1984

INVESTIGATION OF OCULAR ARTEFACTS IN THE HUMAN EEG AND THEIR REMOVAL BY A MICROPROCESSOR-BASED INSTRUMENT

IFEACHOR, EMMANUEL CHIGOZIE

http://hdl.handle.net/10026.1/2341

http://dx.doi.org/10.24382/3872
University of Plymouth

INVESTIGATION OF OCULAR ARTEFACTS IN THE HUMAN EEG AND THEIR REMOVAL BY A MICROPROCESSOR-BASED INSTRUMENT

EMMANUEL CHIGOZIE IFEACHOR BSc(Hons), MSc, D.I.C., AMIEE

This thesis is submitted in partial fulfilment of the requirements of the Council for National Academic Awards for the degree of Doctor of Philosophy.

Department of Communication Engineering,
Plymouth Polytechnic,

in collaboration with the

Department of Neurological Sciences,
Freedom Fields Hospital,
Plymouth

-December 1984-

To

Egoyibe, Azu, Adaobi, and Preye.

Declaration

I hereby declare that whilst registered as a candidate for the degree of Doctor of Philosophy with the Council for National Academic Awards (CNAA) I have not been registered for any other qualifications of the CNAA or any other examining body.

E.C. IFEACHOR

INVESTIGATION OF OCULAR ARTEFACTS IN THE HUMAN EEG AND THEIR REMOVAL BY A MICROPROCESSOR-BASED INSTRUMENT

E.C. IFEACHOR

ABSTRACT

The Electroencephalogram (EEG) is widely used in clinical and psychological situations, but it is often seriously obscured by ocular artefacts (OAs) resulting from movements in the ocular system (eyeball,eyelids etc). The work described in this thesis is concerned with the problems of OAs in the human EEG, their removal both off-line and on-line, and the design and development of an on-line OA removal system, together with a critical review of the literature on the subject.

The work of Jervis and his co-workers was extended to further study OAs, to obtain improved measures of the effectiveness of OA removal, and to find the most effective model for removing OA on-line. A number of criteria were devised to compare the performance of several models, including a more reliable pictorial method. It was found unnecessary to use the vertical and horizontal EOGs for both eyes (ie. four EOGs) in a removal model, as previously reported. This was shown to be due to strong correlation between the EOGs.

It was shown that the assumption of uncorrelated error terms, implicit in present removal models, is invalid. To remedy this, the error terms were modelled as an autoregressive series.

New on-line removal algorithms based on numerically stable factorization algorithms were developed. Compared to the present on-line methods the algorithms are superior, requiring no subjective manual adjustments, or the co-operation of subjects which cannot always be guarranteed. The algorithms were shown to give similar results to their off-line equivalents. A simpler algorithm based on the present on-line method is also proposed as an alternative, but may lead to a reduced performance.

An important part of this research lay in the application of the results to the design and development of a new automatic OA removal system utilizing the algorithms described above.

ACKNOWLEDGEMENTS

I should like to acknowledge the help, advice and co-operation of the following people in this work.

I should like to give my sincere thanks to my three supervisors, Drs Jervis, Allen and Morris, for their support and for the many useful discussions, suggestions and advice throughout this research. This thesis gains from the critical comments of Dr Jervis and for this I am grateful. Dr Allen is to be thanked also for her interest in my welfare. I am grateful to Dr Morris for introducing me to the recursive Least Squares technique. My thanks also go to Drs Johnson and Wright for their expert advice on statistical matters.

The efficiency and assistance of the staff of several departments in the Polytechnic, such as the Computer Centre, the Learning Resources Centre (Library and Media Services), made this work and the preparation of this thesis that much easier. I gratefully acknowledge their help.

I am indebted to the students and members of staff of the departments of Communication Engineering and Electrical & Electronic Engineering for their help in various aspects of this work. My thanks go to Mr Santello for showing a great deal of interest in this work, for arranging and purchasing the various parts used in the project. My thanks also go to Mr Len. Roberts for constructing the ADC/DAC and the decoder boards, to Mr Sweet for allowing me to use the facilities in the Microelectronics Unit which made the program development possible and for useful suggestions during this work.

I should also like to thank all the members of the Neuroscience department of the Freedom Fields Hospital for their cooperation and assistance. In particular, my thanks go to Mr Hudson for his interest, useful discussions, and for arranging for some of the volunteers used to test the OAR system. I should also like to thank Miss Kieron Murdoch for tirelessly arranging for volunteers and assisting in the tests. I should also like to thank the many volunteers who made the evaluation of the OAR system possible.

I should thank the members of the Medical Physics department for their help, and in particular Mr Ivor Jones for constructing the input circuit boards.

I am grateful to Devon county council local education Authority for financial support, to the South West Regional Health Authority for providing funds for the purchase of the WICAT, and to the National Research and Development Corporation for providing the funds for the OAR system.

CONTENTS

Page

Chapter 8 - Review, Future Work, and Conclusion.

Appendices

List of Principal Abbreviations and Symbols

The estimate of a quantity is denoted by the symbol '^' above it. For example, the estimate of θ is $\hat{\theta}$.

EEG	Electroencephalogram
EOG	Electrooculogram
OA	Ocular Artefact
OAR	Ocular Artefact Removal
EMA	Eye Movement Artefact
EM, OM	Eye Movement, Ocular Artefact
HEM,VEM	Horizontal , Vertical Eye movement.
VL,HL	Vertical left ,vertical right EOG
VR, HR	vertical right, horizontal right EOG
$x(i)$,$y(i)$,$e(i)$	Vector of EOG samples,measured EEG, background or true EEG.
P	Covariance matrix of the EOGs
U,D	Upper unit and Diagonal factors of P
θ,γ,ϕ	Vectors of OA parameters,forgetting factor, and autoregressive parameters
OLS,RLS,GLS	Ordinary Least Squares,Recursive Least Squares and Generalized Least Squares.
ADC,DAC	Analogue-to-Digital Converter Digital-to-Analogue Converter
I/O	Input/output
FP	Floating Point
DFT,FFT	Discrete Fourier Transform,Fast Fourier Transform
RAM,ROM,EPROM	Random Access Memory,Read only Mamory, Erasable Programmable ROM
μP	microcomputer or Microprocessor

TTL	Transistor-Transistor Logic
IC	Integrated Circuit
MUX,DMUX	Multiplexer,demultiplexer
PGA	programmable Gain Amplifier
d,Cp	Durbin-Watson Statistic,Mallow Statistic
J	Sum of Squares of the error term
ACF,AR	Autocorrelation function ,Autoregressive
PSD,CSD	Power Spectral Density,Cross Spectral Density
FIR	Finite Inpulse Response

CHAPTER 1 INTRODUCTION

1.0 Preamble

Original work to investigate ocular artefacts (OAs) in the human Electroencephalogram (EEG), and the development of a microprocessor- based instrument for removing OA from the EEG is described in this thesis, together with a critical review of the existing literature on the subject. Chapter 1 constitutes a review of OA in the human EEG and formed the basis of a survey paper [1]. Chapters 2 to 8 describe the original contribution on OA.

The EEG signals are often seriously obscured by the superimposition of large electrical potentials associated with eye movements (EMs) and /or blinks. The source of these potentials is the corneo-retinal dipole and short-circuiting of it by the eyelid [2,3]. Thus eye movement results in dipole motion and scalp potential changes, modified by any associated eyelid movement. Blinks represent the case of temporary closure of the eyelids and the corresponding artefact is due to the motion of the eyelids over the cornea [3,4]. These EM, eyelid and blink artefacts are referred to collectively as ocular artefacts (OAs).

The EEG is widely used in clinical and psychological situations and subjects often exhibit random ocular movement. It is therefore, necessary in these situations to be able to remove the OA from the EEG so that the true EEG record can be analyzed. Such subjects may include brain-damaged babies and patients with frontal tumours. In both cases it is difficult to distinguish between the associated pathological slow waves and ocular artefacts. Patients with Huntington's Chorea, a fatal hereditary condition, exhibit random choreiform movements and their EEGs may be completely obliterated by OA. An event-related potential, known as the Contingent Negative Variation (CNV) [5], which has diagnostic usefulness for this patient category [6] is very vulnerable to ocular artefacts. It has been demonstrated that OAs have shapes very similar to the CNV [7-12] and remain the most serious methodological problem in CNV studies [13].

Ocular artefacts also make it difficult to automate the analysis of the EEG by computers. For example, the accuracy of a technique for locating brain lesions [14,15] which are characterized by slow waves, is reduced in the prescence of OA. In normal EEG recording very large amounts of data are often acquired, and to allow the clinician a better use of his time, automatic screening of the EEG records to indicate records containing clinically relevant features that require further examination is used. Due to OA the screening is often unsuccessful as a large number of normal records are labelled as abnormal [16]. A method for monitoring the effects of anaesthesia on the background EEG, and for studying the EEGs of epileptics and schizophrenics known as "dynamic topographic analysis" [17] will be ineffective in the presence of OAs.

Available literature on OA was surveyed and was found both extensive and distributed widely in several journals reflecting its relevance in the fields of neurology, electroencephalography, psychiatry etc. In the remaining sections of this chapter, the various aspects of the ocular artefact problem are reviewed including, the causes of OAs, methods for their removal, and practical problems involved. Besides covering the existing literature, the text is interspersed with comments on this research, to put it in perspective.

1.1 The Causes of Ocular Artefacts.

The human eye acts like an electric dipole with a positive cornea and negative retina. When the eyes move, the electrical dipole changes orientation so that the associated electrical potential on the scalp (that is OA) changes.

There are several types of ocular movements [22,23] of which the more relevant to this work are described here. In this thesis, the term 'Electrooculogram' (EOG) refers to the electric potential measured between two skin electrodes placed close to the eyes and due to ocular movements.

(a) Blinks

Blinks, figure 1.1, are characterized by a brief artefact potential of between 0.2 and 0.4 s in duration and occur at intervals of 1-10 s. The blink potential was shown to be attributable to the eyelid moving over the functional cornea [3,14], and not to eyeball movement as was previously thought. Movement of the eyelid over the positive cornea produces a shorting effect, which is removed when the eyelid is again raised.

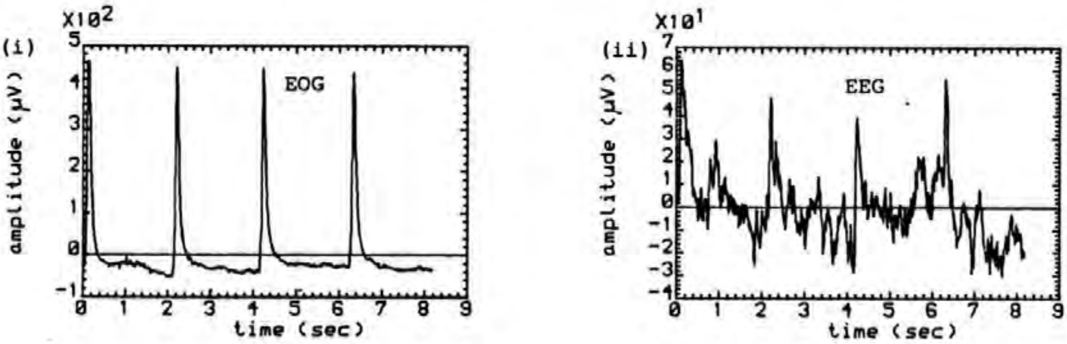


Figure 1.1 Blink Artefact

(i) Vertical right EOG (ii) Contaminated EEG

(b) Saccadic Eye Movements and Rider Artefacts.

Saccadic eye movements, figure 1.2a, are rapid conjugate movements of speeds between 100 and 500 degree/s . Normal everyday movements of the eye from one fixation point to another come under the category. These include the "jump and pause" fixation movements performed when reading or scanning a visual field.

Figure 1.2a

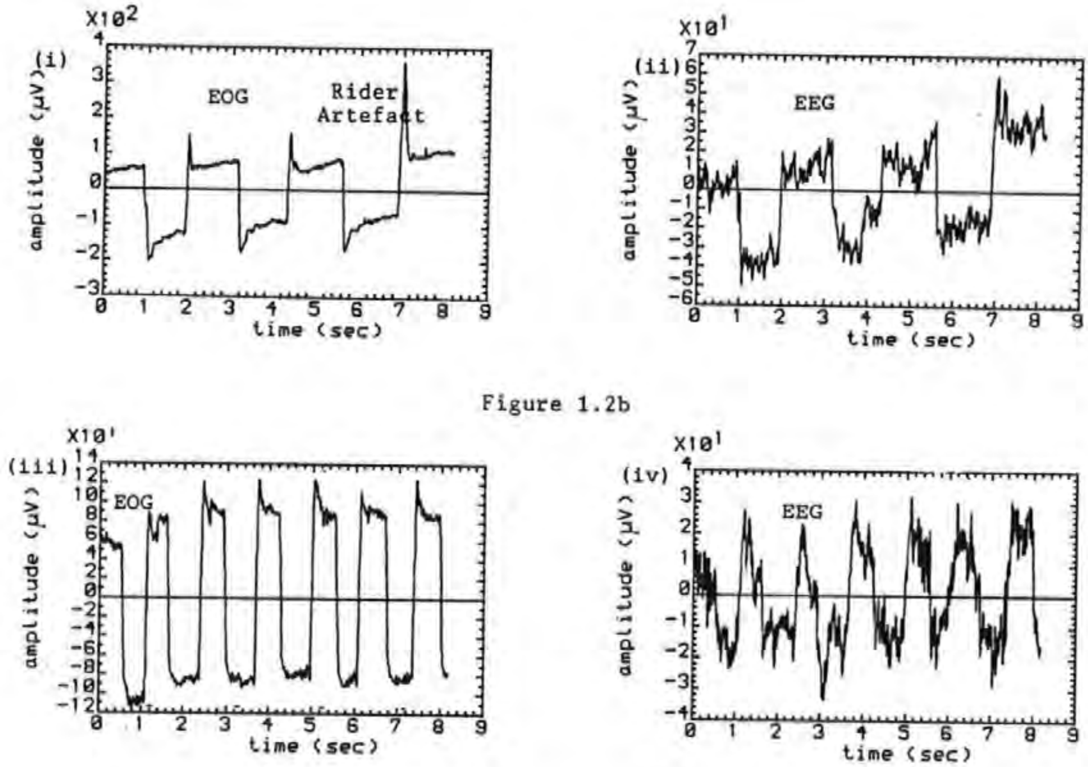


Figure 1.2b

Figure 1.2 Vertical- and Horizontal eye movement records Showing rider artefacts. (i),(iii) EOGs for vertical- and horizontal eye movements, respectively; (ii),(iv) the corresponding contaminated EEGs.

During vertical movements with the eyes open, a brief potential similar to blink potential is sometimes observed in the EOG [3,21], figure 1.2a. This artefact was called rider artefact [21] and was later demonstrated to be due to eyelid movement [3]. Thus it appears that rider artefact has the same cause as blinks. However, the stimulation mechanisms differ being saccades in the former case and blinks in the second. Rider artefact is reported to occur mostly during an upward vertical saccade or blink reflex. However, it has been seen in both vertical and horizontal EOG records, figure 1.2b

(c) Optokinetic Nystagmus

This type of ocular movement is elicited by a visual field which contains repeated patterns. The potential change produced in optokinetic nystagmus follows a characteristic saw tooth pattern. The eye fixates on a part of the field and tracks it as in pursuit movement. Then there is a return saccade and the process repeated.

(d) Vergence Eye Movement

In vergence, the eyeballs move in opposite directions. This disconjugate EM occurs when the eyes focus from far to near objects, or vice versa.

(e) Miniature or Fixation Ocular Movements

Miniature or fixation movements include a number of movements that are generally less than 1 degree in amplitude and occur when the eyes are supposedly stationary, as for example during a fixation on a stationary target. The three main miniature ocular movements are flicks, drift and tremor. Flicks, or microsaccades, are small EMs lasting for about 30mS and which are performed to correct a drift and redirect the eyes to the fixation target. Drift is a slow involuntary eye movement during fixation that makes the eye wander away from a target for short periods of time before correction by flicks. Tremors are rapid oscillatory EMs which may be superimposed on flicks and drifts. The frequency of a tremor may be as high as 30-150 Hz. Flicks and drift from both eyes are partially correlated while tremor is a disconjugate eye movement. The existence of miniature EM means there is always a measurable EOG signal and this is important because it ensures that the signal is 'persistently exciting', a necessary requirement for on-line recursive least squares removal method discussed in chapter 4.

(f) Eye Flutter

Eye flutter is a rapid ocular movement that tends to occur when the eyes are closed or nearly closed. It is increased during mental activity and may be mistaken for a seizure discharge especially during medication [24]. Flutter frequency is usually in the range 3-6 Hz but occasionally in the range of 8-10 Hz and sometimes as high as 14 Hz. Flutter is expected to be of minor importance in experiments where the eyes are kept open.

1.2 Methods of Measuring Ocular Movements

There are three main methods of measuring ocular movements, namely, optical, electro-oculography (EOG) and impedance oculography . Each method produces a voltage that is a function of the degree of ocular movement or position.

(a) Optical

Optical methods of detecting OM have been developed [23]. In most cases they consist of a light source, optical detectors, and a lens system. The limiting disadvantages of optical methods include that they cannot be used on unco-operative subjects such as children and some patient categories, they are bulky, and that they may not give an output when the eyes are closed.

(b) Electro-oculography (EOG)

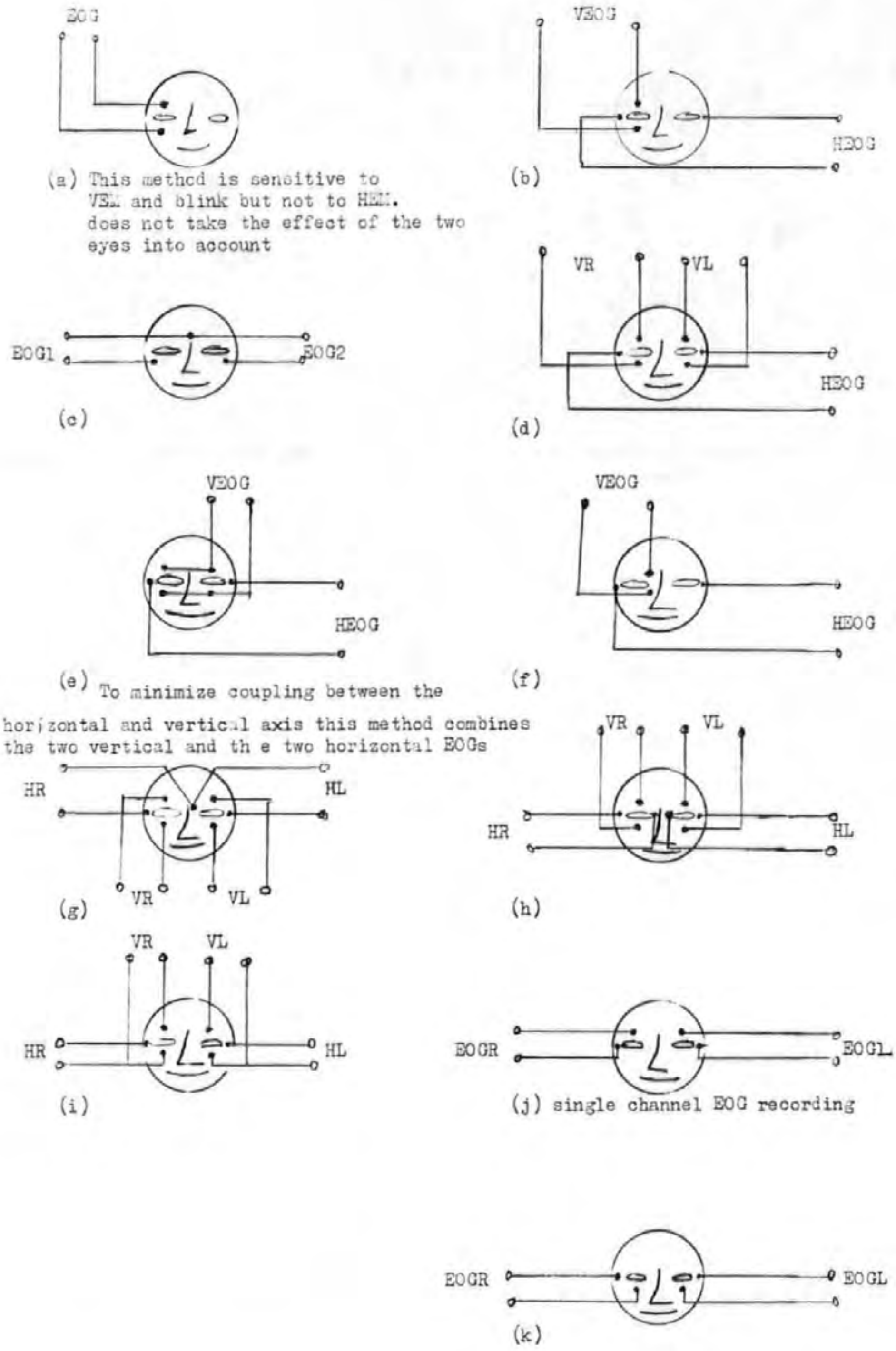
Ocular movements (OMs) (eye-, eyelid movement etc.) can be detected from th resulting EOGs [16,22]. This method has been used extensively to remove OAs [7,16,19,25-32]. Advantages of the EOG method are: It can be used whether the eyes are open or closed, it gives a voltage that is directly related to the degree of OM and hence to the scalp potential change produced by the ocular system (corneo-retinal dipole and/or eyelids). The method is simple to use and can be set up quickly with minimum calibration, and introduces no extra difficulty where electrodes are also being used to detect the EEG. The main disadvantages of the EOG are [22,23]: The measured potential may contain other artefacts unrelated to ocular movement. These may include electromyographic (EMG) signals resulting from muscle action and EEG contamination of the EOG. If both vertical and horizontal EOGs of the eye or both eyes are to be recorded, there may be considerable cross coupling between the axes and the eyes which may introduce some errors, as will be shown in chapter 2. Slow rolling eye movements occuring during drowsiness may be imposssible to distinguish from 'psychogalvanic sways' which are low drifts of potential related to changes in skin impedance. Physical movements of the recording electrode can occur especially if placed over a pulsating artery, or if the electrode is poorly applied.

(c) Impedance Oculography

The disadvantages of the EOG have led to the technique of impedance oculography [33,34]. This method is based on the principle that the impedance across the eye changes as the eye moves. A similar electrode placement method as used for the EOG is employed but in addition a low level high frequency current is injected into the electrodes. Movement of the eye changes the current distribution and produces a modulating signal which is proportional to the change in impedance between the electrodes . This impedance change can be extracted from the output of a demodulator. Although the technique is supposed to eliminate some of the disadvantages of the EOG method it has seldom been used.

1.2.1 Electrode Placements in Electro-oculography.

Figure 1.3 shows some of the commonly used EOG electrode placements. The placements are influenced by a number of factors, which include the need to obtain maximum information regarding an ocular movement and to use a minimum number of electrodes to record and identify the horizontal and vertical components of ocular movements. It is also desirable to reduce EEG contamination of the EOG and to obtain a voltage that is related to the extent of ocular movement. For these reasons electrodes are often placed across and close to the eyes in the horizontal and vertical planes. These may be referred to as the vertical and horizontal EOG channels respectively. The sites most commonly used are the outer and inner canthi (for the horizontal EOG) and the infra- and supra-orbital ridges (for the vertical EOGs). The most common electrode placement is that of figure 1.3b. This has the advantage of using few electrodes and minimizes cross coupling between the horizontal and vertical channels which can be a problem in multiple electrode placement [23], as demonstrated in chapter 2. Unfortunately, the method combines the horizontal EOGs of the two eyes, which may be significantly different in shape especially during horizontal eye movement [16,27]. Electrode placements (g),(h) and (i) are designed to overcome some of the above



(a) This method is sensitive to VEM and blink but not to HEM. does not take the effect of the two eyes into account

(b)

(c)

(d)

(e) To minimize coupling between the horizontal and vertical axis this method combines the two vertical and th e two horizontal EOGs

(f)

(g)

(h)

(i)

(j) single channel EOG recording

(k)

Figure 1.3 Some commonly used EOG measurement methods

disadvantages. Electrode placement (j) is sensitive to all ocular movements, but combines the horizontal and vertical EOGs of each eye. However, it has been pointed out [36] that the separation of the EOG into its vertical and horizontal components may be unnecessary if all that is required is the ability to measure the potential change due to ocular movement. In fact, electrode placement (j) has a number of advantages over placements such as (g), (h) and (i), viz: it eliminates the problem of cross-coupling between the horizontal and vertical EOG channels, and it could lead to a simpler OA removal model (see chapter 7). A similar technique may be used to reduce the contamination of the EOG by the EEG by using the electrode placement (k), although this may be less sensitive to ocular movements than electrode placement (j). In the off-line work the electrode placement (h) was used, but the best result was obtained using only one vertical channel and some combination of the two horizontal channels (see chapter 2). In evaluating the new Ocular Artefact Removal (OAR) system (chapter 7), electrode placements (h), (j), and (k) were all used, but only (h) and (j) gave satisfactory results.

1.3 Scalp Potential Distribution of Ocular Artefacts

A knowledge of scalp potential distribution of OA is useful in the study of surface evoked potentials [18,19]. A number of workers have studied the scalp distribution of OA using multiple electrode placements [18,37,38]. Peters

[37] showed that for subjects with both eyes functional, the scalp potential distribution is symmetrical on both sides of the scalp. Overton and Shagass [18] studied the OA distribution for a number of subjects and reported no significant inter-subject differences. However, they found significant differences in the distribution of OA due to blinks and VEM. A number of workers have, however, reported individual differences in the scalp distribution [25,39]. The OA at any point on the scalp depends on a variety of factors: extent of EM, eyelid position, distance of point from the eyes, type of ocular movement, the possibly distributed nature of the dipole, state of alertness of subjects, anatomical structures, and whether the subject's eyes are defective or not. Because of these factors, OA distribution on the scalp cannot be expected to be the same in all subjects. Visual examination of EEG records of a subject with a defective eye showed significant differences in the distribution of OA on the normal and defective sides. It was also found that in a subject completely blind on one eye that there was OA on the blind side of the scalp, albeit much reduced in amplitude and somewhat distorted. This reinforces the fact that OA on the scalp is a function of both eyes and that symmetrical distribution cannot always be assumed.

1.4 Methods for the Removal and Control of Ocular Artefacts

Several partial solutions have been proposed for removing the OA from the EEG [7,14,16,19,26,27,28,40]. These can be divided into three categories: Rejection, eye fixation, and Electro-oculogram (EOG) subtraction methods.

(a) Rejection Methods.

OA is controlled in the rejection methods by discarding data thought to contain significant OA. This can lead to what may be an unacceptable loss of data, e.g in studies involving unco-operative subjects such as children and psychiatric patients. In some Event Related Potential (ERP) studies sections of the record containing OA also contain useful information, and their deletion could make the data unrepresentative [19,28]. A common technique has been to monitor one EOG channel and when the level exceeded some fixed threshold the corresponding EEG channels were considered to contain unacceptable levels of artefact [41]. The threshold is subjectively determined prior to the experiment as that level of EOG for which no vestige of the EOG can be seen in the EEG. It has been shown that visual inspection of an EEG cannot be used to determine whether all the artefact has been removed [26,27] and so this method leads to the acceptance of EEG records with relatively low levels of artefact as being artefact free. There is also the possibility that ocular movements could be continuous so that the entire data would be rejected [40].

Attempts to implement an automatic rejection method in the frequency domain have been largely unsuccessful [14,29,40]. This is probably because of the inherent problems of the rejection method described above, the occasional spectral overlap between the artefacts and genuine slow waves, and the fact that OA does not necessarily occupy a fixed frequency band [40].

(b) Eye Fixation Methods:

To reduce the amount of data lost by the rejection method, it is often supplemented by asking subjects to fixate their eyes on a target and sometimes to avoid blinking or moving their eyes at critical times during the experiment. However, Rowland [42] claimed that as many as 50% of the subjects could not fixate successfully. A more controlled way of reducing OA by fixation was described by Borda and Hablitz [47]. The subject was required to detect a square wave displayed on a visual display unit (VDU), and this resulted in a reduction of EMA but not of blink artefacts. The method could also cause micro-saccades as the subjects searched the VDU for the square wave. Papakostopoulos et al [43] found that by asking the subject to focus on his own eyes in a mirror, directly in front of him, the amount of rejected data was significantly reduced.

(c) EOG Subtraction Methods.

The EOG subtraction methods probably offer the best means of removing OA from EEGS. At present, however, the various techniques reported do not completely solve the problem. Thus new approaches are continually being developed [16,19,28,30]. All of the methods are based on the principle that the OA is additive to the background EEG. Thus in discrete form:

$$y(i) = OA(i) + e(i) = \sum_{j=1}^{p} \theta_j x_j(i) + e(i) \qquad i=1,2 \ldots,M \qquad (1.1)$$

where y(i) and x(i) are the samples of the measured EEG and the EOGs respectively, and e(i) is the true EEG which may be regarded as an error term. The θ_j are constants of proportionality which will be called the ocular artefact parameters, and p is the number of parameters in the model. If the θ_j can be estimated then an estimate of e(i) can be obtained as:

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

where the $\hat{\theta}_j$ are the estimates of θ_j and $\hat{e}(i)$ is the estimate of e(i). The problem then is one of estimating the θ_j. The various EOG subtraction techniques differ primarily in the way θ_j is estimated, in the number of EOG signals that is used in (1.1) and the way these are measured. (cf, [8,14,16,27,28]).

In some methods it is assumed that the OA parameters are time-invariant [7,25,29]. This assumption, however, is untenable due to the different characteristics of the ocular artefacts [18,39] and the effects of the eye lid position [8].

The EOG subtraction method can be carried out either on-line (that is as the data is being acquired) [25,29]. or off-line at some later time [16,27]. The main advantage of the off-line methods over previously reported on-line methods is that more sophisticated removal techniques can be employed. However, in applications requiring real-time analysis, the delay involved when off-line methods are employed is unacceptable. The trend in EEG signal processing is clearly towards real-time processing [17,44] and it is then necessary to remove the artefacts on-line.

(i) Off-line Methods

Several off-line OA removal methods have been developed [7,14,16,19,26,27,28,40]. In this section some of the more promising ones will be described.

Verleger et al [19] used the electrode placement of 1.3a in their off-line technique. As was pointed out earlier and demonstrated in chapter 2, more than one EOG channel is necessary for OA removal.

Gratton et al [28] recognised that most of the existing removal methods did not distinguish between EM and blink artefacts, and that the θ_j were usually regarded, erroneously, as constants prior to and during the experiments . Thus they identified and obtained separate parameter estimates for blinks and EMs. The method however, is time consuming and will be very expensive in computing time for multi-channel situations. The method may only be suitable for ERP studies, and unsuitable for use on-line

Whitton et al [31] described an EOG subtraction method in the frequency domain. Their method suffers from several disadvantages which include the inability to use more than one EOG and the assumption of time -invariance of θ. The method is also time consuming and not suitable for multichannel use or for use on-line.

Quilter et al [16] proposed an off-line OA removal procedure which involves cross-correlating the EOG and EEG signals for each set of data, without assuming constant θ. These authors used only two EOG signals. Their technique has been extended to three and four parameter models by Jervis et al [26,27], who showed that a four-channel model involving both the vertical and the horizontal EOGs of both eyes was the most effective and that while the use of the vertical EOG due to only one eye might be necessary, the horizontal EOGs of both eyes were essential especially during horizontal eye movements (HEM). The suggested reason was that the dipoles of the two eyes tended to oppose each other during HEM, but not during vertical eye movement when the dipoles move in conjunction.

Fortgens and DeBruin [30] used least squares approach to obtain the θ for a four-EOG channel model. They proved theoretically that four EOG channels were necessary for complete correction. However, they assumed that four uncorrelated EOG channels existed. In chapter 2, it is shown that there are not four uncorrelated EOG channels so that the use of four EOGs is unecessary.

(ii) On-line Methods.

A number of on-line removal methods have been devised but have either been unsatisfactory or unwieldy. These on-line methods required manual calibration. Setting them up was thus time consuming, especially for multichannel recording. Also their calibration required subjects to move their eyes or blink repetitively. This ruled these methods out for use with children and unco-operative adults. Additionally, the method of assessing whether the OA was removed was subjective and the whole procedure required the operator to be familiar with the method. Furthermore, these methods were unable to deal with multiple artefacts, so that if a different ocular artefact occured after calibration, it was not removed properly [12].

McCallum and Walter [32] described one of the first on-line methods for reducing OA in the CNV recordings which used a potentiometer arrangement (see figure 1.4). The vertex EEG signal was referred to the centre terminal

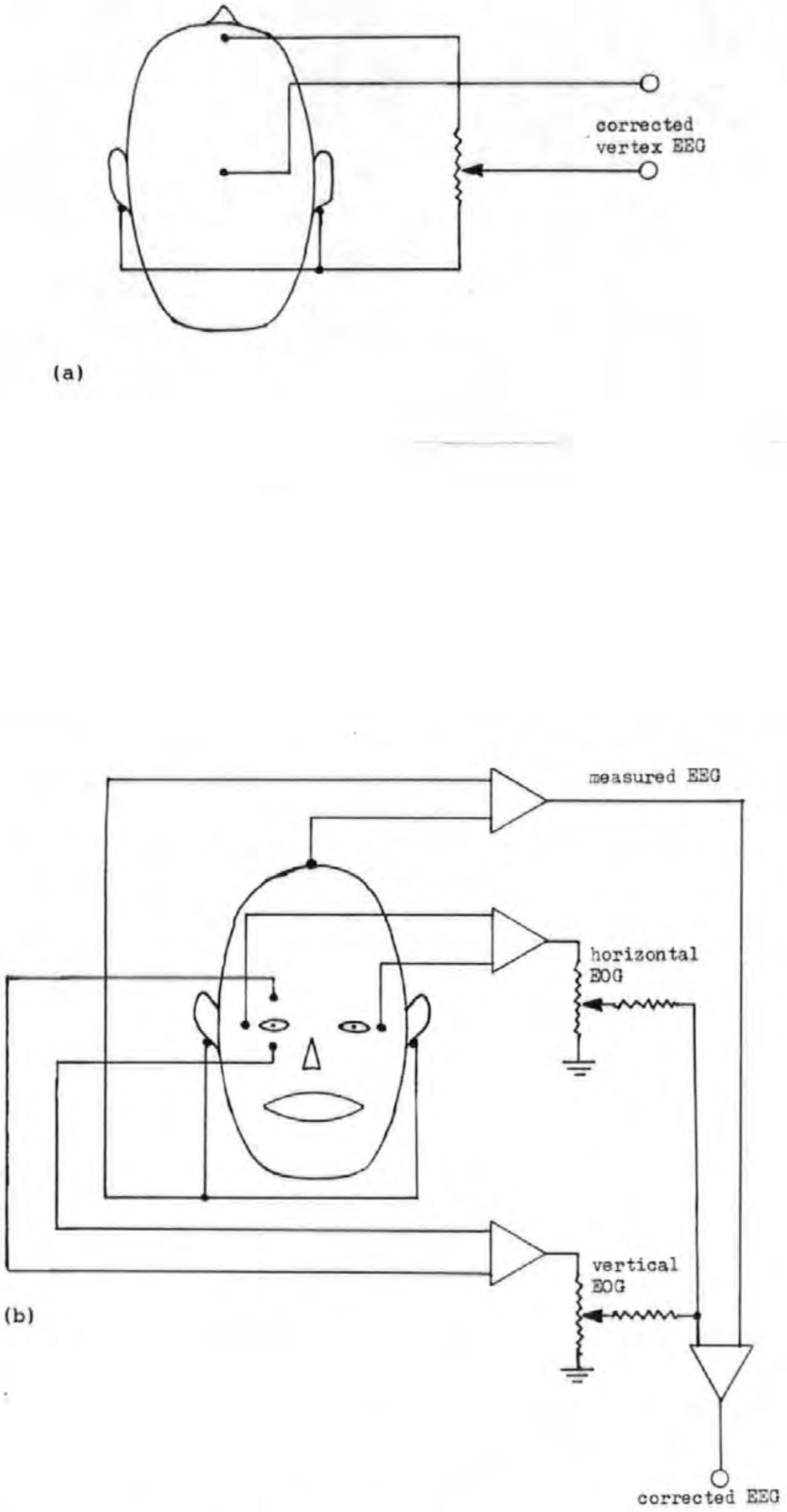


Figure 1.4 On-line Removal methods
(a) McCallum and Walter's method
(b) Girton and Kamiya's method

of the potentiometer. The other terminals of the potentiometer were connected to a frontal EEG electrode and linked earlobes. The 'corrector' was calibrated, prior to recording, by adjusting the centre terminal whilst the subject moved his eyes repetitively until there was no trace of OA in the EEG. The device was then left at this setting during recording. The method suffered from a number of disadvantages including those outlined earlier. It could not deal with lateral eye movement artefacts as these were not taken into account [25]. Jervis et al [27] have used this method and found it unsatisfactory. Jervis et al also showed that this analogue technique may actually be unsuitable for removing OAs due to VEM, as in all cases there was some coupling between the EOG and EEG when the EOG changed rapidly. This was probably due to the RC network formed by the potentiometer resistance and the electro-chemical capacitance of the electrode-skin interface. Wasman et al [11] reported that their attempt to use the method was unsuccessful as it resulted in "positive CNV".

The on-line method described by Girton and Kamiya [25], like the method of McCallum and Walter [32], used a potentiometer arrangement (figure 1.4 b) and required manual calibration. Although a number of modifications were made to cater for lateral eye movement and to improve the signal quality, the method still suffered from the disadvantages mentioned earlier. A number of workers have used the method and found it unwieldy and inefficient in removing OA [14,15,31] .

Barlow and Rémond [29]proposed a method that is similar to that of Girton and Kamiya [25]. The essential differences between the two were that Barlow and Rémond's method required fewer amplifiers when used in multichannel recording and had facilities such as those for changing the sign of the EOGs used to remove the OA.

In chapter 4, a new on-line method for removing ocular artefacts will be described. It uses numerically stable algorithms based on the efficient recursive least squares technique. These algorithms have been implemented in a microprocessor-based system the design and development of which are described in chapters 5 to 7. The on-line removal system requires no subjective manual adjustment and processes all signals digitally and thus has all the advantages of digital signal processing methods.

1.5 Practical Problems of Removing Ocular Artefacts from the EEG

The removal of OA from the EEG poses a number of problems some of which have already been mentioned. Perhaps the most serious problems are (i) the lack of a convincingly accurate method of assessing the reliability and effectiveness of removing OAs and the resulting side effects, (ii) the effects of EOG (or secondary) artefacts on the corrected EEG, (iii) the influence of the EEG/EOG statististics, (iv) the relative effect of pathological slow waves, and (v) the effect of overloading of the data acquisition systems. A brief discussion of these problems will be given here. Solutions to some of them will be given in the main body of the thesis.

1.5.1 Assessment of the Reliability and Effectiveness of Ocular Artefact Removal

(a) Reliability

The method commonly used is to obtain the parameter estimates, θ_i of a number of subjects in two or more experiments or sessions. If these parameter estimates are highly correlated then they are said to be reliable. The implicit assumption of course is that exactly the same factors and ocular movements would be involved in the experiments, which may not be true.

Verleger et al [19] attempted to establish the reliability of subject-specific parameters, and implicitely concluded that they were unreliable. However, the assumption was made that the parameter estimates, θ_i for any individual would be the same for both a comparison- and concept-learning task. Different ocular movements may, however, be elicited in the two tasks which would invalidate the assumption. This would be particularly so with the sixty seven subjects aged ten to thirteen years, of whom twenty five were mentally retarded, used in their experiment. The wide variation in the estimates for each task and between tasks were probably largely due to genuine factors and perhaps to the inadequacy of the model used (see section 1.4(c)).

Gratton et al [28] assessed the constancy of the parameter estimates over different trials for the same subjects in and between experiments. Their results showed that for a given subject the parameter estimate was not constant during or between experimental sessions and that there were inter-subject differences in the parameter estimates. Their results showed that the variation in the subject specific parameter estimates between sessions was larger than during a given session, which they pointed out may be due to not placing the electrodes at exactly the same position in the sessions. This variation emphasises the need for removing the OA not only on an individual basis but also adaptively.

(b) Effectiveness of Ocular Artefact Removal

The most commonly used method of assessing the effectiveness of the removal techniques is visual inspection of the corrected EEG for any trace of OA. Another approach is to compare the slow waves of clinical interest obtained from the corrected EEG and that from an artefact-free EEG. If they are reasonably similar then the correction procedure is said to be effective or valid.

Verleger et al [19] applied both methods to their correction procedure and found it effective. They also compared the covariance between the ERP derived from the corrected EEG and the EOG with that between the ERP derived from the rejection method and the EOG. The former was significantly lower, and was taken as an indication that their correction procedure was superior to the rejection method. However, this result is to be expected since the correction criterion is minimisation of the covariance between the measured EOG and EEG.

Gratton et al [28] assessed the effectivenes of their correction procedure by determining how different the ERP from the corrected EEG was to the 'true' ERP, obtained from trials with insignificant artefact selected by the rejection method. To obtain a quantitative assessment they computed a deviation index which was the sum of squares of the deviations of the corrected ERP from the true ERP. They concluded that in general the corrected ERP resembled the true ERP more than the the uncorrected one. The validity of the method rests upon a common assumption that the ERP from trials selected by the rejection method is a good estimate of the ERP, but as the authors pointed out this assumption may sometimes be incorrect. For example, if the number of trials used to obtain the estimates of the true ERP is small, the noisy EEG would affect the result. Furthermore, rejection methods inevitably accept trials with low level artefact which could affect the result. Such effects may explain why the deviation index never becomes zero.

Jervis et al [27] introduced a technique based on the autocorrelation function (ACF) of the corrected EEG to assess quantitatively the effectiveness of OA removal, where an estimate of the ACF of a data sequence, x(n), n=0,1,..N-1, is given by:

$$\hat{C}_{xx}(k) = \sum_{i=0}^{N-k-1} \frac{x(i)x(i+k)}{\sum_{i=0}^{N-k-1} x^2(i)} \qquad k=0,1,\ldots,N-1 \qquad (1.3)$$

$$\hat{C}_{xx}(-k) = \hat{C}_{xx}(k)$$

Using an EEG record contaminated by a rectangular EOG they found that after OA removal the ACF of the EEG lost its triangularity, an indication that the rectangular artefact has been removed. If the ACF was 'random' the OA was said to have been completely removed. To quantify the effectiveness of OA removal in the case where the ACF was not random the autocorrelation coefficient (ACC) at a lag of 2 seconds was obtained by measuring the peak-to-peak value of the ACF at 2 second lag. The lower the ACC, the more effective the model used was said to be in removing the OA. This was used as the basis for comparing the performance of two methods of removing OA. Thus the authors found that the analogue method of McCallum and Walter [32] was inferior to the correlation technique method [16,26,27], and as described earlier (p18), that a four-parameter model was the most effective in removing OA. This quantitative technique however, was not sensitive enough in many cases, and may not be reliable as a quantitative study of the performance of several models showed (see chapter 2).

1.5.2 Secondary Artefacts

As stated earlier, a disadvantage of using the EOG in removing OA is that it may contain other artefacts not related to ocular movements, such as EMG or EEG. Because in the EOG subtraction method a fraction of one or more of the EOGs is subtracted from the measured EEG, any artefacts in the EOG would be introduced into the EEG as 'secondary' artefacts.

The technique usually used to deal with this problem is to filter out the EOG artefacts which are mostly high frequency activities [21,31,45] However, the filters used have poor amplitude and phase characteristics which could adversely affect the OA removal [31]. An efficient zero phase low pass digital filter that will introduce minimal distortion is described in chapter 3 together with a spectral study of the OA problem.

The extent of the effects of secondary artefact has not been settled. Verleger et al [19] stated that the effect of EEG contamination of the EOG was by and large unnoticeable. Some workers reported only a small increase in noise in the corrected EEG which was only significant in a few cases [16]. Others [22] stated that in most cases the EEG contamination was less in magnitude than the effect of half a degree of eye movement, thus making the problem insignificant. In some subjects with relatively large EOG artefacts, no apparent relationship was found between the scalp EEG and the EOG [22]. The variation of the parameter estimates with time was studied in this research and it was found that the variances of θ for those EOG channels with large artefacts were larger than those for less noisy EOGs. This was interpreted as indicating that secondary artefacts do affect the correction process. This is discussed in more detail in chapter 4.

1.5.3 The Influence of the EEG/EOG Statistics.

It is usually assumed that the background EEG is a random process, uncorrelated with the EOGs [16,28,30]. As will be shown in chapter 2, the assumption of randomness is not valid and requires corrective steps to be taken. Methods will be discussed in chapter 2 to take correlated EEG into account.

To obtain good removal of OA, it is necessary to use more than one EOG signal [16,25,26,27,30]. However, as shown in chapter 2, when more than one EOG signal is used computational and statistical difficulties due to the linear dependencies between the EOGs may arise and these need to be taken account of in OA models.

1.5.4 OA Removal in the Presence of Pathological Waves.

The removal of OAs by the EOG subtraction technique has to be considered carefully when OAs are present together with the ERPs or with pathological slow waves of interest. Since the fraction of the EOG subtracted is obtained from the degree of correlation between the EOG and its component in the EEG, the presence of ERPs or slow waves of similar shape to the OA can lead to the subtraction of a fraction which depends on the ERP or slow waves as well as the EOG. This results in over-correction. For example, it is difficult to distinguish between the electrical potential contributed by eye movement and the true CNV potential [7,10-12]. Muras and Binnie [46] used judicious electrode placement to distinguish between OA and slow waves. It will probably be necessary to distinguish between the various ERP components, slow waves, and OAs by examining the propagation of the electric potentials over the scalp by using a number of scalp electrodes. This is an area that was not investigated in this research, but needs to be studied in the future.

1.5.5 Overloading of the Data Acquisition System

Another problem which has been largely ignored, is the possibility of overloading the amplifier and the analogue to digital conversion system. Quilter et al [16] found this to be their most significant problem. Overloading causes erroneous parameter estimates and hence a reduction in the effectiveness of the correction. The common approach is to set the dynamic range of the analogue-to-digital converter (ADC) such that overloading is infrequent. This often involves a compromise between utilising most of the dynamic range of the converter and avoiding overloading. Verleger et al [19]reported a data loss of about 4% because of overloading. A method is proposed here for dynamically changing the range of the ADC (see chapter 5) which should alleviate the overloading problem.

1.6 Experimental Data.

The data used for the off-line work and computer simulations (chapters 2 to 4), was obtained in a previous study into the problem of ocular artefacts in the EEG [26,27]. Separate vertical and horizontal EOG signals were recorded for each eye from silver-silver chloride electrodes placed above and below the eye, and from similar electrodes at the outer canthus and the nasion. One channel of EEG was obtained from a similar electrode placed at the vertex referred to linked earlobes. Subjects with no known eye defects were asked to make periodic eye movements in the vertical, horizontal and diagonal directions between pairs of light emitting diodes located on a wooden screen. This corresponded to about 20° of eye movement and ensured that the performance of the models were evaluated under a known type of OA.

The EOG and EEG signals were fed to an electroencephalograph to produce paper chart record and to a data acquisition system where they were filtered, digitized and stored on a disc. The data was later transferred to a main frame computer for analysis.

1.6.1 Data Preprocessing

The mean of the data samples of each channel was computed and removed from each sample. Any low frequency trend was removed after removing the mean by differencing [48], and this also helped to reduce the effects of serial correlation of the background EEG on the estimates of θ.

Differencing involves generating a new set of data $\overline{x}(i)$, say, from x(i) [48]. That is,

$$\overline{x}(i) = x(i) - x(i-1) , \quad i= 2,3, \ldots,M \qquad (1.4)$$

Thus after removing the mean and differencing two sets of data were obtained for each EEG record, viz:

(a) Data with the mean removed, and

(b) differenced data (corresponding to $\overline{x}(i)$ above).

References for Chapter 1

[1]JERVIS B.W.,IFEACHOR E.C.,ALLEN E. and MORRIS E.L.
The Removal of Ocular Artefacts from the Electroencephalogram: A Survey. (Submitted to Electroenceph. and Clin. Neurophysiol.)

[2]MOWRER O.H.,RUCH T.C. and MILLER N.E.
The Corneo-Retinal Difference as the Basis of the Galvanometric Methods of Recording Eye Movements Psychological Bulletin,1935,Vol. 32, pp. 566-567.

[3]BARRY W. and JONES G.M.
Influence of Eyelid Movement upon Electro-oculographic Recording of Vertical Eye Movements. Aerospace Medicine, 1965,Vol. 36,pp. 855-858.

[4]MATSUO F.,PETERS J.F. and REILLY E.L.
Electrical Phenomena associated with Movements of the Eyelid. Electroenceph. and Clin. Neurophysiol., 1975,38,pp. 507-511.

[5]WALTER W.G.,COOPER R.,ALDRIDGE V.J.,MCCALLUM W.C. and WINTER A.L.
Contingent Negative Variation: An Electric Sign of Sensorimotor Association and Expectancy in the Human Brain. Nature,1964,Vol. 203,pp380-384.

[6] JERVIS B.W.,ALLEN E.,JOHNSON T.E., NICHOLS M.J. and HUDSON N.R.
The Application of Pattern Recognition Techniques to the Contingent Negative Variation for the Differentiation of Subject Categories. IEEE Trans. Biomed. Eng.,1984,Vol. 31,pp. 342-349.

[7] HILLYARD J.A. and GALAMBOS R.
Eye Movement Artifact in the CNV. Electroenceph. and Clin. Neurophysiol.,1970,24,pp. 173-182.

[8] HILLYARD J.A.
Methodological Issues in CNV Research. In Bioelectric Recording Techniques part B, Electroencephalography and Human Brain Potentials. Thompsom, R.F. and Patterson,M.M. (eds). Academic Press,1974,pp. 259-280.

[9] LOW M.D.,BORDA R.P.,FROST J.D. and KELLAWAY P.
Surface Negative Slow Potential Shift associated with Conditioning in Man. Neurology,1966,16,pp 771-782.

[10] STAUMANIS J.J.,SHAGASS C. and OVERTON D.A.
Problems associated with Applications of the Contingent Negative Variation to Psychiatric Research. JNL of Nervous and Mental Disease,1969,148,pp. 170-179.

[11] WASMAN M.,MOREHEAD S.D.,LEE H. and ROWLAND V.
Interaction of Electro-ocular Potentials with the Contingent Negative Variation. Psychophysiology,1970,vol.

7,pp. 103-111

[12]WEERTS T.C. and LANG P.J.
The Effects of Eye Fixation and Stimulus and Response Location on the Contingent Negative Variation (CNV). Biological Psychology,1,1973,pp. 1-19.

[13]TECCE J.J.
Contingent Negative Variation (CNV) and Psychological Processes in Man. The American Psychological Assoc., Vol. 77,No. 2.,1972,pp. 73-108.

[14] GOTMAN J.,SKUCE D.R.,THOMPSON C.J.,GLOOR P., IVES J.R. and RAY W.F.
Clinical Applications of Spectral Analysis and Extraction of Features from Electroencephalograms with slow Waves in Adult Patients. Electroenceph. and Clin. Neurophysiol., 1973,35,pp. 225-235.

[15] GOTMAN J., GLOOR P. and RAY W.F.
A Quantitative Comparison of Traditional Reading of the EEG and Interpretation of Computer Extracted Features in Patients with Supertentorial Brain Lesions. Electroenceph. and Clin. Neurophysiol.,1975,38, pp. 623-639.

[16] QUILTER P.M.,MACGILLIVRAY and WADBROOK D.G.
The Removal of Eye Movement Artefact from EEG Signals Using Correlation Techniques. Random Signal Analysis, IEE

Conference Publication, No. 159,1977, pp. 93-100.

[17] HARRIS C.J.
Brain Waves appear on T.V. in Real-time. Electronics,24, 1983,pp. 47-48.

[18] OVERTON D.A. and SHAGASS C.
Distribution of Eye Movement and Eye Blink Potentials over the Scalp. Electroenceph. and Clin. Neurophysiol.,1969,27, p. 546.

[19] VERLEGER R.GASSER T. and MOCKS J.
Correction of EOG Artifacts in Event-Related Potentials of the EEG: Aspects of Reliability and Validity. Psychophysiology, 1982,19,pp. 474-480.

[20] FENDER D.H.
Control Mechanisms of the Eye. Scientific American,1964, vol. 211,pp. 24-33.

[21] FORD A.
Significance of Terminal Transients in Electrooculographic Recordings. Archives of Opthalmology,1959,Vol. 61,pp. 899-906.

[22] SHACKEL B.
Eye Movement Recording by Electro-oculography. in A manual of Psychophysiological Methods, VENABLES,P.H. and MARTIN,I. (ed),North Holland Pub. Co., 1967,pp.

301-334.

[23] YOUNG L.R. and SHEENA D.
Survey of Eye Movement Recording Methods. Behaviour Res. Methods. and Instrumentation,1975,Vol. 7,pp. 397-429.

[24] HARLAN W.L.,WHITE P.T. and BICKFORD R.G.
Electrical Activity Produced by Eye Flutter Simulating Frontal Electroencephalographic Rhythms. Electroenceph. and Clin. Neurophysiol.,1958,Vol. 10,pp. 164-169.

[25] GIRTON D.G. and KAMIYA J.
A Simple On-line Technique for Removing Eye Movement Artifacts from the EEG. Electroenceph. and Clin. Neurophysiol.,1973, 34,pp. 212-216.

[26] JERVIS B.W., NICHOLS M.J.,ALLEN E.,HUDSON N.R. and JOHNSON T.E.
The Quantitative Assessment of Electroencephalograms Corrected for Eye Movement Artefacts. First European Conf. on Signal Processing,Lauseane,1980.

[27] JERVIS B.W., NICHOLS M.J.,ALLEN E.,HUDSON,N.R. and JOHNSON T.E.
Comparison of Two Methods for Removing Eye Movement Artefact from the EEG. (submitted to Electroenceph. and Clin. Neurophysiol.)

[28] GRATTON G.,COLES M.G.H. and DOUCHIN E.
A new Method for Off-line Removal of Ocular Artefacts. Electroenceph. and Clin. Neurophysiol.,1983,55,pp. 468-484.

[29] BARLOW J.S. and RÉMOND A.
Eye Movement Artefact Nulling in EEGs by Multichannel On-line EOG Subtraction. Electroenceph. and Clin. Neurophysiol., 1981,55, pp. 418-423.

[30] FORTGENS C. and DE BRUIN M.P.
Removal of Eye Movement and EOG Artifacts from the non Cephalic Reference EEG. Electroenceph. and Clin. Neurophysiol., 1983,56,pp. 90-96.

[31] WHITTON J.L.,LUE F. and MOLDOFSKY H.
A Spectral Method for Removing Eye Movement Artefacts from the EEG. Electroenceph. and Clin. Neurophysiol., 1978,44, pp. 735-741.

[32] McCALLUM W.C. and WALTER W.G.
The Effects of Attention and Distraction on the Contingent Negative Variation in Normal and Neurotic Subjects. Electroenceph. and Clin. Neurophysiol.,1968,25, pp. 319-328.

[33] GEDDES L.A.,MCGRADY J.D. and HOFF H.E.
The Impedance Nystagmogram - A Record of the Level of Anesthesia in the Horse. South Western Verter-

inarian,1965, Vol. 19,pp. 23-25.

[34] SULLIVAN G. and WELFMEN G.
The Impedance Oculogram. JNL. of Applied Physiollogy,1963, Vol. 18,pp 215-216.

[35] HENDERSON C.J.,BUTLER S.R. and GLASS A.
The Localisation of Equivalent Dipoles of EEG Sources by the Application of Electrical Field Theory. Electroenceph. and Clin. Neurophysiol.,1975,39,pp 117-130.

[36] CONNOLLY J.F. and KLEIMAN K.M.
A Single Channel Method for Recording Vertical and Lateral Eye Movements. Electroenceph. and Clin. Neurophysiol.,1978, 45,pp. 128-129.

[37] PETERS P.F.
Surface Electrical Field Generated by Eye Movements. Amer. JNL of EEG Technology,1967,Vol. 7,pp.27-40.

[38] ZAO Z.Z.,GELBIN and RÉMOND A.
Le Champ Electrique de L'oeil. La Semaine des Hopitaux de Paris,1952,Vol. 28,pp. 1-8.

[39] CORBY J.C. and KOPELL B.S.
Differential Contributions of Blinks and Vertical Eye Movements as Artifacts in EEG Recordings. Psychophysiology, 1972,Vol. 9,pp. 640-644.

[40] GEVIN A.S.,YEAGER C.O.,ZEITLIN G.M.,ANCOLI S. and DEDON M.F.
On-line Computer Rejection of EEG Artefacts. Electroenceph. and Clin. Neurophysiol.,1977,42,pp. 267-274.

[41] JOHN E.R.,KARMEL B.Z.,CORNNIG W.C.,EASTON,P.,BROWN D., AHU H.,JOHN M.,HARMONY T.,PRICHEP L.,TORO A.,GERSON I., BARTLETT F.,THATCHER R.,KAYE H. VALDES P. and SCHWARTZ,E.
Neurometrics. Science,June,1977,Vol. 196,pp. 1393-1410.

[42] ROWLAND V.
Cortical Potential (direct Current Potential) in Reinforcement and Learning,In STELLAR,E. and SPRAGUE,J.M. (Eds),Progress in Physiological Psychology,Vol. 2, Academic Press,New York,1968,pp 1-77.

[43] PAPAKOSTOPOULOS D.,WINTER A. and NEWTON P.
New Techniques for the Control of Eye Potential Artefacts in Multi Channel CNV Recordings. Electroenceph. and Clin. Neurophysiol. ,1973,34,pp. 651-653.

[44] BARLOW J.S.
Computerized Clinical Electroencephalography in Perspective. IEEE Trans. Biomed. Eng.,1979,26,pp. 377-390.

[45] REIMAN V.,KOORTH M. and KEIDEL W.D.

Korrelations Analyse Von EEG Und Augenbewengungen. Vision Res. ,1974,Vol. 14,pp. 959-963.

[46] MURAS J.S. and BINNIE C.D.
The Recognition of Frontal Slow Activity in the Prescence of Eye Movements. The Proc. of the Electrophysiol. Technol. Assoc. ,1970,Vol. 17,pp. 131-143.

[47] BORDA R.P. and HABLITZ J.J.
Use of Simple Visual Display to Reduce Eye Movement Artifacts in CNV Recordings. Electroenceph. and Clin. Neurophysiol.,1973,34,pp. 433-436.

[48] JENKINS G.M. and WATTS D.G.
Spectral Analysis and its Applications. Holden-Day, 1968.

Chapter 2 OFF-LINE REMOVAL OF OCULAR ARTEFACTS FROM THE EEG SIGNALS

2.0 Introduction

In this chapter, the ocular artefact problem is described using system identification methods [1,2] and a more formal development of the removal algorithm than was given before is presented. Statistical and computational difficulties in removing OA are discussed and, where possible, solutions given. An extension of the work of Jervis et al [5,6] to find the most effective model for implementation on-line is also described here. The work described in this chapter has been written up for publication [3].

In section 2.1, the OA problem is stated and parameter estimation methods discussed. Section 2.2 gives a formal development of the OA problem using the least squares method, and in section 2.3 the problems of removing OA by this method is discussed. Three problems of importance are, the statistics of the background EEG which is often assumed random, the linear dependence between the EOGs which can lead to difficulties in computation, and the EOG artefacts. In section 2.4 several models are compared with a view to finding the most suitable for on-line removal of the OA. This section fulfils the need to carry out a systematic comparison of models to establish their relative effectiveness in removing OA.

2.1 Ocular Artefact Problem Representation

The fundamental assumption in all EOG subtraction methods is that the OA is additive to the background EEG and that the path between the eye and the EEG electrodes is linear. This situation is represented in figure 2.1.

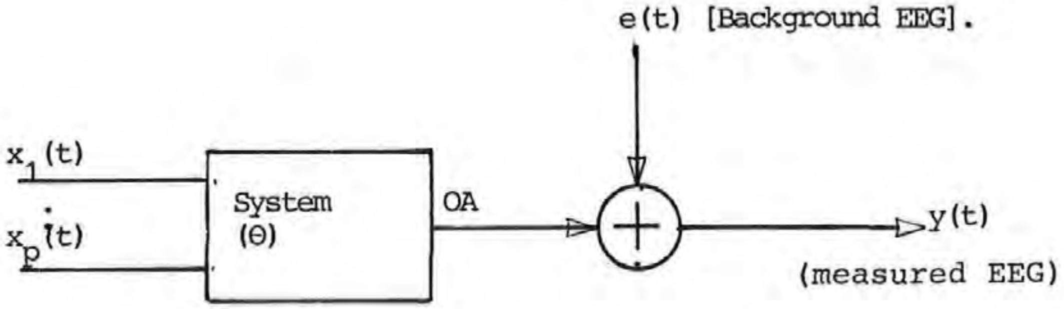


Figure 2.1 Block diagram Representation of the OA System

where $x_i(t)$ and $y(t)$ are, respectively, the measured EOGs and EEG. e(t) represents the error term or the backgound EEG.

Thus, in discrete form, the system output can be written in terms of the inputs:

$$y(i) = \theta_1 x_1(i) + \ldots + \theta_p x_p(i) + e(i)$$
$$= x^T(i)\theta + e(i) \;, \quad i = 1,2, \ldots, M \qquad (2.1)$$

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

and

$\theta^T = [\; \theta_1 \;\; \theta_2 \;\; \ldots \; \theta_p]$

The T indicates a transposition. (It is assumed in this equation and in the rest of this thesis that the appropriate mean values have been computed and removed from the data samples, x(i) and y(i), as described in chapter 1. Otherwise this can be allowed for by including a constant term on the right hand side of the equation 2.1).

The problem is to obtain estimates of Θ, given x(i),y(i), i=1,2,..,M, and then to use these estimates to obtain an estimate of e(i) (the true EEG). This problem is illustrated in figure 2.2.

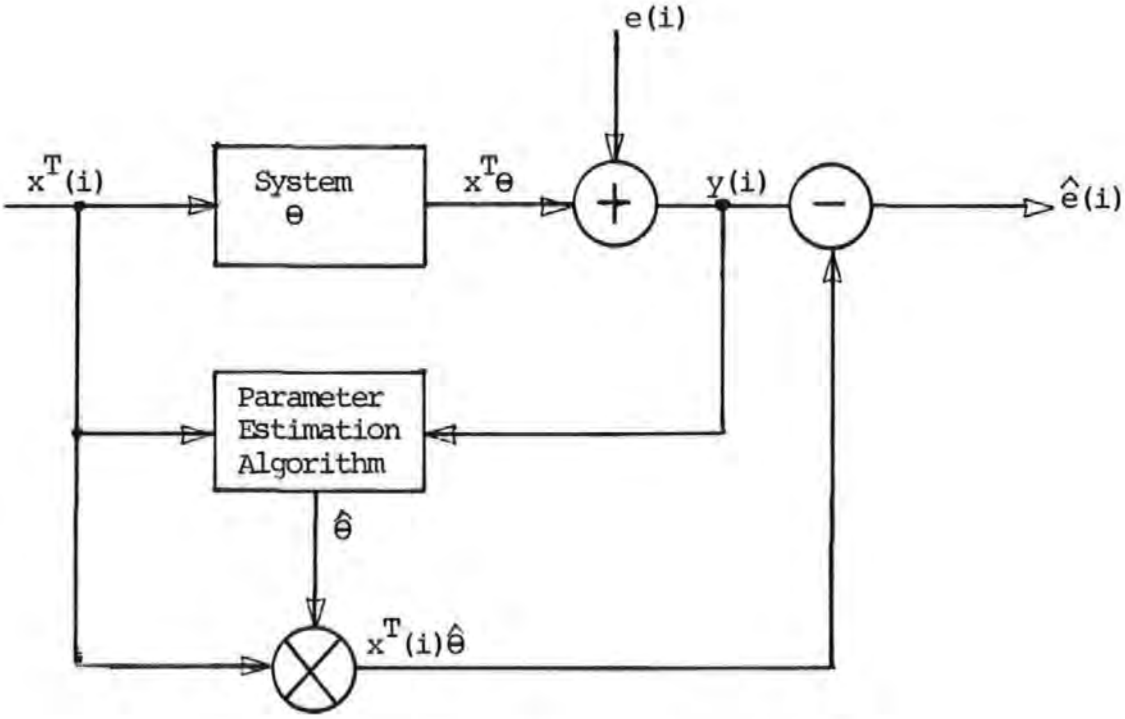


Figure 2.2 Block diagram representation of the removal of ocular artefacts from the EEG

There are various methods that may be used to estimate the system parameters, Θ. These include the ordinary least squares and maximum likelihood estimation methods [1,2].

