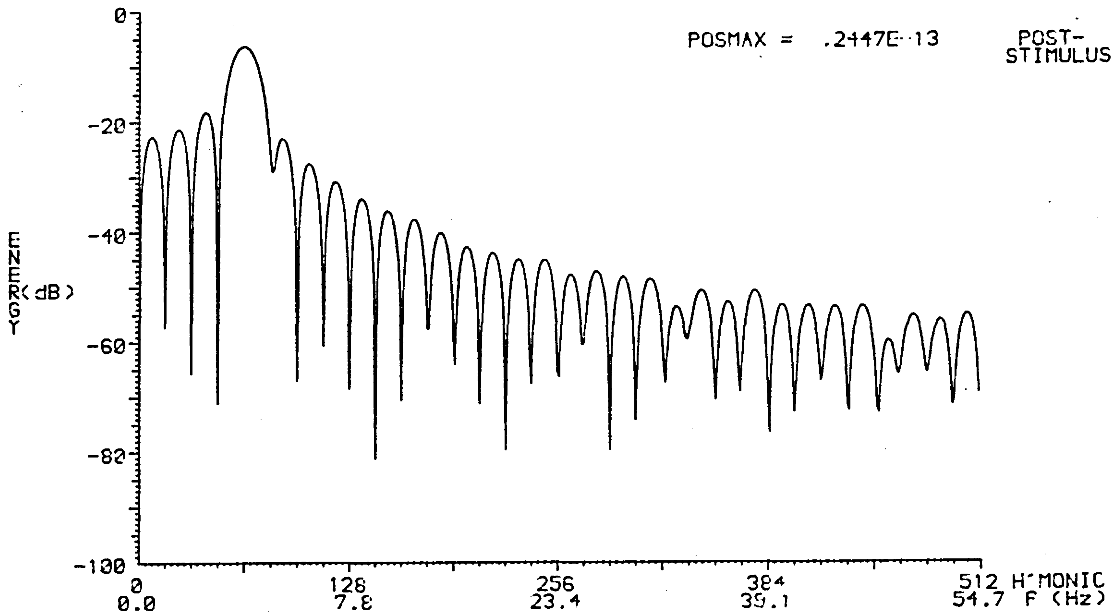

Figure 3.12

Energy spectrum for two sinewaves each harmonically related to 1/(NT) obtained using a rectangular window. 64 data points, 1024 point FFT.

Even with transparent data the sidelobe structure masks the presence of the smaller frequency component. The larger sinewave was now adjusted in frequency to no longer have an integer number of cycles in the data's duration. The use of a rectangular window gave Figure 3.13.

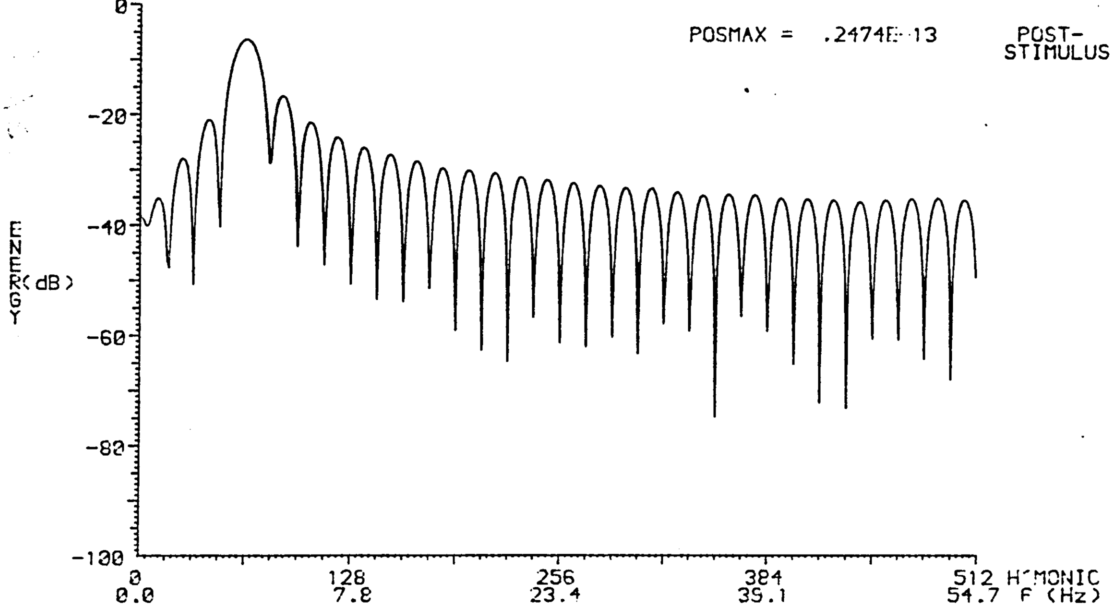

Figure 3.13

Energy spectrum for two sinewaves not harmonically related to 1/(NT) obtained using a rectangular window. 64 data points, 1024 point FFT.

Again there is no indication of the second peak due to the sidelobe structure. A Tukey window with 12.5% taper at each end of the window ($\alpha = 0.25$) was applied to give Figure 3.14.

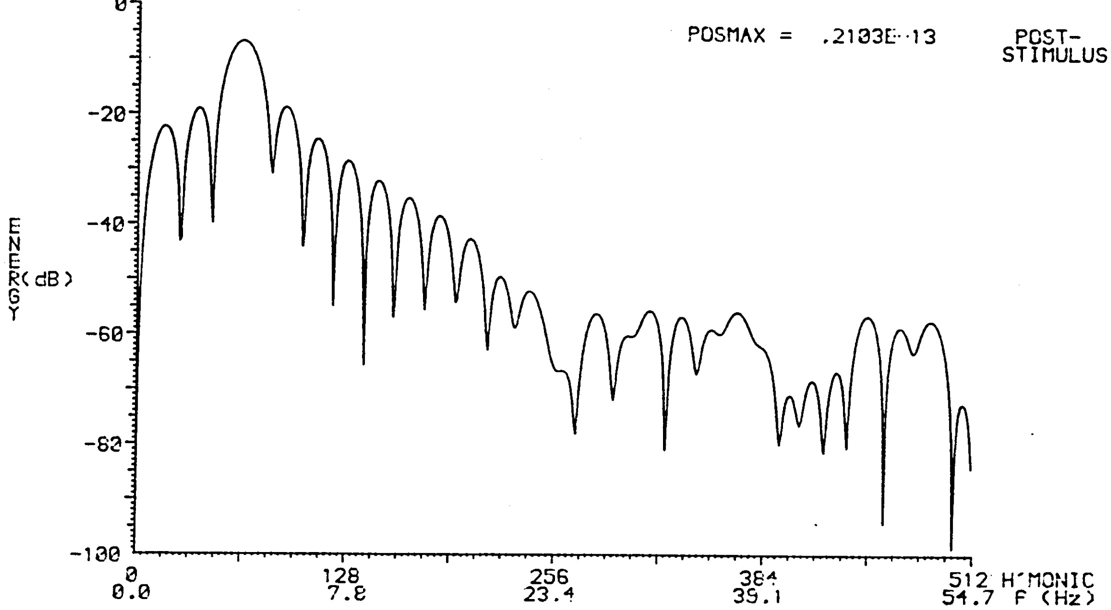

Figure 3.14

Energy spectrum for two sinewaves obtained using a Tukey window with a 12.5% taper. 64 data points, 1024 point FFT.

Although the sidelobes are reduced the second peak cannot be seen. Three examples of the Kaiser-Bessel window with α = 0.5, 0.75 and 1.0 were used to give Figures 3.15, 3.16 and 3.17. It is seen that for this data careful selection of α is necessary to obtain the best result from the trade-off between decreasing sidelobe level and increasing mainlobe width (hence resolution) whilst increasing α. Here α = 0.75 or 1.0 picks out the second peak. The value of 0.75 is to be preferred because of the smaller mainlobe width.

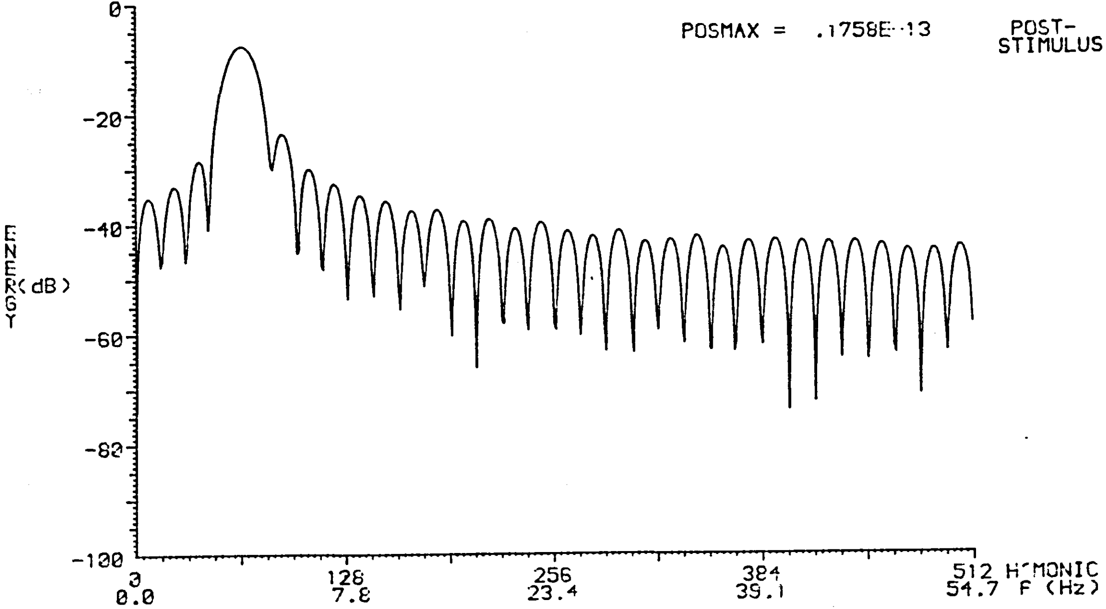

Figure 3.15

Energy spectrum for two sinewaves obtained using a Kaiser-Bessel window with α = 0.5. 64 data points, 1024 point FFT.

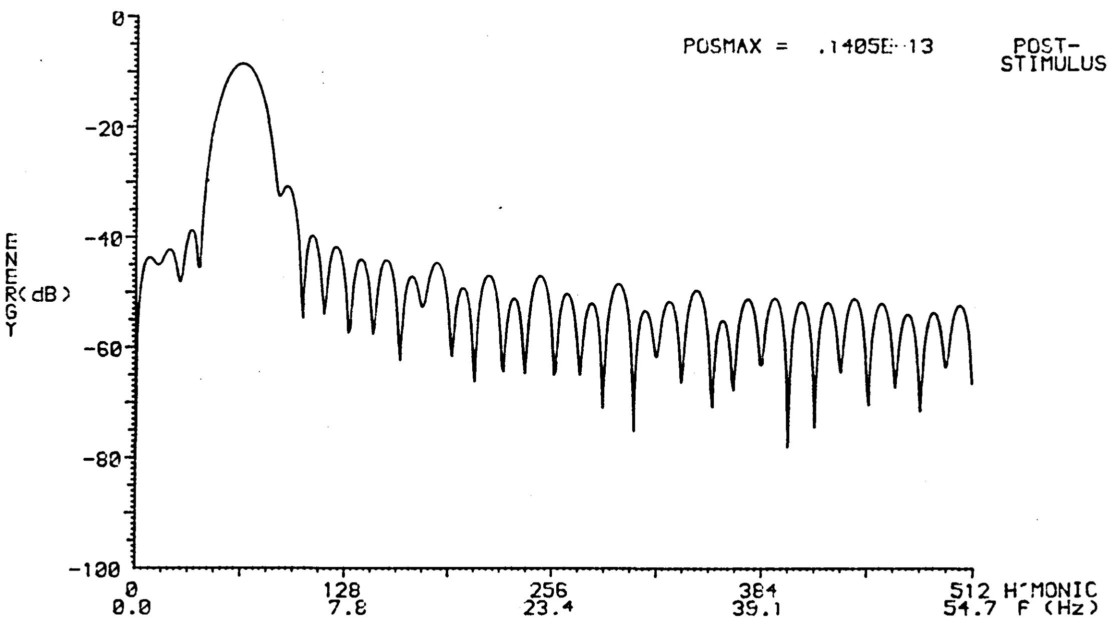

Figure 3.16

Energy spectrum for two sinewaves obtained using a Kaiser-Bessel window with α = 0.75. 64 data points, 1024 point FFT.

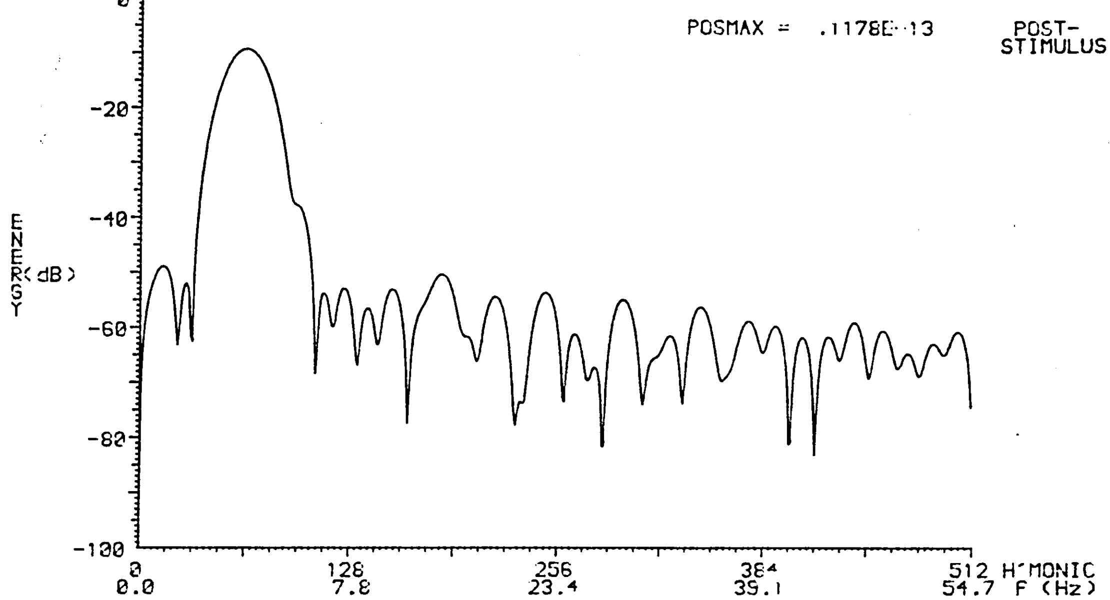

Figure 3.17

Energy spectrum for two sinewaves obtained using a Kaiser-Bessel window with α = 1.0. 64 data points, 1024 point FFT.

Having seen the effect of three different windows the discussion of the signal duration limitations will be illustrated. This was done by performing the above operations once again but now using a data sequence of 1024 points (no augmenting zeroes) representing 10 seconds duration. In addition the larger amplitude sinewave was made 40dB greater than the second signal (a factor of x100).

The transparent representation of the data used signal frequencies of 10 Hz and 12 Hz (the fundamental frequency was 1/(NT) = 0.1 Hz). The spectrum resulting from a rectangular window is shown in Figure 3.18. The two components are clearly seen.

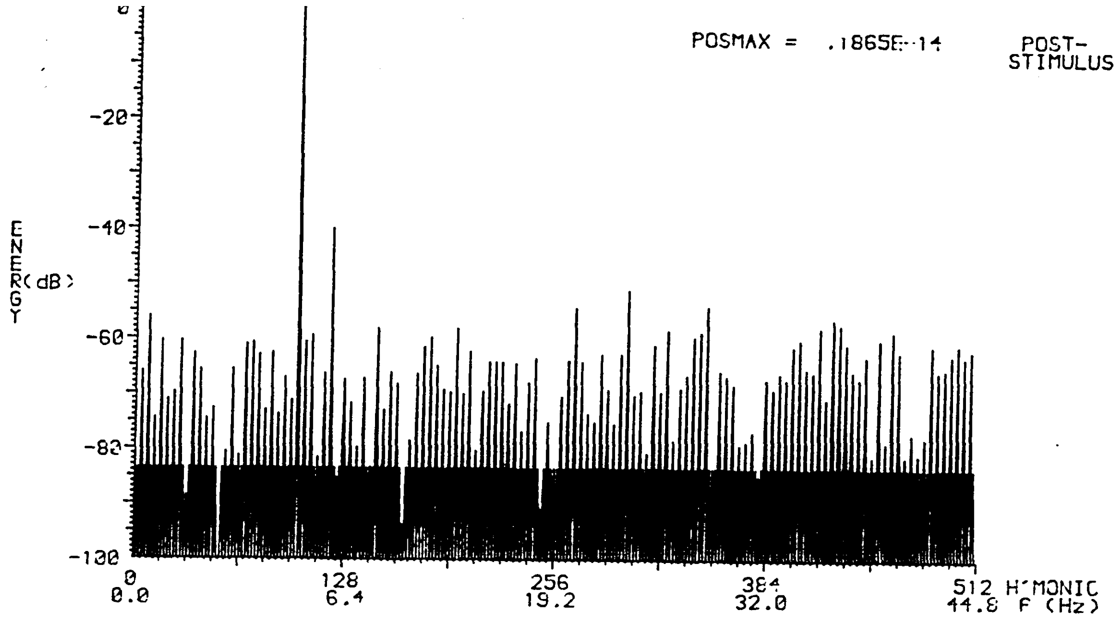

Figure 3.18

Energy spectrum for two sinewaves each harmonically related to 1/(NT) obtained using a rectangular window. 1024 data points

The second peak disappears in the sidelobe structure when the opaque data sequence's spectrum is obtained (Figure 3.19).

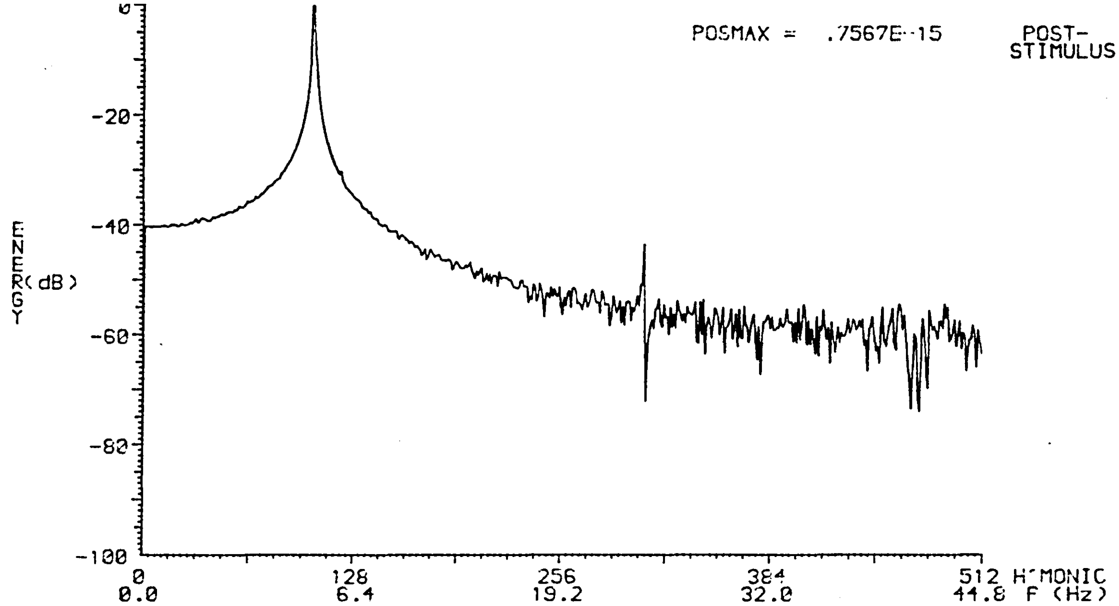

Figure 3.19

Energy spectrum for two sinewaves not harmonically related to 1/(NT) obtained using a rectangular window. 1024 data points

Applying a Tukey window with $\alpha = 0.25$ reveals the second peak again (Figure 3.20).

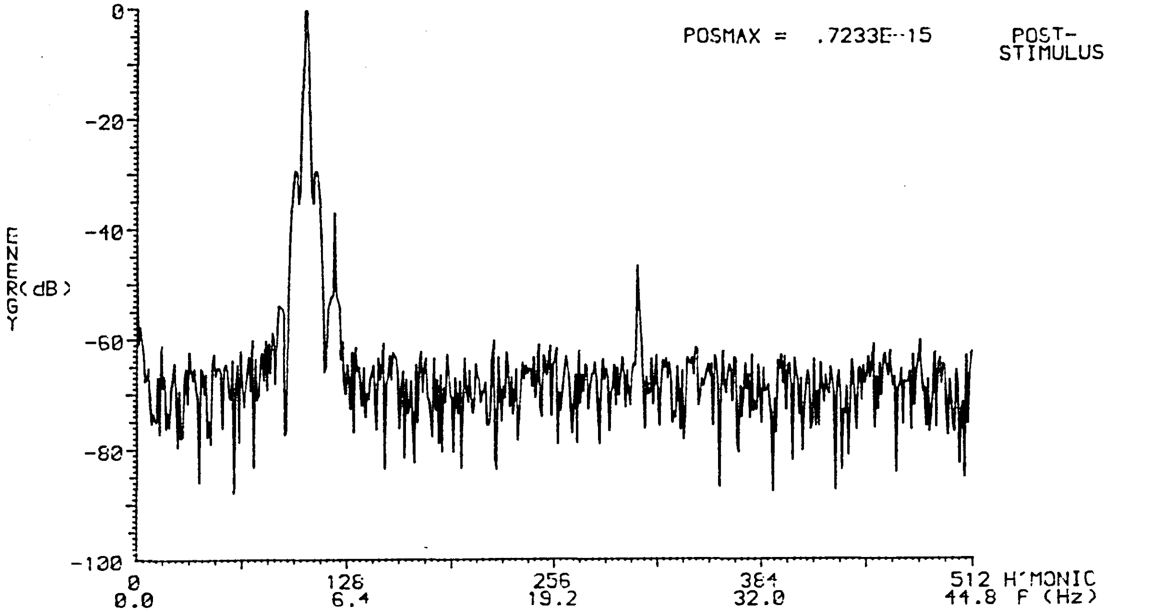

Figure 3.20

Energy spectrum for two sinewaves obtained using a Tukey window with a 12.5% taper. 1024 data points

Figures 3.21, 3.22, 3.23 and 3.24 show the spectra obtained using Kaiser-Bessel windows in which $\alpha = 0.5,\ 0.75,\ 1.0$ and 2.0. As for the 64 data point sequence the mainlobe width increases and the sidelobe levels decrease with increasing α.

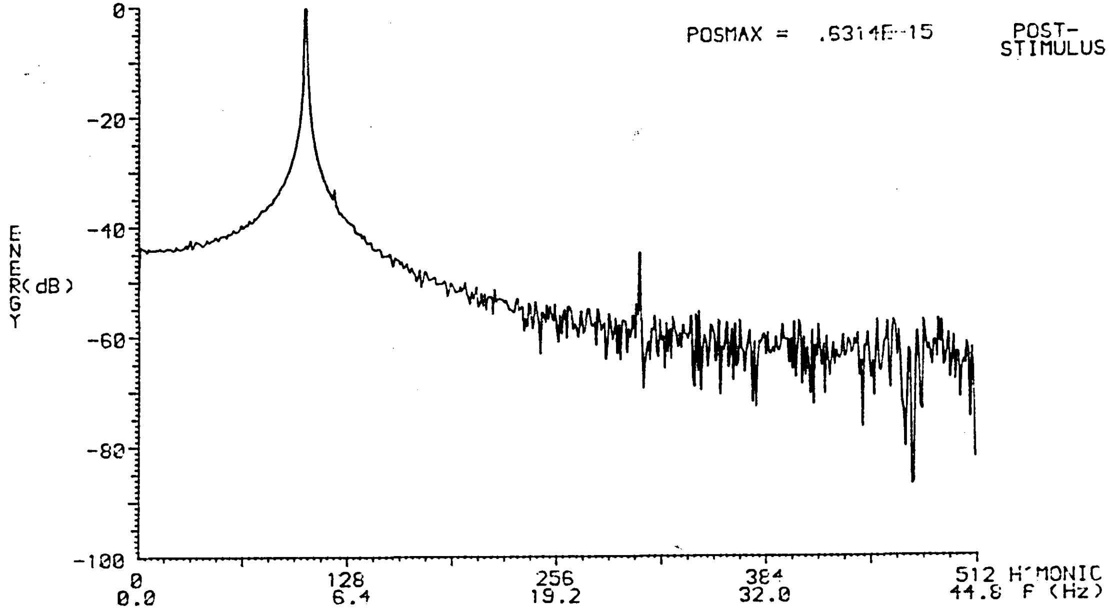

Figure 3.21

Energy spectrum for two sinewaves obtained using a Kaiser-Bessel window with $\alpha = 0.5$. 1024 data points

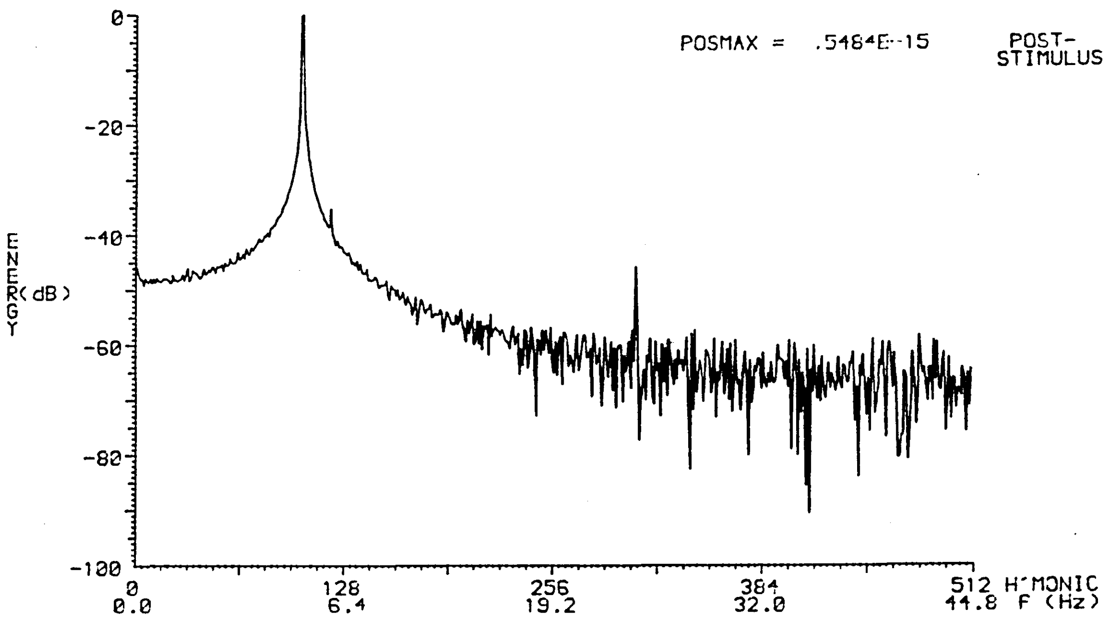

Figure 3.22

Energy spectrum for two sinewaves obtained using a Kaiser-Bessel window with $\alpha = 0.75$. 1024 data points

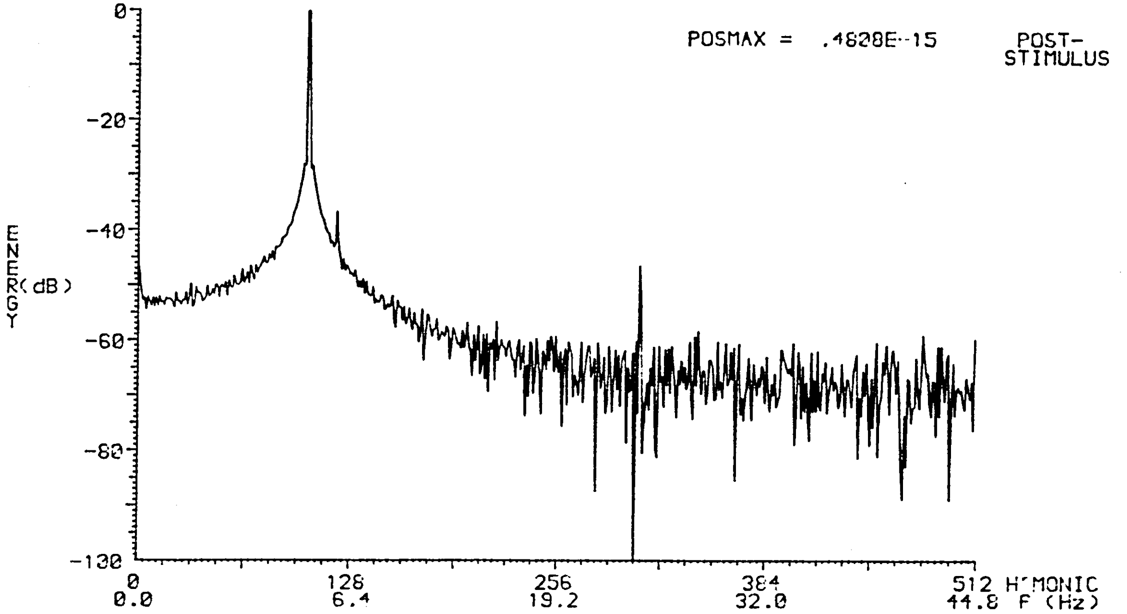

Figure 3.23

Energy spectrum for two sinewaves obtained using a Kaiser-Bessel window with $\alpha = 1.0$. 1024 data points

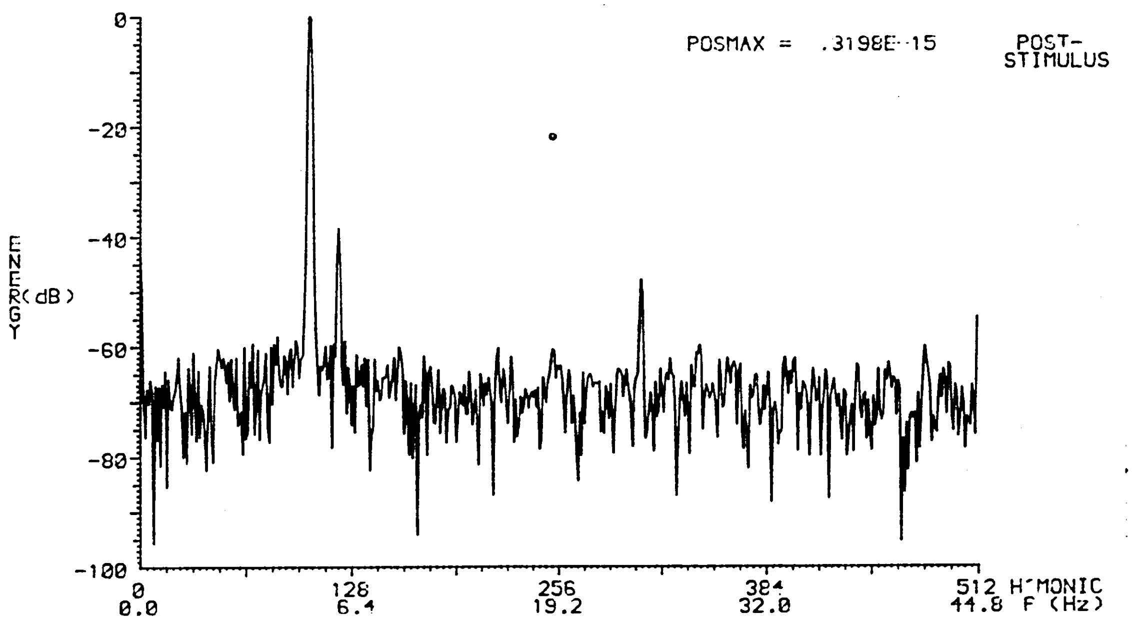

Figure 3.24

Energy spectrum for two sinewaves obtained using a Kaiser-Bessel window with $\alpha = 2.0$. 1024 data points

The value of α = 0.5 only just shows the second peak's presence. With α = 0.75 an improvement is discerned and when α = 1.0 the second peak can be positively identified. The clearest disclosure of this peak comes with α = 2.0. The best display of the second peak comes with the Tukey and Kaiser-Bessel windows (when the latter has α = 1.0 or 2.0). Although comparable the Kaiser-Bessel window is to be preferred since the two signals show smaller sidelobe structures (and hence have better discrimination between close peaks).

3.5.1.3 DISCUSSION OF TEST DATA RESULTS

The above results for the 1024 point data have shown that the Kaiser-Bessel window gives improved performance when compared to the Tukey window. The results for the 64 point data sequence have demonstrated the problems arising due to the use of short epochs of data. The 64 point data was also a testing ground for applications to the real data processing to be used on 1 second CNVs, from which it was determined that a value of α = 0.75 should be used in the Kaiser-Bessel window. In addition investigations were carried out on 216 point data to simulate the 4 second CNV processing. From this a value of α = 1.25 was chosen for use with the 4 second data.

3.5.2 REAL DATA

In order to test the suggested processing procedure derived from the test data it was decided to investigate the effects when a Tukey window (with 12.5% taper at each end of the data) and a Kaiser-Bessel window (with $\alpha = 0.75$) were applied to 1 second CNV data. The data which was to be subject to windowing was that of the averaged waveforms (after each individual trial had been subject to non-recursive OAR without response modelling). The data transformation of equation (3.27) was used.

3.5.2.1 DESCRIPTION OF DATA

The data studied here comprised EEG records of 1024 points sampled at 125Hz (i.e., 8.192 second duration). Each record (trial) contained a 1 second ISI CNV. All the trials were then averaged and two epochs of the averaged EEG were studied (each of 64 point duration). The first segment immediately preceded S_1 while the second segment immediately preceded S_2.

3.5.2.2 RESULTS

Figures 3.25 and 3.26 show, respectively, the energy spectra of 32 trial averaged waveforms (where low pass filtering has been performed) for data which has had Tukey and Kaiser-Bessel data tapering.

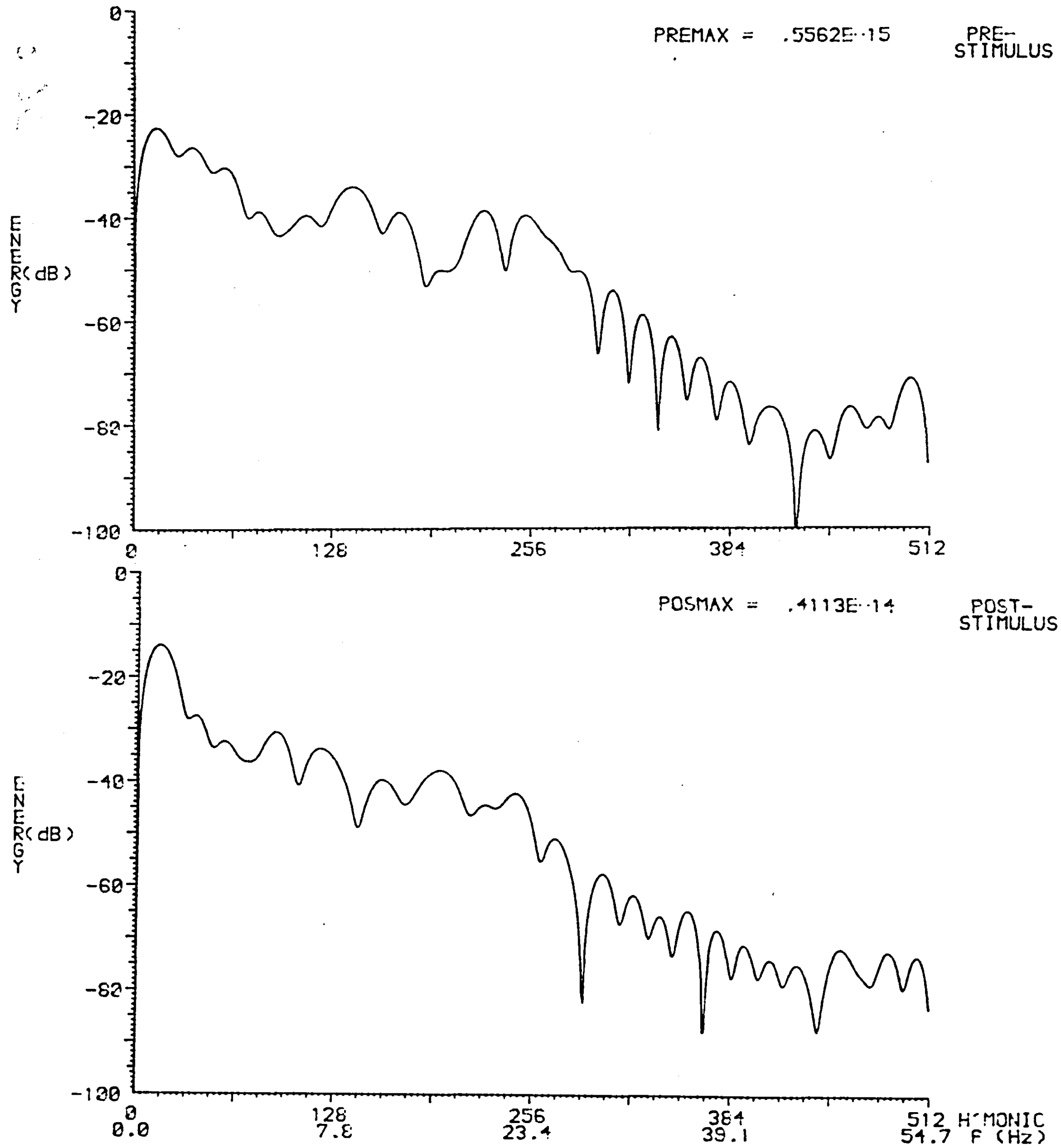

Figure 3.25

Energy spectrum of a 32 trial averaged CNV using a Tukey window with 12.5% taper.