Of the various methods the least squares method is the most widely used. This is due mainly to its simplicity and good statistical properties.

2.2 Ocular Artefact Removal Algorithm- the Ordinary Least Squares Method.

Equation 2.1 can be written more compactly for all m samples as in equation 2.2.

$$Y_m = X_m \theta_m + E_m \tag{2.2}$$

where,

$$Y_m = \begin{bmatrix} y(1) \\ y(2) \\ \vdots \\ y(m) \end{bmatrix}, \quad X_m = \begin{bmatrix} x^T(1) \\ x^T(2) \\ \vdots \\ x^T(m) \end{bmatrix}, \quad \theta_m = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_p \end{bmatrix}, \quad E_m = \begin{bmatrix} e(1) \\ e(2) \\ \vdots \\ e(m) \end{bmatrix}$$

The Suffix m indicates that each matrix was obtained with all m samples. E_m represents the background, or true, EEG.

Assuming that e(i) is an uncorrelated sequence with zero mean and constant variance, then optimum estimates of the coefficients θ_m can be obtained by minimizing J, the sum of squares of the error term E_m:

$$J = \sum_{i=1}^{M} e^2(i) = E_m^T E_m \tag{2.3}$$

This minimization leads to equation 2.4 (see appendix A1 for details):

$$\hat{\theta}_m = [X_m^T X_m]^{-1} X_m^T Y_m \qquad (2.4)$$

This equation gives the ordinary least squares (OLS) estimate of θ_m which can be obtained using any suitable matrix inversion techniques.

Having obtained $\hat{\theta}_m$, estimates of the OA and hence the background EEG, E_m, can be obtained from (2.2). E_m are referred to as the residuals in statistics. Implicit in the OLS method is the assumption that the EOGs, x(i) , are not perfectly collinear, so that the matrix $(X_m^T X_m)$ is invertible.

It is worth noting that all the EOG subtraction methods that appear in the literature under different names are essentially least squares methods and thus subject to the above assumptions. For example, it is shown in appendix A1 that the so-called correlation technique method [5-7] is in fact an OLS method.

2.2.1 Multichannel EEG Ocular Artefact Removal Algorithm.

The removal algorithm described above can be extended to the multichannel case, where there is more than one EEG signal. Thus, following the approach described above a system with p EOG inputs and q measured EEG outputs is shown in figure 2.3.

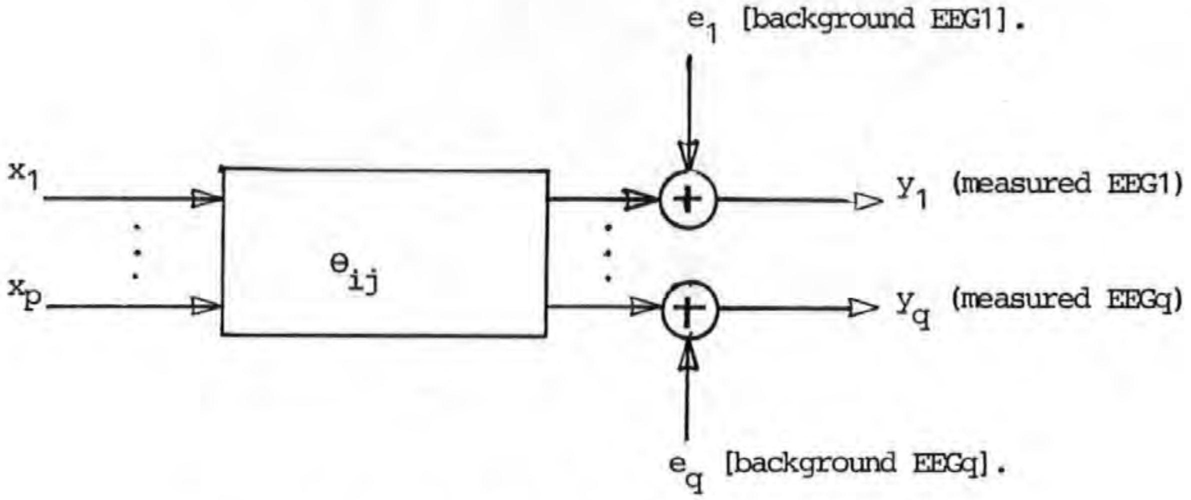


Figure 2.3 Block diagram representation of a Multichannel EEG Ocular Artefact System.

The system equation then becomes:

$$y_j(k) = \sum_{i=1}^{p} \theta_{ij} x_i(k) + e_j(k) \qquad \begin{matrix} j = 1,2, \ldots, q \\ k = 1,2, \ldots, m \end{matrix} \qquad (2.5)$$

For all m data points, these can be written more compactly as in (2.6). $\tilde{Y} = X_m \tilde{\theta}_m + \tilde{E}_m$ (2.6)

where,

$$\tilde{Y} = \begin{bmatrix} y_1(1) & y_2(1) & \ldots & y_q(1) \\ y_1(2) & y_2(2) & \ldots & y_q(2) \\ \vdots & \vdots & & \vdots \\ y_1(m) & y_2(m) & & y_q(m) \end{bmatrix} \qquad X_m = \begin{bmatrix} x_1(1) & x_2(1) & & x_p(1) \\ x_1(2) & x_2(2) & \ldots & x_p(2) \\ \vdots & \vdots & & \vdots \\ x_1(m) & x_2(m) & & x_p(m) \end{bmatrix}$$

$$\tilde{\theta}_m = \begin{bmatrix} \theta_{11} & \theta_{12} & \ldots & \theta_{1q} \\ \theta_{21} & \theta_{22} & & \theta_{2q} \\ \vdots & \vdots & & \vdots \\ \theta_{p1} & \theta_{p2} & & \theta_{pq} \end{bmatrix} \qquad \tilde{E}_m = \begin{bmatrix} e_1(1) & e_2(1) & \cdots & e_q(1) \\ e_1(2) & e_2(2) & & e_q(2) \\ \vdots & \vdots & & \vdots \\ e_1(m) & e_2(m) & & e_q(m) \end{bmatrix}$$

Noting that the form of (2.6) is the same as (2.2) and proceeding as before leads to the OLS estimate:

$$\hat{\tilde{\theta}}_m = [\ X_m^T X_m]^{-1} X_m^T \tilde{Y} \qquad (2.7a)$$

From the form of (2.7a),it is seen that each column of $\tilde{\hat{\theta}}_m$ has the same form as the expresion for the one EEG output considered earlier. Thus, for example, the first column of $\hat{\tilde{\theta}}$ is given by:

$$\hat{\theta}_{i1} = [\ X_m^T X_m\ ]^{-1} X_m^T \tilde{Y}_1 \qquad (2.7b)$$

where

$$\hat{\theta}_{i1} = \begin{bmatrix} \hat{\theta}_{11} \\ \hat{\theta}_{21} \\ \vdots \\ \hat{\theta}_{p1} \end{bmatrix} \quad , \quad \tilde{Y}_1 = \begin{bmatrix} y_1(1) \\ y_1(2) \\ \vdots \\ y_1(m) \end{bmatrix}$$

and the other matrices are as defined earlier.

The foregoing shows that the q output system can be treated as q individual single-output subsystems, and the overall system identifiable in q separate steps. However, it is seen from equations (2.7a) and (2.7b) that the subsystems have the factor $(X_m^T X_m)^{-1}$ in common, and this need be computed only once.

2.2.2 Some Statistical Properties of the OLS Estimates.

If the assumptions of the statistical properties of the error term are valid, the estimate $\hat{\theta}_m$ can be shown to have the following statistical properties [1,2] (more details are given in appendix A1): (i) $\hat{\theta}_m$ is unbiased and consistent. (ii) The error covariance of $\hat{\theta}_m$ is given by $C = \sigma^2 (X^T X)^{-1}$ (where σ^2 is the variance of the error term). The elements of C can be used to estimate the accuracy of the parameter estimates. The diagonal elements of the matrix C give the variances of $\hat{\theta}_m$ and the off-diagonal elements give the covariances between each pair of the elements of $\hat{\theta}_m$.

2.2.3 Models used in the Investigation

The form of the models used is given in equation (2.8)

$$EEG(i) = \theta_1 VL(i) + \theta_2 VR(i) + \theta_3 HL(i) + \theta_4 HR(i) + e(i) \quad (2.8)$$

$$i=1,2,...M$$

where EEG(i), e(i) are the ith samples of the measured and background or true EEG respectively. VL(i), HL(i) are the vertical and horizontal left EOGs respectively. VR(i), HR(i) are the corresponding right EOGs and M is the number of data points for each variable. θ_1 to θ_4, the correction coefficients, are the fractions of the corresponding EOGs that reach the EEG electrodes as artefact.

The variations of equation (2.8) which were used in the investigation are given in table 2.1

Model 2A in the table is an approximation of the model used by Quilter et al [7]. Model 2G simulates what appears to be a popular two-parameter model e.g [20,24]. It has the advantage of having reduced coupling between the vertical and the horizontal EOG signals, but combines the EOGs of both eyes which may be different [7]. Models

Table 2.1 Definition of models

model name	EOGs used in eqn 2.8	θs set to zero in eqn 2.8
2A	VL,HL	θ_2,θ_4
2B	VL,HR	θ_2,θ_3
2C	VR,HL	θ_1,θ_4
2D	VR,HR	θ_1,θ_3
2E	VL,VR	θ_3,θ_4
2F	HL,HR	θ_1,θ_4
2G	VL,(HL-HR)	θ_3,θ_4
3A	VL,HL,HR	θ_2
3B	VL,VR,HL	θ_4
3C	VL,VR,HR	θ_3
3D	VR,HL,HR	θ_1
4A	VL,VR,HL,HR	none
4B	VL,(VLxVL),HL,HR	"
4C	VL,HR,(VL-VR),(HL-HR)	"
4D	VL,HLxHR,HL,HR	"

3A and 4A were used by Jervis et al [5,6]. The latter is also an approximation of the model used by Fortgens and De Bruin [25]. Model 4B attempts to correct for possible non-linearities which may be present. Model 4C attempts to minimize the effects of dependencies between the EOGs (due to coupling and conjugate eye movement), while model 4D attempts to compensate for possible non-linear interraction between the horizontal EOGs. It has been suggested that the ocular dipoles tend to oppose each other in their effects during HEM [6]. Other models were included to ensure that all 'possible' models were compared.

Additionally, four single parameter models, each containing one of the four EOG signals , were used to correct for OA. The aim here was to show that the single parameter models may not always give satisfactory removal of OA.

2.3 Problems with Removing OA by the OLS Method

Although the OLS method is simple and efficient, it suffers from a number of problems. The three most serious problems are correlated errors, collinearity of the EOGs, and 'secondary' artefact in the EOGs themselves.

2.3.1 Autocorrelation in the Background EEG.

A crucial assumption in the least squares method is that the error terms, e(i), are uncorrelated. If this assumption is violated then the parameter estimates obtained by the OLS method are inefficient as they no longer have minimum variance, and tests of significance based on the sum of squares are no longer strictly valid [4,8-10].

(a) Testing for Autocorrelation

The Durbin-Watson statistic, d, [11,12] defined in (2.9), can be used to test for the existence of autocorrelation in the error terms. Since in general the error terms are not directly available, the test is applied to the residuals or the estimates of the error terms (see equation 1.2 and figure 2.2).

$$d = \frac{\sum_{i=2}^{M} [\ e(i) - e(i-1)\ ]^2}{\sum_{i=2}^{M} e^2(i)} \qquad (2.9)$$

d is close to zero when the residuals are highly positively correlated and to 2 when they are not. Tables giving the upper and lower bounds (dL,dU) for d for different percentage points are available for testing the significance of d [9]. If d is significant then some remedial action should be taken.

(b) <u>Correcting for Autocorrelation - The method of Generalized Least Squares</u>.

A significant value for d might be due to model inadequacy or autocorrelation in the errors. If the model is inadequate due to the omission of some important variables, or a linear model is specified where in fact the relationship beween the dependent and independent variables is nonlinear, then the residuals from such a model will be correlated [9]. Model inadequacy may be detected by examining plots of the residuals (or standard or normalized residuals [4]) against the fitted values or in this case estimates of the OAs, and plots of the residuals against the EOGs (including those that were omitted, if any) for systematic effects. Curvatures in any of the plots is indicative of nonlinearity [10], and this may be corrected by the use of a suitable model (e.g.

the use of a polynomial or the addition of cross-product terms to the model [10]). In the case of omitted variables, the plots would show some systematic effects [26]. The inclusion of the omitted variables in the model should eliminate the problem [10]. If the number of variables is not large, as in the OA problem, it is probably best to try out all the 'possible' models and then to examine their Durbin-Watson statistic.

However, if no important variables are omitted and the model is appropriate (ie a linear model has not been specified where the relationship is in fact nonlinear), then the error terms themselves are probably correlated. In this case the OLS method will always give correlated residuals and this can be corrected by transforming the data to yield uncorrelated residuals. A way of doing this is to represent the error term as an autoregressive (AR) series. Thus:

$$e(i) = \sum_{j=1}^{n} \Phi_j e(i-j) + a(i)$$

or

$$a(i) = e(i) - \sum_{j=1}^{n} \Phi_j e(i-j)$$

$$= [1 - \Phi(q^{-1})]e(i) \qquad (2.10)$$

where n is the order of the AR model, chosen such that the model adequately represents the error terms, Φ_j is the autoregressive parameters and a(i) is an uncorrelated sequence. q^{-1} is the shift operator and $\Phi(q^{-1})$ a linear nth order polynomial of q given by:

$$\Phi(q^{-1}) = \Phi_1 q^{-1} + \Phi_2 q^{-2} + \ldots + \Phi_n q^{-n}$$

Now from section 2.1 the OA model is given by:

$$y(i) = \sum_{k=1}^{p} \theta_k x_k(i) + e(i) \tag{2.11}$$

and from equation (2.10),

$$e(i) = \frac{a(i)}{1 - \Phi(q^{-1})}$$

On substituting e(i) into (2.11) and multiplying out equation (2.12) is obtained:

$$\begin{aligned}[1 - \Phi(q^{-1})]y(i) &= [1 - \Phi(q^{-1})] \cdot \sum_{k} \theta_k x_k(i) + a(i) \\ &= \sum_{K} \theta_K [1 - \Phi(q^{-1})] x_k(i) + a(i)\end{aligned} \tag{2.12}$$

Define the variables $\overline{x}_K(i)$ and $\overline{y}$ (i) as:

$$\begin{aligned}\overline{x}_k(i) &= [1 - \Phi(q^{-1})]x_k(i) = x_k(i) - \sum_{j} \Phi_j x_K(i-j) \\ \overline{y}(i) &= [1 - \Phi(q^{-1})]y(i) = y(i) - \sum_{j} \Phi_j y(i-j)\end{aligned} \tag{2.13}$$

Then equation (2.12) becomes:

$$\overline{y}(i) = \sum_{K=1}^{p} \theta_k \overline{x}_K(i) + a(i) \tag{2.14}$$

This model now has the same form as the original model of equation (2.11), but has an uncorrelated error term, a(i), and so will yield optimum estimates of θ_K ,which is the desired goal. However, it is clear from equation (2.14) (which is only a more compact form of equation (2.12)) that before θ_k can be estimated the autoregressive parameters, Φ_j, which are not known apriori, must be obtained. Thus, $\hat{\theta}_K$ cannot be obtained from the usual OLS approach, but may be obtained using a numerical procedure [2]. In this approach, a rough estimate of Φ_j is first obtained, so that $\bar{x}_K(i)$ and $\bar{y}(i)$ may be obtained (equation 2.13). Next $\hat{\theta}_K$ is obtained using (2.14). This value of $\hat{\theta}_K$ is then used to obtain another estimate of Φ_j and the process is then repeated. At each iteration, improved estimates of θ_K and Φ_j are obtained. The process is terminated when some convergence criterion is satisfied.

The procedure is given explicitly in the following algorithm [8,10]:

Step 1

Let $\Phi_j=0$ in equation (2.13), so that equation (2.14) reduces to the OLS equation of (2.11).

Step 2

Use OLS method to obtain an estimate of θ_K using (2.11), and hence an estimate of the error term as:

$$e(i) = y(i) - \sum_{K=1}^{p} \hat{\theta}_K x_K(i)$$

Step 3

Use OLS method to obtain an improved estimate of Φ_j from equation (2.10), that is:

$$\hat{e}(i) = \sum_{j=1}^{p} \Phi_j \hat{e}(i-j) + a(i)$$

Step 4

Obtain the set of variables $\bar{x}_K(i)$ and $\bar{y}$ (i) as in equation (2.13) using Φ_j from step 3.

Step 5

Use OLS method to obtain an improved estimate of θ_k from equation (2.14), viz:

$$\overline{y}(i) = \sum_{k=1}^{P} \theta_k \overline{x}_k(i) + a(i)$$

Use $\hat{\theta}_k$ to obtain an estimate of the error term, e(i), from equation (2.11) and then go back to step 3 and continue until some convergence criterion is satisfied.

For the first order or AR(1) model, Φ could be monitored and the operation terminated when there is no significant change in its value beween iterations. For the general case, that is AR(n), it is more convenient to use the residual variance. When the algorithm has converged, the last estimate of θ_k is then used in (2.11) to obtain estimates of the corrected EEG. If e(i) is uncorrelated then $\Phi_j=0$ and both the OLS and the iterative methods lead to the same results. It is to be noted that differencing (see section 1.6.1) is a special case of the first order AR model, obtained by setting both n and Φ_j to unity in (2.10).

Two objections to the above algorithm, commonly called the Generalized Least Squares (GLS) Algorithm, are that the convergence rate is too slow, and $\hat{\theta}_k$ may converge to false values if the signal-to-noise ratio is small [2]. Hsia [2] has given a modified GLS algorithm that is computationally more efficient.

In the investigation all three estimation methods, viz. OLS with raw and with differenced data, and iterative (i.e AR(1)) methods, were used

(c) Results of the Investigation on Autocorrelation

Several EEG records were corrected for OA with the models described in section 2.2.3 using the OLS with raw data method. In each case the statistic, d, was found to be very low for all ocular movements, suggesting that the residuals were highly correlated. As all the models were affected, including those that contained all the EOGs and those that contained cross-product terms, this suggested that the cause of the high correlation was not model inadequacy. The same records when corrected by the OLS with differenced data and iterative methods gave d statistics that were much closer to 2. For comparison, representative results for the OLS with raw and with differenced data methods are given in table 2.2.

A study of the residuals (or corrected EEG) and their autocorrelation functions (ACFs) for the OLS with raw data method suggested that the error terms,e(i), behaved like an autoregressive series [13,14]. The residuals behaved like 'random walks' and their ACFs decayed slowly to zero. Figure 2.4(i) and 2.4 (ii) give typical examples for a HEM record for the OLS with raw data method. Figure 2.4(iii) to 2.4(vi) give similar plots but for the OLS with differenced data and iterative methods, and these show that all systematic effects have been removed and the ACFs decreased rapidly to zero. This was true for all types of ocular movements and was taken as an indication that the

Table 2.2

Comparison of the Durbin-watson statistics for the OLS with raw and with differenced data methods using various models

model	OLS(Raw)	OLS(differenced)
1A	0.35	1.59
2A	0.35	1.60
2B	0.34	1.52
2C	0.34	1.58
2D	0.37	1.51
2E	0.34	1.59
2F	0.33	1.51
2G	0.37	1.51
3A	0.36	1.51
3B	0.34	1.59
3C	0.38	1.51
3D	0.39	1.49
4A	0.39	1.49
4B	0.36	1.51
4C	0.39	1.49
4D	0.36	1.49

autoregressive representation for e(i) was reasonable.

2.3.2 Collinearity of the EOGs

If the EOGs are linearly dependent on each other they are said to be collinear. Strong collinearity can lead to difficulties in computation and imprecise parameter estimates [9,10,16,17] (see appendix A2).

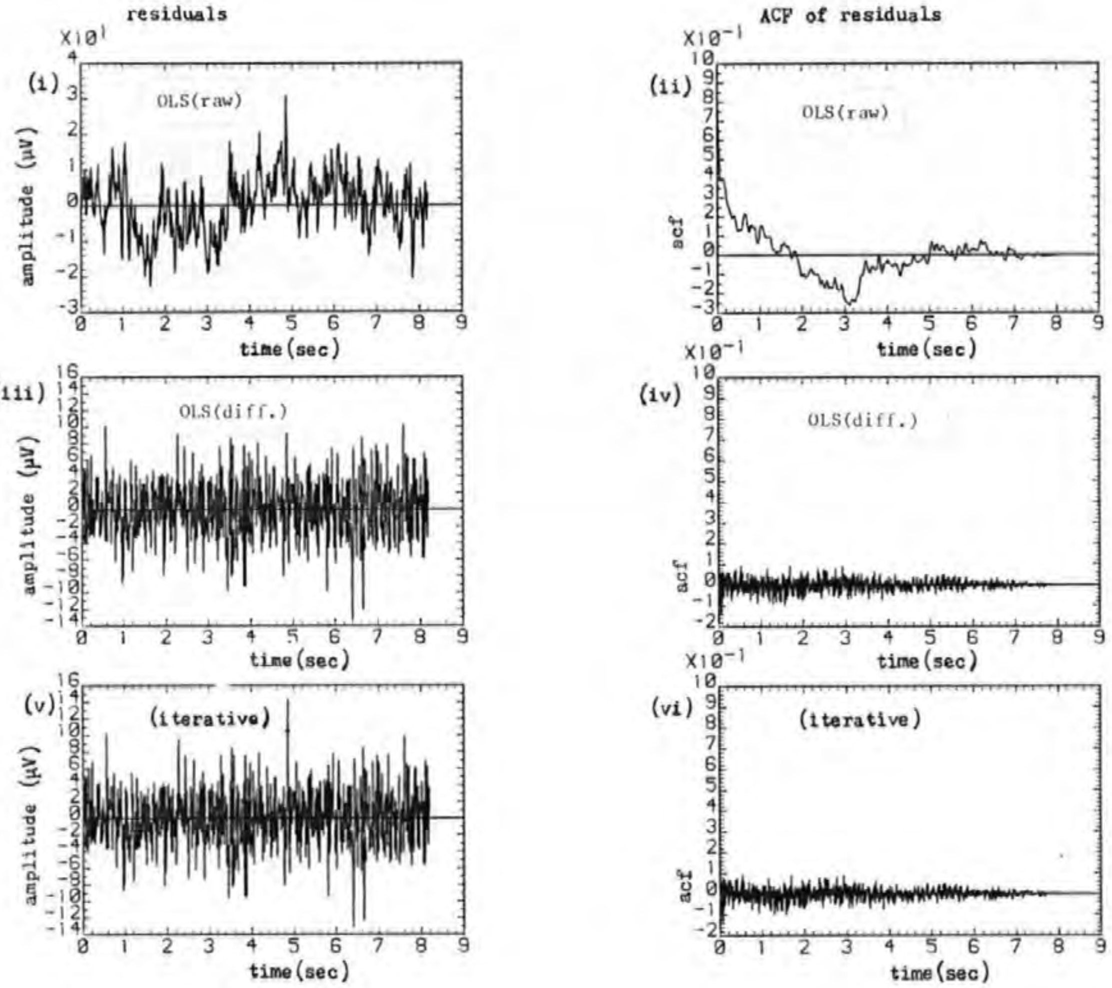


Figure 2.4 Comparison of the residuals and their autocorrelation functions (ACF) for a horizontal eye movement record (JSS48) for the three estimation methods. (i), (iii) and (v) residuals for the OLS with raw data, OLS with differenced data, and iterative methods, respectively. (ii), (iv) and (vi) the corresponding ACFs.

To assess the extent of linear dependence between the EOGs their correlation matrices and the associated eigenvalues were examined. Strong collinearity is accompanied by one or more very small eigenvalues [10,16,18]. Table 2.3 gives a representative selection of the correlation matrices and their eigenvalues for different ocular movements. Examination of the table showed that in each case there was high correlation between some of the EOGs, and at least two of the eigenvalues were very small. In fact, only the first two eigenvalues in each case accounted for nearly 99 % variability [$(\lambda_1+\lambda_2)$ x 100 %/ $\sum_{i=1}^{4} \lambda_i$] suggesting strong collinearity and that about two EOGs were redundant.

For the differenced data, the correlation coefficients were much smaller than for the raw data and no eigenvalue was too small. For example, after differencing the eigenvalues for the record given in table 2.3a were

$\lambda 1= 2.630, \lambda 2=0.647, \lambda 3=0.561, \lambda 4=0.162$. This suggested that differencing could be used to reduce the effects of collinearity.

For all types of ocular movement except lateral eye movement, the vertical EOGs were more strongly correlated than the horizontal EOGs. The correlation coefficients for the differenced data were typically 0.6 for the horizontal EOGs for HEM and 0.75 for the vertical EOGs for blink or VEM. This lends quantitative support to the suggestion that the vertical EOGs during blinks and VEM are more strongly correlated than the horizontal EOGs during HEM [6,7].

Table 2.3

Correlation matrices and their eigenvalues for the EOGs for (a) a vertical eye movement record (JSS36) (b) a blink record (AJS108) (c) a horizontal eye movement record (AJS36) (d) a diagonal eye movement record (AJS54).

	V_L	V_R	H_L	H_R	V_L	V_R	H_L	H_R
V_L	1.000				1.000			
V_R	0.990	1.000			0.987	1.000		
H_L	0.917	0.926	1.000		0.956	0.960	1.000	
H_R	0.894	-.907	0.978	1.000	0.942	0.943	0.943	1.000
Eigenvalues:	$\lambda 1=3.806$, $\lambda 2=0.164$ $\lambda 3=0.020$, $\lambda 4=0.010$				Eigenvalues:	$\lambda 1=3.876$, $\lambda 2=0.078$ $\lambda 3=0.033$, $\lambda 4=0.013$		
(a)	$\sum_i \frac{1}{\lambda_i}=156.36$				(b)	$\sum_i \frac{1}{\lambda_i}=120.30$		

	V_L	V_R	H_L	H_R	V_L	V_R	H_L	H_R
V_L	1.000				1.000			
V_R	0.853	1.000			0.994	1.000		
H_L	-0.662	-0.258	1.000		0.995	1.000	1.000	
H_R	0.799	0.430	-0.957	1.000	-0.761	-0.733	-0.849	1.000
Eigenvalues:	$\lambda 1=3.012$, $\lambda 2=0.933$ $\lambda 3=0.037$, $\lambda 4=0.018$				Eigenvalues:	$\lambda 1=3.628$, $\lambda 2=0.330$ $\lambda 3=0.036$, $\lambda 4=0.006$		
(c)	$\sum_i \frac{1}{\lambda_i}=83.99$				(d)	$\sum_i \frac{1}{\lambda_i}=197.75$		

Scatter plots of the EOGs against one another showed linear relationship between the EOGs, especially between the two vertical EOGs and between the horizontal EOGs (see for example figure 2.5). Although in nearly every case the scatter diagrams for the vertical and horizontal EOGs of any eye showed some curvature, suggesting a nonlinear relationship, the linear relationship was stronger.

vertical left eog (μV)

```
  400.+
      -
      -                                                *   *2
      -                                              **   2*
  300.+                                          *     ***
      -                                          2323* *
      -                                      35*23 **
      -                                   2254 352* *
      -                                 *4 422*3 *    *
  200.+                               5*222 **    2*
      -                             4324 *2* *2
      -                          *24* ** 2
      -                         342*2   * ** *    *
      -                       42 2****    *  *
  100.+                     2534* ** **
      -                   34+**  *    *
      -                 2*23*  2    *   *
      -               23745* 2 **      *
      -            *744**2
    0.+           9952****   *
      -        +++732*
      -     *++++23
      -    ++++
      -    +4
 -100.+
        +----------+----------+----------+----------+----------:
     -100.         0.        100.       200.       300.     . 400

                              vertical right eog (μV)
```

horizontal left eog (μV)

```
  100.+
      -                                  22
      -                                  3*2 2332**
      -                                *3223 3*2
      -                              3*56636* *
   65.+                         *   325655 *
      -                       *  *245224*2  **
      -                        *3*2223222  ** **
      -                      ***232*22*   * *   *
      -                  *  3 6332433** *
   30.+                   223*3 322
      -            *    * 254432
      -              2*3763*
      -         *  22*49**2
      -         *24955647
   -5.+        3+++++83*
      -   22557+++88**
      -  2+++++++++62*
      - 5+++++++++5**
      - *424++422*
  -40.+
        +----------+----------+----------+----------+----------+
      -40.        -5.        30.        65.       100.       135.
```

Figure 2.5 horizontal right eog (μV)

Scatter diagrams of the EOGs against themselves for a vertical eye movement record. The symbol * indicates a single point, the numbers give counts of points that fall on the same spot and the symbol + is used where count exceeds 9.

2.3.3 EOG Artefacts

As stated earlier, any artefact contained in the EOG such as EMG or EEG, would be introduced into the corrected EEG. To assess the problem, several EEG waveforms for both the raw and differenced data were studied visually and were found to be always contaminated by 'noise'. It was found that for VEM and blink records the EOG level relative to noise (loosely called signal-to-noise ratio here) was smaller for the horizontal channels than for the vertical channels. The opposite was true for HEM. A VEM record showing EOG artefacts is given in figure 2.6

The problem of EOG artefact will be re-examined in chapter 3 where an efficient digital filter used to reduce its effect is also discussed.

2.4 Comparison of Models for Removing Ocular Artefacts

It is important to determine whether a model adequately describes the EOG-OA relationships. In particular, which and how many EOG signals should be used in an OA model . Some workers suggest that the best removal can only be achieved by using all four EOGs [5,6,25], others report satisfactory correction with one or two EOG signals [7,19,20,25]. Thus there remained a need to carry out a systematic comparison of the various models to establish their relative effectiveness in removing OA from the EEG. In this section, the effectiveness of the models described

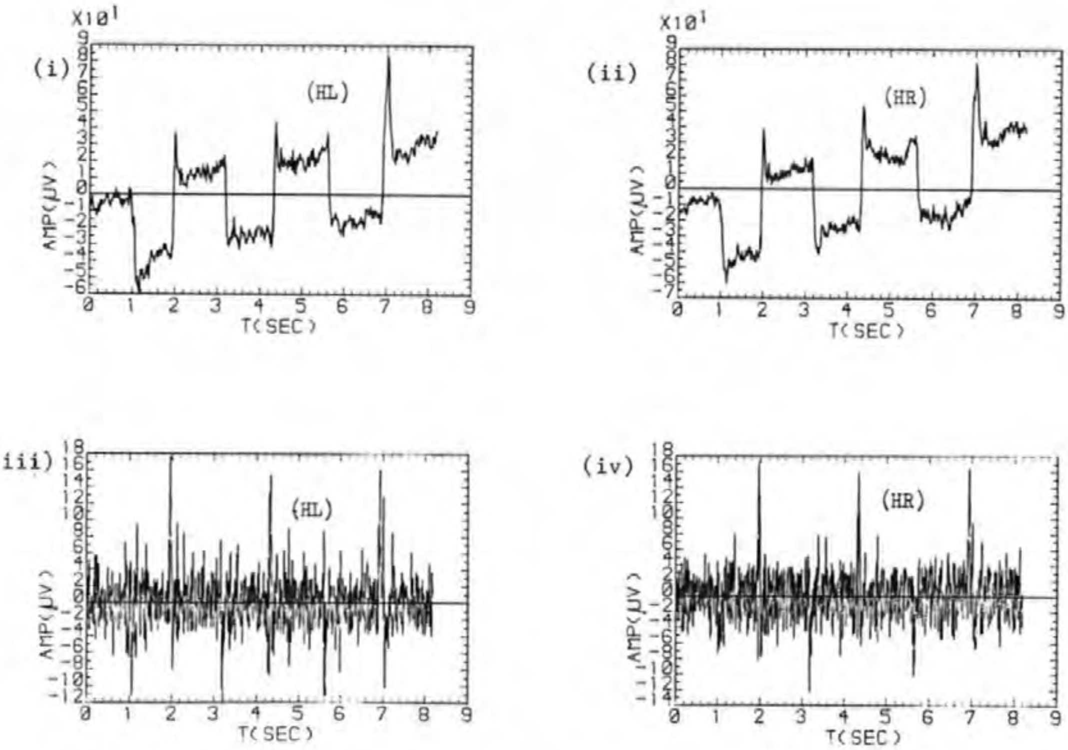


Figure 2.6 EOGs and differenced EOGs showing secondary artefacts (i),(ii) measured EOG waveforms; (iii),(iv) the corresponding differenced waveforms.

in section 2.2.3 are compared, and the work of Jervis et al [5,6] extended.

2.4.1 Criteria Used to Compare the Performance of Models

Quantitative and pictorial criteria were used in the comparison of models. They were intended to be complementary.

(a) Minimization of the Sum of Squares of the EEG.

As discussed in section 2.2, each model minimizes the sum of squares of the error terms. Thus minimum sum of squares of the estimates of the error terms (or residual sum of squares) can be used as a criterion for comparing the models. To take into account the decrease of the residual sum of squares with the number of parameters in a model, the residual variance given by equation 2.15 [18] was used instead.

$$S_p^2 = R_{ssp}/(M-p-1) \qquad (2.15)$$

where R_{ssp} and p are respectively, the residual sum of squares and the number of parameters in the model. Models with small residual variances are considered good models [18,21,22].

(b) Mallows' Statistic, C_p.

A model is a subset of a larger model if the latter contains all the variables in the former plus one or more variables. Subset models yield biased estimates except when the variables omitted are unimportant. Thus a subset model is to be preferred if the bias of its estimates is small. A criterion that can be used to choose between models is Mallow's statistics, C_p [18,21,22]:

$$C_p = (R_{ssp}/S^2{}_k) + 2p - M \qquad (2.16)$$

where p is the number of parameters in the model, S_K^2, is the residual variance from the full model, here taken to be model 4A. A good model should have a C_p value close to p, which in this case should be no more than about 4 as no model has more than four parameters. If models have C_p values that differ by only small amounts, they may have comparable performance. Therefore, a somewhat more relaxed criterion was used to ensure that all 'good' models were included. Thus a model was 'good' if its C_p value was about ten or less. After studying several records a pattern emerged and the 'good' models were then subjected to further analysis.

(c) OA Estimates and their Differences as Measures of Performance

A disadvantage of quantitative tests is that they may be influenced by large secondary artefacts in the EOGs. Thus a model that uses noisy EOGs may be favoured by the residual variance criterion, for example. Also, these tests do not give a good indication of the relative levels of remnant artefact. However, they can be used as a guide. Further discussions of the problems of quantitative tests in the OA work are given in appendix A3.

A complementary method used to study the performance of the models was to compare plots of OA estimates from the various models and also the differences between these estimates. The difference between the OA estimates of any two models is equivalent to the difference between the corresponding corrected EEGs. This approach may be described as a pictorial form of 'F-Test' and it enabled the effects of adding extra terms in the model to be judged visually. To relate the differences to EM, plots of the autocorrelation functions (ACFs) of the differences may also be obtained.

2.4.2 Results of Model Comparison

(a) Single Parameter Models

The four single parameter models described in section 2.2.3 were used to correct several EEG records for OA, and in many cases the corrected EEGs still contained a vestige of the EOGs. It was found that for blinks and vertical eye movements, the models that used vertical EOGs performed better than those using horizontal EOGs, and vice versa for lateral eye movements. Figure 2.7 gives a typical result for two of these models. (Similar results were obtained for the other two models). It is evident from the systematic effects in the corrected EEGs and their ACFs that the OA has not been satisfactorily corrected for by one of the models.

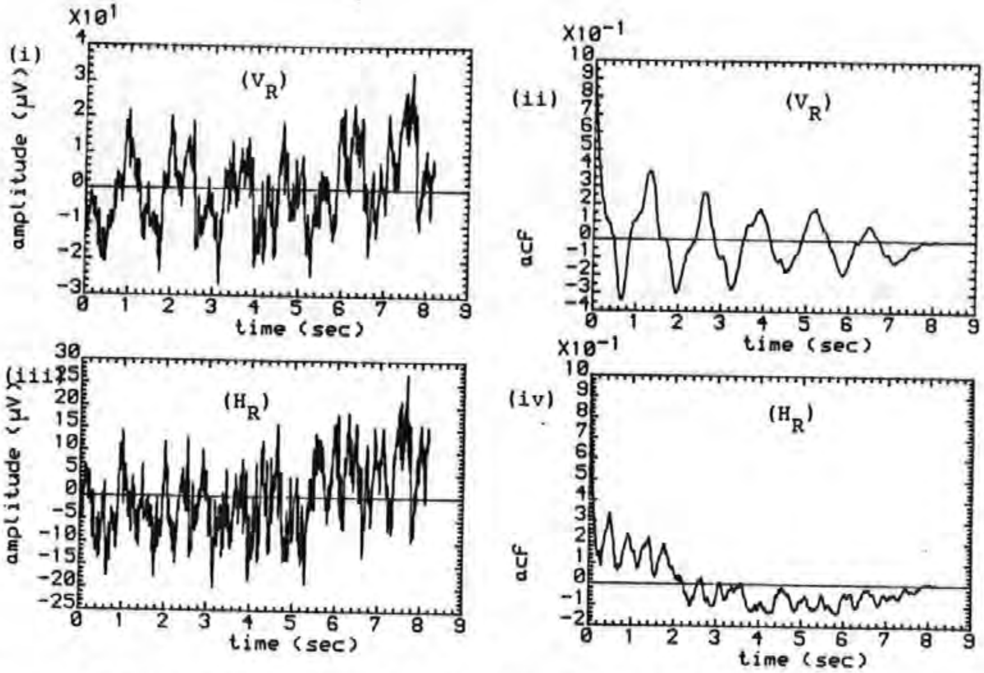


Figure 2.7 Corrected EEGs and their ACFs for a HEM record for one parameter models. (i),(iii) EEGs for models using vertical right EOG and horizontal right EOG, respectively. (ii), (iv) the corresponding ACFs.

Scatter plots of the EEG against the EOGs are shown in figure 2.8, from which it is seen that there is only a weak relationship between the vertical right EOG and the EEG, and this explains its failure in removing the OA (see figure 2.7 (i)). The bunching of points at about ± 60 µV in figure 2.8 (ii) correspond to when the eyes were at rest or fixated (see section 1.6). Examination of these and similar scatter plots for other types of ocular movements showed that there was sometimes a linear relationship between the EOGs and the OA but that this was not true in all cases.

The mean of the absolute values of the parameter estimates and their standard deviations for two single parameter models (using VR and HR respectively) are given in table 2.4 for two subjects. The values of the standard deviations indicated that for VEM, blink and DEM, the single parameter model that used VR gave more reliable estimates than the model that used HR. The opposite was true for lateral eye movement. It is seen that the estimates for the OLS with raw and with differenced data methods differed, but only by small amounts if the standard deviations are taken into account. Similar results were obtained for models using VL and HL, respectively.

(b) Quantitative Comparison of Models

Several EEG records were corrected for OA with the multiparameter models defined in section 2.2.3 using the OLS with raw and with differenced data methods. In each case and for each model the statistics S_p and Cp were computed. Table 2.5 gives typical results for the methods. For the OLS with raw data method, it is seen from the table that only models 2D,3C,3D and 4C satisfied the Cp criterion of about ten or less, whereas for the OLS with differenced data method only models 2D,3C,3D,4C and 4D satisfied the criterion and in this instance are the 'good' models. As the d statistics for the OLS with differenced data method were significantly low (see sec. 2.3.1), the results for the OLS with differenced data method were taken to be more reliable. Similar conclusions can be drawn from the

```
 45.+eeg(µV)
    -
    -
    -
    -       horizontal         *
 30.+       eye movement    *
    -                       *
    -                     **2 *        **  *3 * 2   *
    -                    243 *     2  222  *     *4 *  *
    -                   *36 *      * 3533 2**2*   22 *
 15.+                   245*    * 4*2454* 226*      **
    -                   44 **    24*427*25524222 2* *3*
    -                  *  **  *2832*2526342442 **   **
    -             *    *3***2 *52*22 38735+6722 2 2*    *
    -              3 *3* 23 3**55 254776627962**2  *  * *
  0.+           3 33  **322***4453 698676536825* * *
    -           2 2*2* 57*3*3645 4*377442245943 *2*
    -          *3*       *23*23 4*4+57636456623*4 *
    -     *  *  *  2*2  ** 2*22*226778+53263*5** **
    -        * ** 3*    *2325*2*3237+6+* *422*
-15.+           2       *4*** 2*3358*322*3*2
    -          *         2**232*25 *42* * 4**
    -                  * 2* *2**6 *
    -                       *  *** *
    -                          2
-30.+
      +---------+---------+---------+---------+---------+
    -60.      -40.      -20.       0.       20.       40.
                         vertical right eog (µV)
```

```
 45.+eeg(µV)
    -
    -
    -        horizontal
    -        eye movement        *
 30.+                            *
    -                            *
    -                           *4 2222  2
    -                           65 34223*
    -                         **7428522*5
 15.+                         * 662+42*483
    -                      * ** 45259+94+
    -                           *335*++5+*
    -     *2 *          **  **  2*+56++7+***
    -     82652* 3 *** * **   ** 2385+783922*
  0.+    365+45558***   *  *2* 2*3 688+633* *
    -    4+2+5553+22 *      2  * 5* 5+743 *
    -    *4385+9++262*22 *   * *  **62 2* 3
    -   **2*677+5++4* *2**22 *  *   2     *
    -   2 26492+++7*3*  *
-15.+      **25865457***
    -      **46456223  *
    -        4262  **
    -        * 2*  *
    -         * *
-30.+
      +---------+---------+---------+---------+---------+
   -120.      -60.       0.       60.     120.      180.

                         horizontal right eog (µV)
```

Figure 2.8 Scatter diagrams of the EEG and EOGs for a horizontal eye movement record (AJS36). The symbol * on the diagrams indicates a single point, the numbers give counts of points that fall on the same spot and the symbol + is used where these exceed 9.

Table 2.4

Mean parameter estimates ($\hat{\theta}$) and their standard deviations (SD) for records corrected for OA with single parameter models that used VR and HR for the OLS with raw and with differenced data method

Subject		Vertical EM				Blink			
		OLS(raw)		OLS(diff)		OLS(raw)		OLS(diff)	
		(V_R)	(H_R)	(V_R)	(H_R)	(V_R)	(H_R)	(V_R)	(H_R)
	$\hat{\theta}$	0.16	0.52	0.14	0.33	0.07	0.23	0.09	0.31
AJS	SD	0.03	0.12	0.01	0.06	0.01	0.06	0.01	0.06
	$\hat{\theta}$	0.18	0.76	0.15	0.34	0.10	0.35	0.11	0.40
JSS	SD	0.04	0.38	0.02	0.08	0.01	0.21	0.01	0.12

Subject		Horizontal EM				Diagonal EM			
		OLS(raw)		OLS(diff)		OLS(raw)		OLS(diff)	
		(V_R)	(H_R)	(V_R)	(H_R)	(V_R)	(H_R)	(V_R)	(H_R)
	$\hat{\theta}$	0.20	0.13	0.28	0.16	0.16	0.24	0.15	0.18
AJS	SD	0.14	0.022	0.1	0.037	0.07	0.06	0.05	0.11
	$\hat{\theta}$	0.38	0.13	0.22	0.22	0.18	0.29	0.18	0.36
JSS	SD	0.07	0.04	0.08	0.12	0.08	0.10	0.11	0.24

values of the variance. However, Cp values are easier to interpret, but it is necessary to inspect the variance to avoid making wrong deductions.

Table 2.6 gives a summary of the results based on the Cp values for the OLS with raw and with differenced data methods for the various ocular movements and for two subjects. From the table it is seen that the following models performed well in the OLS with raw and with differenced data methods, respectively: (3D,4C and 4D) and (3D,4C). It is to be noted that these models contained (HL,HR) and one or the other but not both

Table 2.5
Comparison of models using the test statistics S_p and C_p.

Model	OLS(raw)		OLS(differenced)	
	S_p	C_p	S_p	C_p
1A	47.49	67.15	16.45	166.33
2A	45.58	23.33	15.47	95.53
2B	47.50	68.32	14.45	22.96
2C	45.56	23.81	15.48	97.21
2D	44.97	10.30	14.30	11.94
2E	45.55	23.62	15.48	97.47
2F	48.95	101.62	14.43	21.04
2G	47.03	57.49	15.46	95.90
3A	46,78	52.82	14.41	20.97
3B	45.55	24.77	15.50	99.20
3C	44.86	8.90	14.27	10.61
3D	44.57	2.34	14.16	2.56
4A	44.56	*	14.15	*
4B	46.78	53.80	14.42	22.87
4C	44.56	3.00	14.15	3.00
4D	46.74	52.85	14.02	-6.18

* not applicable

vertical EOGs, except model 4C which contained VR indirectly. It appears that there is no need for more than one vertical EOG signal in a model. It is to be noted that models containing the EOGs of the right eye performed better than those containing the EOGs of the left eye. (compare for example, models 2A and 2D or 3A and 3D).

The iterative method was applied to some of the 'best models'. It was found that convergence of the parameter estimates was obtained after about two to three

Table 2.6

Summary of results based on the Cp criterion: the table shows the number of times each model had Cp values of about ten or less for each ocular movement and for two subjects (AJS and JSS).

table 2.6(a): OLS(Raw)

MODEL	VEM		Blink		HEM		DEM		
	AJS	JSS	AJS	JSS	AJS	JSS	AJS	JSS	TOTAL
2A	0	0	0	0	0	1	1	2	4
2B	0	0	0	0	0	0	2	2	4
2C	0	0	0	1	0	0	4	1	6
2D	0	0	2	0	2	0	2	3	9
2E	0	0	0	0	0	0	3	0	3
2F	0	0	0	1	0	1	1	0	3
2G	1	0	0	0	0	0	1	1	3
3A	1	2	0	3	0	3	4	5	16
3B	2	2	1	2	1	0	4	1	16
3C	1	0	2	1	1	0	4	5	14
3D	1	2	3	3	3	0	5	4	21
4B	1	1	0	0	0	4	4	5	15
4C	5	5	3	5	3	4	8	7	40
4D	2	3	3	3	3	3	4	5	26

table 2.6(b): OLS(differenced)

MODEL	VEM		Blink		HEM		DEM		
	AJS	JSS	AJS	JSS	AJS	JSS	AJS	JSS	TOTAL
2A	0	0	0	0	0	0	0	0	0
2B	0	0	0	0	0	1	0	0	1
2C	0	0	0	0	0	0	2	1	3
2D	5	0	3	0	2	1	3	1	15
2E	0	0	0	0	0	0	1	0	1
2F	0	0	1	1	0	0	0	2	4
2G	0	1	0	0	0	0	1	1	3
3A	0	0	1	1	0	1	1	3	7
3B	0	0	0	0	1	1	2	2	6
3C	3	0	2	0	3	1	3	1	13
3D	5	5	3	4	3	4	8	4	36
4B	0	1	0	0	0	1	1	3	6
4C	5	5	3	5	3	4	8	7	40
4D	0	0	2	1	0	2	2	3	10

iterations. Results very similar to those of the differencing method were obtained. For comparison, results for model 3D using the three estimation methods are given in table 2.7. Inspection of the table shows that the results for the OLS with differenced data and iterative methods were in general similar, but differed from those for the OLS with raw data method. Compare for example, the values for $\hat{\theta}3$ in table 2.7(c) for the three methods. This is to be expected as Φ was nearly the same for the OLS with differenced data and iterative methods so that both methods were nearly equivalent.

Table 2.7

Comparison of parameter estimates,Durbin-Watson statistics and the autoregressive parameter Φ for the OLS with raw and with differenced data, and the iterative methods using model 3D
(a) VEM record (JSS36), (b) HEM record (AJS36)
(c) Blink record (AJS114),(d) VEM record (AJS0).

	Parameter Estimates			T-ratio				
	$\hat{\theta}_1$	$\hat{\theta}_2$	$\hat{\theta}_3$	$T\theta_1$	$T\theta_2$	$T\theta_3$	d	Φ
OLS(raw)	0.121	-0.231	0.609	19.36	-4.47	14.61	0.17	0
OLS(diff)	0.129	-0.181	0.331	9.69	-5.19	9.17	1.81	1
iter.	0.140	-0.165	0.355	11.47	-4.74	9.94	1.75	0.935
(a)								
OLS(raw)	0.091	-0.177	-0.108	4.69	-15.41	-7.18	0.27	0
OLS(diff)	0.330	-0.044	0.102	9.99	-2.46	4.47	1.65	1
iter.	0.306	-0.046	0.077	9.39	-2.64	3.50	1.60	0.942
(b)								
OLS(raw)	0.107	0.094	-0.174	10.07	3.16	-4.85	0.39	0
OLS(diff)	0.0515	-0.104	0.304	4.52	-3.37	9.83	1.49	1
iter.	0.047	-0.105	0.274	4.24	3.46	8.79	1.41	0.866
(c)								
OLS(raw)	0.168	-0.115	0.0664	27.68	-3.56	2.43	0.22	0
OLS(diff)	0.129	-0.011	0.105	15.20	-0.44	3.94	1.59	1
iter.	0.132	-0.017	0.097	17.02	-0.66	3.70	1.52	0.899
(d)								

$T\theta_j = \frac{\hat{\theta}_j}{SE}$ where SE is the standard error for $\hat{\theta}_j$, j=1,2,3.

(c) Pictorial Comparison of Models.

Figures 2.9 and 2.10 show, respectively, the OA estimates obtained using a number of models and the differences between these estimates for a horizontal eye movement record corrected by the OLS with raw data method. Figure 2.9 showed that all the oA estimates were not the same in amplitude and shapes, suggesting that the models corrected the EEG to different extents. It was also found that the OA for models 4A and 4C, in general, contained much higher level of noise than any other model. This was probably because the EOGs used in these models contained more secondary artefacts. Referring to figure 2.10, it may be noted that all but figure 2.10(iv) and 2.10(v) contained EOG related signal in the form of rectangular wave and only little noise. Figure 2.10(iv) and (v), associated with models that use all four EOGs (models 4A and 4C), contained significant levels of noise. This suggested that hardly any improvement was achieved by using all four EOGs in a model. Figure 2.10(vi) shows that only small improvement was achieved by including square terms (in this case, HLxHR) in the model. These results hold for lateral eye movement. For vertical eye movement and blink, it was found that models using more than two EOGs did not yield significant improvements, especially if one of the EOGs already included in the model was a vertical EOG.

The corrected EEGs corresponding to the OA estimates given in figure 2.9 are shown in figure 2.11. Some vestige of the EOG can be seen in these waveforms particularly in

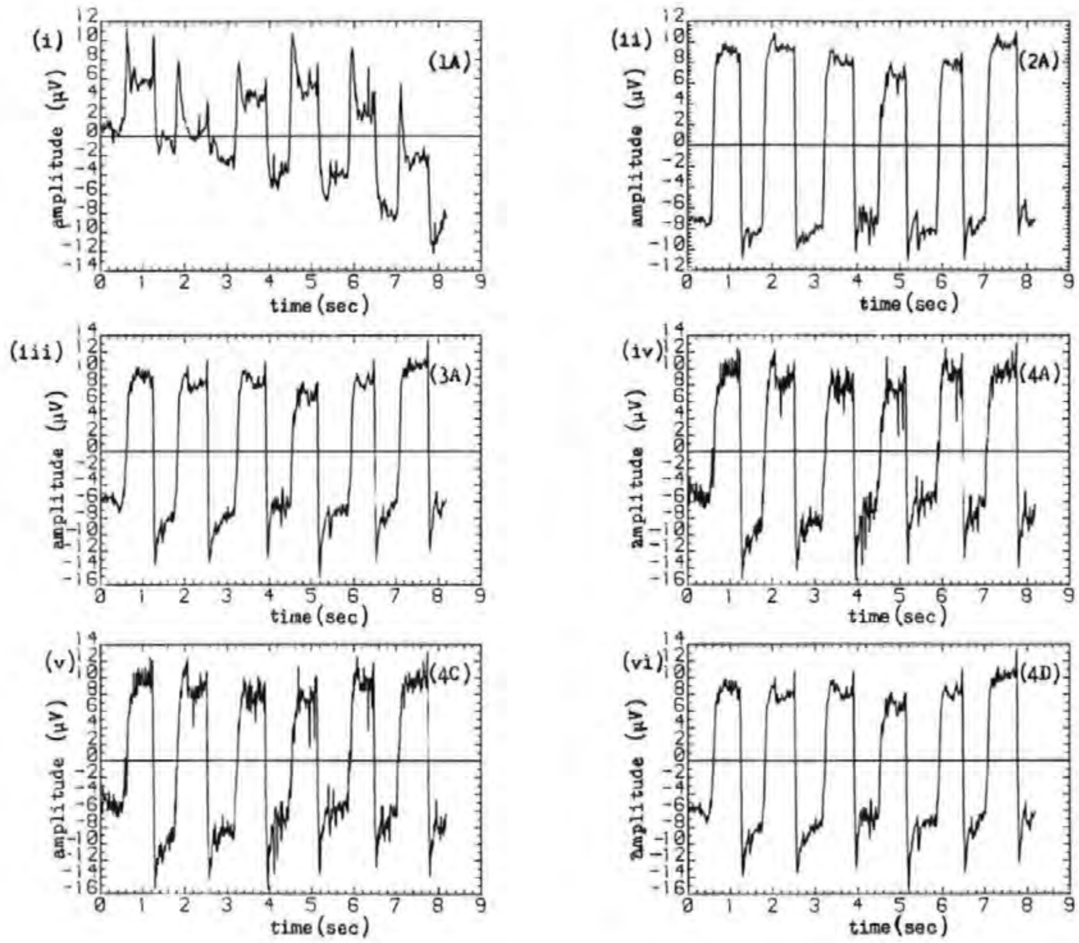


Figure 2.9 Estimates of the ocular artefacts for a HEM record (AJS36) using (i) model 1A (ii) model 2A (iii) model 3A (iv) model 4A (v) model 4C (vi) model 4D.

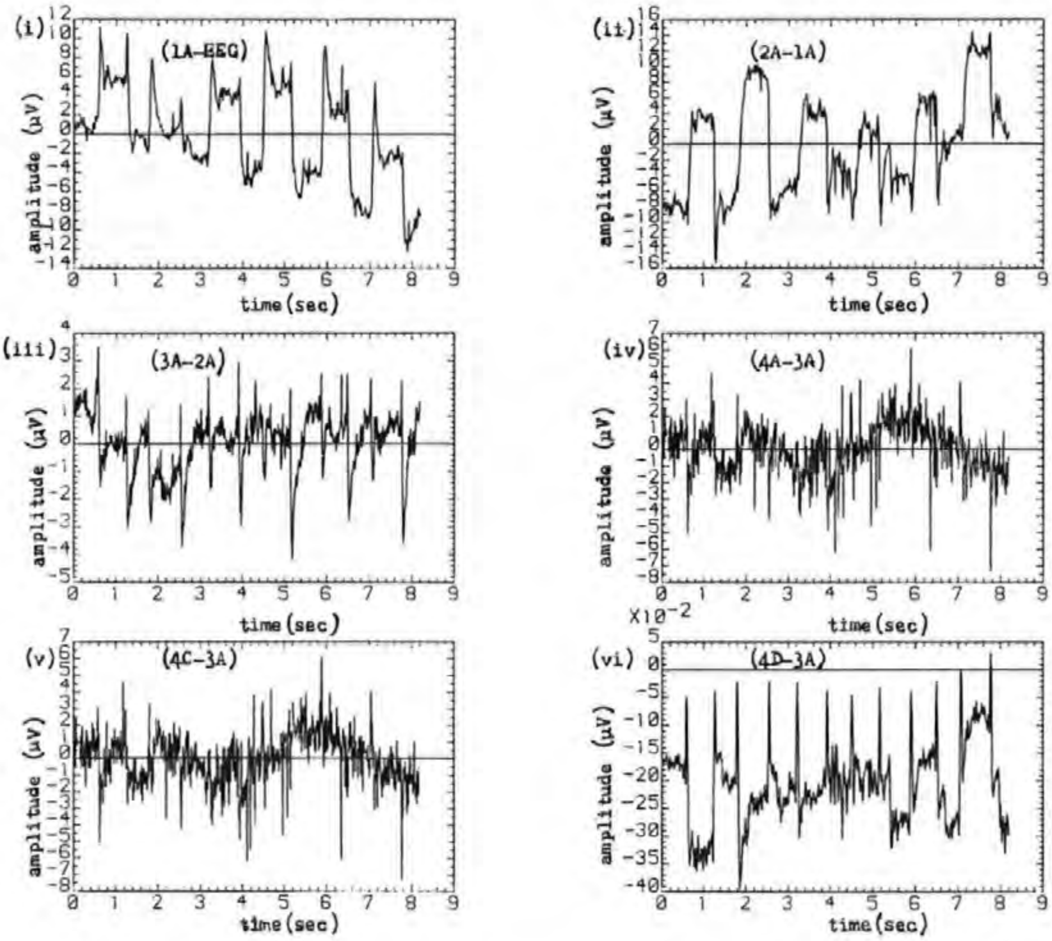


Figure 2.10 Differences between the (i) raw EEG and the corrected EEG using model 1A (ii) OA estimates of models 2A and 1A (iii) OA estimates of models 3A and 2A)iv) OA estimates of models 4A and 3A (v) OA estimates of models 4C and 3A (vi) OA estimates of models 4D and 3A.

that of the single parameter model , figure 2.11(i). It is note worthy that the differences between these EEGs are not as easily seen as they are from figure 2.10, illustrating the sensitivity of the pictorial criterion. A common feature in all these models was that rider artefact, when it occurred, was hardly ever completely removed. Peaks were often observed in the corrected EEG at points corresponding to where the rider artefacts occurred in the raw data.

Similar results to those given above were obtained for the OLS with differenced data and iterative methods. In these cases, however, the improvements were more pronounced. However, model 4D did not perform as well as in the OLS method. Figure 2.12 gives the OA estimates for model 3D for the three methods and the differences between these estimates. The figure showed that the OLS with differenced data and the iterative methods differed only in the size of their OA estimates, whereas the OLS with raw data method gave results that were dissimilar to either methods. Judging from the signs of the peaks in the OA differences, it appears that the OLS with raw data method tended to under estimate the OA.

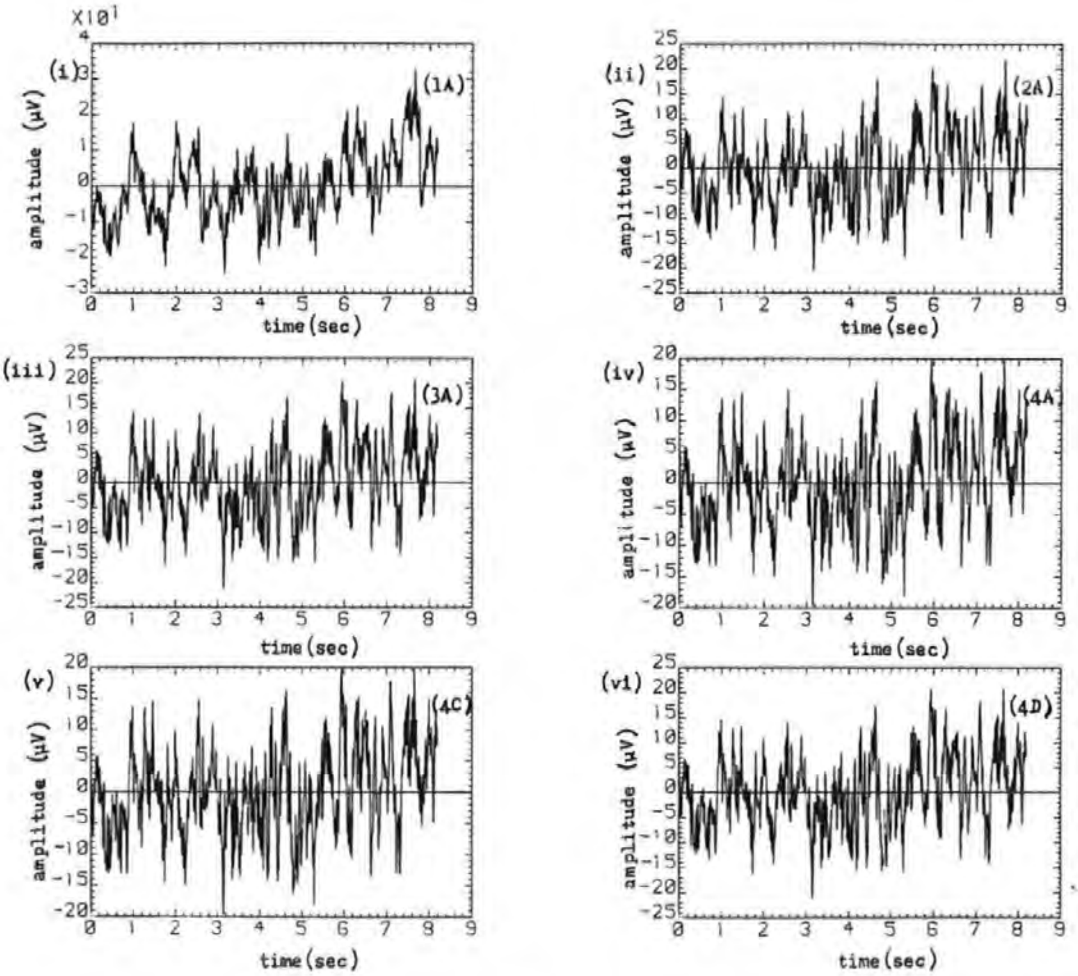


Figure 2.11 Waveforms of the corrected EEG corresponding to the OA estimates of figure 2.9

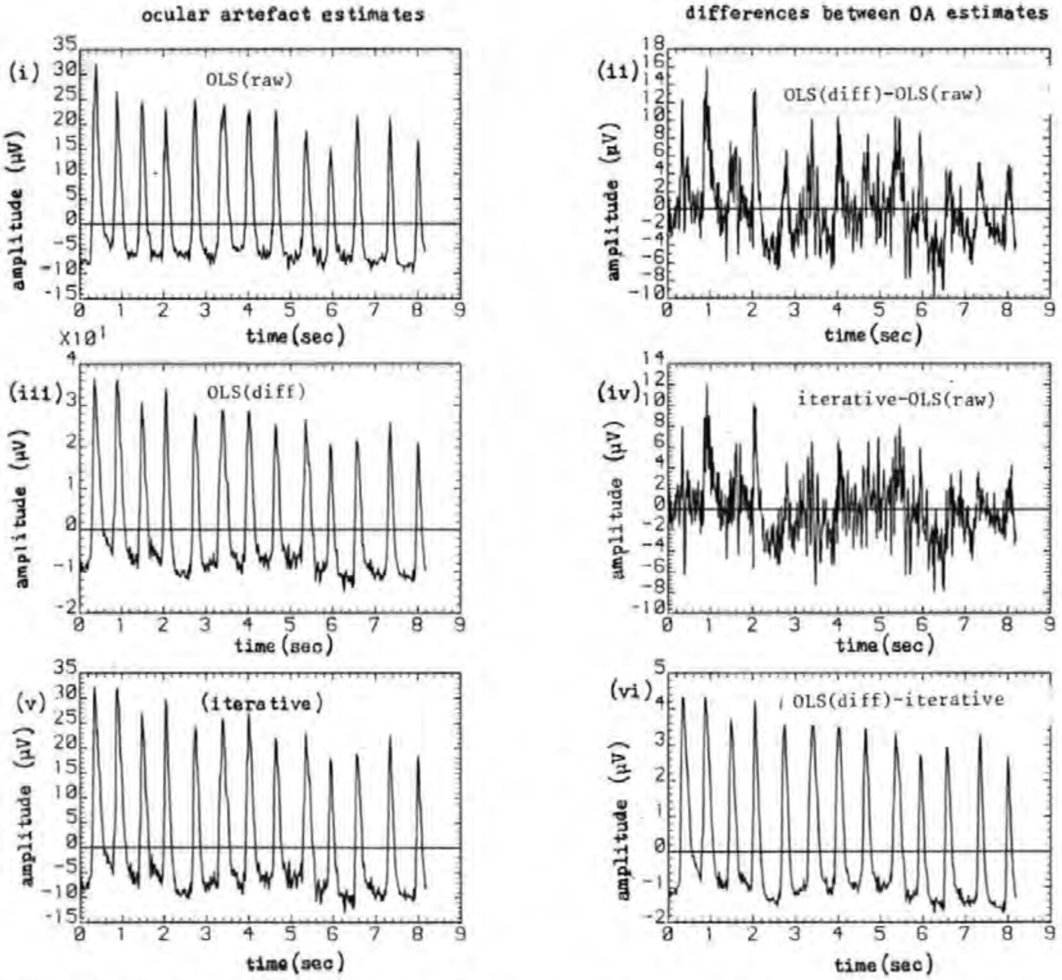


Figure 2.12 Comparison of the OA estimates for a blink record (AJS114) using the three estimation methods with model 3D. (i), (iii) and (v) OA estimates for the OLS with raw and with differenced data, and iterative methods. (ii), (iv) and (vi) Differences between the OA estimates for the OLS with raw and with differenced data methods, OA estimates for the OLS with raw data and iterative methods, and the OA estimates for the OLS with differenced data and iterative methods.

2.5 Discussion of Results

(a) Autocorrelation in the Background EEG

The results of section 2.3.1 showed that the background EEG is highly correlated and thus violates one of the assumptions of the least squares method. Thus optimum estimates of the parameters cannot be obtained. Although the problem was reduced by modelling the EEG as a first order AR series, higher order AR models may give better estimates [15]. However, there are draw backs in using AR models. Firstly, the data points have only a limited accuracy, so that subtraction of terms as in equation (2.13) leads to a reduction in the accuracy of the data. Secondly, these operations are equivalent to high pass filtering and therefore, the parameter etimates are determined only by the changes in the data. This in itself is not a bad thing since the detection and removal of changes in potential due to ocular movement and not eye fixation is the goal in OA removal.

(b) Collinearity

The results of this investigation showed that for any ocular movement, at least a pair of EOGs were highly correlated. The correlation is probably because in most cases both eyes move in unison and there is no isolation between the vertical and horizontal measuring sites. It is possible, therefore, that for some ocular artefacts (e.g. blinks and vertical eye movement) a single parameter model is sufficiently sensitive for satisfactory correction. However, when the best possible correction is required or when there is less correlation between the EOGs as in lateral and random eye movements more EOGs are required. However, when additional EOGs are used computational difficulties due to collinearity may arise [9]. If the EOGs were strongly collinear, as in the cases studied, it is not possible to estimate the exact contribution of each EOG signal and the parameter estimates are less reliable [9,16,17].

(c) Comparison and Choice of Models

A good model should include at least a horizontal and a vertical EOG signal. On the basis of the results it appears that these should be the horizontal and the vertical right EOGs. As mentioned earlier, there is no need to use both vertical EOGs as they are nearly always highly correlated and may introduce significant secondary artefacts into the EEG. Account should also be taken of the various observed effects in the EEG and EOGs such as the opposing effects of the eye dipoles during lateral movements [5,6] and the slight curvatures in the scatter diagrams. Model 4D (which used VL,HLxHR,HL,HR) was chosen for implementation on-line, but with hind sight it was thought that a model that used (VR,HLxHR,HL,HR) would be a better choice. A brief investigation confirmed this to be so. However, if the concept of 'fewer is better' is used then model 3D (which used VR,HL,HR) should be used. As discussed in chapter 5, the new ocular artefact removal system infact implements the three models: 2D, (which uses

VR,HR), 3D (which uses VR,HR and HL) and the modified 4D (which uses VR,HR,HL and HRxHL). The desired model is then selected by the user- again illustrating the versatility of the on-line removal method developed in this work.

Jervis et al [5,6] compared a number of models which included model 4A (which used all the four EOGs) and some of its subsets, using a quantitative test based on the autocorrelation functions of the corrected EEGs. They found that model 4A was the best. However, the presence of strong serial correlation in the EEG and the large secondary artefacts associated with this model could significantly affect the ACF of the corrected EEG and make the quantitative results unreliable. These points are discussed further in the appendix A3. Fortgens and De Bruin [25] reached similar conclusions as Jervis et al [5,6] by considering 2- and 4- parameter models. They reported that at least four EOGs are required for adequate removal of OA, especially in the frontal regions. They also showed theoretically that four EOG signals were necessary assuming uncorrelated signals, but it has been shown that the EOGs are correlated so that four EOGs are unnecessary.

The OA models used in this investigation probably do not remove completely OA which is due to rider artefacts. For the rectangular EOGs used, rider artefacts tended to occur at twice the frequency of the EOGs and appeared as peaks in the corrected EEG. Examination of a number of corrected EEGs from records that contained rider artefact suggested that this may be the source of the second harmonic reported by Jervis et al [5,6]. See for example, figure 2.13. The inability of the OA models to deal with multiple artefacts is mainly because all data are given equal weight in the estimation of θ and only simultaneous changes in the EOGs and the EEG signals are taken into account. Some form of dynamic model that does not give equal weight to all M data points would appear to be more

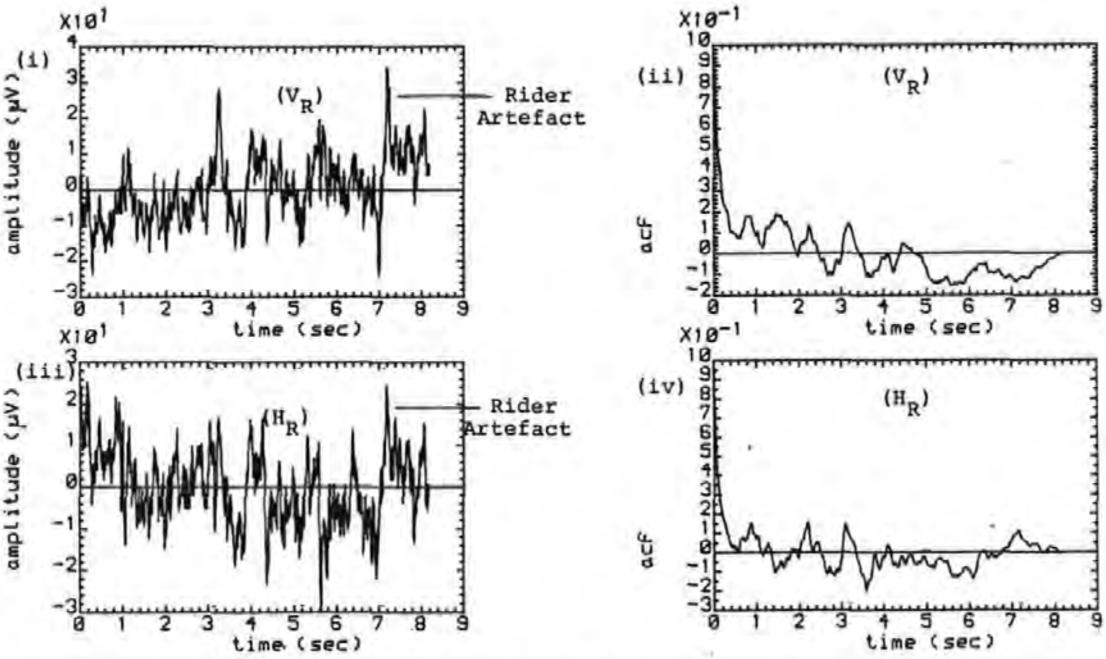


Figure 2.13 Corrected EEG waveforms and their ACFs for a VEM record. showing imperfectly corrected Rider Artefacts. (one parameter models were used. (i), (ii) model that used V_R; (iii), (iv) model that used H_R)

(Raw data are given in figure 1.2a, p5. See also the back pocket).

appropriate. Models incorporating some of these ideas have been simulated on a main frame computer and will be discussed in chapter 4, when attention will be focused on on-line removal of OA.

2.6 Summary

In this chapter a formal ocular artefact removal algorithm using least squares methods has been developed. Three problems of parameter estimation using this method and their solutions have been discussed. These are autocorrelation of the EEG, collinearity of the EOGs and EOG artefacts.

The common assumption, implicit in all EOG subtraction methods, that the error terms (or background EEG) are uncorrelated was shown to be invalid. The effects of this invalid assumption on the ocular artefact parameter estimates were reduced by modelling the error terms as a first order autoregressive series.

It was shown that for most ocular movements there were at least a pair of EOGs that were strongly correlated. Thus there is no need to use all four EOGs (one vertical and one horizontal for each eye) in the model for removing OAs as previously reported [5,6,25].

The performance of several models were compared using quantitative and pictorial criteria and it was found that regardless of the type of ocular movement, there was always a smaller model that has a performance that is comparable or better than the full model. Overall, the model using the vertical right EOG (VR), the two horizontal EOGs (HL and HR) and their product (HLxHR) gave the best performance. The model using the vertical right EOG and the two horizontal EOGs was found to be the best three-parameter model. This updates the conclusions of Jervis et al [5,6] and Fortgens and De Bruin [25].

References for Chapter 2

[1] EYKOFF P.

System Identification, Parameter and State Estimation. John Wiley,1974.

[2] HSIA T.C.

System Identification.Lexington Books,1977.

[3] IFEACHOR E.C.,JERVIS B.W,ALLEN E.,MORRIS E.L., WRIGHT D.W. and HUDSON N.G.

Investigation and Comparison of Some Models for Removing Ocular Artefacts from EEG signals. (To be published)

[4] DRAPER N.R. and SMITH H.

Applied Regression Analysis. John Wiley,New York,1980.

[5] JERVIS B.W.,NICHOLS M.J.,ALLEN E.,HUDSON N.G. and JOHNSON T.E.

The Quantitative Assesment of Electroencephalograms corrected for eye movement artefacts. First European Signal Processing Conference, Lausanne,Sept.,1980.

[6] JERVIS B.W.,NICHOLS M.J.,ALLEN E.,HUDSON N.G. AND JOHNSON T.E. Comparison of two Methods for Removing Eye Movement Artefact from the EEG Signals (Submitted to Electroenceph. Clin. Neurophysiol.).

[7] QUILTER P.M.,MACGILLVRAY B.B. and WADBROOK D.G.

The Removal of Eye Movement Artefact from EEG Signals using Correlation Techniques. Random Signals Analysis,IEE Conference Publication,No. 159,1977,pp93-100.

[8] COCHRANE D. and ORCUTT G.H.
Application of Least Squares Regression to Relationships Containing Autocorrelated Error Terms. J. Amer. Stat. Assoc. 1949,44,pp. 32-61.

[9] JOHNSTON J.
Econometric Methods. 2nd ed,McGraw-Hill,New York,1972.

[10] CHATTERJEE S. and PRICE B.
Regression Analysis By Example. John Wiley,1977.

[11] DURBIN J. and WATSON G.S.
Testing for Serial Correlation in Least Squares Regression I. Biometrika,1950,37,pp. 409-428.

[12] DURBIN J. and WATSON G.S.
Testing for Serial Correlation in Least Squares Regresion II. Biometrika,1951,38,pp. 159-178.

[13] BOX G.E.P. and JENKINS G.M.
Time Series Analysis Forecasting and Control. Holden-Day,1970.

[14] JENKINS G.M. and WATTS D.G.
Spectral Analysis ans its Applications. Holden-Day,1968.

[15] ISAKSSON A.WENNBERG A. and ZETTERBERG L.H.
Computer Analysis of EEG Signals with Parametric Models. Proc. IEEE,1982,69,pp.451-461.

[16] SILVEY S.D.
Multicollinearity and Imprecise Estimation. J. Royal Statist. Soc.,Series B,1969,31,pp. 539-552.

[17] FARRAR D.E. and GLAUBER R.R.
Multicollinearity in Regression Analysis: the Problem Revisited. The Review of Econ. Stat.,1967,49:92-107.

[18] WEISBERG S.
Applied Linear Regression. John Wiley,1980.

[19] GRATTON G.,COLES M.G.H. and DONCHIN H.
A new Method for Offline Removal of Ocular Artefacts. Electroenceph. Clin. Neurophysiol.,1983,35,pp468-484.

[20] BARLOW J.S. and RÉMOND A.
Eye Movement Artifact Nulling in EEGs by Multichannel On-line EOG Subtraction. Electroenceph. Clin. Neurophysiol., 1981,52,pp. 418-423.

[21] MALLOWS C.L.
Some Comments on Cp. Technometrics,1973,15,pp. 661-675.

[22] HOCKING R.R.
The Analysis and Selection of Variables in Linear

Regression, Biometrics,1976,32,pp. 1-49.

[23] DANIEL C. and WOOD F.S.
Fitting Equations to Data.John Wiley,1971.

[24] GIRTON D.G. and KAMIYA J.A.
A Simple On-Line Technique for Removing Eye Movement Artifacts from the EEG. Electroenceph. and Clin. Neurophysiol., 1973, 34, pp212-216.

[25] FORTGENS C. and DE BRUIN M.P.
Removal of Eye Movement and ECG Artifacts from the Non-cephalic Reference EEG. Electroenceph. and Clin. Neurophysiol., 1983, 56, pp90-96.

[26] RYAN T.A., JOINER B.L. and RYAN B.F.
Minitab Reference Manual. Pennsylvannia State University, 1982.

Chapter 3- SPECTRAL ANALYSIS OF THE OCULAR ARTEFACT SYSTEM

3.0 Introduction

In this chapter, several aspects of the OA problem are explored using spectral analysis methods. Although the EEG subtraction technique may be carried out in the frequency domain [1], the main value of spectral analysis in OA is probably as an exploratory tool.

Three aspects of the OA system were investigated, these are:

(i) Power spectra of the corrected EEG to establish whether the OA has been completely removed.

(ii) The relationships between the EOG and its counterpart in the EEG using cross and coherence spectra. These allow an estimate of the range of frequencies over which the EOG and OA are related to be obtained, and this in turn allows appropriate characteristics of a filter for removing the EOG noise to be defined.

(iii) The magnitude-frequency response of the OA system for various types of ocular movement. This method could explain the differences in the ocular artefact parameters for different types of ocular movement.

In section 3.3, an appropriate digital filter for filtering out the EOG noise is discussed. The work on digital filtering was carried out in collaboration with a final year undergraduate student supervised by the author, and formed the basis of a paper [2].

3.1 Summary of Spectral Analysis Methods Used

In this section, an outline of the spectral analysis methods used which is considered necessary to follow subsequent discussions is given. More detailed discussions of both the theory and practice of spectral analysis methods can be found in the literature [3-9]. Four spectral measures were used, namely, Power-, cross-, coherence-, and gain spectra.

(a) Power Spectra and Cross Spectra

Two spectral estimation methods were used in the analysis, namely, the method of modified periodogram or Welch method [3,4,9] and the correlation method [5].

(i) The Method of Modified Periodogram or Welch Method. Estimation of Power Spectra.

In this approach, the data sequence (x(n),n=0,1,...N-1) whose spectrum is to be estimated is first divided into S overlapping segments each of length L. The transform, $X_i(m)$, and periodogram (or raw spectrum), $I_i(w_m)$, of each segment are then obtained as:

$$X_i(m) = \frac{1}{L} \sum_{n=0}^{L-1} W(n) x_i(n) e^{-j(2\pi)nm/L} \quad m=0,\ldots L-1 \qquad (3.1a)$$

$$I_i(w_m) = \frac{L}{2\pi Q} \left| X_i(m) \right|^2 \qquad i=0,1,\ldots (S-1)$$

$$w_m = 2\pi m/L \quad m=0\ldots L/2 \quad (3.1b)$$

where $j = \sqrt{-1}$, $W(n)$ is an appropriate window function (see later) used to reduce the effects of finite length record on the spectral estimates, $x_i(n)$ is the ith data segment and, Q, the window energy [9] defined as:

$$Q = \frac{1}{L} \sum_{n=0}^{L-1} W^2(n)$$

The estimate of the power spectrum of the original sequence x(n), is the average of the periodograms of equation (3.1b):

$$\hat{S}_{xx}(w_m) = \frac{1}{S} \sum_{i=0}^{S-1} I_i(w_m) \qquad w_m = 2\pi m/L \quad m=0,..L/2 \qquad (3.2)$$

The Discrete Fourier Transform (DFT) coefficients, $X_i(m)$ in equation (3.1a) are efficiently obtained via the Fast Fourier Transform (FFT) algorithm [9,10]. To increase the spectral resolution of $I_i(w_m)$ and hence $\hat{S}_{xx}(w_m)$, the length of the segments are extended, after windowing, to say (L+R) by adding R zeros, and the transform of the augmented sequence is then obtained [9].

To obtain optimum estimates of the spectrum, the segment length and/or overlap between segments have to be varied by trial and error. In the investigation the overlap between segments was kept constant at half the segment length, so that only the segment length was varied. The segment, $x_i(n)$, and the original data sequence, x(n), are related as:

$$x_i(n) = x[n + iL/2] \text{ , } i = 0,1, \ldots (S-1) \qquad (3.3)$$

This approach, which is due to Welch [3,4], yields estimates with lower variance than those from a direct transformation of the original sequences or when there is no overlap between the segments, but at the cost of increased computation [4,9].

Estimation of Cross Spectra

For the cross spectra, each of the sequences x(n) and y(n), n=0,1,..N-1, is divided into S segments as described above. Each segment is then windowed and transformed (after zero padding if necessary) to give $X_i(m)$ and $Y_i(m)$ as in equation (3.1). The periodogram and cross spectral estimates are then given by:

$$I_i(w_m) = \frac{L}{2\pi Q} X_i(m) Y_i^*(m) \qquad (3.4a)$$

$$\hat{S}_{xy}(w_m) = \frac{1}{S} \sum_{i=0}^{S-1} I_i(w_m) \qquad (3.4b)$$

$$w_m = 2\pi m/L \quad m=0,..,L/2$$

where $Y^*_i(m)$ is the complex conjugate of the transform of $y_i(m)$. The Cross spectrum is complex, so that it may be expressed as:

$$\hat{S}_{xy}(w_m) = |\hat{S}_{xy}(w_m)| \angle \hat{\Psi}_{xy}(w_m) \qquad (3.5)$$

where $|\hat{S}_{xy}(w_m)|$ and $\angle \hat{\Psi}_{xy}(w_m)$ are, respectively, the magnitude and phase of the cross spectrum.

(ii) Correlation Method

Given a data sequence, x(n),n =0,1,...N-1, the estimate of the spectrum is given by:

$$\widehat{S}_{xx}(w_m) = \sum_{k=-(M-1)}^{M-1} W(k)\widehat{C}_{xx}(k)e^{-j(2\pi/M)mk} \qquad (3.6)$$

$$w_m = 2\pi m/M, \; m=0,\ldots,M/2$$

where $W(k)$ is a lag window [5], M the window length, and $\widehat{C}_{xx}(k)$ is an estimate of the autocovariance function (ACVF) of $x(n)$ given by:

$$\widehat{C}_{xx}(k) = 1/N \sum_{n=0}^{N-k-1} x(n)x(n+k) \quad , \widehat{C}_{xx}(-k) = \widehat{C}_{xx}(k) \qquad (3.7)$$

$$0 \leqslant k \leqslant (N-1)$$

As in the modified periodogram method, the lag window is used to reduce the effects of using a finite length record. Optimum spectral estimates are obtained by varying the window length, M, by trial and error. The 'window closing' technique described by Jenkins and Watt [5] was used, and involves computing spectral estimates with progressively larger window lengths until satisfactory estimates are obtained. In general, the larger the window length the less the bias in the spectral estimates, but the larger the variance. At some point further increase in the window length may lead to unstable (or wildly fluctuating) spectral estimates. Thus the window closing technique is essentially a method of trading-off between bias and variance.

For cross spectra, given the data sequences, $x(n)$ and $y(n)$ ($n=0,1,\ldots N-1$), an estimate of the cross spectra is given as:

$$\hat{S}_{xy}(w_m) = \sum_{k=-(M-1)}^{M-1} W(k)\hat{C}_{xy}(k)e^{-j(2\pi/M)mk} \qquad (3.8)$$

$$w_m = 2nm/M, \quad m=0,\ldots,M/2$$

where $\hat{C}_{xy}(k)$ is an estimate of the crosscovariance function given by:

$$\hat{C}_{xy}(k) = 1/N \sum_{n=0}^{N-k-1} x(n)y(n+k) \qquad 0 \leqslant k \leqslant (N-1)$$

$$\hat{C}_{xy}(-k) = 1/N \sum_{n=0}^{N-k-1} x(n+k)y(n) \qquad 0 \leqslant k \leqslant (N-1) \qquad (3.9)$$

Estimates of C_{xx}, C_{xy}, S_{xx} and S_{xy} may be obtained using the FFT algorithm, but the algorithms given in [5] were used instead so that the various techniques suggested therin could be explored.

Computationally, the method of Welch is more efficient than the correlation method due to the efficiency of the FFT algorithm. For some window functions, e.g Hanning-Tukey, the correlation method can produce negative values of power spectrum [5]. An attraction of the correlation method is that it tends to average out the noise in the data before its transform is taken [11], so that the transform of a recurring signal 'buried' in noise may still be obtained. The Welch method transforms the data (plus any noise) directly, although the averaging of the periodograms in equation (3.2) also reduces the effects of noise. It has been pointed out [11] that the correlation method may not be suitable where the signal is non-stationary in nature, as the method is based on the assumption that the signal source is ergodic, which would no longer be satisfied.

(iii) Window Functions

In the two estimation methods described above, the number of data points used in the transform is finite. A finite length data record can be viewed as the product of a rectangular window of unit height and the actual data. Such a window and its transform are depicted in figure 3.1.

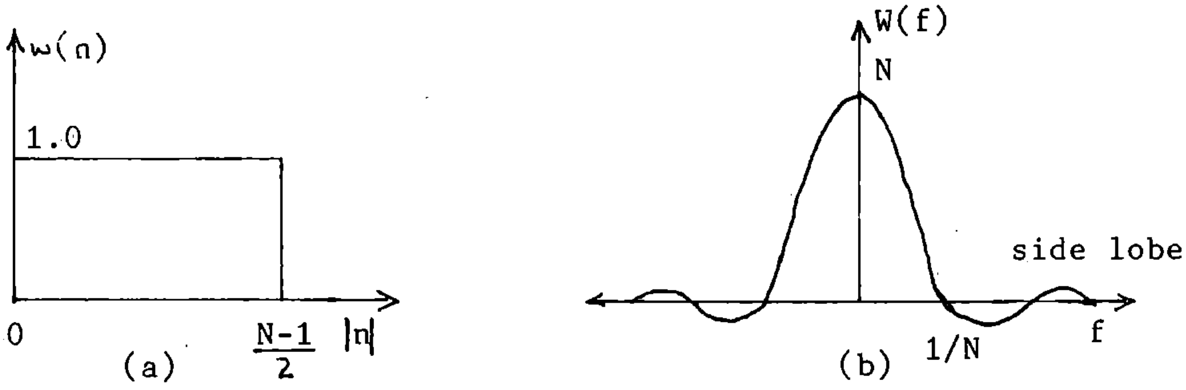


Figure 3.1 (a) A rectangular window and (b) its transform

In the Welch method, the spectra is obtained from the transform of the finite length record (see equation 3.1). Since multiplication in the time domain is equivalent to convolution in the frequency domain [17], the convolution of the spectrum of figure 3.1b and the data spectrum leads to a broadening of the peaks in the estimated spectrum and, due to the side lobes in figure 3.1b, a leakage of energy from one frequency into another. The broadening of peaks may cause two peaks that are very close to fuse together into one peak [11], and energy leakage may lead to spurious peaks in the spectral estimates. Window functions referred to above have reduced sidelobes and are used to reduce leakage, but often at the expense of broader peaks [5,11].

Similar effects to those above are obtained in the correlation method. In this case, however, the use of a finite length record leads to biased estimates of the covariance functions (equations 3.7 and 3.9) since, for example, the expected value of the ACVF is given by [2,5]:

$$\mathcal{E}[\,\hat{C}_{xx}(k)\,] = (1 - k/N)C_{xx}(k) \qquad (3.10)$$

Thus the spectral estimates obtained from the biased covariance functions again can be viewed as the convolution of the true spectrum and the spectrum of the window funtion, (1 - k/N) [3,5].

Two popular windows, the PARZEN and HANNING-TUKEY windows [5,12], were used in the investigation. For the Welch method, they have the form:

PARZEN

$$W\left[\frac{L}{2} + k\right] = 1 - 6\left[\frac{2k}{L-1}\right]^2 + 6\left[\frac{2|k|}{L-1}\right]^3 \qquad 0 \leqslant |k| \leqslant \frac{L}{4}$$

$$= 2\left[1 - \frac{2|k|}{L-1}\right]^3 \qquad \frac{L}{4} < |k| < \frac{L}{2}$$

$$= 0 \quad \text{elsewhere}$$

HANNING-TUKEY

$$W\left[\frac{L}{2} + k\right] = 0.5[1 + \cos(2\pi k/L-1)] \qquad k= -\frac{(L-1)}{2},..,\frac{(L-1)}{2}$$

$$= 0 \quad \text{elsewhere}$$

Following Jenkins and Watt [5], only one half of the windows were used for the correlation method (the windows are even functions), thus they have the form:

PARZEN

$$W(k) = 1 - 6\left[\frac{k}{M}\right]^2 + 6\left[\frac{|k|}{M}\right]^3 \qquad 0 \leqslant |k| \leqslant \frac{M}{2}$$

$$= 2\left[1 - \frac{|k|}{M}\right]^3 \qquad \frac{M}{2} < |k| \leqslant M$$

$$= 0 \text{ elsewhere}$$

HANNING-TUKEY

$$W(k) = 0.5[1 + \cos(\frac{\pi k}{M})] \qquad 0 \leqslant |k| \leqslant M$$

The windows produce similar but not identical spectral estimates due to differences in their shapes.

(b) Coherence Spectrum

The relationship between the EOG (input) and the EEG (output) may be described by the coherence spectrum. This is the counterpart of correlation coefficient (for a single-input-single-output system) in the time domain at each frequency [5].

The coherence spectrum, $\hat{\rho}^2(w_m)$, is given by [5-7]:

$$\hat{\rho}^2(w_m) = \frac{|\hat{S}_{xy}(w_m)|^2}{\hat{S}_{xx}(w_m)\hat{S}_y(w_m)} \qquad (3.11)$$

where $\hat{S}_{xx}(w\ )$, $\hat{S}_y(w_m)$ and $\hat{S}_{xy}(w_m)$ are respectively, estimates of the input-, output-, and cross spectrum using either of the estimation methods described earlier.

Jenkins and Watt [5] have shown that, at least in theory, the coherence function satisfies the inequality: $0 \leq \hat{\rho}^2(w_m) \leq 1$. A coherence of unity at a given frequency implies that the input and output signals are perfectly correlated at that frequency, and a coherence of 0 implies zero correlation [5-7].

(c) Frequency Response

The frequency response of a linear system, such as that given in figure 2.1, but for a single input, is given by:

$$\hat{H}_{xy}(w_m) = \hat{S}_{xy}(w_m)/\hat{S}_{xx}(w_m) \qquad (3.12)$$

This may be expressed as consisting of an absolute value $|\hat{H}_{xy}(w_m)|$, known as the gain, and a phase angle, $\hat{\Psi}_{xy}$. Thus :

$$|\hat{H}_{xy}(w_m)| = |\hat{H}_{xy}(w_m)| \quad \underline{\angle\hat{\Psi}_{xy}(w_m)} = \hat{G}_{xy}(w_m) \quad \underline{\angle\hat{\Psi}_{xy}(w_m)} \qquad (3.13)$$

where ,

$$\hat{G}_{xy}(w_m) = |\hat{H}_{xy}(w_m)|, \quad \underline{\angle\hat{\Psi}_{xy}(w_m)} = \tan^{-1}[\hat{Q}_{xy}(w_m)/\hat{C}_{xy}(w_m)]$$

$\hat{Q}_{xy}(w_m)$ and $\hat{C}_{xy}(w_m)$ are the real and imaginary parts of $\hat{H}_{xy}(w_m)$.

For an n-input system, $\hat{H}_{xy}(w_m)$, and $\hat{S}_{xy}(w_m)$ are complex vectors of dimension n, and $\hat{S}_{xx}(w_m)$ is an nxn complex matrix, so that (where the symbol $\sim$ is used here to distinguish these matrices from the scalars above):

$$\tilde{S}_{xy}(w_m) = \tilde{S}_{xx}(w_m)\tilde{H}_{xy}(w_m) \qquad (3.14)$$

Equation 3.14 above is analogous to the OA model of equation 2.1, and $\tilde{H}_{xy}(w_m)$ may be viewed as the counterpart of θ in the frequency domain [5-7].

In view of the high correlation between the EOGs and for simplicity, only the single input case was investigated. Furthermore, only the gain functions were computed.

3.2 Results of Spectral Analysis

Figure 3.2 shows the power spectra for the EOGs (VR,HR) and the EEGs (measured and corrected) using Welch's method. Each spectrum is normalized to have a minimum value of 0 dB. The horizontal right EOG spectrum, figure 3.2(ii), shows a well defined peak at about 0.9Hz due to the periodic nature of the original data, and a smaller peak at about 2.6Hz. The corresponding peaks are evident in the measured EEG spectrum, figure 3.2(iii), as well as other smaller peaks. Examination of the spectra for the EOG and EEG, figures 3.2 (ii) and (iii), shows that the second peak is more pronounced in the EEG than in the EOG, suggesting that it may have originated from the EEG. The spectrum of the EEG after correcting for OA (using model 3D in section 2.2) is shown in figure 3.2 (iv). It is seen that the fundamental EOG component has been reduced in level, but not completely removed.

Results for the same record using the correlation method are shown in figure 3.3. Comparison of figures 3.2 and

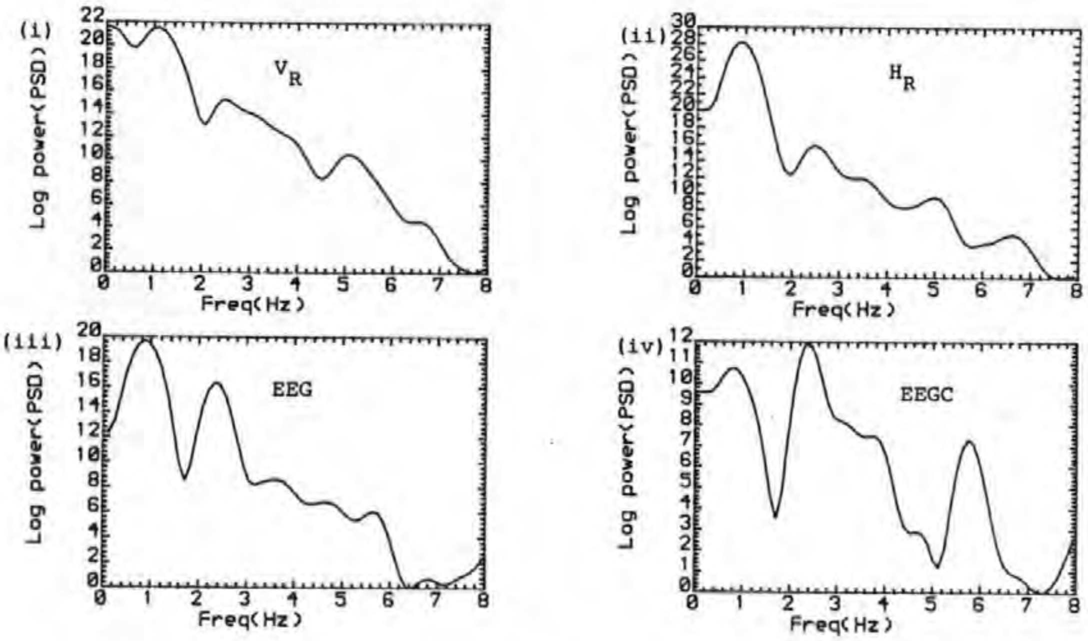


Figure 3.2 Spectra of the EOGs and the EEG for a HEM record
(i) Vertical right EOG (ii) Horizontal right EOG
(iii) measured EEG (iv) Corrected EEG
(the raw data are given in figure 4.1, p123).

3.3 shows that they are similar, but not identical due perhaps to the differences in the estimation methods.

Figure 3.4 gives the results for a blink record using Welch's method. It is seen that the corrected EEG, figure 3.4 (iv), still contains a peak at the EOG frequency (about 1.3Hz), an indication that the OA has not been completely removed. It is to be noted that the OA components in the corrected EEG sometimes occur at lower frequencies than their counterparts in the measured EEG. A consequence of this will be discussed later.

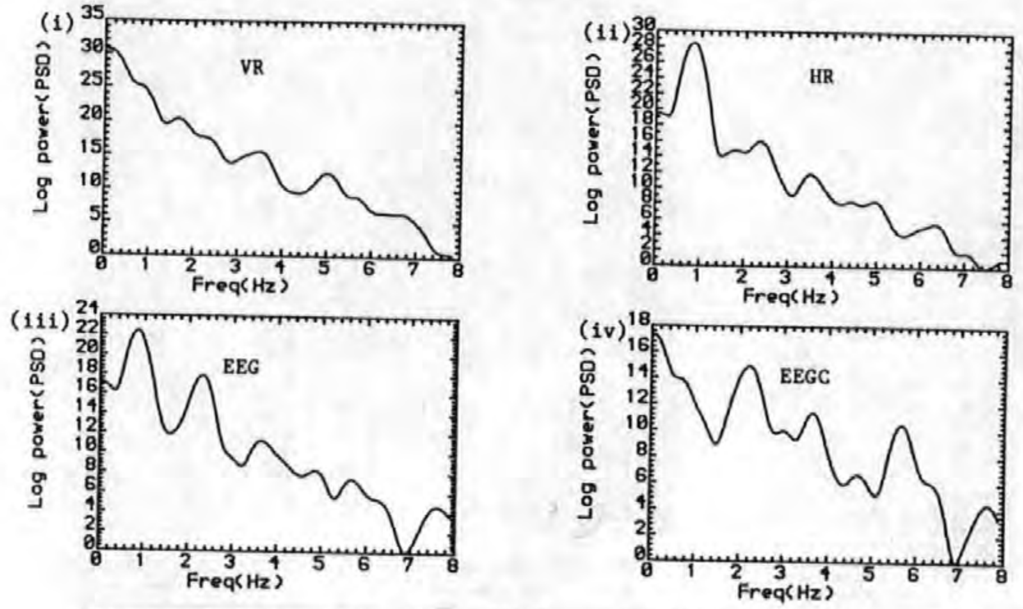


Figure 3.3 Spectra of the EOGs and EEG for a HEM record using the correlation method (M=350). (i) Vertical right EOG (ii) Horizontal right EOG (iii) Measured EEG (iv) Corrected EEG. (the raw data are given in figure 4.1, p123).

The coherence-, gain-, and cross spectra between the EOG and the EEG obtained via the Welch's method for a lateral eye movement record are given in figures 3.5(i), (ii) and (iii) respectively. The coherence spectrum has a large peak at about 1Hz, due to the periodic EOG, and falls off on either side. This indicates that the EOG and the EEG are strongly correlated at this frequency. There are also large coherencies above 2Hz, but these are thought to be spurious since the power and cross spectra indicate little power at those frequencies.

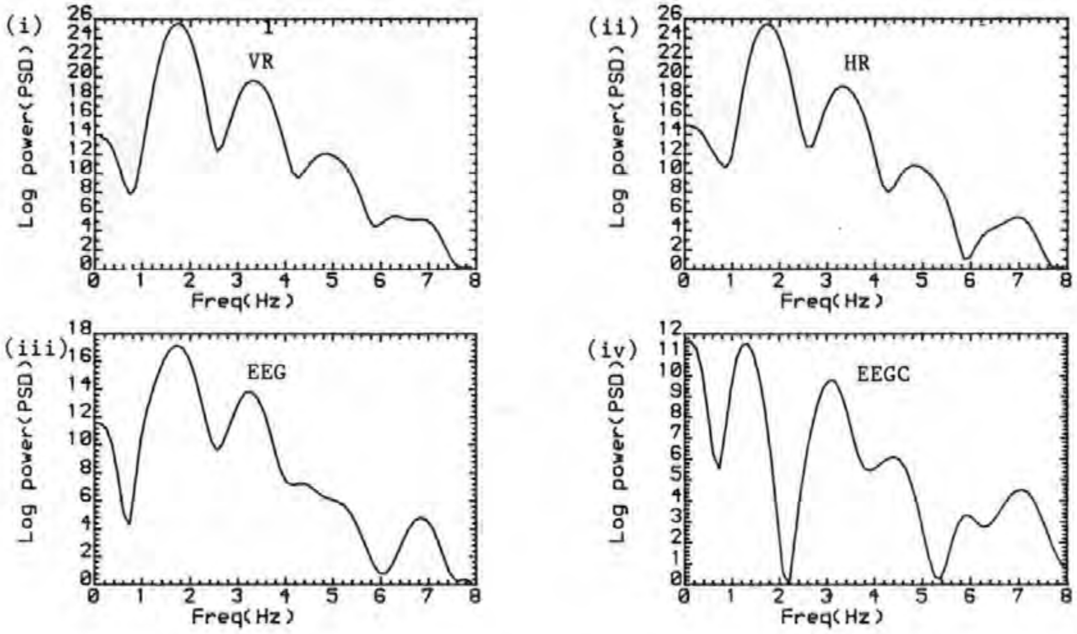


Figure 3.4 Spectra of the EOGs and the EEG for a blink record using the Welch's method (L=350). (i) Vertical right EOG (ii) Horizontal right EOG (iii) Measured EEG (iv) Corrected EEG. (the figure showing the raw data is in the back pocket).

The gain has a peak near the EOG frequency. Above about 2Hz, the gain fluctuates rapidly due perhaps to the weak relationship between the EOG and the EEG at these frequencies.

Figure 3.6 shows the coherence and gain spectra for the same record with the data differenced prior to obtaining the spectral estimates. The behaviour is somewhat similar to figure 3.5, with the notable difference being a broader gain peak. Similar results were obtained for blinks, but in these cases the gain peaks were lower in magnitude.

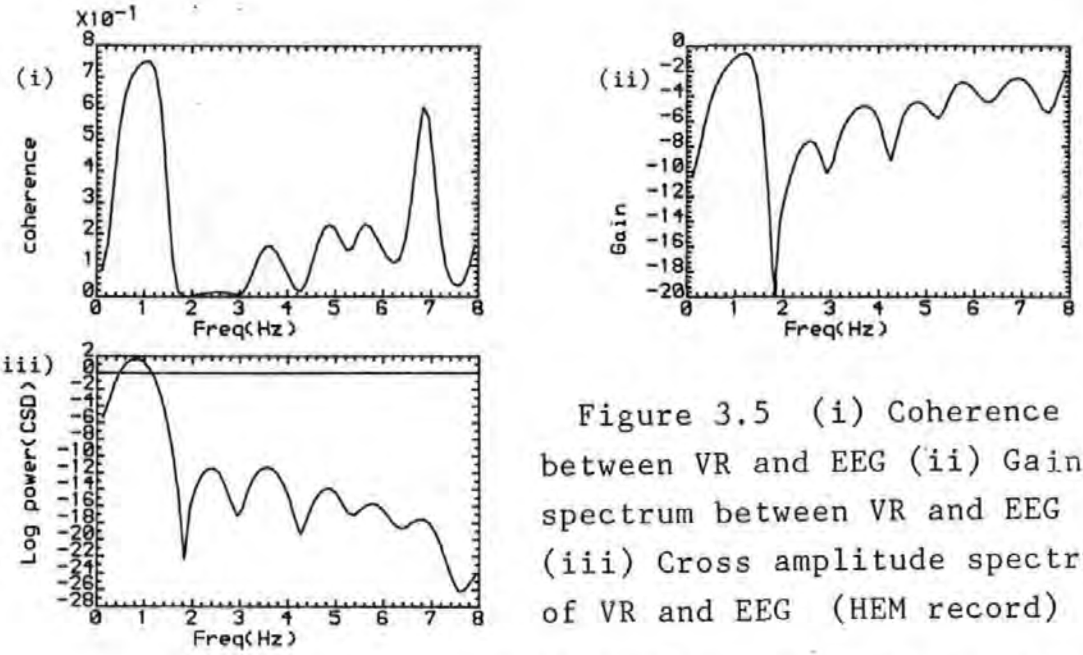


Figure 3.5 (i) Coherence between VR and EEG (ii) Gain spectrum between VR and EEG (iii) Cross amplitude spectrum of VR and EEG (HEM record)

(the raw data are given in figure 4.1, p123).

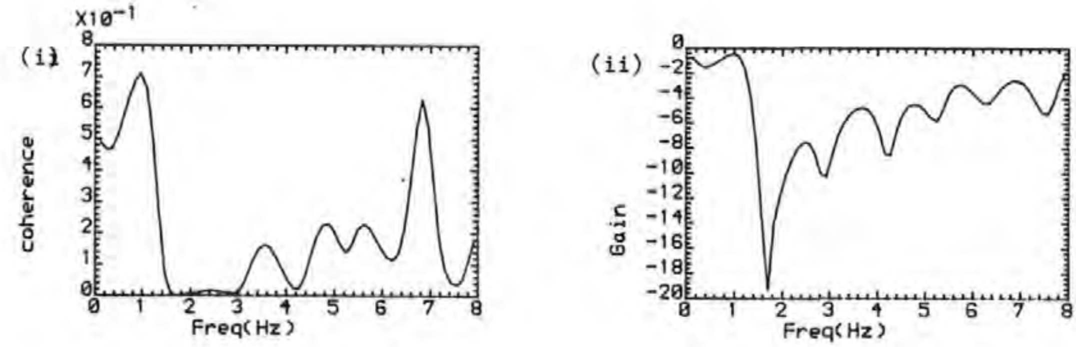


Figure 3.6 Coherence and Gain spectrum for a HEM record with data differenced prior to transformation to reduce the effects of spurious relationships (other details same as 3.5).

(the raw data are given in figure 4.1, p123).

3.3 Digital Filtering of the EOG with Minimal Distortion

As discussed in section 1.6.2, a disadvantage of the EOG subtraction method is that 'noise' in the EOG, unrelated to ocular movement, may be introduced into the corrected EEG. Thus this noise should be removed.

Whitton et al [1] found that there was little shared activity between the EOG and EEG above about 8Hz, and employed a low pass filter to remove any high frequency activity in the EOG above this frequency. The results of section 3.2 agree with their finding. However, it is considered that the use of a simple filter as in [1] is not appropriate. The simple filter with its poor amplitude and phase response would introduce distortion which would impair the removal of ocular artefacts.

The excellent filter design method of Mcllelan et al [9] was selected as the best approach. The method allows filters with specified amplitude response and linear phase to be obtained.

Using the design method, an FIR filter with the frequency response of figure 3.7 was designed. The filter has 29 coefficients, a passband of 0 to 10 Hz (to pass higher frequency components of the EOG) and a stop band loss of -33.43dB. The filter coefficients are listed in table 3.1.

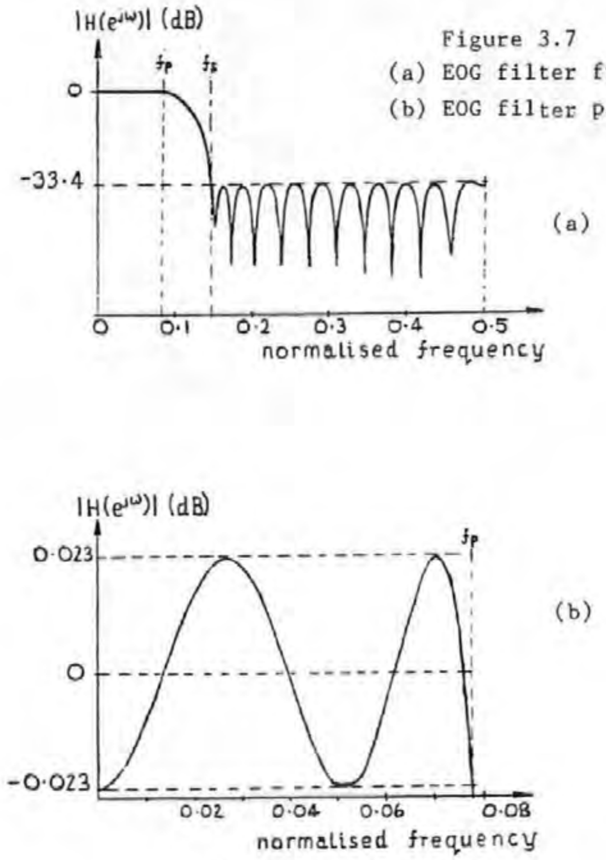


Figure 3.7
(a) EOG filter frequency Response
(b) EOG filter passband response

TABLE 3.1

Filter coefficients for EOG lowpass filter.

H(0) =	-0.11191774E-01	= H(28)
H(1) =	-0.38390798E-02	= H(27)
H(2) =	0.73034801E-02	= H(26)
H(3) =	0.11693427E-01	= H(25)
H(4) =	0.18245798E-01	= H(24)
H(5) =	0.78111943E-02	= H(23)
H(6) =	-0.75105876E-02	= H(22)
H(7) =	-0.31717509E-01	= H(21)
H(8) =	-0.41073106E-01	= H(20)
H(9) =	-0.31341068E-01	= H(19)
H(10) =	0.13298627E-01	= H(18)
H(11) =	0.79011545E-01	= H(17)
H(12) =	0.15616089E-00	= H(16)
H(13) =	0.21316704E-00	= H(15)
H(14) =	0.23729908E-00	

Rabiner's Algorithm [13], modified to take account of the impulse response symmetry for a linear phase filter, was used to realise the filter. A FORTRAN program used to perform the filtering is given in appendix A14, and an example of filtered data is given in figure 3.8.

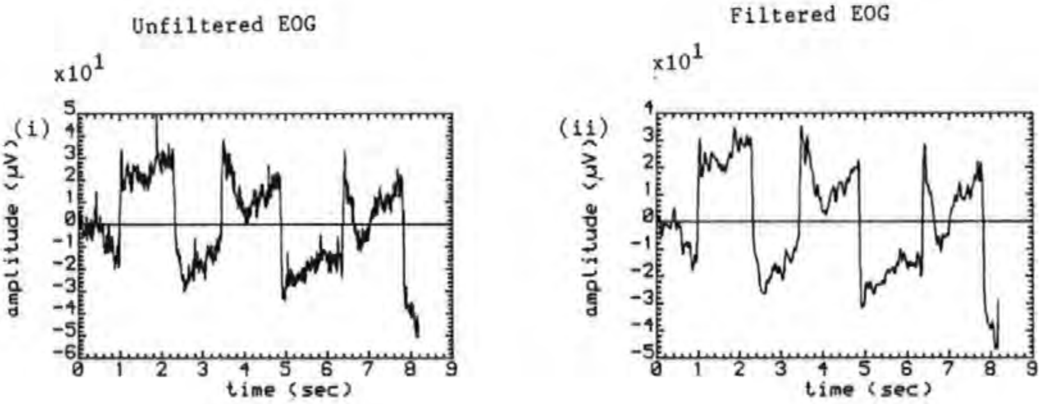


Figure 3.8 Unfiltered and filtered EOG signals

Examination of the figure shows that there is little distortion of the filtered data. However, spectral analysis of the filtered and unfiltered data did not reveal any significant differences, probably because the EOG noise was very small compared to the 'true' EOG. For this reason, the filter was used in subsequent work only in cases where it was thought that the EOG noise would affect the results.

3.4 Discussion of Results

The results of the spectral analysis of the corrected EEG indicates that the present models do not completely remove the OA from the EEG. This supports the report of Whitton et al [1]. A method for removing OA in the frequency domain described by these authors first computes the spectra of the EOG and the EEG. The EOG spectrum is then subtracted from that of the EEG, after both have been scaled so that their low frequency peaks are equal. As was found out, the EOG peaks sometimes occured at different frequencies to those of the OA, so that a direct subtraction of scaled EOG spectrum may not be wise. Also if the peaks are not well defined, as will be the case when the OA level is low, the scaling will be difficult. Furthermore, the gain fuctions suggested that the system does not have a constant gain at all frequencies, so that a scale factor obtained from one peak (which is equivalent to the gain at a single frequency) will not apply to all frequencies. A better approach in the frequency domain would probably be to obtain the corrected EEG as the difference between the product of the gain function and the EOG, and the measured EEG. In the time domain, a possible alternative to the present OA models is to use the impulse response identification method [14-16], with the system impulse duration estimated from a spectral analysis study. This is analogous to the frequency domain methods just suggested, but has the advantage that it could be implemented in real-time.

3.5 Summary

In this chapter, various aspects of the OA problem were explored using spectral analysis methods. It was found that the EOG subtraction method does not completely remove the OA.

Cross and coherency spectral analysis of the EOG and EEG showed that for the 'periodic' EOGs used, there is little shared activity between the EOG and the EEG above about 3Hz. Thus higher components of the EOG are probably due to noise and should be filtered out. A suitable digital filter which introduces minimal distortion into the EOG was discussed.

The gain function of the OA system suggested that all EOG frequency components are not attenuated by the same amount through the scalp.

References for Chapter 3

[1] WHITTON J.L., LUE F. and MOLDDOFSKY H.
A Spectral Method for Removing Eye Movement Artifacts from the EEG. Electroenceph. and Clin. Neurophysiol., 1978,44, pp 735-741.

[2] HAMMER C.F., IFEACHOR E.C. and JERVIS B.W.
Digital Filtering of Physiological Signals with Minimal Distortion (Accepted for publication by Med. & Biol. Eng. and Comp.).

[3] COOLEY J.W., LEWIS P.A.W., and WELCH P.D.
The Application of the Fast Fourier Transform Algorithm to the Estimation of Spectra and Cross Spectra. J. Sound Vib., 1970, 12,pp. 339-352.

[4] WELCH P.D.
The use of Fast Fourier Transform for the Estimation of Power Spectra: A Method Based on Time Averaging Over Short Modified Periodograms. IEEE Trans. Audio Electroacoust. 1967, 15,pp 70-73.

[5] JENKINS G.M. and WATTS D.G.
Spectral Analysis and its Applications. San Fransisco,CA: Holden-Day, 1968.

[6] OATNES R.K. and ENOCHSON L.
Digital Time Series Analysis. John Wiley, 1972.

[7] OATNES R.K. and ENOCHSON L.

Applied Time Series Analysis. Vol 1, John Wiley, 1978.

[8] GRIFITH J.W.R. and PRIETO-DIAZ R.

Spectral Analysis of Natural Seismic Events Using Autoregressive Techniques. IEEE Trans. on Geoscience Electronics, 1977,15, pp13-25.

[9] RABINER L.R. and GOLD B.

Theory and Application of Digital Signal Processing. Prentice-Hall, 1975.

[10] COOLEY J.W. and TUKEY J.W.

An Algorithm for Machine Calculation of Complex Fourier Series. Math., Comput., 1965, 19,pp 297-301

[11] YOGANATHAN A.P, GUPTA R., CORCORAN W.H.

Fast Fourier Transform in the Analysis of Biomedical data. Med. and Biol. Eng., 1976, p239-244.

[12] CHATFIELD C.

The Analysis of Time Series. Chapman and Hall, 2nd Ed, 1980.

[13] Rabiner L.R.

A Simplified Computational Algorithm for Implementing FIR Digital Filters. IEEE Trans. Acoust., Speech and Signal Processing, 1977, 25, pp259-261.

[14] RABINER L.R., CROCHIERE R.E. and ALLEN J.B.
FIR System Modeling and Identification in the Presence of Noise and with Band-limited Inputs. IEEE Trans. Acoust., Speech. and Signal Processing, 26,1978, pp319-333.

[15] WIDROW, B., GOLVER J.R.,McCOOL J.M. KAUNITZ J.,WILLIAMS C.S.,HEARN R.H.,ZEIDLER J.R.,
Adaptive Noise Cancelling: Principles and Applications. Proc. IEEE,63,1975, pp1692-1716.

[16] WEISS P.L., HUNTER I.W. and KEARNEY R.E.
Reduction of Physiological Signal Contamination using Linear Filtering Identification. Med. and Biol. Eng. & Comp. 21,1983, pp521-524.

[17] STREMLER F.G.
Introduction to Communication Systems. Addison-Wesley, 1977.

Chapter 4 ON-LINE REMOVAL OF OCULAR ARTEFACTS FROM THE EEG SIGNALS

4.0 Introduction

The present on-line methods of removing ocular artefacts (OAs) from the EEG signals employ analogue methods requiring the cooperation of subjects which cannot always be guarranteed, and are in general inferior to the off-line methods [1-7]. In this chapter, algorithms are presented which are suitable for on-line removal of OA with a performance that is as good as their off-line equivalents. These algorithms are based on the efficient recursive least squares parameter estimation technique. Compared to the present on-line methods, they are superior, requiring no subjective manual adjustment and the OA removal is adaptive (as the OA parameters are not assumed constant) and therefore more accurate.

The problems of removing OA by these algorithms and solutions are discussed. This chapter has formed the basis of a paper that it is hoped will be published soon [8], and will form part of a longer paper to be written soon.

4.1 On-line Parameter Estimation Algorithms

In chapter 2, it was shown that the ocular artefact parameters are obtained from equation 2.2, by minimising the sum of squares of the error terms,J:

$$J = \sum_i e^2(i) = E_m{}^T E_m \qquad (4.1)$$

This leads to the ordinary least squares (OLS) estimates of equation 4.2 below.

$$\hat{\theta}_m = [X_m^T X_m]^{-1} X_m^T Y_m \qquad (4.2)$$

where

$$\hat{\theta}_m = \begin{bmatrix} \hat{\theta}_1 \\ \hat{\theta}_2 \\ \cdot \\ \cdot \\ \cdot \\ \hat{\theta}_n \end{bmatrix}, \qquad X_m = \begin{bmatrix} x^T(1) \\ x^T(2) \\ \cdot \\ \cdot \\ \cdot \\ x^T(m) \end{bmatrix}, \quad Y_m = \begin{bmatrix} y(1) \\ y(2) \\ \cdot \\ \cdot \\ \cdot \\ y(m) \end{bmatrix}$$

As may be apparent, the time consuming inverse matrix has to be computed each time Θ_m is to be estimated. Clearly, this approach is not suitable for real-time or on-line estimation. In practice when continuous data are being acquired, and it is desired to improve the parameter estimates using the new data, recursive methods are preferred. With a recursive algorithm the estimates can be updated for each new set of data acquired without repeatedly solving the time consuming matrix equation of 4.2 directly.

A suitable recursive least squares (RLS) algorithm, which also tracks slowly varying parameters is obtained by exponentially weighting the data to gradually remove the effects of old data on the estimates [9-11]. Thus :

$$J = \sum_{i=1}^{m} \gamma^{m-i} e^2(i) \qquad 0< \gamma <1 \qquad (4.3)$$

Minimisation of J with respect to the θ's leads to the recursive least squares algorithm given in (4.4) [9-11], the derivation of which is given in appendix A4:

$$\hat{\theta}(m+1) = \hat{\theta}(m) + P(m+1)x(m+1)[\ y(m+1) - x^T(m+1)\hat{\theta}(m)] \qquad (4.4a)$$

$$P(m+1) = \frac{1}{\gamma}\,[P(m) - \frac{1}{\alpha}\,P(m)x(m+1)x^T(m+1)P(m)] \qquad (4.4b)$$

where,

$$\alpha = \gamma + x^T(m+1)P(m)x(m+1)$$

$$x^T(m+1) = [x_1(m+1) \quad x_2(m+1) \quad \ldots \quad x_n(m+1)]$$

and the argument (m) is used to emphasize the fact that the quantities are obtained at each sample point. γ is referred to as the forgetting factor and prevents the matrix P(m+l) from tending to zero (and $\hat{\theta}$(m+l) to a constant) with increased m, thus allowing the tracking of a slowly varying parameter. The number of the previous samples that significantly affect $\hat{\theta}$(m) is referred to as the asymptotic sample length (ASL), given by e.g. [12]:

$$asL = \sum_{j=0}^{\infty} \gamma^{j} \quad = 1/(1-\gamma) \qquad (4.5)$$

Typically, γ is between 0.98 and 1. Smaller values assign too much weight to the more recent data which leads to wildly fuctuating estimates.

As in the off-line case (see section 2.1), the on-line case is extendable to the multi-EEG channel case by similar arguments.

4.1.1 Starting the Algorithm

There are a number of ways of starting the algorithm, two of which are described below [13,14].

(a) Solve for $\theta(m)$ directly using the first m samples. That is, solve:

$$\hat{\theta}(m) = P(m)X_m^T Y_m \tag{4.6}$$

where,

$$P(m) = [X_m{}^T X_m]^{-1} \quad \text{(see appendix A4)}$$

Having obtained P(m) and $\hat{\theta}(m)$, the algorithm then iterates from the (m+1)th sample onwards. To keep the computation time to a minimum, m should be chosen to be small. The advantage of this approach is that the initial transient associated with a 'guessed' initial P(0) and $\hat{\theta}(0)$ is avoided.

(b) Assign arbitrary values to $\hat{\theta}(0)$ and then set $P(0)=\delta I$, where I is a unit matrix and δ is a positive constant. A large δ leads to rapid initial changes in $\hat{\theta}(m)$ as it implies little confidence in the value of $\hat{\theta}(0)$ [12]. A smaller δ is an indication that $\hat{\theta}(0)$ is a reasonable estimate and leads to slow initial changes. It is helpful to choose $\hat{\theta}(0)$ based on previous results and then to set δ to a small value. Having assigned values to $\hat{\theta}(0)$ and P(0), the algorithm then iterates from $\hat{\theta}(1)$ and P(1) onwards.

4.1.2 Factorization Algorithms

The RLS method is very efficient and involves exactly the same number of arithmetic operations between samples, as $\hat{\theta}$ and P in equation 4.4 have fixed dimensions. There are however, two main problems that may be encountered when the RLS algorithm is implemented directly. The first, referred to as 'blow-up', results if the signal is not 'persistently exciting' as, for example, when there is no ocular movement, leading to an exponential increase in the elements of P in equation 4.4b. Thus:

$$\lim_{m \to \infty} P_{ij}(m+1) = \lim_{m \to \infty} \frac{P_{ij}(m)}{\gamma^{m}} \longrightarrow \infty \qquad (4.7)$$

However, because of miniature ocular movements and other activities that are normally picked up in the EOG channels, this problem may not be so serious in OA removal.

The second problem with RLS is its sensitivity to computer round-off errors, which degrades the numerical accuracy and algorithm performance. For successful estimation, the matrix P is required to be positive semidefinite. (This is equivalent to requiring in the off-line case that the matrix $(X^T X)$ be invertible). If this condition is not met, then P does not exist and the algorithm becomes unstable. Because of differencing of terms in equation 4.4b, positive definiteness of P cannot be guaranteed [9,15]. This problem is worse in multiparameter models, especially if the variables (EOGs in this case) are linearly dependent [9] and when the algorithm is implemented on a small system with finite word length. When the algorithm has iterated for a long time the two terms in the bracket in equation 4.4b are very nearly equal and subtraction of such terms in a finite word length system may lead to errors and a negative definite P matrix [12,15,16].

An elegant way of solving the problems of blow-up and numerical instability is to replace the RLS algorithm with one that is numerically better conditioned. Such algorithms factorize the matrix P(m+1) so that the differencing of terms in 4.4b is avoided, and reduce the numerical ranges of the variables so that even when the data is not exciting P does not become excessively large. It is reported that these factorization algorithms have accuracies that are comparable to RLS algorithms that use double precision [15-17]. As an indication of the confidence that is placed on these algorithms, they have been used in many aerospace projects, such as the Appolo lunar mission [15-17]. Two of these algorithms will be described here.

(a) The Square Root Algorithm

In the square root method, the matrix P is factored as:

$$P(m+1) = S(m+1)S^T(m+1) \qquad (4.8)$$

where S(m+1), an upper triangular matrix and S^T (m+1) its transpose, is a square root of P(m+1). Thus if S(m+1) instead of P(m+1) is updated, the positive definiteness of P(m+1) is guarranteed since the product of two square roots is always positive (assuming real roots).

There are several approaches to square root filtering e.g [9,15,18]. The approach presented here is due to Peterka [9]. From equations 4.4 and 4.8, P(m+1) is given by:

$$P(m+1) = \frac{1}{\gamma}\left[I - \frac{1}{\alpha} P(m)x(m+1)x^T(m+1)\right]P(m)$$

$$= S(m+1)S^T(m+1) = \frac{S(m)}{\sqrt{\gamma}} \left[I \;\vdots\; \frac{jf(m+1)}{\sqrt{\alpha}} \right] QQ^T \left[\begin{array}{c} I \\ \hdashline \frac{jf^T(m+1)}{\sqrt{\alpha}} \end{array} \right] \frac{S^T(m)}{\sqrt{\gamma}} \qquad (4.9)$$

where $f^T(m+1) = x^T(m+1)S(m)$, $j=\sqrt{-1}$ and Q is an orthogonal matrix. Noting the symmetry in equation (4.9), S(m+1) may be written as:

$$S(m+1) = \frac{1}{\sqrt{\gamma}} S(m) \left[I \;\vdots\; \frac{jf(m+1)}{\sqrt{\alpha}} \right] Q \qquad (4.10)$$

By choosing an appropriate orthogonal matrix Q, equation 4.10 reduces to:

$$S(m+1) = \frac{1}{\sqrt{\gamma}} S(m)H(m) \qquad (4.11)$$

where H(m) is an upper triangular matrix.

Thus, S(m+1) can be updated from S(m). Peterka [9] has derived a recursive algorithm for updating S(m+1) which is given in appendix A5 together with a more detailed derivation of the square root filter.

(b) U-D Factorization Algorithm

The U-D factorization algorithm is an alternative method of updating the covariance matrix P(m+1) in a numerically stable form [15-17]. It is, in fact, a square-root-free arrangement of the conventional square root algorithm and thus shares the same properties as the latter. In this method P(m+1) is factored as:

$$P(m+1) = U(m+1)D(m+1)U^T(m+1) \tag{4.12}$$

where U(m+1) is a unit upper triangular matrix, $U^T(m+1)$ is its transpose and D(m+1) is a diagonal matrix. Thus instead of updating P its factors U and D are updated.

Using equation 4.12, equation 4.4b may be written as:

$$P(m+1) = \frac{1}{\Upsilon} U(m)[D(m) - \frac{1}{\alpha} vv^T]U^T(m) \tag{4.13}$$

where $v = D(m)U^T(m)x(m+1)$

If the term in the square bracket is further factored into an upper triangular and diagonal matrices, such that

$$\overline{U}(m)\overline{D}(m)\overline{U}^T(m) = [D(m) - \frac{1}{\alpha} vv^T] \tag{4.14}$$

where the bar is used to distinguish the U-D factors of $[D(m) - \frac{1}{\alpha} vvT]$ from those of P, then:

$$P(m+1) = \frac{1}{\Upsilon} U(m)\overline{U}(m)\overline{D}(m)\overline{U}^T(m)U^T(m) \qquad (4.15)$$

comparing equations (4.12) and 4.15), and noting that the product of upper triangular matrices is itself upper triangular and the symmetry in equation (4.15) then:

$$U(m+1) = U(m)\overline{U}(m)$$

$$D(m+1) = \frac{1}{\Upsilon}\overline{D}(m)$$

Thus the problem of updating U(m+1) and D(m+1) depends on finding appropriate recursive formulas for $\overline{U}(m)$ and $\overline{D}(m)$. Bierman [15] has given an algorithm for updating U(m+1),D(m+1) recursively for the Kalman filter, which uses the variance of the error term, e(i), but not Υ. This algorithm has been trivially modified for the OA problem to encorporate Υ instead, as has the presentation given above. The modified algorithm is given in appendix A6.

Compared to the square root filter, the U-D method is more efficient in terms of storage and computation. For an n-parameter estimation problem, the U-D and the square root algorithms require about $(1.5n^2 + 1.5n)$ and $(2n^2 + 2.5n)$ multiplications, respectively per update for P [15-17]. Additionally, the square root method requires the evaluation of n square roots. Also in the U-D method, the elements of D give directly the diagonal elements of P, which can be used for statistical analysis (see appendix A1). For these reasons, the U-D method is preferred and was used in the OAR system discussed in chapters 5 to 7.

4.2 Computer Simulation of On-line Removal of OA

To verify that the recursive algorithms described above when used on-line will give the same results as the off-line (or batch) method, several EEG records contaminated by ocular artefacts were corrected on a main frame computer using both methods.

4.2.1 Experimental Data

The data used in the investigation were the same data that were used in the preceding chapters and described in section 1.6. Figure 4.1 gives typical waveforms of EOGs and the contaminated EEG for a horizontal eye movement.

4.2.2 Models

The models given in equation 4.16 below were used in the investigation. Two of these models (3D and 4D*) had already been found to give the best ocular artefact removal in the previous investigation (see chapter 2), and the third is included to show the effects of EOG 'noise' and correlation among the EOGs on the parameter estimates.