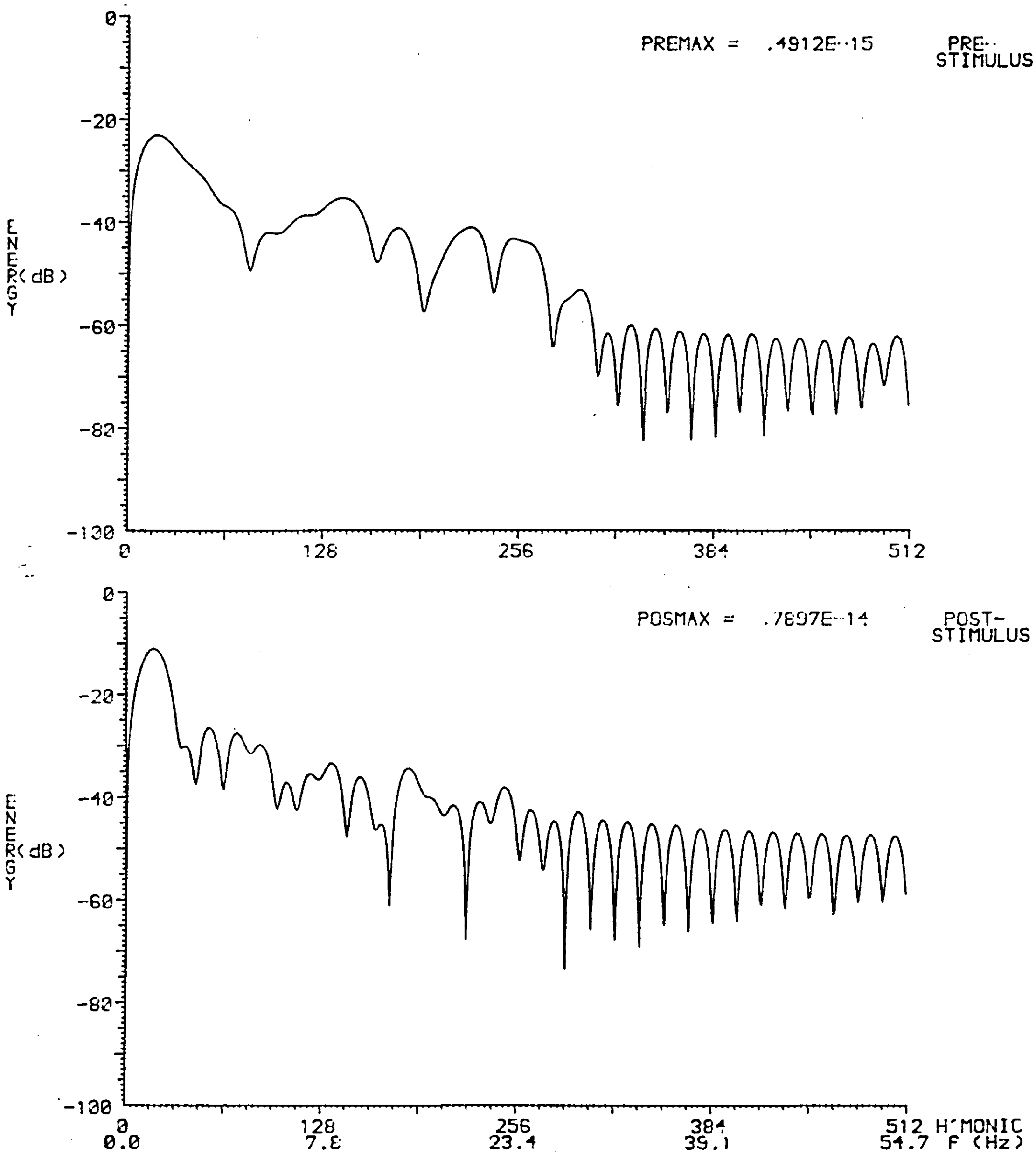

Figure 3.26

Energy spectrum of a 32 trial averaged CNV obtained using a Kaiser-Bessel window with $\alpha = 0.75$

Both show a low frequency peak and most of the energy is in the region up to harmonic 128. In general it is difficult to discern any major or consistent differences between the two figures. It is observed, however, that for high frequencies the KB sidelobe structure is higher than that of the Tukey windowed data (a result consistent with observations on the test data).

3.5.2.3 DISCUSSION OF REAL DATA RESULTS

The results when applying the Tukey and Kaiser-Bessel windows to real data proved inconclusive. This is not surprising since the true spectrum is unknown and on the basis of a simple visual analysis it would be difficult to obtain significant findings when considering short data epochs. However, on the basis of the results of Section 3.5.1 it is felt that the best choice is to use the Kaiser-Bessel window with $\alpha = 0.75$ (for 1 second CNVs) or $\alpha = 1.25$ (for 4 second CNVs).

3.6 CONCLUDING REMARKS

A variety of signal processing topics have been discussed in this chapter a number of which have been investigated practically. It has been seen that windowing of data is desirable in order to reduce spectral leakage, that such windowing should be accompanied by a data transformation given in equation (3.27). A comparison of Tukey and Kaiser-Bessel windows was made on 'short' and 'long' duration test data. Here it was seen, for the 'long' duration data, that the Kaiser-Bessel window gave better performance. When applied to the 'short' duration data a marginally better performance of the Kaiser-Bessel window, as compared to the Tukey window, was observed. Since the 'short' duration test data had been chosen to be of the same length as the real data to which it was to be applied, the value of α for the Kaiser-Bessel window which gave best performance was chosen to be used when processing real data. An example of application of both windows to real data was given. On the basis of the contents of this chapter it was decided to use the following processing steps:

(i) window the data and append augmenting zeroes;

(ii) use the Kaiser-Bessel window with α = 0.75 for 1 second CNVs and α = 1.25 for 4 second CNVs;

(iii) use the mean level correction of equation (3.27)

In investigating the CNVs the computation of energy and phase spectra were based upon data lengths of 64 points (1 second ISI) and 216 points (4 second ISI). S_1 and S_2 were delivered at points 407 and 532 (1 second ISI) and points 219 and 719 (4 second ISI). For the 1 second CNV the pre-stimulus data started at point 342 while the post-stimulus segment commenced at point 432. For the 4 second CNVs the pre-stimulus era began at point 22 and the post-stimulus epochs started at points 284 and 490.

REFERENCES

BELLANGER M. (1984)
"Digital Processing of Signals: Theory and Practice"
John Wiley and Sons, 1984

BERGLAND G.D. (1969)
"A Guided Tour of the Fast Fourier Transform"
IEEE Spectrum, July 1969:41-52

COOLEY J.W. and TUKEY J.W. (1965)
"An Algorithm for the Machine Calculation of Complex Fourier Series"
Math. of Comput. April 1965,19:297-301

COOLEY J.W., LEWIS P.A.W. and WELCH P.D. (1969)
"The Finite Fourier Transform"
IEEE Trans. Audio and Electroacoustics, June 1969, AU-17(2):77-85

DURRANI T.S. and NIGHTINGALE J.M. (1972)
"Data Windows for Digital Spectral Analysis"
Proc. IEE March 1972, 119(3):343-352

HARRIS F.J.(1978)
"On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform "
Proc. IEEE 1978,66(1):51-83

JERVIS B.W., COELHO M.A., MORGAN G.W. and ALLEN E.M.
(Submitted for publication)
"The Spectral Analysis of EEG Responses"

LYNN P.A. (1982)
"An Introduction to the Analysis and Processing of Signals"
2nd ed., MacMillan, 1982

NICHOLS M.J. (1982)
"An Investigation of the Contingent Negative Variation Using Signal Processing Methods"
Ph.D. Thesis, Plymouth Polytechnic, August, 1982

OTNES R.K. and ENOCHSON L. (1972)
"Digital Time Series Analysis"
John Wiley and Sons, 1972

ROBINSON E.A. (1978)
"Multichannel Time Series Analysis with Digital Computer Programs", rev. ed. p.63
Holden-Day, 1978

STARK H. and TUTEUR F.B. (1979)
"Modern Electrical Communications: Theory and Systems"
Prentice-Hall International, 1979

STREMLER F.G. (1977)
"Introduction to Communication Systems"
Addison-Wesley, 1977

4 STATISTICAL METHODS

4.1 CHAPTER OUTLINE

This chapter describes the statistical tests and methods used in analysing the data produced by application of the signal processing methods of Chapters 2 and 3. The results themselves are presented in Chapter 5.

Section 4.2 gives brief details of the statistical tests applied to the amplitude and phase spectra of each segment of the EEG data subsequent to OAR. Section 4.3 discusses Discriminant Analysis (DA) while Section 4.4 covers Predictive Statistical Diagnosis (PrSD). Section 4.5 describes how the above techniques were implemented and Section 4.6 concludes the chapter by indicating how the results so generated were to be interpreted.

4.2 STATISTICAL TESTS

Four tests were applied to the spectra (two to the amplitude spectrum, two to the phase spectrum). These tests were applied by NICHOLS (1982) and JERVIS et al (1983 and 1984) to AEPs and CNVs in an investigation of additive and ordering effects in EEG responses. This involved testing harmonics of the amplitude spectrum of the response (AEP or CNV) for an increased amplitude in the direction of the preferred phase angle (the Nearest and Furthest Mean Amplitude Test) and for an increased amplitude in the harmonics of the response compared with those of the pre-stimulus EEG (Pre- and Post- Stimulus Mean Amplitude Differences Test). The harmonics of the phase spectrum were tested for phase ordering by the Rayleigh Test of Circular Variance and the Modified Rayleigh Test of Circular Variance. Full details of the tests are given in NICHOLS (1982) and JERVIS et al (1983).

4.2.1 NEAREST AND FURTHEST MEAN AMPLITUDE TEST

This test investigates the variation of amplitude with phase angle of a particular post-stimulus harmonic. The mean length of that half of the vectors whose phase angles lay within the smallest arc was calculated as was that of the remaining vectors. A one-tailed t test (with a correction for the possibility of unequal variances) was then performed to determine whether the former mean value was greater than the latter.

4.2.2 PRE- AND POST-STIMULUS MEAN AMPLITUDE DIFFERENCES TEST

In this test the mean of differences between pre- and post-stimulus amplitudes for a given harmonic were calculated. A two-tailed t test was then applied, with a correction for possible unequal variances.

4.2.3 RAYLEIGH TEST OF CIRCULAR VARIANCE

This test is used to determine whether a set of phase angles are distributed in a non-uniform manner. Consider a set of N phase angles Θ_i. The circular variance is computed as:

$$S_o = 1 - \bar{R} \qquad - (4.1)$$

where:

$$\bar{R} = \sqrt{\bar{C}^2 + \bar{S}^2} \qquad - (4.2)$$

and $$\bar{C} = \frac{1}{N} \sum_{i=1}^{N} \cos \Theta_i \qquad - (4.3)$$

$$\bar{S} = \frac{1}{N} \sum_{i=1}^{N} \sin \Theta_i \qquad - (4.4)$$

The standard test is usually applied to the statistic $\bar{R}$. For uniformly distributed phase angles $S_o = 1$ ($\bar{R} = 0$) while a set of identical angles has $S_o = 0$ ($\bar{R} = 1$). Tabulated values of S_o are given in NICHOLS (1982).

4.2.4 MODIFIED RAYLEIGH TEST OF CIRCULAR VARIANCE

This test aims to take both the amplitude and phase angle into account. The statistic, U_o, is computed after ranking of the vectors by magnitude. If there are N phase angles θ_i, U_o is given by:

$$U_o = 1 - \sqrt{\left[\frac{\sum_{i=1}^{N} R_i \cos\theta_i}{\sum_{i=1}^{N} R_i}\right]^2 + \left[\frac{\sum_{i=1}^{N} R_i \sin\theta_i}{\sum_{i=1}^{N} R_i}\right]^2} \qquad - (4.5)$$

The significance levels for U_o are obtained from those of the test statistic R^* by:

$$U_o = 1 - \frac{2\sqrt{N}}{N+1} R^* \qquad - (4.6)$$

Tabulated values of U_o are given in NICHOLS (1982).

4.3 DISCRIMINANT ANALYSIS

In this work the use of DA is an intermediate step (as implemented in the Statistical Analysis System (SAS) package) towards using PrSD. It can also be used as a diagnostic method in its own right (COOPER and WEEKES, 1983) and as such was used to give results for comparison with those of PrSD in Chapter 5.

An introduction to DA is given in Chapter 12 of COOPER and WEEKES (1983) with a more advanced treatment of the linear discriminant function in Chapter 6 of MORRISON (1976). DA was effected using the SAS package and details of the appropriate procedures used will be given in Section 4.5.

DA is concerned with distinguishing between two (or more) groups of individuals on the basis of a common set of variable values for each of the individuals in each group, i.e., it determines how well the groups can be discriminated by the statistical data itself. Of course, the ability to allocate individuals of known groups into those groups is , in its own right, not very useful. However, DA offers a method of classifying individuals of unknown group (type) into one amongst two (or more) known groups. This method requires that DA is applied to those individuals of known classification to yield a set of measures. Against these the variable values of any individual of unknown type can be compared to determine into which group the individual should be classified. As a check on the effectiveness of the DA it can be applied to the individuals of known classification to see if any misclassification arises (the assignment of an individual of known type to the wrong group). The fewer the number of misclassifications the better the discrimination. Since the work described in this thesis refers to data comprising two known types further discussion of DA shall consider two groups but, in general, the method can be applied to more than two types.

DA uses the data obtained for the two known groups to determine a boundary which 'best' divides the two groups of data. This boundary can be expressed mathematically as a discriminant function. If a straight line boundary is considered (curved boundaries are possible) then it is specified by the linear discriminant function, which, for a p-dimensional multivariate data set, takes the general form:

$$f(X) = a_1X_1 + a_2X_2 + \ldots + a_pX_p + c \qquad -(4.7)$$

where X_1, ..., X_p are the p variables and a_1, ..., a_p and c are computed from the data. Substitution of values of X_1, ..., X_p for a particular individual yields a value of the function in equation (4.7) known as the discriminant score.

The criteria for 'best' discriminant boundary and hence discriminant function can now be given. The values of a_1, ..., a_p are chosen so as to minimise the variance of the discriminant scores for each group while maximising the difference between the average discriminant scores of the two groups. The 'c' in equation (4.7) is selected to cause the discriminant boundary to pass through the mid-group centroid (the point midway between the centroids of the two groups).

Once the discriminant function has been determined the test data (the data of individuals of known type is called the calibration data set) can be substituted in the function to yield discriminant scores for this data. These then indicate to which of the known types the unknown individual belongs according to the statistical data.

DA as implemented in SAS does not give values of the discriminant scores but computes two values of probability (the sum of which is unity) which show the probability of belonging to one or other group.

Prior to the use of DA stepwise discriminant analysis was performed (again using a SAS procedure) to select a reduced number of the 56 variables available for each subject (14 harmonics x 4 test statistics). In fact 98 variables were produced, but 42 of these were not of interest as they were for the pre-stimulus data only. The aim of this was to identify those variables which best discriminated between the data for the two known groups.

4.4 PREDICTIVE STATISTICAL DIAGNOSIS

PrSD is concerned with assessing to which of a given set of possible types, t, an individual (or case) belongs on the basis of a vector $\underline{X}$ of observations on that individual. This assessment is most conveniently expressed as probabilities or plausibilities of the possible types (AITCHISON et al, 1977; JERVIS et al, 1985). As with DA, PrSD uses a calibration data set of observations on individuals of known type to compute certain values which are used together with the test data to classify the latter into types.

In addition it is recommended that an index of atypicality is computed for each type. Such indices provide a check on the computed probabilities. High values for all indices or for the index associated with the highest probability may indicate the individual is of an unknown type not included in the calibration set.

Let $p(\underline{X}|t_1,\underline{\theta})$ be the probability density function of $\underline{X}$ for a given type t_1 with parameter vector $\underline{\theta}$. $p(t_1)$ is the incidence rate (the probability of being type t_1 from previous results) and $p(t_1|\underline{X},\underline{\theta})$ is the probability that a case with observation vector $\underline{X}$ is assigned to type t_1. Then using Bayes' theorem:

$$p(t_1|\underline{X},\underline{\theta}) = \frac{p(t_1).p(\underline{X}|t_1,\underline{\theta})}{p(\underline{X})} \qquad -(4.8)$$

but $p(\underline{X})$ has a total probability expression given by:

$$p(\underline{X}) = \sum_{t=t_1}^{t=t_n} p(\underline{X}|t,\underline{\theta})p(t) \qquad -(4.9)$$

and so equation (4.8) becomes:

$$p(t_1|\underline{X},\underline{\theta}) = \frac{p(t_1)p(\underline{X}|t_1,\underline{\theta})}{\sum_{t=t_1}^{t=t_n} p(t)p(\underline{X}|t,\underline{\theta})} \qquad -(4.10)$$

Usually $\underline{\theta}$ is unknown, but the calibration data set $(\underline{Z})$ is and $p(\underline{X}|t,\underline{\theta})$ can be replaced by $q(\underline{X}|t,\underline{Z})$ (AITCHISON et al, 1977) and so equation (4.10) becomes:

$$p(t_1|\underline{X},\underline{\theta}) = \frac{p(t_1)q(\underline{X}|t_1,\underline{Z})}{\sum_{t=t_1}^{t=t_n} p(t)q(\underline{X}|t,\underline{Z})} \qquad -(4.11)$$

where:

$$q(\underline{X}|t,\underline{Z}) = \int_\theta p(\underline{X}|t,\underline{\theta})p(\underline{\theta}|\underline{Z})d\underline{\theta} \qquad -(4.12)$$

Equation (4.12) is known as the predictive density function for a "future" observation $\underline{X}$ on a case of type t assessed on the calibration data $\underline{Z}$ (AITCHISON et al, 1977). This may take a variety of forms (AITCHISON and DUNSMORE, 1975) but for the situation here AITCHISON et al (1977) replace the right hand side of equation (4.12) by:

$$q(\underline{X}|t,\underline{Z}) = St_d(v_t,\underline{m}_t,\{1+1/n_t\}\underline{S}_t) \qquad -(4.13)$$

where: there are n_t individuals of type t with observation vectors $X_1, X_2, \ldots, X_{n_t}$; v_t is the number of degrees of freedom ($= n_t - 1$); $\underline{m}_t$ is the matrix of means and $\underline{S}_t$ is the covariance matrix. St_d denotes a d-dimensional student-type density function defined by:

$$St_d(v,\underline{b},\underline{c}) = \frac{\Gamma[0.5(v+1)]}{\pi^{d/2}.\Gamma[0.5(v-d+1)].|v\underline{c}|^{1/2}} \cdot \frac{1}{[1+(\underline{X}-\underline{b})^T(v\underline{c})^{-1}(\underline{X}-\underline{b})]^{(v+1)/2}} \qquad -(4.14)$$

where Γ is the gamma function (see e.g. STEPHENSON, 1978). Thus using equations (4.14), (4.13) and (4.11) the required values of $p(t|\underline{X},\underline{\theta})$ can be computed for the individuals of known type. To compute the probabilities for the test data equation (4.14) uses the observation vector $\underline{X}$ for the individual of unknown type but retains the mean and covariance matrices for the known type.

The atypicality index for type t is given by AITCHISON et al (1977):

$$A(t) = \beta\ \{\tfrac{1}{2}d,\ \tfrac{1}{2}(n_t - d)\ ;\ \frac{w_t(\underline{X})}{w_t(\underline{X}) + (n_t^2 - 1)/n_t}\ \} \qquad - (4.15)$$

where: $w_t(\underline{X}) = (\underline{X} - \underline{m}_t)^T\ \underline{S}_t^{-1}(\underline{X} - \underline{m}_t)$

and β denotes the incomplete Beta function computed according to the algorithm of MAJUMDER and BHATTACHARJEE (1973).

In applying this technique the variables selected for diagnosis are assumed to have normal distributions and so prior to their inclusion this assumption was tested.

4.5 IMPLEMENTATION OF STATISTICAL METHODS

This section describes the implementation of both DA and PrSD. The work was performed using FORTRAN programs and the SAS (Statistical Analysis System) package on an IBM 4341 mainframe computer with a VM/CMS operating system. Four steps were involved: (i) the selection of the FFT harmonics for investigation; (ii) the compilation of test statistics from the values of the amplitude and phases for these harmonics; (iii) the use of various SAS procedures to select variables for analysis from the above information, (to produce summary data on these variables and to carry out DA); and (iv) the application of PrSD using the variables (of calibration and test data sets) and summary data (of the calibration data set only).

The following definitions are used: each test statistic for any FFT harmonic is called a variable, while the complete set of variable values for a subject is known as an observation; the observations on individuals of known type comprises the calibration data set and the test data set is made up of the observations of the individuals of unknown type.

The selection of the FFT harmonics for processing was made on the basis of the energy spectrum plots of the averaged waveforms (Chapter 3). Scrutiny of the phase spectrum plots gave no useful indicators as to harmonic selection. The energy spectrum for each subject of the two known groups (6 normal and 8 Huntington's Chorea subjects) were produced using the recommended processing options given in Chapters 2 and 3, viz., removal of the mean of the data of each trial from the data prior to application of non-recursive OAR (without response modelling) followed by use of the FFT with a Kaiser-Bessel window (α = 0.75 for

1 second ISI CNV and α = 1.25 for 4 second ISI CNVs) which was accompanied by the data transformation described in Chapter 3. The resulting energy spectra for each of the 14 subjects were then studied to determine the harmonic numbers corresponding to peak and second largest magnitude (i.e. 2 values for each subject) for : the pre-stimulus era (1 and 4 second CNVs), the post-stimulus era (1 second CNVs) and for two segments of the post-stimulus era (4 second CNVs). This gave rise to 140 values (2 harmonics x 14 subjects x 5 separate eras). The aim was to represent this information with the minimum number of harmonics by looking for that sub-set of harmonics which included as many of the observed values as possible and as a result 14 were chosen (harmonic numbers 4, 6, 8, 12, 16, 20, 24, 28, 32, 48, 64, 80, 96 and 112). The chosen harmonics have been adjusted to bring them into line with the nearest harmonic number which was a multiple of 4, apart from harmonic 6 (due to the large number of peak magnitude harmonics occurring here).

These selected harmonics were then subjected to the statistical tests described in Section 4.2 to produce the required test statistics. These test statistics were then processed by a number of SAS procedures in order to select a suitable sub-set of the 56 variables. This was necessary for three reasons: (i) to keep computational processing within manageable limits, (ii) to generate the necessary summary statistics needed for PrSD, and (iii) to ensure differences in the observed distributions between normal and Huntington's Chorea subjects since, as AITCHISON et al

(1977) note, the distribution of observation vectors must differ from type to type if statistical diagnosis is to be useful.

To select those variables which showed different distributions between the two types SAS procedure TTEST (SAS 1982a) was applied to all variables. This produced an F-statistic and two t-statistics (one for the case of equal variances, one for unequal variances), computed by comparing the HC and normal observations. These statistics were used to select the variables with significant differences (between HC and normal groups) of either variance (5% significance level for the F-statistic) and/or mean (10% significance level for the t-statistic). To check whether the selected variables were normally distributed within the two separate groups SAS procedure UNIVARIATE (SAS, 1982a) was used. This provided two measures as to normality of the distribution or otherwise: the Shapiro-Wilk statistic, W (SHAPIRO and WILK, 1965) and a normal probability plot. W takes the values $0 < W < 1$ with small values of W leading to rejection of the null hypothesis that the data are a random sample from the normal distribution. The normal probability plot comprises a graph on which the data are plotted together with a line indicating where data from a normal distribution should fall. Deviation between this line and plotted data indicates that the data are not normally distributed. Any variables which were found to be non-normally distributed were discarded.

The selected variables were then used in SAS procedure STEPDISC (SAS, 1982b). This performed stepwise discriminant analysis in order to identify that subset of these variables which gave best discrimination between HC and normal types. Between 8 and 16 variables were submitted to STEPDISC of which from 1 to 3 were selected by the procedure.

The variables selected by STEPDISC were then processed by the SAS procedure DISCRIM which performs DA as described in Section 4.3. The data supplied to this procedure comprised a calibration data set (the normal and HC subjects) and a test data set (the dementia and at risk subjects). DISCRIM computed the probabilities of being normal and HC (p(N) and p(HC) respectively) for each subject where $p(N) + p(HC) = 1$. For each of the two known subject groups the covariance matrix of the variables and also (for each variable) the means of each type were calculated for use in PrSD. The results of DA are given in Chapter 5.

PrSD computations were performed by a set of FORTRAN programs written for these tasks (see program DISC in Appendix A2). All the data required by the programs was entered into disc files to allow easy verification and avoid data entry errors, each file being read as required by the programs. The computational sequence is now described.

First, the software reads various data items: the number of variables (the "dimension" of student-type density function of equation (4.13)), the total number of subjects (i.e., of both known and unknown types), the number of subjects in each of the HC and normal subject groups, the prior probability of being normal (the p(t) of equations (4.9),(4.10) and (4.11) - assumed to be 0.5 since only two possible types are being considered, i.e., HC or normal) and (for each of the two types) the mean of each of the selected variables (the $\underline{m}_t$s of equations (4.13) and (4.15)). The following steps were then performed twice (once for each known type): the values of the elements of the covariance matrix were read; the items $[v_t(1+1/n_t)\underline{S}]^{-1}$ and $|v_t(1+1/n_t)\underline{S}|$ were computed by means of a matrix inversion subroutine, MINV, from the IBM SSP (Scientific Subroutine Package) library. Then for every subject (taken one at a time) the observation vector (i.e., the value of each variable for that subject) was read in by the software and the density function of equation (4.10) was computed (the FORTRAN intrinsic function GAMMA was used to calculate the values of the gamma function in equation (4.14)). The atypicality indices of being normal or HC (A(N) and A(HC) respectively) were computed by a program unit using the algorithm of MAJUMDER and BHATTACHARJEE (1973) which calculated the incomplete beta function. This was the final step in the computational process, the results of which are given in Chapter 5.

4.6 INTERPRETATION OF RESULTS OF DISCRIMINANT ANALYSIS AND PREDICTIVE STATISTICAL DIAGNOSIS

4.6.1 DISCRIMINANT ANALYSIS RESULTS

As has been described the DA technique as implemented in SAS produces two non-independent probabilities p(N) and p(HC) for each subject. Since p(N) + p(HC) = 1 it is only necessary to consider one of the values (the other being easily obtained from the above expression). Since the PrSD technique as implemented in Section 4.5 produces a value for p(N) this is the probability which will be considered in DA results.

In addition to the value of p(N) SAS indicates (for the subjects of known type) whether a subject has been misclassified by the DA procedure.* Thus when comparing results from different experimental conditions (e.g., 1 second and 4 second CNVs) the condition which gives fewer misclassifications of subjects of known type can be considered to give the better discrimination (COOPER and WEEKES, 1983). Since the closer the value of p(N) to 1 the more likely it is to belong to type normal and the closer the value of p(N) to 0 the more likely it is to belong to type Huntington's Chorea this also provides a means of checking the DA results. Turning to the classification of the subjects of unknown type it is apparent that they will be classified as normal if $p(N) > 0.5$ or Huntington's Chorea if $p(N) < 0.5$.

---

* indicated on SAS printouts by an asterisk.

One possible assessment of a value of p(N) is proposed here : determine the smallest value of p(N) for the known normal subjects ($p(N_{MIN}|N)$) and the largest value of p(N) for the known HC subjects ($p(N_{MAX}|HC)$), if for an AR or dementia subject $p(N) > p(N_{MIN}|N)$ assign to type normal or if $p(N_{MAX}|HC) > p(N)$ assign to type HC, else assign to unclassified type.

4.6.2 PREDICTIVE STATISTICAL DIAGNOSIS RESULTS

PrSD produces three values for each subject: p(N), A(N) and A(HC). When comparing between experimental conditions the assessment of which condition provides better discrimination can use the number of misclassifications criteria given for DA (see page 136).

Analysis of the results for a particular experimental condition are no longer simple above/below threshold comparisons since the atypicality indices have a role to play. These indices are a check against including a subject in a type on the basis of p(N) while in fact belonging to a type not included in the calibration data set e.g., a value of p(N) = 0.98 for a subject of unknown type would suggest membership of the normal subject group, however, a large value of A(N) = 0.95 might suggest that this subject is not typical of that group.

A guide to classification of individuals using the atypicality indices can be inferred from Section 11.4 of

AITCHISON and DUNSMORE (1975). The method consists of determining the maximum values of A(N) from the normal group and of A(HC) from the HC group. The atypicality indices of the test data subjects can then be compared to the maximum values of A(N) and A(HC) of the calibration set to determine to which of the two groups membership is indicated or whether membership of a group outside the two given groups is more likely (see Section 5.2.3.1 for details). Thus when classifying subjects from the test data set it is not sufficient to consider the atypicality indices on their own, they must be used in conjunction with the known values of A(N) and A(HC) of the calibration data set subjects.

REFERENCES

AITCHISON J. and DUNSMORE I.R. (1975)
"Statistical Prediction Analysis"
Cambridge University Press, 1975

AITCHISON J., HABBEMA J.D.F. and KAY J.W. (1977)
"A Critical Comparison of Two Methods of Statistical Discrimination"
Appl. Statist. 1977,26(1) : 15-25

COOPER R.A. and WEEKES A.J. (1983)
"Data Models and Statistical Analysis"
Philip Alan, 1983

JERVIS B.W., ALLEN E.M., JOHNSON T.E., NICHOLS M.J. and HUDSON N.R. (1983)
"A Fundamental Investigation of Auditory Evoked Potentials"
IEEE Trans. Biomed. Eng. Jan. 1983, BME-30(1) : 43-50

JERVIS B.W., ALLEN E.M., JOHNSON T.E., NICHOLS M.J. and HUDSON N.R. (1984)
"The Application of Pattern Recognition Techniques to the Contingent Negative Variation for the Differentiation of Subject Categories"
IEEE Trans. Biomed. Eng., Apr. 1984, BME-31(4): 342-349

JERVIS B.W., MORGAN G., ALLEN E.M. and HUDSON N.R. (1985)
"Diagnosis of CNV Data Quantifies Subject Classification"
7th Annual Conference Frontiers of Engineering and Computing in Health Care, IEEE Biomedical Engineering Society, Chicago, Sept. 1985

MAJUMDER K.L. and BHATTACHARJEE G.P. (1973)
"Incomplete Beta Integral"
Appl. Statist, 1973, 22(3) : 409-411

MORRISON D.F. (1976)
"Multivariate Statistical Methods" 2nd edition
McGraw-Hill Inc., 1976

NICHOLS M.J. (1982)
"An Investigation of the Contingent Negative Variation Using Signal Processing Methods" PhD Thesis, Plymouth Polytechnic, Aug., 1982

SAS (1982a)
"SAS User's Guide: Basics, 1982 edition"
SAS Institute Inc.

SAS (1982b)
"SAS User's Guide: Statistics, 1982 edition"
SAS Institute Inc.

SHAPIRO S.S. and WILK M.B. (1965)
"An Analysis of Variance Test for Normality (Complete Samples)"
Biometrika, 1965, 52(3 and 4) : 591-611

STEPHENSON G (1978)
"Mathematical Methods for Science Students" 2nd edition,
Longman Group Limited

5 RESULTS OF PREDICTIVE STATISTICAL DIAGNOSIS AND DISCRIMINANT ANALYSIS

5.1 CHAPTER OUTLINE

The results discussed in this chapter are those obtained when predictive statistical diagnosis (PrSD) and discriminant analysis (DA) were applied to the data, obtained using the signal processing techniques of Chapters 2 and 3, and the statistical techniques described in Chapter 4. Thus prior to the application of DA or PrSD the data had been prepared as follows:

(i) the mean level of the data of each trial was subtracted from the data for that trial;

(ii) OAR was applied using the non-recursive method without response modelling;

(iii) the energy and phase spectra were obtained using a Kaiser-Bessel window (with $\alpha = 0.75$ for 1 second CNVs and $\alpha = 1.25$ for 4 second CNVs);

(iv) the statistical tests of Section 4.2 had been used to generate the test statistics for each of the given harmonics;

(v) the resulting test statistic variables had been reduced in number by selecting those which were normally distributed and gave best discrimination between normal and Huntington's Chorea subject groups (Section 4.5).

5.2 RESULTS

5.2.1 INTERMEDIATE RESULTS IN IMPLEMENTING PREDICTIVE STATISTICAL DIAGNOSIS AND DISCRIMINANT ANALYSIS

The relationship between the FFT harmonics selected, the statistical tests (of Section 4.2) and the SAS variables are given in Table 5.1

TABLE 5.1
CORRESPONDENCE BETWEEN HARMONIC NUMBER, STATISTICAL TEST AND SAS VARIABLE NUMBER

Harmonic Number	Nearest & Furthest Mean Amplitude Test		Pre- and Post-Stimulus Mean Amplitude Difference Test		Rayleigh Test of Circular Variance		Modified Rayleigh Test of Circular Variance	
4	1	43	-	44	2	45	3	46
6	4	47	-	48	5	49	6	50
8	7	51	-	52	8	53	9	54
12	10	55	-	56	11	57	12	58
16	13	59	-	60	14	61	15	62
20	16	63	-	64	17	65	18	66
24	19	67	-	68	20	69	21	70
28	22	71	-	72	23	73	24	74
32	25	75	-	76	26	77	27	78
48	28	79	-	80	29	81	30	82
64	31	83	-	84	32	85	33	86
80	34	87	-	88	35	89	36	90
96	37	91	-	92	38	93	39	94
112	40	95	-	96	41	97	42	98

Note: (i) each variable is known to the SAS package as Vn, where n = variable number, e.g., the variable number for harmonic 8 of the Nearest and Furthest Mean Amplitude Test (pre-stimulus) is V7

(ii) in each of the cells in this table the left hand figure refers to the pre-stimulus epoch, the right hand figure is for the post-stimulus epoch

(iii) the Pre- and Post-Stimulus Mean Amplitude Difference Test cells do not have pre-stimulus variables since the test is concerned with the difference between pre- and post-stimulus

(iv) for the 4s ISI CNVs two post-stimulus epochs exist of which the first has the variable numbers shown above while for the second epoch each variable is preceded by a '1', e.g. epoch 1 variable for harmonic 80 for Rayleigh Test of Circular Variance is V89 and the correspondong epoch 2 variable is V189.

The steps set out in Section 4.5 on selecting the variables to be candidates for DA (as far as t-statistic, F-statistic, W-statistic and normal probability plots are concerned) were used and the variables so chosen are given in Table 5.2 which also shows those variables determined to give best discrimination between normal and HC types.

TABLE 5.2
SELECTION OF 'BEST' DISCRIMINATORY VARIABLES BY STEPWISE DISCRIMINANT ANALYSIS (SDA)

ISI	Epoch	Outliers Included or Removed	Variables submitted to SDA	Variables Selected by SDA
1s	-	Included	V43-V50 V52-V54 V56 V60 V64 V84-V86	V46
1s	-	Removed	V43 V46 V47 V50 V58 V60 V62 V66 V84	V46
4s	1	Included	V47 V48 V52 V59 V62 V66 V76 V78 V80-V83	V47 V62 V76
4s	1	Removed	V47 V59 V61 V62 V66 V78 V80-V84	V47 V62 V84
4s	2	Included	V145-V149 V175 V176 V182	V148 V149 V175
4s	2	Removed	V144 V145 V148 V149 V152 V173 V174 V182 V194	V145 V148

5.2.2 DIAGNOSTIC RESULTS OF THE LOGIC ALGORITHM

For the purposes of comparison the logic algorithm (LA) method of JERVIS et al (1984) quoted in Table 5.3. is given:

TABLE 5.3
CLASSIFICATION OF SUBJECTS BY USE OF LOGIC ALGORITHM

Known Group	Subject	Classification
Normal	1	N or D
	2	N or D
	3	N or D
	4	N or D
	5	N or D
	6	N or D
Huntington's Chorea	7	HC
	8	HC
	9	HC or D
	10	HC
	11	HC
	12	HC or D
	13	HC
	14	HC
Dementia	15	N or D
	16	N or D
	17	HC or D
At risk	18	N or D
	19	N or D
	20	HC

Notes:
D denotes dementia
HC denotes Huntington's Chorea
N denotes normal

5.2.3 RESULTS OF APPLICATION OF PREDICTIVE STATISTICAL DIAGNOSIS

In the discussion that follows two sets of results are referred to: the full data set (Section 5.2.3.1) and an outliers-removed data set (Section 5.2.3.2). This arose due to the presence of outliers in the 4 second ISI CNV results. These took the form of a known normal subject classified as HC (for the epoch 2 data) and a known HC classified as normal (for the epoch 1 data). To investigate if it was

TABLE 5.4
RESULTS OF PREDICTIVE STATISTICAL DIAGNOSIS APPLIED TO FULL DATA SET

Subject Group	Subject	1s ISI CNV p(N)	A(N)	A(HC)	4s ISI CNV (epoch 1) p(N)	A(N)	A(HC)	4s ISI CNV (epoch 2) P(N)	A(N)	A(HC)
N	1	0.996	0.571	0.999	0.955	0.278	0.862	0.957	0.264	0.955
	2	0.990	0.097	0.995	0.941	0.263	0.832	0.991	0.369	0.987
	3	0.983	0.308	0.992	0.819	0.097	0.527	* 0.294	0.266	0.562
	4	0.786	0.806	0.958	0.898	0.278	0.770	0.971	0.263	0.965
	5	0.997	0.654	0.999	0.998	0.308	0.979	0.668	0.124	0.754
	6	0.994	0.111	0.996	0.664	0.273	0.482	0.830	0.201	0.874
HC	7	0.010	0.995	0.680	0.018	0.852	0.150	0.034	0.452	0.054
	8	0.010	0.993	0.472	* 0.795	0.175	0.561	0.289	0.147	0.447
	9	0.015	0.988	0.057	0.025	0.824	0.140	0.116	0.497	0.608
	10	0.427	0.904	0.898	0.213	0.719	0.543	0.007	0.828	0.343
	11	0.020	0.984	0.160	0.028	0.820	0.166	0.042	0.554	0.277
	12	0.013	0.989	0.179	0.009	0.941	0.512	0.009	0.689	0.064
	13	0.010	0.993	0.509	0.027	0.826	0.176	0.015	0.618	0.043
	14	0.042	0.974	0.482	0.086	0.697	0.183	0.068	0.640	0.554
D	15	0.967	0.503	0.987	0.964	0.022	0.785	0.495	0.643	0.898
	16	0.997	0.588	0.999	0.309	0.381	0.155	0.522	0.629	0.899
	17	0.016	0.987	0.012	0.157	0.620	0.245	0.916	0.383	0.948
AR	18	0.858	0.758	0.968	0.560	0.352	0.444	0.456	0.044	0.529
	19	0.997	0.754	0.999	0.998	0.303	0.979	0.043	0.571	0.314
	20	0.997	0.842	0.999	0.027	0.872	0.383	0.807	0.327	0.895

Notes:

N denotes normal
HC denotes Huntington's Chorea
D denotes dementia
AR denotes at risk
* denotes misclassification of subject of known type

possible to improve classification these two subjects were transferred from the known (calibration) data to the unknown (test) data to be classified. The resultant calibration data set is the outliers-removed data set.