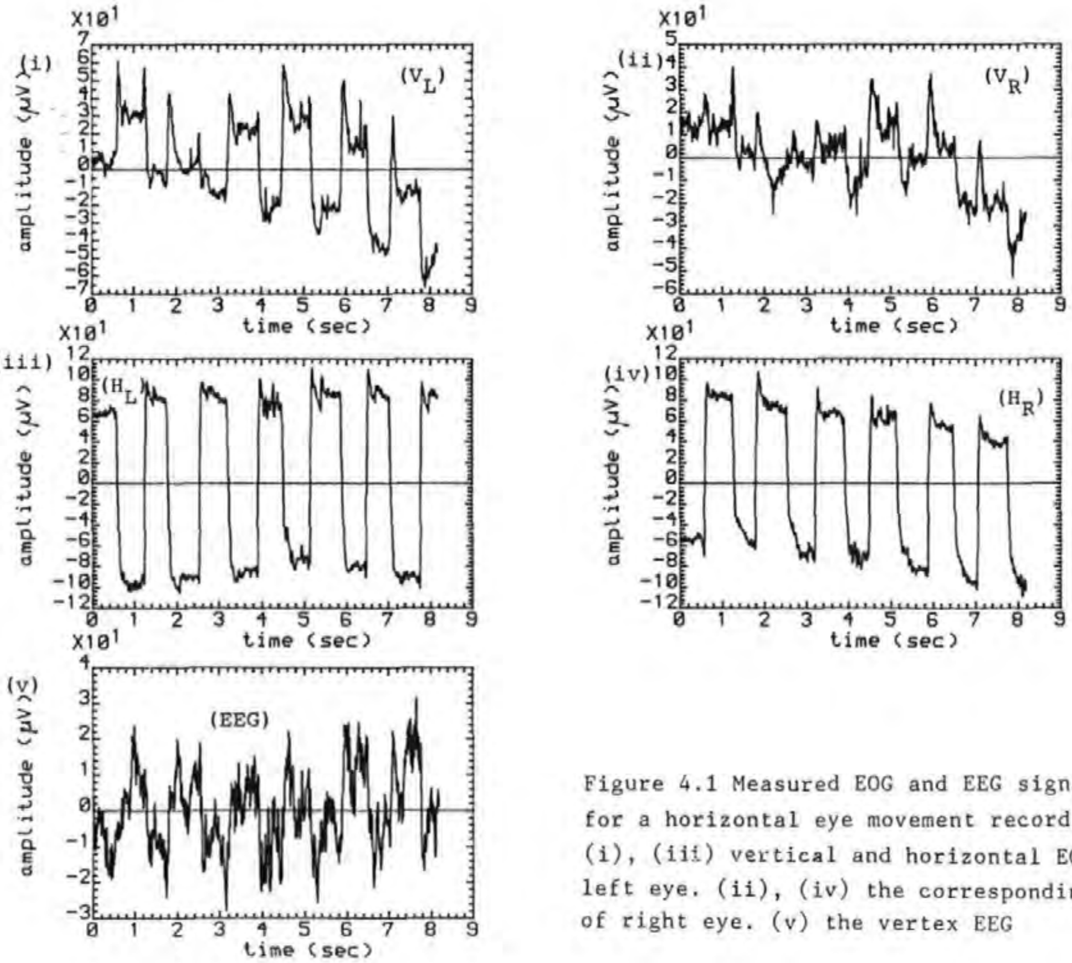


Figure 4.1 Measured EOG and EEG signals for a horizontal eye movement record. (i), (iii) vertical and horizontal EOGs of the left eye. (ii), (iv) the corresponding EOGs of right eye. (v) the vertex EEG

3D $y(i) = \theta_1 VR(i) + \theta_2 HR(i) + \theta_3 HL(i) + e(i)$ (4.16a)

4A $y(i) = \theta_1 VR(i) + \theta_2 HR(i) + \theta_3 HL(i) + \theta_4 VL(i) + e(i)$ (4.16b)

4D* $y(i) = \theta_1 VR(i) + \theta_2 HR(i) + \theta_3 HL(i) + \theta_4 HL(i) x HR(i) + e(i)$ (4.16c)

where y(i), e(i) are, respectively the ith samples of the measured scalp signal and background EEG. VL(i) and HL(i) are, respectively, the ith samples of the vertical and horizontal left eye EOGs. VR(i) and HR(i) are the corresponding right eye EOGs. The θ_j have the usual meaning. (the symbol * is used to indicate that this model is not the same as the model 4D defined in chapter 2 which used VL instead of VR as above, and the EOG variables have been re-ordered for convenience).

4.2.3 Results and Discussions

The results presented in this section for the on-line methods were obtained using the UD algorithm discussed above, although similar results were obtained for the recursive least squares- (equation 4.4) and the square root algorithms, as will be pointed out.

(a) Comparison of the On-line Methods with the Off-line Method

Figures 4.2 and 4.3 give the ocular artefact estimates and the corrected EEGs for these models for the batch and the U-D methods respectively. Ignoring the initial transients, both the OA estimates and the corrected EEG for the two methods are similar. Table 4.1 compares the parameter estimates for the batch or conventional off-line method with those for the U-D algorithm after 1024 iterations. An examination of the table shows that the parameter estimates for both methods are essentially the same, an indication that they are equivalent. Similar results to figures 4.2 and 4.3 were obtained for other types of ocular movement (blinks,vertical eye movement and diagonal eye movement).

(b) Convergence of the Parameter Estimates

To study the behaviour of the recursive algorithm between samples and its convergence properties, plots of the variation with time of the parameter estimates, the sample variance S^2 and the multiple correlation coefficient R^2 [21] were obtained and studied. The sample variance and multiple correlation coefficient were obtained with the following approximate recursive equations (see appendix A7 for more details):

$$S^2(m+1) = SS(m+1)/[asL(m+1)-n] \ , \ asL(m+1) > n \qquad (4.17a)$$

$$R^2(m+1) = SS(m+1)/SS_T(m+1) \qquad (4.17b)$$

where SS(m+1) and $SS_T(m+1)$ are, respectively, the sum of squares of the residuals and the total sum of squares and given by:

$$SS(m+1) = \gamma SS(m) + e^T(m+1)e(m+1) \qquad (4.18a)$$

$$SS_T(m+1) = \gamma SS_T(m) + y^T(m+1)y(m+1) \qquad (4.18b)$$

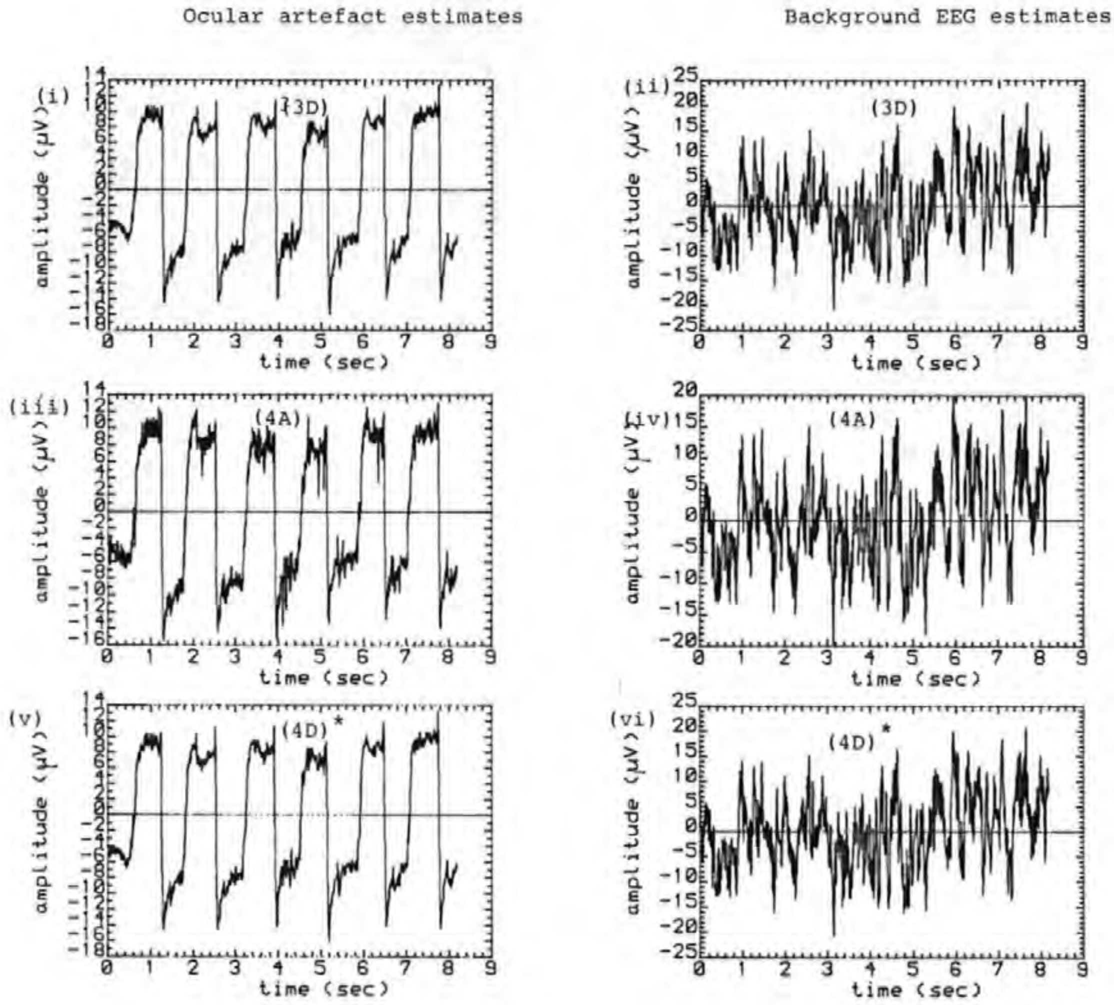


Figure 4.2

Estimates of the ocular artefacts and the background EEG by the conventional batch method. (i), (iii), and (v) ocular artefact estimates using models 3D,4A and 4D*, respectively. (ii), (iv), and (vi) the corresponding estimates of the background EEGs.

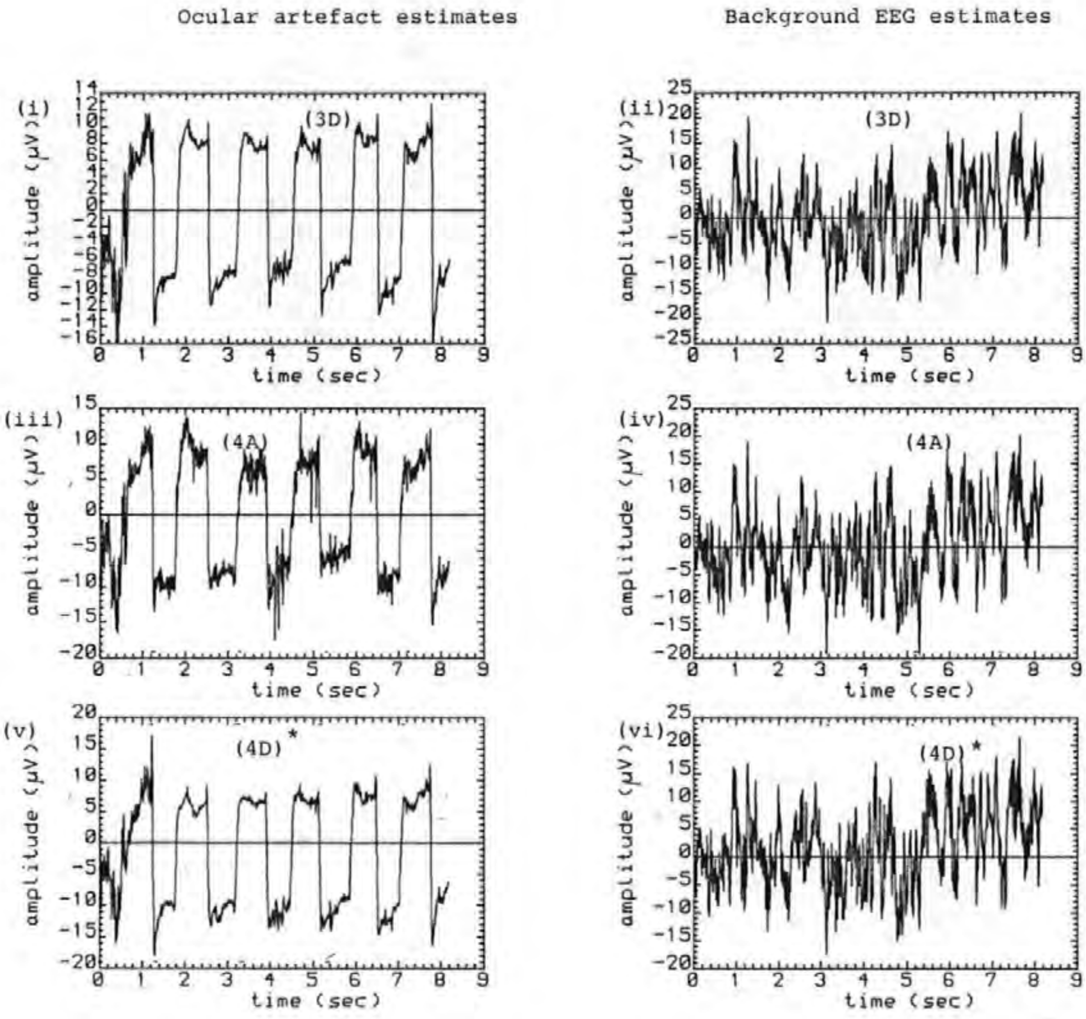


Figure 4.3
Estimates of the ocular artefacts and the background EEG by the UD recursive algorithm. (δ =0.05, γ =1.0, $\hat{\theta}_0$ = 0.1)
Other details are the same as for figure 4.2.

	Model	$\hat{\theta}_1$	$\hat{\theta}_2$	$\hat{\theta}_3$	$\hat{\theta}_4$
UD	3D	0.091	-0.108	-0.177	+
	4A	0.310	-0.053	-0.169	-0.216
	4D*	0.090	-0.109	-0.178	0.000
Batch	3D	0.091	-0.108	-0.178	+
	4A	0.312	-0.053	-0.169	-0.218
	4D*	0.090	-0.109	-0.176	0.000

+ not applicable

Table 4.1 Comparison of the parameter estimates for the batch and the UD algorithm. The estimates for the UD algorithm are the values at the end of the iterations.

n is the number of parameters in the model and asL(m+1) is given by:

$$asL(m+1) = 1 + \gamma asL(m) = \sum_{j=0}^{m} \gamma^{j}$$

Figures 4.4(i), (iii) and (v) illustrate the variation of the parameter estimates with time for the U-D algorithm. It is seen that there are initial transients as a result of guessed initial values for U,D and $\hat{\theta}$, after which they appeared to converge to some value about which they varied slowly. A comparison of figures 4.4(i) and (iii) showed that there were large changes in the parameter estimate, $\hat{\theta}_1$, when VL is included in the model. (in figures 4.1(iii), VL and VR correspond to $\hat{\theta}_4$ and $\hat{\theta}_1$,

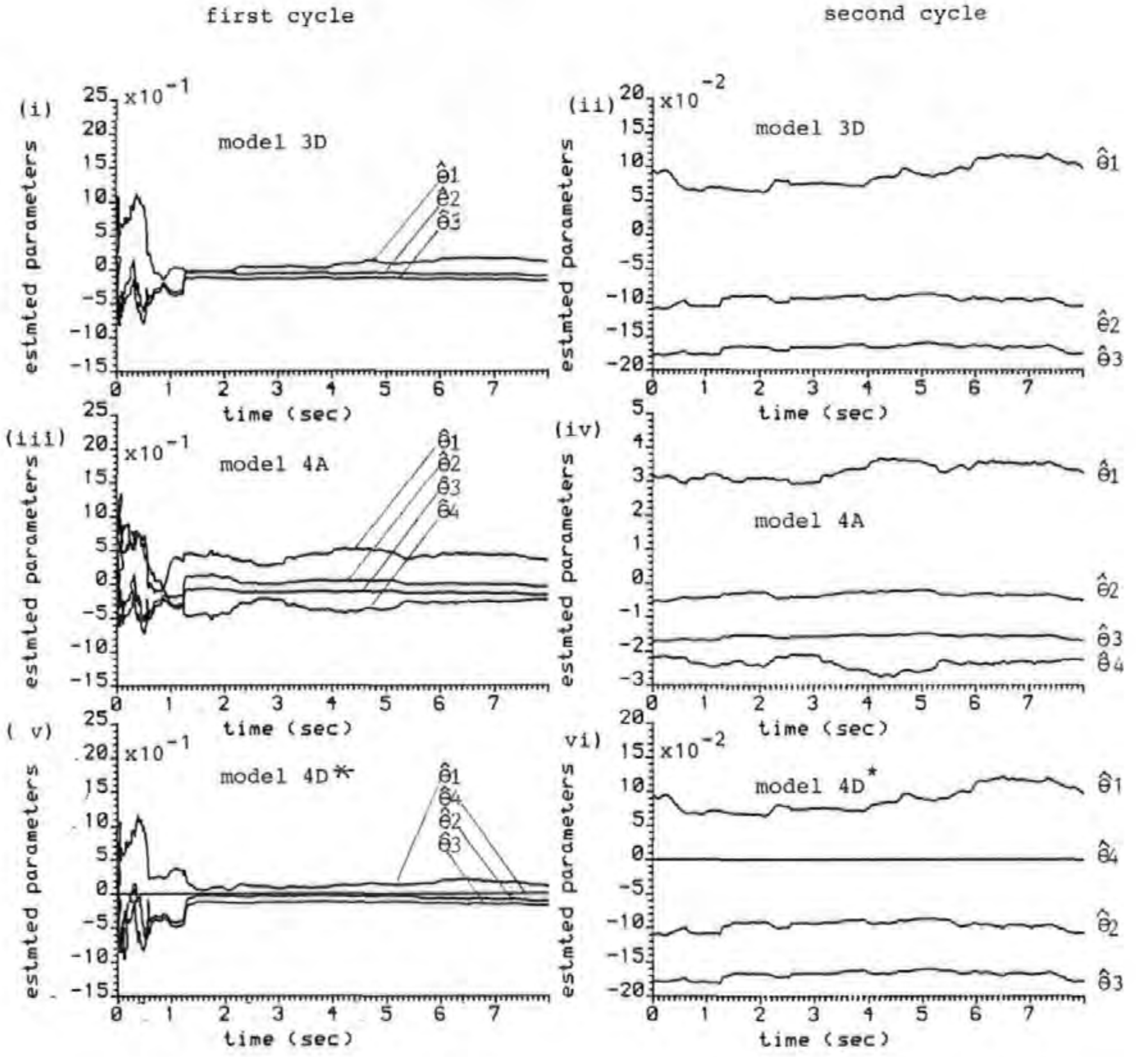


Figure 4.4 The evolution of the parameter estimates. (i), (iii), and (v) estimates for models 3D, 4A, and 4D* respectively, for the first cycle. (ii),(iv), and (vi) the corresponding estimates for the second cycle.

respectively). In fact, it was found that when only VL or VR was in the model the variations in the parameter estimates were much less than when both variables were in the model (cf figures 4(i) and (iii)). As discussed in reference [24], these large changes in the parameter estimates when a variable is added or dropped from a model are due to collinearity. Thus when there is collinearity, a given parameter estimate is determined by not only the corresponding EOG but also by the other EOGs which are related to the corresponding EOG. These points and other effects of collinearity are discussed further in appendix A2. It should be pointed out that the large variations in $\hat{\theta}_1$ and $\hat{\theta}_4$ in figure 4.4(iii) were probably due in part to the relatively large amount of noise unrelated to ocular movement in the corresponding EOGs (see figure 4.1(i) and (ii)). The large variations in the parameter estimates were observed mostly in horizontal eye movement records, but sometimes they were observed in the other types of ocular movements also. These effects suggested that it is undesirable to include all the EOGs, especially the vertical ones, in the model, and this agrees with the previous findings in chapter 2.

The variation with time of the estimates of the residual variance S^2 and the multiple correlation coefficient, R^2, are depicted in figure 4.5 (i) and (iii). It is seen that both S^2 and R^2 increase with time, rapidly at first and then slowly. This is an indication that these quantities were converging.

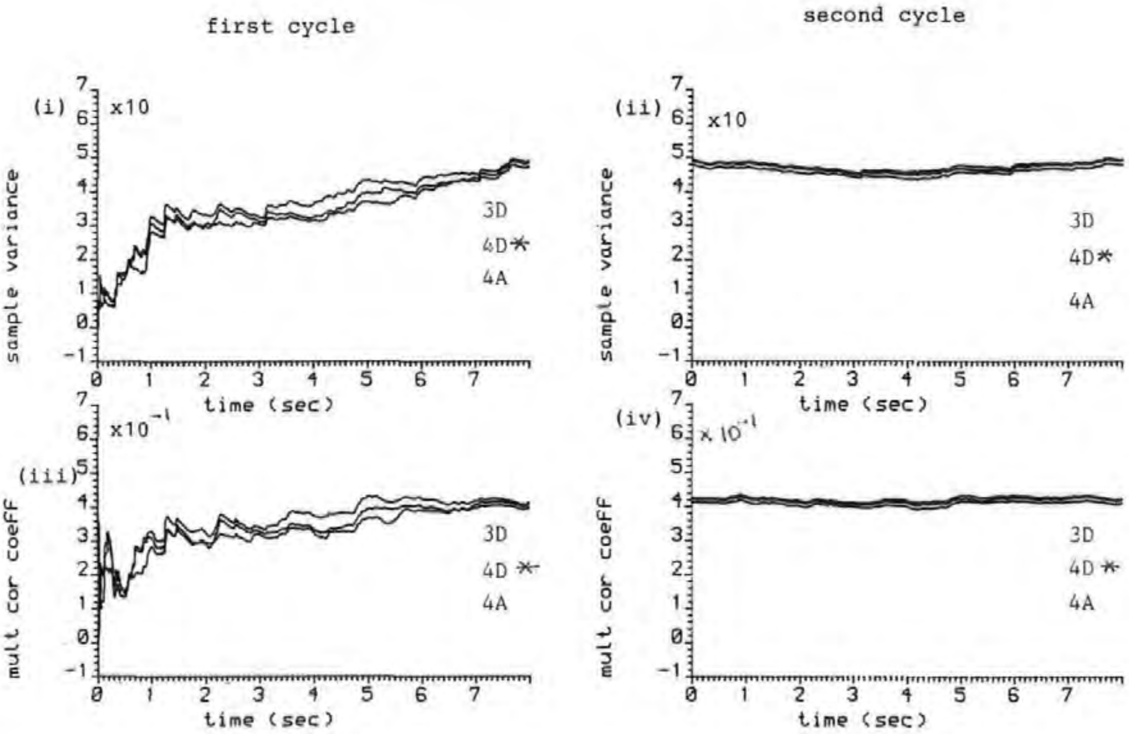


Figure 4.5 The evolution of the sample variance and multiple correlation coefficient for the first and second cycles.(i) and (iii) estimates of the sample variance and multiple correlation coefficient, respectively for the first cycle. (ii) and (iv) the corresponding estimates for the second cycle.

No of cycles	Model	Parameter estimates $\hat{\theta}_1$	$\hat{\theta}_2$	$\hat{\theta}_3$	$\hat{\theta}_4$	S^2	R^2
1	3D	0.091	-0.108	-0.177	+	49.22	0.42
	4A	0.310	-0.053	-0.169	-0.216	47.75	0.41
	4D*	0.090	-0.109	-0.178	0.000	48.94	0.42
2	3D	0.090	-0.106	-0.176	+	49.88	0.43
	4A	0.311	-0.053	-0.169	-0.217	48.20	0.42
	4D*	0.090	-0.109	-0.178	0.000	49.64	0.43
3	3D	0.09	-0.107	-0.176	+	49.95	0.43
	4A	0.312	-0.053	-0.178	-0.218	48.37	0.42
	4D*	0.090	-0.109	-0.178		49.64	0.43
Batch	3D	0.091	-0.108	-0.178	+	50.34	0.56
	4A	0.312	-0.053	-0.169	-0.218	49.02	0.41
	4D*	0.090	-0.109	-0.176	0.000	50.21	0.56

+ not applicable

Table 4.2 Final values of the estimates of the parameter the sample variance (S^2), and the multiple correlation coefficient (R^2)

To simulate longer lengths of data and to verify whether the parameter estimates have converged, the operation was recycled for the U-D algorithm. That is, at the end of the data the algorithm was resumed at the begining of the data without reassigning starting values for U,D and $\hat{\theta}$. Table 4.2 compares the final values of the parameter estimates, the sample variance and the multiple correlation coefficient for the first, second and third cycles with those of the batch method. It is evident that the quantities have converged by the end of the first cycle. It is noteworthy that the final values of both the variance and the multiple correlation coefficients are reasonably close to those of the batch method, thus demonstrating that equations 4.17a and 4.17b are good estimates of these quantities. The figures on the right hand side of figures 4.4 and 4.5 give the parameter estimates, the sample variance and the multiple correlation coefficient for the second cycle. They confirm that the algorithm has converged. The estimates of the ocular artefact and the background EEG in this case were very similar to the results for the batch method.

Results very similar to those of the UD algorithm were obtained for the recursive least squares algorithm (equation 4.4) and the square root algorithm. The notable difference between them was that they always produced different initial transients due to the starting values for P(0) (or S(0) for the square root algorithm) and $\hat{\theta}(0)$.

4.3 Recursive Generalized Least Squares (GLS) Algorithm

A problem in the ordinary least squares (OLS) method that was discussed in chapter 2 for the off-line case was correlation in the error terms, e(i). From section 2.3.1, the autoregressive model for the error term and the resulting GLS equation are:

$$e(i) = \sum_j \Phi_j e(i-j) + a(i) \qquad (4.19a)$$

$$\bar{y}(i) = \sum_k \bar{x}_k(i)\theta_k + a(i) \qquad (4.19b)$$

where, $\bar{y}(i) = y(i) - \sum_j \Phi_j y(i-j)$

$\bar{x}_k(i) = x_k(i) - \sum_j \Phi_j x_k(i-j)$

As observed in chapter 2, equations (4.19a) and (4.19b) have the same form as the OA model. Thus the recursive GLS for the parameter estimates, $\bar{\theta}$, and the covariance matrix, P, have the same form as those for the RLS algorithm given in equation (4.4a). Thus,

$$\hat{\bar{\theta}}(m+1) = \hat{\bar{\theta}}(m) + \bar{P}(m+1)\bar{x}(m+1)[\bar{y}(m+1) - \bar{x}^T(m+1)\hat{\bar{\theta}}(m)\ ] \qquad (4.20a)$$

$$\bar{P}(m+1) = \frac{1}{\Upsilon}[\bar{P}(m) - \frac{1}{\bar{\alpha}}\bar{P}(m)\bar{x}(m+1)\bar{x}^T(m+1)\bar{P}(m)] \qquad (42.0b)$$

where,

$$\bar{\alpha} = \Upsilon + \bar{x}^T(m+1)\bar{P}(m)\bar{x}(m+1)$$

$$\bar{x}^T(m+1) = [\bar{x}_1(m+1) \quad \bar{x}_2(m+1) \quad \ldots \quad \bar{x}_p(m+1)]$$

where the bar is used to distinguish the GLS terms from those of the RLS.

The autoregressive parameter, Φ_j , which is required to obtain $\bar{x}^T$ and $\bar{y}$ also has the usual RLS form:

$$\hat{\Phi}(m+1) = \hat{\Phi}(m) + Q(m+1)W(m+1)[\hat{e}(m+1) - W^T(m+1)\hat{\Phi}(m)] \qquad (4.21a)$$

$$Q(m+1) = \frac{1}{\Upsilon}[Q(m) - \frac{1}{\beta}Q(m)W(m+1)W^T(m+1)Q(m)] \qquad (4.21b)$$

where

$$\beta = \Upsilon + W^T(m+1)Q(m)W(m+1)$$

$$W(m+1) = [\hat{e}(m) \quad \hat{e}(m-1) \quad \ldots \hat{e}(m-n+1)]^T$$

$$\hat{e}(m+1) = y(m+1) - \sum_K \hat{\theta}_K x_K(m+1)$$

As in the off-line case, $\hat{\Phi}$ may be obtained iteratively. In fact, at each sample instant, a similar algorithm to that used for the off-line case is applied to the GLS. To start the algorithm, initial values are assigned to $\hat{\theta}(0)$, $P(0)$, $\hat{\Phi}(0)$ and $Q(0)$ in the manner described in section 4.1.1. Then at each subsequent sample instant when a new set of data is acquired, the following algorithm is performed (the symbol @ is used to indicate the relevant equations)

1. Set $\hat{\Phi}(m+1)=\hat{\Phi}(m)$
2. Obtain the variables $\overline{x}_K(i),\overline{y}(i)$ @(4.19)
3. Update $p(m+1)$ and $\overline{\theta}(m+1)$ using @(4.20)
4. Estimate the error term, $\hat{e}(m+1)=y(m+1)-x^T(m+1)\hat{\overline{\theta}}(m+1)$
5. Obtain improved estimates of $\Phi(m+1),Q(m+1)$ @(4.21)
6. Obtain $\overline{x}_k(i)$, $\overline{y}(i)$ @(4.19)
7. Obtain improved estimates of $\hat{\overline{\theta}}(m+1),P(m+1)$ @(4.20)
8. Test for convergence and go to step 4 if not satisfied. When the algorithm converges, the latest $\hat{\overline{\theta}}(m+1)$, $\overline{P}(m+1)$, $\hat{\Phi}(m+1)$ are the desired updates.

To ensure that the algorithm does not go on indefinitely at some sample instant, a limit is imposed on the maximum number of iterations allowable for each sample point or update based on previous experience. To avoid numerical problems of the kind associated with the RLS algorithm discussed earlier, the update of $P(m+1)$ and $Q(m+1)$ in steps 3, 5, and 7, is made via one of the factorization algorithms described earlier. Of course, if AR(1) model is used $\hat{\Phi}$ and Q are scalars and their update is greatly simplified and can be made directly. It should be pointed out that the recursive GLS algorithm suggested above has not been verified in practice.

4.4 Fixed Parameter Models

A disadvantage of all the recursive algorithms discussed above is that where there are many EEG channels to correct for OA (e.g 16 channels), the number of arithmetic operations between samples becomes so large that they cannot be implemented entirely in software on a general purpose microcomputer. Possible solutions include, reduction in the number of EEG channels, reduction in the number of parameters in the model (to say 2), and the reduction in the sampling frequency to give more time for calculations between samples. If none of these is desired then hardware floating point arithmetic devices may be required, as discussed in chapters 7 and 8.

However, if only predominantly one type of OA is expected so that the OA parameters, θ, may be assumed constant and subjects are able to co-operate, then an automated form of the present OA removal method may give satisfactory OA removal (see section 1.4), and in this case no special arithmetic device is necessary. An approach that can be used is proposed here.

4.4.1 Algorithm

1. Calibration Stage

(a) Acquire M samples (e.g. M = 250) from each EOG and EEG channel, whilst subjects are performing purposeful ocular movements.

(b) Use samples to compute the values for $\hat{\theta}$ for each EEG channel:

$$\hat{\theta} = [X_m^T Y_m]^{-1} X_m^T Y_m$$

where the matrices X_m, Y_m are as defined in section 4.2.1

2. Removal Stage

Use $\hat{\theta}$ obtained in 1(b) to correct subsequent EEG samples for artefact on-line:

$$\hat{e}(i) = y(i) - x^T(i)\hat{\theta} \qquad i=1,2 \ldots \infty$$

Remarks

(i) As the number of arithmetic operations is small, the above algorithm can be carried out entirely by software.

(ii) If desired, $\hat{\theta}$ may be obtained for each type of ocular movements, e.g blinks and vertical eye movement, and a criterion for distinguishing between vertical eye movement and blink [22] used, so that when blink occurs θblink is used and when ordinary eye movement occurs θvem is used.

(iii) In the previously reported on-line methods [4,5,6], 'estimates' of θ are obtained whilst subjects are performing purposeful ocular movements by adjusting one or more potentiometers until no vestige of the EOG can be seen in the EEG (see section 1.4). This can be time consuming and requires a subjective judgement and familiarity with the instrument by the operator. The calibration method suggested above, which is essentially an off-line technique similar to those discussed in chapter 2 and practiced by Jervis et al [2], overcomes these disadvantages. However, the removal stage for both methods are equivalent, since both assume that θ is time-invariant. However, this assumption may be untenable, so that the fixed parameter approach will be inferior to the 'full' or adaptive on-line methods described in the preceding sections of this chapter.

4.5 Summary

In this chapter efficient recursive least squares (RLS) algorithms suitable for removing OA on-line were developed. Two main problems that beset the RLS algorithm, namely, autocorrelation and numerical instability, and ways of solving these were discussed in some depth. For autocorrelation, the generalized least squares method was suggested, and for stability, the two factorization algorithms with excellent properties, viz, the square root and U-D algorithms were used. These numerically stable algorithms were shown to give similar results to their off-line equivalents. Compared to the present on-line methods, this approach is superior, requiring no subjective manual adjustment and does not assume that the OA parameters are constant. An automated form of the present on-line method is also proposed as an alternative method of on-line removal of OA where it is not desired to use fast but expensive hardware arithmetic devices, subjects are able to cooperate, and less than optimum performance is acceptable.

References for Chapter 4

[1] JERVIS B.W.,NICHOLS M.J.,ALLEN E.,HUDSON N.R. and JOHNSON T.E.
The Quantitative Assessment of Electroencephalograms Corrected for Eye Movement Artefacts. First European Conference on Signal Processing,Lausanne, Switzerland, 1980.

[2] JERVIS B.W.,NICHOLS M.J.,ALLEN E.HUDSON N.R. and JOHNSON T.E.
Comparison of two Methods for Removing Eye Movement Artefact from the EEG. (submitted to Electroenceph. and Clin. Neurophysiol.)

[3] GOTMAN J.,GLOOR P. and RAY W.F.
A Quantitative Comparison of Traditional Reading of the EEG and Interpretation of Computer Extracted Features in Patients with Supertentorial Brain Lesions. Electroenceph. Clin. Neurophysiol.,1975,38,PP. 623-639.

[4] BARLOW J.S. and RÉMOND A.
Eye Movement Artifact Nulling in EEGs by Multichannel on-line EOG Subtraction. Electroenceph. and Clin. Neurophysiol.,1981,55,pp. 418-423.

[5] GIRTON D.G. and KAMIYA J.
A simple On-line Technique for Removing Eye Movement Artifacts from the EEG. Electroenceph. Clin.

Neurophysiol., 1973,34,pp. 212-216.

[6] McCALLUM W.C. and WALTER W.G.
The Effects of Attention and Distraction on the Contingent Negative Variation in Normal and Neurotic Subjects. Electroenceph. Clin. Neurophysiol.,1968,25,pp. 319-329.

[7] QUILTER P.M., MACGILLVRAY B.B. and WADBROOK D.G.
The Removal of Eye Movement Artefact from EEG Signals using Correlation Techniques. Random Signals Analysis, IEE Conference Publication, No. 159, 1977, pp93-100.

[8] IFEACHOR E.C.,JERVIS B.W.,MORRIS E.L.,ALLEN E. and HUDSON,N.R.
A NEW ON-LINE METHOD FOR REMOVING OCULAR ARTEFACTS FROM EEG SIGNALS (to be published)

[9] PETERKA C.
A Square Root Filter for Real-Time Multivariate Regression. Kybernetika,1975,11,pp. 53-67.

[10] MORRIS E.L. and ABAZA B.A.
Adaptive Control of Steam Turbine. Proc. IEE,1976, 123,pp. 549-553.

[11] YOUNG P.
Recursive Approaches to time Series Analysis. Bull. I.M.A.,1974,10,pp. 209-224.

[12] CLARKE D.W.
Implementation of Self-Tuning Controllers,in HARRIS,C.J. and BILLINGS,S.A. (Eds.) Self-Tuning and Adaptive Control. Peter Peregrinus,1981,pp. 36-71.

[13] HSIA T.C.
System Identification. Lexington Books,Mass.,1977.

[14] EYKHOFF P.
System Identification John Wiley,1974.

[15] BIERMAN G.J.
Measurement Updating Using the U-D Factorization. Automatica,1976,12,pp. 375-382.

[16] THORNTON C.L. and BIERMAN G.J.
Filtering and Error Analysis via the UDU Covariance Factorization. IEEE Trans. on Automatic Control, 1978,23,pp. 901-907.

[17] THORNTON C.L. and BIERMAN G.J.
Givens Transformation Techniques for Kalman Filtering Acta Automatica,1977,4,pp. 847-863.

[18] CARLSON N.A.
Fast Triangular Formulation of teh Square Root Filter. AIAA JNL,1973,11,pp. 1259-1265.

[19] HSIA T.C.

On Least Squares Algorithms for System Parameter Identification. IEEE Trans. Automatic Control,Feb. 1976, PP. 104-108.

[20] CLARKE D.W.
Generalized-Least-Squares Estimation of Parameters of a Dynamic Model. IFAC Symposium- Identification in Automatic Control Systems,1967.

[21] JOHNSTON J.
Econometric Methods. McGraw-Hill,1972.

[22] GRATTON G., COLES M.G.H. and DONCHIN E.
A new method for Off-line Removal of Ocular Artifacts. Electroenceph. and Clin. Neurophysiol., 1983, 55, pp. 468-484.

[23] SINHA N.K. and SEN M.S.
Critical Evaluation of On-line Identification Methods. Proc. IEE,1975,122,pp. 1153-1158.

[24] CHATTERJEE S. and PRICE B.
Regression Analysis By Example. John Wiley, 1977.

Chapter 5 DESIGN AND DEVELOPMENT OF THE HARDWARE FOR THE ON-LINE OCULAR ARTEFACT REMOVAL SYSTEM

5.0 Introduction

In chapter 1, the need for on-line correction of EEG signals for OA was discussed. It was stated that the present on-line methods, however, are unsatisfactory and unwieldy for routine clinical use, among other disadvantages. In chapters 2 and 4 an approach to solving these problems was presented. In this chapter and the next, the design and development of an Ocular Artefact Removal (OAR) system which incorporates the work in those chapters are described.

First, target specifications for the system are set out. Next, a suitable system is described at system and block diagram level. The hardware needed to meet the specifcations and the detailed design of the system are then discussed.

5.1 Instrument Specification

5.1.1 General Specification

The OAR system should meet the following requirements:

(i) Be compatible with the standard EEG machines.

(ii) It should be able to acquire data from upto 16 and 4

EOG channels at 128 samples/sec. for each channel.

(iii) Provide a continuous real-time correction for OA in the EEG signals.

(iv) Output the corrected EEG and/or the uncorrected EEG/EOG to the EEG machine, to allow instant comparison of the corrected and uncorrected EEG.

(v) To avoid saturation, which would reduce the corrector's effectiveness, the system should have some autoranging facility.

(vi) The instrument should be suitable for use by unskilled persons.

In view of the requirements and the criteria set out above, and the desire to implement the U-D algorithm described in chapter 4, it was decided that the OAR system should be microprocessor-based. This decision was reinforced by the following advantages offered by the use of a microprocessor-based instrument.

(i) Software controlled design yields a very flexible system. Several OA removal algorithms and models can be implemented on one system, and the models used in any application specified by the user.

(ii) New models or ideas can be investigated by mere software modifications without having to build a new instrument. Thus a software-controlled OA removal system could be an excellent research aid.

(iii) A programmed instrument allows the provision of house-keeping routines for self-checking, automatic calibration, reduction in overload problems etc. Data processing on the removal system may include zero phase digital filtering of the EOGs to reduce the effects of secondary artefacts.

5.1.2 System Description

The block diagram of the OAR system is given in figure 5.1. Each EEG/EOG signal from the auxillary output of the EEG machine is first amplified and then bandlimited to 30 Hz by a low pass filter which feeds a sample and hold circuit. The twenty channels are then simultaneously sampled at the positive transition of the sampling signal,FS. Simultaneous sampling is employed to avoid the introduction of delays between corresponding time points. The negative transition of the signal, FS, also interrupts the processor, and signals the beginning of a cycle during which the twenty samples are sequentially selected by the multiplexer (MUX) and digitized by the analogue-to-digital converter (ADC) under the the control of the microprocessor (μP).

Some form of autoranging is provided by the programmable

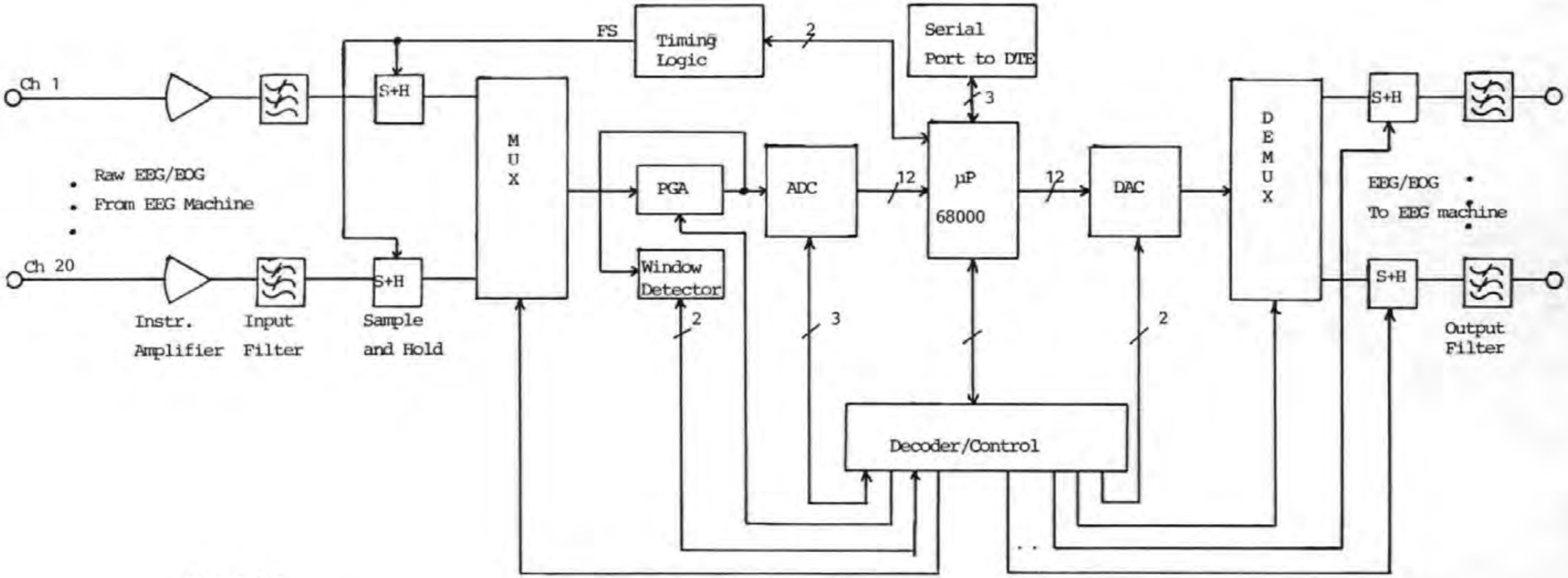


Figure 5.1

Block diagram of a microcomputer-based

Ocular Artefact Removal System.

gain amplifier (PGA) and the window detector. This is to extend the dynamic range of the ADC and to avoid saturating it. If a channel sample falls outside a predefined window (± 9.75V), the µP selects the lower gain of the PGA, which halves the sample level before initiating an ADC cycle. Account is taken of this before the digitized sample is saved in the memory.

The digitized samples are then processed by the OA removal algorithm to obtain the corrected EEGs. The corrected EEG samples, together with the raw EEG/EOG if desired, are output to the auxilliary inputs of the EEG machine via the digital-to-analogue converter (DAC) and the associated network.

5.1.3 Specifications and Choice of Hardware

(a) Microprocessor

It was realised at an early stage that to perform the complex tasks of processing the OA algorithm, albeit recast in an efficient form (see chapter 4), and control the entire structure require a very fast processor. Also it was desirable that the arithmetic operations be performed using floating point scheme (see section 6.3), which involves the manipulation of several bytes of data. In a time-critical system such as this these operations are best carried out using register-to-register operations to gain speed. The 68000 µP with internal 32-bit data size for registers is ideally suited for these operations, in addition to the fact that this processor was considered the best 16-bit procesor at the time .

The MVE110 single card Microcomputer was selected for this application, as it has essentially all the facilities that were required.

The MVE110 Single Card Microcomputer

The MVE110, a 68000 μP-based microcomputer, was chosen for the OAR system. It can function as either a single CPU element in a multi-processing VME-bus configuration, as a single CPU/controller in a VMEbus system or as a stand-alone microcomputer [1]. It has a number of ports which are available to the user. These include the VMEbus, which contains all the signal lines from the processor, and the serial port, which is used for communication with a data terminal equiptment (DTE). The MVE110 also comes with a monitor program which allows line-by-line assembly of programs and various debugging operations. It has four pairs of memory sockets which can be easily configured to accept ROM and/or RAM devices.

(b) Amplifiers, Filters and Sample and Holds

The amplifier, the filter and the sample and holds form the input signal conditioner [2,3].

(i) Amplifiers

The two main factors that were considered in selecting amplifiers were range of the amplitudes of the signals and noise [3]. The gain of the amplifier is chosen so that the smallest signal of interest can be detected. It is desirable for the amplifier to have low internal noise (~1μV) and high common mode rejection ratio (~100dB), so that low level signals in a noisy environment can be measured as faithfully as possible. The amplifier should also be able to amplify both differential and single-ended signals.

These requirements are easily met by a suitable instrumentation amplifier, which has become an industrial standard in medical instruments [4,5]. The amplifier chosen is the AD524. This device is a low drift, high linearity and low noise amplifier suitable for high speed data acquisition systems. It has the added feature of being programmable, which may become useful in the future. Additionally, the high input impedance of this device (~$10^9\Omega$) ensures that there is negligible loading of the EEG machine which has an output impedance of the order of 100ohms (see appendix 8 for more details).

(c) Input and Output Filters

The filters are 3rd order Butterworth active filters with a cut-off frequency of 30 Hz, in line with current practice in EEG work [4,7,8]. The Butterworth filter is chosen as it has a reasonably good amplitude and phase characteristics, and so would not alter significantly the signals amplitude and latency [4]. The use of a third order filter provides a loss of about 15 dB at half the sampling frequency, which is considered adequate to reduce aliasing. The input and output filters are identical.

(iii) Sample and Hold

The devices should have low droop, and in view of the number required should be cheap. The LF398 was chosen for these reasons. The sampling frequency chosen is 128 Hz, in line with current practice in EEG work [7,8]. A higher sampling rate would put severe restrictions on real-time computations.

(c) Multiplexer/Demultiplxer

The multiplexer and the demultiplexers are required to allow resource-sharing by the input and output channels. Thus, they allow each channel to be digitized or converted to analogue using only one ADC and one DAC. Analogue devices AD7501/06, with excellent characteristics were used to perform both functions, as they are bidirectional switches.

An alternative approach would be to provide separate ADC and DAC for each channel [2,3,6]. This approach reduces to a minimum system noise due to cross-talk between channels, but was considered rather too expensive.

(d) Analogue-to-Digital Converter/Digital-to-Analogue Converter

In line with current practice in physiological work [4,5,7,8], the ADC/DAC have a resolution of 12 bits. In view of the number of channels that need to be converted, it is desirable for the ADC to have a conversion time of less than 40μS. For easy interfacing to the μP, it would also be desirable for both the ADC and DAC to be microprocessor-compatible. The ADC and DAC devices selected are the Analog Devices successive approximation converter AD574 and AD567, respectively. For the ADC, 12 bit conversion takes between 15 to 35 μS, and the DAC has a settling time of 500ns. Both devices meet easily the system requirements and are reasonably cheap. In the bipolar mode, the ADC can accept analogue signals in the range ±10V.

(e) Digital Integrated Circuits

All digital integrated circuits (IC) were TTL (LS versions) for speed and compatibility with other circuits (μP,ADC,DAC,MUX etc.).

(f) Window Detector

The LM311 was selected to perform window detection. This device has features that make it stand out from most other comparators, (e.g 710), and these include:

(i) Operation from standard supplies,e.g ±15V, unlike most comparators which operate from asymmetrical supplies (e.g 710).

(ii) Ability to perform a wired-OR fuction, and gives output that is TTL compatible.

5.2 System Design

The entire system can be divided broadly into six different circuits or subsystems, namely: The microcomputer which was bought off-the-shelf, the Analogue input circuit, the ADC circuit, the DAC circuit, the decoder, and the Analogue output circuit. These will now be described in turn (except the microcomputer which has already been described in section 5.1.3):

5.2.1 Analogue Input Circuit

The analogue input circuit is shown in figure 5.2. It consists of three sections; the amplifier, the filter and the sample and hold. The signal level at any point in the system should not exceed 12V peak to allow for the internal volt-drops and possible non-linearity in the amplifiers (± 15V supplies are used), and since the input signal to the system is expected to be in the range 1 to 2V peak the input system gain should not exceed 6. Thus, the gain is provided by the instrumentation amplifier, connected for a gain of 2, and the programmable gain amplifier (see later).

(a) Filter

The filter is required to have a cut-off frequency of 30Hz and a loss at 60Hz (twice the cutoff frequency) of 15dB so that n, the filter order, lies between 2 and 3 [9]. In view of the system gain requirements, the filter should

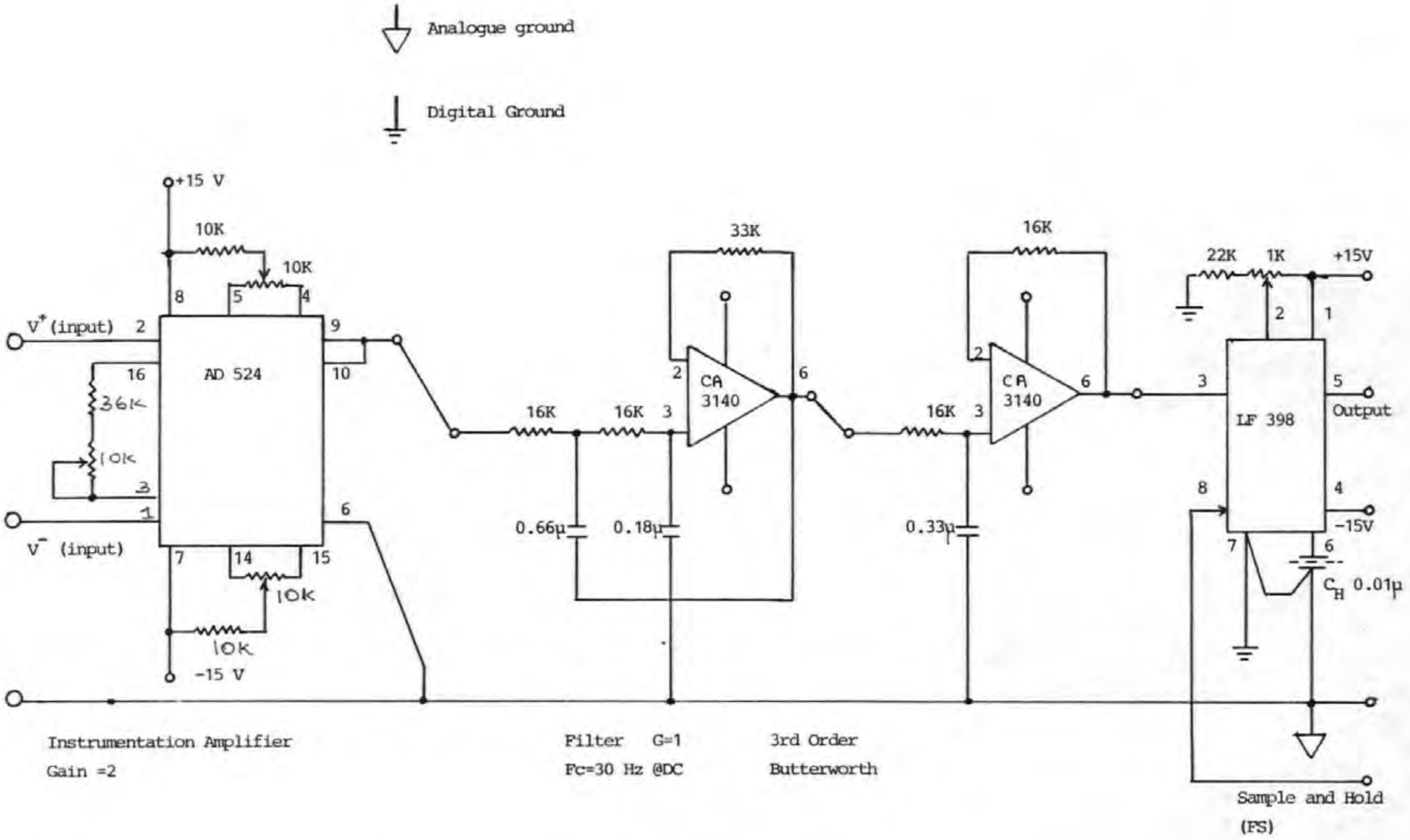


Figure 5.2 Analogue Input Circuit

have a gain of unity (at d.c.). A suitable configuration for these requirements [10] is given in figure 5.2. Following the design steps in [10], the given components yield the desired filter response.

(b) Sample and Hold

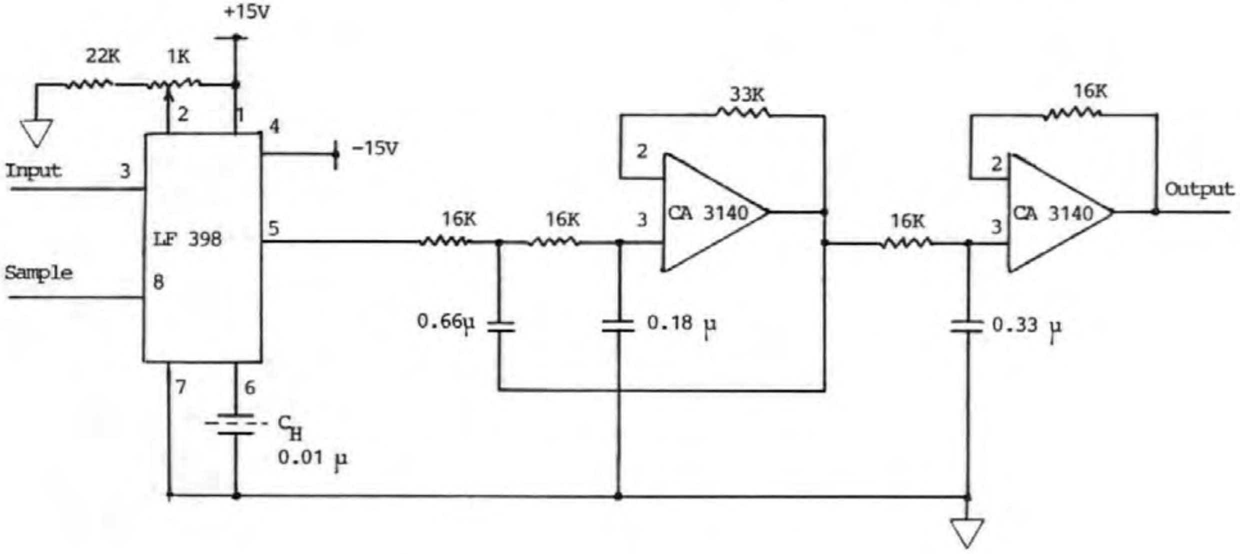
The sample and hold circuit connections are straight forward. It is necessary, however, to choose the hold capacitor carefully. The hold capacitor used is polypropylene, which has a low loss dielectric and suitable for continuous use, has excellent frequency performance and can withstand fast rise time pulses.

5.2.2 Analogue Output Circuit

The analogue output circuit consists of the sample and hold and the filter, which are identical to those of the analogue input circuit. See figure 5.3. Each sample and hold has a separate sampling signal line which is controlled via the decoder (see later). The sample and hold is used to hold the analogue sample until the next analogue sample is obtained. This stretches the sample pulses and increases the signal power, but introduces aperture distortion [17,18], which is considered small here. The filter reduces the unwanted sideband and sampling signal components at the output of the sample and hold.

Figure 5.3

Analogue Output Circuit



5.2.3 Analogue-to-Digital Converter Circuit

The ADC circuit consists of the multiplexer (MUX), the programmable gain amplifier, the window detector and the ADC as shown in figure 5.4. The 20-input to 1-output multiplexer is made up of one 16-channel MUX and one 8-channel MUX (there are four spare inputs).

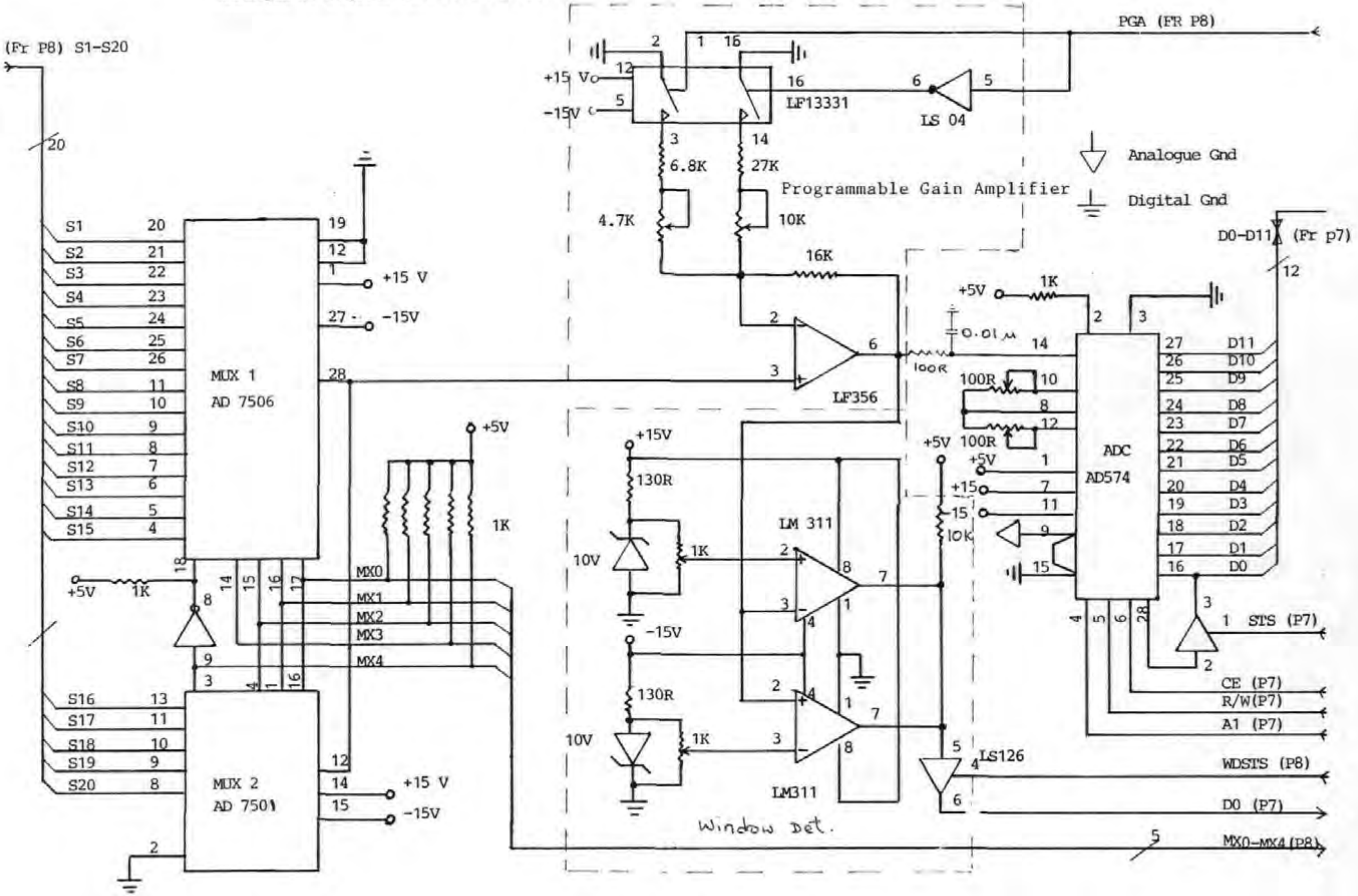
The MUX occupies a word in memory ($FFFF70), and channels are selected by writing to this location. The value of the word, which determines the logic levels of the control lines MX0-MX4, determines which channel is selected. To minimize crosstalk between channel selection, the MUX output is grounded after the selected channel sample has been converted to digital.

The output of the MUX is connected to the input of the programmable gain amplifier (PGA), which also acts as a buffer between the ADC and the MUX. The MUX should operate from a low impedance source, which is provided by the sample and hold, into a high impedance load to reduce loading which affects accuracy.

It is important to include a filter at the input of the ADC to reduce the error due to noise. The output of the MUX/PGA is broadband, so that it is necessary to limit the noise bandwidth to a value consistent with the response times of the multiplexer [17,18]. This can be

Figure 5.4

Analogue-to-Digital Converter Circuit



accomplished by a simple RC filter. The choice of R and C, however, is a compromise between improving signal-to-noise ratio and linearity. Additionally, only very little series resistance can be tolerated if loading of the ADC, which has an input impedance of only about 15KOhm, is to be avoided.

The PGA and the window detector are used to extend the dynamic range of the ADC. If the output of the PGA exceeds the range of the ADC, which is ±10V, there is saturation or overloading which would affect the removal algorithm. The window detector, consisting of the two comparators, LM311 , and the associated diode/resistor network, is used here to monitor the PGA output and to indicate to the controller (µP) before conversion when it exceeds the window limits. To allow for the effects of noise and power supply variations, the window limits are set at ±9.75V. If these limits are exceeded, the output of the detector goes low. On detection of a low level by the µP when the status of the detector is read, the gain of the PGA is set to a lower value. The PGA is normally in the higher gain which is twice the lower gain (gains of 3 and 1.5). Selection of the lower gain automatically halves the sample value and brings it to within the dynamic range of the ADC and thus avoids saturation of the ADC.

The ADC, AD574, is connected for bipolar operation and an input signal range of ±10V. The device has internal register control inputs that are used to control the output data format and conversion cycle. The data format input (pin 2) is hard-wired high so that all 12 bits of data can be read in one operation. Pin 4 controls the conversion length. If it is held low before conversion is initiated a full 12 bit cycle results (25µS), otherwise an 8 bit cycle is initiated. The two functions, start conversion and read data are controlled by the $R/\overline{W}$ line when the enable signal, CE, is at logic 1. To start conversion, $R/\overline{W}$ is at logic 0, and to read data it is at logic 1. The signal line, CE, is generated in the decoder and the $R/\overline{W}$ comes direct from the processor (via buffers). The status line (pin 28) goes high at the start of conversion to indicate that conversion is in progress and goes low when the cycle is completed. This line is connected to the data bus via the tri-state device controlled by the 'STS' line.

5.2.4 Digital-to-Analogue Converter Circuit

The DAC circuit consists of the 12-bit DAC, attenuator and the demultiplexer, figure 5.5. The DAC is connected in the transparent mode. Thus, writing a word to the DAC address ($FFFF68) automatically initiates a 12 bit digital-to-Analogue conversion. Only two control lines, $\overline{CS}$ and $R/\overline{W}$, are required in this case to control the DAC.

The output amplifier, IC 2, is connected to give a +5V span at its output corresponding to $000 and $FFF (Hex) at

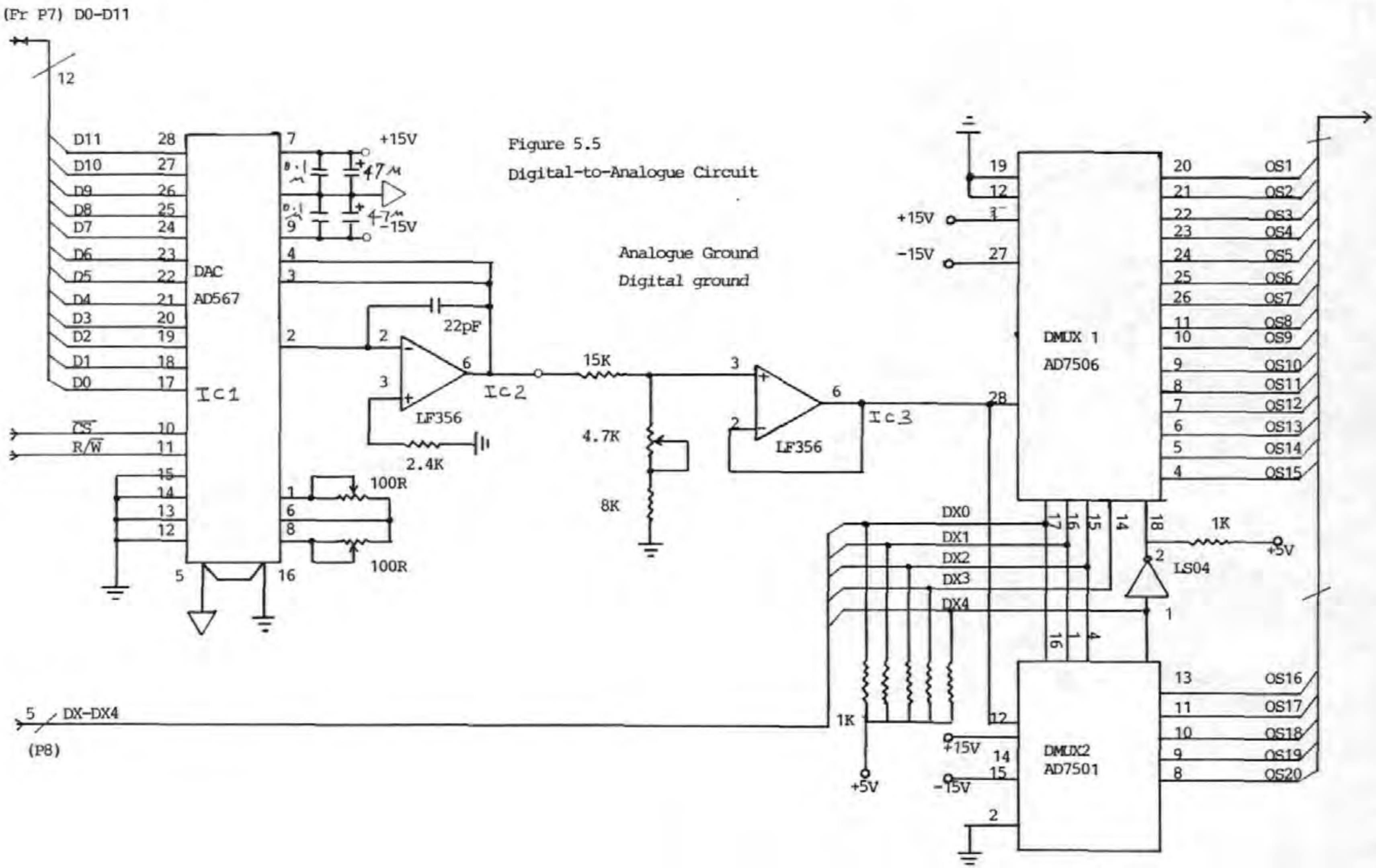


Figure 5.5
Digital-to-Analogue Circuit

the input of the DAC. To avoid saturation in the EEG machine, the output of the output amplifier is limited to a maximum of 2V peak by the attenuator (the potential divider preceding IC3). A buffer (IC3) between the attenuator and demultiplexer provides the necessary low impedance source. The demultiplexer performs the inverse operation to the multiplexer, so that under the control of the µP it connects the output of the buffer to one of 20 ouput circuits. It should be pointed out that it is possible to do without the demultiplexer [6]. In this case the output sample and holds would be used to perform the demultiplexing function as well as the function described in section 5.2.2.

Since the final destination of the demultiplexed signal is the EEG machine, and given the speed of the DAC (500 nS) the delay introduced between channel samples is insignificant. Thus, it was not considered necessary to use 'simultaneous sampling' as was the case at the input circuit. A way of implementing simultaneous sampling, if the cost of doing so is justifiable, is to use the per-channel DAC arrangement described in section 5.3

5.2.5 The Decoder/Control Interface

The circuit diagram of the decoder is given in figure 5.6. The circuit decodes and generates from the VMEbus the various control signals for the I/O devices. I/O devices connected to the VME bus via this interface can occupy the address range $FFFF00 to $ FFFF7F (128 bytes), as the I/O enable signal (I/O EN), (pin 10, IC2), is only valid when an address in this range is accessed.

Whenever the processor accesses any address in the range $FF0000 to $FFFFFF, the so-called VME short address range, an address modifier code is placed on the AM0 to AM4 lines of the VMEbus. When the processor is in the supervisory mode, as in this case, the code on the AM0-AM4 lines is $2D (Hex). By decoding these lines, the address lines A7 to A15 and the address strobe $\overline{AS}$ (using IC1 and 2), it is ensured that the I/O devices connected to the VMEbus respond only to accesses in the range $FFFF00 to $FFFF7F.

The device select IC (IC 8) is enabled when the (I/O EN) signal is high. The device selected then depends on the states of the address lines, A4 to A6, and in the case of MUX,DMUX,PGA, window detector and output sample and holds also on the states of A3,A2 and A1.

The ADC and DAC occupy, respectively, three and two words in memory, and either may be selected when the (I/OEN) line is high and the address lines (A6,A5,A4) are at (1,1,0). The ADC is selected when additionally A3 = 0, and the DAC when A3 =1. IC 18 is used to decode the address lines A3 and A2 to generate the DAC and ADC select and/or control signals. To start 12 bit AD conversion, a write to location $FFFF60 or $FFFF62 is performed. The converted sample is read from the same location. The

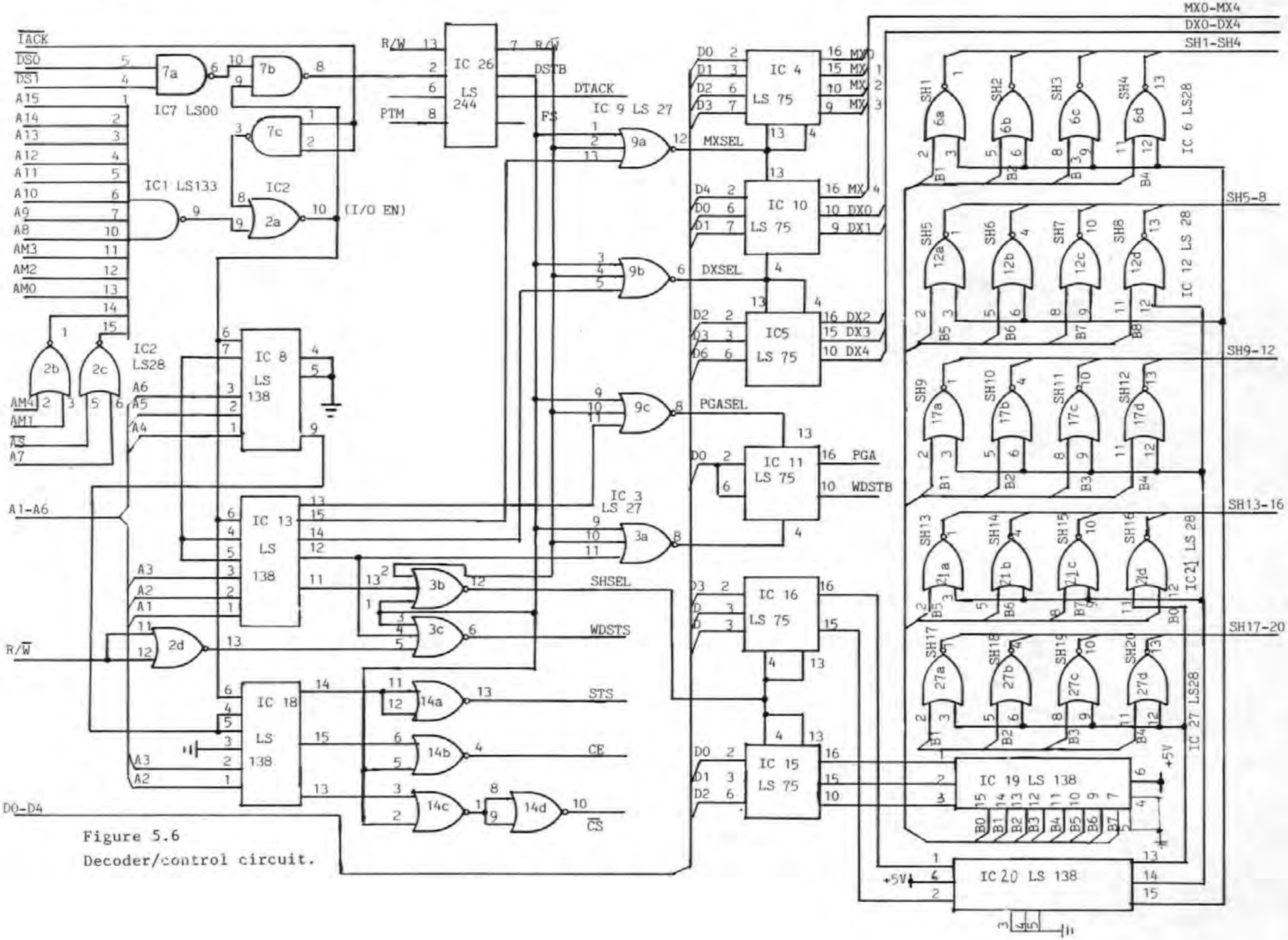


Figure 5.6
Decoder/control circuit.

status of the ADC can be read from location $FFFF64.

The DAC can be controlled from one of two locations, $FFFF68 and $FFFF6A. Writing to either of these locations loads the 12 bit data into the DAC and initiates conversion.

Writing to location $FFFF74 strobes the data on D0 to the PGA latch (IC11, pin 16) and hence to the PGA control line. The logic level of D0 determines which of the gains of the PGA is to be selected. Similarly, the window detector status can be read by reading from address $FFFF76 which causes WDSTS line (pin 6, IC3) to go high.

The output sample and holds require 20 control lines, SH1 to SH20 (see the right hand side of figure 5.6). One word is allocated to control these. Writing to this location ($FFFF78) causes SHSEL (pin 12, IC3b) to go high and strobes the data on data lines D0-D4 to the latches (IC15 and 16). The 5-line to 32-line decoder formed by IC6,12,17,19-21 and 27, then generates from the output latches the appropriate logic level to select the desired channel.

The control signals for the other devices are generated in a similar way as described above. The output equations for the various device select signals are given below, and the memory map for the I/O devices in table 5.1.

Table 5.1 Memory map for the I/O devices connected to the VMEbus.

Address	Device	Signal	operation
		mnemonic	
FFFF50-5F	FP chips (reserved).		
FFFF60-61	ADC	CE	start convsn.
FFFF62-63	"	"	read 12 bit data
FFFF64-65	"	STS	read status
FFFF68-69	DAC	$\overline{CS}$	load 12 bit data
FFFF6A-6B	"	$\overline{CS}$	"
FFFF70-71	MUX	MX0-4	select an i/p ch
FFFF72-73	DMUX	DX0-4	select an o/p ch
FFFF74-75	PGA	PGA	select gain
FFFF76-77	WDSTS	WDSTS	read WD status
FFFF78-79	S+HLD	SHSEL	select o/p s+hold.

$(I/O\ EN) . A6 . \overline{A5} . A4 = MPY/ALU$

$(I/O\ EN) . A6 . A5 . \overline{A4} . \overline{A3} . \overline{A2} = CE$

$(I/O\ EN) . A6 . A5 . \overline{A4} . A3 . \overline{A2} = \overline{CS}$

$(I/O\ EN) . A6 . A5 . A4 . \overline{A3} . \overline{A2} . \overline{A1} = MXSEL$

$(I/O\ EN) . A6 . A5 . A4 . \overline{A3} . \overline{A2} . A1 = DXSEL$

$(I/O\ EN) . A6 . A5 . A4 . \overline{A3} . A2 . \overline{A1} = PGASEL$

$(I/O\ EN) . A6 . A5 . A4 . \overline{A3} . A2 . A1 = WDSEL$

$(I/O\ EN) . A6 . A5 . A4 . A3 . \overline{A2} . \overline{A1} = SHSEL$

The control signals that go via latches are required to be stable throughout the period they are needed. In each case, a data strobe signal $\overline{DSTB}$, derived from the the data strobe lines of the processor, is used to strobe data into these latches.

The VME110 generates a bus time-out and aborts a data transfer cycle if an addressed I / O device does not acknowledge the data transfer within 200 µS. This is to prevent the system hanging.

Acknowledge is accomplished by asserting the $\overline{DTACK}$ line when the bidirectional lines, ($\overline{DS0}$,$\overline{DS1}$), that indicate a valid data transfer on the data bus are active low. For the purposes of terminating the bus cycle, the I/O devices are divided into two groups. The Floating point (FP) chips (see chapters 7 and 8), requiring no wait states [13,16] because they are very fast, belong to one group, and the rest of the I/O devices belong to the

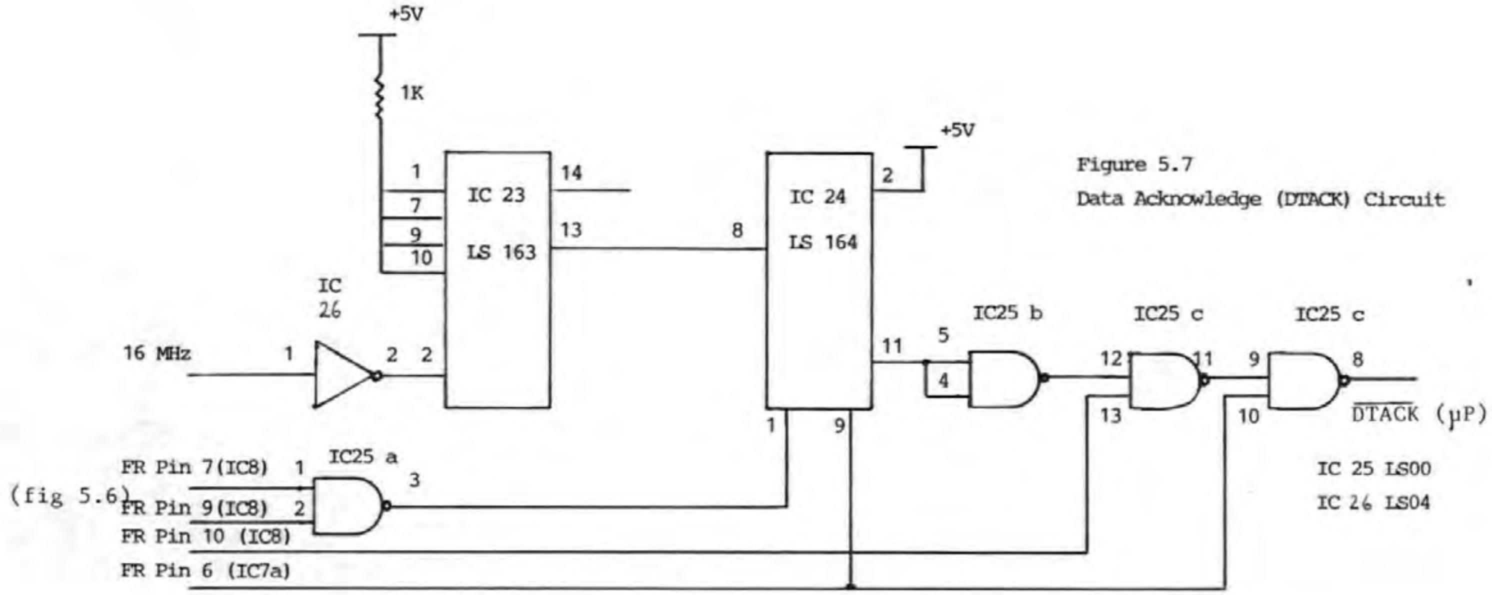


Figure 5.7
Data Acknowledge (DTACK) Circuit

other. For the group 1 devices, a valid $\overline{\text{DTACK}}$ is generated when any of the addresses assigned to the FP devices (see table 5.1) is accessed and the data is valid. For the group 2 devices, IC24 (figure 5.7), a clearable shift register, is used to lengthen the read or write cycles to 1.5µS to allow valid data to be presented or received from the data bus by the slowest device, which is the the multiplexer or demultiplexer

5.3 Hardware Construction

Plate 5.1 depicts the OAR system . The OAR system hardware was constructed on double Eurocard boards, dictated by the dimensions of the computer board. Initially, these boards were contructed with wire-wrapping techniques so that the designs could be modified if necessary, but later printed circuit boards (PCBs) were used for good finishing and to keep the system noise down. The boards plug onto DIN 41612 connectors mounted on an equipment rack and interconnected at the backplane with wire-wraps. The details of the backplane and Eurocard connectors for the system are given in appendix A8.

The six analogue input circuits (section 5.2.1) were constructed on one board and the corresponding analogue circuits on another. (Infact at the present, the six input circuits were constructed on two P.C.Bs which were then mounted on another board containing the appropriate DIN connector. The P.C.Bs were originally intended to be

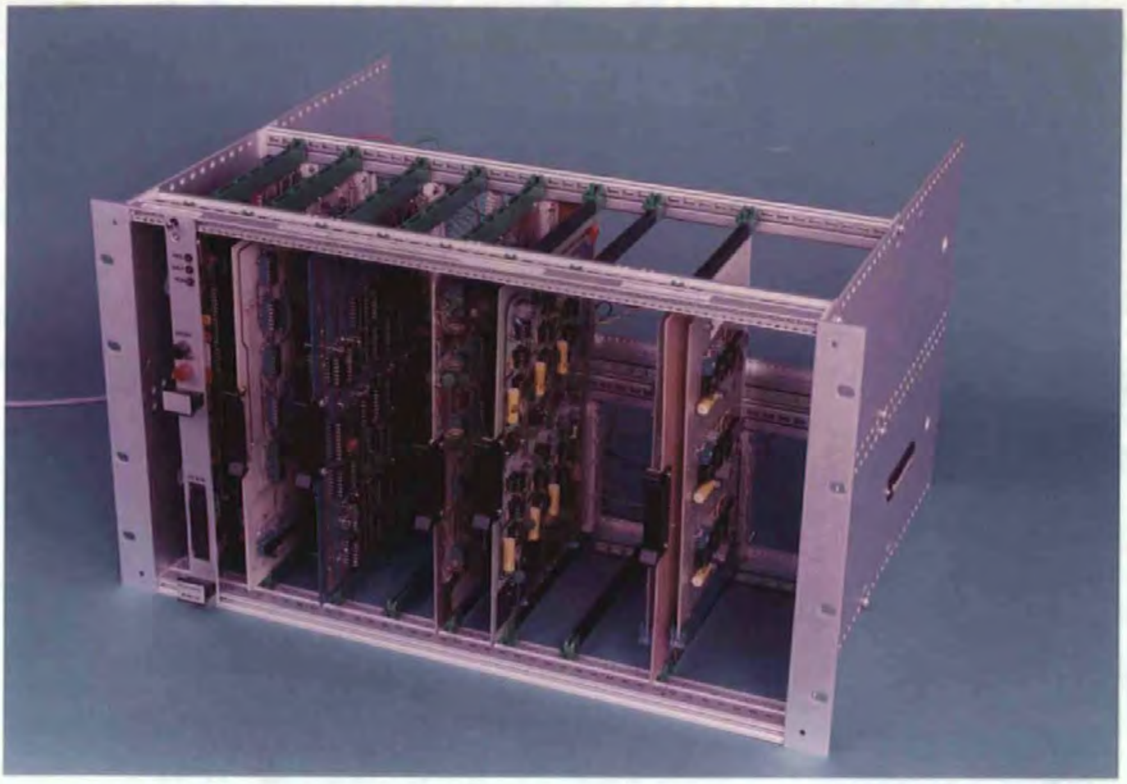


Plate 5.1 The OAR system

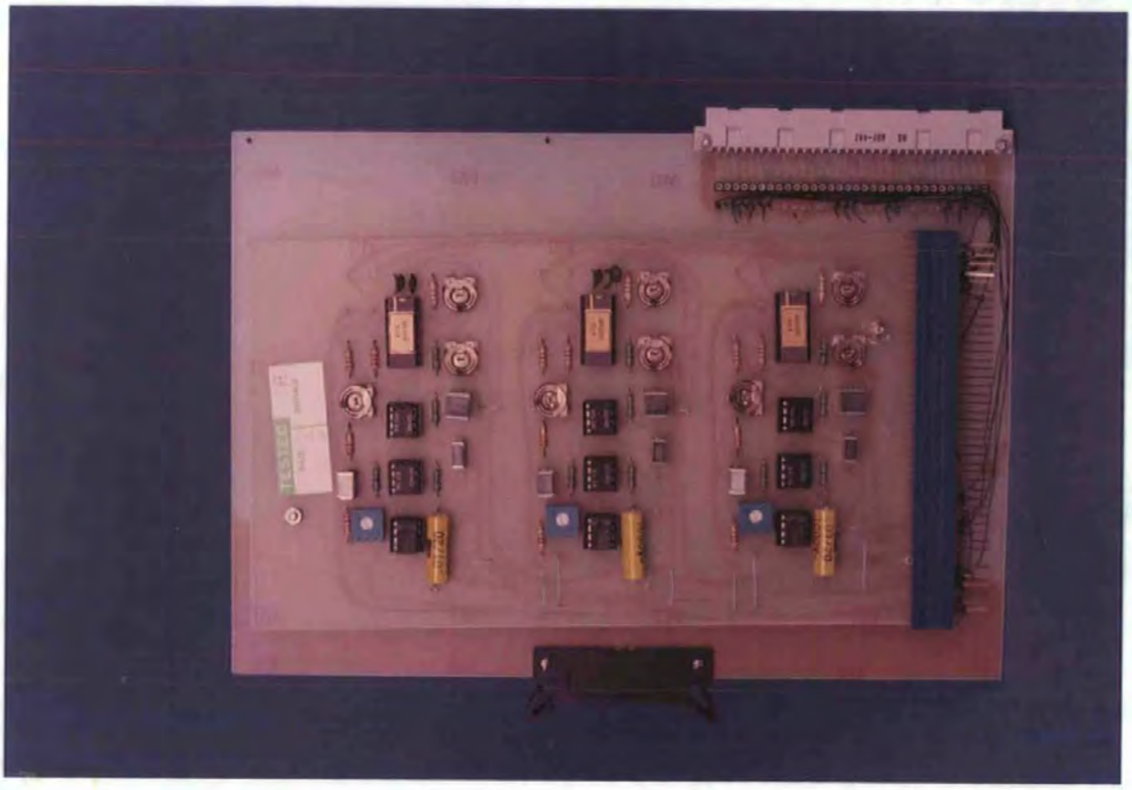


Plate 5.2 The OAR system Analogue input board.

mounted in a smaller equipment rack). Plates 5.2 and 5.3 show these boards. To minimize digital-feedthrough from fast logic signals into analogue signals, Logic lines to the sample and holds were kept as far as possible from the analogue input and output lines and by using guard tracks [20]. The hold capacitors were guarded by tracks connected to the output [20]. Where possible, ground planes were used.

The ADC and DAC circuits (figures 5.4 and 5.5) were contructed on one board as shown in plate 5.4. The board supply voltages were suitably decoupled at the edges of the board and near the supply pins of the various integrated circuits. Each of the analogue supply pins of the ADC and DAC was decoupled directly to analogue common with a 47μF tantallum capacitor in parallel with a 0.1μF disc ceramic capacitor. The +5V supply pin of the ADC was decoupled directly to digital ground. Each of the other ICs on board was decoupled to digital ground with a 0.1 μF disc ceramic capacitor. The ADC and associated analogue circuitry were located as far as possible from logic circuitry. Special note should be made of the ground geometry. Two separate grounds, the analogue ground and the digital ground, were used. The only connections between the two grounds were at the analogue ground pins of the ADC and DAC.

The PCB for the decoder is given in plate 5.5 and follows similar considerations as above. The PCB construction and

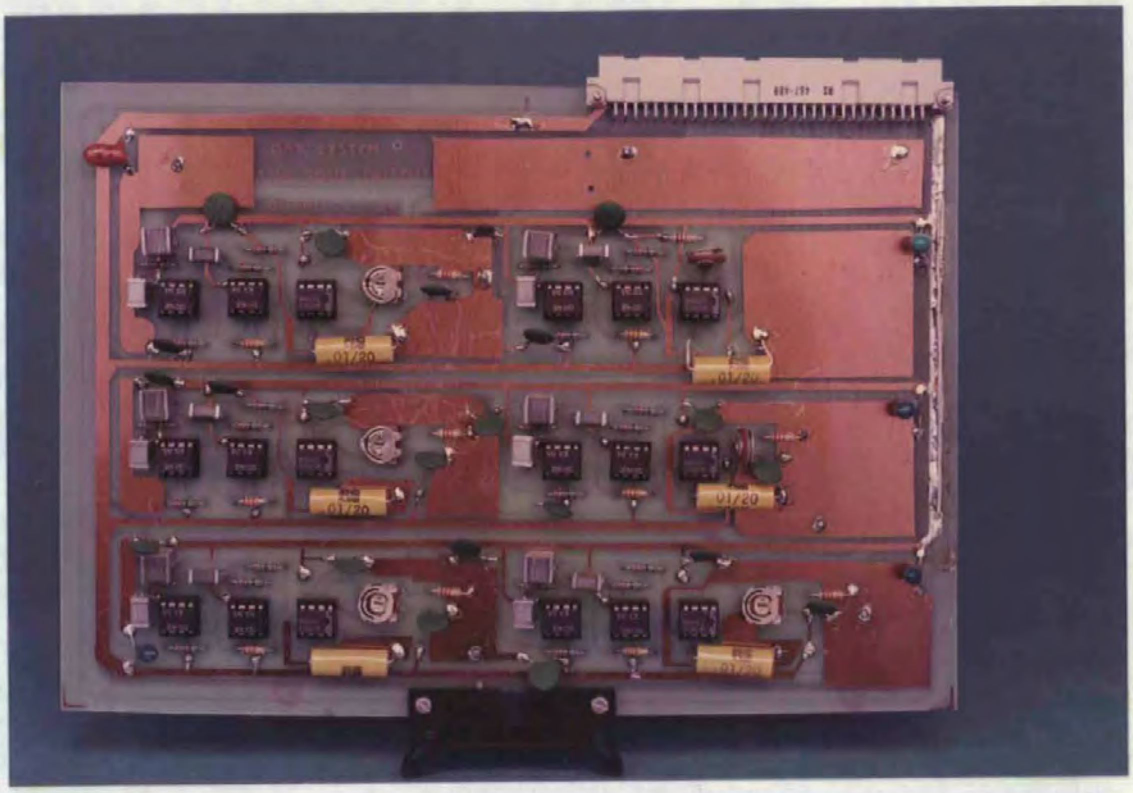


Plate 53 The OAR system Analogue output board

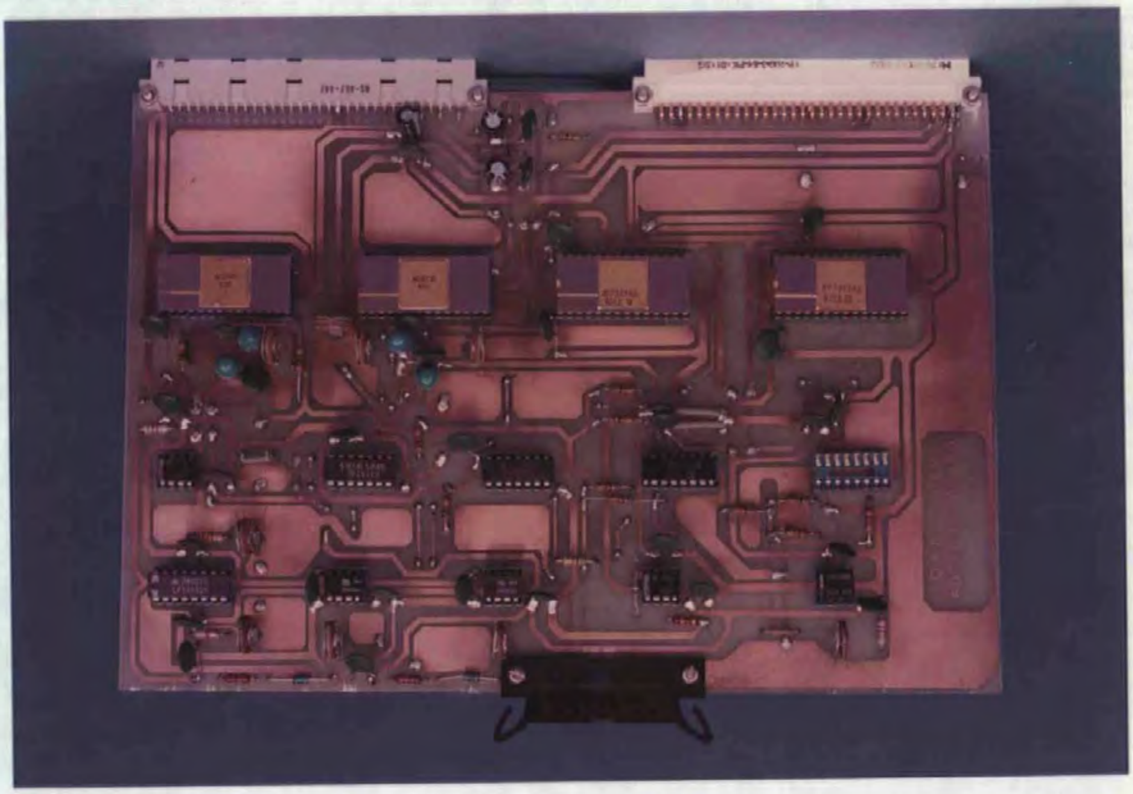


Plate 54 The OAR system ADC/DAC board

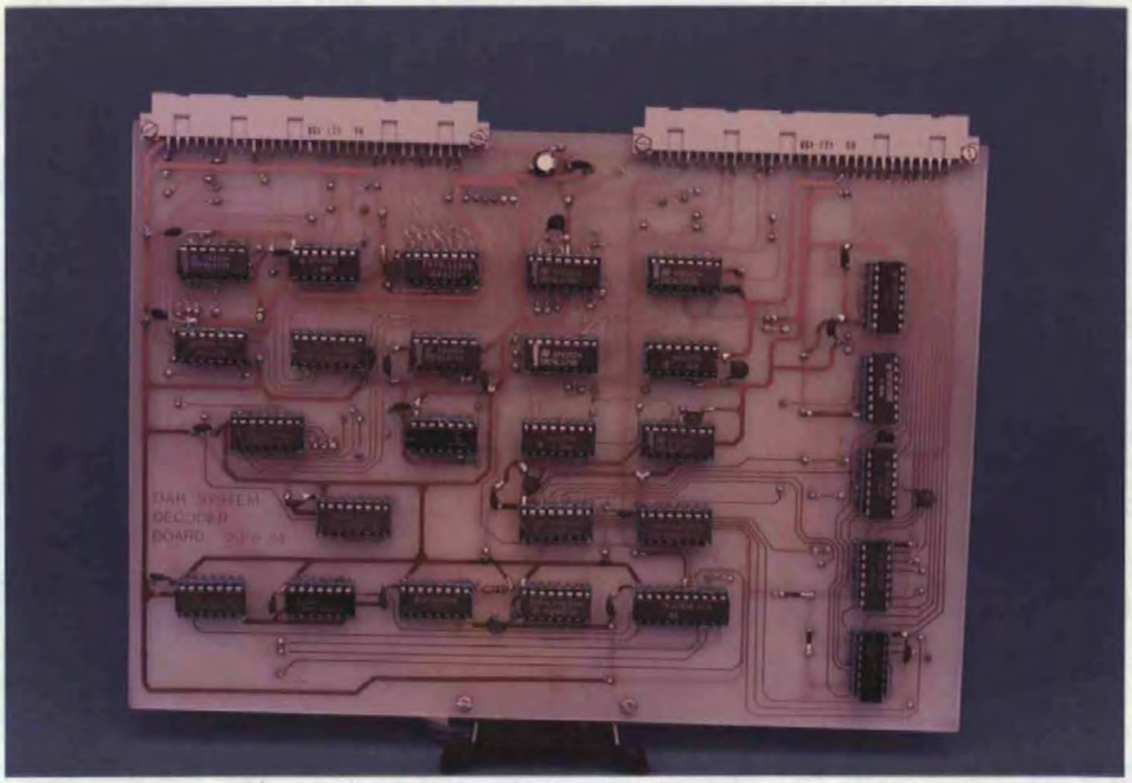


Plate 55 The OAR system decoder.

layout were in general spacious with ground planes on both sides of the board, where possible.

The costing for the OAR system parts is giving in appendix A16.

5.4 Summary

In this chapter, the specification, design and development of the hardware for a new Ocular Artefact Removal system were described. The OAR system takes advantage of the advances in semiconductor technology: use is made of an advanced microprocessor, the 68000 µP and new microprocessor compatible ADC and DAC.

The design of the various parts of the system is presented. These included the analogue signal conditioners, the data converters,
and the interface between the advanced microcomputer and the various input output (I / O) devices. The system is designed so that it is compatible with the standard EEG machines and with possible future expansion in mind.

References for Chapter 5

[1] VMEmodule Monoboard Microcomputer User's Manual. Motorola Semiconductor, 1983.

[2] GARRET P.H.
Analog I/O Design. Acquisition: Conversion: Recovery. Reston Publishing Co.,Inc.,Reston,Virginia,1981.

[3] VANDOREN A.H.
Data Acquisition Systems. Prentice-Hall,1982.

[4] MINTZ G.D., FRECKER R.C., JOY M.L.G.
Specification and Design of an Evoked-Response Data Acquisition System. Med. and Biol. Eng. & Comp., 18,1980,pp167-178.

[5] TOMPKINS W.J. and WEBSTER J.G. (ed)
Design of Microcomputer-Based Medical Instrumentation. Prentice-Hall,1981.

[6] SHEINGOLD D.H.
Analog-Digital Conversion Handbook. Analog Devices, Inc. Norwood,Mass.,1972.

[7] JERVIS B.W.,NICHOLS M.J.,ALLEN E.,HUDSON N.R., and JOHNSON T.E.
Comparison of two Methods for Removing Eye Movement Artefact from the EEG. (submitted to Electroenceph. and

Clin. Neurophysiol.).

[8] NICHOLS M.J.
An Investigation of the Contingent Negative Variation Using Signal Processing Methods. PhD Thesis, Plymouth Polytechnic, 1982.

[9] HEULSMAN L.P. and ALLEN P.E.
Introduction to the Theory and Design of Active Filters. McGraw-Hill,1980. ch. 2.

[10] COUGHLIN R.F. and DRISCOLL F.F.
Operational Amplifiers and Linear Integrated Circuits. Prentice-Hall,1977.

[11] TAUB H. and SCHILLING D.
Digital Integrated Electronics. McGraw-Hill,1977.

[12] TITUS C.A., TITUS J.A. and LARSEN D.G.
Microcomputer-Analog Converter Sofware and Hardware Interfacing. Howard W. Sams and Co,1978.

[13] MC68000 16-bit Microprocesor User's Manual. Austin,Texas: Motorola Semiconductor,Inc.,1980.

[14] VMEbus Specification Manual. Motorola, Semiconductor,Oct. 1981.

[15] SCANLON L.J.

The 68000: Principles and Programming. Howard W. Sams and co.,1981.

[16] STOCKTON J. and SCHERER V.
Learn the Timing and Interfacing of MC68000 Peripheral Circuits. Electronic Design, Nov. 8, 1979,pp. 58-64.

[17] BOLTON L.J. and BENNET G.H.
Design Features and Applications of the BPO 24-channel PCM System. POEEJ, Vol. 61,Jul. 1968, pp. 95-100

[18] VOGEL E.C. and MCLINTOCK R.W.
30-Channel Pulse-Code Modulation System, Part 1- Multiplex Equipment. POEEJ, Vol. 71,1978, pp.5-11.

[19] WOO B.,LIN L., and OWEN R.E.
ALU, Multiplier chips zip through IEEE Floating-Point Operations. Electronics, 19,1983,pp.

[20] Designing with the WTL 1032/1033. Application Note. Weitek, 1983.

[21] Linear Databook, National Semiconductor,1982.

Chapter 6 DEVELOPMENT OF THE SOFTWARE FOR THE ON-LINE OCULAR ARTEFACT REMOVAL SYSTEM

6.0 Introduction

In chapter 5, it was stated that the OAR system is sofware controlled. In this chapter, the design and development of the software are described.

First, a general description is presented to give an overview of the sofware, then a more detailed description of the OA removal routines is given. The heart of these routines, written in 68000μp assembly language, is the UD algorithm described in chapter 4. Next, the software floating point routines required by the OA removal system is described. Finally, a brief description of the software development tools used is given.

6.1 General Description of Sofware

The OAR system is interrupt driven. The interrupt signal is derived from the programmable Timer Module (PTM) on board the system controller [1,2], and has a frequency of 128 Hz. On interrupt, the OAR software is used to acquire data from up to 20 channels of EOG/EEG signals, remove OA from the EEG samples using the UD algorithm, and output the corrected EEG and/or the raw data to the EEG machine so that a paper chart record can be produced. The system operation is summarized in the flow chart of figure 6.1.

During the initialisation phase, the user is invited to specify the various system constants viz, the number of EEG channels to be corrected for artefact, the number of parameters and hence the model that should be used in the removal algorithm, and the number of corrected EEG and/or raw EEG signals to be output to the EEG machine. Some EOG signals and parameter estimates may also be output to the EEG machine. These constants are checked and if valid are used to initialize the system. A default value is used for any constant that is invalid. See appendix A15 for more details on the choice of, and default values for these constants.

The models given below are implemented, but only one may be used in the removal process. The number of parameters, specified by the user at the outset, determines which model is to be used. These models were found to give the best OA removal as discussed in chapters 2 and 4. It is to be noted that the software was written with the third model in mind, but with careful ordering of the variables, it has been made general enough to permit the implementation of the other two models as well.

1. $y(i) = \theta_1 VR(i) + \theta_2 HR(i) + e(i)$
2. $y(i) = \theta_1 VR(i) + \theta_2 HR(i) + \theta_3 HL(i) + e(i)$
3. $y(i) = \theta_1 VR(i) + \theta_2 HR(i) + \theta_3 HL(i) + \theta_4 HL(i) x HR(i) + e(i)$

After initialisation, the program loops around endlessly until valid data is available (figure 6.1a). (In future

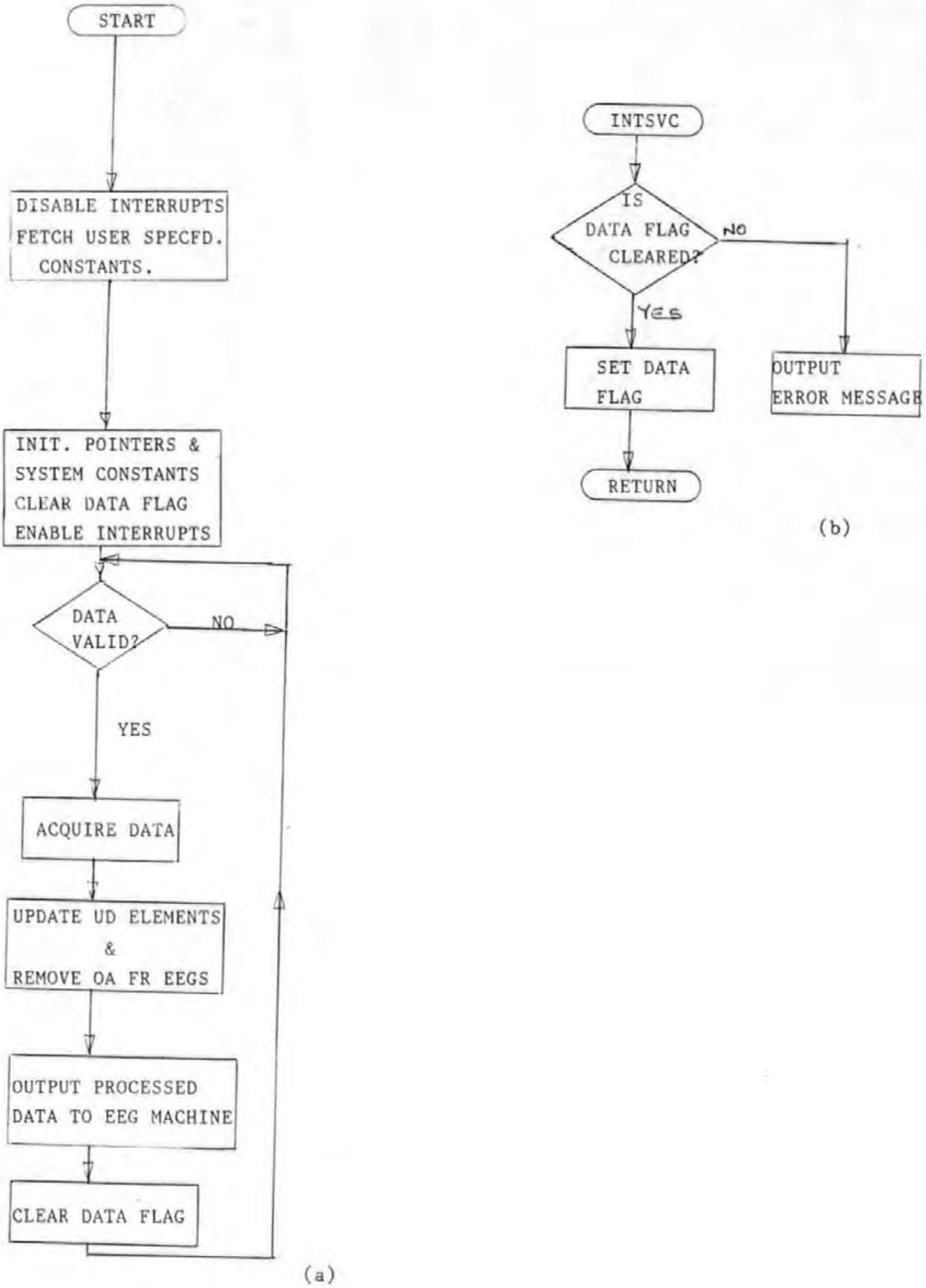


Figure 6.1 OAR system software
(a) Main program (b) Interrupt service routine.

OAR systems, this 'background' program will be replaced by a more useful one, e.g displaying the corrected EEG or parameter estimates). A flag (DATA FLAG) is set in the interrupt service routine (see figure 6.1b) each time the interrupt occurs and this is the indication to the main program that valid data is now available. After the data has been acquired, the elements of the UD algorithm are updated, the OAs are removed from the EEGs, and the corrected EEGs and/ or raw data are output to the EEG machine. Finally, the DATA FLAG is cleared to indicate that the current data samples have been successfully processed.

An error message is output to the visual display unit (VDU) and the program halted if an interrupt occurs before the previous one has been serviced . This will normally occur if too many parameters and/or EEG channels are specified than the software can process within the sampling interval of 8mS, and prevents the accumulation of unserviceable interrupts and the eventual system failure.

6.2 Description of OAR System Software

The OAR system software consists of several subroutines in addition to the main program and interrupt service routine described above. These include, the data acquisition, the ocular artefact removal, and output routines.

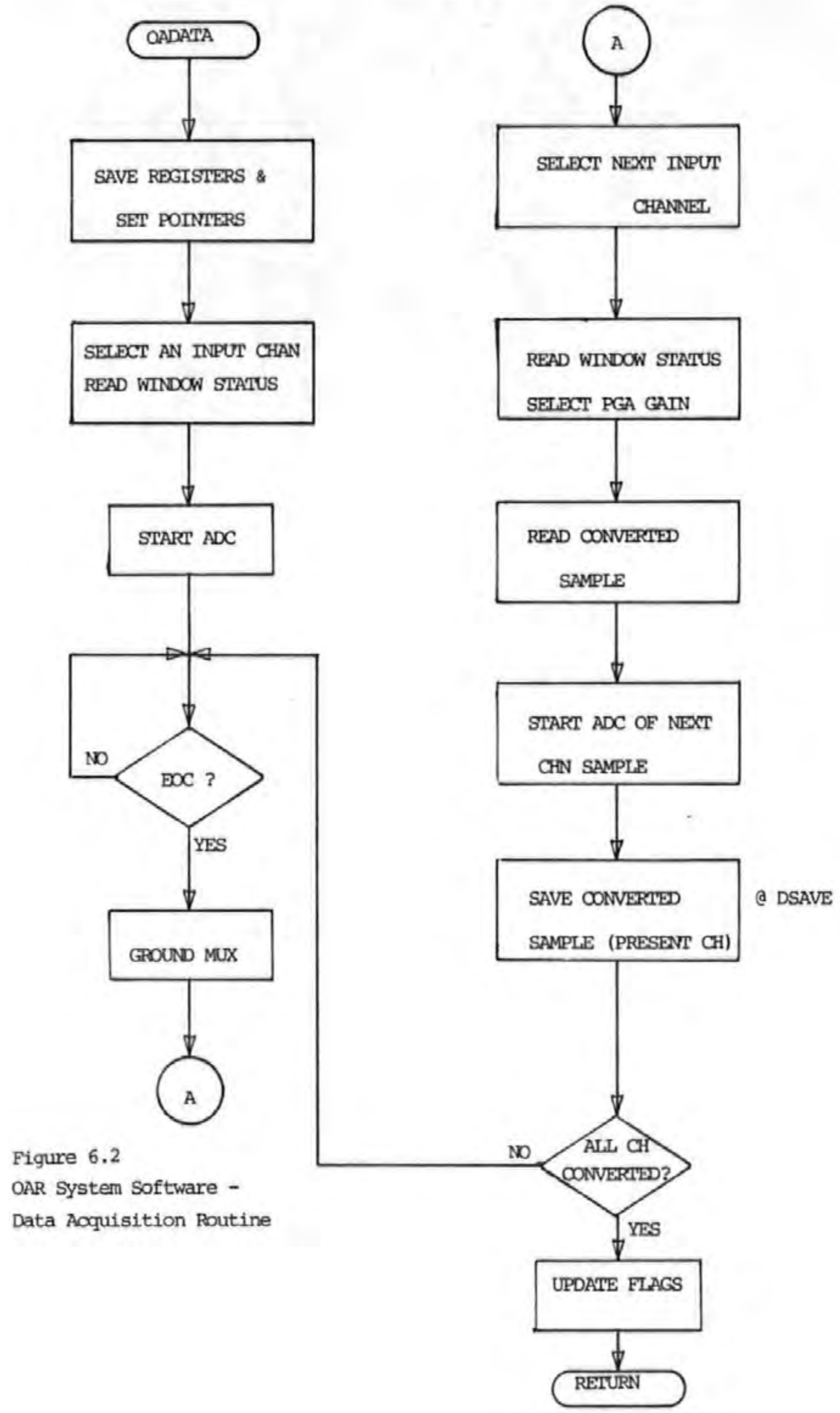


Figure 6.2
OAR System Software -
Data Acquisition Routine

6.2.1 Data Acquisition Routine (OADATA)

The main purpose of this routine is to acquire and save data from the EEG machine.

With reference to the flow chart,figure 6.2, on entry to the routine the various address registers are initialized. The use of these registers greatly facilitates subsequent memory accesses. The input channels are then sequentially selected and their samples digitized and saved using the subroutine DSAVE (see later). The PGA is always set to the higher gain before a channel is selected for digitization. After a channel is selected, the window detector status, WDSTS, is read and used to select the appropriate gain. If WDSTS is at logic '0', then the lower gain is selected, if at '1', the higher gain is selected. To reduce crosstalk between channels, the output of the multiplexer is grounded in between channel selection.

In anticipation of converting the digitized samples to floating point format in the DSAVE routine, the 12 bit samples are converted to 16 bit words. To save time, the conversion of the next channel sample is started before saving the present digitized sample. These operations are summarized in the flow chart of figure 6.2.

Data Saving (DSAVE) Routine

This subroutine saves the acquired sample in the data area of the Random Access Memory, (RAM) (see figure 6.3).

Two main operations are performed on the 16 bit word representing the converted sample before it is saved. First, it is converted into a floating point number as arithmetic operations in the OAR system are carried out using floating point scheme. To account for PGA, the exponent of the floating point number is incremented by 1 if a lower gain was selected before digitization, otherwise it is left unchanged. Next, the mean value is computed and removed from the sample. The following recursion formula is used to compute the mean:

$$\overline{S}(m+1) = \Upsilon\overline{S}(m) + (1-\Upsilon)S(m+1) \qquad (6.1)$$

where S(m+1) and $\overline{S}$(m+1) are, respectively, the (m+1)th sample and sample mean, and Υ is the forgetting factor (see chapter 4).

It is to be noted that this routine saves the EOG and EEG data in contiguous memory locations as:

Ch1 data (VR)

Ch2 data (HR)

Ch3 data (HL)

Ch4 data (VL)

Ch5 data (EEG1)

: :

Ch20 data (EEG16)

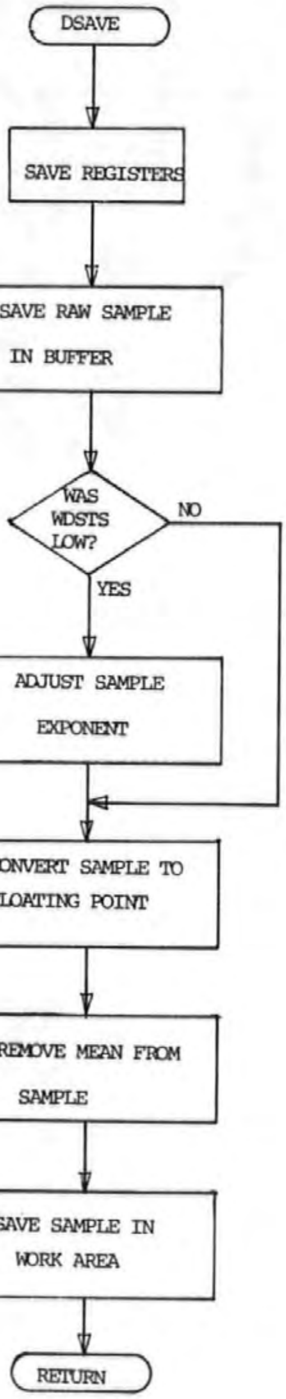


Figure 6.3
OAR System Software -
Routine Used to save Acquired Data

Thus, the EOG data are saved first, and in the order given, then the EEG data. The order in which the EEGs are stored is irrelevant, provided they start after the EOG.

The data is in fact stored in two identical sets. This is to simplify the output routine, where it may be required to output both the raw data and the corrected EEGs. The data saving routine is summarized in figure 6.3.

6.2.2 Ocular Artefact Removal Routines

Three auxiliary subroutines together with a control subroutine form the OA removal algorithm. The auxiliary subroutines are, the UDUFLT, used to update the elements of the UD filter; PERR, used to compute the prediction error or the difference between the EEG sample and the predicted EEG sample; and the ESTIMATE, which is used to update the parameter estimates for each EEG channel.

Now, the objective is to obtain for each EEG channel, an estimate of the background EEG, e(m+1) given by :

$$\hat{e}(m+1) = y(m+1) - \hat{\theta}(m+1)x^T(m+1) \qquad (6.2)$$

where y(m+1) is the measured EEG sample, $x^T(m+1)$ the EOG vector, $\hat{\theta}(m+1)$ the parameter estimate vector.

It was shown in chapter 4 that estimates $\hat{\theta}(m+1)$ can be obtained at each sample point recursively, using the following expression:

$$\hat{\theta}(m+1) = \hat{\theta}(m) + G\Delta e \qquad (6.3)$$

where G is the Kalman gain vector, given by $P(m)x(m+1)/\alpha$ and Δe the prediction error is given by: $y(m+1)-\hat{\theta}(m)x^T(m+1)$. Thus, the removal algorithm may be summarized as follows:

(i) Update the elements of the UD filter and obtain G

(ii) Compute the prediction error, Δe, for the given channel.

(iii) update the parameter estimates for the channel and obtain $\hat{e}(m+1)$ from 6.2 .

(iv) Repeat (ii) and (iii) for each EEG channel.

Remark

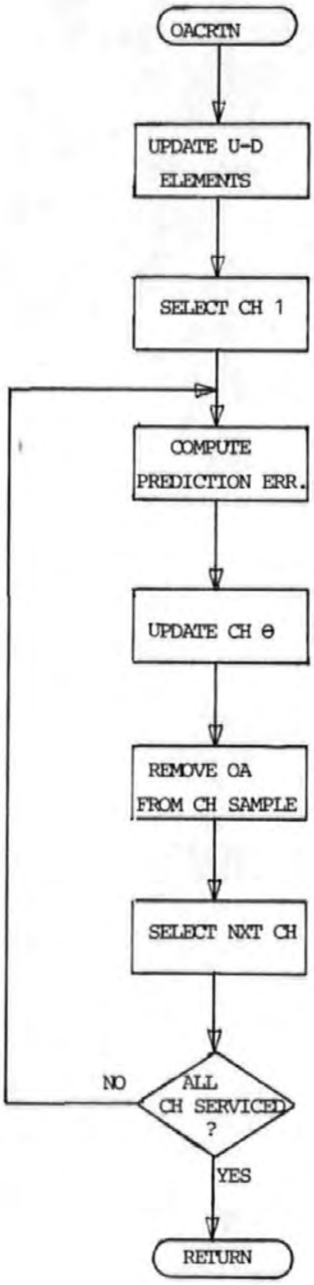
Experience with the various recursive algorithms showed that the prediction error, Δe, and $\hat{e}(m+1)$ do not differ significantly, so that step (iii) may be ignored to ease time contraint imposed by the number of arithmetic operations involved.

The subroutine UDCRTN, figure 6.4, controls the removal of OA from the EEG samples. The three auxiliary subroutines will now be described.

(a) The UDUFLT

In chapter 4, it was shown that the covariance matrix, P, which is used to update the parameter estimates can be factored as:

$$P = UDU^T$$



@ $y(i) - x^T(i)\,\hat{\theta}(i)$

Figure 6.4
OAR System Software -
Routine that controls the
Removal of OA from EEG samples.

A modified recursive algorithm for updating the UD factors of P which was described in chapter 4 is reproduced in figure 6.5 for ease of reference.

```
1. $v = U^T(m)x$
2. $b_i = D_i(m)v$          , $i=2,..,n$
3. $\alpha_1 = \gamma + b_1 v_1$
4. $D_1(m+1) = D_1(m)/\alpha_1$
5. $b_1 = D_1(m+1)v_1$
For $j = 2,3, .. ,n$  recursively evaluate (6) to (10)
    6. $\alpha_j = \alpha_{j-1} + v_j b_j$
    7. $\rho_j = -v_j/\alpha_{j-1}$
    For $k = 1,2, ... ,j-1$  recursively evaluate (8) and (9)
        8. $U_{kj}(m+1) = U_{kj}(m) + b_k \rho_j$
        9. $b_k = b_k + b_j U_{kj}(m)$
    10. $D_j(m+1) = D_j(m)\alpha_{j-1}/(\alpha_j \gamma)$
11. $G = b/\alpha_n$
```

Figure 6.5 Modified UD factorization Algorithm

The Kalman gain vector, G, obtained at step 11 is used to update the parameter estimates, as indicated in eqn (6.3).

The subscripts (k,j) in the algorithm indicate that U is stored as a two dimensional array. Because U is upper triangular, it is sufficient to describe it by n(n+1)/2 of its elements. Further, since all the diagonal elements of U are unity, the elements of D can be stored along the diagonal. Thus, for n=4, U becomes:

$$U = \begin{bmatrix} D_1(1) & U_{12}(2) & U_{13}(4) & U_{14}(7) \\ 0 & D_2(3) & U_{23}(5) & U_{24}(8) \\ 0 & 0 & D_3(6) & U_{34}(9) \\ 0 & 0 & 0 & D_4(10) \end{bmatrix}$$

To save storage and for ease of programming, the elements of U are stored columnwise , with the zeros omitted, in contiguous memory locations as indicated by the numbers in brackets. This is the so-called vector subscripting [3], whereby a matrix such as U above is stored as a vector or one-dimensional array instead of as a two dimensional array.

The UD algorithm is summarized in the flow chart of figure 6.6.

The FORTRAN and assembly language coding of the UD algorithm are, respectively, given in appendices A12 and A15. In implementing the algorithm in assembly language, the various powerful instructions of the 68000 µP were exploited [4,5]. The excellent features of the processor are evident from examination of the listing.

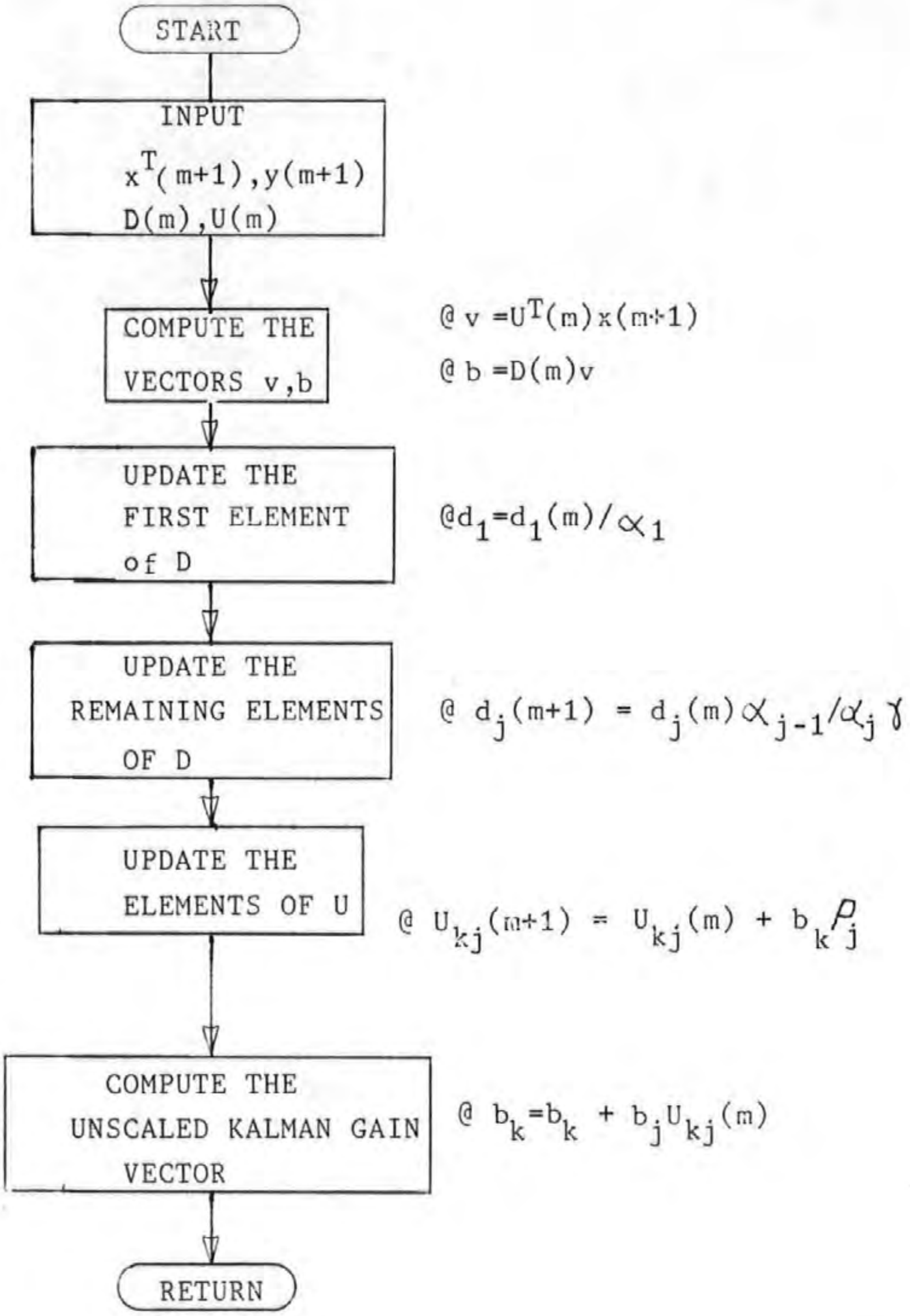

START
INPUT
$x^T(m+1), y(m+1)$
$D(m), U(m)$
COMPUTE THE VECTORS v,b
@ $v = U^T(m)x(m+1)$
@ $b = D(m)v$
UPDATE THE FIRST ELEMENT of D
@$d_1 = d_1(m)/\alpha_1$
UPDATE THE REMAINING ELEMENTS OF D
@ $d_j(m+1) = d_j(m)\alpha_{j-1}/\alpha_j \gamma$
UPDATE THE ELEMENTS OF U
@ $U_{kj}(m+1) = U_{kj}(m) + b_k \rho_j$
COMPUTE THE UNSCALED KALMAN GAIN VECTOR
@ $b_k = b_k + b_j U_{kj}(m)$
RETURN

Figure 6.6 UD factorization Algorithm

(b) Prediction Error (PERR) Routine

This routine is used to compute $[y - x^T\hat{\theta}]$ given in (6.3). Because of the similarity of the expressions for Δe and $\hat{e}(m+1)$ (cf eqns 6.2 and 6.3), this routine is also used to compute $\hat{e}(m+1)$

(c) Parameter Estimate update (ESTIMATE)

This routine is used to update the parameter estimates that is, to solve for $\hat{\theta}(m+1)$ in (6.3).

6.2.3 Output Routines.

A control subroutine, OUTPUT, together with two auxiliary subroutines are used to output the corrected EEG samples, the EOG and / or the raw EEG (measured EEG with mean removed) to the EEG machine, depending on the user specified (or default) constants.

The two auxiliary routines are the 'FLTINT' and the 'DACOUT' which are, respectively, used to convert a given data sample from floating point to integer and from digital to analogue. Additionally, the DACOUT selects the output channel to which the converted sample should be sent.

(a) Conversion from Floating Point to Integer (FLTINT)

The input to the DAC must be a 12 bit integer in the range 000 to $FFF (Hex). Negative numbers are in the range 000 to $7FF, and positive numbers $800 to $FFF. Since the

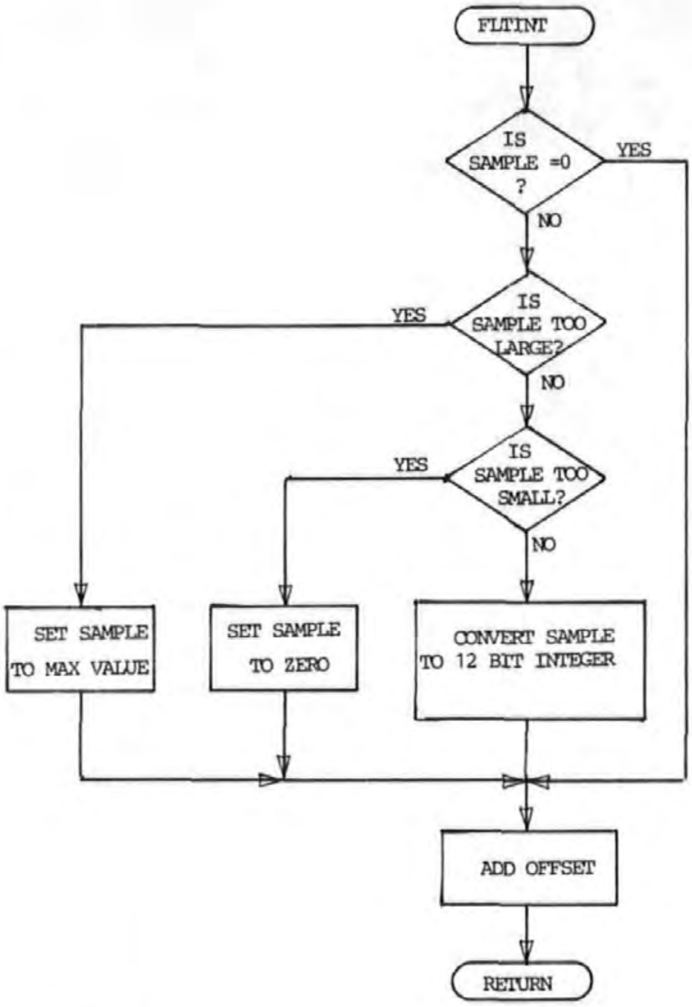


Figure 6.7

OAR System Sofware -

Subroutine to convert sample from

Floating-point to 12-bit integer suitable for DAC.

arithmetic operations in the OAR system are in floating point scheme, it is necessary to convert the ouput data or result to 12 bit integer. The subroutine FLTINT is used to perform this conversion.

With reference to the flowchart in figure 6.7, it is seen that three decisions have to be made before conversion:

(i) Is the sample zero? If it is, then an offset is simply added to obtain directly the desired integer number.

(ii) If the sample is not zero, is it too large? A sample is too large if it is outside the range of the DAC. This is easily determined by examining the exponent of the floating point number to be converted. Samples that are too large are set to the maximum integer value, with the appropriate sign.

(iii) Is the sample not zero, but too small? If it is too small, it is treated as a zero. A sample is too small if its integer equivalent is less than 1.

(iv) If (i) to (iii) above are not true, then the integer equivalent of the sample is within the DAC range. The sample is then converted to integer by shifting right its mantissa (see section 6.3) an amount determined by its exponent.

(b) <u>Subroutine DACOUT</u>

This subroutine outputs the 12 bit sample to the DAC where it is converted into analogue form. Next it selects the demultiplexing channel and the output sample and hold where the analogue sample should be sent. This program is listed along with the other routines used to output data in appendix A15 .

6.3 Software Floating Point Arithmetic Routines

Fixed and floating point arithmetic are the two most common types of arithmetic in a digital system [6,7]. Fixed point representation has the advantage of being fast, but limits the range of numbers that can be represented. To prevent results of arithmetic operation going outside the number range, the operands have to be scaled before and/or after each arithmetic operation. To overcome the problem of scaling associated with fixed point approach floating point representation is often used. A floating point scheme greatly increases the dynamic range of the numbers and it is preferred in applications where the magnitude of numbers vary widely [6]. Thus, floating point (FP) arithmetic operations were preferred for the OAR system. Furthermore, FP arithmetic was used to develop the removal algorithms on the main frame computer (see chapter 4), so that the use of FP follows naturally.