PrSD computes three values: probability of being of type normal, p(N) and atypicality indices of membership of normal and Huntington's Chorea (HC) subject groups (A(N) and A(HC) respectively).

5.2.3.1 PREDICTIVE STATISTICAL DIAGNOSIS AND THE FULL DATA SET

The results for PrSD (p(N),A(N) and A(HC)) when applied to 1 second CNV data and to both epochs of 4 second CNV data are given in Table 5.4.

For the 1 second CNV data it is apparent that all known members of both types are correctly classified. Both epochs of the 4 second CNV data suffer misclassification of one subject of known type.

Following the recommendation of Section 4.6 regarding interpretation of atypicality indices the maximum values of A(N) for the normal group and A(HC) for the HC group are given in Table 5.5.

TABLE 5.5
MAXIMUM VALUES OF ATYPICALITY INDICES FOR FULL DATA SET

Experimental condition	N		HC	
	Value of largest A(N)	Subject	Value of largest A(HC)	Subject
1s CNV	0.806	4	0.898	10
4s CNV (epoch 1)	0.308	5	0.561	8
4s CNV (epoch 2)	0.369	2	0.608	9

Using the values given here (denoting the maximum value of A(N) for the normal group as $A(N_{MAX})$ and the maximum value of A(HC) for the HC group as $A(HC_{MAX})$) in association with the results in Table 5.4 the following can be noted:

(i) for the 1 second CNV all values of A(HC) > $A(HC_{MAX})$ for the normal subject group and all values of A(N) > $A(N_{MAX})$ for the HC subject group;

(ii) for the 4 second CNV (epoch 1) two values of A(HC) fall below $A(HC_{MAX})$ for the normal subject group while one value of A(N) falls below $A(N_{MAX})$ for the HC subject group;

(iii) for the 4 second CNV (epoch 2) 1 value of A(HC) < $A(HC_{MAX})$ for the normal subject group and one value of A(N) < $A(N_{MAX})$ for the HC group.

Comparing all three experimental conditions for the dementia and at risk groups it is seen that in terms of p(N) all six subjects suffer inconsistent classification on the basis of p(N). Table 5.6 shows the classification of the six subjects for each experimental condition. The classification was as follows:

(i) assign to type U if both $A(N) > A(N_{MAX})$ and $A(HC) > A(HC_{MAX})$;

(ii) assign to type HC if $p(N) < 0.5$, $A(N) > A(N_{MAX})$ and $A(HC) < A(HC_{MAX})$ and

(iii) assign to type N if $p(N) > 0.5$, $A(N) < A(N_{MAX})$ and $A(HC) > A(HC_{MAX})$.

Once again there is inconsistent classification across experimental conditions.

TABLE 5.6
CLASSIFICATION OF DEMENTIA AND AT RISK SUBJECTS WITH FULL DATA SET BY PrSD

Subject Group	Subject	1s CNV	4s CNV (epoch 1)	4s CNV (epoch 2)
D	15	N	N	U
	16	N	HC	U
	17	HC	HC	U
AR	18	N	N? a	U b
	19	N	N	HC
	20	N? a	HC	N

Notes:

N denotes normal type
HC denotes Huntington's Chorea type
U denotes a type outside of classification possibility since both $A(N) > A(N_{MAX})$ and $A(HC) > A(HC_{MAX})$

? denotes conflicting indications between p(N) and A(N) or A(HC), details as follows:

a $p(N) > 0.5$ but $A(N) > A(N_{MAX})$

b $p(N) < 0.5$ but $A(N) < A(N_{MAX})$ and $A(HC) < A(HC_{MAX})$

Returning to Table 5.4 it is observed that for every subject in the HC group A(N) is greater for the 1 second CNV than for either of the two 4 second CNV epochs. This indicates that the HC group values of the 1 second data are more atypical of normal type membership than those for either of the 4 second data epochs. For the normal group every subject has a value of A(HC) greater in the 1 second CNV data than epoch 1 of the 4 second CNV data and five subjects out of six have greater values of A(HC) in the 1 second CNV data than in epoch 2 of the 4 second CNV data. This indicates that normal group values of the 1 second CNV data are more atypical of HC type membership than those for either of the 4 second CNV epochs. These two results taken together suggest better discrimination between types in the 1 second data than either epoch of the 4 second data.

The above observations lead to the conclusion that more reliable discrimination is obtained with the data from the 1 second CNV data since:

(i) there are no misclassifications of subjects of known type in the 1s data;

(ii) the 1 second data achieves better discrimination between normal and HC types (as indicated by atypicality indices)

5.2.3.2 PREDICTIVE STATISTICAL DIAGNOSIS AND THE OUTLIERS-REMOVED DATA SET

The results for PrSD (p(N), A(N) and A(HC)) when applied to the three experimental conditions are given in Table 5.7.

TABLE 5.7

RESULTS OF PREDICTIVE STATISTICAL DIAGNOSIS APPLIED TO OUTLIERS-REMOVED DATA SET

Subject Group	Subject	1s ISI CNV			4s ISI CNV (epoch 1)			4s ISI CNV (epoch 2)		
		p(N)	A(N)	A(HC)	p(N)	A(N)	A(HC)	p(N)	A(N)	A(HC)
N	1	1.000	0.538	1.000	0.981	0.165	0.914	0.997	1.000	1.000
	2	0.999	0.431	0.999	0.975	0.123	0.890	0.981	0.999	1.000
	3	0.998	0.644	0.999	--	--	--	--	--	--
	4	--	--	--	0.979	0.199	0.917	0.995	1.000	1.000
	5	1.000	0.656	1.000	0.999	0.248	0.983	0.783	0.985	0.999
	6	1.000	0.165	1.000	0.625	0.178	0.501	0.976	0.998	1.000
HC	7	0.002	0.998	0.740	0.018	0.805	0.378	* 0.951	0.966	1.000
	8	0.002	0.997	0.438	--	--	--	--	--	--
	9	0.003	0.996	0.228	0.026	0.648	0.045	0.318	0.539	0.698
	10	--	--	--	0.007	0.882	0.471	* 0.937	0.749	0.988
	11	0.005	0.995	0.514	0.020	0.758	0.249	0.460	0.963	0.990
	12	0.002	0.997	0.040	0.029	0.761	0.376	0.065	0.861	0.719
	13	0.002	0.998	0.497	0.019	0.739	0.186	* 0.832	0.987	0.999
	14	0.017	0.992	0.809	0.029	0.736	0.289	* 0.974	0.998	1.000
O	3	--	--	--	* 0.071	0.853	0.789	* 0.042	0.807	0.421
	4	0.902	0.941	0.995	--	--	--	--	--	--
	8	--	--	--	0.213	0.696	0.707	* 0.980	0.998	1.000
	10	* 0.524	0.973	0.984	--	--	--	--	--	--
D	15	0.994	0.791	0.999	0.612	0.712	0.893	0.997	0.999	1.000
	16	1.000	0.563	1.000	0.315	0.370	0.393	0.993	1.000	1.000
	17	0.003	0.996	0.292	0.090	0.897	0.887	0.526	0.969	0.993
AR	18	0.947	0.922	0.996	0.800	0.612	0.900	0.983	0.999	1.000
	19	1.000	0.784	1.000	0.999	0.219	0.982	0.963	0.997	1.000
	20	1.000	0.880	1.000	0.187	0.499	0.385	0.596	0.972	0.995

Notes:

N denotes normal
HC denotes Huntington's Chorea
O denotes outlier
D denotes dementia
AR denotes at risk
* denotes misclassification of subject of known type

For both 1 second and 4 second (epoch 1) data all members of the calibration data set are correctly classified but one outlier is misclassified. The 4 second (epoch 2) data shows severe misclassification with 4 out of 7 HC subjects and both outliers incorrectly assigned.

The maximum values of A(N) and A(HC) are given in Table 5.8.

TABLE 5.8
MAXIMUM VALUES OF ATYPICALITY INDICES FOR OUTLIERS-REMOVED DATA SET

Experimental condition	N: Value of largest A(N)	N: Subject	HC: Value of largest A(HC)	HC: Subject
1s CNV	0.656	5	0.809	14
4s CNV (epoch 1)	0.248	5	0.471	10
4s CNV (epoch 2)	1.000	1 and 4	1.000	7 and 14

For both 1 second and 4 second (epoch 1) data all values of A(HC) > $A(HC_{MAX})$ for the normal subject group and all values of A(N) > $A(N_{MAX})$ for the HC subject group. It is not possible to comment on the 4 second (epoch 2) data as both $A(N_{MAX})$ and $A(HC_{MAX})$ are unity.

The classification of subjects 15 - 20 using the same rules as described in the previous section is given in Table 5.9.

TABLE 5.9
CLASSIFICATION OF DEMENTIA AND AT RISK SUBJECTS WITH OUTLIERS REMOVED DATA SET BY PrSD

Subject Group	Subject	1s CNV	4s CNV (epoch 1)	4s CNV (epoch 2)
D	15	U n	U n	x
	16	N	HC	x
	17	HC	U h	x
AR	18	U n	U n	x
	19	U n	N	x
	20	U n	HC	x

Notes:

- N denotes normal type
- HC denotes Huntington's Chorea type
- U denotes type outside classification possibility since both $A(N) > A(N_{MAX})$ and $A(HC) > A(HC_{MAX})$
- n $p(N) > 0.5$ but both $A(N) > A(N_{MAX})$ and $A(HC) > A(HC_{MAX})$
- h $p(N) < 0.5$ but both $A(N) > A(N_{MAX})$ and $A(HC) > A(HC_{MAX})$
- x unable to classify since $A(N_{MAX}) = 1.0000$ and $A(HC_{MAX}) = 1.0000$

In terms of p(N) subjects 15, 18 and 19 are consistently classified as normal across all three experimental conditions. Using the classification rule of the previous section two thirds of the 1 second data are unclassifiable into a known type and half of the 4 second (epoch 1) data cannot be assigned to either normal or HC groups. Since the values of $A(N_{MAX})$ and $A(HC_{MAX})$ for the 4 second (epoch 2) case are both unity no sensible classification can be attempted.

From Table 5.7 it is seen that (comparing the 1 second data to each of 4 second data epochs in turn, considering only those subjects where comparison is possible) for the normal subjects all four values of A(HC) are larger in the 1 second data than in the 4 second (epoch 1) data while for the six HC subjects all values of A(N) are larger in the 1 second data than the 4 second (epoch 1) data. Comparing the 1 second data with the 4 second (epoch 2) data the A(N) of two subjects are identical between experimental conditions while one value of A(N) is larger in the 1 second data than the 4 second (epoch 2) data and one smaller. These results suggest that the 1 second data gives better discrimination between types than the 4 second (epoch 1) data.

The above lead to the following conclusions:

(i) the 4 second (epoch 2) data cannot be used for successful classification;

(ii) the 1 second and the 4 second (epoch 1) data give comparable performance in terms of number of misclassifications of known subjects and the number of unclassifiable dementia and at risk subjects;

(iii) the 1 second data has better discrimination between normal and HC types when the values of atypicality indices are considered.

5.2.4 RESULTS OF APPLICATION OF DISCRIMINANT ANALYSIS

Results are given for both full and outliers-removed data sets. DA gave probabilities for each subject, p(N) and p(HC), of which p(N) alone is recorded since p(N) + p(HC) = 1.

5.2.4.1 DISCRIMINANT ANALYSIS AND THE FULL DATA SET

Table 5.10 shows the results of applying DA to the full data set.

TABLE 5.10
RESULTS OF DISCRIMINANT ANALYSIS APPLIED TO FULL DATA SET

Subject Group	Subject	1 second ISI CNV p(N)	4 second epoch 1 p(N)	ISI CNV epoch 2 p(N)
N	1	1.0000	0.9975	1.0000
	2	0.9999	0.9943	1.0000
	3	0.9994	0.8990	* 0.4301
	4	0.8880	0.9780	1.0000
	5	1.0000	1.0000	0.9128
	6	1.0000	0.7470	0.9923
HC	7	0.0000	0.0000	0.0233
	8	0.0000	* 0.8870	0.3984
	9	0.0002	0.0002	0.1817
	10	0.3718	0.0372	0.0000
	11	0.0006	0.0002	0.0203
	12	0.0001	0.0000	0.0011
	13	0.0000	0.0002	0.0040
	14	0.0039	0.0115	0.0266
D	15	0.9971	0.9948	0.9093
	16	1.0000	0.2778	0.9300
	17	0.0002	0.0513	0.9999
AR	18	0.9485	0.6051	0.6293
	19	1.0000	1.0000	0.0193
	20	1.0000	0.0000	0.9940

The 1 second data shows correct classification of all subjects of known type while both epochs of the 4 second data give incorrect assignment of 1 subject each and these are the same subjects erroneously classified by PrSD.

On the basis of the number of misclassifications of subjects of known type it is inferred that the 1 second data will give better classification of subjects of unknown type.

Use of a simple above/below threshold criterion (of $p(N) = 0.5$) to classify dementia and at risk subjects gave Table 5.11 while the rule proposed in Section 4.6 gave Table 5.12.

TABLE 5.11
CLASSIFICATION OF DEMENTIA AND AT RISK SUBJECTS WITH FULL DATA SET BY SIMPLE THRESHOLD VALUE

Subject Group	Subject	1s CNV	4s CNV (epoch 1)	4s CNV (epoch 2)
D	15	N	N	N
	16	N	HC	N
	17	HC	HC	N
AR	18	N	N	N
	19	N	N	HC
	20	N	HC	N

TABLE 5.12
CLASSIFICATION OF DEMENTIA AND AT RISK SUBJECTS FROM THE FULL DATA SET WITH RULE OF SECTION 4.6

	Critical Probabilities	1s CNV	4s CNV (epoch 1)	4s CNV (epoch 2)
	$p(N_{MIN} \mid N)$	0.888	0.747	0.9128
	$p(N_{MAX} \mid HC)$	0.3718	0.0372	0.3984
Subject Group	Subject			
D	15	N	N	U
	16	N	U	N
	17	HC	U	N
AR	18	N	U	U
	19	N	N	HC
	20	N	HC	N

5.2.4.2 DISCRIMINANT ANALYSIS AND THE OUTLIERS-REMOVED DATA SET

Table 5.13 gives the results for the outliers-removed data set when DA is applied.

TABLE 5.13
RESULTS OF DISCRIMINANT ANALYSIS APPLIED TO OUTLIERS-REMOVED DATA SET

Subject Group	Subject	1 second ISI CNV p(N)	4 second epoch (1) p(N)	ISI CNV epoch (2) p(N)
N	1	1.0000	1.0000	0.9999
	2	1.0000	0.9999	0.6709
	3	1.0000	--	--
	4	--	1.0000	0.9988
	5	1.0000	1.0000	* 0.4212
	6	1.0000	0.7714	0.9853
HC	7	0.0000	0.0000	0.3406
	8	0.0000	--	--
	9	0.0000	0.0004	0.0181
	10	--	0.0000	0.0046
	11	0.0000	0.0000	0.2438
	12	0.0000	0.0000	0.0612
	13	0.0000	0.0000	0.0362
	14	0.0000	0.0000	0.3012
O	3	--	* 0.0000	* 0.2074
	4	0.9979	--	--
	8	--	0.0145	* 0.8556
	10	0.3311	--	--
D	15	1.0000	0.7119	0.7319
	16	1.0000	0.2609	0.9996
	17	0.0000	0.0000	0.7549
AR	18	0.9998	0.9906	0.7671
	19	1.0000	1.0000	0.5382
	20	1.0000	0.0556	0.0593

All known subjects (including the two outliers) are successfully classified with the 1 second data. Epoch 1 of the 4 second data has one known subject (an outlier) mis-classified and epoch 2 of the 4 second data incorrectly assigns three known subjects. Comparing the number of known subject misclassifications given in Table 5.13 it is seen that the 1 second data is best in this respect (no misclassifications) and that the 4 second (epoch 1) data is better than the 4 second (epoch 2) data.

The classification of the dementia and at risk subjects when a simple above/below rule of $p(N) = 0.5$ and the rule of Section 4.6 were used are shown below in Tables 5.14 and 5.15 respectively.

TABLE 5.14
CLASSIFICATION OF DEMENTIA AND AT RISK SUBJECTS WITH OUTLIERS-REMOVED DATA SET BY SIMPLE THRESHOLD VALUE

Subject Group	Subject	1s CNV	4s CNV (epoch 1)	4s CNV (epoch 2)
D	15	N	N	N
	16	N	HC	N
	17	HC	HC	N
AR	18	N	N	N
	19	N	N	N
	20	N	HC	HC

TABLE 5.15
CLASSIFICATION OF DEMENTIA AND AT RISK SUBJECTS FROM THE OUTLIERS-REMOVED DATA SET BY DA WITH RULE OF SECTION 4.6

	Critical Probabilities	1s CNV	4s CNV (epoch 1)	4s CNV (epoch 2)
	$p(N_{MIN} \mid N)$	1.0000	0.7714	0.6709
	$p(N_{MAX} \mid HC)$	0.0000	0.0004	0.3406
Subject Group	Subject			
D	15	N	U	N
	16	N	U	N
	17	HC	HC	N
AR	18	U	N	N
	19	N	N	U
	20	N	U	HC

5.3 DISCUSSION

Evidence was given in Sections 5.2.3 and 5.2.4 that results obtained for both PrSD and DA from 1 second CNV data gave better discrimination than those results obtained from either epoch of the 4 second CNV data. Given this the question of whether the full or outliers-removed data results distinguish between types better is addressed.

For PrSD (Tables 5.4 and 5.7) the removal of the outliers gave higher values of p(N) for all normal subjects in the normal group and lower values of p(N) for all HC subjects in the HC group, which indicates better discrimination between the two types. However, values of A(N) became larger (worse) in four out of five cases in the normal group and four out of seven values of A(HC) were larger (worse) in the HC group when the outliers were removed, This indicates impaired type discrimination. Although the normal outlier (subject 4) has a large value of p(N) both it and the HC outlier (subject 10) suffer from larger values of their associated atypicality indices when outliers-removed data is compared to full data set results. Against this is set the observation that the two outliers have larger (better) atypicality index values relating to the group of which they are not a member, indicating that they are less typical of that group. The HC outlier is also misclassified on the basis of its p(N) value. In addition the two outliers would be assigned to an as yet unknown type on the basis of their atypicality indices.

In contrast the results of DA (Tables 5.10 and 5.13) give equal or larger values of p(N) for all six normal subjects when outliers have been removed than when included in the calibration data set. All values of p(N) for all eight HC subjects are improved (i.e., lowered) when outliers are removed from the calibration data set.

The comments lead to the conclusion that the PrSD discrimination is best performed when the calibration data set includes the outliers while DA discriminates better when the outliers are excluded from the calibration data.

The final issue concerns whether PrSD or DA is superior in discriminating between known types and classifying subjects of unknown type. Consider first Table 5.4 and 5.10. Both PrSD and DA successfully classify the known subjects, they both indicate the same two subjects as outliers (as measured by their values of p(N)) and both give the same classification of dementia and at risk subjects (Tables 5.6 and 5.11) for five out of six cases. The remaining case (subject 20) is assigned by DA to type normal. PrSD assigns the subject to type normal on the basis of high p(N) (= 0.997) but $A(N) > A(N_{MAX})$ which questions whether it belongs to this type. However, the difference $[A(HC) - A(HC_{MAX})] > [A(N) - A(N_{MAX})]$ indicating the subject is more atypical of type HC than of type normal.

Turning now to the outliers-removed data set results (Tables 5.7 and 5.13) PrSD and DA correctly classify the known normal and HC subjects of the calibration data set. DA also correctly assigns both outlier subjects using the simple above/below p(N) = 0.5 criteria but fails to assign either subject when the 'critical probabilities' measure is used (Tables 5.14 and 5.15). PrSD allocates one subject correctly and one incorrectly on the basis of p(N). Both are indicated as not belonging to either normal or HC types by their atypicality indices.

Considering the above points it is felt that the PrSD technique is to be preferred since it provides two additional scores with which to test the validity of type assignment on the basis of probability of being normal and provides consistent results (classifications) with DA. Also better discrimination is obtained by using the 1 rather than 4 second data.

5.4 CONCLUSIONS

This chapter has given and discussed the results of using PrSD and DA techniques to distinguish between two known groups (normal and Huntington's Chorea) and to classify subjects from dementia and at risk groups into one or other of the above types. An attempt at improving discrimination between known types (and hence hopefully improving the reliability of classifying the subjects from dementia and at risk groups) proved inadequate. However, this is felt to be more due to the small sample size involved than to other factors.

In light of the current understanding of the multi-component nature of the CNV (TECCE and CATTANACH, 1982; SIMONS et al. 1983; LOVELESS and SANFORD, 1974a and 1974b; and ROHRBAUGH et al, 1976 and 1986) it had been felt, prior to the work reported here, that use of a 4 second CNV which was split into epochs might give better discrimination in one or other of the epochs than for the 1 second data. The results presented here, however, show the contrary. For the recorded data the 1 second results consistently led to better classification than the results of either epoch of the 4 second data. A comparison of the results of PrSD and DA showed them to give consistent results (where classification of subjects of unknown type was possible) for the conditions considered best for discrimination (1 second data with a full calibration data set). However, the existence of two atypicality indices in PrSD to check for misclassification of subjects or the possibility that the subject belongs to a type which is not normal or HC is considered to make this technique preferable to DA.

REFERENCES

JERVIS B.W. ,ALLEN E.,JOHNSON T.E., NICHOLS M.J. and HUDSON N.R. (1984)
"The Application of Pattern Recognition Techniques to the Contingent Negative Variation for the Differentiation of Subject Categories" IEEE Trans. Biomed. Eng. BME-31(4) : 342-349

LOVELESS N.E. and SANFORD A.J. (1974a)
"Effects of Age on the Contingent Negative Variation and Preparatory Set in a Reaction-Time Task" Journal of Gerontology, 1974, 29 (1) : 52-63

LOVELESS N.E. and SANFORD A.J. (1974b)
"Slow Potential Correlates of Preparatory Set" Biol. Psychol. 1974, 1 : 303-314

ROHRBAUGH J.W., SYNDULKO K. and LINDSLEY, D.B.(1976)
"Brain Wave Components of the Contingent Negative Variation in Humans" Science, Mar. 1976, 191: 1055-1057

ROHRBAUGH J.W., McCALLUM W.C., GAILLARD A.W.K., SIMONS R.F., BIRBAUMER N. and PAPAKOSTOPOULOS D. (1986)
"ERPs Associated with Preparatory and Movement-Related Processes: a Review" Electroenceph. and Clinical Neurophysiol. Suppl. (Event-Related Potentials) 1986, 38 : 189-229

SIMONS R.F., HOFFMAN J.E. and MACMILLAN F.W. (1983)
"The Component Structure of Event-Related Slow Potentials: Task, ISI and Warning Stimulus Effects on the 'E' Wave" Biol. Psychol. 1983, 17 : 193-219

TECCE J.J. and CATTANACH L. (1982)
"Contingent Negative Variation" - Chapter 36 of "Electroencephalography: Basic Principles, Clinical Applications and Related Fields" Urban and Schwarzenberg, 1982. Eds. Niedermeyer E. and Lopes da Silva F.

6 CONCLUSIONS

This chapter is a summary of the results of Chapters 2,3 and 5 and on this basis recommendations on processing procedures are given. Some suggestions for future investigations are put forward. The content of this chapter is kept to a minimum since the only aim is to draw together the most essential points which should be considered in any future work in this field. The detailed conclusions and the reasoning behind them are presented in each chapter.

6.1 RESULTS AND PROCESSING METHODS

The results of the test data of Chapter 2 have shown the need to include a model of the response to obtain efficient OAR (for NR-OAR and R-OAR) or reduced distortion representation (for R-OAR) of a response present in the EEG. However, it was also seen that mismodelling of the response can cause its own distortion. Results for a recorded CNV showed that the NR-OAR method was relatively insensitive to the inclusion or omission of a model, but for R-OAR modelling was essential. Since this work was performed off-line NR-OAR (without modelling) was chosen.

In the frequency domain it was shown that windowing (or tapering) of the data is desirable. The Kaiser-Bessel window when applied to 1024 point test data gave much better performance than the Tukey window. When 64 point data is considered the improvement is difficult to discern. In light of the results for 1024 point data the Kaiser-Bessel window is to be preferred. The need for a transformation of the data along with the windowing process has also been demonstrated.

The statistical techniques of Discriminant Analysis (DA) and Predictive Statistical Diagnosis (PrSD) have been applied to both 1 second and 4 second ISI CNV data in an attempt to classify subjects of an unknown type (at risk of having Huntington's Chorea) into one of two subject groups (normal or HC). It was found that the 1 second data gave better discrimination as measured by the number of correct classifications of individuals of known type. A method of classification was described and applied to the subjects of unknown type. Of the three at risk subjects all were classified as normal. As of October/November 1987 all three showed no HC symptoms*. Of course, this is not entirely conclusive proof on two counts: first, the sensitivity of classification with varying experimental conditions, i.e., classification was not always consistent and second, the impossibility of stating whether or not HC symptoms will appear later.

On the basis of the results of Chapters 2,3 and 5 the following processing method is recommended:

---

* Personal communication with Dr. E.M. Allen, Freedom Fields Hospital, Plymouth.

(i) the use of modelling of the response is essential if R-OAR is performed and if NR-OAR is applied response modelling (if an accurate model is available) would be beneficial but not essential;

(ii) use of the Kaiser-Bessel window is desirable and should be accompanied by transformation of the windowed data as described in Section 3.4.4;

(iii) the use of PrSD is to be preferred to DA.

6.2 SUGGESTIONS FOR FUTURE INVESTIGATION

The most important investigation required for OAR is to determine a means of obtaining accurate response models. This, however, will present difficulties because of the evolution of the CNV with trial number.

As a further step in the use of PrSD it is suggested that the existing two type classification is increased to three by including the dementia results in the calibration data set. This might provide better measures of the atypicality indices (on the basis of the recommended method in Chapter 4) for the at risk group and the probability of being normal p(N) and so lead to more definite classification.

The most important extension of the PrSD method requires the acquisition of a substantially larger dataset. The data on the individuals of known types should be split into two sections: the first, to be used as a calibration data set, the second, to allow assessment of the correctness or otherwise of the 'rules' derived for classification purposes. Provided a successful classification of the second section is achieved, more confidence could be placed on the classification of the at risk subjects.

6.3 CONCLUDING REMARKS

The overall contribution of this work has been in implementing and developing techniques to improve the investigation of the CNV response of the human EEG. These techniques comprised signal processing and statistical methods. The signal processing work on OAR gave clear evidence of the need to modify the existing NR-OAR and R-OAR methods. The results of the application of windowing to real data were less conclusive. This was due to the restrictions arising from the short duration of the CNV data but the methods involved are still considered relevant to the field of EEG work. The application of the statistical methods gave useful results, although they must be treated with caution because of the small sample size. However the procedures followed here will be of value in any future work with hoped for larger data sets. It is felt that the techniques described here (whether taken as a whole or individually) have helped lay the foundations for future work and it is hoped that they will be of benefit to others.

APPENDIX A1 : DERIVATION OF THE ESTIMATES FOR THE $\hat{\theta}_j$s

Denote the measured EEG as y(i), the true (background EEG) as e(i), the measured EOGs as $x_j(i)$ and the transmission coefficients as θ_j. Let there be p measured EOGs and q model components. Then the general model is:

$$y(i) = \sum_{j=1}^{p+q} \theta_j x_j(i) + e(i) \qquad - \text{(A1)}$$

and so:

$$e(i) = y(i) - \sum_{j=1}^{p+q} \theta_j x_j(i) \qquad - \text{(A2)}$$

Denoting the estimates of e(i) as $\hat{e}(i)$ and of θ_j as $\hat{\theta}_j$, equation (A2) becomes:

$$\hat{e}(i) = y(i) - \sum_{j=1}^{p+q} \hat{\theta}_j x_j(i) \qquad - \text{(A3)}$$

Let the sum of squares of the error term be J, then:

$$J = \sum_{i=1}^{n} [\hat{e}(i)]^2 = \sum_{i=1}^{n} [y(i) - \sum_{j=1}^{p+q} \hat{\theta}_j x_j(i)]^2 \qquad - \text{(A4)}$$

Use of the least squares method of parameter estimation requires that the sum of squares of the error terms, J, is a minimum (this is achieved by computation of appropriate values of the $\hat{\theta}_j$s). For the k th parameter it is necessary to differentiate J partially with respect to $\hat{\theta}_k$:

$$\frac{\partial J}{\partial \hat{\theta}_k} = 0 = \frac{\partial}{\partial \hat{\theta}_k} \sum_{i=1}^{n} [y(i) - \sum_{j=1}^{p+q} \hat{\theta}_j x_j(i)]^2$$

$$= \sum_{i=1}^{n} [-2y(i)x_k(i) + 2\sum_{j=1}^{p+q} \hat{\theta}_j x_j(i)x_k(i)] \qquad - (A5)$$

$$\sum_{i=1}^{n} y(i)x_k(i) = \sum_{i=1}^{n} [\sum_{j=1}^{p+q} \hat{\theta}_j x_j(i)x_k(i)] \qquad - (A6)$$

$$\hat{\theta}_k = \frac{\sum_{i=1}^{n} y(i)x_k(i) - \sum_{i=1}^{n} [\sum_{j\neq k}^{p+q} \hat{\theta}_j x_j(i)x_k(i)]}{\sum_{i=1}^{n} x_k^2(i)} \qquad - (A7)$$

where the second sum for the second term of the numerator is for all values of j except k.

To show that this value of $\hat{\theta}_k$ does give a minimum value of J differentiate equation (A5) with respect to $\hat{\theta}_k$:

$$\frac{\partial^2 J}{\partial \hat{\theta}_k^2} = \frac{\partial^2}{\partial \hat{\theta}_k^2} \{\sum_{i=1}^{n} [-2y(i)x_k(i) + 2\sum_{j=1}^{p+q} \hat{\theta}_j x_j(i)x_k(i)]\}$$

$$= \sum_{i=1}^{n} [\sum_{j=1}^{p+q} 2x_k^2(i)] \qquad - (A8)$$

which must always be positive and so the value of $\hat{\theta}_k$ given in equation (A7) causes J to be a minimum in equation (A4).

As an example put $y(i) = EEG_m(i)$, $e(i) = EEG_t(i)$. Also let there be one $x_j(i)$, $EOG_m(i)$, and no response modelling so that $p = 1$ and $q = 0$. Thus there is only one transmission coefficient θ_1 and so its estimate is:

$$\hat{\theta}_1 = \frac{\sum_{i=1}^{n} EEG_m(i) EOG_m(i)}{\sum_{i=1}^{n} EOG_m^2(i)} \qquad - \text{(A9)}$$

which is of the form of equation (2.22) in Section 2.4.

APPENDIX A2 : PROGRAM UNIT LISTINGS AND DESCRIPTIONS

The contents of this appendix are as follows: Section A2.1 contains diagrams illustrating the calling sequences of the various program units; Section A2.2 is a list of brief descriptions (in alphabetical order) of the various program units used in the signal processing work; and Section A2.3 gives the source code listings.

A2.1 CALLING SEQUENCES

The three main program units are DISC (for PrSD), IDVRES (for generating the test statistics) and MAINPLOT (for producing energy and phase spectra and the averaged waveforms).

In what follows the MAIN program units are given on the left most side of the page. Moving rightwards across the page indicates deeper levels of program invocation, e.g. subroutine MINV is called by subroutine DISP which is called by MAIN program unit DISC. Moving down the page indicates the order in which the program units are first encountered, e.g. MAIN program unit IDVRES first calls GETNAM followed by HRINIT then BATNOS, etc.

Note that the lower case letters following a program unit have the following meanings:

'i' denotes a FORTRAN 66 intrinsic function,

's' denotes a subroutine from the IBM Scientific

Systems Subroutine (SSP) package,

'v' denotes an IBM VM/CMS Systems Subroutine,

'd' denotes that the subroutine has further calls to program units as described earlier in the calling sequence.

```
DISC      DISP   MINV s
          STN    ALGAMA i
                 BETAIN

IDVRES    GETNAM NAMTST
          HRINIT
          BATNOS
          DEVICE
          NROARM MODEL
                 DATIN  NAMCON FILEDF v
                               LBMOVE v
                        FILEDF v
                        DECODE LBMOVE v
                               CORE v
                        I2TOI4
                 MEAN
                 ARRAY  s
                 DGELG  s
          RCOARM MODEL
                 RDATA  GETNAM d
                        DATIN  d
                        MEAN
                        ADAPTV XYVAL  ONLSUB XVAL1
                                             UDUFLT
                               MAXVAL
                               OAGRAF NGRAPH
                               WAIT
                               POSDET
                               MODTHG NGRFS1 NPLOT
                               PRTTHT
                               THENVL
                               MODOAG
          FILTER
          TAPCOS COSGEN
          TAPKAI KAIGEN SUFACT
                        BSSL
          TAPER2 WINDY
          NLOGN
          ASTAT1
          VSTAT2
          VSTAT3
          STPRNT
          STATIO NAMCON
```

```
MAINPLOT  RETYPE v
          INPUT  GETNAM d
                 BATNOS
          PROCES NROARM d
                 RCOARM d
                 FILTER
                 TAPER2 d
                 TAPCOS d
                 TAPKAI d
                 NLOGN
          REDUCT
          DATOUT HEADER
                 DATAXS
                 DATPLT
          PSDOUT HEADER
                 PSDAXS
                 PSDPLT
          PHAOUT HEADER
                 PHAPLT
```

Calls to subroutines not listed above are to subroutines which are part of the GINO-F and GINOGRAF packages.

A2.2 PROGRAM UNIT DETAILS

This section gives a brief description of each of the program units (which are given in alphabetic order).*

ADAPTV OAR-F66

Acts as an interface between the various program units which are responsible for recursive OAR.

ASTAT1 STAT-F66

Carries out 'Rayleigh Test of Circular Variance'.

BATNOS CNV-F66

Determines how many trials to be processed and which are to be included.

---

* Brief descriptions of program units BETAIN, DISC, DISP and STN are given on p. A13.

BSSL TPFL-F66

Computes Bessel functions.

COSGEN TPFL-F66

Generates Tukey window weights.

DATAXS PLOT-F66

Produces axes and associated labels used in generating plots of CNV/ERA or windowed data.

DATIN CNV-F66

Reads information from a specified file pair which consists of data from file 'fname' BINARY and header information (comprising patient name, sample rate and EEG and EOG scale factors) from file 'fname' HEADER. 'fname' is common to both files.

DATOUT PLOT-F66

Controls production of CNV/ERA or windowed data plots.

DATPLT PLOT-F66

Responsible for plotting CNV/ERA or windowed data.

DECODE COMM-F66

Mimics function carried out by PRIME FORTRAN command DECODE as invoked on line DAT 32 of program unit DATIN, n.b. it is not a general purpose replacement for this command.

DEVICE OAR-F66

Selects graphics device to be used in plotting.

FILTER TPFL-F66

Performs low pass filtering.

GETNAM CNV-F66

Reads name of file to be processed. Also determines what type of file it is by invoking NAMTST (see below).

HEADER PLOT-F66

Affixes to each plot (produced under program MAINPLOT) details of name of file, processing options requested and trials included in processing.

HRINIT COMM-F66

Assigns the harmonics to be investigated according to whether a CNV or ERA file is being processed. Harmonic numbers are stored in an array by means of FORTRAN DATA statements and can be changed by altering these statements.

I2TOI4 COMM-F66

Converts INTEGER*2 to standard INTEGER representation. Files are stored in INTEGER*2 format to conserve disc space.

IDVRES OAR-F66

Main program unit responsible for generating details of phase/amplitude results for each harmonic on a trial by trial basis.

INPUT PLOT-F66

When using MAINPLOT this subroutine determines required processing and plotting options.