As speed is vital in this application, hardware floating point was considered the best approach, but it was found that hardware floating point devices available at the time were both expensive and too slow. Therefore, the number of EEG channels to correct was scaled down so that software FP could be used until fast FP devices became available.

An overview of the binary floating point routines which were used to perform the basic arithmetic operations of add, subtract, multiply and divide in 68000 microprocessor assembly language is presented here. A more detailed description of the assembly language routines, which are of interest in their own right, is given in appendix A9. The material presented here and in appendix A9 has been written up for publication [8].

The floating point number representation and the data format used will be described in section 6.3.1 and 6.3.2 respectively, and in section 6.3.3 the floating point arithmetic routines are described.

6.3.1 Binary Floating Point Representation

A floating point number X is represented as the product of two signed numbers, the mantissa M and the exponent, E:

$$X = M.2^E$$

where 2 is the base of the binary system.

The exponent primarily determines the range of the numbers that can be represented, whereas the mantissa determines the accuracy of the numbers. In this system, the exponent and the mantissa are represented, respectively, by 8 and

16 bits. This choice is a suitable compromise between speed and accuracy. Similar choice of exponent and mantissa lengths have been used in a control application [9] and were found to be adequate. The range of the floating point numbers is the difference between the largest and the smallest numbers that can be represented, which in this case is:

$$(0.5 \times 2^{-64}) \text{ to } (1 - 2^{-15}) \times 2^{63}$$

Of the 16 bits used to represent the mantissa, 1 bit is the sign bit and the least significant bit may be of a doubtful accuracy due to rounding effects. Thus, the accuracy of the floating point numbers is 1 in 2^{14} (0.61×10^{4}), that is about 4 decimal digits.

6.3.2 Floating Point Data Format

The format for the floating point number (or word) is given in figure 6.8. The mantissa is expressed as a 16 bit fractional two's complement value with the binary point assumed to the right of the sign bit. The exponent is 8 bits long in excess 64 form. The exponent normally has a range of $-64 \leq E \leq 63$, but by adding a fixed constant to the exponent (in this case 64) such that it is always positive the range becomes: $0 \leq E \leq 127$. This ensures that when the mantissa is zero the exponent is also zero, giving an all zero floating point number. Additionally, this form of representation has the advantage of making the detection of over- and under-flow simple [10]. However, when performing arithmetic operations the

e7	e6	...	e0	S	m14	m13	...	m0

exponent (8bits) mantissa (16bits)

(a)

Address	XX
Addrass+1	Exponent
Address+2	High byte of mantissa
Address+3	LOw byte of mantissa

(b)

Figure 6.8 (a) Floating point data format (b) Floating point memory allocation.

exponent has to be restored to its unbiased form.

All numbers going into or out of the floating point routines must be normalized. A normalized floating point number is one in which the sign bit and the bit to the right of the binary point (bits S and m14 in figure 6.8) of the mantissa are different. In this system, a normalized mantissa satisfies the condition: $0.5 \leq |M| < 1$. If the result of an arithmetic operation produces a mantissa that is outside this range, it is said to have overflowed (mantissa overflow) if the result is equal to or greater than unity, and underflowed (mantissa underflow) if it is less than 0.5.

The floating point numbers in memory are allocated four consecutive bytes of memory (2 words), although only the first three are used as shown in figure 6.8b This is to simplify reading and writing to the memory of the 68000 µP. It also allows the number of bits in a floating point word (mantissa + exponent) to be increased easily, if desired.

6.3.3 Floating Point Arithmetic Routines

The floating point arithmetic routines consist of three routines for the basic arithmetic operations: FADD (for addition), FMUL (for multiplication), and FDIV (for division). Exit from these routines is made via a common routine, 'EXIT'.

On entry into any of the arithmetic routines, registers D3 and D2 contain, respectively, the first and second operands. In the case of division, D3 and D2 contain, respectively, the dividend and the divisor. On exit, D3 contains the result. Apart from D3, none of the registers are modified on exit by the floating point routines.

The 68000 µP has an abundance of registers and these have been used freely to improve the execution speed. For the same reason, the use of some otherwise efficient instructions have been avoided. Arithmetic operations can be chained with the result of the previous operation left in D3 to be used as the first operand of the next operation.

A detailed description of these routines can be found in appendix A9.

6.3.4 Execution Times of the Floating Point Routines

Estimates of the execution times of the floating point routines are given below. These times are only approximate as the execution times depend on the data.

Routine	Execution time (µS)
Addition	70
Multiplication	70
Division	85

6.4 Software Development

Due to lack of adequate development tools, the OA system software was developed on a number of systems, each system being used to develop only a part of the software. These were, the VME110 microcomputer, the Wicat, the Future data and the Exorset.

The VME110 is the system controller board used to develop the programs initially and to test later programs after they have been put on EPROMs. It is equiped with a monitor program with line by line assembler and several debugging facilities. These facilities were used extensively in the program development. However, the VME110 board lacked suitable permanent storage facilities.

The WICAT, a 68000 µP based system at present is used only as a terminal equipment for the VME board. It also holds a fully documented copy of the OAR software from which a hard copy may be obtained. The Wicat has a resident assembler which can be used to assemble programs, but has no debugging facilities. When the system was bought about half way into this research, it was hoped that it would be used to develop programs. However, due to insufficient memory, inadequate software, and hardware problems encountered with the system this was not possible.

The Future Data system was the only system in the polytechnic at the time that had facilities for developing

68000 assembly language programs, but these were limited . The limitations included the inability to split object codes into upper and lower bytes, which is necessary for 16 bit processors and the inability to program memory devices above 2K bytes (at least 4K EPROMs were required). Thus another system, the EXORSET, was used to perform these two functions.

The EXORSET is a development system for the 6800 series which does not support 68000 programs. However, it can program a variety of memory devices and supports at least a high level language.

Thus, the approach adopted was to cross assemble programs on the Future Data system, and after editing put these on one or more 2K EPROMs (depending on the size of program), read the codes from the EPROMs into the EXORSET, use a BASIC program on the EXORSET to split the codes into upper and lower bytes, and then put the upper and lower bytes of the codes on separate 4K EPROMs.

Ideally, it would have been better to use a single system to develop the programs and then download on to the VME110 card. However, the method devised above although less efficient was quite straight forward and took little time.

6.5 Memory Requirement

The on-board memory of the VME110 card is organized into four socket pairs, as shown in figure 6.9. Each socket pair can be configured to accept a wide range of RAM or ROM, except the first which must be a ROM. This socket pair contains the monitor program.

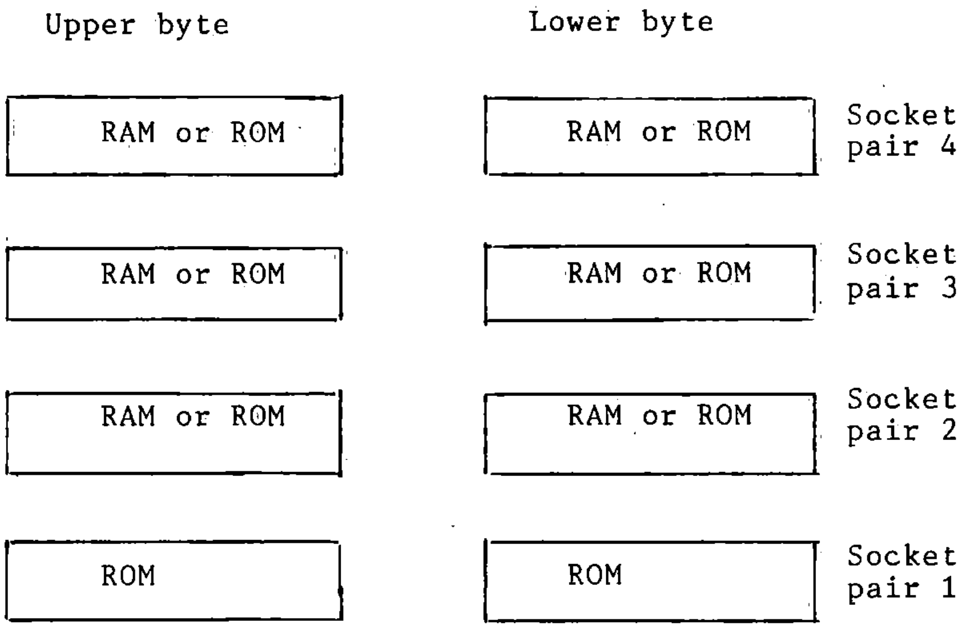


Figure 6.9 VME110 local memory organization

Headers are provided for configuring the sockets. Additionally, a map decoder PROM is required to provide information about the memory [11,12]. This includes, whether an address is local or global, whether a local memory is ROM or RAM, whether a local RAM location is write protected and implicitly, the size of the memories in each socket pair.

The user programs the decoder to provide this information. The memory requirements chosen for the OAR system are as follows: 8K RAM and 24 K ROM. This includes two 8K x 8 ROMs for the monitor, two 4K x 8 ROMs for the OAR system software (socket pair 2), and four 2K x 8 RAMs used as work area (socket pairs 3 and 4). These memories occupy the following address ranges:

System Monitor	: address \$F00000 - \$F03FFF
OAR Sytem	: address \$F04000 - \$F05FFF
Write protected	: address \$000000 - \$000FFF
+ Vectors +	: address \$001000 - \$001FFF
work area	

The approach used by Motorola whereby ROM is placed in the high address (the HI block) and RAM in the low address (LO block) [11,12] has been adopted.

6.6 Summary

A detailed description of the OAR system sofware, its design and development was presented in this chapter. The heart of the software is the numerically stable UD algorithm. Also discussed are the software floating point routines which were used in the OAR sofware, and routines required to control the hardware described in chapter 5.

References for Chapter 6

[1] BRYCE H.D.

Versatile Programmable Module meets µP Timing Needs. EDN, Oct.,1980, pp. 145 - 153.

[2] HEMENWAY J. and GRAPPLE R.

To Construct a Real-Time Clock, combine Hardware and Software. EDN, Oct. 5,1980, pp115 - 120.

[3] BIERMAN G.J.

Factorization Methods for Discrete Sequential Estimation. Academic Press, New York,1977.

[4] MC68000 16-Bit Microprocessor User's Manual.

Austin, Texas, Motorola Semiconductor, Inc., 1980.

[5] SCANLON L.J. The 68000 : Principles and Programming.

Howard W. Sams and Co., 1981.

[6] FLORES I.

The Logic of Computer Arithmetic. Prentice Hall, 1963.

[7] RABINER L.R. and GOLD B.

Theory and Application of Digital Signal Processing. Prentice Hall, 1975.

[8] IFEACHOR E.C., JERVIS B.W., MORRIS E.L. and ALLEN E.

Software Floating Point Arithmetic Routines for the

Motorola 68000 Microprocessor. (to be published).

[9] CLARKE D.W., COPE, S.N. and GAWTHROP P.J.
Feasibility Study of the Application of Microprocessor to Self-tuning Controllers. O.U.E.L. Report 1137/75, 1975.

[10] COPE S.N.
Floating-Point Arithmetic Routines and Macros for an Intel 8080 Microprocessor. O.U.E.L Report 1123/75, 1975.

[11] VMEmodule Monoboard Microcomputer User's Manual. Motorola Semiconductor, 1983.

[12] VME Debug Packages User's Manual, Motorola Semiconductor, 1982.

Chapter 7 SYSTEM TESTING AND EXPERIMENTAL RESULTS

7.0 Introduction

In the last two chapters, the design and development of the hardware and software of the ocular artefact removal (OAR) system were discussed. In this chapter tests on the system are described. The tests were carried out in two phases, namely, simulated tests and pre-clinical tests. In the simulated tests, the various subsystems (hardware and software) were tested using test data, signals and programs to discover the blatant errors and bugs. Next, the entire system was tested with test signals that resemble the EOGs/EEG to verify that the system worked in the real-time situation. The test program and circuits for generating the test signals will also be described.

In the pre-clinical tests, both normal and patient subjects were used. Experimental results from these tests will be presented and discussed.

7.1 Simulated Tests of the OAR System

7.1.1 Hardware and Software Tests

(a) Hardware

The various hardware subsystems (multiplexer, ADC, PGA etc) were tested with small programs. The debugging facilities on the VME110 microcomputer were used in these tests. These facilities allowed reading from or writing to the subsystems, single stepping through the test programs, and the display of the microprocessor registers on the terminal.

Firstly, the decoder which generates the control signals was tested. For each function listed in table 5.1 (chapter 5), short assembly language programs of three or four lines were used to generate the correct control signals which were then monitored on an oscilloscope. For example, to generate the multiplexer control signals (MX0 to MX4), the desired channel number was entered in a data register and then a write to the multiplexer address was performed. The logic levels of the control lines were then monitored on the oscilloscope to verify that they were correct. Similar tests were carried out for other control signals and enabled the discovery of wiring errors etc. Next, the analogue input circuit was tested, to verify that with both sinusoidal and sampling signals applied, its output was a sampled version of the sinusoid. (the sampling signal was derived from the Programmable Timer Module on the VME110).

After testing the hardware subsystems and correcting any errors, it was necessary to test the overall hardware. To do this it was decided to connect the system 'back-to-back'. Thus, sine wave signals were applied to the system inputs, on interrupt these were sequentially selected and digitized by the ADC, the digitized samples were then converted back to analogue by the DAC, and demultiplexed via the DMUX and the output sample and holds. After smoothing the samples with the output filters, the resulting analogue signals were then compared with the original input signals to verify that they were similar.

In all the hardware tests the programs were carefully chosen to simulate operations that were similar to those of the OAR software, so that in effect the OAR software was also being tested.

(b) Software Tests

As was stated earlier, some aspects of the software were tested in the hardware tests. The removal routines are, however, the most important part of the OAR system software. These were tested with test data obtained from the data used in the investigation described in the previous chapters. An on-line OA removal was simulated on the main frame computer as described in chapter 4, and values of the EOG and EEG data, parameter estimates as well as elements of the UD matrices at a few sample points were printed.

The EOG/EEG data were then stored in the RAM of the OAR system, after converting to floating point format. With the system initialized with similar starting values for $\hat{\theta}$ and the UD elements as on the main frame computer, and using the debugging facilities, the EOG and EEG samples were fetched and used in the removal algorithm, see figure 7.1. At the end of each iteration, the values of $U, D, \hat{\theta}$ and the corrected EEG were compared to those produced by the main frame computer. Typical values from the test are given in table 7.1, from which it is seen that the values produced by the OAR and the main frame computer are the same.

This test revealed a number of software errors, which were later corrected. It will be recalled that the OAR system arithmetic is carried out in floating point scheme. Thus it was desirable to perform floating point/decimal conversion to allow communication with the system in easily understandable form and to check the validity of the results. Routines developed for performing these conversions are described in appendix A10. The values given in table 7.1 were obtained with these routines.

7.1.2 System Testing with Real-Time Signals

Although the hardware and software of the OAR system have been shown to work separately, it was necessary to verify that the entire system worked before proceding to preclinical tests. This required generating test signals with characteristics similar to the EOG and EEG.

Circuit for Generating Test Signals

The circuit used to generate the EOGs/EEG is shown in figure 7.2. The requirements of the circuits were:

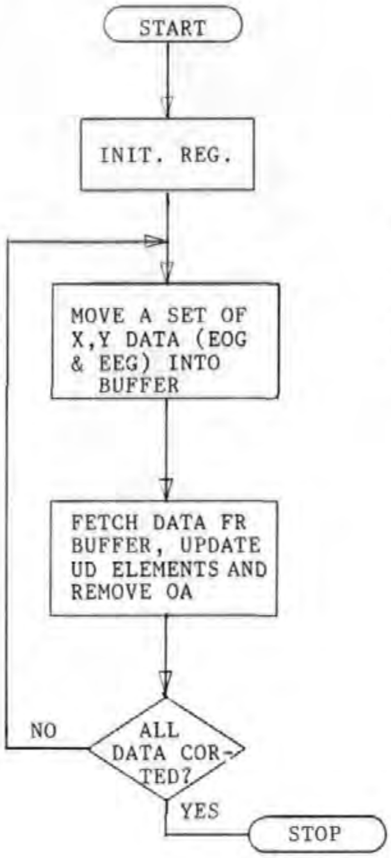

START
INIT. REG.
MOVE A SET OF X,Y DATA (EOG & EEG) INTO BUFFER
FETCH DATA FR BUFFER, UPDATE UD ELEMENTS AND REMOVE OA
ALL DATA COR-TED?
NO
YES
STOP

Figure 7.1

Flow chart for the program used to test the assembly language coding of of the UD algorithm

		m=1		m=2		m=3		m=4	
		MF	OAR	MF	OAR	MF	OAR	MF	OAR
(a)	$\hat{\theta}_1(m)$	0.1416	0.416	0.2463	0.2463	0.2667	0.2667	0.7074	0.7073
	$\hat{\theta}_2(m)$	-0.0260	-0.0260	0.0190	0.0189	0.0488	0.0488	0.1962	0.1962
(b)	$\hat{\theta}_1(m)$	0.1030	0.1030	0.2159	0.2158	0.2285	0.2285	0.7089	0.7089
	$\hat{\theta}_2(m)$	0.0910	0.0910	0.1280	0.1281	0.1843	0.1841	0.1870	0.1864
	$\hat{\theta}_3(m)$	0.1106	0.1106	0.1011	0.1012	0.1259	0.1258	-0.0083	-0.0088
(c)	$\hat{\theta}_1(m)$	0.1005	0.1005	0. 045	0.2043	0.1858	0.1858	0.1857	0.6627
	$\hat{\theta}_2(m)$	0.0984	0.0984	0.0599	0.0601	-0.0667	-0.0670	-0.0711	-0.0717
	$\hat{\theta}_3(m)$	0.1019	0.1018	0.1722	0.1722	0.3642	0.3640	0.2373	0.2366
	$\hat{\theta}_4(m)$	-0.0030	-0.003	0.0023	0.0023	0.0079	0.0079	0.0081	0.0081

Table 7.1 Comparison of results of simulation of the UD on-line OA removal on the main frame (MF) computer and on the OAR system. (a) model 2D (VR,HR) (b) model 3D (VR,HR,HL) (c) model 4D (VR,HR,HL,[HLxHR]).

(i) Generate signals that were similar to the EOGs and EEG signals. Thus the EOGs should be correlated (but not perfectly), and have a low frequency ($\sim$ 10Hz). It was also useful to simulate random ocular movement. The EEG should be approximately random.

(ii) The signal levels should be similar to those at the auxilliary outputs of the EEG machine ($<$ 2V peak).

Referring to figure 7.2, the output of an oscillator running at a frequency of 20 Hz (a lower frequency made viewing on oscilloscope difficult) was used to generate a pseudorandom sequence (PRS) of length 15 bits, by logically controlling the shift register. The PRS allows the simulation of random ocular movement.

The resistance-capacitance networks were used to shape the PRS output to obtain the desired signal levels and shapes [1]. The resulting 'EOG' signals were added to a noise signal from a noise generator to give a contaminated signal, reminiscent of the OA process on the scalp.

A word or two on the test circuit is in order. The PRS output is maximal length (15 bits), made possible by exclusive-Oring the last two bits of the shift register, IC3. To avoid the all zero state at the shift register output on turning the power on, for example, which would inhibit the PRS [2] the NAND gate was used to detect the all zero state and force the PRS to sequence.

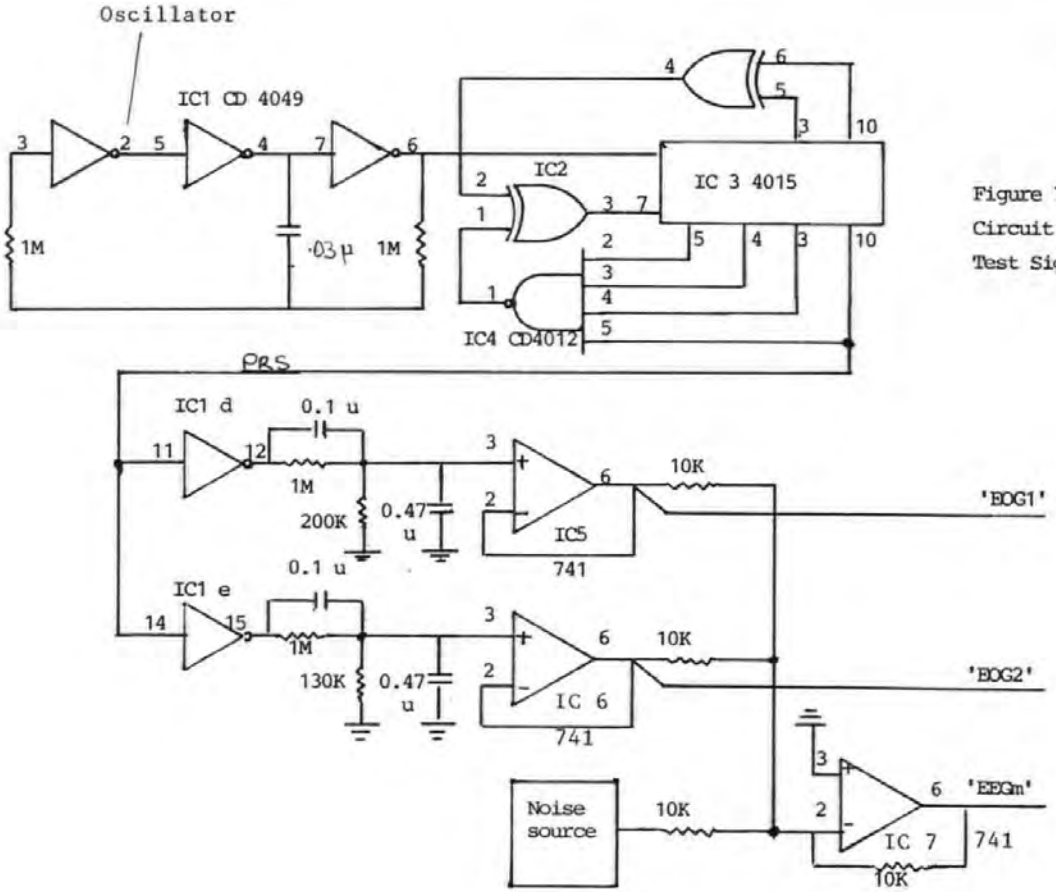


Figure 7. 2
Circuit for generating
Test Signal

To prevent the interaction of the signals (EOGs, EEG), an inverting adder was used. This feature was desirable to allow each input signal to be individually controlled. The RC network arrangement has the feature that its components can be adjusted to obtain different waveshapes.

The simulated signals were used to test the OAR system which was found to work satisfactorily.

7.2 Pre-Clinical Tests

The pre-clinical testing was carried out at the Freedom Fields Hospital, Plymouth, and consisted of two phases. In the first phase, six normal subjects were used to allow extensive tests to be carried out. The objective here was to assess the reliability of the OAR system, which can only be obtained from trying the instrument out on people. This phase was also a 'learning' phase, allowing the understanding of how the system behaved and discovering any bugs. In the second phase two patient subjects exhibiting slow waves were used, to assess how the OA removal process affected the waves. One of the subjects could not co-operate fully and also suffered from chronic epileptic condition, and the other was unco-operative and suffered from a mental illness.

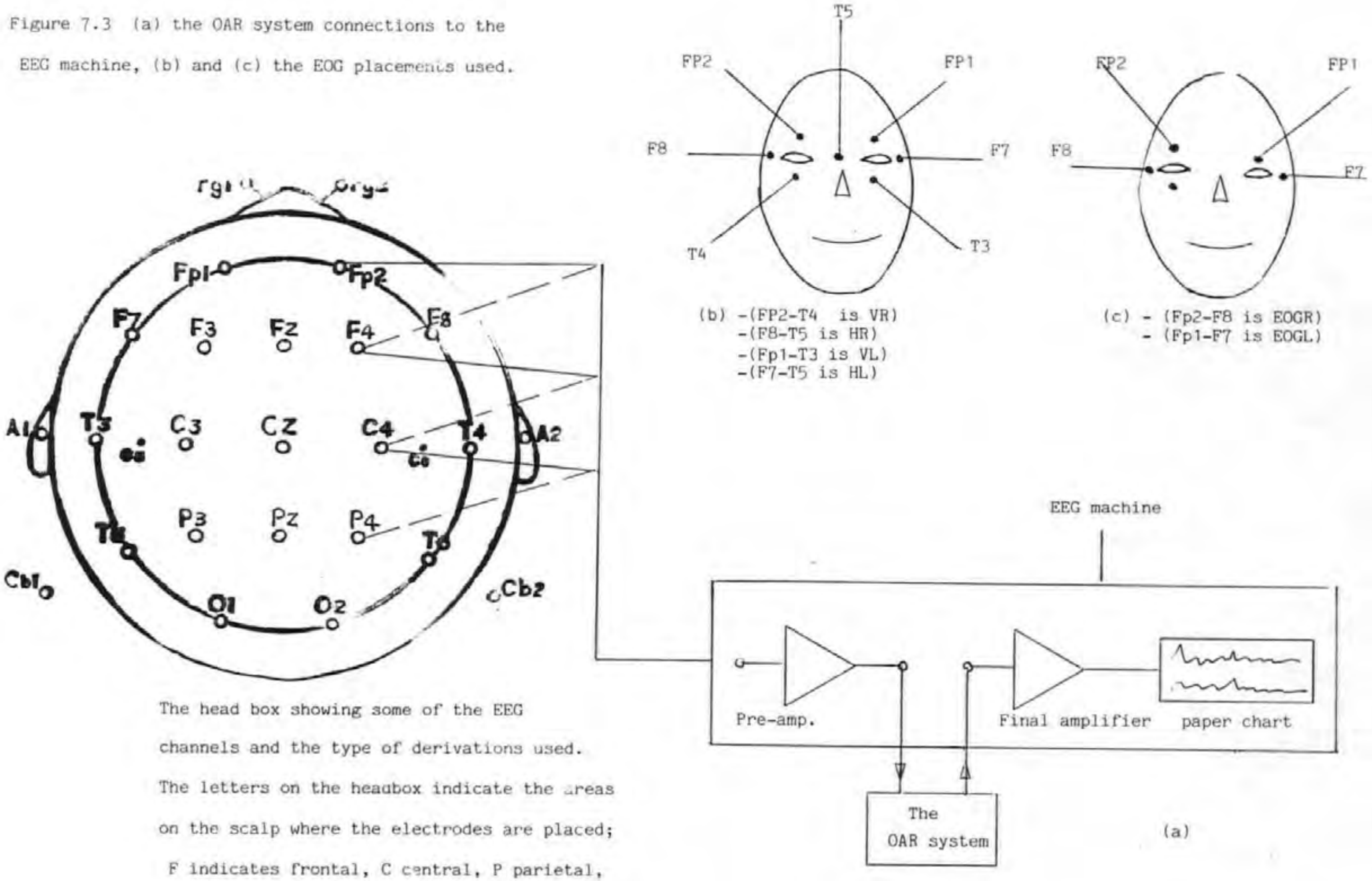
7.2.1 Methods and Material

(a) Experimental Apparatus

Ten EEG signals were derived from electrodes placed as shown in figure 7.3. These were FP2-F4, F4-C4, C4-P4, FP1-F7, FP1-F7, F3-C3, C3-P3, Fz-Cz, Cz referred to the right ear lobe, and Cz-Pz. The EOG signals were derived from electrodes placed near the eyes as described in chapter 1, sections 1.2.1 and 1.6. Two different EOG electrode placements were used (see figure 7.3) and these have been discussed in details in chapter 1, section 1.2.1. (Other derivations for both the EEG and the EOGs were also sometimes used, as will be pointed out later). It is perhaps worth pointing out that the electrode placement 7.3(c) can be derived from that of 7.3(b) (as indeed most of those given in figure 1.3 in chapter 1) by merely selecting the appropriate pairs of EOG electrodes at the EEG machine. Thus, it was only necessary to apply the EOG electrodes as shown in figure 7.3(b).

The EEG and EOG signals were fed into an eight-channel EEG machine via the head box, and after amplification they were fed into the OAR system via a 37-way D-type connector (see appendix A8). After removing the OA from the EEG signals, both the corrected and raw EEGs (with the means removed) and/or the EOGs were fed into the final amplifiers of the EEG machine and thence to the paper chart for examination (see figure 7.3a). It should be noted that the first four channels of the EEG machine were

Figure 7.3 (a) the OAR system connections to the EEG machine, (b) and (c) the EOG placements used.



(b) -(FP2-T4 is VR)
-(F8-T5 is HR)
-(Fp1-T3 is VL)
-(F7-T5 is HL)

(c) - (Fp2-F8 is EOGR)
- (Fp1-F7 is EOGL)

The head box showing some of the EEG channels and the type of derivations used. The letters on the headbox indicate the areas on the scalp where the electrodes are placed; F indicates frontal, C central, P parietal, O occipital and T temporal.

reserved for the EOG signals so that only four EEG signals could be applied to the EEG machine simultaneously. Furthermore, only six input and output signal conditioners were fitted to the OAR system , so that effectively only two EEG signals could be fed to the OAR system at the same time. The two EEG signals were selectable using the channel-selection switches on the EEG machine. Additional EEG signals can be accommodated by using a bigger EEG machine and by fitting more signal conditioners. However, since the OAR system software can only remove OA from a limited number of EEG signals at the same time due to the slow speed of the floating point arithmetic software these expansions were considered unnecessary at the present time (see later).

(b) Models Used in the Tests.

Four models were used in the tests. Three of these utilized the EOGs derived from electrode placement of figure 7.3(b) and were described in chapters 2 and 6. These three models are reproduced below for easy reference:

2D $y(i) = \theta_1 VR(i) + \theta_2 HR(i) + e(i)$

3D $y(i) = \theta_1 VR(i) + \theta_2 HR(i) + \theta_3 HL(i) + e(i)$

4D* $y(i) = \theta_1 VR(i) + \theta_2 HR(i) + \theta_3 HL(i) + \theta_4 HL(i) x HR(i) + e(i)$

(7.1)

The fourth model, which will be called model 2H in line with the nomenclature of chapter 2 , used the EOGs derived from the electrode placement of figure 7.3(c):

2H $y(i) = \theta_1 EOGR(i) + \theta_2 EOGL + e(i)$ (7.2)

where EOGR(i) and EOGL(i) are the right and left eye EOGs.

(c) Ocular Artefact Removal

With the OAR system connected to the subject via the EEG machine and the head box as shown in figure 7.3(q), the OAR program was started . After entering the various system constants (including the desired model) as described in chapter 6, section 6.1, the subject was asked to carry out various ocular movements, reminiscent of the experiments described in section 1.6. (For the unco-operative subjects, the OAR system was simply used to remove the random OAs). It should be noted that if a two-parameter model was specified, then either model 2D or model 2H was used to remove the OA, depending on the pairs of EOG electrodes that were selected to feed channels 1 and 2 of the EEG machine (see figure 7.3(b)). In fact, the selection switch on the EEG machine can be used to 'force' the OAR system to implement a variety of models by selecting the appropriate pairs of EOG electrodes to feed channels 1 to 4.

The OAs in the EEGs were then removed on-line using the U-D factorization algorithm described in chapters 4 and 6.

7.2.2 Experimental Results with Normal Subjects

(a) Comparison of Models

Figure 7.4 gives typical results for models 2H, 3D, and 4D* for vertical eye movement experiments at the FP2-F4 EEG electrodes. This is one of the EEG electrodes that are most susceptible to OA. One model was used at a time to remove the OA while the subject moved his eyes vertically. It is seen that all three models gave satisfactory removal of OA. Compare for example the corrected and uncorrected EEGs i.e traces (iii) and (iv) in each of figures 7.4(a), (b) and (c). However, it is seen that the rider artefacts were not completely removed. Satisfactory OA removal was also obtained for the other types of OA. Figure 7.5 gives the results for a lateral or horizontal eye movement experiment at the same electrodes.

It was found that in all cases, model 2H had a performance that was as good as the other models. As this model is simpler than either model 3D or 4D* (cf equations 7.1 and 7.2) and uses fewer EOG electrodes, it was decided that it should be used in subsequent investigation in preference to models 3D and 4D*. Figure 7.6 gives additional results for the model 2H at FP2-F4 electrodes for periodic blinks and random ocular movements. These results also showed that the OAR system satisfactorily removed the OAs.

Electrode placement 1.3(k) in chapter 1, was tried but gave an unsatisfactory results, especially for blinks.

Figure 7.4 Comparison of models at the FP2-F4 electrodes for VEM experiments.

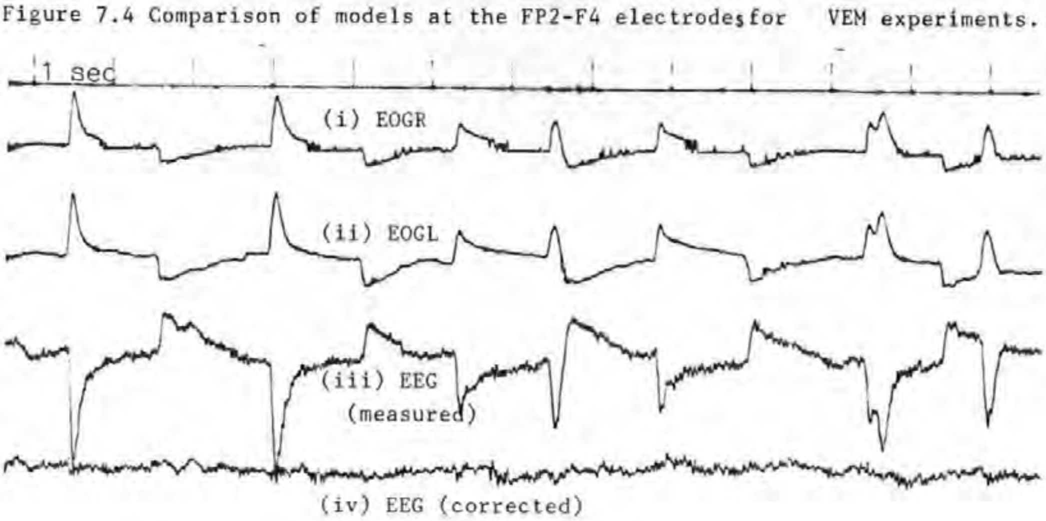


Figure 7.4(a) Model 2H
(i) The right eye EOG (EOGR) (ii) The left eye EOG (EOGL)
(iii) Measured EEG (iv) Corrected EEG

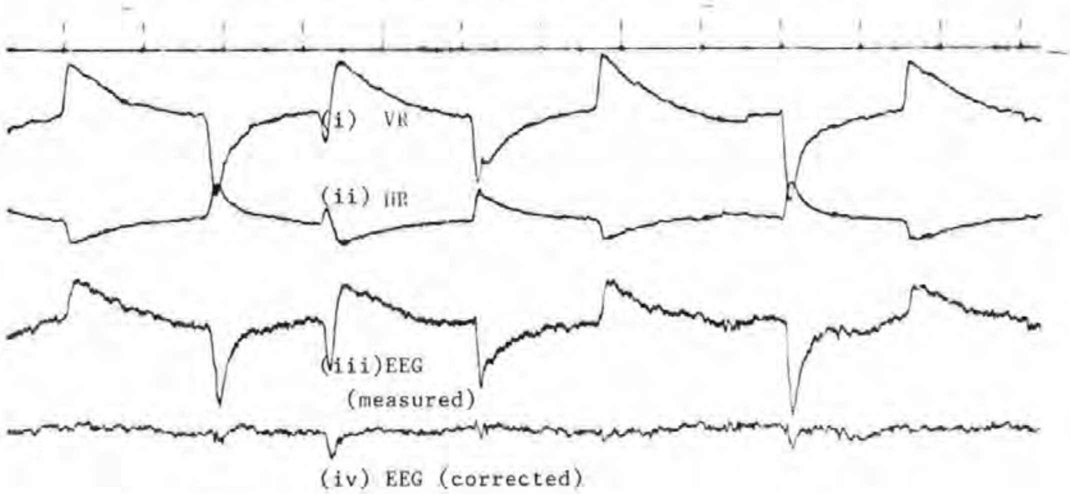


Figure 7.4(b) Model 3D
(i) The vertical right EOG (ii)The horizontal right EOG
(iii) Measured EEG (iv) Corrected EEG

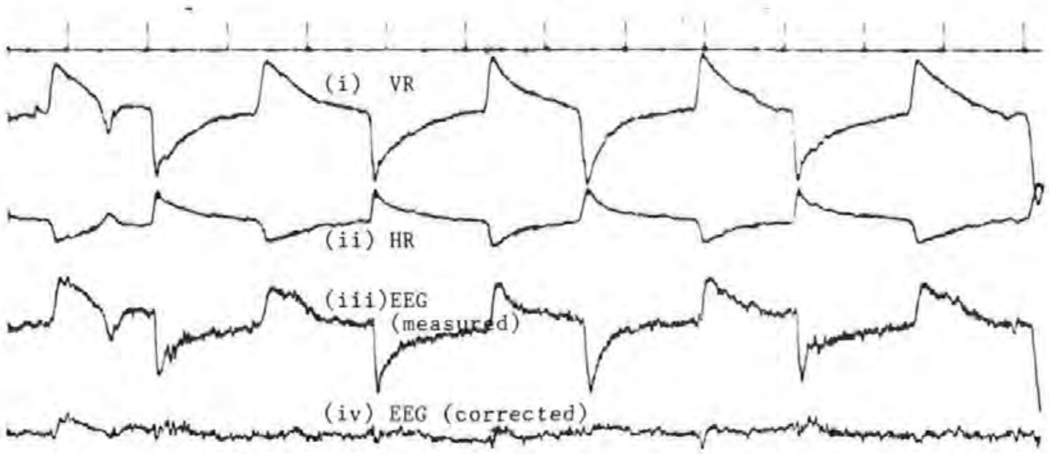


Figure 7.4(c) Model 4D*
(i) The vertical right EOG (ii) The horizontal right EOG
(iii) Measured EEG (iv) Corrected EEG

Figure 7.5 Comparison of models at the Fp2-F4 electrodes for HEM experiments.

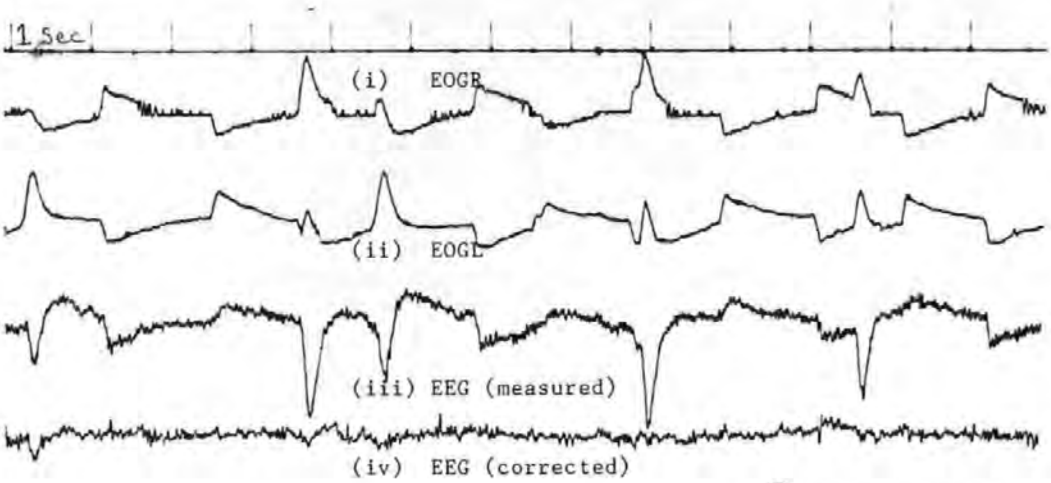


Figure 7.5(a) Model 2H
(i) The right eye EOG (ii) The left eye EOG
(iii) Measured EEG (iv) Corrected EEG

200 µV (EOGs)
70 µV (EOGs)

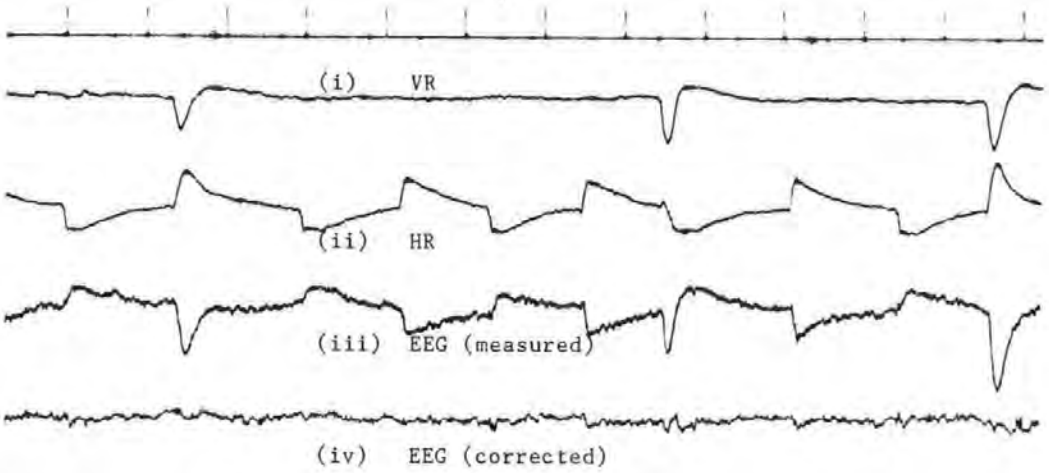


Figure 7.5(b) Model 3D
(i)The vertical right EOG (ii) The horizontal right EOG
(iii) Measured EEG (iv) Corrected EEG

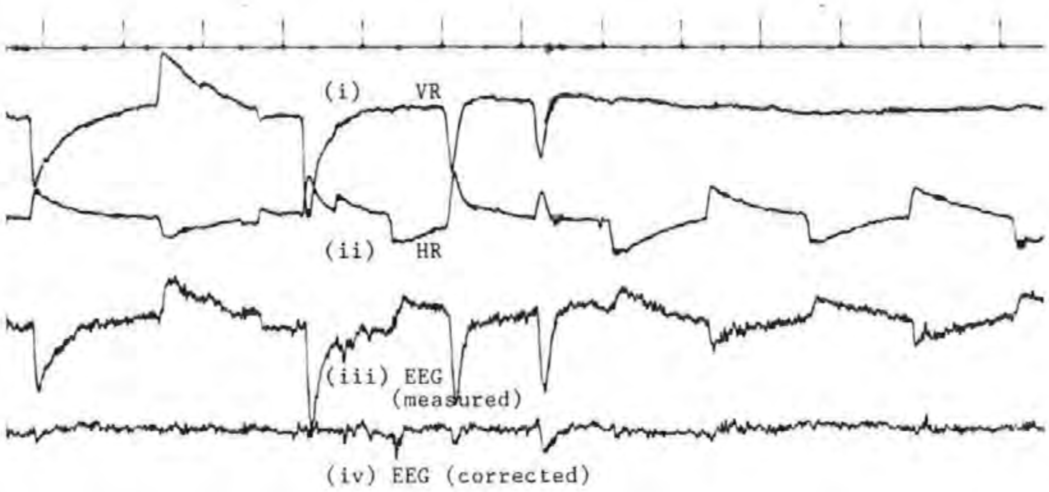


Figure 7.5(c) Model 4D*
(i) The vertical right EOG (ii) The horizontal right EOG
(iii) Measured EEG (iv) Corrected EEG

Figure 7.6 Ocular artefact Removal at the Fp2-F4 electrodes using model 2H for (a) periodic blinks (b) random ocular movements.

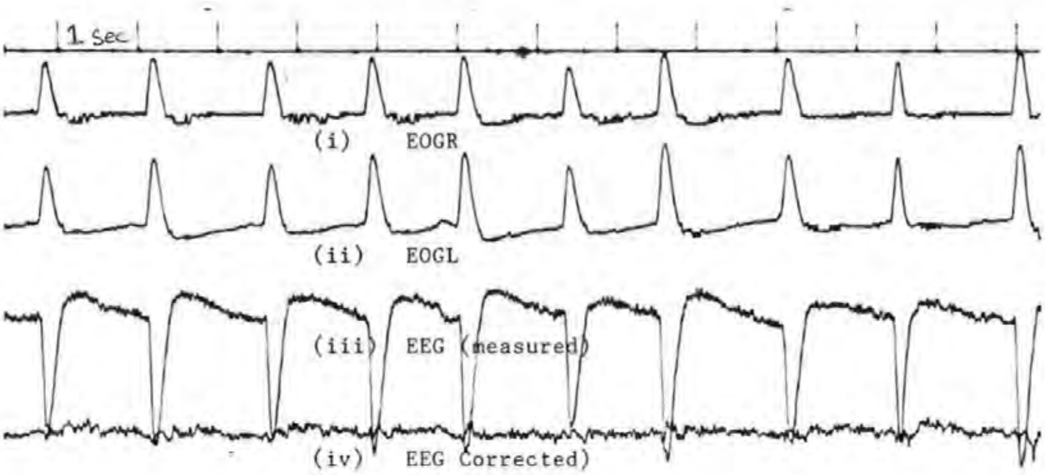


Figure 7.6(a) Periodic blinks
(i) The right eye EOG (ii) The left eye EOG
(iii) Measured EEG (iv) Corrected EEG

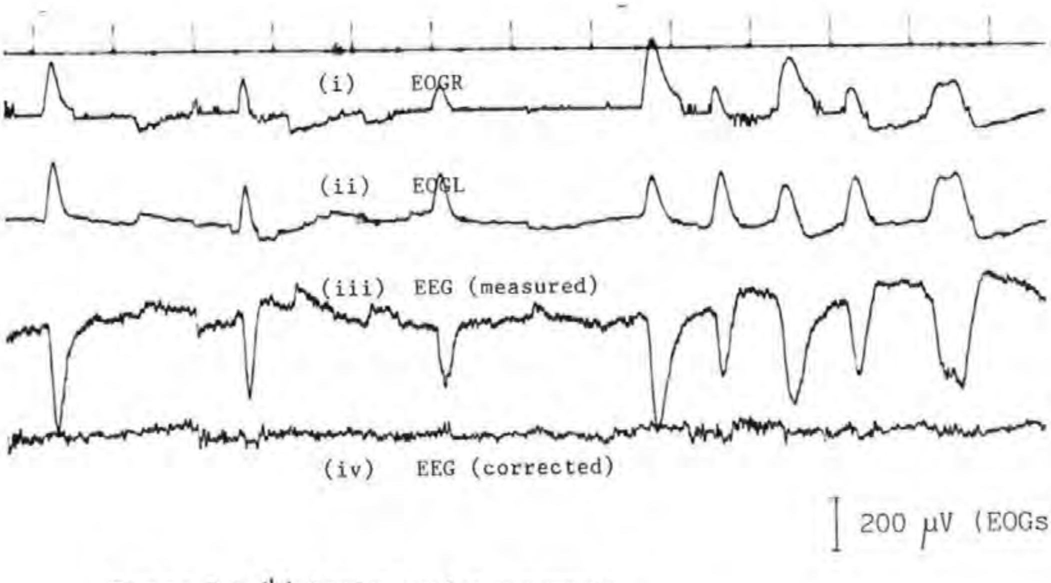


Figure 7.6 (b) Random ocular movements
(i) The right eye EOG (ii) The left eye EOG
(iii) Measured EEG (iv) Corrected EEG

Model 2D did not give complete OA removal at the FP2-F4 electrodes, especially for large ocular movements (see for example, figure 7.7).

(b) Evaluation of the OAR System Performance at different EEG electrodes.

The OAR system was used to remove OAs from a number of other EEG electrodes (see figure 7.3(a)) using model 2H (and occasionally models 3D and 4D*). It was found that in all cases the OA was satisfactorily removed, and this included all the frontal EEG channels where the OA was largest. Figures 7.8(a) and (b) give results for four different electrodes (Fz-Cz, Cz referred to the earlobe, F4-C4, and F3-C3) for blink experiments. In both figures two different EEG signals were simultaneously corrected for OA. Comparison of the corrected and uncorrected EEGs in both sets of figures showed that the system had satisfactorily removed the OAS. (compare traces (v) and (vi) with traces (iii) and (iv) in each set of figures)

There was little OA contamination at the more posteriorly placed EEG electrodes (e.g Cz-Pz and C4-P4). In these cases all the models performed equally well.

(c) The effects of OA removal on the background EEG

To assess the effects of OA removal on the background or true EEG, subjects were asked to keep their eyes as 'still as possible', so that there was effectively no OA in the EEG. The raw and 'corrected' EEGs were then compared and in all cases there appeared to be no significant

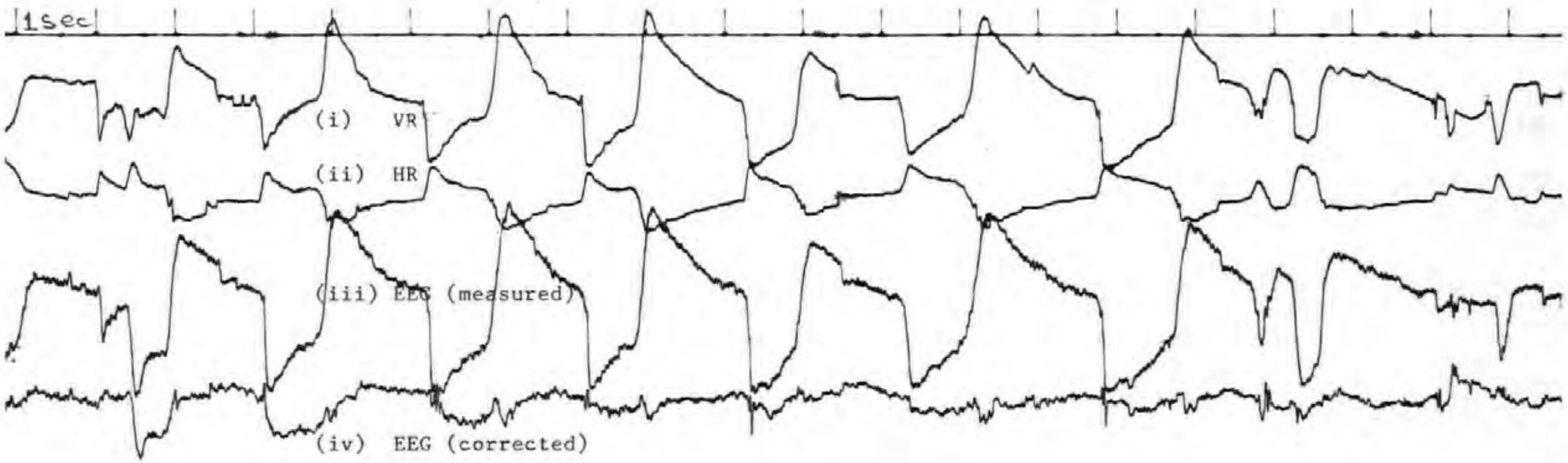


Figure 7.7 Ocular artefact removal at the Fp2-F4 electrodes using model 2D for a VEM experiment.
(i) The vertical right EOG (ii) The horizontal right EOG
(iii) The measured EEG (iv) The corrected EEG

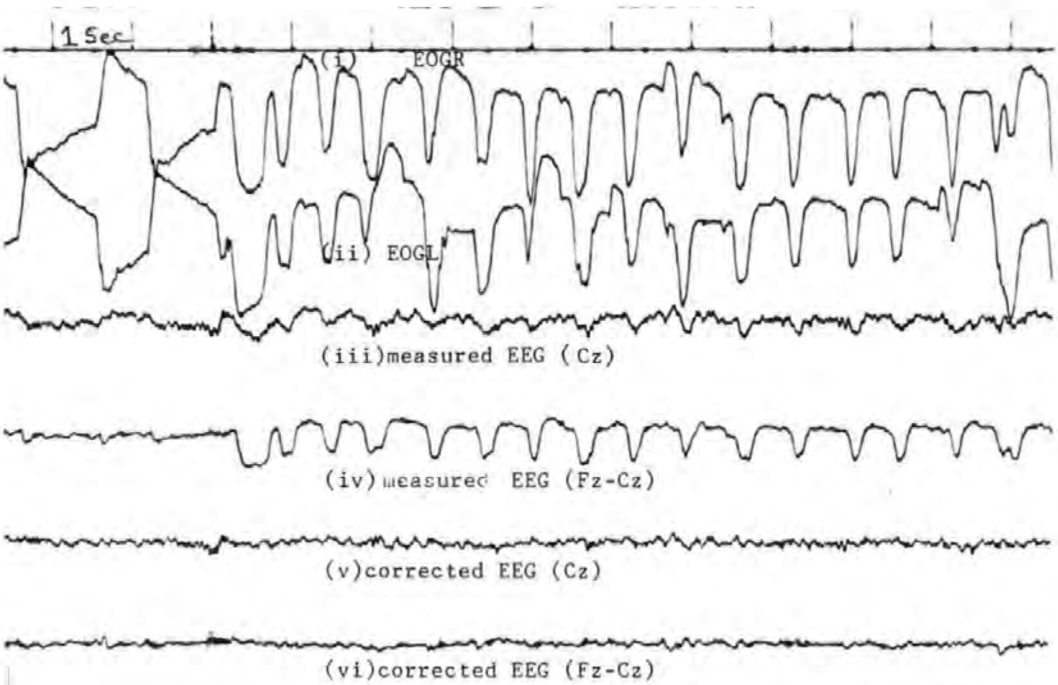


Figure 7.8 (a) Multichannel EEG ocular artefact removal at the Cz and Fz-Cz electrodes.

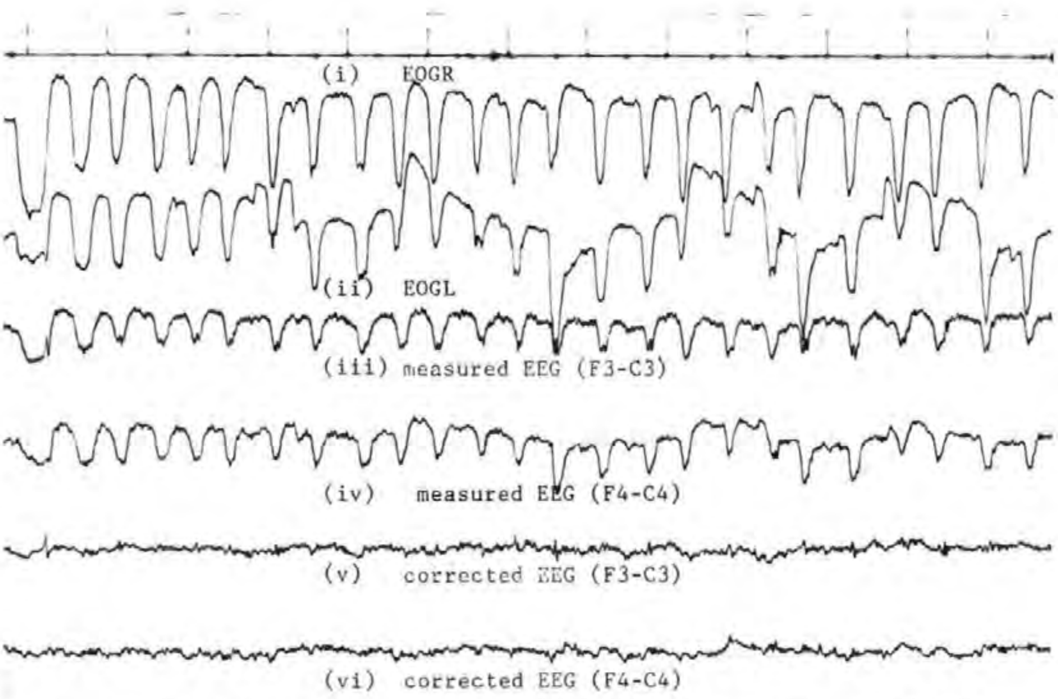


Figure 7.8 (b) Multichannel EEG ocular artefact removal at the F3-C3 and F4-C4 electrodes/

150 µV

differences between them, e.g. figure 7.9. This suggested that the OAR system does not have any serious adverse effects on the true EEG.

(d) Simulation of the Previous On-line OA Removal Methods

In the investigation reported above, it was necessary to use a lower sampling frequency (95Hz) than the target specification (128Hz) when model 4D* was employed, or when more than one or two EEG signals were to be simultaneously corrected for OA using the simpler models. This limitation was due to the floating point arithmetic routines used which were too slow to perform the large number of arithmetic operations involved in these cases within the sampling interval. Table 7.2 sets out the maximum number of EEG signals that can be simultaneously corrected for OA at each of the two sampling frequencies. (The sampling frequency can be easily changed in software as described in appendix A15.)

Table 7.2 Maximum number of EEG signals that can be corrected for OA simultaneously at the two sampling frequencies used.

model	95Hz	128Hz
2D or 2H	4	2
3D	2	1
4D*	1	0

It should be pointed out that no significant differences could be seen when the results at the two frequencies were compared. For this reason most of the tests were carried out at the lower sampling frequency, including all those

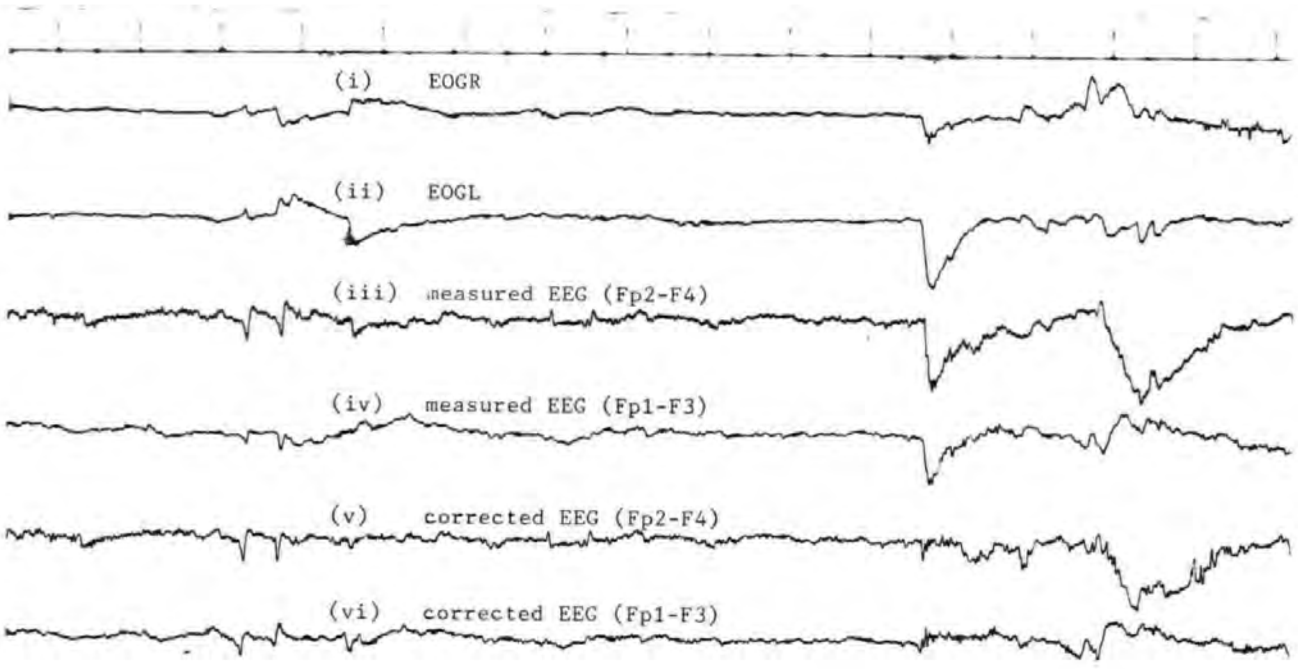


Figure 7.9 Effects of OA removal on the background EEG at the Fp2-F4 and Fp1-F3 electrodes using model 2H.
(i) The right eye EOG (ii) The left eye EOG
(iii) The measured EEG (Fp2-F4) (iv) The measured EEG (Fp1-F3)
(v) and (vi) The corresponding corrected EEGs

described in this chapter.

The number of EEG signals that can be corrected for OA simultaneously can be significantly increased by using a suitable hardware floating point (FP) arithmetic devices, such as the Weitek devices [4].

As described in chapter 4 section 4.4, another way of avoiding the limitations imposed by the slow speed of the FP routines is to use the fixed parameter approach. As mentioned in chapter 4, this approach can be viewed as an automated form of the previous on-line methods [6-8]. Thus the fixed parameter approach is a two-step OA removal method. In step 1 or calibration stage, estimates of the OA parameters are obtained whilst subjects are performing a given ocular movement, and in step 2 or removal stage the parameter estimates are used to remove the OAs on-line. In chapter 4, an off-line technique was suggested as a way of obtaining the parameter estimates required in step 1. An alternative approach is to use the on-line routines of the OAR system to estimate the parameters. This approach was used to simulate the previous on-line methods. Thus the OAR system's on-line program was started as described in chapter 6, and the subject was asked to perform a given ocular movement. The program was then stopped after several seconds, and the resulting parameter estimates used in the second step of the algorithm. (A special program of only a few lines of codes was used for the removal stage, see appendix A15).

Figure 7.10 (a) to (c) show results for the fixed parameter algorithm. In these cases, estimates of θ were obtained whilst the subject was performing vertical eye movement experiment. After the parameter estimates were obtained the subject was then asked to perform horizontal-, blinks-, and vertical ocular movements, and in each case the same parameter estimates were used to remove the OAs. As Examination of these figures showed, the OA removal in all cases was not satisfactory although it is seen that the system performed best for the vertical eyemovement (figure 7.10(a)). These results demonstrated that parameter estimates obtained for one type of ocular movement cannot give adequate OA removal for other types of ocular movements.

7.2.3 Experimental Results with Patient Subjects

In the results given in the preceding sections, the raw EEGs were in fact partially processed (e.g. low pass filtered, sampled, mean-corrected, converted to digital etc). For the patients, it was considered necessary to record the raw unprocessed EEG on to the paper chart along with the corrected EEG to allow an unbiased analysis of the EEG.

Thus the raw EEG to be corrected was fed direct to the

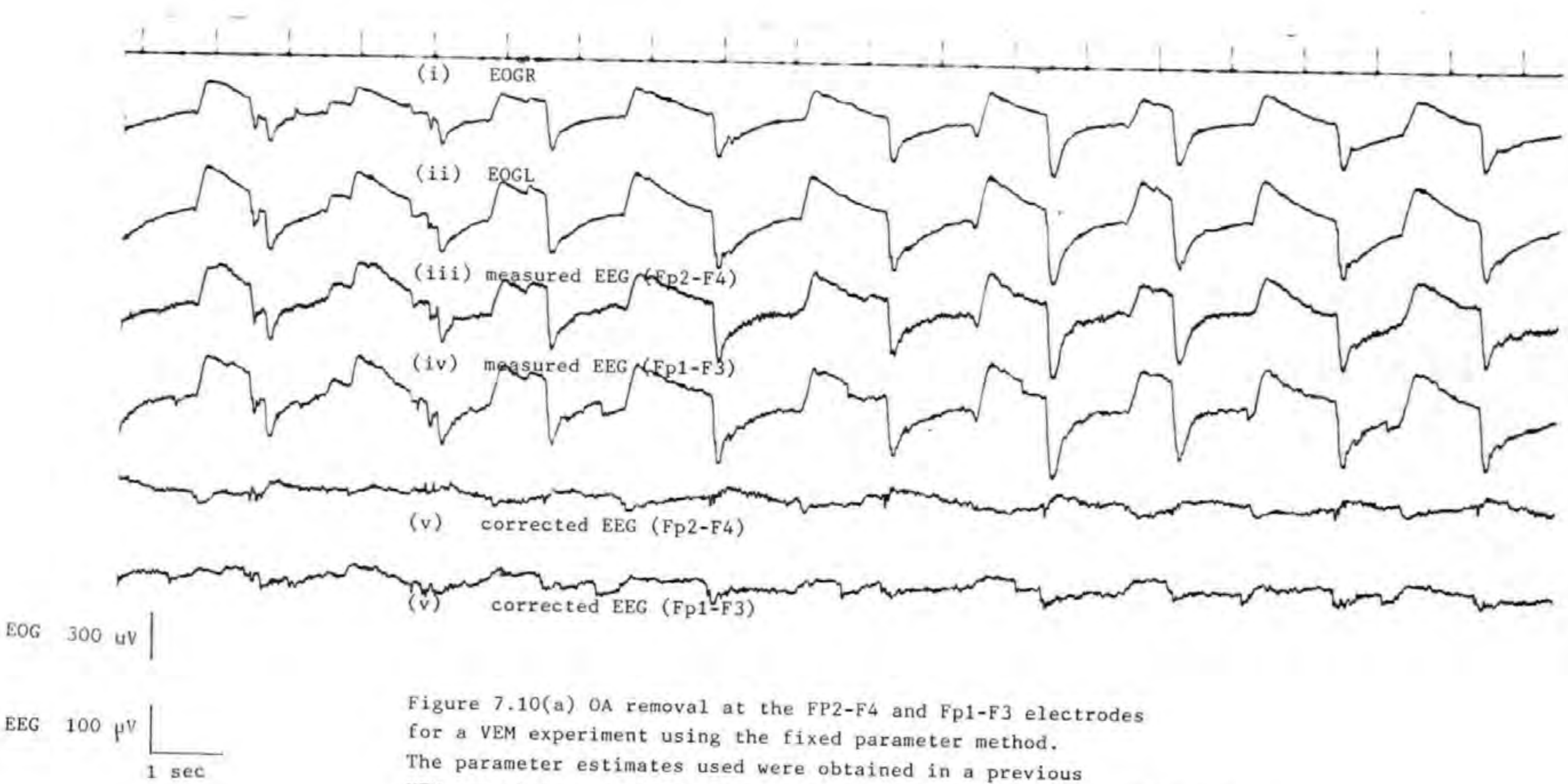


Figure 7.10(a) OA removal at the FP2-F4 and Fp1-F3 electrodes for a VEM experiment using the fixed parameter method. The parameter estimates used were obtained in a previous VEM experiment. Details of the traces are the same as for figure 7.9.

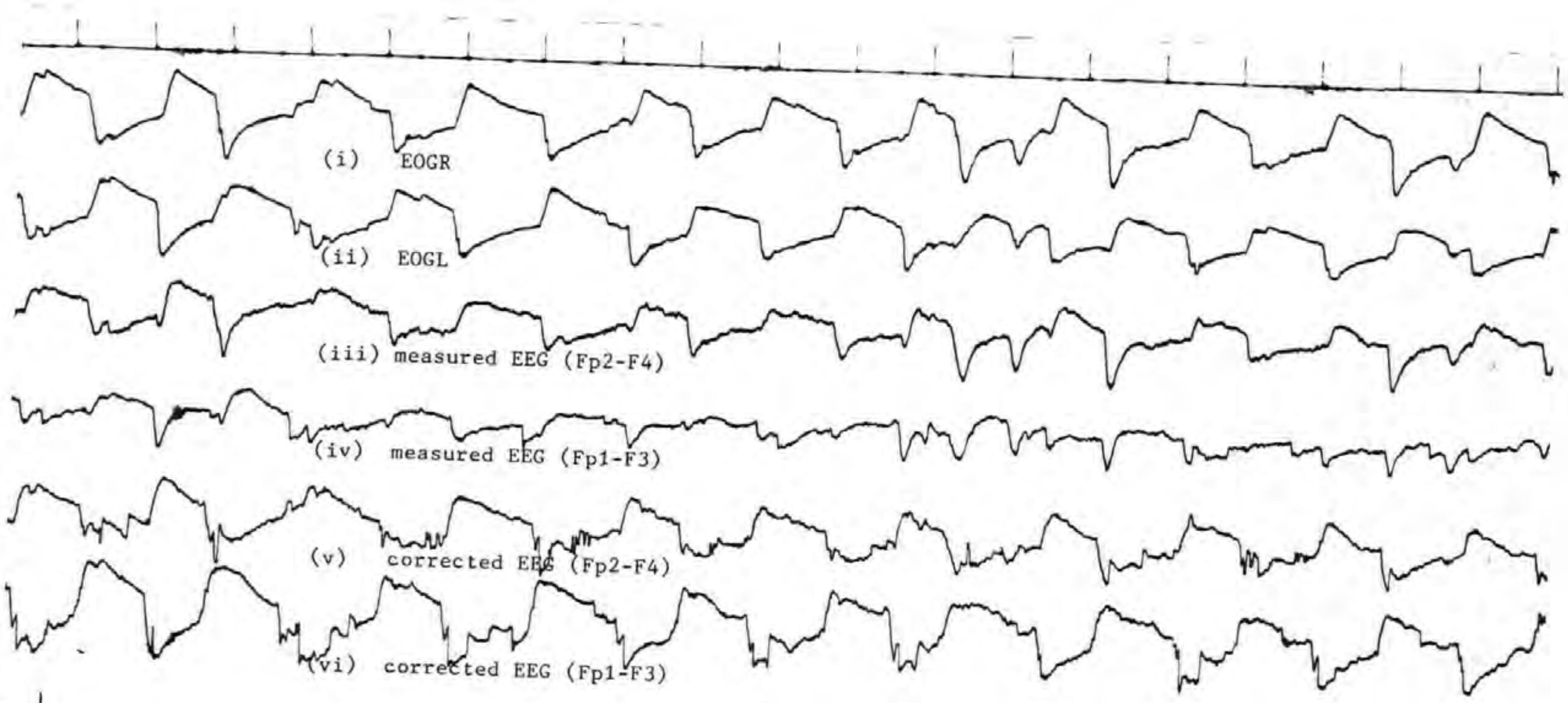


Figure 7.10(b) OA removal at the Fp2-F4 and Fp1-F3 electrodes for a HEM experiment using the fixed parameter method.
The parameter estimates used and details of the traces are the same as for figure 7.10(a).

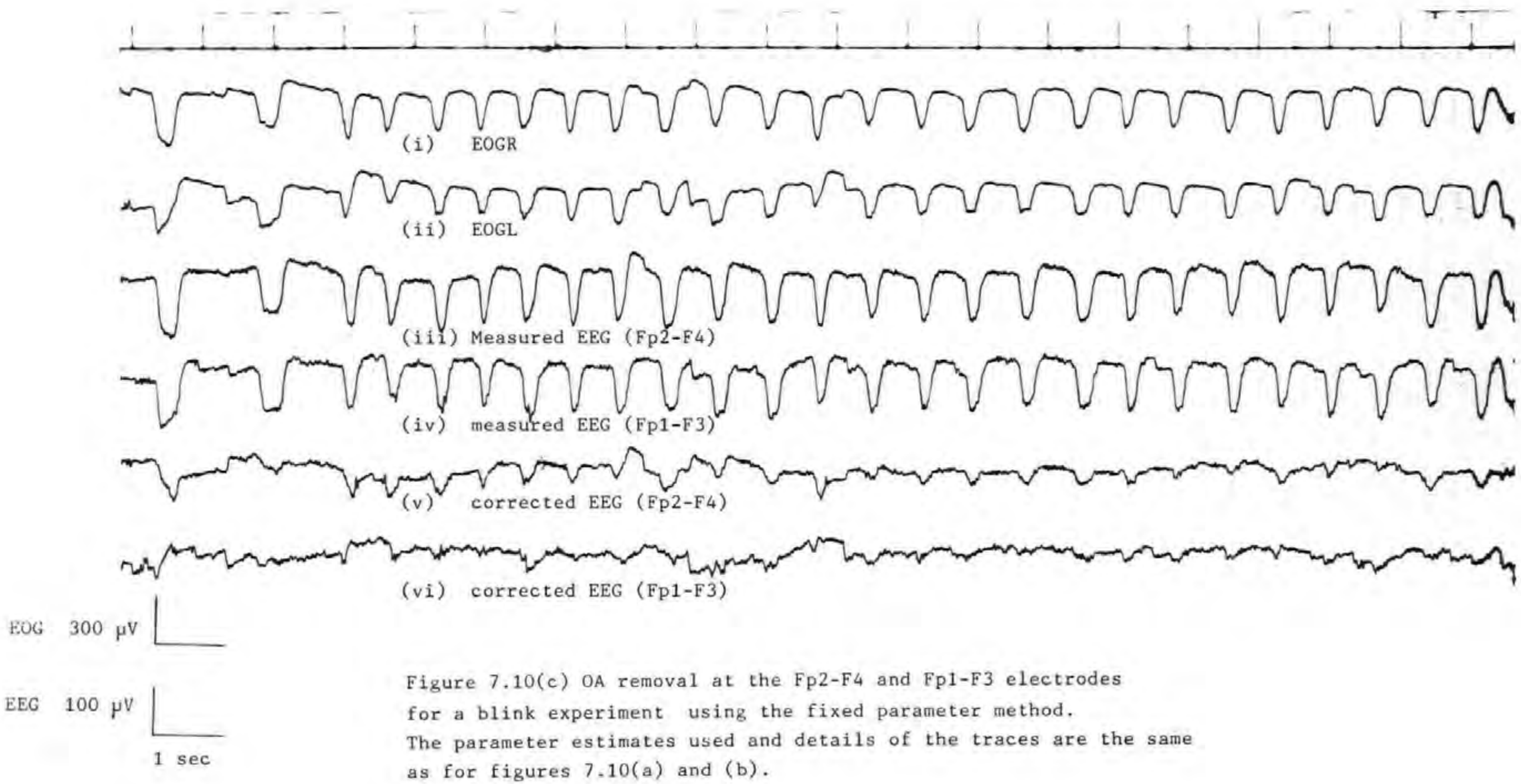


Figure 7.10(c) OA removal at the Fp2-F4 and Fp1-F3 electrodes for a blink experiment using the fixed parameter method. The parameter estimates used and details of the traces are the same as for figures 7.10(a) and (b).

paper charts via a spare channel as well as to the OAR system as described earlier.

(a) Epileptic Subject

This 28-year old female subject suffered from chronic epilepsy and could not perform the ocular movement experiments successfully. Even when she was aided in the experiments by gently holding her head to prevent head movements and with instructions such as 'look at my head, look at my foot' to get her to perform vertical eye movements, her ocular movements were still uncontrolled.

Figure 7.11 shows the result for a horizontal eye movement experiment. An examination of the EOGs (traces (i) and (ii)) showed that they contained blinks as well as some lateral eye movement and perhaps other types of ocular movements as well, showing that the subject could not control her ocular movements. However, it is seen from a comparison of traces (iii) or (v) with (iv) that the OAs have been satisfactorily removed.

A problem with patients in this category is that they exhibit slow waves which could be confused for OA as well as the characteristic epileptic spikes. It is desirable to remove the OAs from their EEGs without adversely affecting the waves or the spikes.

Figures 7.12 shows an EEG record for the patient which contains the epileptic spikes and slow waves. Comparison

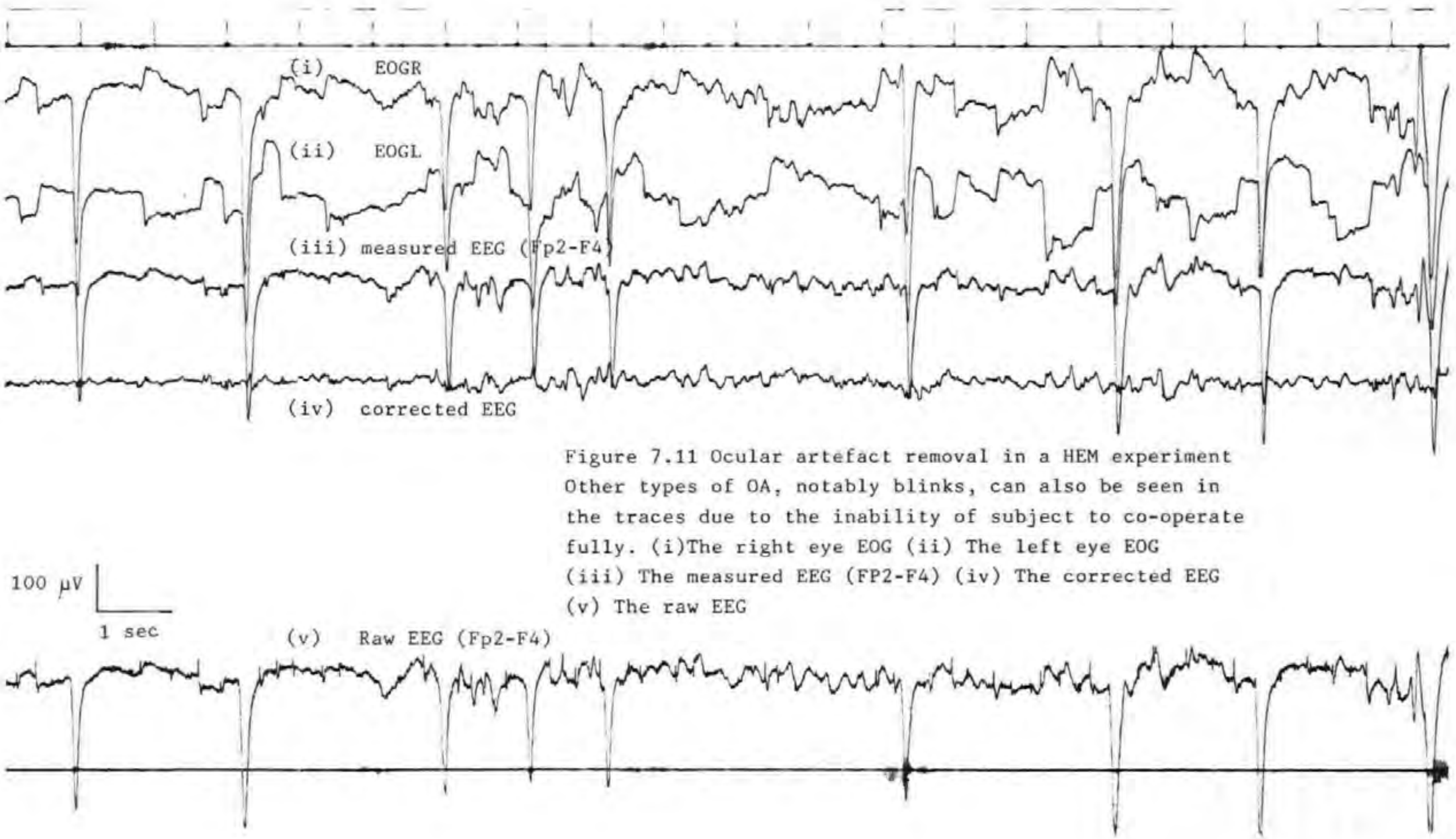


Figure 7.11 Ocular artefact removal in a HEM experiment Other types of OA, notably blinks, can also be seen in the traces due to the inability of subject to co-operate fully. (i)The right eye EOG (ii) The left eye EOG (iii) The measured EEG (FP2-F4) (iv) The corrected EEG (v) The raw EEG

of the corrected EEG (trace (iv)) and the raw EEG (trace (v)) showed that the OAs have been removed, but not the spikes and waves. Another example is given in figure 7.13. Again this showed a satisfactory performance.

In some instances, the EOG electrode pairs of figure 7.3(c) which were used for model 2H may be used as EEG channels. It was therefore decided to verify how the OAR system would behave at these sites which are right at the usual EOG electrodes. Thus the EEG electrode pair F1-F7 was applied to the OAR system as an EEG channel and model 3D used to remove the OA. Figure 7.14 shows the result in this case. Examination of the corrected EEG showed that the spike and wave complex have been substantially reduced (figure 7.14(iv)). It is seen that the vertical EOG (figure 7.14(i)) and the measured EEG (figure 7.14(iii) or (v)) were essentially the same because of the proximity of the EEG site to the eye, and this was why the spikes and waves have been reduced. This result suggested that it may not be wise to record EEG signals from sites that are as close to the eyes as the F1-F7 electrodes.

(a) Patient with Mental Illness

This patient could not co-operate and exhibited slow waves. To enhance the slow waves he was occasionally asked to keep his eyes still but this met with limited success.

Figure 7.15 shows results for this patient. The slow

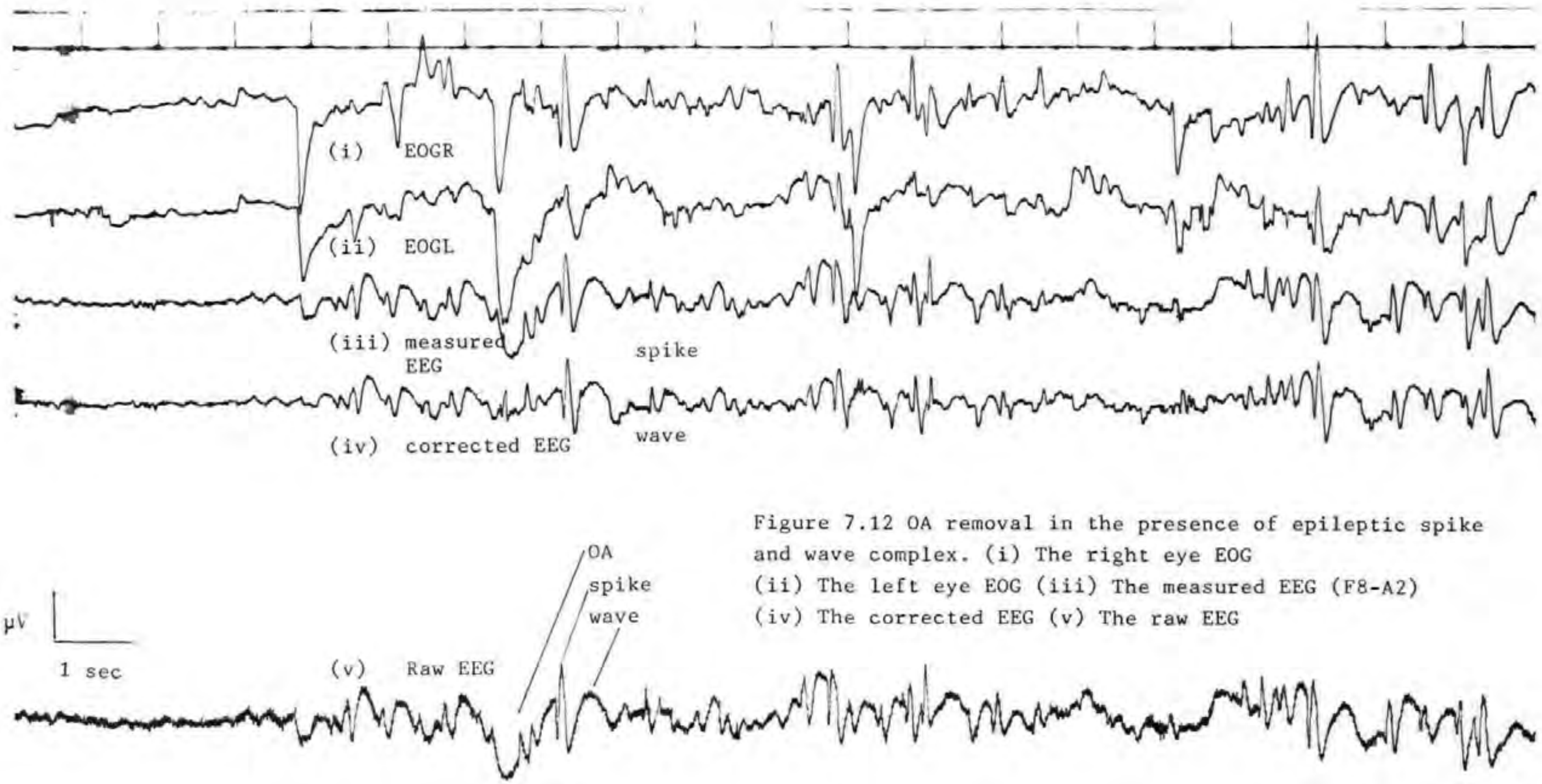


Figure 7.12 OA removal in the presence of epileptic spike and wave complex. (i) The right eye EOG (ii) The left eye EOG (iii) The measured EEG (F8-A2) (iv) The corrected EEG (v) The raw EEG

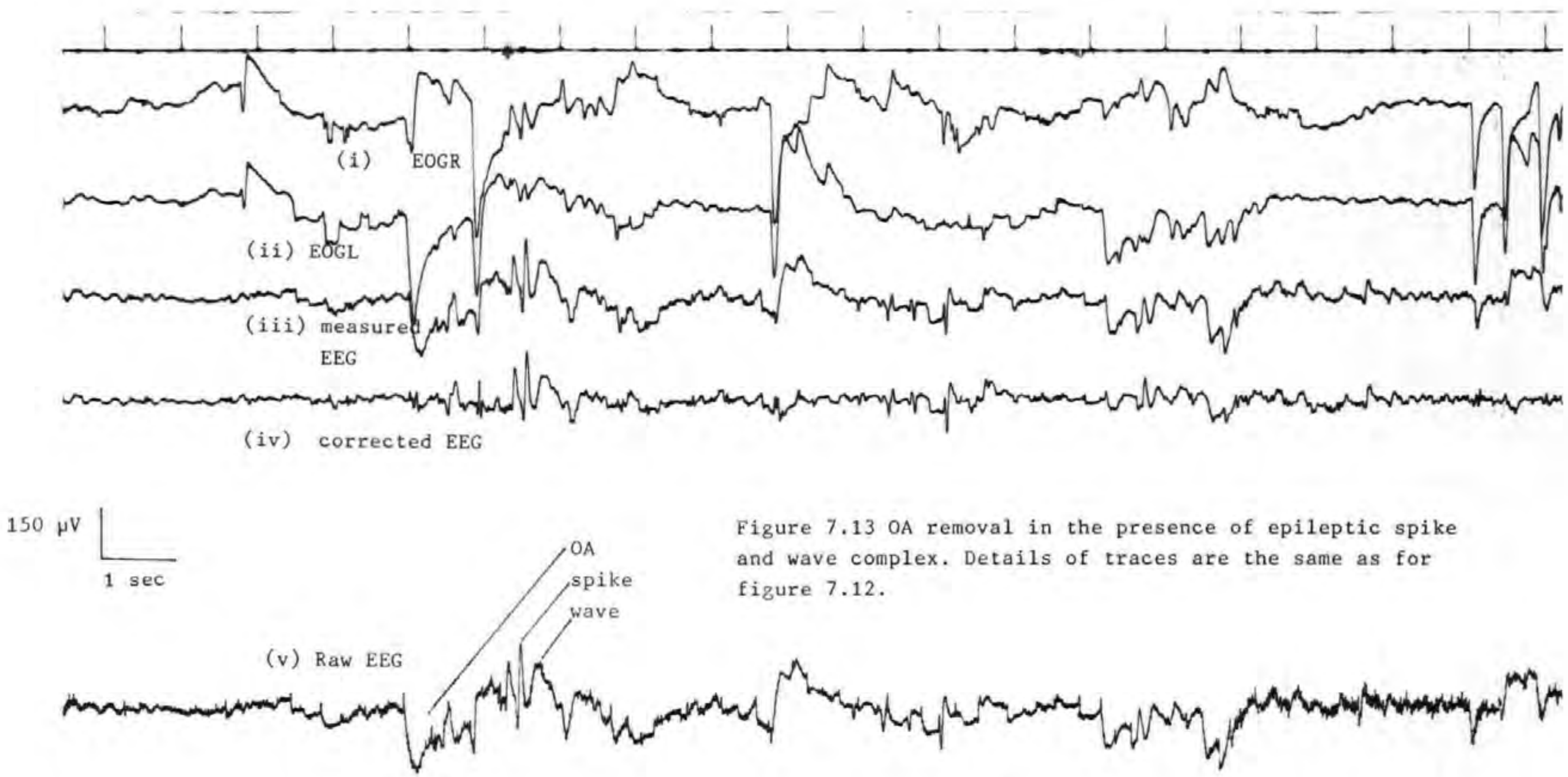


Figure 7.13 OA removal in the presence of epileptic spike and wave complex. Details of traces are the same as for figure 7.12.

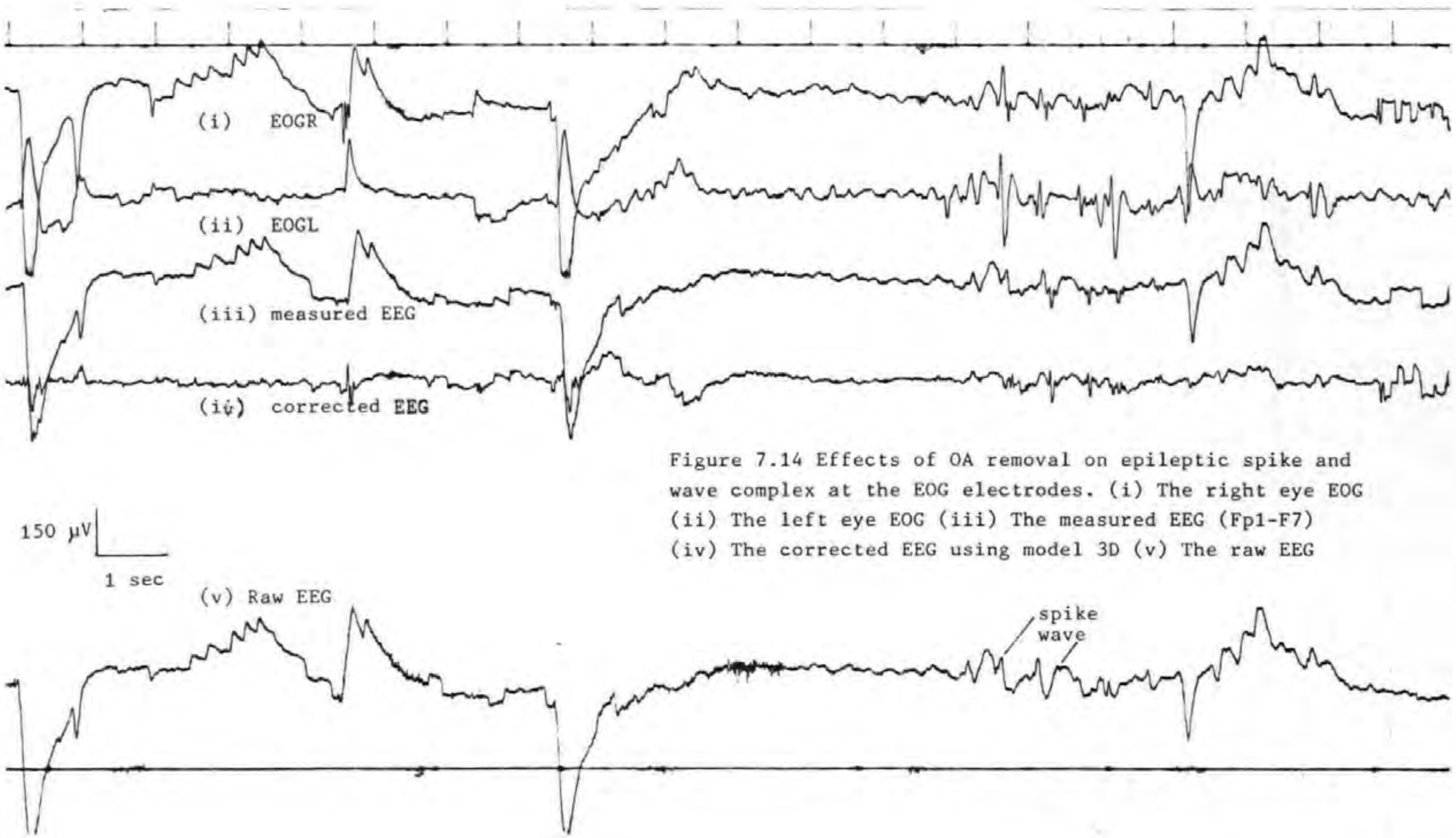


Figure 7.14 Effects of OA removal on epileptic spike and wave complex at the EOG electrodes. (i) The right eye EOG (ii) The left eye EOG (iii) The measured EEG (Fp1-F7) (iv) The corrected EEG using model 3D (v) The raw EEG

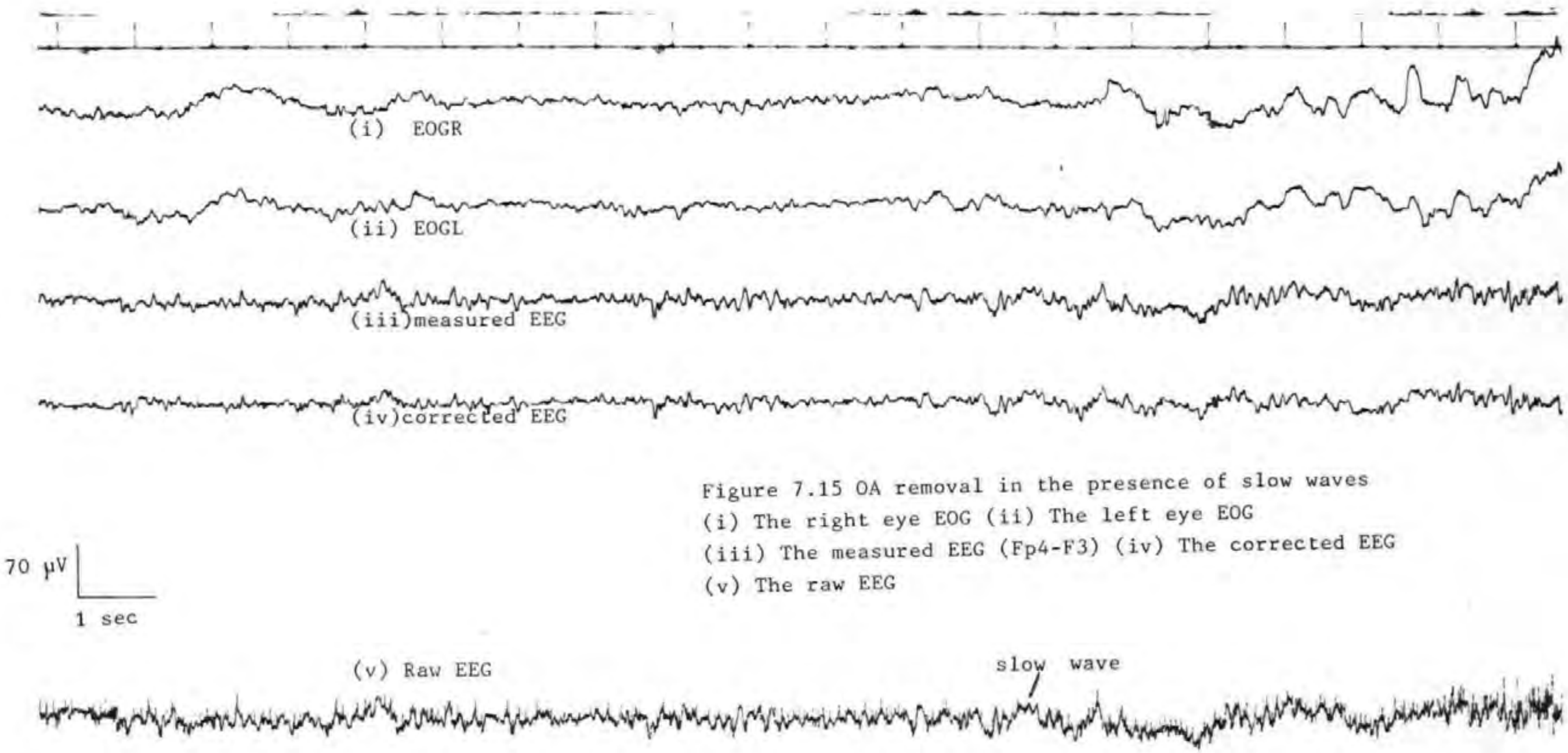


Figure 7.15 OA removal in the presence of slow waves
(i) The right eye EOG (ii) The left eye EOG
(iii) The measured EEG (Fp4-F3) (iv) The corrected EEG
(v) The raw EEG

waves in the EEG appeared to have been reduced after correction, but not removed.

7.2.4 Discussion of Results

(a) Normal Subjects

The results of the various tests showed that in most cases the models defined in section 7.2 .1 gave satisfactory OA removal. Model 2D (which used VR,HR) in some cases, especially for large ocular movements did not give satisfactory removal of OA in the frontal channels. This agrees with the results of Fortgens and De Bruin [5] who concluded that more than two EOGs were required to achieve satisfactory OA removal at the frontal channels. However, the good performance of model 2H (which used only two EOGs) suggested that the reason model 2D failed was probably because it failed to take into account the contribution of both eyes to the OA in the EEG and not because it used only two EOGs.

The tests also showed that it is possible to achieve satisfactory on-line OA removal at all EEG sites. It was found that in the more posterior EEG sites, very little OA contamination of the EEG was observed and in these cases all the models performed well.

In the absence of OA the background EEG did not suffer any serious adverse effects by the OA removal process. However, the effects of OA removal on the true EEG depends largely on the secondary artefact in the EOG channels, and this may vary from subject to subject.

(b) Patients

The results showed that when pathological slow waves and spikes occur in the absence of OA, they were in general not significantly affected by the OA removal process. However, when they ocur simultaneously with the OA, they may be reduced in amplitude, but not removed completely. The reduction in amplitude occurs mainly at the frontal EEG channels.

A simulation of the previously published on-line methods showed that in general they were inferior to the OAR system. It is perhaps worth mentioning that the un-coperative patients used in the test would have been unsuitable for use in the previous on-line methods where subjects' co-operation is necessary (see section 1.4).

7.4 Summary

In this chapter, tests used to evaluate the OAR system were discussed. These included simulated tests, and tests on normal subjects as well as patient subjects exhibiting pathological slow waves. The tests showed that in general the OAR system gave satisfactory OA removal.

Tests on normal subjects showed that the OAR system satisfactorily removed the OA at various EEG sites, but suggested that to obtain satisfactory OA removal in the frontal EEG channels account should be taken of the EOGs of both eyes in the OA model.

Tests on patients indicated that at the frontal channels when pathological slow waves and spikes occur simultaneaously with the OA, the OA removal process could reduce the amplitudes of the slow waves and the spikes. At the more posterior channels or when the waves occur in the absence of the OA they are in general insignificantly affected by the OA process.

References for Chapter 7

[1] MILLMAN J. and TAUB H.
Pulse, Digital, and Switching Waveforms. McGraw-Hill, 1965.

[2] TAUB H. and SCHILLING D.
Digital Integrated Electronics. McGraw-Hill, 1977.

[3] TOMPKINS, W.J., WEBSTER, J.G. (ed)
Design of Microcomputer-Based Medical Instrumentation. Prentice-Hall, 1981.

[4] WOO B.,LIN L., and OWEN R.E.
ALU, Multiplier chips zip through IEEE Floating-Point Operations. Electronics, 19,1983,pp.

[5] FORTGENS C. and DE BRUIN M.P.
Removal of Eye Movement and ECG Artifacts from the Non-cephalic Reference EEG.' Electroenceph. and Clin. Neurophysiol., 1983, 56, pp90-96.

[6] BARLOW J.S. and RÉMOND A.
Eye Movement Artifact Nulling in EEGs by Multichannel on-line EOG Subtraction. Electroenceph. and Clin. Neurophysiol.,1981,55,pp. 418-423.

[7] GIRTON D.G. and KAMIYA J.
A simple On-line Technique for Removing Eye Movement

Artifacts from the EEG. Electroenceph. Clin. Neurophysiol., 1973,34,pp. 212-216.

[8] McCALLUM W.C. and WALTER W.G.
The Effects of Attention and Distraction on the Contingent Negative Variation in Normal and Neurotic Subjects. Electroenceph. Clin. Neurophysiol.,1968,25,pp. 319-329.

Chapter 8 REVIEW, FUTURE WORK AND CONCLUSIONS

8.0 Introduction

The work described in this thesis is concerned with the problems of ocular artefacts in the human EEG, their removal both off-line and on-line, and the design and development of an on-line ocular artefact removal system. This chapter contains a review of the main points in the thesis, suggestions for future work, and conclusions.

8.1 Review

(a) Off-line Removal of Ocular Artefact

As discussed in chapter 1, ocular artefacts in the EEG resulting from movements in the ocular system (eyeball,eyelids etc.), need to be removed so that the true EEG can be analyzed. The EOG subtraction method offers the best solution because it does not lead to wastage of data and gives satisfactory ocular artefact removal. On-line removal of OA is preferred because the delay involved when off-line techniques are employed is unnacceptable, especially in real-time applications.

In chapter 2, the work of Jervis et al [1,2] and Fortgens and De Bruin [3] was extended to find the most effective EOG subtraction model for implementation on-line. A number of criteria were devised, including the more reliable pictorial (waveform subtraction) method, and used to compare the performance of several models. It was found that there was no need to use four EOG channels, as suggested by Jervis et al [1,2] and Fortgens and De Bruin [3] because of the strong correlation between the EOGs. The EOGs of the right eye were found to be more important in removing OA than those of the left eye. The best model overall used the vertical right EOG (VR), the two horizontal EOGs (HL and HR), and their product (HL x HR). The best three-parameter model used VR,HL and HR.

It was shown that the assumption of uncorrelated error terms in the EOG subtraction models is invalid. To remedy this, the error terms were modelled as an autoregressive series.

A disadvantage of the EOG subtraction method is that if the EOGs themselves contained artefacts, these would be introduced into the corrected EEG. The exact effect of this is difficult to asses, but would appear to be small [4,5]. However, because the EOG noise is mostly high frequency, the effect can be reduced by low-pass filtering. A suitable low-pass digital filter that removes the EOG noise and introduces only minimal distortion was discussed in chapter 3.

(b) On-line Removal of Ocular Artefacts

New on-line removal algorithms based on the numerically stable UD factorization and square root algorithms have been developed. Compared to the present on-line methods the algorithms are superior, requiring no subjective manual adjustments, or the co-operation of subjects which cannot always be guarranteed. The new algorithms were shown to give similar results to their off-line equivalents.

(c) Automatic Ocular Artefact Removal System

An important part of this research was the design and development of an automatic ocular artefact removal system. The system utilizes the efficient on-line algorithms described above. The specification, design and development of the system's hardware and software were discussed in detail. Preliminary results obtained with both normal and patient subjects showed that the OAR system gives satisfactory OA removal. This system, which is the first of its kind, is compatible with standard EEG machines, so that it could be bought as an accessory. It is hoped that the system will in the near future be manufactured.

8.2 Future work.

(a) Improvements to the OAR system

It may be desirable in routine clinical work that the OAR system takes as small a space as possible. This would require the system to be reduced in size. At present, the main constraint on size is the microcomputer board. It is possible to implement the software using a single chip processor, and utilizing only the necessary components;

this would reduce the processor board to a minimum. Furthermore, programmable array logic (PAL) can be used to perform most of the functions of the decoder, and the analogue boards could be reduced by putting the discrete components on an integrated circuit. By these methods, it should be possible to reduce the size of the OAR system to at least a half of its present size.

A limitation of the OAR system at the present is that it cannot remove the OA from more than a few EEG channels simultaneously, due to the slow speed of the floating point arithmetic routines which take typically 70µS to perform an arithmetic operation. A solution to this problem is to use fast hardware floating point arithmetic devices, such as the Weitek devices [6], capable of performing an arithmetic operation in 1 or 2 µS.

The use of the fast floating point arithmetic devices should allow the incorporation of the EOG filter discussed in chapter 3 into the OAR software. At the present the filter was not used because of the additional arithmetic operations that would be involved.

(b) <u>OA Removal in the Presence of Pathological Slow Waves</u>

AS was discussed in chapter 1, when ocular artefacts and pathological slow waves and/or event related potentials, such as the CNV, are simultaneously present, the EOG subtraction method may not give optimum correction. As was found in chapter 7, the slow waves could be reduced in these cases especially at the frontal EEG channels. Thus, it is necessary to distinguish between the OA and slow waves.

It is known that slow waves and OA have different scalp potential distribution [7,8]. Slow waves for example, are uncommon along the midline, so that waves that occur only laterally are probably slow waves. Thus several electrodes could be used to separate out the two types of waves. A similar technique could be applied to the CNV which occurs abundantly only at specific sites, such as the vertex electrodes. A technique could be developed for recording the topography and quantifying the propagating waves and to investigate the spatial variation of EOGs.

(c) Further Clinical Tests

Further tests on the OAR system need to be carried out. The effectiveness of the correction procedure should be tested further on various categories of patients, particularly those exhibiting slow frontal waves and un-coperative patients (eg children) to establish its usefulness. The results from such tests should provide the basis for an assessment of the clinical usefulness of the corrector.

8.3 Conclusions.

The work described in this thesis has shown that the OA in EEG signals can be satisfactorily removed on-line, and that the realization of an automatic ocular artefact removal system is technically feasible. However, the usefulness of the instrument can only be fully assessed after extensive clinical tests.

It is hoped that the instrument will significantly alleviate the problem of OA in the human EEG.

Although the OAR system was designed specifically for removing the OA from the EEG, it could be used as a general-purpose artefact (or noise) removal system in most physiological situations where both the contaminating and the contaminated signals can be separately measured [9-11]. An example is the problem of measuring the foetal Electrocardiogram (ECG) in the presence of large contaminating maternal ECG [9,10]. Another example is the case where it is necessary to remove both the OA and the ECG artefacts from the EEG [3]. In both applications the OAR system could be used to remove the artefacts, after possible minor modifications to the software and hardware.

The OAR system could also be used as a data acquisition system. In this case, it may be necessary to link the OAR system to a suitable storage medium so that the acquired data may be saved.

It is also possible to reprogram the OAR system and then to use it in real-time digital filtering applications, e.g. [12,13].

References for Chapter 8.

[1] JERVIS B.W.,NICHOLS M.J.,ALLEN E.,HUDSON N.G. and JOHNSON T.E.
The Quantitative Assesment of Electroencephalograms Corrected for Eye Movement Artefacts. First European Signal Processing Conf. Lausane, Sept.,1980.

[2] JERVIS B.W.,NICHOLS M.J.,ALLEN E. HUDSON N.G., and JOHNSON T.E.
Comparison of two Methods for Removing Eye Movement Artefact from the EEG Signals (in preparation).

[3] FORTGENS C. and DE BRUIN M.P.
Removal of Eye Movement and ECG Artifacts from the Non-cephalic Reference EEG. Electroenceph. and Clin. Neurophysiol., 1983, 56, pp90-96.

[4] QUILTER P.M., MACGILLVRAY B.B., and WADBROOK D.G.
The Removal of Eye Movement Artefact from EEG Signals Using Correlation Techniques. Random Signals Analysis, I.E.E. Conference Publication, No. 159, 1977, pp93-100.

[5] VERLEGER R., GASSER T. and MOCKS J.
Correction of EOG Artefacts in Event-Related Potentials of the EEG: Aspects of Reliability and Validity. Psychophysiology, 1982, 19, pp. 474-480.

[6] WOO B., LIN L., and OWEN R.E.
ALU, Multiplier chips zip through IEEE Floating-Point

Operations. Electronics, 19, 1983, pp

[7] GOTMAN J., GLOOR P. and RAY W.F.
Clinical Applications of Spectral Analysis and Extraction of Features from Electroencephalograms with Slow Waves in Adult Patients. Electoenceph. and Clin. Neurophysiol., 1973, 35, pp. 225-235.

[8] MURAS J.S. and BINNIE C.D.
The Recognition of Frontal Slow Activity in the Presence of Eye Movements. The proc. of the Electrophysiol. Technol. Assoc., 1970, 17, pp. 131-143.

[9] WIDROW B., GOLVER J.R., McCOOL J.M., KAUNITZ J., WIILIAMS C.S., HEARN R.H., ZEIDLER J.R., DONG E. and GOODLIN R.C.
Adaptive Noise Cancelling: Principles ans Applications. Proc. IEEE, 1975, 63, pp1692-1716.

[10] BERVELD P. and MEIJER W.J.H.
A new Technique for the Suppression of the MECG. IEEE Trans. Biomed., 1981, 28, pp.348-354.

[11] COUPLAND S.G. and KIRKHAM T.H.
Electro-retinography and Nystagmus: Subtraction of Eye Movement Artifact. Can. J. Ophthalmol., 1981, 16, pp.192-194.

[12] GOTMAN J., IVES J.R. and GLOOR P.A.
A digital Filter for Eliminating EMG artefact from Recordings of Epileptic Seizures. Electroenceph.and Clin. Neurophysiol., 1977, 43, pp475-476.

[13] BARLOW J.

EMG Artifact Minimization during Clinical EEG Recordings by Special Analog Filtering. Electroenceph. and Clin. Neurophysiol., 1984, 58, pp161-174.

Appendix A1 The Least Squares and Correlation Techniques Methods.

(a) The least Squares Method.

The form of models used in the investigation was given in equation (2.1) which may be written more compactly in matrix form as:

$$Y = X\theta + E \qquad (A1.1)$$

where Y,X,θ and E are given by:

$$Y = \begin{bmatrix} EEG(1) \\ EEG(2) \\ \vdots \\ EEG(m) \end{bmatrix} \qquad X = \begin{bmatrix} VL(1) & VR(1) & HL(1) & HR(1) \\ VL(2) & VR(2) & HL(2) & HR(2) \\ \vdots & \vdots & \vdots & \vdots \\ VL(m) & VR(m) & HL(m) & HR(m) \end{bmatrix}$$

$$\theta = [\theta_1 \quad \theta_2 \quad \theta_3 \quad \theta_4]^T, \quad E = [e(1) \quad e(2) \; \ldots \; e(m)]^T$$

The suffix T indicates transposition.

Ordinary least squares (OLS) estimates of θ are obtained by minimizing the sum of squares of the error terms, J:

$$J = \sum_{i=1}^{m} e^2(i) = E^TE = (Y-X\theta)^T(Y-X\theta) \qquad (A1.2)$$

From linear algebra [1,2], the following theorems and identities hold. Given matrices A,B, and C:

(i) $(A + B)^T = A^T + B^T$

(ii) $(AB)^T = B^TA^T$

(iii) $(ABC)^T = C^TB^TA^T$

(iv) $\dfrac{\partial(A^TBA)}{\partial A} = 2BA = 2A^TB$

where A^TBA is a quadratic in A.

Thus using relations (i), (ii), equation (A1.2) becomes:

$$(Y-X\theta)^T(Y-X\theta) = (Y^T-\theta^TX^T)(Y-X\theta) = Y^TY - \theta^TX^TY - Y^TX\theta + \theta^TX^TX\theta \qquad (A1.3)$$

Noting that $\theta^T X^T Y$ is a scalar so that it is equal to its transpose, that is:

$$Y^T X\theta = (Y^T X\theta)^T = \theta^T X^T Y$$

then (A1.3) becomes:

$$J = Y^T Y - 2\theta^T X^T Y + \theta^T X^T X\theta \qquad \text{(A1.4)}$$

Differentiating (A1.4) with respect to θ and equating to zero:

$$\left.\frac{\partial(J)}{\partial\theta}\right|_{\theta=\hat{\theta}} = -2X^T Y + 2X^T X\hat{\theta} = 0$$

where the relation (iv) above has been used to obtain the last term. On simplifying,

$$X^T Y = (X^T X)\hat{\theta} \qquad \text{(A1.5)}$$

(b) Correlation Techniques Method

Writing out the matrices in (A1.5) explicitely, gives:

$$\begin{bmatrix} \sum EEG(i)VL(i) \\ \sum EEG(i)VR(i) \\ \sum EEG(i)HL(i) \\ \sum EEG(i)HR(i) \end{bmatrix} = \begin{bmatrix} \sum VL^2(i) & \sum VL(i)VR(i) & \sum VL(i)HL(i) & \sum VL(i)HR(i) \\ \sum VL(i)VR(i) & \sum VR^2(i) & \sum VR(i)HL(i) & \sum VR(i)HR(i) \\ \sum VL(i)HL(i) & \sum VR(i)HL(i) & \sum HL^2(i) & \sum HL(i)HR(i) \\ \sum VL(i)HR(i) & \sum VR(i)HR(i) & \sum HL(i)HR(i) & \sum HR^2(i) \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \end{bmatrix}$$

(A1.6

If the elements of $(X^T Y)$ and $(X^T X)$ are defined as follows:

$M_{VL} = \sum EEG(i)VL(i)$; $P_{VL} = \sum VL^2(i)$

$MVR = \sum EEG(i)VR(i)$; $P_{VR} = \sum VR^2(i)$

$M_{HL} = \sum EEG(i)HL(i)$; $P_{HL} = \sum HL^2(i)$

$M_{HR} = \sum EEG(i)HR(i)$; $P_{HR} = \sum HR^2(i)$

$A = \sum HR(i)HL(i)$; $B = \sum VR(i)VL(i)$; $C = \sum HR(i)VL(i)$

$D = \sum HL(i)VR(i)$; $C_{CL} = \sum HL(i)VL(i)$; $C_{CR} = \sum HR(i)VR(i)$

If these elements are substituted into (A1.6) equation (A1.7) which is identical to the results of Jervis et al [3] is obtained.

$$\begin{bmatrix} M_{VL} \\ M_{VR} \\ M_{HL} \\ M_{HR} \end{bmatrix} = \begin{bmatrix} P_{VL} & B & C_{CL} & C \\ B & P_{VR} & D & C_{CR} \\ C_{CL} & D & P_{HL} & A \\ C & C_{CR} & A & P_{HR} \end{bmatrix} \begin{bmatrix} \hat{\theta}_1 \\ \hat{\theta}_2 \\ \hat{\theta}_3 \\ \hat{\theta}_4 \end{bmatrix} \qquad (A1.7)$$

Identical results to those of Quilter et al [4] can be obtained by considering only a 2-parameter model, e.q.

$$EEG(i) = \theta_1 VL(i) + \theta_2 HL(i) + e(i) \qquad (A1.8)$$

Proceeding as before leads to the following normal equation:

$$\begin{bmatrix} \sum EEG(i)VL(i) \\ \sum EEG(i)HL(i) \end{bmatrix} = \begin{bmatrix} \sum VL^2(i) & \sum VL(i)HL(i) \\ \sum VL(i)HL(i) & \sum HL^2(i) \end{bmatrix} \begin{bmatrix} \hat{\theta}_1 \\ \hat{\theta}_2 \end{bmatrix}$$

(A1.9)

If the elements of the vectors and matrix are defined as:

$C_V = \sum EEG(i)VL(i)$; $C_H = \sum EE(i)HL(i)$

$C_C = \sum VL(i)HL(i)$; $P_V = \sum VL^2(i)$; $P_H = \sum HL^2(i)$

Equation (A1.10) is obtained:

$$\begin{bmatrix} C_V \\ C_H \end{bmatrix} = \begin{bmatrix} P_V & C_C \\ C_C & P_H \end{bmatrix} \begin{bmatrix} \hat{\theta}_1 \\ \hat{\theta}_2 \end{bmatrix} \qquad (A1.10)$$

which if multiplied out is identical to equation (8) and (9) of Quilter et al [4]. Thus it is seen that the least squares approach is simply a more formal way of deriving the normaL equations than the so-called correlation technique.

The correlation techniques approach is instructive in the sense that it reveals the need to make certain assumptions about the error terms which is not apparent in the formal least squares method. This can be illustrated as follows:

If equation (A1.8) is multiplied by VL(i) and HL(i) separately, and sum each results over all M data points (using a slightly more refined method than in [3,4]), (A1.11) results:

$$\begin{bmatrix} \sum EEG(i)VL(i) \\ \sum EEG(i)HL(i) \end{bmatrix} = \begin{bmatrix} \sum VL^2(i) & \sum VL(i)HL(i) \\ \sum VL(i)HL(i) & \sum HL^2(i) \end{bmatrix} \begin{bmatrix} \hat{\theta}_1 \\ \hat{\theta}_2 \end{bmatrix} + \begin{bmatrix} \sum e(i)VL(i) \\ \sum e(i)HL(i) \end{bmatrix} \qquad (A1.11)$$

Notice the similarity between (A1.6),(A1.9) and (A1.11) Thus (A1.11) may be written more compactly as:

$$X^TY = (X^TX)\theta + X^TE \qquad (A1.12)$$

It is seen that to obtain $\hat{\theta}$, it is necessary that $X^TE = 0$. That is, that the EOG and EEG (or error terms) are uncorrelated. This is the explicit assumption that is made in the correlation techniques method and is met if the error terms e(i) are a random sequence, It has been shown that this assumption is untenable in the OA problem.

(c) Properties of the OLS Estimates.

The least squares estimator, $\hat{\theta}$, posseses a number of useful statistical properties [5,6]. These properties are based on some assumptions, namely, that the error terms, e(i) is a stationary random variable with zero mean and constant variance. Some of the properties of $\hat{\theta}$, based on the these assumptions, will be discussed here.

(i) Bias

If equation (A1.1) is substituted into (A1.5) and, dropping the subscripts for convenience, then:

$$\hat{\theta} = (X^TX)^{-1}X^T[X\theta + E] = \theta + (X^TX)^{-1}X^TE \qquad (A1.13)$$

Taking expectation of both sides of (A1.13) and using the assumption that $\mathcal{E}[E] = 0$, then

$$\mathcal{E}[\hat{\theta}] = \mathcal{E}[\theta] + \mathcal{E}[(X^TX)^{-1}X^T]\mathcal{E}[E] = \hat{\theta}$$

where $\mathcal{E}[.]$ is the expectation of the function in the bracket. Thus the least squares estimator is unbiased.

(ii) Error Covariance

The covariance matrix corresponding to the estimate error $(\hat{\theta}-\theta)$, is obtained from (A1.13):

$$\begin{aligned} C &= \mathcal{E}\{(\hat{\theta}-\theta)(\hat{\theta}-\theta)^T\} \\ &= \mathcal{E}\{[(X^TX)^{-1}X^TE][(X^TX)^{-1}X^TE]^T\} = (X^TX)^{-1}X^T\mathcal{E}\{E^TE\}X(X^TX)^{-1} \\ &= (X^TX)^{-1}X^TRX(X^TX)^{-1} \qquad \text{(A1.14)} \end{aligned}$$

where

$R = \mathcal{E}[E^TE]$ which is the expected value of the sum of squares of the error terms, or the 'mean-square-error'. If the error terms satisfy the assumptions stated earlier, then

$$R = \mathcal{E}[E^TE] = \sigma^2 I \qquad \text{(A11.15)}$$

and (A1.14) becomes:

$$C = \sigma^2(X^TX)^{-1} = \sigma^2 P \qquad \text{(A1.16)}$$

C is useful in estimating the accuracy of the parameter estimates, but require the computation of the variance, σ^2, and this may be obtained from:

$$S^2 = \frac{\sum e^2(i)}{M-n-1}$$

(iii)Consistentcy.

The error covariance C may be written in the form:

$$C = \sigma^2(X^TX)^{-1} = \frac{\sigma^2}{m}[\tfrac{1}{m}X^TX]^{-1}$$

where M is the number of samples used in computing C. Assuming that $m \rightarrow \infty$, the matrix $(\frac{1}{m}X^TX)$ tends to a a non-singular matrix so that:

$$\lim_{m\to\infty} C = \lim_{m\to\infty} \frac{\sigma^2}{m}\left(\frac{1}{m} X^T X\right)^{-1} = 0.$$

Since $C = \mathcal{E}[(\hat{\theta}-\theta)(\hat{\theta}-\theta)^T]$ is only zero when $\hat{\theta}=\theta$, it follows that $\hat{\theta}$ is a consistent estimate of θ.

References for appendix A1

[1] JOHNSTON J.

Econometric Methods. 2nd ed. McGraw Hill, 1972.

[2] HADLEY G.

Linear Algebra, Addison-Wesley, 1961.

[3] JERVIS B.W., NICHOLS M.J., ALLEN E., HUDSON N.G. and JOHNSON T.E.

Comparison of two Methods for Removing Eye Movement Artefact from the EEG Signals (submitted to Electroenceph. and Clin. Neurophysiol.)

[4] QUILTER P.M., MACGILLVRAY B.B. and WADBROOK D.G.

The Removal of Eye Movement Artefact from EEG Signals using Correlation Techniques. Random Signals Analysis, IEE Conf. Publication, 159, 1977, pp93-100.

[5] EyKOFF P.

System Identification, Parameter and State Estimation. John Wiley, 1974.

[6] HSIA T.C.

System Identification. Lexington Books, 1977.

Appendix A2 Problems of Collinearity.

In the OA problem, collinearity always exists because the interdependence of the EOGs is an inherent characteristic of the OA system. As discussed in the main text, the existence of collinearity among the EOGs can lead to some difficulties. These include large changes in the parameter estimates, $\hat{\theta}$, when a variable is added or dropped from the model, a change in the sign of $\hat{\theta}$ from that which is expected, values of $\hat{\theta}$ that are less precise [1,2], and in the extreme case inestimable parameters. A reliable way of detecting collinearity is the examination of the correlation matrices of the EOGs and the associated eigenvalues [1,2], and this was used in the main text to demonstrate the strong linear dependence between the EOGs. In this appendix, the problem of collinearity will be re-examined to show how it affects the parameter estimates.

(a) The Influence of Collinearity on the Magnitude and Sign of the Parameter Estimates.

Given two data sequences, x_2, x_3, the simple (or zero order) correlations between them is defined as [2]:

$$r_{23} = \frac{\sum_i x_2(i)x_3(i)}{\left[\sum_i x_2^2(i)\right]^{1/2}\left[\sum_i x_3^2(i)\right]^{1/2}} \qquad (A2.1)$$

Given several variables, y, x_2, x_3 ... x_n, the correlations between them arranged in matrix form is called the correlation matrix, which is symmetric [2].

Thus:

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} & \cdots & r_{1n} \\ r_{12} & r_{22} & r_{23} & \cdots & r_{2n} \\ \vdots & & & & \\ r_{1n} & r_{2n} & r_{3n} & \cdots & r_{nn} \end{bmatrix} \qquad (A2.2)$$

where r_{1j}, j=2, ...,n is the correlation between y and x_j, and $r_{jj}=1$. (The x index starts from 2, to avoid a clash in subscripts).

Now, the OA model may be written as:

$$y(i) = \theta_2 x_2(i) + \theta_3 x_3(i) \; .... \; \theta_n x_n(i) \; e(i) \qquad (A2.3)$$

where $x_j(i)$ represent the EOGs and y(i) the measured EEG.

It is shown in [2] that the parameter estimates are functions of the co-factors of the correlation matrix, R. Thus:

$$\hat{\theta}_j = - \frac{S_1 \; R_{1j}}{S_j \; R_{11}} \quad , \; j=2,3, \; ...,n \qquad (A2.4)$$

where S_1, S_j are the standard deviations of y, and x_j respectively, and given by:

$$S_1 = \left[\frac{\sum y^2(i)}{M} \right]^{\frac{1}{2}}, \quad S_j = \left[\frac{\sum x_j^2(i)}{M} \right]^{\frac{1}{2}} \qquad (A2.5)$$

R_{1j} is the co-factor of r_{1j} in the matrix R above [3].

In equation (A2.4) above, the sign of $\hat{\theta}_j$ depends on the sign of R_{1j}, since S_1 , S_j and R_{11} must be positive [2].

When there is no correlation or collinearity between the EOGs, the ratio R_{1j}/R_{11} in (A2.4) reduces to the simple correlation between the x_j and y (see later). In the presence of collinearity, the ratio R_{1j}/R_{11} contains terms that takes this into account, so that both the magnitude and sign of $\hat{\theta}_j$ depend on the collinearity.

A two parameter model will now be used to illustrate the point. The arguments presented are extendable to larger models, but these involve considerably more algebraic manipulations.

In the case of the simple two-parameter model, the OA model of (A2.3) becomes:

$$y(i)=\theta_2 x_2(i) + \theta_3 x_3(i) + e(i) \tag{A2.6}$$

The three variables (x_2, x_3, and y) lead to the 3x3 correlation matrix:

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{12} & r_{22} & r_{23} \\ r_{13} & r_{23} & r_{33} \end{bmatrix} \tag{A2.7}$$

where as before, $r_{11} = r_{22} = r_{33} = 1$,

R_{1j}, the co-factors of r_{1j} in this case are [3]:

$$R_{11} = \begin{vmatrix} r_{22} & r_{23} \\ r_{23} & r_{33} \end{vmatrix} = 1 - r_{23}^2$$

$$R_{12} = - \begin{vmatrix} r_{12} & r_{13} \\ r_{23} & r_{33} \end{vmatrix} = -(r_{12} - r_{13}r_{23}) \tag{A2.8}$$

$$R_{13} = \begin{vmatrix} r_{12} & r_{13} \\ r_{22} & r_{23} \end{vmatrix} = r_{12}r_{23} - r_{13}$$

The symbol $|\quad|$ denotes the determinant.

Thus from (A2.4) and (A2.8):

$$\hat{\theta}_2 = \frac{S_1 (r_{12} - r_{13}r_{23})}{S_2 (1 - r_{23}^2)} \qquad \text{(A2.9)}$$

$$\hat{\theta}_3 = \frac{-S_1 (r_{12}r_{23} - r_{13})}{S_3 (1 - r_{23}^2)}$$

Now, if there is no correlation between x_2 and x_3, then $r_{23} = 0$, and $\hat{\theta}_2$, $\hat{\theta}_3$ reduce to:

$$\hat{\theta}_2 = \frac{S_1 r_{12}}{S_2} , \quad \hat{\theta}_3 = \frac{s_1 r_{13}}{S_3} \qquad \text{(A2.10)}$$

and the sign and magnitude of each parameter estimate depend entirely on the correlation between the corresponding x and y (and of course the standard deviations). However, if there is collinearity equation (A2.9) holds and each parameter estimate is determined by the corresponding x and y as well as by the correlation between the corresponding x and the other x . The extent the other x affects the parameter estimates depends on the degree of correlation. Thus if the correlation is strong, the sign of a parameter estimate could be the exact opposite of the sign where there is no correlation (cf A2.9 and A2.10).

(b) <u>The Effects of Collinearity on the Precision of the Parameter Estimates.</u>

The reduction in the precision of the parameter estimates may be seen from a consideration of the simple two parameter model [2]:

$$y(i) = \theta_2 x_2(i) + \theta_3 x_3(i) + e(i) \quad i = 1, 2, \ldots, M.$$

Following the procedure given in appendix A1, this leads to the normal equations below:

$$\begin{bmatrix} \sum y(i)x_2(i) \\ \sum y(i)x_3(i) \end{bmatrix} = \begin{bmatrix} \sum x_2^2(i) & \sum x_2(i)x_3(i) \\ \sum x_2(i)x_3(i) & \sum x_3^2(i) \end{bmatrix} \cdot \begin{bmatrix} \hat{\theta}_2 \\ \hat{\theta}_3 \end{bmatrix} \qquad (A2.11)$$

or in compact form:

$$X^TY = [\ X^TX\ ]\hat{\theta} \qquad (A2.12)$$

Now the standard deviations S_1 and S_j are given in (A2.5). Thus if the first row of (A2.11) is divide by MS_2 and the second by MS_3, the equation A2.11 becomes:

$$\begin{bmatrix} r_{12}S_1 \\ r_{31}S_1 \end{bmatrix} = \begin{bmatrix} r_{22}S_2 & r_{23}S_3 \\ r_{23}S_2 & r_{33}S_3 \end{bmatrix} \begin{bmatrix} \hat{\theta}_2 \\ \hat{\theta}_3 \end{bmatrix}$$

Thus :

$$(X^TX) = \begin{bmatrix} S_2 & r_{23}S_3 \\ r_{23}S_2 & S_3 \end{bmatrix} \quad \text{and} \quad (X^TX)^{-1} = \frac{1}{S_2S_3(1-r_{23}^2)}\begin{bmatrix} S_3 & -r_{23}S_3 \\ -r_{23}S_2 & S_2 \end{bmatrix} \qquad (A2.14)$$

where use has been made of the fact that:

$r_{22} = r_{33} = 1$, and $r_{23} = r_{32}$

The variance-covariance matrix of the parameter estimates $\mathrm{Var}(\hat{\theta})$, is given by [2]:

$\mathrm{Var}(\theta) = S^2\ (X^TX)^{-1}$, where S^2 is the residual variance of the error term, e(i).

Thus using (A2.14):

$$\mathrm{Var}(\hat{\theta}_2) = \frac{S^2}{S_2(1-r_{23}^2)}, \quad \mathrm{Var}(\hat{\theta}_3) = \frac{S^2}{S_3(1-r_{23}^2\ )} \qquad (A2.15)$$

It is seen from these equations that as the correlation between x_2 and x_3, r_{23}, increases the larger the variances of the parameter estimates become. In the limit when $r_{23}=1$, the variances become infinite.

The exact effect of collinearity on the parameter estimates in a given situation can be seen from a consideration of the difference between the parameter estimates and the parameters themselves, i.e. $(\hat{\theta}-\theta)$. From appendix A1

$$(\hat{\theta}-\theta) = (X^TX)^{-1}X^TE \qquad (A2.16)$$

where E is the vector of the error terms, e(i). Thus

$$X^TE = \begin{bmatrix} x_2(1) & x_3(1) \\ \vdots & \vdots \\ x_2(m) & x_3(m) \end{bmatrix}^T \begin{bmatrix} e(1) \\ \vdots \\ e(m) \end{bmatrix} = \begin{bmatrix} \sum x_2e \\ \\ \sum x_3e \end{bmatrix} \qquad (A2.17)$$

For simplicity, assume that x_3 is related to x_2 as

$$x_3(i) = \alpha x_2(i) + v(i) \qquad (A2.18)$$

where v is a disturbance term uncorrelated with x_2. Assume further that $\sum x_2^2(i) = \sum x_3^2(i) = 1$, so that $(X^TX)^{-1}$ has the simple form [2]:

$$(X^TX)^{-1} = \frac{1}{(1-\alpha^2)} \begin{bmatrix} 1 & -\alpha \\ -\alpha & 1 \end{bmatrix} \qquad (A2.19)$$

Substituting (A2.17) and (A2.19) into (A2.16) and simplifying:

$$(\hat{\theta}-\theta) = \begin{bmatrix} \hat{\theta}_2 - \theta_2 \\ \hat{\theta}_3 - \theta_3 \end{bmatrix} = \frac{1}{1-\alpha^2} \begin{bmatrix} \sum x_2e - \alpha \sum x_3e \\ -\alpha \sum x_2e + \sum x_3e \end{bmatrix} \qquad (A2.20)$$

Using (A2.18) in (A2.20) and simplifying gives:

$$\hat{\theta}_2 - \hat{\theta}_2 = \sum x_2 e(i) \frac{-\alpha\sum e(i)v(i)}{(1-\alpha^2)} = -\frac{\sum e(i)\ v(i)}{(1-\alpha^2)} \qquad \text{(A2.21a)}$$

$$\hat{\theta}_3 - \theta_3 = \frac{1\sum e(i)v(i)}{(1-\alpha^2)} \qquad \text{(A2.21b)}$$

Noting that both equations (A2.21a) and (A2.21b) contain the term $\sum e(i)v(i)$ then if α is large and positive (i.e x_2 and x_3 are highly positively correlated), large and opposite errors will be produced in the estimates θ_2 and θ_3. E.g, if $\hat{\theta}_2$ underestimates θ_2, then $\hat{\theta}_3$ overestimates θ_3.