KAIGEN TPFL-F66

Generates Kaiser-Bessel taper.

MAINPLOT PLOT-F66

Interface between program units used in plotting.

MAXVAL OAR-F66

Ensures that plot data fits into plotting area by determining maximum positive and negative values of y axis data.

MEAN CNV-F66

Removes mean value from data record.

MODEL COMM-F66

Initialises components to be used if modelling has been requested (constructs necessary models according to data file name).

MODOAG OAR-F66

Controls production of plots of model components.

MODTHG OAR-F66

Plots graphs of point by point THETA variation.

NAMCON COMM-F66

Converts name of file to be processed from a form consistent with original PRIME programs to one consistent with VM/CMS file handling.

NAMTST COMM-F66

Determines whether CNV or ERA data is being processed.

NGRAPH OAR-F66

Plots a graph in one of six possible positions.

NGFRS1 OAR-F66

Plots multiple curves on a single graph with selectable y axis scaling.

NLOGN CNV-F66

Performs FFT and inverse FFT.

NPLOT OAR-F66

Used to allow multiple curves to be plotted on one graph.

NROARM COMM-F66

Carries out non-recursive OAR including modelling or not as required.

OAGRAF OAR-F66

Controls production of plots of corrected and uncorrected EEG graphs.

ONLSUB OAR-F66

Carries out actual recursive correction by invoking one of three correction algorithms.

PHAOUT PLOT-F66

Controls production of phase v. frequency plots.

PHAPLT PLOT-F66

Responsible for plotting phase v. frequency graphs.

POSDET OAR-F66

Determines position of end point for each correction parameter on graphs of (THETA) values.

PROCES PLOT-F66

Carries out processing requirements as dictated by program unit INPUT, invoked by program MAINPLOT.

PRTTHT OAR-F66

Appends to THETA plots information showing batch number being corrected and which curve corresponds to which THETA.

PSDAXS PLOT-F66

Produces axes and associated labels used in generating plots of ESD.

PSDOUT PLOT-F66

Controls plotting of ESD.

PSDPLT PLOT-F66

Responsible for plotting ESDs.

RCOARM OAR-F66

Carries out recursive OAR including modelling or not as required replaces EYEREC.

RDATA OAR-F66

Reads data (stored as integers) from a given unformatted (binary) file and converts to real format.

REDUCT COMM-F66

Reduces size of GINO-F plots if so desired (user can select between a choice of sizes).

STATIO COMM-F66

Reads/writes statistics from/to file.

STPRNT COMM-F66

Creates two files (PRE-STAT LISTING and POS-STAT LISTING) containing a condensation of the statistical results for each harmonic for both pre- and post-stimulus epochs.

SUFACT TPFL-F66

Creates an array in COMMON storage containing values of 1! to 32!.

TAPCOS TPFL-F66

Interface to program unit COSGEN and allows mean removal from windowed data.

TAPER2 TPFL-F66

Carries out alternative form of Tukey windowing.

TAPKAI TPFL-F66

Interface to program unit KAIGEN which allows 'complex' mean removal from windowed data.

THENVL OAR-F66

Creates a file (THETA ENDVALS) containing end point values for each THETA.

UDUFLT OAR-F66

UDU algorithm for recursive OAR (the one currently used).

VSTAT2 STAT-F66

Calculates 'Nearest and Furthest Mean Amplitude Difference' test statistics.

VSTAT3 STAT-F66

Calculates 'Modified Rayleigh Test of Circular Variance' statistics.

WAIT COMM-F66

Pauses between displayed screens until user enters a '1'.

WINDY TPFL-F66

Called by TAPER2 (see above) to create window weights.

XVAL1 OAR-F66

Used to set up data array for use when ONLSUB invoked. Contains model components.

XYVAL OAR-F66

If OA model 4D is being used for recursive OAR sets XYVAL vertical left EOG equal to product of horizontal left and right EOG channels.

A2.3 PROGRAM UNIT LISTINGS

In the following the program units have been seperated into two sections: the first lists the signal processing program units; the second lists the program units used for PrSD.

In the signal processing section the program units are listed in alphabetic order. The PrSD program units are listed in the order in which they are called (they are held in a single source file). Listings of program units DATAXS, DATOUT, DATPLT, HEADER, PHAOUT, PHAPLT, PSDAXS, PSDOUT and PSDPLT are not given since they do not perform any processing and are graphics output routines only.

A2.2 PROGRAM UNIT DETAILS (cont.)

Below are brief descriptions of the program units used in predictive statistical diagnosis.

BETAIN

Computes incomplete beta function.

DISC

Main program unit used in implementing predictive statistical diagnosis. Acts as an interface between the required subroutines.

DISP

Generates dispersion matrix.

STN

Generates d-dimensional student-type density function.

```
C  FILE I.D : ADAPTV  OAR-F66               LAST REV : 9 FEB 87
C
C  22 APR 87 : ADDITION OF CALLS TO SUBROUTINE WAIT.
C
      SUBROUTINE ADAPTV(NPT,N,L1,ITYPE,IMODEL,ONLFLG,MPLFLG)
      INTEGER    THEPOS(7),VDUOUT,KYBDIN,ISIVAL
      REAL       THETMP(7),MODEND(3)
      LOGICAL    CNVFLG,ERAFLG,ONLFLG,MPLFLG
      DIMENSION  EEGCD(1024),EEGC6(1024),EEGC7(1024)
      DIMENSION  THETAD(7168),THETA6(7168),THETA7(7168),THEMOD(3072)
      DIMENSION  RSSD(1024),RSS6(1024),RSS7(1024)
      DIMENSION  RSQD(1024),RSQ6(1024),RSQ7(1024)
      DIMENSION  OAD(1024),OA6(1024),OA7(1024)
      DIMENSION  THDEND(4),TH6END(6),TH7END(7)
      COMMON     VL(1024),VR(1024),HL(1024),HR(1024)
      COMMON     /BLNKEX/ CMPNT1(1024),CMPNT2(1024),CMPNT3(1024)
      COMMON     /FLAGS/  CNVFLG,ERAFLG,ISIVAL
      COMMON     /DEVNUM/  IDEV
      DATA       NPAR4,NPAR6,NPAR7 /4,6,7/
      DATA       IDD,ID6,ID7 /5,6,7/
      DATA       VDUOUT,KYBDIN /6,5/
      IF ((IMODEL .NE. 0) .AND. (.NOT.(ONLFLG))) GO TO 200
C  NO CNV OR AER MODELLING REQUIRED.
      CALL XYVAL(THETAD,EEGCD,RSSD,RSQD,OAD,NPAR4,IDD,ITYPE)
      IF (MPLFLG) GO TO 700
      IF (IDEV .EQ. 9) GO TO 700
      CALL MAXVAL(BBEG,BEND,THETAD,N,4)
      DO 100 I=1,4
         THDEND(I) = THETAD(1024*I)
         THETMP(I) = THDEND(I)
  100    CONTINUE
      CALL OAGRAF(VL,VR,HL,HR,OAD,EEGCD,N)
      IF (IDEV .EQ. 1) CALL WAIT
      CALL POSDET(NPAR4,THDEND,THETMP,THEPOS)
      CALL MODTHG(THETAD,N,4,4096,BBEG,BEND)
      CALL PRTTHT(THEPOS,NPAR4,THDEND,1)
      IF (IDEV .EQ. 1) CALL WAIT
      CALL THENVL(THDEND,4)
  150 GO TO 700
  200 IF (.NOT.(CNVFLG)) GO TO 400
C  CNV 2 COMPONENT MODELLING.
      CALL XYVAL(THETA6,EEGC6,RSS6,RSQ6,OA6,NPAR6,ID6,ITYPE)
      IF (MPLFLG) GO TO 700
      IF (IDEV .EQ. 9) GO TO 700
      CALL MAXVAL(BBEG,BEND,THETA6,N,6)
      DO 300 I=1,6
         TH6END(I) = THETA6(1024*I)
         IF (I .LE. 4) THETMP(I) = TH6END(I)
         IF (I .GE. 5) MODEND(I-4) = TH6END(I)
  300    CONTINUE
      CALL OAGRAF(VL,VR,HL,HR,OA6,EEGC6,N)
      IF (IDEV .EQ. 1) CALL WAIT
      CALL POSDET(NPAR4,TH6END,THETMP,THEPOS)
      CALL MODTHG(THETA6,N,4,4096,BBEG,BEND)
      CALL PRTTHT(THEPOS,NPAR4,TH6END,1)
      IF (IDEV .EQ. 1) CALL WAIT
      CALL MODOAG(CMPNT1,CMPNT2,CMPNT3,N)
      IF (IDEV .EQ. 1) CALL WAIT
      DO 340 I=4097,6144
         THEMOD(I-4096) = THETA6(I)
  340    CONTINUE
      THETMP(1) = MODEND(1)
      THETMP(2) = MODEND(2)
      CALL POSDET(2,MODEND,THETMP,THEPOS)
      CALL MODTHG(THEMOD,N,2,2048,BBEG,BEND)
      CALL PRTTHT(THEPOS,2,MODEND,5)
      IF (IDEV .EQ. 1) CALL WAIT
      CALL THENVL(TH6END,6)
      GO TO 700
  400 IF (.NOT.(ERAFLG)) GO TO 1000
C  AER 3 COMPONENET MODELLING.
  420 CALL XYVAL(THETA7,EEGC7,RSS7,RSQ7,OA7,NPAR7,ID7,ITYPE)
      IF (MPLFLG) GO TO 700
      IF (IDEV .EQ. 9) GO TO 700
      CALL MAXVAL(BBEG,BEND,THETA7,N,7)
```

```
      DO 500 I=1,7
         TH7END(I) = THETA7(1024*I)
         IF (I .LE. 4) THETMP(I) = TH7END(I)
         IF (I .GE. 5) MODEND(I-4) = TH7END(I)
  500    CONTINUE
      CALL DAGRAF(VL,VR,HL,HR,DA7,EEGC7,N)
      IF (IDEV .EQ. 1) CALL WAIT
      CALL POSDET(NPAR4,TH7END,THETMP,THEPOS)
      CALL MODTHG(THETA7,N,4,4096,BBEG,BEND)
      CALL PRTTHT(THEPOS,NPAR4,TH7END,1)
      IF (IDEV .EQ. 1) CALL WAIT
      CALL MODDAG(CMPNT1,CMPNT2,CMPNT3,N)
      IF (IDEV .EQ. 1) CALL WAIT
      DO 540 I=4097,7168
         THEMOD(I-4096) = THETA7(I)
  540    CONTINUE
      THETMP(1) = MODEND(1)
      THETMP(2) = MODEND(2)
      THETMP(3) = MODEND(3)
      CALL POSDET(3,MODEND,THETMP,THEPOS)
      CALL MODTHG(THEMOD,N,3,3072,BBEG,BEND)
      CALL PRTTHT(THEPOS,3,MODEND,5)
      IF (IDEV .EQ. 1) CALL WAIT
      CALL THENVL(TH7END,7)
  700 RETURN
C
C  EXCEPTION HANDLER.
C
 1000 WRITE(VDUOUT,1020)
 1020 FORMAT(/,' *** MODELLING OF NON-CNV/ERA DATA REQUESTED. ***',
     1       /,4X,'''0'' TO TERMINATE ''1'' TO CONTINUE')
      READ(KYBDIN,1040)IENQRY
 1040 FORMAT(I1)
      IF ((IENQRY .NE. 0) .AND. (IENQRY .NE. 1)) GO TO 1000
      IF (IENQRY .EQ. 1) GO TO 420
      STOP
      END



C  FILE I.D. : ASTAT1  STAT-F66
C
      SUBROUTINE ASTAT1(ANGLE,N)
C
C     THIS SUBROUTINE CALCULATES SUMMARY STATISTICS FOR THE N ANGULAR
C     VALUES (RADIANS) STORED IN ARRAY 'ANGLE'.
C
C     MEAN DIRECTION ....  THETA=ATAN(S/C)
C
C          WHERE   C = AVERAGE COSINE VALUE
C                  S = AVERAGE SINE VALUE
C
C     CIRCULAR VARIANCE ....  V0=1-SQRT(7C*C+S*S)
C
C          V0 HAS A VALUE 1 FOR COMPLETE UNIFORMITY ON THE CIRCLE
C                         0 FOR A SET OF IDENTICAL ANGLES
C
      INTEGER   VDUOUT
      REAL PRNFST(32),PRNFDF(32),PRPPST(32),PRCVST(32),PRMDST(32)
      REAL NFSTPO(32),NFDFPO(32),PPSTPO(32),CVSTPO(32),MDSTPO(32)
      INTEGER IHRPRE,IHRPOS,PRPPDF,PPDFPO
      LOGICAL PRE,POS
      COMMON /PREST/ PRNFST,PRNFDF,PRPPST,PRPPDF,PRCVST,PRMDST,IHRPRE,
     1               PRE
      COMMON /POSST/ NFSTPO,NFDFPO,PPSTPO,PPDFPO,CVSTPO,MDSTPO,IHRPOS,
     1               POS
      DATA      VDUOUT /14/
C     DATA      VDUOUT /6/
      DIMENSION ANGLE(N)
      DATA PI/3.1415926536/
      C=0
      S=0
      DO 1 I=1,N
         C=C+COS(ANGLE(I))
         S=S+SIN(ANGLE(I))
```

```
    1     CONTINUE
      C=C/N
      S=S/N
      THETA=ATAN2(S,C)
      V0=1-SQRT(C*C+S*S)
      WRITE(VDUOUT,100) THETA
      THETA=THETA*180/PI
      WRITE(VDUOUT,102) THETA
      WRITE(VDUOUT,101) V0
 100  FORMAT(/' MEAN DIRECTION',13X,'= ',F10.5,'  RADIANS')
 101  FORMAT(' CIRCULAR VARIANCE',10X,'= ',F10.5)
 102  FORMAT(28X,'= ',F10.5,'  DEGREES')
      IF (PRE) PRCVST(IHRPRE) = V0
      IF (POS) CVSTPO(IHRPOS) = V0
      RETURN
      END
```

```
C  FILE I.D. : BATNOS  CNV-F66
C
C       READS NUMBER OF BATCHES TO BE PROCESSED FROM KEYBOARD. IF LESS
C  THAN 32 DETERMINES WHICH BATCHES TO BE INCLUDED.
C
      SUBROUTINE BATNOS(BATS,MAX)
      INTEGER BATS(32),MAX,IYES,INO,ANSWER
      DATA IYES,INO /'Y','N'/
   10 WRITE(6,20)
   20 FORMAT(' HOW MANY TRIALS TO BE PROCESSED',
     1       /,5X,'ENTER A 2 DIGIT NUMBER, E.G. 09')
      READ(5,30,ERR=200)MAX
   30 FORMAT(I2)
      IF(MAX .GT. 32 .OR. MAX .LT. 1)GO TO 10
      IF(MAX .EQ. 32)GO TO 100
      ITBAT=0
      I=1
   60 WRITE(6,70)ITBAT
   70 FORMAT(' BATCH ',I3,' TO BE INCLUDED - ''Y'' OR ''N''')
      READ(5,80,ERR=300)ANSWER
   80 FORMAT(A1)
      IF (ANSWER .EQ. INO) GO TO 90
      IF (ANSWER .NE. IYES) GO TO 60
      BATS(I)=ITBAT
      I=I+1
   90 ITBAT=ITBAT+6
      IF(I .LE. MAX)GO TO 60
      RETURN
  100 DO 120 I=1,MAX
          BATS(I)=(I-1)*6
  120     CONTINUE
      RETURN
  200 WRITE(6,210)
  210 FORMAT(' ERROR IN READING NUMBER OF BATCHES TO BE PROCESSED',
     1        /,10X,'PLEASE RE-ENTER VALUE')
      GO TO 10
  300 WRITE(6,310)
  310 FORMAT(' ERROR IN READING ANSWER, PLEASE RE-ENTER')
      GO TO 60
      END
```

```
C  FILE I.D. : BSSL TPFL-F66               LAST REV : 16 FEB 87
C
C  COMPUTES BESSEL FUNCTIONS.
C
      FUNCTION BSSL(X)
      INTEGER I
      REAL BSSL,X,XT,XTSQRT
      COMMON /ARFACT/ FACT(32)
      BSSL = 1.0
      DO 10 I=1,32
         XTSQRT = (X/2)**I/FACT(I)
         IF (XTSQRT .LT. .734684E-39) XTSQRT = .734684E-39
         XT = XTSQRT**2
```

```
         BSSL = BSSL + XT
   10    CONTINUE
      RETURN
      END


C  FILE I.D. : COSGEN  TPFL-F66                    LAST REV : 7 NOV 85
C
C  THIS PROGARM GENERATES A COSINE-BELL WINDOW FUNCTION AND APPLIES IT
C  TO DATA CONSISTING OF 'DATNUM' POINTS STORED IN ARRAY DATA ().
C
      SUBROUTINE COSGEN(DATA,DATNUM)
      INTEGER  DATLEN,DATNUM,TAPNUM,I,J,K
      REAL     DATA(DATNUM),PI,TAPLEN,ARGMNT
      DATA     PI       /3.141592654/
      TAPLEN = 0.125
      DATLEN = DATNUM - 1
      TAPNUM = IFIX(TAPLEN*DATNUM+0.5)
      ARGMNT = PI/(TAPLEN*FLOAT(DATLEN))
      DO 100 I=1,TAPNUM
         J = I - 1
         DATA(I) = 0.5*(1.0-COS(FLOAT(J)*ARGMNT))*DATA(I)
  100    CONTINUE
      K = DATNUM - TAPNUM + 1
      DO 200 I=K,DATNUM
         J = I - 1
         DATA(I) = 0.5*(1.0-COS(FLOAT(DATLEN-J)*ARGMNT))*DATA(I)
  200    CONTINUE
      RETURN
      END


C  FILE I.D. : DATIN  CNV-F66           LAST REV : 10 MAR 87
C
      SUBROUTINE DATIN(IBATNO,I4DATA,RNAME,SF1,SF2,SAMRAT,DFILOP)
C
C       THIS SUBROUTINE READS DATA FROM A FILE SPECIFIED BY ARGUMENT
C  RNAME.  THE BATCH SPECIFIED IS READ INTO IDATA.  THE PROGRAM ALSO
C  RETURNS THE SCALE FACTORS SF1, SF2, AND THE SAMPLE RATE (SAMRAT).
C  SETTING DFILOP = -1 CLOSES ANY OPEN FILE AND RETURNS.
C
      INTEGER I4DATA(1024),RNAME(20),TITLE(36),NME(6)
      INTEGER NBAT,IBATNO,IAA1(4),I,MAXBAT,IL
      INTEGER RETCD,UNITAS,UNITBN,NPT
      INTEGER*2 FMODEA,FMODEB,I2DATA(1024)
      LOGICAL DFILOP
      REAL DEVICE
      REAL*8 NAME,FTYPEA,FTYPEB,OPTSAS(4),OPTSBN(4)
      COMMON /SMPFRQ/ TMPFRQ
      DATA UNITAS,UNITBN,NPT /10,11,1024/
      DATA FTYPEA,FTYPEB /'HEADER','BINARY'/
      DATA FMODEA,FMODEB,DEVICE /'C1','C4','DISK'/
      DATA OPTSAS /'(','RECFM','F',')'/
      DATA OPTSBN /'(','RECFM','VS',')'/
      IF (DFILOP) GO TO 1100
      CALL NAMCON(NAME,RNAME)
      CALL FILEDF(RETCD,UNITAS,DEVICE,NAME,FTYPEA,FMODEA,OPTSAS)
      READ(10,3000,END=1800,ERR=1900)NME,SF1,SF2
      READ(10,3010,END=1800,ERR=1900)MAXBAT
      READ(10,3020,END=1800,ERR=1900)TITLE
      DO 234 IL=1,4
          IAA1(IL)=TITLE(IL+20)
  234     CONTINUE
      CALL DECODE(IAA1,SAMRAT)
      TMPFRQ = SAMRAT
      CALL FILEDF(RETCD,UNITBN,DEVICE,NAME,FTYPEB,FMODEB,OPTSBN)
      DFILOP = .TRUE.
 1100 READ(11,END=1800,ERR=1900)NBAT
      READ(11,END=1800,ERR=1900)(I2DATA(I),I=1,1024)
      IF(NBAT .NE. IBATNO)GO TO 1100
      NBAT=NBAT+1
      CALL I2TOI4(I4DATA,I2DATA,NPT)
      RETURN
```

```
 1800 WRITE(6,1810)RNAME
 1810 FORMAT('*** END OF FILE ',20A1,' ***')
      STOP 3
 1900 WRITE(6,1910)IBATNO,RNAME
 1910 FORMAT('*** ERROR READING BATCH ',I5,' FORM FILE ',20A1,' ***')
      STOP 4
 3000 FORMAT(6A2,2F8.6)
 3010 FORMAT(I4)
 3020 FORMAT(36A2)
      END



C  FILE I.D. : DECODE  COMM-F66                   LAST REV : 23 SEP 86
C
C  THIS SUBROUTINE PERFORMS THE FUNCTION CARRIED OUT BY THE 'DECODE'
C  COMMAND IN PRIME FORTRAN. IT DOES SO USING A CALL TO AN IBM SYSTEMS
C  SUBROUTINE :
C                  CALL CORE(BUFFER,LENGTH)
C  'BUFFER' IS AN ARRAY OF 'LENGTH' BYTES. IT CONTAINS THE CHARACTER
C  (FORMATTED) REPRESENTAION WHICH IS TO BE CONVERTED TO INTERNAL BINARY
C  (UNFORMATED) REPRESENTATION. FOR FURTHER DETAILS SEE SCP CS DOCUMENT
C  'V2/1.6 VM/CMS SYSTEMS SUBROUTINES' JAN 1986 EDITION. N.B. THIS IS
C  NOT A GENERAL PURPOSE REPLACEMENT FOR THE PRIME COMMAND.
C
      SUBROUTINE DECODE(IAA1,SAMRAT)
      INTEGER IAA1(4),ITEMP(2)
      REAL SAMRAT
C
C  CONVERT FROM A2 TO A4 CHARACTER FORMAT.
C
      CALL LBMOVE(IAA1(1),1,ITEMP(1),1,2)
      CALL LBMOVE(IAA1(2),1,ITEMP(1),3,2)
      CALL LBMOVE(IAA1(3),1,ITEMP(2),1,2)
      CALL LBMOVE(IAA1(4),1,ITEMP(2),3,2)
      CALL CORE(ITEMP,8)
      READ(5,10)SAMRAT
   10 FORMAT(F8.4)
      RETURN
      END



C  FILE I.D. : DEVICE  DAR-F66               LAST REV : 21 NOV 86
C
      SUBROUTINE DEVICE(IDEV)
C
C THIS SUBROUTINE NOMINATES THE GRAPHICS DEVICE TO
C BE USED FOR PLOTTING GRAPHS.
C
      COMMON   /DEVNUM/   IDEVNO
   20 WRITE(6,40)
   40 FORMAT(' ENTER 1 TO VIEW, 2 FOR C1051 OR 9 FOR NO GINO-F ACTION')
      READ(5,60,ERR=100)IDEV
   60 FORMAT(I1)
      IF((IDEV .NE. 1) .AND. (IDEV .NE. 2) .AND. (IDEV .NE. 9))GO TO 200
      IF (IDEV .EQ. 1) CALL T4010
      IF (IDEV .EQ. 2) CALL C1051N
      IDEVNO = IDEV
      RETURN
C
C  ERROR MESSAGES.
C
  100 WRITE(6,120)
  120 FORMAT(' ***   ERROR IN READING DEVICE NUMBER, RE-ENTER ***')
      GO TO 20
  200 WRITE(6,220)IDEV
  220 FORMAT(' *** INVALID DEVICE CODE, IDEV = ',I2,' ***')
      GO TO 20
      END
```

```
C  FILE I.D. : FILTER TPFL-F66                        LAST REV : 19 DEC 86
C
      SUBROUTINE FILTER(NPTS,XT)
      DIMENSION XT(NPTS),DATOUT(1024),H(128)
      REAL H
      LOGICAL GOTEM
      INTEGER WTFILE(20)
      INTEGER*2 FMODE
      INTEGER RETCD,UNIT
      REAL DEVICE
      REAL*8 FNAME,FTYPE
      DATA UNIT,DEVICE,FTYPE,FMODE /9,'DISK','FILTER','C1'/
      DATA FNAME /'BP1L21'/
      DATA GOTEM / .FALSE. /
      IF(GOTEM)GO TO 60
      CALL FILEDF(RETCD,UNIT,DEVICE,FNAME,FTYPE,FMODE)
      REWIND 9
      READ(9,10)
   10 FORMAT(/)
      READ(9,15)ICASE
   15 FORMAT(6X,I2)
C             CASE 1 = ODD LENGTH, SYMMETRICAL
C             CASE 2 = EVEN LENGTH, SYMMETRICAL
C             CASE 3 = ODD LENGTH, ANTI-SYMMETRICAL
C             CASE 4 = EVEN LENGTH, ANTI-SYMMETRICAL
      ISGN=1
      IF(ICASE .EQ. 3 .OR. ICASE .EQ. 4)ISGN=-1
      READ(9,20)N
   20 FORMAT(16X,I4)
      N2=(N+1)/2
      DO 50 I=1,N2
          READ(9,40)H(I)
   40     FORMAT(9X,E15.8)
          H(N+1-I)=H(I)*FLOAT(ISGN)
   50     CONTINUE
      GOTEM=.TRUE.
   60 DO 80 I=1,NPTS
          STORE=0.
          IF(N .LT. I)GO TO 70
          STORE=XT(I)
          GO TO 80
   70     DO 75 K=1,N
              STORE=STORE + XT(I-K+1)*H(K)
   75         CONTINUE
          DATOUT(I)=STORE
   80     CONTINUE
      DO 85 I=1,NPTS
          XT(I)=DATOUT(I)
   85     CONTINUE
      RETURN
      END
```

```
C  FILE I.D. : GETNAM  CNV-F66                    LAST REV : 1 MAY 87
C
C       THIS PROGRAM READS A FILE NAME FROM THE USER. IT IS ENTERED INTO
C  ARRAY NAME() IN A1 FORMAT.
C
      SUBROUTINE GETNAM(NAME)
      INTEGER NAME(20),ANSWER,IYES,INO,I
      DATA IYES,INO /'Y','N'/
   10 WRITE(6,20)
   20 FORMAT(' PLEASE ENTER NAME OF FILE TO BE PROCESSED',
     1      /,10X,' - A MAXIMUM OF 8 CHARACTERS')
      READ(5,40,ERR=100)(NAME(I),I=1,8)
   40 FORMAT(8A1)
   70 WRITE(6,80)(NAME(I),I=1,8)
   80 FORMAT(' PLEASE CONFIRM FILE NAME : ',8A1,
     1      /,10X,' - ENTER ''Y'' OR ''N''')
      READ(5,90,ERR=200)ANSWER
   90 FORMAT(A1)
      IF (ANSWER .EQ. INO) GO TO 10
      IF (ANSWER .NE. IYES) GO TO 70
      CALL NAMTST(NAME)
```

```
      RETURN
  100 WRITE(6,110)
  110 FORMAT(' ERROR IN READING FILE NAME, PLEASE RE-ENTER DETAILS')
      GO TO 10
  200 WRITE(6,210)
  210 FORMAT(' ERROR IN READING ANSWER, PLEASE RE-ENTER')
      GO TO 70
      END



C  FILE I.D. : HRINIT  COMM-F66                  LAST REV : 2 MAR 87
C
C      THIS PROGRAM UNIT ASSIGNS THE HARMONIC NUMNERS TO BE INVESTIGATED
C  ACCORDING TO WHICH TYPE OF FILE IS BEING PROCESSED, VIZ. CNV OR ERA
C  FILE.
C
      SUBROUTINE HRINIT(NUMHRM,HRMTAB,FNAME)
      INTEGER   CNVHRM,ERAHRM,NUMHRM,FNAME(20)
      INTEGER   CNVTAB(32),ERATAB(32),HRMTAB(32)
      INTEGER   ISIVAL
      LOGICAL   CNVFLG,ERAFLG
      COMMON   /FLAGS/ CNVFLG,ERAFLG,ISIVAL
C
C  INITIALISATION FOR CNV PROCESSING
C
      DATA CNVHRM /14/
      DATA CNVTAB(1),CNVTAB(2),CNVTAB(3)    /4,6,8/
      DATA CNVTAB(4),CNVTAB(5),CNVTAB(6)    /12,16,20/
      DATA CNVTAB(7),CNVTAB(8),CNVTAB(9)    /24,28,32/
      DATA CNVTAB(10),CNVTAB(11),CNVTAB(12)   /48,64,80/
      DATA CNVTAB(13),CNVTAB(14)   /96,112/
C     DATA CNVTAB(1),CNVTAB(2),CNVTAB(3)   /16,20,36/
C     DATA CNVTAB(4),CNVTAB(5),CNVTAB(6)   /56,76,96/
C     DATA CNVTAB(7),CNVTAB(8),CNVTAB(9)   /124,144,176/
C     DATA CNVTAB(10),CNVTAB(11),CNVTAB(12)  /196,208,244/
C     DATA CNVTAB(13),CNVTAB(14)   /244,272/
C
C  INITIALISATION FOR ERA PROCESSING
C
      DATA ERAHRM /12/
      DATA ERATAB(1),ERATAB(2),ERATAB(3)   /16,24,36/
      DATA ERATAB(4),ERATAB(5),ERATAB(6)   /64,72,92/
      DATA ERATAB(7),ERATAB(8),ERATAB(9)   /116,156,176/
      DATA ERATAB(10),ERATAB(11),ERATAB(12)  /204,224,240/
C     DATA ERATAB(13),ERATAB(14)   /212,232/
C
C  ASSIGN HARMONICS TO BE TESTED ACCORDING TO FILE BEING PROCESSED.
C
      IF (CNVFLG) NUMHRM = CNVHRM
      IF (ERAFLG) NUMHRM = ERAHRM
      DO 100 I=1,NUMHRM
         IF (CNVFLG) HRMTAB(I) = CNVTAB(I)
         IF (ERAFLG) HRMTAB(I) = ERATAB(I)
  100    CONTINUE
      RETURN
      END



C  FILE I.D. : I2TOI4 COMM-F66      CREATED  : 29 JAN 87
C                                   LAST REV : 29 JAN 87
C
C
C  CONVERTS INTEGER*2 DATA TO INTEGER DATA ( 2- TO 4- BYTE ) FORM.
      SUBROUTINE I2TOI4(I4DATA,I2DATA,NPT)
      INTEGER   I4DATA(1024),NPT,VDUOUT
      INTEGER*2 I2DATA(1024)
      DATA  VDUOUT /6/
      IF ((NPT .LE. 0) .OR. (NPT .GT. 1024)) GO TO 9000
      DO 100 I=1,NPT
         I4DATA(I) = I2DATA(I)
100      CONTINUE
      RETURN
9000  WRITE(VDUOUT,9020)NPT
9020  FORMAT(/,' *** ERROR IN SUBROUTINE I2TOI4 ***',/
```

```
     1 ,4X,'VALUE OF NPT = ',I8)
      STOP
      END



C  FILE I.D. : IDVRES  OAR-F66           LAST REV : 27 APR 87
C
      INTEGER FNAME(20),CHANAN,BATNRS(32),SSR,DEL,BLINE
      INTEGER TDLEN,BATNO,STATOU,IOFLAG
      INTEGER PRES,POSS,IDEV,ISTART,IFILT,IS1,IS2,ITL,ITRIAL,LGTDLN
      INTEGER EYETYP,HRMTAB(32),ANSWER,WINTYP,MBAT,MBAT1,NUMHRM
      INTEGER PREPOS,HRMNIC,VDUOUT,KYBDIN,IMEAN
      INTEGER I,K,L1,N,NP,N1,ISIVAL,IEPOCH
      INTEGER IHAR,IHRPRE,IHRPOS,PRPPDF,PPDFPO
      LOGICAL PRE,POS,MPLFLG,STAFLG,CNVFLG,ERAFLG,OPNFIL,DFILOP,WFILOP
      REAL DATA(1024),ALPHA,ECORR,PI,RMEAN,SF1,SF2,SAMRAT,STDEV
      REAL SUM,SUMSQ,TSTAT,WINVAL(1024),WISUM1,WISUM3
      REAL TRDAT(1024),ANGLE(32),RAD(32),RADDIF(256,32)
      REAL PRNFST(32),PRNFDF(32),PRPPST(32),PRCVST(32),PRMDST(32)
      REAL NFSTPO(32),NFDFPO(32),PPSTPO(32),CVSTPO(32),MDSTPO(32)
      REAL  PCADAT(500,32)
      COMPLEX TDATA(1024),PRES1(1024,32),S1(1024,32)
      REAL*8    FTYPE
      INTEGER*2 FMODE
      COMMON /IFLSEL/ IFILT
      COMMON /TRLNUM/ BATNO
      COMMON /PREST/ PRNFST,PRNFDF,PRPPST,PRPPDF,PRCVST,PRMDST,IHRPRE,
     1               PRE
      COMMON /POSST/ NFSTPO,NFDFPO,PPSTPO,PPDFPO,CVSTPO,MDSTPO,IHRPOS,
     1               POS
      COMMON /EPOCH/ IEPOCH
      COMMON /FLAGS/ CNVFLG,ERAFLG,ISIVAL
      COMMON /DTFLST/ DFILOP
      COMMON /WFILST/ WFILOP,WISUM1,WISUM3,WINVAL
      DATA N,CHANAN,PI /1024,4,3.141592654/
      DATA PRES,POSS,IS1,IS2,DEL /342,472,407,532,125/
      DATA TDLEN,LGTDLN,NP,SSR /1024,10,64,1/
      DATA VDUOUT,KYBDIN,STATOU /6,5,14/
      DATA MPLFLG,OPNFIL /.FALSE.,.FALSE./
      DATA   FTYPE,FMODE   /'STATFILE','C'/
      DFILOP = .FALSE.
      WFILOP = .FALSE.
      ITL = TDLEN/2 + 1
      N1 = NP*SSR
      DO 20 I=1,20
         FNAME(I) = 0
   20    CONTINUE
   50 CALL GETNAM(FNAME)
      CALL HRINIT(NUMHRM,HRMTAB,FNAME)
      IF (NUMHRM .GT. 32) GO TO 4000
      IF (ERAFLG .OR. (CNVFLG .AND. (ISIVAL .EQ. 1))) GO TO 80
55    WRITE(VDUOUT,60)
60    FORMAT(' PROCES POST STIMULUS EPOCH 1 OR 2 - ENTER NUMBER')
      READ(KYBDIN,70,ERR=55)IEPOCH
70    FORMAT(I1)
      IF ((IEPOCH .NE. 1) .AND. (IEPOCH .NE. 2)) GO TO 55
      PRES = 22
      IF (IEPOCH .EQ. 1) POSS = 288
      IF (IEPOCH .EQ. 2) POSS = 490
      IS1 = 219
      IS2 = 719
      DEL = 500
      NP = 216
      N1 = NP*SSR
   80 CALL BATNOS(BATNRS,MBAT)
      CALL DEVICE(IDEV)
  100 WRITE(VDUOUT,110)
  110 FORMAT('  (1) FOR OA REMOVAL (0) NOT TO DO SO')
      READ(KYBDIN,*,ERR=100)ECORR
C
C  SET EYETYP = 0 TO ENSURE DATA READ FROM FILES BY MEANS OF SUBROUTINE
C  NRDARM
C
```

```
      EYETYP = 0
      IF (ECORR .EQ. 0.0) GO TO 200
  150 WRITE(VDUOUT,160)
  160 FORMAT(' RECURSIVE (1) OR NON-RECURSIVE(0)')
      READ(KYBDIN,5020,ERR=150)EYETYP
  200 WRITE(VDUOUT,210)
  210 FORMAT('  (1) TO FILTER , (0) OTHERWISE')
      READ(KYBDIN,5020,ERR=200)IFILT
  250 WRITE(VDUOUT,260)
  260 FORMAT(' PERFORM BASELINE CORRECTIONS')
      READ(KYBDIN,5020,ERR=250)BLINE
  300 WRITE(VDUOUT,310)
  310 FORMAT('  ''0'' FOR TUKEY WINDOW, ''1'' FOR KAISER-BESSEL WINDOW',
     1       /,'  ''2'' FOR MJN TAPER2 WINDOW')
      READ(KYBDIN,5020,ERR=300)WINTYP
  312 IF (WINTYP .NE. 1) GO TO 320
      WRITE(VDUOUT,315)
  315 FORMAT(' ENTER ALPHA VALUE')
      READ(KYBDIN,318,ERR=312)ALPHA
  318 FORMAT(F5.0)
  320 WRITE(VDUOUT,330)
  330 FORMAT('  REMOVE MEAN FROM WINDOWED DATA ''1'' OR NOT ''0''')
      READ(KYBDIN,5020,ERR=320)IMEAN
  340 WRITE(VDUOUT,350)
  350 FORMAT(' TYPE OF MODELLING REQUIRED -',/,
     1   4X,'''0'' FOR NO MODELLING',/,4X,'''10'' FOR STRAIGHT LINE',
     2   ' MODELLING',/,6X,'ENTER AS A 2 DIGIT NUMBER, E.G. 09')
      READ(KYBDIN,360,ERR=340)IMODEL
  360 FORMAT(I2)
  400 WRITE(VDUOUT,410)
  410 FORMAT(' STATISTICS TO BE INCLUDED, ENTER (T) FOR YES, (F)',
     1       ' FOR NO')
      READ(KYBDIN,420,ERR=400)STAFLG
  420 FORMAT(L1)
      IF ((.NOT.(STAFLG)) .AND. (.NOT.(.NOT.(STAFLG)))) GO TO 400
      IF (IFILT .EQ. 0) PRES = PRES - 21
      IF (IFILT .EQ. 0) POSS = POSS - 21
      IF (IFILT .EQ. 1) IS1 = IS1 + 21
      IF (IFILT .EQ. 1) IS2 = IS2 + 21
      DEL = DEL - 48
C
C           THESE PARAMETERS ARE GOVERNED
C           BY THE CNV PARADIGM
C           IS1 IS THE S1 STIMULUS
C           IS2 IS THE S2 STIMULUS
C           PRES IS THE START OF THE PRE. STIM. DATA TO BE ANALYSED.
C           POSS IS THE START OF THE POST STIM. DATA TO BE ANALYSED.
C           ALL VALUES ARE IN TERMS OF 'SAMPLE NUMBER' 1  TO 1024
C
C                 NOW THE ITERATIVE BIT
C
      DO 1200 ITRIAL=1,NBAT
          BATNO=BATNRS(ITRIAL)
          WRITE(VDUOUT,800)BATNO
  800     FORMAT(' PROCESSING BATCH ',I4)
          IF (EYETYP .EQ. 0) CALL NRDARM(FNAME,BATNO,CHANAN-3,DATA,SF1,
     1                                   SF2,SAMRAT,ECORR,IMODEL)
          IF((ECORR .EQ. 1.).AND.(EYETYP .EQ. 1))CALL RCOARM(FNAME,BATNO,
     1         ,DATA,IMODEL,MPLFLG)
          IF(IFILT .EQ. 1)CALL FILTER(N,DATA)
C
C                    PRE STIMULUS
C
C      COPY DATA FOR PCA PROCESSING.
C
        IBEG = IS1 - 1 + 48
        IEND = IS2 - 1
        DO 820 I=1,DEL
           PCADAT(I,ITRIAL) = DATA(I+IBEG)*1.E6
820        CONTINUE
         ISTART=PRES-1
         L1=0
         DO 900 I=1,N1,SSR
             L1=L1+1
             TRDAT(L1)=DATA(I+ISTART)
```

```
  900          CONTINUE
          IF (WINTYP .EQ. 0)CALL TAPCOS(TRDAT,NP,IMEAN)
          IF (WINTYP .EQ. 1)CALL TAPKAI(TRDAT,NP,ALPHA,IMEAN)
          IF (WINTYP .EQ. 2)CALL TAPER2(TRDAT,NP,IMEAN)
          DO 950 I=1,TDLEN
               TDATA(I)=CMPLX(0.,0.)
               IF(I .LE. NP)TDATA(I)=CMPLX(TRDAT(I),0.)
  950          CONTINUE
  151     CALL NLOGN(LGTDLN,TDATA,-1.)
C
C         ***  STUDY HARMONIC   ***
C
          DO 1000 IHAR=1,NUMHRM
               HRMNIC = HRMTAB(IHAR)
               PRES1(IHAR,ITRIAL)=TDATA(HRMNIC+1)*2./FLOAT(NP)
 1000          CONTINUE
C
C                     POST STIMULUS
          ISTART=POSS-1
          L1=0
          DO 1050 I=1,N1,SSR
               L1=L1+1
               TRDAT(L1)=DATA(I+ISTART)
 1050          CONTINUE
          IF (WINTYP .EQ. 0)CALL TAPCOS(TRDAT,NP,IMEAN)
          IF (WINTYP .EQ. 1)CALL TAPKAI(TRDAT,NP,ALPHA,IMEAN)
          IF (WINTYP .EQ. 2)CALL TAPER2(TRDAT,NP,IMEAN)
          DO 1100 I=1,TDLEN
               TDATA(I)=CMPLX(0.,0.)
               IF(I .LE. NP)TDATA(I)=CMPLX(TRDAT(I),0.)
 1100          CONTINUE
          CALL NLOGN(LGTDLN,TDATA,-1.)
          DO 1150 IHAR=1,NUMHRM
               HRMNIC = HRMTAB(IHAR)
               S1(IHAR,ITRIAL)=TDATA(HRMNIC+1)*2./FLOAT(NP)
 1150          CONTINUE
 1200     CONTINUE
      CALL DEVEND
C
C      CREATE FILE CONTAINING DATA FOR PCA PROCESSING.
C
      DO 1280 ISAMP=1,DEL
         DO 1260 ILINE=1,4
              K = (ILINE-1)*8 + 1
              L = K + 7
              WRITE(27,1240)(PCADAT(ISAMP,ITRIAL),ITRIAL=K,L)
1240          FORMAT(8(F8.3,1X))
1260          CONTINUE
1280     CONTINUE
      IF (.NOT.(STAFLG)) GO TO 3100
      WRITE(STATOU,1300)(FNAME(I),I=1,8)
 1300 FORMAT(20X,'DATA FILE = ',8X,8A1)
      WRITE(STATOU,1320)NBAT
 1320 FORMAT(20X,'NUMBER OF BATCHES = ',I2)
      WRITE(STATOU,1340)NP
 1340 FORMAT(20X,'NUMBER OF DATA POINTS = ',I4)
      WRITE(STATOU,1360)TDLEN
 1360 FORMAT(20X,'FFT LENGTH = ',I4,' POINTS')
      WRITE(STATOU,1380)PRES
 1380 FORMAT(20X,'PRE-STIMULUS ANALYSED DATA EXTENDS FROM POINT ',I4)
      WRITE(STATOU,1400)POSS
 1400 FORMAT(20X,'POST-STIMULUS ANALYSED DATA EXTENDS FROM POINT ',I4)
      IF ((ECORR .EQ. 1.) .AND. (EYETYP .EQ. 0)) WRITE(STATOU,1420)
      IF ((ECORR .EQ. 1.) .AND. (EYETYP .EQ. 1)) WRITE(STATOU,1440)
      IF (BLINE .EQ. 1) WRITE(STATOU,1460)
      IF (IFILT .EQ. 1) WRITE(STATOU,1480)
      IF (WINTYP .EQ. 0) WRITE(STATOU,1500)
      IF (WINTYP .EQ. 1) WRITE(STATOU,1520)ALPHA
      IF (WINTYP .EQ. 2) WRITE(STATOU,1540)
      IF (IMEAN .EQ. 1) WRITE(STATOU,1560)
      IF (IMEAN .EQ. 0) WRITE(STATOU,1570)
1420 FORMAT(20X,'NON RECURSIVE OA REMOVAL APPLIED')
1440 FORMAT(20X,'RECURSIVE OA REMOVAL APPLIED')
1460 FORMAT(20X,'BASELINE CORRECTION APPLIED')
1480 FORMAT(20X,'FILTERING OF DATA PERFORMED')
```

```
 1500 FORMAT(20X,'25 PER-CENT TUKEY WINDOW APPLIED')
 1520 FORMAT(20X,'KAISER-BESSEL WINDOW APPLIED WITH ALPHA = ',F8.4)
 1540 FORMAT(20X,'MJN 25 PER-CENT ''TAPER2'' TUKEY WINDOW APPLIED')
 1560 FORMAT(20X,'MEAN OF WINDOWED DATA REMOVED')
 1570 FORMAT(20X,'MEAN OF WINDOWED DATA INCLUDED')
      WRITE(STATOU,1580)
 1580 FORMAT(///)
      DO 1650 IHAR=1,NUMHRM
      IHRPRE = IHAR
      IHRPOS = IHAR
          HRMNIC = HRMTAB(IHAR)
          DO 1600 K=1,MBAT
              ANGLE(K)=ATAN2(AIMAG(PRES1(IHAR,K)),REAL(PRES1(IHAR,K)))
              RAD(K)=CABS(PRES1(IHAR,K)) .
 1600         CONTINUE
          WRITE(STATOU,1610)HRMTAB(IHAR)
 1610     FORMAT(20X,'ANGULAR STATISTICS FOR PRE-STIMULUS HARMONIC ',I3)
         PRE = .TRUE.
         POS = .FALSE.
C   STATISTICAL TESTS FOR PRE-STIMULUS EPOCH
          CALL ASTAT1(ANGLE,MBAT)
          CALL VSTAT3(ANGLE,RAD,MBAT)
          CALL VSTAT2(ANGLE,RAD,MBAT)
          WRITE(STATOU,1620)
 1620     FORMAT(/////)
          DO 1630 K=1,MBAT
              ANGLE(K)=ATAN2(AIMAG(S1(IHAR,K)),REAL(S1(IHAR,K)))
              RAD(K)=CABS(S1(IHAR,K))
              RADDIF(IHAR,K)=CABS(S1(IHAR,K))-CABS(PRES1(IHAR,K))
 1630         CONTINUE
          WRITE(STATOU,1635)HRMTAB(IHAR)
 1635     FORMAT(20X,'ANGULAR STATISTICS FOR POST-STIMULUS HARMONIC '
     1    ,I3)
         PRE = .FALSE.
         POS = .TRUE.
C  STATISTICAL TESTS FOR POST-STIMULUS EPOCH
          CALL ASTAT1(ANGLE,MBAT)
          CALL VSTAT3(ANGLE,RAD,MBAT)
          CALL VSTAT2(ANGLE,RAD,MBAT)
          WRITE(STATOU,1640)
 1640     FORMAT(////)
 1650 CONTINUE
      WRITE(STATOU,2100)
 2100 FORMAT(//,' RESULTS OF A PAIRED T-TEST ON THE PRE-POST RADIUS LENGT
     +HS',///)
      MBAT1=MBAT-1
      DO 3000 IHAR=1,NUMHRM
          SUM=0.
          DO 2500 I=1,MBAT
              SUM=SUM+RADDIF(IHAR,I)
 2500         CONTINUE
          RMEAN=SUM/FLOAT(MBAT)
          SUMSQ=0.
          DO 2600 I=1,MBAT
              SUMSQ=SUMSQ+(RADDIF(IHAR,I)-RMEAN)*(RADDIF(IHAR,I)-RMEAN)
 2600         CONTINUE
          STDEV=SQRT(SUMSQ/FLOAT(MBAT1))
          TSTAT=RMEAN/(STDEV/SQRT(FLOAT(MBAT)))
          PPSTPO(IHAR) = TSTAT
          PPDFPO = MBAT1
          WRITE(STATOU,2710)HRMTAB(IHAR),RMEAN,STDEV,TSTAT,MBAT1
 2710     FORMAT(' HARMONIC= ',I3,5X,'MEAN= ',E14.8,5X,
     1    'ST. DEV= ',E14.8,5X,'T= ',F8.4,5X,'WITH',5X,I3,'  DF')
 3000     CONTINUE
      PREPOS = 0
      CALL STPRNT(PREPOS,NP,TDLEN,PRES,POSS,ECORR,BLINE,IFILT,
     1     WINTYP,ALPHA,NUMHRM,HRMTAB,MBAT,BATNRS,FNAME,EYETYP,IMEAN,
     2     PRNFST,PRNFDF,PRPPST,PRPPDF,PRCVST,PRMDST,IMODEL)
      IOFLAG = 1
      CALL STATIO(PREPOS,NP,TDLEN,PRES,POSS,ECORR,BLINE,IFILT,
     1     WINTYP,ALPHA,NUMHRM,HRMTAB,MBAT,BATNRS,FNAME,EYETYP,IMEAN,
     2     PRNFST,PRNFDF,PRPPST,PRPPDF,PRCVST,PRMDST,IMODEL,
     3     IOFLAG,FTYPE,FMODE,OPNFIL)
      PREPOS = 1
      CALL STPRNT(PREPOS,NP,TDLEN,PRES,POSS,ECORR,BLINE,IFILT,
```

```
     1      WINTYP,ALPHA,NUMHRM,HRMTAB,MBAT,BATNRS,FNAME,EYETYP,IMEAN,
     2      NFSTPO,NFDFPO,PPSTPO,PPDFPO,CVSTPO,MDSTPO,IMODEL)
      CALL STATIO(PREPOS,NP,TDLEN,PRES,POSS,ECORR,BLINE,IFILT,
     1      WINTYP,ALPHA,NUMHRM,HRMTAB,MBAT,BATNRS,FNAME,EYETYP,IMEAN,
     2      NFSTPO,NFDFPO,PPSTPO,PPDFPO,CVSTPO,MDSTPO,IMODEL,
     3      IOFLAG,FTYPE,FMODE,OPNFIL)
 3100 STOP
C  ERROR MESSAGES
 4000 WRITE(VDUOUT,4020)
 4020 FORMAT(2X,'*** ERROR IN PROGRAM UNIT IDVRES ***')
      WRITE(VDUOUT,4040)NUMHRM
 4040 FORMAT(5X,'NUMHRM = ',I3,' EXCEEDS HRMTAB DIMENSION')
      GO TO 3100
C  MULTI-REFERERENCED FORMAT STATEMENTS.
 5020 FORMAT(I1)
      END



C  FILE I.D. : INPUT  PLOT-F66            LAST REV : 18 MAR 87
C
C  5 JUN 87  :  ADDITION OF CODE TO SKIP PSD QUESTIONS IF NOT REQUIRED
C
C  THIS SUBROUTINE INTERROGATES THE USER FOR DESIRED PROCESSING OPTIONS.
C
      SUBROUTINE INPUT(FNAME,IDEV,ECORR,BLINE,IFILT,PLOTAN,BATNRS,
     1            NUMBAT,WINDOW,PLTYPE,EYETYP,LOGTYP,IMEAN,
     2            IMODEL,ALPHA)
      INTEGER  FNAME(20),PLOTAN(4),BATNRS(32)
      INTEGER  IDEV,BLINE,IFILT,NUMBAT,WINDOW
      INTEGER  ANSWER,IYES,INO,PLTYPE,LOGTYP,EYETYP,IMEAN
      INTEGER  VDUOUT,KYBDIN,IMODEL,ISIVAL,IEPOCH
      LOGICAL  CNVFLG,ERAFLG
      REAL     ECORR,ALPHA
      DATA     IYES,INO,VDUOUT,KYBDIN /'Y','N',6,5/
      COMMON   /EPOCH/ IEPOCH
      COMMON   /FLAGS/ CNVFLG,ERAFLG,ISIVAL
  100 WRITE(VDUOUT,110)
  110 FORMAT(///)
      CALL GETNAM(FNAME)
      IF (ERAFLG .OR. (CNVFLG .AND. (ISIVAL .EQ. 1))) GO TO 118
112   WRITE(VDUOUT,114)
114   FORMAT(' FOR 4S ISI CNV PROCES POST STIMULUS EPOCH 1 OR 2',
     1       ' - ENTER APPROPRIATE NUMBER')
      READ(KYBDIN,1020)IEPOCH
      IF ((IEPOCH .NE. 1) .AND. (IEPOCH .NE. 2)) GO TO 112
  118 WRITE(VDUOUT,120)
  120 FORMAT(' DO YOU WANT TO DISPLAY ON TEKTRONIX (0) OR CALCOMP(1)')
      READ(KYBDIN,1020,ERR=118)IDEV
  140 WRITE(VDUOUT,150)
  150 FORMAT(' DO YOU WANT EYE MOVEMENT CORRECTIONS - ''Y'' OR ''N''')
      READ(KYBDIN,1040,ERR=140)ANSWER
      IF ((ANSWER .NE. IYES) .AND. (ANSWER .NE. INO)) GO TO 140
      ECORR = 0.0
      IF (ANSWER .EQ. IYES) ECORR = 1.
      IF (ECORR .EQ. 0.) GO TO 230
  170 WRITE(VDUOUT,180)
  180 FORMAT(' NON-RECURSIVE ''0'' OR RECURSIVE ''1'' OA REMOVAL')
      READ(KYBDIN,1020,ERR=170)EYETYP
      IF ((EYETYP .NE. 0) .AND. (EYETYP .NE. 1)) GO TO 170
  200 WRITE(VDUOUT,210)
  210 FORMAT(' TYPE OF MODELLING REQUIRED -',/,
     1  4X,'''0'' FOR NO MODELLING',/,4X,'''10'' FOR STRAIGHT LINE',
     2  ' MODELLING',/,6X,'ENTER AS 2 DIGIT NUMBER, E.G. 04')
      READ(KYBDIN,220,ERR=200)IMODEL
  220 FORMAT(I2)
      IF ((IMODEL .NE. 0) .AND. (IMODEL .NE. 10))GO TO 200
  230 WRITE(VDUOUT,240)
  240 FORMAT(' DO YOU WANT BASELINE CORRECTIONS - ''Y'' OR ''N''')
      READ(KYBDIN,1040,ERR=230)ANSWER
      IF ((ANSWER .NE. IYES) .AND. (ANSWER .NE. INO)) GO TO 230
      BLINE = 0
      IF (ANSWER .EQ. IYES) BLINE = 1
  260 WRITE(VDUOUT,270)
  270 FORMAT(' DO YOU WANT FILTERING - ''Y'' OR ''N''')
```

```
      READ(KYBDIN,1040,ERR=260)ANSWER
      IF ((ANSWER .NE. IYES) .AND. (ANSWER .NE. INO)) GO TO 260
      IFILT = 0
      IF (ANSWER .EQ. IYES) IFILT = 1
      WRITE(VDUOUT,290)
  290 FORMAT(' REQUIRED PLOTTING OPTIONS (ENTER ''Y'' OR''N'') :')
  300 WRITE(VDUOUT,310)
  310 FORMAT(' AVERAGED CNV')
      READ(KYBDIN,1040,ERR=300)PLOTAN(1)
      IF ((PLOTAN(1) .NE. IYES) .AND. (PLOTAN(1) .NE. INO)) GO TO 300
  330 WRITE(VDUOUT,340)
  340 FORMAT(' WINDOWED DATA')
      READ(KYBDIN,1040,ERR=330)PLOTAN(2)
      IF ((PLOTAN(2) .NE. IYES) .AND. (PLOTAN(2) .NE. INO)) GO TO 330
  360 WRITE(VDUOUT,370)
  370 FORMAT(' POWER SPECTRAL DENSITY')
      READ(KYBDIN,1040,ERR=360)PLOTAN(3)
      IF ((PLOTAN(3) .NE. IYES) .AND. (PLOTAN(3) .NE. INO)) GO TO 360
      IF (PLOTAN(3) .EQ. INO) GO TO 500
  390 IF (PLOTAN(3) .EQ. IYES) WRITE(VDUOUT,400)
  400 FORMAT(' LINEAR ''0'' OR LOGARITHMIC ''1'' PLOT')
      IF (PLOTAN(3) .EQ. IYES) READ(KYBDIN,1020,ERR=390)PLTYPE
      IF ((PLTYPE .NE. 0) .AND. (PLTYPE .NE. 1)) GO TO 390
  420 IF (PLTYPE .EQ. 1) WRITE(VDUOUT,430)
  430 FORMAT(' RELATIVE ''0'' OR ABSOLUTE ''1'' VALUES')
      IF (PLTYPE .EQ. 1) READ(KYBDIN,1020,ERR=420)LOGTYP
  500 WRITE(VDUOUT,510)
  510 FORMAT(' PLOT PHASE ANGLE V. FREQUENCY, ''Y'' OR ''N''')
      READ(KYBDIN,1040,ERR=500)PLOTAN(4)
      IF ((PLOTAN(4) .NE. IYES) .AND. (PLOTAN(4) .NE. INO)) GO TO 500
      CALL BATNOS(BATNRS,NUMBAT)
  540 WRITE(VDUOUT,550)
  550 FORMAT(' DO YOU WANT COSINE-BELL (0), OR KAISER-BESEL (1) WINDOW')
      WRITE(VDUOUT,560)
  560 FORMAT(4X,'RECTANGULAR WINDOW (8)')
      WRITE(VDUOUT,570)
  570 FORMAT(4X,'INVESTIGATE WINDOW ITSELF (9)')
      WRITE(VDUOUT,580)
  580 FORMAT(4X,'USE MJN ''TAPER2'' WINDOW (2)')
      READ(KYBDIN,1020,ERR=540)WINDOW
      IF ((WINDOW .NE. 0) .AND. (WINDOW .NE. 1) .AND. (WINDOW .NE. 8)
     1    .AND. (WINDOW .NE. 9) .AND. (WINDOW .NE. 2)) GO TO 540
  585 IF(WINDOW .NE. 1) GO TO 630
      WRITE(VDUOUT,590)
  590 FORMAT(' ENTER ALPHA VALUE')
      READ(KYBDIN,595,ERR=585)ALPHA
  595 FORMAT(F5.0)
  630 WRITE(VDUOUT,640)
  640 FORMAT(' REMOVAL OF MEAN FROM WINDOWED DATA',
     1        /,'    ''0'' FOR NO, ''1'' FOR YES')
      READ(KYBDIN,1020,ERR=630)IMEAN
      IF ((IMEAN .NE. 0) .AND. (IMEAN .NE. 1)) GO TO 630
      WRITE(VDUOUT,660)
  660 FORMAT(///,' PLEASE CONFIRM THE FOLLOWING REQUIREMENTS :')
      WRITE(VDUOUT,670)(FNAME(I),I=1,8)
  670 FORMAT(' FILE TO BE PROCESSED : ',8A1)
      IF (CNVFLG .AND. (ISIVAL .EQ. 4)) WRITE(VDUOUT,675)IEPOCH
675   FORMAT(' PROCES POST-STIMULUS EPOCH ',I2)
      IF (IDEV .EQ. 0) WRITE(VDUOUT,680)
      IF (IDEV .EQ. 1) WRITE(VDUOUT,690)
  680 FORMAT(' SELECTED OUTPUT DEVICE  :  TEKTRONIX 4010')
  690 FORMAT(' SELECTED OUTPUT DEVICE  :  CALCOMP 1051')
      IF (ECORR .EQ. 0.) WRITE(VDUOUT,700)
      IF ((ECORR .EQ. 1.) .AND. (EYETYP .EQ. 0) .AND. (IMODEL .EQ. 0))
     1  WRITE(VDUOUT,710)
      IF ((ECORR .EQ. 1.) .AND. (EYETYP .EQ. 1) .AND. (IMODEL .EQ. 0))
     1  WRITE(VDUOUT,720)
      IF ((ECORR .EQ. 1.) .AND. (EYETYP .EQ. 0) .AND. (IMODEL .EQ. 10))
     1   WRITE(VDUOUT,740)
      IF ((ECORR .EQ. 1.) .AND. (EYETYP .EQ. 1) .AND. (IMODEL .EQ. 10))
     1   WRITE(VDUOUT,750)
  700 FORMAT(' NO EYE MOVEMENT CORRECTIONS REQUESTED')
  710 FORMAT(' PERFORM NON-RECURSIVE EYE MOVEMENT CORRECTIONS',
     1  ' (NO MODELLING)')
  720 FORMAT(' PERFORM RECURSIVE EYE MOVEMENT CORRECTIONS',
```

```
     1  ' (NO MODELLING)')
  740 FORMAT(' PERFORM NON-RECURSIVE EYE MOVEMENT CORRECTIONS',
     1  ' (STRAIGHT LINE MODELLING)')
  750 FORMAT(' PERFORM RECURSIVE EYE MOVEMENT CORRECTIONS',
     1  ' (STRAIGHT LINE MODELLING)')
      IF (BLINE .EQ. 1) WRITE(VDUOUT,760)
  760 FORMAT(' PERFORM BASELINE CORRECTIONS')
      IF (IFILT .EQ. 1) WRITE(VDUOUT,770)
  770 FORMAT(' FILTER AVERAGED CNV')
      IF (PLOTAN(1) .EQ. IYES) WRITE(VDUOUT,780)
  780 FORMAT(' PLOT AVERAGED CNV')
      IF (PLOTAN(2) .EQ. IYES) WRITE(VDUOUT,790)
  790 FORMAT(' PLOT WINDOWED DATA')
      IF ((PLOTAN(3) .EQ. IYES) .AND. (PLTYPE .EQ. 0))WRITE(VDUOUT,800)
      IF ((PLOTAN(3) .EQ. IYES) .AND. (PLTYPE .EQ. 1) .AND.
     1    (LOGTYP .EQ. 0)) WRITE(VDUOUT,810)
      IF ((PLOTAN(3) .EQ. IYES) .AND. (PLTYPE .EQ. 1) .AND.
     1    (LOGTYP .EQ. 1)) WRITE(VDUOUT,820)
  800 FORMAT(' LINEAR PLOT OF POWER SPECTRAL DENSITY')
  810 FORMAT(' RELATIVE LOGARITHMIC PLOT OF POWER SPECTRAL DENSITY')
  820 FORMAT(' ABSOLUTE LOGARITHMIC PLOT OF POWER SPECTRAL DENSITY')
      IF (PLOTAN(4) .EQ. IYES) WRITE(VDUOUT,830)
  830 FORMAT(' PLOT OF PHASE ANGLE V. FREQUENCY')
      IF (NUMBAT .EQ. 32) WRITE(VDUOUT,840)
      IF (NUMBAT .LT. 32) WRITE(VDUOUT,850)NUMBAT
      IF (NUMBAT .LT. 16) WRITE(VDUOUT,860)(BATNRS(I),I=1,NUMBAT)
      IF ((NUMBAT .GT. 16) .AND. (NUMBAT .LT. 32))
     1       WRITE(VDUOUT,860)(BATNRS(I),I=1,16)
      IF ((NUMBAT .GT. 16) .AND. (NUMBAT .LT. 32))
     1       WRITE(VDUOUT,860)(BATNRS(I),I=17,NUMBAT)
  840 FORMAT(' ALL 32 BATCHES')
  850 FORMAT(1X,I4,' BATCHES, COMPRISING THE FOLLOWING BATCHES :')
  860 FORMAT(16(1X,I3,1X))
      IF (WINDOW .EQ. 0) WRITE(VDUOUT,900)
      IF (WINDOW .EQ. 1) WRITE(VDUOUT,920)ALPHA
      IF (WINDOW .EQ. 2) WRITE(VDUOUT,930)
      IF ((WINDOW .EQ. 8) .OR. (WINDOW .EQ. 9)) WRITE(VDUOUT,940)WINDOW
  900 FORMAT(' COSINE-BELL WINDOW')
  920 FORMAT(' KAISER-BESEL WINDOW WITH ALPHA = ',F5.2)
  930 FORMAT(' MJN ''TAPER2'' WINDOW')
  940 FORMAT(' WINDOW = ',I1)
      IF (IMEAN .EQ. 0) WRITE(VDUOUT,950)
      IF (IMEAN .EQ. 1) WRITE(VDUOUT,960)
  950 FORMAT(' MEAN OF WINDOWED DATA TO BE INCLUDED')
  960 FORMAT(' MEAN OF WINDOWED DATA TO BE REMOVED')
  970 WRITE(VDUOUT,980)
  980 FORMAT('  -  ENTER ''Y'' OR ''N''')
      READ(KYBDIN,1040,ERR=970)ANSWER
      IF (ANSWER .EQ. INO) GO TO 100
      IF (ANSWER .NE. IYES) GO TO 970
      RETURN
C
C  MULTI-REFERENCED FORMAT STATEMENTS.
 1020 FORMAT(I1)
 1040 FORMAT(A1)
      END



C  FILE I.D.  : KAIGEN TPFL-F66                 LAST REV : 16 FEB 87
C
      SUBROUTINE KAIGEN(W,N,ALPHA)
      INTEGER   IDEST,ISORCE,I,NT,NW,N2
      REAL      W(1024),PI,ALPHA,X
      REAL      TEMP,BSSLPI
      DATA      PI /3.141592654/
      N2 = N/2
      NT = N2 + 1
      CALL SUFACT
      BSSLPI = BSSL(PI*ALPHA)
      DO 130 NW=1,N
         IF (NW .LT. NT) GO TO 110
         X = (FLOAT(N2-NW)/FLOAT(N2))**2
         GO TO 120
  110    X = (FLOAT(NW-N2)/FLOAT(N2))**2
```

```
 120    W(NW) = (BSSL(PI*ALPHA*SQRT(1.0-X))/BSSLPI)*W(NW)
 130    CONTINUE
        TEMP = W(N)
        DO 140 I=2,N
           IDEST = N + 2 - I
           ISORCE = IDEST - 1
           W(IDEST) = W(ISORCE)
 140    CONTINUE
        W(1) = TEMP
        RETURN
        END
```

```
C  FILE I.D. : MAINPLOT  PLOT-F66            LAST REV : 24 MAR 87
C
C  20 MAR 87 : ALTERED SELECTION OF UNITS TO ALLOW REDUCED SIZE PLOTS
C
C         THIS PROGRAM ACTS AS THE INTERFACE BETWEEN THE VARIOUS SUB-
C    ROUTINES. IT MAKES USE OF THE GINO-F GARPHICS PACKAGE. IF A TEK-
C    TRONIX TERMINAL WAS THE NOMINATED GRAPHICS DEVICE, THEN THE USER
C    IS ASKED WHETHER A CALCOMP PLOT OF THE PREVIOUSLY DISPLAYED
C    SCREEN(S) IS (ARE) REQUIRED.
C
C         THE SUBROUTINES USED, WITH A BRIEF DESCRIPITION, ARE :
C
C    INPUT  - INTEROGATES THE USER FOR NAME OF DATA FILE TO BE USED,
C             PROCESSING OPTIONS TO BE CARRIED OUT, AND OUTPUT
C             GRAPHICS DEVICE ( TEKTRONIX 4010  OR  CALCOMP 1051 )
C    PROCES - READS DATA FROM NOMINATED FILE AND PROCESSES IT ACCORD-
C             ING TO SPECIFIED OPTIONS
C    DATOUT - RESPONSIBLE FOR PRODUCING GRAPHICAL DISPLAY OF DATA OR
C             WINDOWED DATA (I.E. BEFORE OR AFTER THE DATA HAS BEEN
C             WINDOWED)
C    PSDOUT - PRODUCES GRAPHICAL DISPLAY OF PRE- AND POST- STIMULUS
C             ENERGY SPECTRAL DENSITY
C    PHAOUT - PRODUCES GRAPHICAL DISPLAY OF PRE- AND POST- STIMULUS
C             PHASE PLOT V. FREQUENCY.
C
       INTEGER  BLINE,IDEV,NUMBAT,TDLEN,NPOINT,IYES,INRVAL,HARNUM
       INTEGER  IFILT,STIM1,STIM2,PRES,POSS,NPANLY,WINDOW,SIZE
       INTEGER  FNAME(20),BATNRS(32),PLOTAN(4),EYETYP,LOGTYP,PLTYPE
       INTEGER  IMEAN,I,IMODEL
       LOGICAL  IRDUCE
       REAL     ECORR,SAMRAT,STMINT,HSIZE,VSIZE,ALPHA
       REAL     AVDATA(1024),PREPSD(1024),POSPSD(1024),TAPDAT(1024)
       REAL     PREANG(1024),POSANG(1024)
       REAL     TEXUNI,CALUNI
       COMMON   /IFLSEL/ IFILT
       DATA     IYES  /'Y'/
       DATA     TEXUNI,CALUNI,IRDUCE /0.18,0.25,.TRUE./
C
C  READ IN DESIRED PROCESSING OPTIONS.
       CALL RETYPE
C
       CALL INPUT(FNAME,IDEV,ECORR,BLINE,IFILT,PLOTAN,BATNRS,NUMBAT,
      1     WINDOW,PLTYPE,EYETYP,LOGTYP,IMEAN,IMODEL,ALPHA)
C
C  PERFORM REQUIRED PROCESSING OPTIONS.
C
       CALL PROCES(FNAME,ECORR,BLINE,IFILT,BATNRS,NUMBAT,AVDATA,PREPSD,
      1     POSPSD,INRVAL,SAMRAT,STIM1,STIM2,PRES,POSS,NPANLY,TDLEN,
      2     NPOINT,WINDOW,ALPHA,PLOTAN,TAPDAT,EYETYP,IMEAN,IMODEL,
      3     PREANG,POSANG)
       STMINT = FLOAT(INRVAL)/SAMRAT
       SIZE = TDLEN
       HARNUM = TDLEN
C
C  SELECTS REQUIRED GRAPHCS DEVICE AND SIZE OF UNITS IN MM.
C
       IF (IRDUCE) CALL REDUCT(TEXUNI,CALUNI,NSIZE)
       IF (IDEV .EQ. 1) GO TO 10
       CALL T4010
       CALL UNITS(TEXUNI)
       IF (NSIZE .NE. 0) CALL CHASIZ(14.,14.)
```

```
      GO TO 20
   10 CALL C1051N
      CALL UNITS(CALUNI)
      CALL CHASIZ(12.,12.)
C
C  DEPENDING ON SELECTED PLOTTING OPTINS PRODUCES GRAPHICAL DISPLAY OF
C  AVERAGED DATA, WINDOWED DATA OR ENERGY SPCTRAL DENSITY OR ANY
C  COMBINATION OF THESE.
C
   20 IF (PLOTAN(1) .EQ. IYES) CALL DATOUT(FNAME,STMINT,SAMRAT,ECORR,
     1    BLINE,IFILT,STIM1,STIM2,PRES,POSS,NPANLY,TDLEN,WINDOW,IDEV,
     2    AVDATA,NUMBAT,BATNRS,ALPHA,1,EYETYP,IMEAN,IMODEL)
      IF (PLOTAN(2) .EQ. IYES) CALL DATOUT(FNAME,STMINT,SAMRAT,ECORR,
     1    BLINE,IFILT,STIM1,STIM2,PRES,POSS,NPANLY,TDLEN,WINDOW,IDEV,
     2    TAPDAT,NUMBAT,BATNRS,ALPHA,2,EYETYP,IMEAN,IMODEL)
      IF (PLOTAN(3) .EQ. IYES) CALL PSDOUT(FNAME,STMINT,SAMRAT,ECORR,
     1    BLINE,IFILT,STIM1,STIM2,PRES,POSS,NPANLY,TDLEN,WINDOW,IDEV,
     2    PREPSD,POSPSD,SIZE,NUMBAT,BATNRS,ALPHA,PLTYPE,3,EYETYP,
     3    IMEAN,IMODEL,LOGTYP)
      IF (PLOTAN(4) .EQ. IYES) CALL PHAOUT(FNAME,STMINT,SAMRAT,ECORR,
     1    BLINE,IFILT,STIM1,STIM2,PRES,POSS,NPANLY,TDLEN,WINDOW,IDEV,
     2    PREANG,POSANG,SIZE,NUMBAT,BATNRS,ALPHA,PLTYPE,4,EYETYP,
     3    IMEAN,IMODEL,LOGTYP)
C
C  IF TEKTRONIX TERMIANL USED PROMPT USER AS TO WHETHER A CALCOMP
C  PLOT OF VIEWED SCREENS REQUIRED.
C
      HSIZE = 14.
      VSIZE = 20.
      IF (IDEV .EQ. 1) GO TO 30
      CALL PICCLE
      CALL MOVTO2(0.,0.)
      CALL MOVBY2(11.*HSIZE,17.*VSIZE)
      CALL CHAHOL(49HDO YOU WANT A CALCOMP PLOT OF DISPLAYED SCREENS*.)
      CALL MOVBY2(-47.*HSIZE,-2.*VSIZE)
      CALL CHAHOL(19HYES (1) OR NO (0)*.)
      CALL MOVTO2(0.,0.)
      CALL CHAHOL(10HRESPONSE*.)
      CALL CURSOR(I,X,Y)
      IF (I .NE. 1) GO TO 30
      CALL DEVEND
C
C  PRODUCE CALCOMP PLOT OF DISPLAYED SCRENN(S).
C
      IDEV = 1
      CALL C1051N
      CALL UNITS(CALUNI)
      CALL CHASIZ(12.,12.)
      IF (PLOTAN(1) .EQ. IYES) CALL DATOUT(FNAME,STMINT,SAMRAT,ECORR,
     1    BLINE,IFILT,STIM1,STIM2,PRES,POSS,NPANLY,TDLEN,WINDOW,IDEV,
     2    AVDATA,NUMBAT,BATNRS,ALPHA,1,EYETYP,IMEAN,IMODEL)
      IF (PLOTAN(2) .EQ. IYES) CALL DATOUT(FNAME,STMINT,SAMRAT,ECORR,
     1    BLINE,IFILT,STIM1,STIM2,PRES,POSS,NPANLY,TDLEN,WINDOW,IDEV,
     2    TAPDAT,NUMBAT,BATNRS,ALPHA,2,EYETYP,IMEAN,IMODEL)
      IF (PLOTAN(3) .EQ. IYES) CALL PSDOUT(FNAME,STMINT,SAMRAT,ECORR,
     1    BLINE,IFILT,STIM1,STIM2,PRES,POSS,NPANLY,TDLEN,WINDOW,IDEV,
     2    PREPSD,POSPSD,SIZE,NUMBAT,BATNRS,ALPHA,PLTYPE,3,EYETYP,
     3    IMEAN,IMODEL,LOGTYP)
      IF (PLOTAN(4) .EQ. IYES) CALL PHAOUT(FNAME,STMINT,SAMRAT,ECORR,
     1    BLINE,IFILT,STIM1,STIM2,PRES,POSS,NPANLY,TDLEN,WINDOW,IDEV,
     2    PREANG,POSANG,SIZE,NUMBAT,BATNRS,ALPHA,PLTYPE,4,EYETYP,
     3    IMEAN,IMODEL,LOGTYP)
   30 CALL DEVEND
      STOP
      END



C  FILE I.D. : MAXVAL   OAR-F66
C
      SUBROUTINE MAXVAL(BBEG,BEND,DIN,N,NGRF)
      DIMENSION DIN(10240)
C
C SUBROUTINE TO ASSIGN THE START OF Y-AXIS (BBEG) TO
C THE MAX NEG DATA VALUE AND THE END OF Y-AXIS (BEND)
```

```
C TO THE MAX POSITIVE DATA VALUE.
C
      BBEG = 0.0
      BEND = 0.0
      NTOT = N*NGRF
      DO 10 I=1,NTOT
         IF (DIN(I) .LT. BBEG)BBEG = DIN(I)
         IF (DIN(I) .GT. BEND)BEND = DIN(I)
   10    CONTINUE
      WRITE(6,20)BBEG,BEND
   20 FORMAT(' BBEG,BEND',2F12.4)
      RETURN
      END



C  FILE I.D. : MEAN  CNV-F66
C
      SUBROUTINE MEAN(NPTS,DATA,RMEAN)
      DIMENSION DATA(NPTS)
      RMEAN=0.
      DO 20 I=1,NPTS
          RMEAN=RMEAN+DATA(I)
   20     CONTINUE
      RMEAN=RMEAN/FLOAT(NPTS)
      DO 30 I=1,NPTS
          DATA(I)=DATA(I)-RMEAN
   30     CONTINUE
      RETURN
      END



C  FILE I.D. : MODEL  COMM-F66                     LAST REV : 30 APR 87
C
C       THIS PROGRAM UNIT INITIALISES THE COMPONENTS TO BE USED IF
C  'MODELLING' IS REQUIRED. IT DETERMINES WHICH FORM OF MODEL IS RE-
C  QUIRED (FOR CNV OR ERA DATA) AND, IF CNV, WHETHER 1S OR 4S ISI
C  MODEL IS NEEDED. IT ALSO REQUIRES FROM THE CALLING PROGRAM UNIT
C  THREE FLAGS INDICATING WHETHER CNV OR ERA DATA IS BEING PROCESSED,
C  AND (IF CNV DATA) THE ISI VALUE.
C
      SUBROUTINE MODEL(FNAME,CMPNT1,CMPNT2,CMPNT3,IMODEL)
      INTEGER FNAME(20),ISIVAL,CHAR1,CHAR2,IFILT
      INTEGER VDUOUT,KBDIN,INDEX,IMODEL
      LOGICAL CNVFLG,ERAFLG
      COMMON  /FLAGS/ CNVFLG,ERAFLG,ISIVAL
      COMMON  /IFLSEL/ IFILT
      REAL     CMPNT1(1024),CMPNT2(1024),CMPNT3(1024)
      DATA     CHAR1,CHAR2,KBDIN,VDUOUT /'1','2',5,6/
C
C  CHECK IF IMODEL CORRESPONDS TO VALID MODEL TYPE.
C
      IF (IMODEL .NE. 10) GO TO 9000
      DO 130 I=1,1024
         CMPNT1(I) = 0.
         CMPNT2(I) = 0.
         CMPNT3(I) = 0.
  130    CONTINUE
      IF ((CNVFLG) .AND. (ISIVAL .EQ. 1)) INDEX = 407 + 32 - 21
      IF ((CNVFLG) .AND. (ISIVAL .EQ. 4)) INDEX = 267 + 32 - 21
      IF (ERAFLG) INDEX = 203 - 21
      IF (IFILT .EQ. 1)INDEX = INDEX + 21
      IF (ERAFLG) GO TO 800
      IF (CNVFLG .AND. (ISIVAL .EQ. 4)) GO TO 500
C
C  INITIALISE COMPONENTS FOR 1S CNV MODEL.
C
C     DO 300 I=1,93
      DO 300 I=1,46
         J = I - 1
         CMPNT1(INDEX+I) = -1.E-6*FLOAT(J)
  300    CONTINUE
      DO 400 I=47,93
         J = I - 47
```

```
          CMPNT1(INDEX+I) = CMPNT1(INDEX+46)
          CMPNT2(INDEX+I) = -1.E-6*FLOAT(J)
  400     CONTINUE
      GO TO 1500
C
C  INITIALISE COMPONENTS FOR 48 CNV MODEL.
C
C 500 DO 600 I=1,250
  500 DO 600 I=1,234
          J = I - 1
          CMPNT1(INDEX+I) = -1.E-6*FLOAT(J)
  600     CONTINUE
C     DO 700 I=251,500
      DO 700 I=235,468
          J = I - 235
          CMPNT1(INDEX+I) = CMPNT1(INDEX+234)
          CMPNT2(INDEX+I) = -1.E-6*FLOAT(J)
  700     CONTINUE
      GO TO 1500
C
C  INITALISE COMPONENTS FOR ERA MODEL.
C
  800 DO 900 I=1,8
          J = I - 1
          CMPNT1(INDEX+I) = -1.E-6*FLOAT(J)
  900     CONTINUE
      DO 1000 I=9,24
          J = I - 9
          CMPNT1(INDEX+I) = CMPNT1(8)
          CMPNT2(INDEX+I) = 1.E-6*FLOAT(J)
 1000     CONTINUE
      DO 1100 I=25,32
          J = I - 25
          CMPNT1(INDEX+I) = CMPNT1(8)
          CMPNT2(INDEX+I) = CMPNT2(24)
          CMPNT3(INDEX+I) = -1.E-6*FLOAT(J)
 1100     CONTINUE
 1500 RETURN
C
C  ERROR HANDLERS.
C
 9000 WRITE(VDUOUT,9020)
 9020 FORMAT(//,' INVALID MODEL TYPE REQUESTED, VALUE IS ',I2,
     1   //,' *** STOP IN SUBROUTINE ''MODEL'' ***')
      STOP
      END



C  FILE I.D. : MODDAG  OAR-F66             LAST REV : 9 FEB 87
C
C  A PROGRAM TO PRODUCE PLOTS OF MODEL COMPONENTS.
C
      SUBROUTINE MODDAG(CMPNT1,CMPNT2,CMPNT3,N)
      INTEGER GRFTYP(3),ISIVAL
      LOGICAL CNVFLG,ERAFLG
      COMMON /XVAL/ X(1024)
      COMMON /FLAGS/ CNVFLG,ERAFLG,ISIVAL
      REAL  CMPNT1(1024),CMPNT2(1024),CMPNT3(1024)
      DATA ICMPA,ICMPB,ICMP1,ICMP2,ICMP3 /'CM','PN','T1','T2','T3'/
      CALL PICCLE
C  PLOT MODEL COMPONENT 1.
      GRFTYP(1) = ICMPA
      GRFTYP(2) = ICMPB
      GRFTYP(3) = ICMP1
      CALL NGRAPH(N,X,CMPNT1,1,0,GRFTYP)
C  PLOT MODEL COMPONENT 2.
      GRFTYP(3) = ICMP2
      CALL NGRAPH(N,X,CMPNT2,2,0,GRFTYP)
      IF ((CNVFLG) .AND. (.NOT.(ERAFLG))) GO TO 1000
C  PLOT MODEL COMPONENT 3.
      GRFTYP(3) = ICMP3
      CALL NGRAPH(N,X,CMPNT3,3,0,GRFTYP)
 1000 RETURN
      END
```

```
C  FILE I.D. : MODTHG OAR-F66            LAST REV : 14OCT 86
C
C  A PROGRAM TO PRODUCE PLOTS OF THETAS FOR MODEL COMPONENTS.
C
      SUBROUTINE MODTHG(THETA,N,NGRF,NSIZE,BBEG,BEND)
      INTEGER NGRF,NSIZE
      REAL THETA(NSIZE)
      COMMON /XVAL/ X(1024)
      ISCALE = 3
      CALL PICCLE
      CALL NGRFS1(N,NGRF,X,THETA,4,3,ISCALE,BBEG,BEND)
      RETURN
      END
```

```
C  FILE I.D. : NAMCON  COMM-F66                    LAST REV : JUN 85
C
C  A SUBROUTINE TO CONVERT NAME OF THE FILE TO BE PROCESSED FROM A FORM
C  CONSISTANT WITH THE ORIGINAL M.J.N. PROGRAMS TO ONE WHICH ALLOWS THE
C  USE OF THE CMS SUBROUTINE :
C       CALL FILEDF(RETCD,UNIT,DEVICE,FNAME,FTYPE,FMODE,OPTS)
C  THE CALL TO 'LBMOVE' IS TO A VM/CMS SYSTEM SUBROUTINE, SEE SCP
C  COMPUTER SERVICES DOCUMENT V2/1.6 'VM/CMS SUBROUTINES'
C
      SUBROUTINE NAMCON(FNAME,NAME)
      INTEGER NAME(20),I
      REAL*8 FNAME
      DO 10 I=1,8
          CALL LBMOVE(NAME(I),1,FNAME,I,1)
   10     CONTINUE
      RETURN
      END
```

```
C  FILE I.D. : NAMTST  COMM-F66                    LAST REV : 9 FEB 87
C
C       THIS PROGRAM UNIT DETERMINES WHETHER FILE NAME CORRESPONDS TO
C  CNV OR ERA DATA BY INSPECTING FIRST THREE CHARACTERS OF FILE NAME. IF
C  SO THE APPROPRIATE FLAG IS SET TRUE, ELSE BOTH FLAGS ARE RETURNED
C  FALSE TO THE CALLING PROGRAM UNIT.
C
      SUBROUTINE NAMTST(FNAME)
      INTEGER FNAME(20),CNVTST(3),ERATST(3),IENQRY,VDUOUT,CHAR1,CHAR2
      INTEGER ISIVAL
      LOGICAL CNVFLG,ERAFLG
      COMMON  /FLAGS/  CNVFLG,ERAFLG,ISIVAL
      DATA    VDUOUT,KBDIN,CHAR1,CHAR2 /6,5,'1','2'/
      DATA    CNVTST,ERATST /'C','N','V','E','R','A'/
      CNVFLG = .FALSE.
      ERAFLG = .FALSE.
      IF ((FNAME(7) .NE. CHAR1) .AND. (FNAME(7) .NE. CHAR2)) GO TO 9100
      IF ((CNVTST(1) .EQ. FNAME(1)) .AND. (CNVTST(2) .EQ. FNAME(2))
     1    .AND. (CNVTST(3) .EQ. FNAME(3))) CNVFLG = .TRUE.
      IF ((ERATST(1) .EQ. FNAME(1)) .AND. (ERATST(2) .EQ. FNAME(2))
     1    .AND. (ERATST(3) .EQ. FNAME(3))) ERAFLG = .TRUE.
      IF (.NOT.(CNVFLG .OR. ERAFLG)) GO TO 9000
      IF (CNVFLG .AND. ERAFLG) GO TO 9200
      IF (FNAME(7) .EQ. CHAR1) ISIVAL = 1
      IF (FNAME(7) .EQ. CHAR2) ISIVAL = 4
  120 RETURN
C
C  ERROR HANDLERS.
C
 9000 WRITE(VDUOUT,9020)
 9020 FORMAT(//,' NON CNV/ERA DATA (NAME DOESN''T START WITH ',
     1  '''CNV'' OR ''ERA''), ''1'' TO CONTINUE, ''9'' TO ABORT')
      READ(KBDIN,9040,ERR=9000)IENQRY
 9040 FORMAT(I1)
      IF ((IENQRY .NE. 1) .AND. (IENQRY .NE. 9)) GO TO 9000
      IF (IENQRY .EQ. 1) GO TO 120
      GO TO 9900
 9100 WRITE(VDUOUT,9120)
 9120 FORMAT(' NON CNV/ERA DATA (NAME DOESN''T CONTAIN A ''1'' OR'
```

```
     1         ,' ''2'' AT EXPECTED LOCATION), ''1'' TO CONTINUE'
     2         ,' ''9'' TO ABORT')
      READ(KBDIN,9140,ERR=9000)IENQRY
 9140 FORMAT(I1)
      IF ((IENQRY .NE. 1) .AND. (IENQRY .NE. 9)) GO TO 9100
      IF (IENQRY .EQ. 1) GO TO 120
 9200 WRITE(VDUOUT,9220)
 9220 FORMAT(' ERROR IN ''NAMTST'', CNVFLG AND ERAFLG BOTH TRUE')
 9900 WRITE(VDUOUT,9920)
 9920 FORMAT(//,' ***  STOP IN SUBROUTINE NAMTST ***')
      STOP
      END


C  FILE I.D. : NGRAPH  DAR-F66      LAST REV : 8 OCT 86
C
      SUBROUTINE NGRAPH(N,X,Y,ING,IST,GRFTYP)
C
C ROUTINE TO PLOT A GRAPH IN ONE OF SIX PARTS OF
C A PAGE. THUS UPTO SIX GRAPHS CAN BE PLOTTED ON
C ONE PAGE IF CALLED SIX TIMES
C
C  X    HOLDS X-AXIS DATA
C  Y    HOLDS DATA TO BE PLOTTED
C  ING  SPECIFIES PART OF THE PAGE TO PLOT GRAPH
C  IST  SPECIFIES THE Y-AXIS LABEL
C
      DIMENSION X(N),Y(N)
      INTEGER ILAB1(11),ILAB2(2),ILAB3(2),GRFTYP(3)
      DATA ILAB1 /3*'  ',' A','MP','LI','TU','DE',' (','UV',') '/
      DATA ILAB2,ILAB3 /'AC','F ','CC','F '/
      ILAB1(1) = GRFTYP(1)
      ILAB1(2) = GRFTYP(2)
      ILAB1(3) = GRFTYP(3)
      X1 = 0.0
      IF (MOD(ING,2) .EQ. 1) X1 = 110.0
      Y1 = 57.0
      IF (ING .LT. 2) Y1 = 115.0
      IF (ING .GT. 3) Y1 = 0.0
      X2 = X1 + 100.0
      Y2 = Y1 + 70.0
      IF (ING .GT. 0) GO TO 15
      CALL PICCLE
   15 CALL WINDO2(X1,X2,Y1,Y2)
      CALL CHASIZ(2.4,2.4)
      CALL GRAF(X,Y,N,0)
      AX1 = X2 - 50.0
      AX2 = X1 + 10.0
      AY1 = Y1 + 5.0
      AY2 = Y1 + 10.0
      CALL MOVTO2(AX1,AY1)
      CALL CHAHOL('TIME (SEC)*.')
      CALL CHAANG(90.0)
      CALL MOVTO2(AX2,AY2)
      IF (IST .EQ. 0)CALL CHAARR(ILAB1,11,2)
      IF (IST .EQ. 1)CALL CHAARR(ILAB2,2,2)
      IF (IST .EQ. 2)CALL CHAARR(ILAB3,2,2)
      CALL CHAANG(0.)
      RETURN
      END


C  FILE I.D. : NGRFS1 DAR-F66               LAST REV : 15 OCT 86
C
      SUBROUTINE NGRFS1(N,NGRF,X,DIN,ING,IST,ISCALE,BBEG,BEND)
C
C     THIS SUBROUTINE PLOTS MULTIPLE CURVES ON A SET
C     OF AXES WITH SELECTABLE Y-AXIS SCALING AND POSITION OF
C     ORIGIN. THE PART OF THE PAPER TO PLOT THE GRAPHS IS
C     ALSO SELECTABLE.
C
C  N       -THE NUMBER OF POINTS PER CURVE
C  NGRF    -THE NUMBER OF CURVES TO BE PLOTTED
```

```
C  X        -CONTAINS THE VALUES FOR THE X-AXIS
C  DIN      -CONTAINS ALL THE DATA POINTS TO BE PLOTTED
C            WITH DATA FOR CURVE 1 OCCUPYING THE FIRST N POSITIONS
C  ING      -DETERMINES WHAT PART OF THE PAPER THE GRAPHS
C            ARE PLOTTED ON
C  IST      -DETERMINES THE Y-AXIS LABELLING
C  ISCALE  -DETERMINES THE TYPE OF LABELLING FOR Y-AXIS
C
      DIMENSION Y(1025),DIN(10240),X(1025)
      INTEGER ILAB0(4),ILAB1(2),ILAB2(2),ILAB3(9),ILAB4(5)
      INTEGER ILAB5(5),ILAB6(8),ILAB7(7)
      INTEGER ITYPE,ICOL,IPEN(4)
      REAL    WIDTH
      DATA ILAB0,ILAB1,ILAB2 /'AM','P(','UV',') ','AC','F ','CC','F '/
      DATA ILAB3 /'ES','TM','TE','D ','PA','RA','ME','TE','RS'/
      DATA ILAB4 /'ER','RO','R ','NO','RM'/
      DATA ILAB5 /'PA','RA','ME','TE','RS'/
      DATA ILAB6 /'SA','MP','LE',' V','AR','IA','NC','E '/
      DATA ILAB7 /'MU','LT',' C','OR',' C','OE','FF'/
      DATA IPEN(1),IPEN(2),IPEN(3),IPEN(4) /1,2,5,7/
      DATA WIDTH,ITYPE /0.0,0/
C
C DETERMINE THE PART OF THE PAGE TO PLOT GRAPH
C
      X1 = 0.0
      IF (MOD(ING,2) .EQ. 1) X1 = 110.0
      Y1 = 57.0
      IF (ING .LT. 2) Y1 = 115.0
      IF (ING .GT. 3) Y1 = 0.0
      X2 = X1 + 100.0
      Y2 = Y1 + 70.0
      CALL WINDO2(X1,X2,Y1,Y2)
      CALL CHASIZ(2.4,2.4)
      NLEN = 8
C
C DETERMINE THE RANGE OF DATA TO PLOT
C
      IF (X(1) .GE. 0.0) GO TO 50
      IOR = 0
      X3 = X1 + 50.0
      Y3 = Y1 + 35.0
      VBEG = -5.0
      VEND = 5.0
      NINTS = 10
      AX1 = X2 - 7.0
      AX2 = X3
      AY1 = Y3 - 5.0
      AY2 = Y2 - 6.0
      GO TO 60
  50  IOR = 1
      X3 = X1 + 20.0
      Y3 = Y1 + 17.0
      VBEG = 0.0
      VEND = 9.0
      NINTS = 9
      AX1 = X3 + 20.0
      AX2 = X3 - 15.0
      AY1 = Y1 + 8.0
      AY2 = Y3
      ISCLEX = 3
      ISCLEY = 2
C
C POSITION AXES, DRAW AND LABEL THEM
C
  60  CALL AXIPOS(IOR,X3,Y3,90.0,1)
      CALL AXIPOS(IOR,X3,Y3,50.0,2)
      CALL AXISCA(ISCLEX,NINTS,VBEG,VEND,1)
      CALL AXISCA(ISCLEY,NLEN,BBEG,BEND,2)
      CALL AXIDRA(2,1,1)
      CALL AXIDRA(-2,-1,2)
      CALL MOVTO2(AX1,AY1)
      CALL CHAHOL('TIME (SEC)*.')
      CALL MOVTO2(AX2,AY2)
      CALL CHAANG(90.0)
C
```

```
C PLOT THE CURVES
C
      IF (IST .EQ. 0)CALL CHAARR(ILAB0,4,2)
      IF (IST .EQ. 1)CALL CHAARR(ILAB1,2,2)
      IF (IST .EQ. 2)CALL CHAARR(ILAB2,2,2)
      IF (IST .EQ. 3)CALL CHAARR(ILAB3,9,2)
      IF (IST .EQ. 4)CALL CHAARR(ILAB4,5,2)
      IF (IST .EQ. 5)CALL CHAARR(ILAB5,5,2)
      IF (IST .EQ. 6)CALL CHAARR(ILAB6,8,2)
      IF (IST .EQ. 7)CALL CHAARR(ILAB7,7,2)
      CALL CHAANG(0.0)
      DO 70 I=1,NGRF
         CALL PENSEL(IPEN(I),WIDTH,ITYPE)
         CALL NPLOT(DIN,N,Y,I)
         CALL GRACUR(X,Y,N)
  70     CONTINUE
      CALL PENSEL(1,WIDTH,ITYPE)
      RETURN
      END
```

```
C  FILE I.D. : NLOGN  CNV-F66
C
      SUBROUTINE NLOGN(N,X,SIGN)
C              THIS PROGRAM PERFORMS THE FFT.
C              N=BASE  2 LOG OF NO. OF POINTS.
C              X= COMPLEX ARRAY OF DATA FOR TRANSFORMATION.
C              SIGN= -1. FOR FFT.
C              SIGN= +1. FOR IFFT.
C              TRANSFORMED DATA IS RETURNED IN X.
      DIMENSION M(12)
      COMPLEX WK,HOLD,Q,X(2048)
      LX=2**N
      DO 1 I=1,N
    1 M(I)=2**(N-I)
      DO 4 L=1,N
      NBLOCK=2**(L-1)
      LBLOCK=LX/NBLOCK
      LBHALF=LBLOCK/2
      K=0
      DO 4 IBLOCK=1,NBLOCK
      FK=K
      FLX=LX
      V=SIGN*6.283185308*FK/FLX
      WK=CMPLX(COS(V),SIN(V))
      ISTART=LBLOCK*(IBLOCK-1)
      DO 2 I=1,LBHALF
      J=ISTART+I
      JH=J+LBHALF
      Q=X(JH)*WK
      X(JH)=X(J)-Q
      X(J)=X(J)+Q
    2 CONTINUE
      DO 3 I=2,N
      II=I
      IF(K.LT.M(I))GO TO 4
    3 K=K-M(I)
    4 K=K+M(II)
      K=0
      DO 7 J=1,LX
      IF(K.LT.J)GO TO 5
      HOLD=X(J)
      X(J)=X(K+1)
      X(K+1)=HOLD
    5 DO 6 I=1,N
      II=I
      IF(K.LT.M(I))GO TO 7
    6 K=K-M(I)
    7 K=K+M(II)
      IF(SIGN.LT.0.)RETURN
      DO 8 I=1,LX
    8 X(I)=X(I)/FLX
      RETURN
      END
```

```
C  FILE I.D. : NPLOT  OAR-F66
C
      SUBROUTINE NPLOT(DIN,N,Y,IST)
C
C  THIS SUBROUTINE ALLOWS N CURVES TO BE PLOTTED
C  ON ONE GRAPH
C
      DIMENSION Y(1025),DIN(10240)
      I1 = (IST-1)*N
      DO 10 I=1,N
         Y(I) = DIN(I1+I)
   10    CONTINUE
      RETURN
      END



C  FILE I.D : NRDARM  COMM-F66                      LAST REV : 13 MAR 87
C
      SUBROUTINE NRDARM(FNAME,BATNO,ICHAN,COR1,SF1,SF2,SAMRAT,ECORR,
     1                  IMODEL)
C
C        THIS PROGRAM IS INTENDED TO MINIMISE THE AMOUNT OF E.O.G. POWER
C  IN THE E.E.G.  IT USES THE MODIFIED QUILTER TECHNIQUE.  HORIZONTAL
C  AND VERTICAL COMPONENTS OF BOTH EYES ARE TAKEN INTO CONSIDERATION.
C  THE PROGRAM REMOVES D.C. OFFSET ON ANY OF THE INPUT DATA CHANNELS.
C
      REAL  VL(1024),VR(1024),HL(1024),HR(1024),E1(1024),COR1(1024)
      REAL  CMPNT1(1024),CMPNT2(1024),CMPNT3(1024),VECX1(49)
      REAL  PVL,PVR,PHL,PHR,PN1,PN2,PN3
      REAL  B,CCL,C,D,CCR,A
      REAL  N1VL,N2VL,N3VL,N1VR,N2VR,N3VR
      REAL  N1HL,N2HL,N3HL,N1HR,N2HR,N3HR
      REAL  N12,N13,N23
      REAL  MVL,MVR,MHL,MHR,MN1,MN2,MN3
      DIMENSION X1(7,7),RM1(7),RHS(7)
      DOUBLE PRECISION DRHS(7),DVECX1(49)
      INTEGER BATNO,INP(1024),FNAME(20),ISIVAL
      INTEGER I,L,N,VDUOUT,KBDIN,ACTROW,ACTCOL,DIMROW,DIMCOL
      LOGICAL CSE,DFILOP
      LOGICAL CNVFLG,ERAFLG
      COMMON /FLAGS/ CNVFLG,ERAFLG,ISIVAL
      COMMON  /DTFLST/  DFILOP
      DATA CSE /.FALSE./
      DATA KBDIN,VDUOUT /5,6/
C  INITIALISE CORRELATION SUMS OF PRODUCTS TO ZERO.
      DATA  PVL,B,CCL,C,N1VL,N2VL,N3VL   /7*0./
      DATA  PVR,D,CCR,N1VR,N2VR,N3VR     /6*0./
      DATA  PHL,A,N1HL,N2HL,N3HL         /5*0./
      DATA  PHR,N1HR,N2HR,N3HR           /4*0./
      DATA  PN1,N12,N13                  /3*0./
      DATA  PN2,N23                      /2*0./
      DATA  PN3                          /0./
      DATA  MVL,MVR,MHL,MHR,MN1,MN2,MN3  /7*0./
      N=1024
      IF (IMODEL .EQ. 10)CALL MODEL(FNAME,CMPNT1,CMPNT2,CMPNT3,IMODEL)
C
C       THE DATA IS ASSUMED TO BE IN THE FOLLOWING ORDER -
C
C                VL , VR , HL , HR , M1 , M2
C
C  M1 IS THE CHANNEL TO BE CORRECTED BY THE MODIFIED QUILTER TECHNIQUE.
C  M2 IS THE CHANNEL CORRECTED BY THE BURDEN TECHNIQUE.
C  SF1 FOR EOG DATA. SF2 FOR EEG DATA.
C
C             CHECK THAT THE BATCH NUMBER IS VALID.
      IF (BATNO .LT. 0 .OR. BATNO .GT. 191)GO TO 2999
      IF(FLOAT(BATNO/6) .NE. FLOAT(BATNO)/6.)GO TO 2999
      L=BATNO
C             READ THE DATA AND CONVERT TO REAL FORMAT.
      CALL DATIN(L,INP,FNAME,SF1,SF2,SAMRAT,DFILOP)
      DO 22 I=1,N
         VL(I)=FLOAT(INP(I))*SF1*1.E-06
   22    CONTINUE
      L=L+1
```

```
      CALL DATIN(L,INP,FNAME,SF1,SF2,SAMRAT,DFILOP)
      DO 24 I=1,N
          VR(I)=FLOAT(INP(I))*SF1*1.E-06
   24     CONTINUE
      L=L+1
      CALL DATIN(L,INP,FNAME,SF1,SF2,SAMRAT,DFILOP)
      DO 26 I=1,N
          HL(I)=FLOAT(INP(I))*SF1*1.E-06
   26     CONTINUE
      L=L+1
      CALL DATIN(L,INP,FNAME,SF1,SF2,SAMRAT,DFILOP)
      DO 28 I=1,N
          HR(I)=FLOAT(INP(I))*SF1*1.E-06
   28     CONTINUE
      L=L+1
C
C       CHECK WHICH CHANNEL IS TO BE CORRECTED.
C                IF CHANNEL 5 INCREMENT L.
C
      IF(ICHAN .GE. 2)L=L+1
      CALL DATIN(L,INP,FNAME,SF1,SF2,SAMRAT,DFILOP)
      DO 30 I=1,N
          E1(I)=FLOAT(INP(I))*SF2*1.E-06
   30     CONTINUE
C              SUBTRACT THE MEAN OF EACH DATA BATCH FROM THE DATA.
      CALL MEAN(N,E1,E1AV)
      IF(ECORR .NE. 1.)GO TO 1000
      CALL MEAN(N,VL,VLM)
      CALL MEAN(N,VR,VRM)
      CALL MEAN(N,HL,HLM)
      CALL MEAN(N,HR,HRM)
      CALL MEAN(N,CMPNT1,C1AV)
      CALL MEAN(N,CMPNT2,C2AV)
      CALL MEAN(N,CMPNT3,C3AV)
C  SETS UP MODIFIED EOG MODEL 4D (AS USED IN RECURSIVE OAR).
C  I.E. MODEL COMPRISES VR,HL,HR,HL*HR.
      DO 40 I=1,1024
         VL(I) = HL(I)*HR(I)
   40    CONTINUE
C              FORM THE CORRELATION SUMS OF PRODUCTS.
      DO 100 I=1,N
         PVL = PVL + VL(I)**2
         B = B + VL(I)*VR(I)
         CCL = CCL + VL(I)*HL(I)
         C = C + VL(I)*HR(I)
         N1VL = N1VL + VL(I)*CMPNT1(I)
         N2VL = N2VL + VL(I)*CMPNT2(I)
         N3VL = N3VL + VL(I)*CMPNT3(I)
         PVR = PVR + VR(I)**2
         D = D + VR(I)*HL(I)
         CCR = CCR + VR(I)*HR(I)
         N1VR = N1VR + VR(I)*CMPNT1(I)
         N2VR = N2VR + VR(I)*CMPNT2(I)
         N3VR = N3VR + VR(I)*CMPNT3(I)
         PHL = PHL + HL(I)**2
         A = A + HL(I)*HR(I)
         N1HL = N1HL + HL(I)*CMPNT1(I)
         N2HL = N2HL + HL(I)*CMPNT2(I)
         N3HL = N3HL + HL(I)*CMPNT3(I)
         PHR = PHR + HR(I)*HR(I)
         N1HR = N1HR + HR(I)*CMPNT1(I)
         N2HR = N2HR + HR(I)*CMPNT2(I)
         N3HR = N3HR + HR(I)*CMPNT3(I)
         PN1 = PN1 + CMPNT1(I)**2
         N12 = N12 + CMPNT1(I)*CMPNT2(I)
         N13 = N13 + CMPNT1(I)*CMPNT3(I)
         PN2 = PN2 + CMPNT2(I)**2
         N23 = N23 + CMPNT2(I)*CMPNT3(I)
         PN3 = PN3 + CMPNT3(I)**2
  100    CONTINUE
      DO 200 I=1,N
         MVL = MVL + E1(I)*VL(I)
         MVR = MVR + E1(I)*VR(I)
         MHL = MHL + E1(I)*HL(I)
         MHR = MHR + E1(I)*HR(I)
```

```
         MN1 = MN1 + E1(I)*CMPNT1(I)
         MN2 = MN2 + E1(I)*CMPNT2(I)
         MN3 = MN3 + E1(I)*CMPNT3(I)
  200    CONTINUE
      X1(1,1) = PVL
      X1(1,2) = B
      X1(1,3) = CCL
      X1(1,4) = C
      X1(1,5) = N1VL
      X1(1,6) = N2VL
      X1(1,7) = N3VL
      X1(2,2) = PVR
      X1(2,3) = D
      X1(2,4) = CCR
      X1(2,5) = N1VR
      X1(2,6) = N2VR
      X1(2,7) = N3VR
      X1(3,3) = PHL
      X1(3,4) = A
      X1(3,5) = N1HL
      X1(3,6) = N2HL
      X1(3,7) = N3HL
      X1(4,4) = PHR
      X1(4,5) = N1HR
      X1(4,6) = N2HR
      X1(4,7) = N3HR
      X1(5,5) = PN1
      X1(5,6) = N12
      X1(5,7) = N13
      X1(6,6) = PN2
      X1(6,7) = N23
      X1(7,7) = PN3
      RHS(1) = MVL
      RHS(2) = MVR
      RHS(3) = MHL
      RHS(4) = MHR
      RHS(5) = MN1
      RHS(6) = MN2
      RHS(7) = MN3
C
C          SET UP SYMMETRICAL MATRIX.
      DO 300 I=1,6
         IPL1 = I + 1
         DO 250 J=IPL1,6
            X1(J,I) = X1(I,J)
  250       CONTINUE
  300    CONTINUE
C
C  CHANGE TWO DIMENSIONAL ARRAY TO SINGLE DIMENSION VECTOR AS REQUIRED
C  BY SSP SUBROUTINES.
C
      MODE = 2
      ACTROW = 99
      IF (IMODEL .EQ. 0) ACTROW = 4
      IF ((IMODEL .EQ. 10) .AND. (CNVFLG)) ACTROW = 6
      IF ((IMODEL .EQ. 10) .AND. (ERAFLG)) ACTROW = 7
      ACTCOL = 99
      IF (IMODEL .EQ. 0) ACTCOL = 4
      IF ((IMODEL .EQ. 10) .AND. (CNVFLG)) ACTCOL = 6
      IF ((IMODEL .EQ. 10) .AND. (ERAFLG)) ACTCOL = 7
      IF ((ACTROW .EQ. 99) .OR. (ACTCOL .EQ. 99)) GO TO 9100
      DIMROW = 7
      DIMCOL = 7
      CALL ARRAY(MODE,ACTROW,ACTCOL,DIMROW,DIMCOL,VECX1,X1)
C
C PERFORM SOLUTION OF GENERALISED LINEAR EQUATIONS BY SSP CALL.
C
      NUMEQU = 99
      IF (IMODEL .EQ. 0) NUMEQU = 4
      IF ((IMODEL .EQ. 10) .AND. (CNVFLG)) NUMEQU = 6
      IF ((IMODEL .EQ. 10) .AND. (ERAFLG)) NUMEQU = 7
      IF (NUMEQU .EQ. 99) GO TO 9300
      NUMVEC = 1
C  EPS VALUE FOR DOUBLE PRECISION SSP CALL.
      EPS = 1.E-15
```

```
      DO 310 I=1,49
         DVECX1(I) = DBLE(VECX1(I))
  310    CONTINUE
      DO 315 I=1,7
         DRHS(I) = DBLE(RHS(I))
  315    CONTINUE
      CALL DGELG(DRHS,DVECX1,NUMEQU,NUMVEC,EPS,IER)
      IF (IER .NE. 0) GO TO 9000
      DO 320 I=1,7
         RHS(I) = SNGL(DRHS(I))
  320    CONTINUE
C     IF (IMODEL .EQ. 0)WRITE(VDUOUT,400)(RHS(I),I=1,NUMEQU)
C     IF ((IMODEL .NE. 0) .AND. (CNVFLG))
C    1    WRITE(VDUOUT,410) (RHS(I),I=1,NUMEQU)
C     IF ((IMODEL .NE. 0) .AND. (ERAFLG))
C    1    WRITE(VDUOUT,420) (RHS(I),I=1,NUMEQU)
C 400 FORMAT(//,' K1 = ',F12.5,2X,' K2 = ',F12.5,2X,' K3 = ',F12.5,2X,
C    1      ' K4 = ',F12.5)
C 410 FORMAT(//,' K1 = ',F12.5,2X,' K2 = ',F12.5,2X,' K3 = ',F12.5,2X,
C    1  ' K4 = ',F12.5,2X,'K5 = ',F12.5,' K6 = ',F12.5)
C 420 FORMAT(//,' K1 = ',F12.5,2X,' K2 = ',F12.5,2X,' K3 = ',F12.5,2X,
C    1  ' K4 = ',F12.5,2X,'K5 = ',F8.6,' K6 = ',F12.5,2X,' K7 = ',F12.5)
C
C  APPLY CORRECTION.
C
      DO 600 I=1,N
         COR1(I) = E1(I)-(RHS(1)*VL(I)+RHS(2)*VR(I)+RHS(3)*HL(I)
     1         +RHS(4)*HR(I))
  600    CONTINUE
      GO TO 1100
 1000 DO 1010 I=1,N
         COR1(I) = E1(I)
 1010    CONTINUE
 1100 RETURN
C
C  ERROR MESSAGES
C
 2999 WRITE(6,4000)BATNO
 4000 FORMAT(///,' BATCH NUMBER INCORRECT',I6)
      STOP 8
 9000 WRITE(VDUOUT,9020)IER
 9020 FORMAT(' ERROR IN USE OF SSP SUBROUTINE GELG, ERROR CODE =',I4)
      GO TO 9900
 9100 WRITE(VDUOUT,9120)
 9120 FORMAT(' ERROR IN VALUE OF ''ACTROW'' OR ''ACTCOL''')
      GO TO 9900
 9300 WRITE(VDUOUT,9220)
 9220 FORMAT(' ERROR IN VALUE OF ''NUMEQU''')
 9900 WRITE(VDUOUT,9920)
 9920 FORMAT(//,' *** STOP IN SUBROUTINE NROARM ***')
      STOP
      END



C  FILE I.D. : DAGRAF OAR-F66           LAST REV : 7 OCT 86
C
      SUBROUTINE DAGRAF(D0,D1,D2,D3,D4,D5,N)
      INTEGER IVL,IVR,IHL,IHR,IEEG1,IEEG2,IOA1,IOA2,GRFTYP(3)
      DIMENSION D0(1024),D1(1024),D2(1024)
      DIMENSION D3(1024),D4(1024),D5(1024)
      COMMON /XVAL/X(1024)
      DATA IVL,IVR,IHL,IHR,IEOG1,IEOG2 /'VL','VR','HL','HR',' E','OG'/
      DATA IEEG1,IEEG2,IOA1,IOA2,IBLANK  /'CE','EG','DE','EG','  '/
      CALL PICCLE
C  PLOT DIFFERENCE BETWEEN CORRECTED AND UNCORRECTED EEGS.
      GRFTYP(1) = IOA1
      GRFTYP(2) = IOA2
      GRFTYP(3) = IBLANK
      CALL NGRAPH(N,X,D4,0,0,GRFTYP)
C  PLOT CORRECTECD EEG.
      GRFTYP(1) = IEEG1
      GRFTYP(2) = IEEG2
      CALL NGRAPH(N,X,D5,1,0,GRFTYP)
C  PLOT VL EOG.
```

```
      GRFTYP(1) = IVL
      GRFTYP(2) = IEOG1
      GRFTYP(3) = IEOG2
      CALL NGRAPH(N,X,D0,2,0,GRFTYP)
C  PLOT VR EOG.
      GRFTYP(1) = IVR
      CALL NGRAPH(N,X,D1,3,0,GRFTYP)
C  PLOT HL EOG.
      GRFTYP(1) = IHL
      CALL NGRAPH(N,X,D2,4,0,GRFTYP)
C  PLOT HR EOG.
      GRFTYP(1) = IHR
      CALL NGRAPH(N,X,D3,5,0,GRFTYP)
      RETURN
      END



C  FILE I.D. : ONLSUB  OAR-F66                  LAST REV : 20 OCT 87
C
      SUBROUTINE ONLSUB(PARM,DOUT,RSS,RSQ,OA,NPAR,ITYPE)
C
C XN      VECTOR OF EOG DATA
C THETA   VECTOR OF PARAMETER ESTIMATES
C PS      COVARIANCE MATRIX
C S       SQUARE ROOT VECTOR
C U       VECTOR FOR THE UPPER TRIANGULAR MATRIX
C
      DIMENSION XN(7),PS(7,7),THETA(7),SST(1024),S(28),U(28)
      DIMENSION PARM(7168),DOUT(1024),RSS(1024),RSQ(1024),OA(1024)
      INTEGER FNAME(20)
      COMMON /VDATA/ EEG1(1024),EEG2(1024),FNAME
      COMMON /INTLV/ ALPHA,GAMMA,THETA0
      DATA J1,J2,J3,J4,J5,J6 /1024,2048,3072,4096,5120,6144/
      N = 1024
      NPT = N
      DO 40 I=1,28
         S(I) = 0.0
         U(I) = 0.0
 40      CONTINUE
      DO 50 I=1,7
         DO 55 J=1,7
            PS(I,J) = 0.0
 55         CONTINUE
 50      CONTINUE
      NTIMES = 2
C
C INITALISE THETA AND THE COVARIANCE MATRIX PS
C
      DO 10 I=1,NPAR
         THETA(I) = THETA0
         PS(I,I) = ALPHA
         IK = (I+1)*I/2
         S(IK) = ALPHA
         U(IK) = ALPHA
 10      CONTINUE
      WRITE(6,111)ALPHA,GAMMA,NPAR
 111  FORMAT(1X,'ALPHA=',F6.3,'GAMMA=',F6.3,'NPAR=',I5)
      SF1 = NPT/(NPT-NPAR)
      SF2 = 1.0 - SF1
      RSS1 = 0.0
      SST1 = 0.0
      DF1 = 0.0
C
C REMOVE OCULAR ARTEFACT RECURSIVELY
C
      DO 25 M1=1,NTIMES
         DO 20 I1=1,NPT
            CALL XVAL1(XN,NPAR,I1)
            Y = EEG1(I1)
            IF (ITYPE .EQ. 0)CALL RLSFLT(Y,XN,PS,THETA,NPAR,E,GAMMA,I1)
            IF (ITYPE .EQ. 1)CALL SQRTFL(Y,XN,S,THETA,NPAR,E,GAMMA)
            IF (ITYPE .EQ. 2)CALL UDUFLT(Y,XN,U,THETA,NPAR,E,GAMMA,I1)
            PARM(I1) = THETA(1)
            PARM(I1+J1) = THETA(2)
```

```
            PARM(I1+J2) = THETA(3)
            PARM(I1+J3) = THETA(4)
            PARM(I1+J4) = THETA(5)
            PARM(I1+J5) = THETA(6)
            PARM(I1+J6) = THETA(7)
            DOUT(I1) = E
            OA(I1) = Y-E
            RSS1 = GAMMA*RSS1 + E*E
            SST1 = GAMMA*SST1 + EEG1(I1)*EEG1(I1)
            DF1 = 1.0 + GAMMA*DF1
            RSS(I1) = RSS1/DF1
            SST(I1) = SST1
            IF (ABS(SST1) .GT. 0.000001)RSQ(I1) = (SF2+SF1*RSS1/SST1)
   20       CONTINUE
   25     CONTINUE
        DO 100 I=1,NPT
          EEG1(I) = DOUT(I)
  100     CONTINUE
        RETURN
        END


C  FILE I.D. : POSDET  OAR-F66              LAST REV : 21 AUG 87
C
      SUBROUTINE POSDET(PARMNO,THEVAL,TMPVAL,THEPOS)
      INTEGER   PARMNO,THEPOS(PARMNO)
      REAL      THEVAL(PARMNO),TMPVAL(PARMNO)
C
C  CARRY OUT BUBBLE SORT
C
 1    IF (PARMNO .EQ. 1) GO TO 3000
      DO 1000 I=1,PARMNO
         K = PARMNO - 1
         DO 900 J=1,K
              TEMP = TMPVAL(J)
              IF (TMPVAL(J+1) .LE. TMPVAL(J)) GO TO 900
              TMPVAL(J) = TMPVAL(J+1)
              TMPVAL(J+1) = TEMP
 900          CONTINUE
 1000    CONTINUE
C
C  DETERMINAE THETA VALUE POSITIONS
C
      DO 2000 I=1,PARMNO
         DO 1600 J=1,PARMNO
              IF (TMPVAL(I) .EQ. THEVAL(J)) GO TO 1800
 1600         CONTINUE
 1800    THEPOS(I) = J
 2000    CONTINUE
 3000 IF (PARMNO .EQ. 1) THEPOS(1)=1
      RETURN
      END


C  FILE I.D. : PROCES  PLOT-F66             LAST REV : 13 MAR 87
C
C       THIS SUBROUTINE PERFORMS THE REQUIRED PROCESSSING OPTIONS.
C
C            THE FOLLOWING CHANGES HAVE BEEN MADE TO VARIABLE NAMES
C            BETWEEN IDVRES AND MAINPLOT PROGRAM UNITS.
C
C    IDVRES  , MAINPLOT ,  IDVRES  , MAINPLOT ,  IDVRES  , MAINPLOT
C   -------- , -------- , -------- , -------- , -------- , --------
C    IS1     ,  STIM1   ,  N       ,  NPOINT  ,   MBAT   ,  NUMBAT
C    IS2     ,  STIM2   ,  NP      ,  NP      ,    L1    ,  INDEX
C    ISTRT   ,  ISTART  ,  DEL     ,  INRVAL  ,  LAB1    ,  LABEL
C    PRES1   ,  PREST1  ,  S1      ,  POSST1
C
      SUBROUTINE PROCES(FNAME,ECORR,BLINE,IFILT,BATNRS,NUMBAT,AVDATA,
     1                  PREPSD,POSPSD,INRVAL,SAMRAT,STIM1,STIM2,PRES,
     2                  POSS,NP,TDLEN,NPOINT,WINDOW,ALPHA,PLOTAN,
     3                  TAPDAT,EYETYP,IMEAN,IMODEL,PREANG,POSANG)
      INTEGER   CHANAN,SSR,INRVAL,BLINE,TDLEN,BATNO,PRES,POSS
```

```
      INTEGER   IFILT,STIM1,STIM2,ITL,NPOINT,NP,N1,NUMBAT,IEPOCH
      INTEGER   ISTART,INDEX,ITRIAL,HARNUM,INO,WINDOW
      INTEGER   FNAME(20),BATNRS(32),PLOTAN(4),EYETYP
      INTEGER   IMEAN,IMODEL,VDUOUT,KBDIN,LGTDLN,IHAR,ISIVAL
      REAL      ECORR,ALPHA,SF1,SF2,WISUM1,WISUM3,WINVAL(1024)
      REAL      AVDATA(1024),DATA(1024),TRDAT(1024),TAPDAT(1024)
      REAL      PREPSD(1024),POSPSD(1024),PREANG(1024),POSANG(1024)
      COMPLEX   TDATA(1024),PREST1(1024),POSST1(1024)
      LOGICAL CNVFLG,ERAFLG,MPLFLG,WFILOP
      COMMON /FLAGS/ CNVFLG,ERAFLG,ISIVAL
      COMMON /EPOCH/  IEPOCH
      COMMON /SMPFRQ/ TMPFRQ
      COMMON /WFILST/ WFILOP,WISUM1,WISUM3,WINVAL
      DATA   MPLFLG /.TRUE./
      DATA   CHANAN     /4/
      DATA   SSR,HARNUM,INO           /1,1024,'N'/
      DATA   VDUOUT,KBDIN  /6,5/
      WFILOP = .FALSE.
      TDLEN = 1024
      LGTDLN = 10
      NPOINT = 1024
      NP = 64
      IF (CNVFLG .AND. (ISIVAL .EQ. 1)) NP = 64
      IF (CNVFLG .AND. (ISIVAL .EQ. 4)) NP = 216
      IF (ERAFLG) NP = 32
      ITL = TDLEN/2 + 1
      N1 = NP*SSR
      PRES = 342
      IF (CNVFLG .AND. (ISIVAL .EQ. 1)) PRES = 342
      IF (CNVFLG .AND. (ISIVAL .EQ. 4)) PRES = 22
      IF (ERAFLG) PRES = 168
      POSS = 472
      IF (CNVFLG .AND. (ISIVAL .EQ. 1)) POSS = 472
      IF (CNVFLG .AND. (ISIVAL .EQ. 4) .AND. (IEPOCH .EQ. 1)) POSS = 288
      IF (CNVFLG .AND. (ISIVAL .EQ. 4) .AND. (IEPOCH .EQ. 2)) POSS = 490
      IF (ERAFLG) POSS = 204
      STIM1 = 407
      IF (CNVFLG .AND. (ISIVAL .EQ. 1)) STIM1 = 407
      IF (CNVFLG .AND. (ISIVAL .EQ. 4)) STIM1 = 219
      STIM2 = 532
      IF (CNVFLG .AND. (ISIVAL .EQ. 1)) STIM2 = 532
      IF (CNVFLG .AND. (ISIVAL .EQ. 4)) STIM2 = 719
      INRVAL = 125
      IF (CNVFLG .AND. (ISIVAL .EQ. 1)) INRVAL = 125
      IF (CNVFLG .AND. (ISIVAL .EQ. 4)) INRVAL = 500
      IF (IFILT .EQ. 0) PRES = PRES - 21
      IF (IFILT .EQ. 0) POSS = POSS - 21
      IF (IFILT .EQ. 1) STIM1 = STIM1 + 21
      IF (IFILT .EQ. 1) STIM2 = STIM2 + 21
      DO 100 I=1,NPOINT
         AVDATA(I) = 0.
100      CONTINUE
      DO 250 ITRIAL=1,NUMBAT
         BATNO = BATNRS(ITRIAL)
         WRITE(VDUOUT,110)BATNO
110      FORMAT(' PROCESSING BATCH ',I4)
         IF (EYETYP .EQ. 0) CALL NROARM(FNAME,BATNO,
     1             CHANAN-3,DATA,SF1,SF2,SAMRAT,ECORR,IMODEL)
         IF ((ECORR .EQ. 1.) .AND. (EYETYP .EQ. 1)) CALL RCOARM(FNAME,
     1        BATNO,DATA,IMODEL,MPLFLG)
         DO 220 I=1,NPOINT
            AVDATA(I) = AVDATA(I) + DATA(I)
220         CONTINUE
250      CONTINUE
      DO 300 I=1,NPOINT
         AVDATA(I) = AVDATA(I)/FLOAT(NUMBAT)
300      CONTINUE
      IF (IFILT .EQ. 1) CALL FILTER(NPOINT,AVDATA)
C
C              PRE - STIMULUS
C
      ISTART = PRES - 1
      INDEX = 0
      DO 350 I=1,N1,SSR
         INDEX = INDEX + 1
```

```
         TRDAT(INDEX) = AVDATA(I+ISTART)
  350    CONTINUE
      IF (WINDOW .EQ. 2) CALL TAPER2(TRDAT,NP,IMEAN)
      IF (WINDOW .EQ. 0) CALL TAPCOS(TRDAT,NP,IMEAN)
      IF (WINDOW .EQ. 1) CALL TAPKAI(TRDAT,NP,ALPHA,IMEAN)
      INDEX = 0
      DO 375 I=1,NP
         INDEX = INDEX + 1
         TAPDAT(INDEX+ISTART) = TRDAT(I)
 375     CONTINUE
      DO 400 I=1,TDLEN
         TDATA(I) = CMPLX(0.,0.)
         IF (I .LE. NP) TDATA(I) = CMPLX(TRDAT(I),0.)
  400    CONTINUE
      CALL NLOGN(LGTDLN,TDATA,-1.)
      DO 450 IHAR=1,HARNUM
         PRESTI(IHAR) = TDATA(IHAR)*2./FLOAT(NP)
  450    CONTINUE
C
C                POST - STIMULUS
C
      ISTART = POSS - 1
      INDEX = 0
      DO 500 I=1,N1,SSR
         INDEX = INDEX + 1
         TRDAT(INDEX) = AVDATA(I+ISTART)
  500    CONTINUE
      IF (WINDOW .EQ. 2) CALL TAPER2(TRDAT,NP,IMEAN)
      IF (WINDOW .EQ. 0) CALL TAPCOS(TRDAT,NP,IMEAN)
      IF (WINDOW .EQ. 1) CALL TAPKAI(TRDAT,NP,ALPHA,IMEAN)
      INDEX = 0
      DO 570 I=1,NP
         INDEX = INDEX + 1
         TAPDAT(INDEX+ISTART) = TRDAT(I)
570      CONTINUE
      DO 600 I=1,TDLEN
         TDATA(I) = CMPLX(0.,0.)
         IF (I .LE. NP) TDATA(I) = CMPLX(TRDAT(I),0.)
  600    CONTINUE
      CALL NLOGN(LGTDLN,TDATA,-1.)
      DO 700 IHAR=1,HARNUM
         POSSTI(IHAR) = TDATA(IHAR)*2./FLOAT(NP)
  700    CONTINUE
      IF (EYETYP .EQ. 1) SAMRAT = TMPFRQ
      IF ((PLOTAN(3) .EQ. INO) .AND. (PLOTAN(4) .EQ. INO)) GO TO 900
      DO 800 IHAR=1,HARNUM
         PREPSD(IHAR) = (2./(SAMRAT*FLOAT(NP)))*(CABS(PRESTI(IHAR))**2)
         POSPSD(IHAR) = (2./(SAMRAT*FLOAT(NP)))*(CABS(POSSTI(IHAR))**2)
         PREANG(IHAR) = ATAN2(AIMAG(PRESTI(IHAR)),REAL(PRESTI(IHAR)))
         POSANG(IHAR) = ATAN2(AIMAG(POSSTI(IHAR)),REAL(POSSTI(IHAR)))
  800    CONTINUE
  900 RETURN
      END


C  FILE I.D. : PRTTHT  DAR-F66          LAST REV : 14 OCT 86
C
C       THIS SUBROUTINE APPENDS TO THE PLOTS (OF THETAS, CORRECTED AND
C  CONTAMINATED EEGS, ETC.) INFORMATION DISPLAYING BATCH NUMBER OF TRIAL
C  AND THE FINAL VALUES OF THE THETA PARAMETERS. THE LATTER IS INCLUDED
C  TO HELP IN IDENTIFYING THE DIFFERENT THETAS ON THE PLOTS.
C
      SUBROUTINE PRTTHT(THEPOS,PARMNO,ENDVAL,OFFSET)
      INTEGER    PARMNO,THEPOS(PARMNO),BATNO,FNAME(20),OFFSET
      INTEGER    ICOL,IPEN(4),ITYPE
      REAL       ENDVAL(PARMNO),WIDTH
      COMMON     /TRLNUM/ BATNO
      COMMON     /VDATA/ EEG1(1024),EEG2(1024),FNAME
C  PEN COLOURS : 1 - BLACK, 2 - RED, 5 - GREEN, 7 - BLUE
      DATA   IPEN /1,2,5,7/
      DATA   WIDTH,ITYPE  /0.0,0/
C
C  TEMPORARILY SWITCH OFF WINDOW.
C
```

```
      CALL WINDOW(0)
      CALL MOVTO2(110.0,60.0)
      CALL CHAA1(FNAME,8)
      CALL MOVTO2(110.0,52.0)
      CALL CHAHOL(8HBATCH *.)
      CALL CHAINT(BATNO,3)
      CALL MOVTO2(110.0,44.0)
      DO 100 I=1,PARMNO
         ICOL = IPEN(THEPOS(I))
         CALL PENSEL(ICOL,WIDTH,IYTPE)
         CALL CHAHOL(8HTHETA *.)
         CALL CHAINT((THEPOS(I)+OFFSET-1),1)
         CALL CHAHOL(3H *.)
         CALL CHAFIX(ENDVAL(THEPOS(I)),9,4)
         CALL MOVTO2(110.0,(44.0-6.0*FLOAT(I)))
  100    CONTINUE
      CALL MOVTO2(110.0,(44.0-6.0*FLOAT(PARMNO+1)))
C
C  RESTORE WINDOW.
C
      CALL WINDOW(1)
      ICOL = 1
      CALL PENSEL(ICOL,WIDTH,ITYPE)
      RETURN
      END



C  FILE I.D. : RCOARM  OAR-F66          LAST REV : 9 FEB 87
C
C       THIS PROGRAM IS AN ADAPTATION OF PROGRAM UNIT ONLCRTN.
C  IT HAS BEEN SO ADAPTED IN ORDER TO ALLOW ITS INTEGRATION WITH OTHER
C  PROGRAM UNITS.  THIS IS TO ALLOW IT'S USE IN PRODUCING AVERAGED
C  RESULTS AND INDIVIDUAL TRIAL RESULTS FOR MULTIPLE TRIALS.
C
      SUBROUTINE RCOARM(TNAME,L1,TEMP,IMODEL,MPLFLG)
C
C THIS PROGRAM, TOGETHER WITH SEVERAL OTHER ROUTINES,
C IS USED TO REMOVE OCULAR ARTEFACTS FROM THE EEG
C USING ONE OF THREE USER-DEFINED ON-LINE ALGORITHMS
C (THESE ARE RLS,SQRT AND UD FILTERS).
C
      INTEGER FNAME(20),TNAME(20)
      LOGICAL ONLFLG,MPLFLG
      REAL TEMP(1024)
      COMMON VL(1024),VR(1024),HL(1024),HR(1024)
      COMMON /BLNKEX/  CMPNT1(1024),CMPNT2(1024),CMPNT3(1024)
      COMMON /VDATA/EEG1(1024),EEG2(1024),FNAME
      COMMON /XVAL/X(1024)
      COMMON /INTLV/ALPHA,GAMMA,THETA0
      DATA L2 /0/
      DO 10 I=1,20
         FNAME(I) = TNAME(I)
  10     CONTINUE
      IF (IMODEL .EQ. 10) CALL MODEL(FNAME,CMPNT1,CMPNT2,CMPNT3,IMODEL)
      NPT = 1024
      N = 1024
C  THE FOLLOWING DO LOOP IS PRESENET TO ALLOW FOR DIFFERENCES IN
C  IMPLEMENTATIONS OF M.J.N. AND E.I.C. PROGRAMS.
      DO 30 I=1,N
         CMPNT1(I) = CMPNT1(I)*1.E6
         CMPNT2(I) = CMPNT2(I)*1.E6
         CMPNT3(I) = CMPNT3(I)*1.E6
   30    CONTINUE
      DO 40 I=1,N
         IM1 = I - 1
         X(I) = FLOAT(IM1)/125.0
  40     CONTINUE
C
C SPECIFY GAMMA AND THE INITAL VALUES FOR
C P (OR S OR U).
C
C ENTER ALPHA - INITAL VALUE FOR PS
C       GAMMA - THE FORGETTING FACTOR
C   AND THETA(0) - THE INITAL VALUE FOR THETA
```

```
      ALPHA = 0.05
      GAMMA = 0.99609375
      THETA0 = 0.1
C
C SPECIFY AND THEN READ THE EEG RECORD TO PROCES
C
      L4 = 0
      ONLFLG = .FALSE.
      CALL RDATA(L1,L2,L3,L4,ONLFLG)
      IF (L3 .EQ. 0) GO TO 100
C
C REMOVE MEAN FROM THE DATA
C
      CALL MEAN(NPT,VL,VLM)
      CALL MEAN(NPT,VR,VRM)
      CALL MEAN(NPT,HL,HLM)
      CALL MEAN(NPT,HR,HRM)
      CALL MEAN(NPT,EEG1,EEGM)
      CALL MEAN(NPT,CMPNT1,C1AV)
      CALL MEAN(NPT,CMPNT2,C2AV)
      CALL MEAN(NPT,CMPNT3,C3AV)
C
C SELECT ON-LINE ALGORITM TO USE
C
C 0 FOR RLS, 1 FOR SORT, 2 FOR UDU
      ITYPE = 2
      CALL ADAPTV(NPT,N,L1,ITYPE,IMODEL,ONLFLG,MPLFLG)
  100 CONTINUE
C  RETURN CORRECTED DATA TO CALLING PROGRAM, ALLOWS FOR DIFFERENCES
C  BETWEEN M.J.N. AND E.I.C. PROGRAMS
      DO 500 I=1,NPT
          TEMP(I) = EEG1(I)*1.E-06
  500     CONTINUE
      RETURN
      END



C  FILE I.D. : RDATA  DAR-F66                      LAST REV : 10 MAR 87
C
      SUBROUTINE RDATA(L1,L2,L3,L4,ONLFLG)
C
C THIS SUBROUTINE READS DATA FORN A GIVEN BINARY FILE
C AND CONVERTS IT TO REAL FORMAT
C
      REAL MEEG1,MEEG2
      INTEGER FNAME(20)
      INTEGER BATNO,INP(1024)
      LOGICAL CSE,ONLFLG,DFILOP
      COMMON VL(1024),VR(1024),HL(1024),HR(1024)
      COMMON /BLNKEX/ CMPNT1(1024),CMPNT2(1024),CMPNT3(1024)
      COMMON /VDATA/ EEG1(1024),EEG2(1024),FNAME
      COMMON /SMPFRQ/ SAMRAT
      COMMON /DTFLST/ DFILOP
      DATA N,CSE /1024,.FALSE./
      BATNO = L1
      IF (L4 .GT. 0) GO TO 25
      IF (ONLFLG) CALL GETNAM(FNAME)
      WRITE(6,8)FNAME
    8 FORMAT(' DATA FILE ',20A1,///)
C
C CHECK WHETHER WHOLE FILE IS TO BE READ OR NOT
C
      IF (L2 .EQ. 0) GO TO 25
   10 WRITE(6,15)
   15 FORMAT(' ENTER BATCH NUMBER OR (-1) TO QUIT')
      READ(6,*)BATNO
      IF ((BATNO .LT. 0) .OR. (BATNO .GT. 191))GO TO 299
   25 WRITE(6,20)BATNO
   20 FORMAT(' BATNO=',I5)
      L = BATNO
C
C READ DATA AND CONVERT TO REAL FORMAT
C
```

```
      CALL DATIN(L,INP,FNAME,SF1,SF2,SAMRAT,DFILOP)
      DO 30 I=1,N
         VL(I) = FLOAT(INP(I))*SF1
 30      CONTINUE
      L = L + 1
      CALL DATIN(L,INP,FNAME,SF1,SF2,SAMRAT,DFILOP)
      DO 40 I=1,N
         VR(I) = FLOAT(INP(I))*SF1
 40      CONTINUE
      L = L + 1
      CALL DATIN(L,INP,FNAME,SF1,SF2,SAMRAT,DFILOP)
      DO 50 I=1,N
         HL(I) = FLOAT(INP(I))*SF1
 50      CONTINUE
      L = L + 1
      CALL DATIN(L,INP,FNAME,SF1,SF2,SAMRAT,DFILOP)
      DO 60 I=1,N
         HR(I) = FLOAT(INP(I))*SF1
 60      CONTINUE
      L = L + 1
      CALL DATIN(L,INP,FNAME,SF1,SF2,SAMRAT,DFILOP)
      DO 70 I=1,N
         EEG1(I) = FLOAT(INP(I))*SF2
 70      CONTINUE
      L = L + 1
      CALL DATIN(L,INP,FNAME,SF1,SF2,SAMRAT,DFILOP)
      DO 80 I=1,N
         EEG2(I) = FLOAT(INP(I))*SF2
 80      CONTINUE
      WRITE(6,90)SF1,SF2
 90   FORMAT(' SF1= ',F8.6,'  SF2= ',F8.6)
      L3 = 1
      IF ((L2 .EQ. 0) .AND. (L4 .LT. 31))RETURN
 240  CSE = .TRUE.
      CALL DATIN(L,INP,FNAME,SF1,SF2,SAMRAT,DFILOP)
      RETURN
 299  L3 = 0
      IF (BATNO .EQ. -1) GO TO 240
      WRITE(6,300)
 300  FORMAT(' ILLEGAL BATCH NO. - NO. SHOULD BE BETWEEN 0 AND 191')
      GO TO 10
      END


C  FILE I.D. : REDUCT COMM-F66                    CREATED  : 20 MAR 87
C                                                 LAST REV : 25 MAR 87
C
C  DETERMINES IF ANY REDUCTION IS TO BE APPLIED TO THE PLOT DEFAULT
C  UNITS SIZE OF 0.18 FOR TEKTRONIX AND 0.25 FOR CALCOMP. THESE ARE
C  VARIABLES 'TEXUNI' AND 'CALUNI' RESPECTIVELY. 'NSIZE' INDICATES
C  WHETHER FULL SIZE (0) OR REDUCED (1) PLOTS ARE TO BE PRODUCED.
C
      SUBROUTINE REDUCT(TEXUNI,CALUNI,NSIZE)
      INTEGER     IRDUCE,ICONT,KYBDIN,VDUOUT,NSIZE
      REAL        TEXUNI,CALUNI
      DATA  VDUOUT,KYBDIN  /6,5/
      NSIZE = 0
C  CHECK FOR VALID GRAPHICAL OUTPUT UNIT SIZE
      IF ((TEXUNI .NE. 0.18) .OR. (CALUNI .NE. 0.25)) GO TO 9000
C  DETERMINE REQUIRED REDUCTION
 100  WRITE(VDUOUT,120)
 120  FORMAT(//,' ENTER REQUIRED REDUCTION (IF ANY) - ',
     1       /,10X,'''1'' FOR NONE',/,10X,'''2'' FOR 2/3 RD. SIZE',
     2     /,10X,'''3'' FOR 58 PER CENT',/,10X,'''4'' FOR HALF SIZE',
     3     /,10X,'''5'' QUARTER SIZE')
      READ(KYBDIN,140,ERR=100)IRDUCE
 140  FORMAT(I1)
      IF ((IRDUCE .NE. 1) .AND. (IRDUCE .NE. 2) .AND. (IRDUCE .NE. 4)
     1   .AND. (IRDUCE .NE. 3) .AND. (IRDUCE .NE. 5)) GO TO 100
      IF (IRDUCE .EQ. 2) TEXUNI = (2./3.)*TEXUNI
      IF (IRDUCE .EQ. 2) CALUNI = (2./3.)*CALUNI
      IF (IRDUCE .EQ. 3) TEXUNI = 0.58*TEXUNI
      IF (IRDUCE .EQ. 3) CALUNI = 0.58*CALUNI
      IF (IRDUCE .EQ. 4) TEXUNI = 0.5*TEXUNI
```

```
      IF (IRDUCE .EQ. 4) CALUNI = 0.5*CALUNI
      IF (IRDUCE .EQ. 5) TEXUNI = 0.25*TEXUNI
      IF (IRDUCE .EQ. 5) CALUNI = 0.25*CALUNI
      IF (IRDUCE .NE. 1) NSIZE = 1
      WRITE(VDUOUT,200)TEXUNI,CALUNI,NSIZE
 200  FORMAT('  TEXUNI = ',F8.4,5X,'CALUNI = ',F8.4,5X,'NSIZE = ',I1)
  500 RETURN
C
C  ERROR EXCEPTION.
C
 9000 WRITE(VDUOUT,9020)
 9020 FORMAT(///,'***  ERROR IN SUBROUTINE REDUCT ***')
 9040 WRITE(VDUOUT,9060)
 9060 FORMAT(5X,'UNITS SIZE ENTERED FROM CALLING PROGRAM UNIT IN ERROR')
      WRITE(VDUOUT,9080)TEXUNI,CALUNI
 9080 FORMAT(5X,'TEKTRONIX UNITS = ',F8.4,5X,'CALCOMP UNITS = ',F8.4)
 9100 WRITE(VDUOUT,9120)
 9120 FORMAT(//,'*** ENTER ''1'' TO CONTINUE, ''0'' TO ABORT ***')
      READ(KYBDIN,9140,ERR=9100)ICONT
 9140 FORMAT(I1)
      IF ((ICONT .NE. 1) .AND. (ICONT .NE. 0)) GO TO 9100
      IF (ICONT .EQ. 1) GO TO 500
      STOP
      END


C  FILE I.D. : STATIO COMM-F66                  CREATED  : 24 FEB 87
C                                               LAST REV : 26 FEB 87
C
C  PROGRAM TO -
C     1)  CREATE FILES TO HOLD STATISTICAL INFORMATION FOR SUBSEQUENT
C         PROCESSING IN PREDICTIVE STATISTICAL ANALYSIS.
C     2)  READ BACK DATA FROM FILES CREATED IN STEP (1).
C
C  THE FOLLWING ARE THE ARGUMENTS TRANSMITTED TO/FROM THIS SUBROUTINE
C  AND HAVE THE DENOTED MEANING AND RANGE OF VALUES.
C
C  PREPOS - EPOCH 0 = PRE-STIMULUS   ,  TDLEN - FFT LENGTH
C                 1 = POST-STIMULUS  ,
C  NP - NUMBER OF DATA POINTS        ,  ECORR - OAR 0 = NO
C                                    ,              1 = YES
C  BLINE - BASELINE     0 = NO       ,  IFILT - LOW PASS 0 = NO
C          CORRECTION   1 = YES      ,          FILTER  1 = YES
C  PRES - START OF PRE-STIMULUS      ,  POSS - START OF POST-STIMULUS
C         EPOCH                      ,         EPOCH
C  ALPHA - ALPHA VALUE FOR KB WINDOW ,  EYETYP - OAR TYPE 0 = NON-REC.
C                                    ,                    1 = REC.
C  WINTYP - WINDOW 0 = MJN TUKEY     ,  NBAT - NUMBER OF TRIALS
C                  1 = KAISER-BESSEL ,
C                  2 = TUKEY         ,
C  FNAME - FILE TO BE PROCESSED      ,  IEPOCH - POST-STIM EPOCH
C                                    ,              1 = EARLY
C                                    ,              2 = LATE
C  IMODEL - MODEL TYPE 0 = NONE      ,  NUMHRM - NUMBER OF HARMONICS
C                      1 = STRAIGHT  ,
C                              LINE  ,
C  HRMTAB - HARMONIC NUMBERS TO BE   ,  BATNRS - BATCH NUMBERS TO BE
C           INCLUDED                 ,          INCLUDED
C  FTYPE - FILE-TYPE OF FILE TO BE   ,  FMODE - FILE-MODE OF FILE TO BR
C          READ, IF ANY              ,         READ, IF ANY
C  IOFLAG - READ OR WRITE STATS FILE 0 = INPUT (READ)
C                                    1 = WRITE (OUTPUT)
C  OPNFIL - INDICATES IF FILE ALREADY OPEN
C               .FALSE. = NOT OPEN
C               .TRUE.  = OPEN
C  NFST - NEAREST AND FURTHEST MEAN AMPLITUDE TEST (STATISTICS)
C  NFDF -                 ''                        (D.F.)
C  PPST - PRE- AND POST-STIMULUS MEAN AMPLTUDE TEST (STATISTICS)
C  PPDF -                 ''                        (D.F.)
C  CVST - RAYLEIGH TEST OF CIRCULAR VARIANCE
C  MDST - MODIFIED RAYLEIGH TEST OF CIRCULAR VARIANCE
C
      SUBROUTINE STATIO(PREPOS,NP,TDLEN,PRES,POSS,ECORR,BLINE,IFILT,
     1   WINTYP,ALPHA,NUMHRM,HRMTAB,NBAT,BATNRS,FNAME,EYETYP,IMEAN,
```

```
     2   NFST,NFDF,PPST,PPDF,CVST,MDST,IMODEL,IOFLAG,FTYPE,FMODE,OPNFIL)
      INTEGER  IHAR,PPDF,PREPOS,NP,TDLEN,PRES,POSS,BLINE,IFILT
      INTEGER  WINTYP,NUMHRM,MBAT,HRMTAB(32),BATNRS(32),EYETYP
      INTEGER  DATSET,FNAME(20),IMEAN,IMODEL,IOFLAG,IEPOCH,ISIVAL
      LOGICAL CNVFLG,ERAFLG,OPNFIL
      REAL    NFST(32),NFDF(32),PPST(32),CVST(32),MDST(32)
      REAL    ECORR,ALPHA
      INTEGER*2 FMODE,MODE
      INTEGER   RETCD,WRNAME(20),IBLANK,IDASH,CHEP1,CHEP2
      REAL      DEVICE
      REAL*8    NAME,FTYPE,TYPE
      COMMON /FLAGS/ CNVFLG,ERAFLG,ISIVAL
      COMMON /EPOCH/ IEPOCH
      DATA   DATSET,IDASH,IBLANK,CHEP1,CHEP2  /21,'-',' ','1','2'/
      DATA TYPE,MODE,DEVICE     /'STATFILE','C','DISK'/
C
C  CREATE WRITE FILE NAME.
C
      DO 30 I=1,20
         WRNAME(I) = IBLANK
30       CONTINUE
      WRNAME(1) = FNAME(1)
      WRNAME(2) = FNAME(4)
      WRNAME(3) = FNAME(5)
      WRNAME(4) = FNAME(6)
      WRNAME(5) = FNAME(7)
      IF (ISIVAL .EQ. 4) WRNAME(6) = IDASH
      IF ((ISIVAL .EQ. 4) .AND. (IEPOCH .EQ. 1)) WRNAME(7) = CHEP1
      IF ((ISIVAL .EQ. 4) .AND. (IEPOCH .EQ. 2)) WRNAME(7) = CHEP2
C
C  IF FILE ALREADY OPEN WRITE/READ DATA FROM NEXT SECTION OF FILE
C
      IF (OPNFIL) GO TO 70
      IF (IOFLAG .EQ. 1) CALL NAMCON(NAME,WRNAME)
      IF (IOFLAG .EQ. 0) CALL NAMCON(NAME,FNAME)
      IF (IOFLAG .EQ. 1) GO TO 50
      TYPE = FTYPE
      MODE = FMODE
50    CALL FILEDF(RETCD,DATSET,DEVICE,NAME,TYPE,MODE)
      IF (RETCD .NE. 0) GO TO 9000
      OPNFIL = .TRUE.
C
C  DETERMINE WHETHER READ OR WRITE FILE HEADER INFORMATION
C
70    IF ((IOFLAG .NE. 0) .AND. (IOFLAG .NE. 1)) GO TO 9000
      IF (IOFLAG .EQ. 0) GO TO 500
C
C  WRITE HEADER INFORMATION
C
      WRITE(DATSET,90)
90    FORMAT(1X,70('*'))
      WRITE(DATSET,100)(FNAME(I),I=1,8),NP,TDLEN,PREPOS,IEPOCH
100   FORMAT(' FILE  ',8A1,5X,'NP = ',I4,5X,'FFT LENGTH = ',I4,5X,
     1      'PREPOS = ',I1,4X,'IEPOCH = ',I1)
      WRITE(DATSET,120)PRES,POSS,ECORR,EYETYP,IMODEL
120   FORMAT(' PRES = ',I4,5X,'POSS = ',I4,5X,'ECORR = ',F3.1,5X,
     1      'EYETYP = ',I1,5X,'IMODEL = ',I2)
      WRITE(DATSET,140)BLINE,IFILT,IMEAN,WINTYP,ALPHA
140   FORMAT(' BLINE = ',I1,5X,'IFILT = ',I1,5X,'IMEAN = ',I1,5X,
     1      'WINTYP = ',I1,2X,'ALPHA = ',F5.2)
      WRITE(DATSET,160)NUMHRM,MBAT
160   FORMAT(' NUMHRM = ',I3,5X,'MBAT = ',I2)
      WRITE(DATSET,90)
C
C  WRITE STATISTICS TO FILE
C
      DO 300 I=1,NUMHRM
         WRITE(DATSET,260)HRMTAB(I),NFST(I),NFDF(I),PPST(I),PPDF,CVST(I)
     1              ,MDST(I)
260      FORMAT(1X,I4,3X,F8.5,1X,F5.2,3X,F8.5,1X,I2,2(3X,F7.5))
300      CONTINUE
      GO TO 900
C
C  READ HEADER INFORMATION
C
```

```
500   READ(DATSET,520)(FNAME(I),I=1,8),NP,TDLEN,PREPOS,IEPOCH
520   FORMAT(//,7X,8A1,5X,5X,I4,5X,13X,I4,5X,9X,I1,4X,9X,I1)
      READ(DATSET,540)PRES,POSS,ECORR,EYETYP,IMODEL
540   FORMAT(8X,I4,5X,7X,I4,5X,8X,F3.1,5X,9X,I1,5X,9X,I2)
      READ(DATSET,560)BLINE,IFILT,IMEAN,WINTYP,ALPHA
560   FORMAT(9X,I1,5X,8X,I1,5X,8X,I1,5X,9X,I1,2X,8X,F5.2)
      READ(DATSET,580)NUMHRM,MBAT
580   FORMAT(10X,I3,5X,7X,I2,//)
C
C  READ STATISTICS FROM FILE
C
      DO 700 I=1,NUMHRM
         READ(DATSET,660)HRMTAB(I),NFST(I),NFDF(I),PPST(I),PPDF,CVST(I)
     1             ,MDST(I)
660      FORMAT(1X,I4,3X,F8.5,1X,F5.2,3X,F8.5,1X,I2,2(3X,F7.5))
700      CONTINUE
900   RETURN
C
C  ERROR HANDLERS
C
9000  CONTINUE
      STOP
      END



C  FILE I.D. : STPRNT  COMM-F66                     LAST REV : JUL 86
C
      SUBROUTINE STPRNT(PREPOS,NP,TDLEN,PRES,POSS,ECORR,BLINE,IFILT,
     1   WINTYP,ALPHA,NUMHRM,HRMTAB,MBAT,BATNRS,FNAME,EYETYP,IMEAN,
     2   NFST,NFDF,PPST,PPDF,CVST,MDST,IMODEL)
      INTEGER  IHAR,PPDF,PREPOS,NP,TDLEN,PRES,POSS,BLINE,IFILT
      INTEGER  WINTYP,NUMHRM,MBAT,HRMTAB(NUMHRM),BATNRS(MBAT),EYETYP
      INTEGER  DATSET,FNAME(20),IMEAN,IMODEL
      REAL    NFST(32),NFDF(32),PPST(32),CVST(32),MDST(32)
      REAL    ECORR,ALPHA
      IF (PREPOS .EQ. 0) DATSET = 19
      IF (PREPOS .EQ. 1) DATSET = 20
      IF ((PREPOS .EQ. 0) .AND. (MBAT .GT. 1))
     1    WRITE(DATSET,100)(FNAME(I),I=1,8),MBAT
      IF ((PREPOS .EQ. 0) .AND. (MBAT .EQ. 1))
     1    WRITE(DATSET,120)(FNAME(I),I=1,8),MBAT
      IF ((PREPOS .EQ. 1) .AND. (MBAT .GT. 1))
     1    WRITE(DATSET,140)(FNAME(I),I=1,8),MBAT
      IF ((PREPOS .EQ. 1) .AND. (MBAT .EQ. 1))
     1    WRITE(DATSET,160)(FNAME(I),I=1,8),MBAT
  100 FORMAT('1',29X,' PRE-STIMULUS RESULTS FROM DATA FILE  -  ',
     1   8A1,4X,'COMPRISING ',I2,' BATCHES',10X,'PAGE ',I3)
  120 FORMAT('1',29X,' PRE-STIMULUS RESULTS FROM DATA FILE  -  ',
     1   8A1,4X,'COMPRISING ',I1,' BATCH   ',10X,'PAGE ',I3)
  140 FORMAT('1',29X,'POST-STIMULUS RESULTS FROM DATA FILE  -  ',
     1   8A1,4X,'COMPRISING ',I2,' BATCHES',10X,'PAGE ',I3)
  160 FORMAT('1',29X,'POST-STIMULUS RESULTS FROM DATA FILE  -  ',
     1   8A1,4X,'COMPRISING ',I1,' BATCH   ',10X,'PAGE ',I3)
      WRITE(DATSET,200)NP,TDLEN,PRES,POSS
  200 FORMAT(/,1X,I4,' DATA POINTS',6X,I4,' POINT FFT',6X,
     1    'PRE-STIMULUS EPOCH EXTENDS FROM POINT ',I4,6X,
     2    'POST-STIMULUS EPOCH EXTEND FROM POINT ',I4)
      WRITE(DATSET,300)
  300 FORMAT(/)
      IF ((ECORR .EQ. 1.) .AND. (EYETYP .EQ. 0) .AND. (IMODEL .EQ. 0))
     1    WRITE(DATSET,320)
      IF ((ECORR .EQ. 1.) .AND. (EYETYP .EQ. 0) .AND. (IMODEL .EQ. 10))
     1    WRITE(DATSET,330)
      IF ((ECORR .EQ. 1.) .AND. (EYETYP .EQ. 1) .AND. (IMODEL .EQ. 0))
     1    WRITE(DATSET,340)
      IF ((ECORR .EQ. 1.) .AND. (EYETYP .EQ. 1) .AND. (IMODEL .EQ. 10))
     1    WRITE(DATSET,350)
      IF (BLINE .EQ. 1) WRITE(DATSET,360)
      IF (IFILT .EQ. 1) WRITE(DATSET,380)
  320 FORMAT(20X,'NON-RECURSIVE OA REMOVAL APPLIED (NO MODELLING)')
  330 FORMAT(20X,'NON-RECURSIVE OA REMOVAL APPLIED (CNV/ERA MODELLED)')
  340 FORMAT(20X,'RECURSIVE OA REMOVAL APPLIED (NO MODELLING)')
  350 FORMAT(20X,'RECURSIVE OA REMOVAL APPLIED (CNV/ERA MODELLED)')
  360 FORMAT(20X,'BASELINE CORRECTION APPLIED')
```

```
  380 FORMAT(20X,'FILTERING OF DATA PERFORMED')
      IF (WINTYP .EQ. 0) WRITE(DATSET,500)
      IF (WINTYP .EQ. 1) WRITE(DATSET,520)ALPHA
      IF (WINTYP .EQ. 2) WRITE(DATSET,530)
  500 FORMAT(20X,'25 PER CENT TUKEY WINDOW')
  520 FORMAT(20X,'KAISER-BESSEL WINDOW WITH ALPHA = ',F8.4)
  530 FORMAT(20X,'MJN 25 PER CENT ''TAPER2'' TUKEY WINDOW')
      IF (IMEAN .EQ. 0) WRITE(DATSET,540)
      IF (IMEAN .EQ. 1) WRITE(DATSET,550)
  540 FORMAT(20X,'MEAN OF WINDOWED DATA INCLUDED')
  550 FORMAT(20X,'MEAN OF WINDOWED DATA REMOVED')
      WRITE(DATSET,580)NUMHRM
  580 FORMAT(/,10X,I4,1X,'HARMONICS HAVE BEEN SELECTED FOR THE',
     1   ' STATTISTICAL TESTS')
      WRITE(DATSET,700)
  700 FORMAT(' HARMONIC.',5X,'NEAREST AND FURTHEST',5X,
     1   '. PRE- AND POST- STIMULUS MEAN .',3X,
     2   'RAYLEIGH TEST OF CIRCULAR  .',3X,
     3   'MODIFIED RAYLEIGH TEST OF')
      WRITE(DATSET,720)
  720 FORMAT(' ========.',5X,'MEAN AMPLITUDE TEST',6X,
     1   '.  AMPLITUDE DIFFERENCES TEST  .',11X,
     2   'VARIANCE',11X,'.',7X,'CIRCULAR VARIANCE')
      WRITE(DATSET,740)
  740 FORMAT(9X,'.',5X,'====================',5X,
     1   '. ============================ .',3X,
     2   '========================  .',3X,
     3   '========================')
      WRITE(DATSET,760)
  760 FORMAT(9X,'.(1 TAIL)',22X,'.(2 TAIL)',22X,
     1   '.CIRCULAR',22X,'.MOD. DISP.')
      WRITE(DATSET,780)PPDF
  780 FORMAT(9X,'. T-STAT  D.F.',17X,'. T-STAT  ALL WITH ',I2,
     1   ' D.F.',5X,'.VARIANCE',22X,'.  FACTOR')
      WRITE(DATSET,800)
C 800 FORMAT(9X,'.------- ----',17X,'.------- ---------------',
C    1   5X,'.--------',22X,'.---------')
  800 FORMAT(9X,4('.',30('-')))
      DO 1000 I=1,NUMHRM
         IF (PREPOS .EQ. 0) WRITE(DATSET,900)
     1      (HRMTAB(I),NFST(I),NFDF(I),CVST(I),MDST(I))
  900    FORMAT(4X,I3,2X,'.',F8.5,' ',F4.1,17X,'. ',7X,
     1               22X,'. ',F7.5,22X,'. ',F7.5)
C        IF (PREPOS .EQ. 1) WRITE(DATSET,950)
C    1      (HRMTAB(I),NFST(I),NFDF(I),PPST(I),CVST(I),MDST(I))
C 950    FORMAT(4X,I3,2X,'.',F8.5,' ',F4.1,17X,'. ',F7.4,
C    1               22X,'. ',F7.5,22X,'. ',F7.5)
         IF (PREPOS .EQ. 1) WRITE(DATSET,950)
     1      (HRMTAB(I),NFST(I),NFDF(I),PPST(I),CVST(I),MDST(I))
  950    FORMAT(4X,I3,2X,'.',F8.5,' ',F4.1,17X,'. ',F7.4,
     1               22X,'. ',F7.5,22X,'. ',F7.5)
         WRITE(DATSET,970)
  970    FORMAT(9X,4('.',30('-')))
 1000    CONTINUE
      RETURN
      END


C  FILE I.D. : SUFACT TPFL-F66               LAST REV : 16 FEB 87
C
C  COMPUTES FACTORIAL 1 TO FACTORIAL 32 , STORING THE RESULT IN A
C  COMMON BLOCK TO SAVE REAPEATED CALCULATIONS.
C
      SUBROUTINE SUFACT
      COMMON /ARFACT/ FACT(32)
      FACT(1) = 1.0
      DO 200 I=2,32
         REFACT = 1.0
         DO 100 J=1,I
            REFACT = REFACT * FLOAT(J)
100         CONTINUE
         FACT(I) = REFACT
200      CONTINUE
      RETURN
      END
```

```
C  FILE I.D. : TAPCOS TPFL-F66                      LAST REV : 13 MAR 87
C
C       THIS PROGRAM IS AN INTERFACE BETWEEN THE MAIN PORGRAM UNITS RE-
C  SPONSIBLE FOR SIGANL PROCESSING AND THE PROGRAM UNITS WHICH GENERATE
C  THE REQUIRED WINDOWS.
C  ALSO THE WINDOWED DATA MAY, OR MAY NOT (DEPENDING ON A SUBROUTINE
C  ARGUMENT), HAVE ANY MEAN REMOVED.
C
      SUBROUTINE TAPCOS(TRDAT,NP,IMEAN)
      INTEGER  NP,VDUOUT,IMEAN,IDCTYP
      REAL     TRDAT(NP),WINVAL(1024),RMEAN,SUM1,SUM2,SUM3,WINDAT(1024)
      LOGICAL  WFILOP
      COMMON /WFILST/ WFILOP,SUM1,SUM3,WINVAL
      DATA     VDUOUT /6/
C  SET IDCTYP = 0 FOR 'SIMPLE' MEAN LEVEL REMOVAL.
C  SET IDCTYP = 1 FOR 'COMPLEX' MEAN LEVEL REMOVAL.
      DATA IDCTYP /1/
      IF ((IMEAN .NE. 0) .AND. (IMEAN .NE. 1)) GO TO 1300
      IF (WFILOP) GO TO 200
      DO 100 I=1,NP
         WINVAL(I) = 1.0
100      CONTINUE
      CALL COSGEN(WINVAL,NP)
C  TAPER DATA
200   DO 400 I=1,NP
         TRDAT(I) = TRDAT(I)*WINVAL(I)
400      CONTINUE
C
C  DETERMINE WHETHER OR NOT TO SUBTRACT MEAN
C
  600 IF (IMEAN .NE. 1) GO TO 900
C
C  SUBTRACT MEAN FROM WINDOWED DATA
C
      IF (IDCTYP .EQ. 1) GO TO 700
      SUM1 = 0.0
      DO 650 I=1,NP
         SUM1 = SUM1 + TRDAT(I)
  650    CONTINUE
      RMEAN = SUM1/FLOAT(NP)
      DO 680 I=1,NP
         TRDAT(I) = TRDAT(I) - RMEAN
  680    CONTINUE
      GO TO 900
  700 SUM2 = 0.0
      IF (WFILOP) GO TO 740
      SUM1 = 0.0
      SUM3 = 0.0
      DO 730 I=1,NP
         SUM1 = SUM1 + WINVAL(I)
         SUM3 = SUM3 + WINVAL(I)*WINVAL(I)
  730    CONTINUE
      SUM3 = SQRT(FLOAT(NP)/SUM3)
  740 DO 750 I=1,NP
         SUM2 = SUM2 + TRDAT(I)*WINVAL(I)
  750    CONTINUE
      SUM2 = SUM2/SUM1
      DO 800 I=1,NP
         TRDAT(I) = (TRDAT(I)-SUM2)*SUM3*WINVAL(I)
  800    CONTINUE
      WFILOP = .TRUE.
  900 RETURN
C
C  ERROR MESSAGES
C
 1300 WRITE(VDUOUT,1320)
 1320 FORMAT(' **** INVALID IMEAN VALUE ****')
 2000 WRITE(VDUOUT,2020)
 2020 FORMAT(' STOP IN SUBROUTINE TAPCOS')
      STOP
      END
```

```
C  FILE I.D. : TAPER2  TPFL-F66          LAST REV : 15 DEC 86
C
      SUBROUTINE TAPER2(X,N,IMEAN)
C
C     TAPERS SUBROUTINE AND SUBTRACTS MEAN
C
      INTEGER I,IMEAN,IDCTYP
      REAL SUM1,SUM2,SUM3,W,WINDY,X(N),XMEAN
C
C  SET IDCTYP = 0 FOR 'SIMPLE' MEAN LEVEL REMOVAL.
C  SET IDCTYP = 1 FOR 'COMPLEX' MEAN LEVEL REMOVAL.
      DATA  IDCTYP /1/
      IF ((IMEAN .EQ. 1) .AND. (IDCTYP .EQ. 1)) GO TO 200
C  TAPER DATA.
      DO 100 I=1,N
         X(I) = X(I)*WINDY(I,N)
  100    CONTINUE
      IF (IMEAN .EQ. 0) GO TO 300
      XMEAN = 0.0
      DO 120 I=1,N
         XMEAN = XMEAN + X(I)
  120    CONTINUE
      XMEAN = XMEAN/FLOAT(N)
      DO 140 I=1,N
         X(I) = X(I) - XMEAN
  140    CONTINUE
      GO TO 300
  200 SUM1 = 0.0
      SUM2 = 0.0
      SUM3 = 0.0
      DO 210 I=1,N
         W = WINDY(I,N)
         SUM1 = SUM1 + W
         SUM3 = SUM3 + W*W
         SUM2 = SUM2 + W*X(I)
  210    CONTINUE
      SUM2 = SUM2/SUM1
      SUM3 = SQRT(N/SUM3)
      DO 220 I=1,N
         X(I) = (X(I)-SUM2)*SUM3*WINDY(I,N)
  220    CONTINUE
  300 RETURN
      END


C  FILE I.D. : TAPKAI  TPFL-F66                   LAST REV : 13 MAR 87
C
C       THIS PROGRAM IS AN INTERFACE BETWEEN THE MAIN PORGRAM UNITS RE-
C  SPONSIBLE FOR SIGANL PROCESSING AND THE PROGRAM UNITS WHICH GENERATE
C  THE REQUIRED WINDOWS.
C  ALSO THE WINDOWED DATA MAY, OR MAY NOT (DEPENDING ON A SUBROUTINE
C  ARGUMENT), HAVE ANY MEAN REMOVED.
C
      SUBROUTINE TAPKAI(TRDAT,NP,ALPHA,IMEAN)
      INTEGER  NP,VDUOUT,IMEAN,IDCTYP
      REAL     TRDAT(NP),ALPHA,RMEAN
      REAL     WINVAL(1024),SUM1,SUM2,SUM3
      LOGICAL  WFILOP
      COMMON /WFILST/ WFILOP,SUM1,SUM3,WINVAL
C  SET IDCTYP = 0 FOR 'SIMPLE' MEAN LEVEL REMOVAL.
C  SET IDCTYP = 1 FOR 'COMPLEX' MEAN LEVEL REMOVAL.
      DATA IDCTYP /1/
      DATA     VDUOUT /6/
      IF ((IMEAN .NE. 0) .AND. (IMEAN .NE. 1)) GO TO 1300
      IF (WFILOP) GO TO 200
      DO 100 I=1,NP
         WINVAL(I) = 1.0
100      CONTINUE
      CALL KAIGEN(WINVAL,NP,ALPHA)
C   TAPER DATA
200   DO 400 I=1,NP
         TRDAT(I) = SNGL(DBLE(TRDAT(I))*DBLE(WINVAL(I)))
400      CONTINUE
C
```

```
C  DETERMINE WHETHER OR NOT TO REMOVE MEAN
C
  600 IF (IMEAN .NE. 1) GO TO 900
C
C  SUBTRACT MEAN FROM WINDOWED DATA
C
      IF (IDCTYP .EQ. 1) GO TO 700
      SUM1 = 0.0
      DO 650 I=1,NP
         SUM1 = SUM1 + TRDAT(I)
  650    CONTINUE
      RMEAN = SUM1/FLOAT(NP)
      DO 680 I=1,NP
         TRDAT(I) = TRDAT(I) - RMEAN
  680    CONTINUE
      GO TO 900
  700 SUM2 = 0.0
      IF (WFILOP) GO TO 740
      SUM1 = 0.0
      SUM3 = 0.0
      DO 730 I=1,NP
         SUM1 = SUM1 + WINVAL(I)
         SUM3 = SUM3 + WINVAL(I)*WINVAL(I)
  730    CONTINUE
      SUM3 = SQRT(FLOAT(NP)/SUM3)
  740 DO 750 I=1,NP
         SUM2 = SUM2 + TRDAT(I)*WINVAL(I)
  750    CONTINUE
      SUM2 = SUM2/SUM1
      DO 800 I=1,NP
         TRDAT(I) = (TRDAT(I)-SUM2)*SUM3*WINVAL(I)
  800    CONTINUE
      WFILOP = .TRUE.
  900 RETURN
C
C  ERROR MESSAGES
C
 1300 WRITE(VDUOUT,1320)
 1320 FORMAT(' **** INVALID IMEAN VALUE ****')
 2000 WRITE(VDUOUT,2020)
 2020 FORMAT(' STOP IN SUBROUTINE TAPKAI')
      STOP
      END
```

```
C  FILE I.D. : THENVL  DAR-F66            LAST REV : 13 OCT 86
C
C       THIS SUBROUTINE SENDS TO A DATA FILE THE VALUES OF THE THETAS
C  FOR THE FINAL DATA POINT IN EACH TRIAL.  IT SUPPLIES THE TRIAL NUM-
C  BER, AND THE VALUES OF THE THETAS IN THE ORDER THETA 1, THETA 2,
C  THETA 3 AND THETA 4 (PRINTED LEFT TO RIGHT).
C
      SUBROUTINE THENVL(THDEND,NPAR)
      INTEGER  BATNO,DATSET,NPAR
      REAL     THDEND(NPAR)
      COMMON   /TRLNUM/  BATNO
      DATA     DATSET /13/
      WRITE(DATSET,100)BATNO,(THDEND(I),I=1,NPAR)
  100 FORMAT(4X,I4,7(1X,F8.4,1X))
      RETURN
      END
```

```
C  FILE I.D. : UDUFLT  DAR-F66            LAST REV : 1 DEC 86
C
      SUBROUTINE UDUFLT(Y,X,U,THETA,NPAR,E,GAMMA,I1)
C
C THIS SUBROUTINE IS BASED ON BIERMAN'S CODING OF THE UDU
C FILTERING ALGORITHM. IT COMPUTES THE PARAMETER ESTIMATES,
C THE EMA ESTIMATES AND THE CORRECTED EEG USING RLS METHOD.
C INPUTS:
C       U       UPPER TRIANGULAR MATRIX WITH D(I) STORED IN U(I,I)
```

```
C       Y,X      OUTPUT SAMPLE AND EOG DATA VECTOR
C       THETA    VECTOR OF PREVIOUS PARAMETER ESTIMATES
C       GAMMA    THE FORGETTING FACTOR
C       NPAR     NUMBER OF PARAMETERS IN THE MODEL
C       I1       THE SAMPLE NUMBER
C
C OUTPUTS:
C       U         UPDATED TRIANGYLAR MATRIX
C       B         THE UNWEIGTHED KALMAN GAIN
C       THETA     UPDATED PARAMETER ESTIMATES
C       E         CORRECTED EEG DATA
C
      DIMENSION X(7),U(28),THETA(7),B(7),V(7)
      SF = 1.0/GAMMA
      PERR = Y
      DO 2 J=1,NPAR
         PERR = PERR - X(J)*THETA(J)
    2    CONTINUE
      M = 1
      V(1) = X(1)
      IF (NPAR .EQ. 1) GO TO 15
      DO 10 J=2,NPAR
         V(J) = X(J)
         J1 = J - 1
         DO 5 K=1,J1
            M = M + 1
            V(J) = V(J) + U(M)*X(K)
    5       CONTINUE
         M = M + 1
         B(J) = U(M)*V(J)
   10    CONTINUE
   15 B(1) = U(1)*X(1)
      ALPHA = GAMMA + B(1)*V(1)
      DELTA = 1.0/ALPHA
      U(1) = U(1)*DELTA
      M = 1
      IF (NPAR .EQ. 1) GO TO 27
      DO 25 J=2,NPAR
         BETA1 = ALPHA
         ALPHA = ALPHA + B(J)*V(J)
         P = -V(J)*DELTA
         DELTA = 1.0/ALPHA
         J1 = J - 1
         DO 20 K=1,J1
            M = M + 1
            BETA = U(M)
            U(M) = BETA + B(K)*P
            B(K) = B(K) + B(J)*BETA
   20       CONTINUE
        M = M + 1
        U(M) = U(M)*BETA1*DELTA*SF
   25   CONTINUE
   27 PERR = PERR/ALPHA
      DO 30 J=1,NPAR
         THETA(J) = THETA(J) + B(J)*PERR
   30    CONTINUE
      E = Y
      NPAR2 = NPAR
      IF (NPAR2 .GT. 4)NPAR2 = 4
      DO 35 I=1,NPAR2
         E = E - X(I)*THETA(I)
   35    CONTINUE
      RETURN
      END


C  FILE I.D. : VSTAT2  STAT-F66
C
      SUBROUTINE VSTAT2(ANGLE,RAD,N)
C
C     THIS SUBROUTINE EXAMINES THE N VECTORS WHOSE DIRECTIONS ARE STORED
C     IN ARRAY 'ANGLE' AND WHOSE MAGNITUDES ARE STORED IN ARRAY 'RAD'. THE
C     PROGRAM IDENTIFIES THE (N/2) VECTORS WHICH LIE IN THE SMALLEST ARC
C     AND COMPARES THE AVERAGE LENGTH OF THESE VECTORS WITH THE AVERAGE
```

```
C     LENGTH OF THE REMAINING VECTORS. THE COMPARISON IS MADE BY A T-TEST
C     WITH AN APPROXIMATE CORRECTION FOR (POSSIBLE) UNEQUAL VARIANCES.
C
C
      INTEGER  VDUOUT
      DIMENSION ANGLE(N),RAD(N),TEMP(100)
      DOUBLE PRECISION SUM11,SUM12,SUM21,SUM22
      REAL PRNFST(32),PRNFDF(32),PRPPST(32),PRCVST(32),PRMDST(32)
      REAL NFSTPO(32),NFDFPO(32),PPSTPO(32),CVSTPO(32),MDSTPO(32)
      INTEGER IHRPRE,IHRPOS,PRPPDF,PPDFPO
      LOGICAL PRE,POS
      COMMON /PREST/ PRNFST,PRNFDF,PRPPST,PRPPDF,PRCVST,PRMDST,IHRPRE,
     1               PRE
      COMMON /POSST/ NFSTPO,NFDFPO,PPSTPO,PPDFPO,CVSTPO,MDSTPO,IHRPOS,
     1               POS
      DATA PI/3.1415926536/
      DATA  VDUOUT /14/
C     DATA  VDUOUT /6/
C
C     STORES ANGLES IN A TEMPORARY ARRAY AND ARRANGES IN ASCENDING ORDER
C     USING A BUBBLE SORT.
C
      DO 1 I=1,N
          TEMP(I)=ANGLE(I)
    1     CONTINUE
      DO 2 I=2,N
          MAX=1
          AMAX=TEMP(1)
          NMAX=N-I+2
          DO 3 J=2,NMAX
              IF(TEMP(J).LE.AMAX) GO TO 3
              AMAX=TEMP(J)
              MAX=J
    3         CONTINUE
          TEMP(MAX)=TEMP(NMAX)
    2     TEMP(NMAX)=AMAX
C
C     FIND THE (N/2) VALUES LYING IN THE SMALLEST ARC.
C
      N2=N/2
      DMIN=TEMP(N2)-TEMP(1)
      AMIN=TEMP(1)
      AMAX=TEMP(N2)
      DO 4 I=2,N
          IU=I+N2-1
          IF(IU .LE. N)UPPER=TEMP(IU)
          IF(IU .GT. N)UPPER=TEMP(IU-N)+2.*PI
          IF((UPPER-TEMP(I)) .GE. DMIN)GO TO 4
          DMIN=UPPER-TEMP(I)
          AMIN=TEMP(I)
          AMAX=UPPER
    4     CONTINUE
C
C     CALCULATES MEAN AND VARIANCE FOR BOTH SETS OF DATA
C
      SUM11=0.0
      SUM12=0.0
      SUM21=0.0
      SUM22=0.0
      N2=0
      N21=0
      CONST=0.
      IF(AMAX.GT.PI) CONST=2*PI
      DO 5 I=1,N
          A=ANGLE(I)
          IF(A .LT. 0.)A=A+CONST
          IF(A .LT. AMIN .OR. A .GT. AMAX)GO TO 6
          N2=N2+1
          SUM11=SUM11+RAD(I)
          SUM12=SUM12+RAD(I)*RAD(I)
          GO TO 5
    6     SUM21=SUM21+RAD(I)
          N21=N21+1
          SUM22=SUM22+RAD(I)*RAD(I)
    5     CONTINUE
```

```
      SUM11=SUM11/N2
      SUM21=SUM21/N21
      SUM12=(SUM12/N2-SUM11*SUM11)/(N2-1)
      SUM22=(SUM22/N21-SUM21*SUM21)/(N21-1)
C
C     CALCULATES T-STATISTIC AND OUTPUTS RESULTS
C
      TSTAT=SNGL((SUM11-SUM21)/DSQRT(SUM12+SUM22))
      DF=(SUM12+SUM22)*(SUM12+SUM22)
      DF=DF/(SUM12*SUM12/(N2+1)+SUM22*SUM22/(N21+1))-2
      SUM12=DSQRT(SUM12*N2)
      SUM22=DSQRT(SUM22*N21)
      WRITE(VDUOUT,100)N2,SUM11,SUM12
      WRITE(VDUOUT,101)N21,SUM21,SUM22
      WRITE(VDUOUT,102)TSTAT,DF
 100  FORMAT(/' MEAN AND S.D. OF LENGTHS OF THE',I4,' CLOSEST VECTORS
     += ',E16.8,E16.8)
 101  FORMAT(' MEAN AND S.D. OF LENGTHS OF THE',I4,' REMAINING VECTORS =
     + ',E16.8,E16.8)
 102  FORMAT(/' T-STATISTIC =',F10.5,' WITH ',F6.1,
     +'  DEGREES OF FREEDOM')
      IF (PRE) PRNFST(IHRPRE) = TSTAT
      IF (PRE) PRNFDF(IHRPRE) = DF
      IF (POS) NFSTPO(IHRPOS) = TSTAT
      IF (POS) NFDFPO(IHRPOS) = DF
      RETURN
      END



C  FILE I.D. : VSTAT3  STAT-F66
C
      SUBROUTINE VSTAT3(ANGLE,RAD,N)
C
C     THIS SUBROUTINE CALCULATES A 'MODIFIED DISPERSION' STATISTIC FOR THE 'N'
C     VECTORS WHOSE DIRECTIONS ARE STORED IN ARRAY 'ANGLE' AND WHOSE
C     MAGNITUDES ARE STORED IN ARRAY 'RAD'.
C
C     MODIFIED DISPERSION FACTOR ....  UM=1-SQRT(S*S+C*C)
C
C     WHERE   S= WEIGHTED AVERAGE OF SINE VALUES
C             C= WEIGHTED AVERAGE OF COSINE VALUES
C
C     AND THE WEIGHTING FACTORS ARE THE RANK ORDERS OF THE VECTORS
C
C     UM HAS VALUE 1 FOR A ZERO MAGINTUDE RESULTANT VECTOR
C                  0 FOR A SET OF ALIGNED VECTORS
C
      INTEGER  VDUOUT
      DIMENSION ANGLE(N),RAD(N),TEMPA(100),TEMPR(100)
      REAL PRNFST(32),PRNFDF(32),PRPPST(32),PRCVST(32),PRMDST(32)
      REAL NFSTPO(32),NFDFPO(32),PPSTPO(32),CVSTPO(32),MDSTPO(32)
      INTEGER IHRPRE,IHRPOS,PRPPDF,PPDFPO
      LOGICAL PRE,POS
      COMMON /PREST/ PRNFST,PRNFDF,PRPPST,PRPPDF,PRCVST,PRMDST,IHRPRE,
     1               PRE
      COMMON /POSST/ NFSTPO,NFDFPO,PPSTPO,PPDFPO,CVSTPO,MDSTPO,IHRPOS,
     1               POS
      DATA  VDUOUT /14/
C     DATA  VDUOUT /6/
      DO 1 I=1,N
           TEMPA(I)=ANGLE(I)
           TEMPR(I)=RAD(I)
   1       CONTINUE
      C=0.0
      S=0.0
      DO 2 I=2,N
           MAX=1
           AMAX=TEMPR(1)
           NMAX=N-I+2
           DO 3 J=2,NMAX
                IF (TEMPR(J).LE.AMAX) GO TO 3
                AMAX=TEMPR(J)
                MAX=J
 3              CONTINUE
```

```
          C=C+NMAX*COS(TEMPA(MAX))
          S=S+NMAX*SIN(TEMPA(MAX))
          TEMPR(MAX)=TEMPR(NMAX)
          TEMPA(MAX)=TEMPA(NMAX)
 2        CONTINUE
      C=C+COS(TEMPA(1))
      S=S+SIN(TEMPA(1))
      SUMN=N*(N+1)/2
      C=C/SUMN
      S=S/SUMN
      UM=1.0-SQRT(S*S+C*C)
      WRITE(VDUOUT,100) UM
 100  FORMAT(/' MODIFIED DISPERSION FACTOR = ',F10.5)
      IF (PRE) PRMDST(IHRPRE) = UM
      IF (POS) MDSTPO(IHRPOS) = UM
      RETURN
      END


C  FILE I.D. : WAIT COMM-F66          CREATED  : 22 APR 87
C
      SUBROUTINE WAIT
10    CALL MOVTO2(0.,0.)
      CALL CURDEF(3H1*,)
      CALL CURSOR(I,X,Y)
      IF (I .NE. 1) GO TO 10
      RETURN
      END


C  FILE I.D. : WINDY  TPFL-F66
C
      REAL FUNCTION WINDY(J,N1)
      INTEGER J,N,NTL,N1
      DATA PI /3.1415926536/
      TL = .125
      AN = FLOAT(N1)
      AN1 = AN - 1
C
C       TL IS THE TAPER LENGTH
C
      NTL = IFIX(TL*AN+0.5)
      WINDY = 1.0
      AJ = J - 0.5
      IF (J .GE. NTL .AND. J .LE. (N1-NTL))RETURN
      WINDY = (1. - COS(PI*AJ/(AN1*TL)))/2.
      IF (J .GT. (N1-NTL))
     1    WINDY = (1. + COS(PI*(AJ+NTL-AN1)/(AN1*TL)))/2.
      RETURN
      END


C  FILE I.D. : XVAL1  DAR-F66
C
      SUBROUTINE XVAL1(XN,NPAR,I1)
C
C PRIME THE VECTOR X WITH EOG DATA
C
      DIMENSION XN(7)
      COMMON VL(1024),VR(1024),HL(1024),HR(1024)
      COMMON /BLNKEX/ CMPNT1(1024),CMPNT2(1024),CMPNT3(1024)
      XN(1) = VR(I1)
      XN(2) = HR(I1)
      XN(3) = HL(I1)
      XN(4) = VL(I1)
      XN(5) = CMPNT1(I1)
      XN(6) = CMPNT2(I1)
      XN(7) = CMPNT3(I1)
      RETURN
      END
```

```
C  FILE I.D. : XYVAL  OAR-F66
C
      SUBROUTINE XYVAL(PARM,DOUT,RSS,RSD,DA,NPAR,ID,ITYPE)
C
C XN      IS A VECTOR OF EOG DATA
C THETA   IS A VECTOR OF PARAMETER ESTIMATES (NPAR*1)
C PS      IS COVARIANCE MATRIX OF THE EOG DATA (NPAR*NPAR)
C
      DIMENSION PARM(7168),DOUT(1024),RSS(1024),RSD(1024),DA(1024)
      COMMON VL(1024),VR(1024),HL(1024),HR(1024)
      COMMON /BLNKEX/ CMPNT1(1024),CMPNT2(1024),CMPNT3(1024)
      N = 1024
      IF (ID .LT. 5) GO TO 110
      DO 150 I=1,N
          VL(I) = HR(I)*HL(I)
  150     CONTINUE
  110 CONTINUE
      CALL ONLSUB(PARM,DOUT,RSS,RSD,DA,NPAR,ITYPE)
      RETURN
      END
```

```
C    FILE I.D. : DISC FORTRAN                LAST REV : 15 DEC 87
C
C    15 DEC 87 : CHANGED READ STATEMENTS TO TAKE DATA STRAIGHT
C                FROM FILE RATHER TAHN FROM KEYBOARD
C
C    PROGRAM UNITS TO PERFORM PREDICTIVE STATISTICAL DIAGNOSIS
C
C    M     = NUMBER OF VARIABLES (TESTS)
C    N     = NUMBER OF OBSERVATIONS TO BE CLASSIFIED
C    PRIOR = PRIOR PROBABILITY OF BEING NORMAL
C    N1    = NUMBER OF NORMAL SUBJECTS
C    N2    = NUMBER OF HC SUBJECTS
C    AM1   = ARRAY HOLDING MEANS FOR NORMAL SUBJECTS
C    AM2   = ARRAY HOLDING MEANS FOR HC SUBJECTS
C    X     = ARRAY HOLDING DATA TO BE CLASSIFIED
C
C    THE FOLLOWING SUBROUTINES ARE USED
C    - STN    COMPUTES D-DIMENSIONAL STUDENT TYPE DENSITY AND
C             ATYPICALITY INDICES
C    - DISP   COMPUTES DISPERSION MATRIX
C    - MINV   IS FROM THE IBM SCIENTIFIC SYSTEMS
C             SUBROUTINE (SSP)
C
      DIMENSION X(10),C1(10,10),C2(10,10),AM1(10),AM2(10)
      READ(12,*) M,N,PRIOR
      READ(12,*) N1,(AM1(J),J=1,M)
      READ(12,*) N2,(AM2(J),J=1,M)
      V1=FLOAT(N1)-1.0
      V2=FLOAT(N2)-1.0
      CALL DISP(M,N1,C1,V1,DET1)
      CALL DISP(M,N2,C2,V2,DET2)
      DO 4 I=1,N
      READ(12,*) (X(J),J=1,M)
      CALL STN(M,V1,AM1,C1,X,S1,DET1,A1)
      CALL STN(M,V2,AM2,C2,X,S2,DET2,A2)
      PA=S1*PRIOR/(S1*PRIOR+(1.0-PRIOR)*S2)
    4 WRITE(6,100) I,PA,A1,A2
  100 FORMAT(I10,3F10.3)
      STOP
      END
C             SUBROUTINE STN(M,V,B,C,X,S,DET,A)
C             DIMENSION B(M),X(M),C(M,M)
C
      SUBROUTINE STN(M,V,B,C,X,S,DET,A)
      DIMENSION B(M),X(M),C(M,M)
      P=0.5*(V+1)
      Q=0.5*(V-FLOAT(M)+1)
      D=3.14159**(0.5*FLOAT(M))
      A=GAMMA(P)/(D*GAMMA(Q)*SQRT(DET))
      F=0.0
      DO 1 J=1,M
      DO 1 K=1,M
    1 F=F+(X(J)-B(J))*(X(K)-B(K))*C(J,K)
      S=1.0/(1.0+F)**P
      S=A*S
      P1=FLOAT(M)/2.0
      P2=P-P1
      BETA=ALGAMA(P1)+ALGAMA(P2)-ALGAMA(P)
      Z=F/(F+1)
      A=BETAIN(Z,P1,P2,BETA,IFAULT)
      IF(IFAULT.NE.0) A=-1
      RETURN
      END
C
C
C
```

```
      SUBROUTINE DISP(M,N,C,V,DET)
      DIMENSION C(M,M),LW1(10),LW2(10)
      DO 2 I=1,M
    2 READ(12,*) (C(I,J),J=1,M)
      CR=(1.0+1.0/FLOAT(N))*V
      DO 1 I=1,M
      DO 1 J=1,M
    1 C(I,J)=C(I,J)*CR
      CALL MINV(C,M,DET,LW1,LW2)
      RETURN
      END
C
      FUNCTION BETAIN(X,P,Q,BETA,IFAULT)
      LOGICAL INDEX
      DATA ACU /0.1E-7/
      BETAIN=X
      IFAULT=1
      IF(P.LE.0.0.OR.Q.LE.0.0) RETURN
      IFAULT=2
      IF(X.LT.0.0.OR.X.GT.1.0) RETURN
      IFAULT=0
      IF(X.EQ.0.0.OR.X.EQ.1.0) RETURN
      PSQ=P+Q
      CX=1.0-X
      IF(P.GE.PSQ*X) GOTO 1
      XX=CX
      CX=X
      PP=Q
      QQ=P
      INDEX=.TRUE.
      GOTO 2
    1 XX=X
      PP=P
      QQ=Q
      INDEX=.FALSE.
    2 TERM=1.0
      AI=1.0
      BETAIN=1.0
      NS=QQ+CX*PSQ
      RX=XX/CX
    3 TEMP=QQ-AI
      IF(NS.EQ.0) RX=XX
    4 TERM=TERM*TEMP*RX/(PP+AI)
      BETAIN=BETAIN+TERM
      TEMP=ABS(TERM)
      IF(TEMP.LE.ACU.AND.TEMP.LE.ACU*BETAIN) GOTO 5
      AI=AI+1.0
      NS=NS-1
      IF(NS.GE.0) GOTO 3
      TEMP=PSQ
      PSQ=PSQ+1.0
      GOTO 4
    5 BETAIN=BETAIN*EXP(PP*ALOG(XX)+(QQ-1.0)*ALOG(CX)-BETA)/PP
      IF(INDEX) BETAIN=1.0-BETAIN
      RETURN
      END
```