(c) Partial Correlation Coefficients.

The problems discussed in (a) above can also be studied by considering partial correlations, which is essentially the simple correlation between each x and y after the effects of the other x's have been removed [2,4]. Thus, if the partial correlation between a given EOG (or x) and EEG (or y) is small, it suggests that the particular EOG makes little or no contribution to the OA in the EEG [4]. On the other hand, if the partial correlation coefficient is larger than the simple correlation coefficient between the EOG and the EEG, then some other EOG(s) are masking the contribution of the first EOG to OA [4].

References for Appendix A2.

[1] CHATTERJEE S. and PRICE B.
Regression Analysis By Example. John Wiley, 1977.

[2] JOHNSTON J.
Econometric Methods. 2nd ed, McGraw Hill, 1972.

[3] BAJPAI A.C., CALUS I.M., and FAIRLEY J.A.
Mathematics for Engineers and Scientists. Vol. 1.
John Wiley, 1973.

[4] KENDALL M. and STUART A.
The Advanced Theory of Statistics, Vol 2, 4th ed.
Griffin and CO, 1979.

Appendix A3 On the Quantitative Assessment of EEGs Corrected for Ocular Artefacts.

(a) It was stated in the main text that quantitative methods of assessing the effectiveness of OA removal may give misleading results, especially in the presence of large secondary artefacts and serial correlation in the error terms. In this appendix results are given to illustrate this point. These results were obtained in a preliminary investigation into ways of obtaining more sensitive and more reliable measures of the effectiveness of OA removal than the method of Jervis et al [1] (see chapter 1).

The models used in the investigation included those used by Jervis et al [1] and some new ones. These models are defined below. A discussion of these models, with the exception of of model 4E, was given in chapter 2. Model 4E attempts to compensate for possible time delays between the EOGs and OA.

Model name	EOGs used
2A	VL,HL
3A	VL,HL,HR
4A	VL,VR,HL,HR
4B	VL,VLxVL,HL,HR
4D	VL,HLxHR,HL,HR
4E	VL,[VL(i-1) + VL(i+1)],HL,HR

One significant problem in assessing the effectiveness of OA removal is that the true EEG is not known, so that it is impossible to judge from the corrected EEG exactly how well the OA removal has been. Thus following Jervis et al [1], the models given above were used to correct several EEG records for OA, and in each case the performance of the models were compared using a number of criteria.

(b)Summary of Criteria used in the Investigation.

Several criteria were used in the investigation, but only the most promising ones are discussed here.

(i) Test of Additi onal term.

Starting with the model 2A and for each EEG record the F-test [2] was used to assess how much improvements or otherwise the addition of an extra term to the OA model made to the fit. The form of the F-test used was [2]:

$$F_{1,n} = \frac{(M-k)S^2_{k-1} - (M-k-1)S^2_k}{S^2_k} \qquad (A3.1)$$

where S^2_k and S^2_{k-1} are ,the residual variances of the models with k terms (or parameters) and (k-1) parameters, respectly, M is the number of data points used to obtain the residual variances.

A significant value of $F_{1,n}$ indicates that the model with k terms is better than that with (k-1) terms. That is, the addition of the extra term is justified. The opposite is also true. I.e. an insignificant value of $F_{1,n}$ indicates that the addition of the extra term is not justified.

In the investigation, the value of $F_{1,n}$ obtained from (A3.1) is compared to critical values of F obtained from tables [2]. If $F_{1,n}$ for a given model is greater than the critical values then the $F_{1,n}$ is said to be significant.

For large n, the critical values at 0.1%, 1%, and 5% levels are, respectively, 10.83, 6.63 and 3.84 [2].

Remarks

(1) The F-test assumes that the error terms have certain distributional properties. For example, that they are normally distributed and are not serially correlated

(2) The test can be used to compare the performance of a large model and its subsets (see chapter 2), but cannot be directly used to compare models where one is not a subset of the other, for example in comparing models with the same number of parameters [3]. This was one of the reasons the Mallow's statistic was used to compare models in the main text instead of the F-test.

(ii) Pictorial Method - OA estimates and their differences

This criteria allows the visual assessment of the improvements or otherwise of adding an extra term to the OA model (see chapter 2). Thus it may be described as the pictorial form of the F-test. There are two aspects of the test. Firstly, the estimates of the OAs for the various models are compared to identify the 'good' models. The OA estimates for the good models should contain most (if not all) the information about the OA in the EEG and little (if any) of the noise or other contamination. The more the noise in the OA estimates the more genuine EEG that will be removed in the OA process . Secondly, the differeces between the OA estimates (or equivalently, the differences between the corrected EEGs) are studied. As in the F-test, starting with the model 2A, the improvements or otherwise of adding an extra term to the model is studied.

(c) Results and Discussions.

(i) Test of additional term - the F-test

Several EEG records were corrected for OA by the OLS method using the models defined in section (a). In each case and for each model the F-value (eqn A3.1 was computed (where appropriate). Table A3.1 gives the results obtained from 21 EEG records for three subjects. The results are grouped separately for each subject and according to the type of ocular movement to reduce any effects of individual differences or differences in the characteristics of the various ocular movements on the conclusions.

A comparison of the F values in the table with the 5% level (or critical value) showed the following:

model name	number of times model's F value was significant.
3A	17
4A	17
4B	12
4D	12
4E	5

This result suggested that in general, model 3A was better than model 2A, model 4A was the best, models 4B and 4D may be preferred to model 3A, but that model 4E does not offer any improvement over model 3A. These findings parallel those of Jervis et al [1], although their method was not sensitive enough to distinguish clearly between the models. For example, they applied their method to model 4A and in 96% of the cases no measurable value could be obtained for their test statistic (see chapter 1).

Table A3.1 F values for models (5 % significant level= 3.84

Note: Although the significant levels of the F-test are not strictly valid [2] due to serial correlation in the error terms, the F values in many cases were either too large or too small that the validity question may be overlooked [4]).

		F Values				
Type of OM and Subject	Data File	3A	4A	4B	4D	4E
AJS	42	0.02	63.45	5.97	2.64	3.49
VEM	54	22.53	109.81	1.34	10.06	7.99
	48	96.43	171.18	2.27	0.29	11.12
AJS	36	25.34	46.75	17.96	8.34	2.06
HEM	60	8.34	34.06	14.88	12.19	0.07
	84	69.64	31.84	20.76	1.61	0.59
AJS	102	12.96	35.89	0.05	14.26	0.02
Blinks →	108	47.25	21.35	51.82	5.12	0.03
	114	2.52	51.82	3.25	6.27	0.37
JSS →	36	251.45	0.31	21.00	12.65	14.83
VEM	66	46.18	0.51	9.50	64.23	0.96
	90	12.42	136.33	0.06	18.93	2.89
JSS	24	22.04	136.33	0.06	18.93	0.99
HEM →	48	0.08	170.76	15.17	19.26	0.79
	96	9.29	7.59	0.94	19.73	0.39
JSS	126	5.76	35.17	0.15	22.73	2.33
Blinks	132	92.46	4.80	33.48	1.07	5.37
	138	3.94	45.46	2.18	28.10	1.08
DEA	24	79.31	27.76	52.20	1.79	19.00
VEM	54	29.09	0.92	0.48	0.37	7.06
	72	0.09	0.77	72.41	0.02	8.74

(ii) Pictorial Method - OA estimates and their differences.

The pictorial method was applied to several EEG records. It was found that for HEM, model 3A gave significant improvements over model 2A, but not for blinks and VEM where the improvements were small or none at all. For blinks and VEM model 4A sometimes gave significant improvements over model 3A although in each case, it also introduced a high level of noise into the corrected EEGs (ie its OA estimates contained high levels of noise). For HEM, model 4A did not give any improvements. Models 4B and 4D always gave some improvements over model 3A, but these were small in many cases. Model 4E in general did not give any improvements over model 3A. These results suggested that for HEM, it is necessary to include both horizontal EOGs in the model, but that it is not necessary to include both vertical EOGs for VEM or blinks (cf the variables in models 4A and 3A).

Figure A3.1 gives the differences between the OA estimates using the OLS method for a horizontal eye movement record. Examination of the figure showed that figure A3.1(ii) and (v) contained large amount of noise and suggested that the corresponding models (models 4A and 4E) did not give a significant improvements over model 3A. In this case these models would in fact significantly affect the true EEG. As examination of table A3.1 will show, the result for the pictorial method agreed with the F-test for models 4B, 4E, and 4D, but not for models 3A and 4A. Examination of the OA differences and table A3.1 showed that for model 3A, both the OA differences and the F value were small. For Model

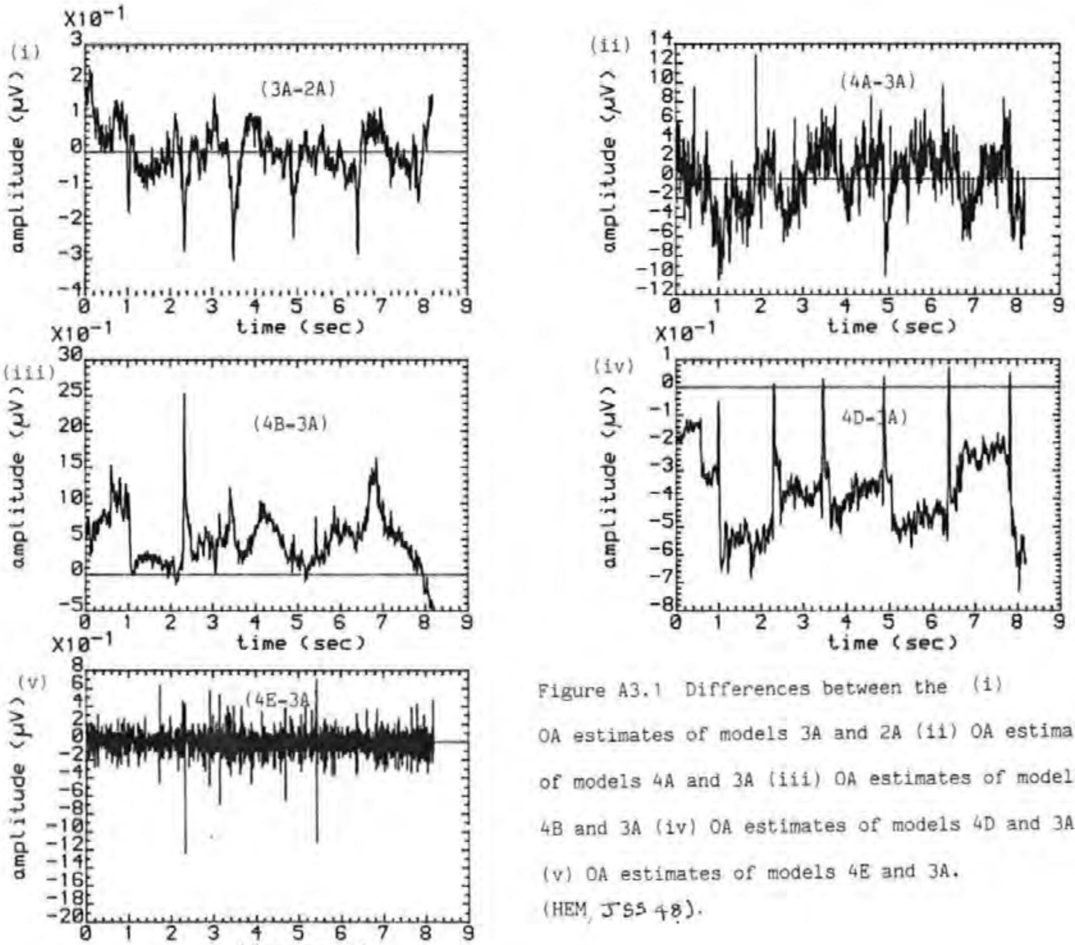


Figure A3.1 Differences between the (i) OA estimates of models 3A and 2A (ii) OA estimates of models 4A and 3A (iii) OA estimates of models 4B and 3A (iv) OA estimates of models 4D and 3A (v) OA estimates of models 4E and 3A.
(HEM, JSS 48).

4A, both the OA differences and the F value were very large although the OA differences were mostly noise. It was found that in general when the OA differences associated with a model were large the corresponding F value was also large, and hence highly significant irrespective of whether the large differences were mostly noise or OA. Also when the differences were small, the corresponding F value was insignificant. Thus a model whose OA estimate contained the most noise which led to a substantial reduction in the true EEG (an undesirable side effect) would give a highly significant result which could be misleading.

Figure A3.2 gives results for a VEM record. It is seen that the differences between the OAs for models 3A and 2A, and the OAs for models 4E and 3A, are mostly noise, so that the use of models 3A or 4E in this case is not desirable. However, the F values for models 3A and 4E are significant and suggest that model 3A is to be preferred to model 2A and model 4E is to be preferred to model 3A. Both methods agree that model 4D is significantly better than model 3A. As mentioned earlier, the quantitative test only takes into account how large the differences between the OAs are, but not the composition of these differences.

Figure A3.3 shows the OA differences for the models for a blink record. It is seen that the OA differences in figure A3.3(i) and (v) contain substantial noise, so that the corresponding models (3A and 4E, respectively) have not performed well. Figure A3.3(ii) which corresponds to model 4A also contains a fair amount of noise, but also contains significant level of OAs and so its performance on this occassion may be acceptable. The F values for most of

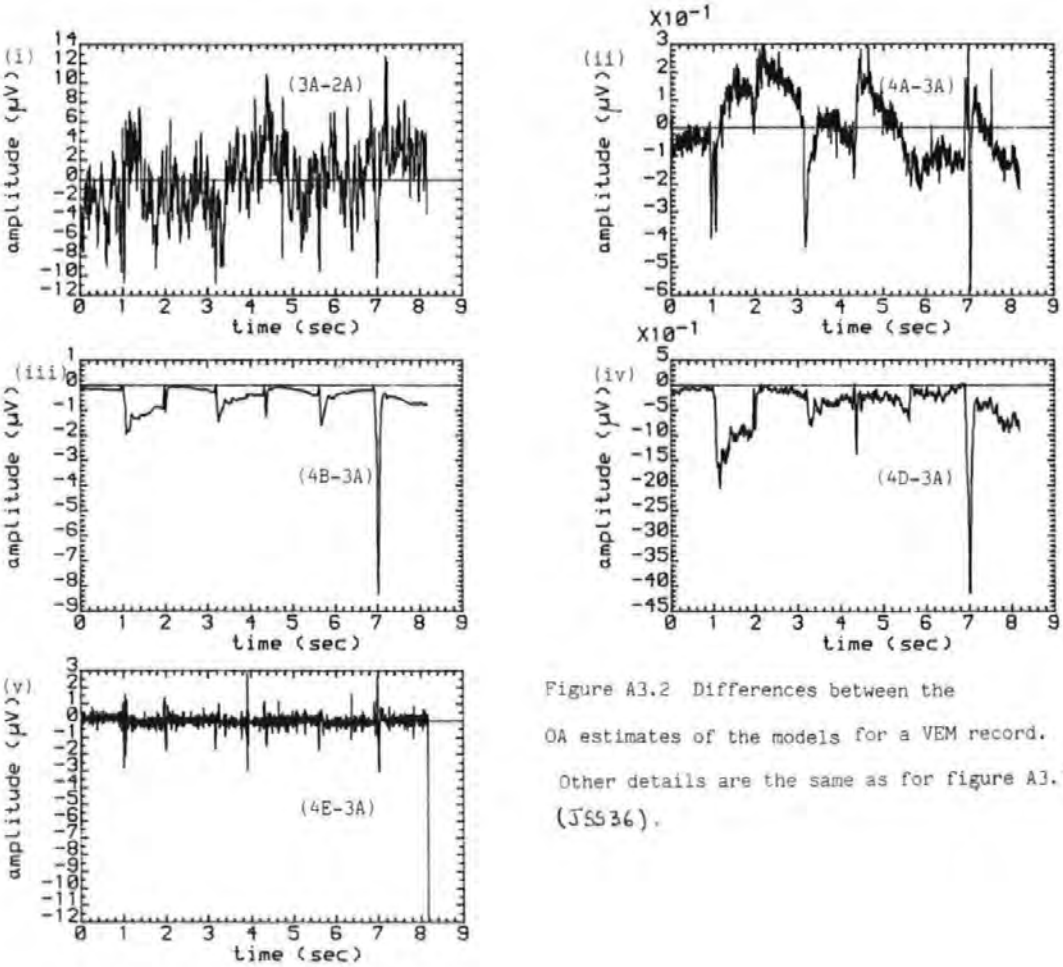


Figure A3.2 Differences between the OA estimates of the models for a VEM record. Other details are the same as for figure A3.1 (JSS36).

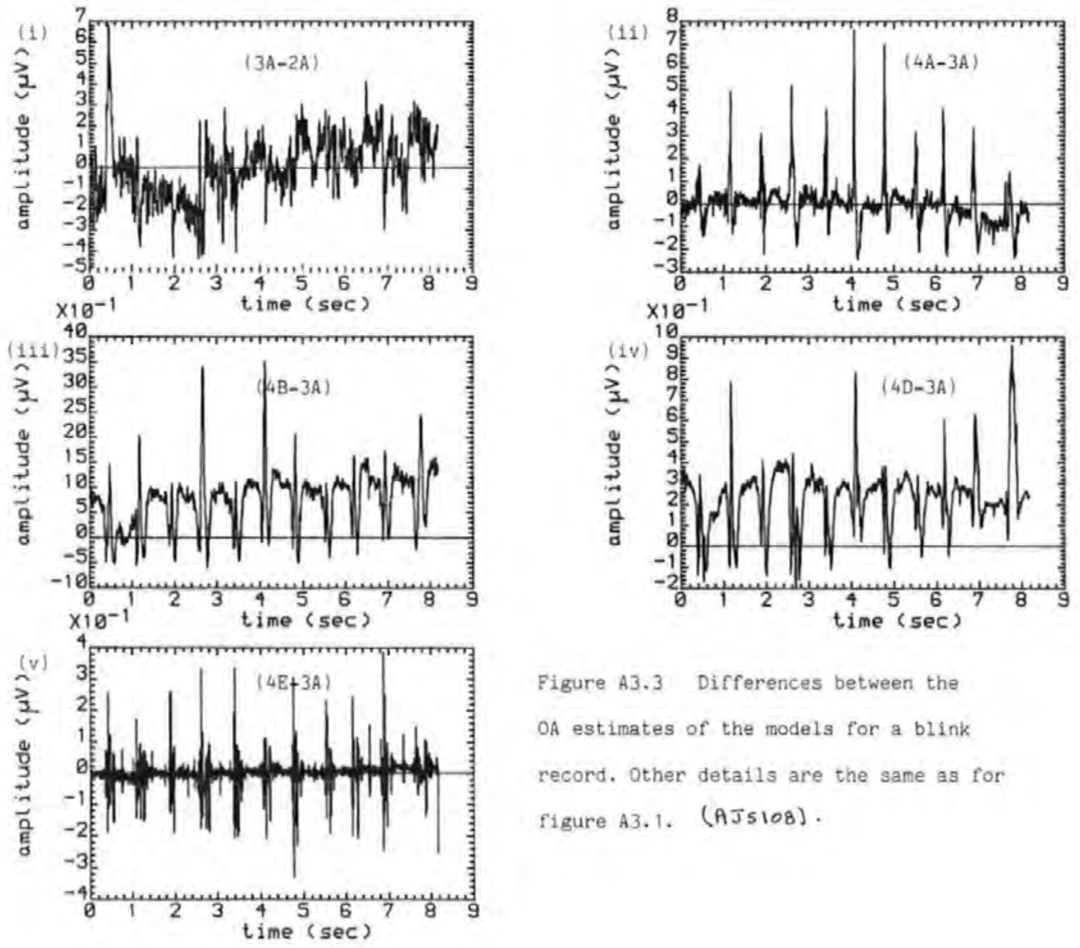


Figure A3.3 Differences between the OA estimates of the models for a blink record. Other details are the same as for figure A3.1. (AJs108).

the models (3A, 4A, 4B, 4D) are significant, and as said earlier have not taken into acount the exact content of the OA differences.

(d) Summary.

The quantitative tests suggested that in general model 4A was the best, irrespective of the type of ocular movement. However, the waveform subtraction or pictorial method showed that no single model was the best for all types of OM. It showed that in nearly all the cases considered model 4A had an undesirable side effects - that of introducing EOG artefact or noise into the true EEG. The quantitative test did not take account of factors such as noise in the EOGs, whereas the pictorial method revealed the noise. It was found that the F values which were exceptionally high corresponded to cases where the OA estimates and the associated OA differences contained significant levels of noise.

References for Appendix A3.

[1] JERVIS B.W., NICHOLS M.J., ALLEN E., HUDSON N.G. and JOHNSON T.E.
Comparison of two methods for Removing Eye Movement Artefacts from the EEG Signals (submitted to Electroenceph. and Clin. Neurophysiol.)

[2] JOHNSTON J.
Econometric Methods. 2nd ed, McGraw Hill, 1972.

[3] WEISBERG S.
Applied Linear Regression. John Wiley, 1980.

[4] WRIGHT D.E.
Personal Communication.

Appendix A4 Recursive Least Squares Algorithm.

From section 4.1, the matrix equation for the OA model may be written as:

$$Y_{m+1} = X_{m+1}\theta_{m+1} + E_{m+1} \qquad \text{(A4.1)}$$

where these matices were defined in section 4.1. In the on-line case, the sum of squares of the error term, J, is minimized as follows [1-4].

$$J = \sum_{i=1}^{m} \gamma^{m-i} e^2(i) \qquad 0 < \gamma < 1 \qquad \text{(A4.2)}$$

This means that Y_{m+1}, X_{m+1} and E_{m+1}, in partitioned form, become :

$$Y_{m+1} = [\,\gamma^{m/2}y(1) \quad \gamma^{(m-1)/2}y(2) \;\ldots\; \gamma y(m-1) \quad \gamma^{1/2}y(m) \;\vdots\; y(m+1)\,]^T$$

$$= [\,\gamma^{\frac{1}{2}}\, Y_m \;\vdots\; y(m+1)\,]^T$$

$$X_{m+1} = \left[\begin{array}{c} \gamma^{m/2} \quad x^T(1) \\ \gamma^{m-1/2} \quad x^T(2) \\ \vdots \\ \gamma^{\frac{1}{2}} \quad x^T(m) \\ \hline x^T(m+1) \end{array}\right] = \left[\begin{array}{c} \gamma^{\frac{1}{2}}\; X_m \\ \hline x^T(m+1) \end{array}\right]$$

$$E_{m+1} = [\gamma^{m/2}\; e(1) \quad \gamma^{(m-1)/2}\; e(2) \quad \ldots \quad \gamma e(m-1) \;\; \gamma^{\frac{1}{2}}e(m) \;\vdots\; e(m+1)]^T$$

It is to be noted that if γ is set to unity, these expressions reduce to those of equations 2.2.

Proceeding as in appendix A1, then:

$$\hat{\theta}_{m+1} = [\; X^T_{m+1}X_{m+1} \;]^{-1}X^T_{m+1}Y_{m+1}$$

If the matrix P(m) is defined as $P(m) = [X^T_m X_m]^{-1}$, then:

$$P(m+1) = [\; X^T_{m+1}X_{m+1}]^{-1}$$

Using X_{m+1}, above, P(m+1) becomes:

$$P(m+1) = \left\{ \left[\gamma^{\frac{1}{2}} X_m^T \;\middle|\; x(m+1) \right] \begin{bmatrix} \gamma^{\frac{1}{2}} X_m \\ \hdashline x^T(m+1) \end{bmatrix} \right\}^{-1}$$

$$= [\gamma X_m^T X_m + x(m+1)x^T(m+1)]^{-1}$$

$$= [\gamma P^{-1}(m) + x(m+1)x^T(m+1)]^{-1} \qquad (A4.3)$$

Invoking the matrix inversion lemma of equation A4.4 [1]

$$(A + BCD)^{-1} = A^{-1} - A^{-1}B(C^{-1} + DA^{-1}B)^{-1}DA^{-1} \qquad (A4.4)$$

where the matrice A is non singular, then (A4.3) becomes (in this case $C \cong I$):

$$P(m+1) = \frac{1}{\gamma}P(m) - \frac{1}{\gamma} P(m)x(m+1)[\gamma + x^T(m+1)P(m)x(m+1)] \;.\; x^T(m+1)P(m)$$

$$= \frac{1}{\gamma}\left[P(m) - \frac{P(m)x(m+1)x^T(m+1)P(m)}{\alpha}\right] \qquad (A4.5)$$

where $\alpha = \gamma + x^T(m+1)P(m)x(m+1)$

Equation (A4.5) can be written in various forms. For example, it may be written as:

$$P(m+1) = \frac{1}{\gamma}\left[I - \frac{1}{\alpha} P(m)x(m+1)x^T(m+1)\right]P(m)$$

$$= \frac{1}{\gamma}[I - G(m)x^T(m+1)]P(m) \qquad (A4.6)$$

where G(m) is the Kalman gain vector, given by:

$$G(m) = P(m)x(m+1)/[\gamma + x^T(m+1)P(m)x(m+1)] \qquad (A4.7)$$

In this form P(m+1) can be related to the well known Kalman filter [2].

Now,

$$\hat{\theta}(m+1) = [X^T_{m+1}X_{m+1}]^{-1}X^T_{m+1}Y_{m+1}$$

Using the definition for P(m), $\hat{\theta}(m+1)$ may be written as

$$\hat{\theta}(m+1) = P(m+1)X^T_{m+1}Y_{m+1} \quad , \text{ and}$$

$$X_{m+1}^{T}Y_{m+1} = [\,\gamma^{\frac{1}{2}}X_{m}^{T} \mid x(m+1)\,] \begin{bmatrix} \gamma^{\frac{1}{2}} Y_m \\ \hline y(m+1) \end{bmatrix}$$

$$= \gamma X_m^T Y_m + x(m+1)y(m+1)$$

Therefore,

$$\hat{\theta}(m+1) = P(m+1)[\gamma P^{-1}(m)\hat{\theta}(m) + x(m+1)y(m+1)\,].$$

From (A4.3), $P^{-1}(m+1) = \gamma P^{-1}(m) + x(m+1)x^T(m+1)$, so that $\gamma P^{-1}(m)\hat{\theta}(m) = [P^{-1}(m+1) - x(m+1)x^T(m+1)\,]\hat{\theta}(m)$.

Thus,

$$\hat{\theta}(m+1) = P(m+1)[P^{-1}(m+1) - x(m+1)x^T(m+1)]\hat{\theta}(m) + P(m+1)x(m+1)y(m+1)$$

$$= \hat{\theta}(m) - P(m+1)x(m+1)x^T(m+1)\hat{\theta}(m) + P(m+1)x(m+1)y(m+1)$$

$$= \hat{\theta}(m) + P(m+1)x(m+1)[y(m+1) - x^T(m+1)\hat{\theta}(m)] \qquad (A4.8)$$

or in words,

New estimate = old estimate + Gain x Prediction error.

Equation (A4.5) or one of its variants and equation (A4.8) give the real-time recursive least squares algorithm.

(a) <u>Algorithm for updating P(m+1) and $\hat{\theta}$(m+1)</u>

To obtain P(m+1) and $\hat{\theta}$(m+1) on the computer, equation (A4.5) can be used directly. However, the coding can be made more efficient by exploiting the symmetry of P(m+1) and some matrix identity [5].

Thus writing P(m+1) and $\hat{\theta}$(m+1) as:

$$P(m+1) = \frac{1}{\gamma}[P(m) - G(m)x^T(m+1)P(m)]$$

$$\hat{\theta}(m+1) = \hat{\theta}(m) + G(m)[y(m+1) - x^T(m+1)\hat{\theta}(m)]$$

where,

$$G(m) = P(m)x(m+1)/[\gamma + x^T(m+1)P(m)x(m+1)].$$

The algorithm becomes:

(i) $V(m+1) = P(m)x(m+1)$

(ii) $\delta(m+1) = x^T(m+1)V(m+1) + \gamma$

(iii) $G(m\ ) = V(m+1)/\delta(m+1)$

(iv) $\Delta e = y(m+1) - x^T(m+1)\hat{\theta}(m)$

(v) $\hat{\theta}(m+1) = \hat{\theta}(m) + G(m)\Delta e$

(vi) $P(m+1) = \frac{1}{\gamma}[P(m) - G(m)V(m+1)]$

Notes.

1. Because of symmetry, $P(m)x(m+1) = x^T(m+1)P(m)$

2. In a computer program the symmetry of $P(m+1)$ would be used to compute the lower triangular half of $P(m+1)$.

(b) Matrix Inversion Lemma.

Assuming that the inverse of a matrix, A, exists then it can be modified by adding a term of the form BCD to it. This modification normally reduces the number of operations required to obtain the inverse of A.

Lemma: If A is a non-singular square matix then the following identity holds [1,6]:

$$(A + BCD)^{-1} = A^{-1} - A^{-1}B(C^{-1} + DA^{-1}B)^{-1}DA^{-1} \qquad (A4.9)$$

Proof.

Premultiply both sides of (A4.9) by (A + BCD) to obtain:

$$I = (A + BCD)[A^{-1} - A^{-1}B\left\{C^{-1} + DA^{-1}B\right\}^{-1}DA^{-1}] \qquad (A4.10)$$

The aim in this proof is to show that the right hand side of (A2.10) can be reduced to an identity matrix, I.

$(A + BCD)[A^{-1} - A^{-1}B(C^{-1} + DA^{-1}B)^{-1}DA^{-1}]$

$= I + BCDA^{-1} - B(C^{-1} + DA^{-1}B)^{-1}DA^{-1}$

$- BCDA^{-1}B(C^{-1} + DA^{-1}B)^{-1}DA^{-1}$

$= I + BCDA^{-1} - B[I + CDA^{-1}B][C^{-1} + DA^{-1}B]^{-1}DA^{-1}$

$$= I + BCDA^{-1} - BC[C^{-1} + DA^{-1}B][C + DA^{-1}B]^{-1}DA^{-1}$$

$$= I + BCDA^{-1} - BCDA^{-1} = I.$$

References for Appendix A4.

[1] HSIA T.C.

System Identification. Lexington books, 1977.

[2] PETERKA C.

A Square Root filter for Real-time Multivariate Regression Kybernetika, 1975, 11, pp53-67.

[3] MORRIS E.L. and ABAZA B.A.

Adaptive control of Steam Turbine. Proc. IEE, 1976, 123 pp. 549-553.

[4] YOUNG P.

Recursive Approaches to time series Analysis. Bull. I.M.A., 1974, 10, pp. 209-224.

[5] BIERMAN G.J.

Factorization Methods for Discrete Sequential Estimation Academic Press, 1977.

[6] EYKOFF P.

System Identification, Parameter and State Estimation. John Wiley, 1974.

Appendix A5 The Square Root Algorithm.

In this appendix, the orthogonal matrix (see later) will be used to factor the covariance matrix into the product of two matrices. The algorithm for updating S, the square root of P will then be given. the derivation of the algorithm presented here is due to Peterka [1,2]

The equation for P can be written as:

$$P(m+1) = \frac{1}{\Upsilon}\left[I - \frac{1}{\alpha}P(m)x(m+1)x^T(m+1)\right]P(m) \qquad (A5.1)$$

Define the positive definite matrix, P(m), as:

$$P(m) = S(m)S^T(m) \qquad (A5.2)$$

where S(m) is an upper triangular matrix. Using (A5.2), equation (A5.1) becomes:

$$\begin{aligned} P(m+1) &= \frac{1}{\Upsilon}\left[I - \frac{1}{\alpha}S(m)S^T(m)x(m+1)x^T(m+1)\right]S(m)S^T(m) \\ &= \frac{1}{\sqrt{\Upsilon}}S(m)\left[I - \frac{1}{\alpha}S^T(m)x(m+1)x^T(m+1)S(m)\right]S^T(m)\frac{1}{\sqrt{\Upsilon}} \end{aligned} \qquad (A5.3)$$

define $f(m+1) = S^T(m)x(m+1)$, then $f^T(m+1) = x^T(m+1)S(m)$ and (A5.3) becomes:

$$P(m+1) = S(m+1)S^T(m+1) = \frac{1}{\sqrt{\Upsilon}}S(m)\left[I - \frac{1}{\alpha}f(m+1)f^T(m+1)\right]S^T(m)/\sqrt{\Upsilon} \qquad (A5.4)$$

where

$$\begin{aligned} \alpha &= \Upsilon + x^T(m+1)P(m)x(m+1) \\ &= \Upsilon + f^T(m+1)f(m+1) \end{aligned} \qquad (A5.5)$$

The crucial step in the square root algorithm is to rearrange (A5.4) to get (A5.6):

$$S(m+1)S^T(m+1) = \frac{1}{\sqrt{\Upsilon}}S(m)\left[I \;\vdots\; \frac{jf(m+1)}{\sqrt{\alpha}}\right]QQ^T\begin{bmatrix} I \\ \hline \frac{jf^T(m+1)}{\sqrt{\alpha}} \end{bmatrix}\frac{S(m)}{\sqrt{\Upsilon}} \qquad (A5.6)$$

where $j = \sqrt{-1}$ and Q is an orthogonal matrix, such that:

$$QQ^T = I \qquad (A5.7)$$

Noting that the right hand side of (A5.6) is symmetrical then S(m+1) may be written as:

$$S(m+1) = \frac{1}{\sqrt{\gamma}} S(m)\left[I \;\vdots\; j\,\frac{f(m+1)}{\sqrt{\alpha}}\right]Q \qquad (A5.8)$$

Next choose the orthogonal matrix Q such that:

$$\left[I \;\vdots\; j\,\frac{f(m+1)}{\sqrt{\alpha}}\right]Q = [H(m) \;\vdots\; 0] \qquad (A5.9)$$

If a Q can be found to perform the reduction of (A5.9) then (A5.8) becomes:

$$S(m+1) = \frac{1}{\sqrt{\gamma}} S(m)[H(m) \;\vdots\; 0] = \frac{1}{\sqrt{\gamma}} S(m)H(m) \qquad (A5.10)$$

It is evident from (A5.9) that Q cannot be real.

It is to be noted at this point that if the number of parameters to be estimated is n, then the dimensions of f,x and Q are n, n, and (n+1)x(n+1),respectively

Express the orthogonal matrix Q as a product:

$$Q = \prod_{i=1}^{n} Q_i \qquad (A5.11)$$

where Q_i is the generalized elementary orthogonal matrix given by:

$$\begin{array}{cc} \left[\begin{array}{ccccc} 1 & & & & \\ & 1 & 0 & & \\ 0 & & & & \\ & & C_i & & jS_i \\ & & -jS_i & 1\;\ddots & C_i \\ & & & 1 & \end{array}\right] & \begin{array}{c} \\ \\ \\ \dots(i) \\ \dots(n+1) \\ \\ \end{array} \\ \begin{array}{ccccc} \quad & \quad & (i) & \quad & (n+1) \end{array} & \end{array} \qquad (A\ \ 12)$$

This matrix is a special form of Givens matrix [3], with imaginary coefficients for S_i. The condition of orthogonality requires that:

$$C_i^2 - S_i^2 = 1 \qquad (A5.13)$$

In the factored form each elementary orthogonal matrix is used to annul one element of the vector $jf/\sqrt{\alpha}$ and at the same time obtain a column of H. The final H is obtained by successively post multiplying $[I \vdots jf/\alpha\ ]$ by Q_i. As the vector $jf/\sqrt{\alpha}$ is of dimension n, a total of n elementary orthogonal matrices is required to accomplish the decomposition of (A5.9).

The procedure will now be applied to a 3-parameter system to illustrate the point, and later generalized. In this case there are a maximum of three orthogonal matrices, $Q=Q_3Q_2Q_1$, and given by:

$$Q_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & C_3 & jS_3 \\ 0 & 0 & -jS_3 & C_3 \end{bmatrix} \qquad Q_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C_2 & 0 & jS_2 \\ 0 & 0 & 1 & 0 \\ 0 & -jS_2 & 0 & C_2 \end{bmatrix}$$

$$Q_1 = \begin{bmatrix} C_1 & 0 & 0 & jS_1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -jS_1 & 0 & 0 & C_1 \end{bmatrix}$$

It is desired to accomplish the decomposition:

$$[I \vdots j/\sqrt{\alpha}\ ]Q_3Q_2Q_1 = [H \vdots 0] \qquad \text{(A5.16)}$$

<u>Step 1:</u> $[I \vdots jf/\sqrt{\alpha}\ ]Q_3$

$$= \left[\begin{array}{ccc|c} 1 & 0 & 0 & jf_1/\sqrt{\alpha} \\ 0 & 1 & 0 & jf_2/\sqrt{\alpha} \\ 0 & 0 & 1 & jf_3/\sqrt{\alpha} \end{array}\right] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & C_3 & jS_3 \\ 0 & 0 & -jS_3 & C_3 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & f_1 S_3/\sqrt{\alpha} & j f_1 C_3/\sqrt{\alpha} \\ 0 & 1 & f_2 S_3/\sqrt{\alpha} & j f_2 C_3/\sqrt{\alpha} \\ 0 & 0 & (C_3 + f_3 S_3/\sqrt{\alpha}) & j(S_3 + f_3 C_3/\sqrt{\alpha}) \end{bmatrix}$$

From which it is seen that only the last two columns of $[I \quad jf/\sqrt{\alpha}]$ are affected by the multiplication by Q_3 ; that is, the n and (n+1)th columns. The aim in this step is to annul the last element of the 4th or (n+1)th column This is achieved by setting:

$$S_3 + f_3 C_3/\sqrt{\alpha} \qquad = 0 \qquad\qquad (A5.17)$$

Also from (A5.13)

$$C_3^2 - S_3^2 = 1 \qquad\qquad (A5.18)$$

Define:

$$\alpha_i = \gamma + \sum_{K=1}^{n} f_k^2 = \zeta_i^2 \qquad\qquad (A5.19)$$

From (A5.16) and (A5.17) and using (A5.18):

$$C_3 = \zeta_3/[\zeta_3^2 - f_3^2]^{\frac{1}{2}} = \zeta_3/\zeta_2 \qquad\qquad (A5.20)$$

$$S_3 = -f_3 C_3/\zeta_3 = - f_3/\zeta_2 \qquad\qquad (A5.21)$$

Equations (A5.20) and (A5.21) give the values for C_3 and S_3 to ensure that the last element of column 4 is annuled. Thus, after the first step $[I \quad jf/\sqrt{\alpha}]Q_3$ becomes:

$$\left[\begin{array}{ccc|c} 1 & 0 & H_{13} & jf_1/\zeta_2 \\ 0 & 1 & H_{23} & jf_2/\zeta_2 \\ 0 & 0 & H_{33} & 0 \end{array}\right] \qquad\qquad (A5.22)$$

where,

$$H_{k3} = \frac{f_k S_3}{\zeta_3} = \frac{-f_k f_3}{\zeta_3 \zeta_2} \qquad k=1,2$$

$$H_{33} = C_3 + \frac{f_3 S_3}{\zeta_3} = \frac{\zeta_3}{\zeta_2} - \frac{f_3^2}{\zeta_2 \zeta_3}$$

$$= (\zeta_3^2 - f_2^2)/\zeta_2 \zeta_3 = \zeta_2/\zeta_3$$

Step 2:

$$\left[\begin{array}{ccc|c} 1 & 0 & H_{13} & jf_1/\zeta_2 \\ 0 & 1 & H_{23} & jf_2/\zeta_2 \\ 0 & 0 & H_{33} & 0 \end{array}\right] Q_2 \qquad \text{(A5.23)}$$

$$= \left[\begin{array}{ccc|c} 1 & 1 & H_{13} & jf_1/\zeta_2 \\ 0 & 1 & H_{23} & jf_2/\zeta_2 \\ 0 & 0 & H_{33} & 0 \end{array}\right] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C_2 & 0 & jS_2 \\ 0 & 0 & 1 & 0 \\ 0 & -jS_2 & 0 & C_2 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & f_1/\zeta_2 & H_{13} & jf_1C_2/\zeta_2 \\ 0 & (C_2 + f_2S_2/\zeta_2) & H_{23} & j(S_2 + f_2C_2/\zeta_2) \\ 0 & 0 & H_{33} & 0 \end{bmatrix}$$

It is to be noted that only columns 2 and 4, or the (n-1)th and (n+1) columns, respectively, were affected. The aim in this step is to annul the last but one element in the 4th column and to obtain the 2nd column of H. To do this set:

$$S_2 + f_2C_2/\zeta_2 = 0 \qquad \text{(A5.24)}$$

It is to be noted that (A5.24) is similar to (A5.17). Thus similar results are obtained for S_2 and C_2, and (A5.22) reduces to:

$$\begin{bmatrix} 1 & H_{12} & H_{13} & jf_1/\zeta_1 \\ 0 & H_{22} & H_{23} & 0 \\ 0 & 0 & H_{33} & 0 \end{bmatrix} \qquad (A5.25)$$

<u>Step</u> 3

$$\begin{bmatrix} 1 & H_{12} & H_{13} & jf_1/\zeta_1 \\ 0 & H_{22} & H_{23} & 0 \\ 0 & 0 & H_{33} & 0 \end{bmatrix} Q_1$$

$$= \left[\begin{array}{ccc|c} (C_1 + f_1S_1/\zeta_1) & H_{12} & H_{13} & j(S_1 + f_1C_1/\zeta_1) \\ 0 & H_{22} & H_{23} & 0 \\ 0 & 0 & H_{33} & 0 \end{array}\right]$$

To annul the 1st element of the 4th column set:

$$S_1 + f_1C_1/\zeta_1 = 0 \qquad (A5.26)$$

This leads to :

$$\left[\begin{array}{ccc|c} H_{11} & H_{12} & H_{13} & 0 \\ 0 & H_{22} & H_{23} & 0 \\ 0 & 0 & H_{33} & 0 \end{array}\right] = \left[H \,\middle|\, 0 \right] \qquad (A5.27)$$

where,

$$H_{11} = \frac{\zeta_0}{\zeta_1} \; ; \; H_{12} = \frac{-f_1f_2}{\zeta_2\zeta_1} \; ; \; H_{22} = \frac{\zeta_1}{\zeta_2}$$

$$H_{13} = \frac{-f_1f_3}{\zeta_3\zeta_2} \; ; \; H_{23} = \frac{-f_2f_3}{\zeta_3\zeta_2} \; ; \; H_{33} = \frac{\zeta_2}{\zeta_3}$$

Or in general,

$$H_{ki} = -f_kf_i/\zeta_i\,\zeta_{i-1} \; , \quad k < i \qquad (A5.28a)$$

$$H_{ii} = \zeta_{i-1}/\zeta_i \qquad (A3.28b)$$

The general formula for computing the elements of H can be used to update S(m+1) in (A5.10) directly. Thus,

$$S(m+1)_{ij} = \frac{1}{\sqrt{\Upsilon}} \sum_{K=1}^{j} [S(m)]_{ik} H_{kj} \qquad (A5.29)$$

$$= \frac{1}{\sqrt{\Upsilon}} \frac{\zeta_{j-1}}{\zeta_j} \left\{ [S(m)]_{ij} - \frac{f_j}{\zeta^2_{j-1}} \sum_{K=1}^{j-1} [S(m)]_{ik} f_k \right\} \qquad (A5.30)$$

If a vector g_i is defined as:

$$g_i^{(1)} = \sum_{K=1}^{\ell} [S(m)]_{ik} f_k \; ; \; i=1,2, \ldots \ell \qquad (A5.31)$$

$$g_i^{(\ell)} = 0; \; i > \ell$$

$$[S(m+1)]_{ij} = \frac{1}{\sqrt{\Upsilon}} \frac{\zeta_{j-1}}{\zeta_j} \left\{ [S(m)]_{ij} - \frac{f_j g_i^{(j-1)}}{\zeta^2_{j-1}} \right\} \qquad (A5.32)$$

The algorithm given in the section can be used to update S(m+1).

Peterka's Square Root Algorithm for Updating S(nm+1)

1. $\zeta_0 = \sqrt{\Upsilon}$

2. $\zeta_0^2 = \Upsilon$

For j=1,2, ...,n recursively evaluate (3) to (13)

3. $f_j = \sum_{i=1}^{j} [S(m)]_{ij} \, [x(m+1)]_i$

4. $a = \zeta_{j-1}/\sqrt{\Upsilon}$

5. $b = f_j/\zeta^2_{j-1}$

6 $\zeta_j^2 = \zeta^2_{j-1} + f_j^2$

7. $\zeta_j = \sqrt{\zeta_j^2}$

8. $C = a/\zeta_j$

9. $g_j = [S(m)]_{jj} f_j = C[S(m)]_{jj}$

For i=1,2, ... j-1 recursively evaluate (11) to (13).

11 $d = [S(m)]_{ij}$

12 $[S(m+1)]_{ij} = C[d - bg_i]$

13 $g_i = g_i = df_j + g_i$

Notes

1. The variables f_j, ζ_j, and ζ_j^z do not need to be indexed and may be considered as scalars in a computer program

2. The argument m for S and x does not play any part in a computer program, but is included to make the algorithm esay to understand.

3. $\zeta = \alpha$

A FORTRAN program for the square root filter is given in appendix A12.

References for appendix A5.

[1] PETERKA C.

A Square Root Filter for Real-time Multivariate Regression Kybernetika, 11, 1975, pp53-67.

[2] CLARKE D.W. COPE S.N. and GAWTHROP P.J.

Feasibility Study of the Application of Microprocessors to Self-tuning Controllers. O.U.E.L report, 1137/75, Oxford, 1975.

[3] GILL P.E., GOLUB G.H., MURRAY W. and SAUNDERS M.A.

Modifying Matrix Factorization . Math. Comp., 28, 1974, pp505-535.

Appendix A6 U-D Factorization Algorithm.

The factorization given here is due to Bierman [1,2].

The covariance matrix P can be updated as:

$$\begin{aligned} P(m+1) &= \frac{1}{\gamma}[P\ m) - \frac{1}{\alpha}P(m)x(m+1)x^T(m+1)P\ m)] \\ &= \frac{1}{\gamma}[P(m) - Gx^T(m+1)P(m)] \qquad \text{(A6.1)} \end{aligned}$$

where $\alpha = \gamma + x^T(m+1)P(m+1)x(m+1)$ and

$G = P(m)x(m+1)/\alpha$

If P(m) is factored as $P(m) = U(m)D(m)U^T(m)$, where U(m) is an upper triangular matrix with unit diagonal elements, and D(m) is a diagonal matrix, then (A6.1) can be written as

$$\begin{aligned} P(m+1) &= \frac{1}{\gamma}[U(m)D(m)U^T(m) - \frac{1}{\alpha}U(m)D(m)U^T(m)x(m+1) \\ &\qquad . \ x^T(m+1)U(m)D(m)U^T(m)] \\ &= \frac{1}{\gamma}U(m)[D(m) - \frac{1}{\alpha}D(m)U^T(m)x(m+1)x^T(m+1)U(m)D(m)]U^T(m) \\ &= \frac{1}{\gamma}U(m)[D(m) - \frac{1}{\alpha}vv^T]U^T(m) \qquad \text{(A6.2)} \end{aligned}$$

where,

$v = D(m)U^T(m)x(m+1)$

and use has been made of the fact that $D = D^T$, D being a diagonal matrix.

If the term in the square bracket is factored as:

$$\bar{U}(m)\bar{D}(m)\bar{U}^T(m) = [D(m) - \frac{1}{\alpha}vv^T]$$

where the bar is used to distinquish the U-D factors of $[D(m) - vv^T]$ from those of P, then (A6.2) becomes:

$$\begin{aligned} P(m+1) &= U(m+1)D(m+1)U^T(m+1) \\ &= \frac{1}{\gamma}U(m)\bar{U}(m)\bar{D}(m)\bar{U}^T(m)U^T(m) \qquad \text{(A6.3)} \end{aligned}$$

Noting that the product of two upper triangular matrices is itself an upper triangular matrix, and the symmetry in (A6.3), then:

$$U(m+1) = U(m)\overline{U}(m) \; ; \; D(m+1) = \frac{\overline{D}(m)}{\gamma} \qquad \text{(A6.4)}$$

Thus, if suitable forms of $\overline{U}(m)$ and $\overline{D}(m)$ can be found, $U(m+1)$ and $D(m+1)$ can be updated.

If γ in (A6.1) is set equal to unity, and $\gamma = \sigma^2$ in (A6.1a) where σ^2 is the variance of the error term, the presentation given here is the same as that given by Bierman [1]. Bierman's algorithm for updating $U(m+1)$ and $D(m+1)$, modified as stated above, is given below. To reduce the number of arithmetic operations and storage, Bierman's algorithm avoids the matrix multiplication implied in (A6.4) by using a special matrix factorization. The algorithm is given below:

For j=2,..,n recursively evaluate (1) to (4):

1. $\alpha_j = \alpha_{j-1} + v_j f_j$
2. $\overline{d}_j(m) = d_j(m)\alpha_{j-1}/\alpha_j \; ; \; d_j(m+1) = \overline{d}_j(m)/\gamma$
3. $U_j(m+1) = U_j(m)\overline{U}_j(m) = U_j(m) + \lambda_j b_j ; \; \lambda_j = -f_j/\alpha_{j-1}$
4. $b_{j+1} = b_j + v_j U_j(m+1)$

where $f = U^T(m)x$

$v = D(m)f$

$\overline{d}_1 = d_1(m)/\alpha_1 \; ; \; d_1(m+1) = \overline{d}_1$

$\alpha_1 = \gamma + v_1 f_1$

$b^T{}_2 = [v_1 \; \underbrace{0 \quad 0 \ldots \; 0}_{n-1}]$

Note 1

$D = [\; d_1 \quad d_2 \quad \ldots \quad d_n]$

$b_j^T = [b_j(1) \quad b_j(2) \; \ldots \; b_j(n)]$

$U_j^T(m) = [U_j(1) \; \ldots . \; U_j(n)]$

$U_j(m)$, $U_j(m+1)$ are the jth columns of the matrices $U(m)$ and $U(m+1)$, respectively.

Note 2.

$$\alpha = \gamma + x^T(m+1)U(m)D(m)U(m)x(m+1)$$

$$= \gamma + vf = \gamma + \sum_{i=1}^{n} f_i v_i$$

Writing this out explicitely, for each step in the iteration

$i = 1 \quad \alpha_1 = \gamma + f_1 v_1$

$i = 2 \quad \alpha_2 = \gamma + f_1 v_1 + f_2 v_2 = \alpha_1 + f_2 v_2$

$i = j \quad \alpha_j = \alpha_{j-1} + f_j v_j$

After the last iteration, α_j in step 1 of the algorithm gives α of equation (A6.1a).

Note 3.

Step 2 of the algorithm updates the elements of D, the diagonal elements of P. Step 3 updates the elements of U(m+1); Step 4 computes the unweighted Kalman gain vector:

$b_j = \alpha G$

From section 4.2 in chapter 4, $\hat{\theta}$ may be written as:

$$\hat{\theta}(m+1) = \hat{\theta}(m) + G[y(m+1) - x^T(m+1)\hat{\theta}(m)].$$

so that all that is necessary to update $\hat{\theta}$ is G, and this is readily obtained from b_j. Thus although P(m+1) can be obtained from $P(m+1) = U(m+1)D(m+1)U^T(m+1)$, it is unnecessary to compute P(m+1).

A FORTRAN program for updating the U-D factors of P is given in appendix A12.

Some Properties of the U-D Algorithm.

To gain insight into some properties of the U-D algorithm it is useful to write out the algorithm explicitely. Thus for a 2-parameter model (using the more direct form of the U-D algorithm given in chapter 6):

1. $v_1 = x_1(m+1)$; $v_2 = x_2(m+1) + U_{12}(m)x_1(m+1)$

2. $b_1 = d_1(m)x_1(m+1)$; $b_2 = d_2(m)v_2$

3. $\alpha_1 = \gamma + b_1v_1 = \gamma + d_1(m)x_1(m+1)x_1(m+1)$

4. $d_1(m+1) = d_1(m)/\alpha_1$

5. $\alpha_2 = \alpha_1 + b_2v_2 = \alpha_1 + d_2(m)v_2v_2$

6. $U_{12}(m+1) = U_{12}(m+1) - (b_1v_2)/\alpha_1$

7. $b_1 = b_1 + b_2U_{12}(m) = d_1(m)x_1(m+1) + d_2(m)v_2U_{12}(m)$

8. $d_2(m+1) = (d_2\alpha_1)/(\gamma\alpha_2)$

The U-D elements are updated at steps (4), (6) and (8). An examination of the algorithm above will show that (steps 3 and 4) if the values used to initialize the U-D elements were positive, then α_1 and hence $d_1(m+1)$ will always be positive. For the same reason α_2 (which is the same as α in the general RLS algorithm - see note 2 above and equation 4.4 in chapter 4) and d_2 will also be positive. Thus it is seen that the D elements are always positive, and this guarrantees the positive definiteness of the covariance matrix P.

Notes.

(i) The analysis above assumes of course that γ is positive.

(ii) A matrix P is positive definite if and only if $x^T P x > 0$ [3,4], except when $x_1 = x_2 = \dots x_n = 0$

Now, $\alpha = \gamma + x^T U D U^T x = \gamma + x^T P x$ (see note 2 above), which for a 2-parameter model becomes (see steps 3 and 5 of algorithm):

$$\alpha_2 = \alpha_1 + d_2 v_2 v_2 = \gamma + d_1(m) x_1^2(m+1) + d_2(m) v_2^2$$

so that in this case: $x^T P x = d_1(m) x_1^2(m+1) + d_2(m) v_2^2$.

As stated earlier, both $d_1(m)$ and $d_2(m)$ are always positive, so that the matrix P satisfies the positive definiteness property stated earlier. Thus, the U-D factorization method guarrantees the positive definiteness of P.

(iii) If there is no data, ie $x_1 = x_2 = \dots x_n = 0$, then $\alpha_1 = \gamma$ (step 3), and d_1 will be continuously scaled by γ as will d_2. As discussed in the main text, this will lead to P increasing exponentially (since it is being divided by a quantity that is < 1). Thus it is seen that the problem of blow-up is not altogether eliminated as suggested in [5] by matrix factorization. However, as Bierman [3] pointed out the use of factorization algorithm considerably reduces the numerical ranges of numbers involved so that the blow-up problem may be less serious. More importantly in the OA work, the inherent system noise and/or noise picked up at the EOG electrodes ensure that the x's never become zero.

References for Appendix A6.

[1] BIERMAN G.J

Measurement Updating Using the UD Factorization Algorithm Automatica, 1976, 12, pp. 375-382.

[2] THORNTON C.L. and BIERMAN G.J.

Filtering and Error Analysis via the UDU Covariance Fcatorization. IEEE Trans. on Automatic Control, 1978, 23, pp. 901-907.

[3] BIERMAN G.J.

Factorization Methods for Discrete Sequential Estimation. Academic Press, 1977.

[4] BAJPAI A.C., CALUS I.M., and FAIRLEY J.A.

Mathematics for Engineers and Scientists, Vol 2, John Wiley, 1973.

[5] CLARKE D.W.

Some Implementation Considerations of Self-tuning Controllers. In Numerical Techniques for Stochastic Systems, ed ARCHETTI F. and CUGIANI, North Holland, 1980, pp81-110.

Appendix A7 Recursive Estimation of the Residual Variance and the Multiple correlation Coefficient.

The residual variance, S^2, and the multiple correlation coefficient, R^2, for the batch method may be defined as in (A7.1) [1]:

$$S^2 = SS/M-n \tag{A7.1a}$$

$$R^2 = SS/SS_T \tag{A7.1b}$$

where M and n are the number of samples and the number of parameters in the model, respectively. SS, the sum of squares of the residuals and SS_T, the total sum of squares are, respectively, given by:

$$SS = \sum_i e^2(i) \quad ; \quad SS_T = \sum_i y^2(i)$$

y(i), e(i) are, respectively, the ith samples of the measured and corrected EEG. In the recursive algorithms it is desirable to obtain S^2 and R^2 recursively. The recursive forms of S^2 and R^2 used in the investigation are justified here. From chapter 4, the sum of squares at each sample point may be written as:

$$\begin{aligned} SS(m) &= \gamma\, SS(m-1) + e^T(m)e(m) \\ &= \gamma\, SS(m-1) + e^2(m) \end{aligned} \tag{A7.2}$$

since e(m) is a scalar. Similarly,

$$SS(m-1) = \gamma\, SS(m-2) + e^2(m-1)$$

Thus on substituting into (A7.2), SS(m) becomes:

$$SS(m) = \gamma^2 SS(m-2) + \gamma\, e^2(m-1) + e^2(m)$$

By successive substitution for SS(m-2), SS(m-3) etc and simplfying SS(m) becomes:

$$SS(m) = \sum_{i=1}^{m} \gamma^{m-1} e^2(i) \tag{A7.3}$$

which is the same as the expression for the sum of squares for the exponentially weighting scheme (eqn 4.3 in chapter 4). Thus (A7.2) converges to the required form. In view of (A7.2), S(m+1) may be written as:

$$SS(m+1) = \gamma\, SS(m) + e^2(m+1) \qquad (A7.4)$$

By a similar argument, the total sum of squares given by (A7.5) also converges to the true total sum of squares.

$$SS_T(m+1) = \gamma\, SS_T(m) + y^2(m+1) \qquad (A7.5)$$

In the exponential weighting scheme, the number of samples that effectively contribute to the estimation of θ(m+1) at each sample point is called the asymptotic sample length (ASL) [2] and is given by:

$$ASL = \sum_{i=0}^{\infty} \gamma^i = \frac{1}{1-\gamma} \qquad (A7.6)$$

Thus m in (A7.1a) may be replaced by ASL. A recursive form for ASL is given in (A7.7):

$$asL(m+1) = 1 + \gamma\, asL(m) \qquad (A7.7)$$

By a similar argument to that for the sum of squares, asL(m+1) can be shown to converge to the true ASL, that is

$$asL(m+1) = 1 + \gamma\, asL(m) = \sum_{i=0}^{\infty} \gamma^i \qquad (A7.8)$$

To compute S^2 and R^2 recursively, all that is now required is to use equations (A7.2), (A7.5) and (A7.8) as in equations equations (4.17) and (4.18) in chapter 4.

References for Appendix A7.

[1] JOHNSTON J.

Econometric Methods. McGraw Hill, 1972.

[2] CLARKE D.W.

Introduction to self-tuning Controllers, in HARRIS C.J. and BILLINGS S.A. (Eds) Self-tuning and Adaptive Control. Peter Peregrinus, 1981, pp. 36-71.

Appendix A8 OAR System Input/Output Connections to the EEG Machine and Backplane Connections

This appendix describes the interconnections to the OAR system from the EEG machine, to the data terminal equipment, and the backplane connections. The OAR system has one analogue port, one Rs-232 serial port, and several slots for the boards which are interconnected at the backplane.

(a) The Analogue I/O Port.

The analogue I/O port is a 37-pin female D connector. The interconnections to the EEG machine are made via this port. Table A8.1 lists pin numbers, signals, and their descriptions for the port. The pin assignment of table A8.1 allows a direct connection of the OAR system to a port of the 8 channel Universal Electroencephalograph (or simply EEG machine) via a shielded cable. For connection to other EEG machines, reference should be made to the manufactures handbook and a suitable cable and connector used.

(b) RS-232 Serial I/O Port.

This port connects the OAR system to a data terminal equipment and allows communication with the system. The connector on the OAR system is a standard 25-pin female D connector. Table A8.2 lists the pin numbers, signal names and descriptions for the port. Reference should be made to the 'VMVE110 Monoboard Microcomputer User's Manual', by Motorola, for a detailed description of how to communicate with the microcomputer board of the OAR system.

(c) Backplane and Eurocard Connections.

Tables A8.3 to A8. list the pin assignments for the connectors at the backplane of the OAR system. Three types of connectors are used. A 96 way connector consisting of three rows of pins labelled rows A, B, and C; a 64-way connector consisting of two rows labelled rows A and B; and a 32-way connector consisting of a single row, labelled row A. These connectors are all standard DIN 41612 connectors. The connectors on the backplane are designated P1, P2 etc. from left to right on the OAR system starting from the microcomputer board as shown below.

Port number	Description or name of port
Port 1 (P1)	VMEbus
Port 2 (P2)	I/O channel (not used)
Port 3 (P3)	Decoder (A)
Port 4 (P4)	Decoder (B)
Port 5 (P5)	FP hardware board (reserved)
Port 6 (P6)	(not used)
Port 7 (P7)	ADC/DAC (A)
Port 8 (P8)	ADC/DAC (B)
Port 9 (P9)	(not used)
Port 10(P10)	Analogue O/P board
Port 11(P11)	(not used)
Port 12(P12)	Analogue O/P board (reserved)
Port 13(P13)	(not used)
Port 14(P14)	Analogue I/P board
Port 15(P15)	(not used)
Port 16(P16)	Analogue I/P board (reserved)

(d) Characteristics of the EEG Machine

In this section, typical signal levels and input/output impedances of the EEG machine are given. These parameters relate to the Universal EEG machine in particular, but are in general typical of most EEG machines.

Output signal level from EEG machine	up to 2V peak
Input signal level to the EEG machine	"
Input impedance	200MΩ, 500pF
Analogue inputs	0.5V/CM, 100KΩ

Table A8.1 Analogue I/O port pin assignment.

pin no.	Input signal name and description	Pin no.	Output signal name and description
1	EOG1 signal	20	EOG/EEG
2	EOG2 signal	21	"
3	EOG3 signal	22	"
4	EOG4 signal	23	"
5	EEG1 signal	24	"
6	EEG2 signal	25	"
7	EEG3 signal	26	"
8	EEG4 signal	27	"
9	EEG5 signal	28	"
10	EEG6 signal	29	"
11	EEG7 signal	30	"
12	EEG8 signal	31	not connected
13	EEG9 signal	32	EOG/EEG
14	EEG10 signal	33	"
15	EEG11 signal	34	"
16	EEG12 signal	35	"
17	Ground	36	not connected
18	not connected	37	not connected
19	not connected		

EOG/EEG indicates that either the EOG or the EEG (corrected or uncorrected) could be output through the pin , depending on the response of the user to the requests for system constants when the OAR program is run (see appendix A15).

Table A8.2 RS232 Serial Port Connector pin assignments.

Pin number	Signal Mnemonic	Signal name and description
2	TXD	Transmit data from terminal
3	RXD	Receive data to terminal
5	CTS	Clear to send
6	DSR	Data set ready
7	GND	Signal ground
8	DCD	Data carrier detect
20	DTR	Data terminal ready.

Table A8.3 VMEbus Backplane and Eurocard Connectors. Port 1.

Pin Number	Row A Signal Mnemonic	Row B Signal Mnemonic	Row C Signal Mnemonic
1	D00		D08
2	D01		D09
3	D02		D10
4	D03		D11
5	D04		D12
6	D05		D13
7	D06		D14
8	D07		D15
9	GND		GND
10	SYSCLK		$\overline{\mathrm{SYSFAIL}}$
11	GND		$\overline{\mathrm{BERR}}$
12	$\overline{\mathrm{DS1}}$		$\overline{\mathrm{SYSRESET}}$
13	$\overline{\mathrm{DS0}}$		
14	R/$\overline{\mathrm{W}}$		
15	GND		
16	$\overline{\mathrm{DTACK}}$	AM0	
17	GND	AM1	GND
18	$\overline{\mathrm{AS}}$	AM2	
19	GND	AM3	GND
20	$\overline{\mathrm{IACK}}$	GND	
21	$\overline{\mathrm{IACKIN}}$		
22			
23	AM4	GND	A15
24	A07		A14
25	A06	IRQ6	A13
26	A05		A12
27	A04		A11
28	A03		A10
29	A02		A09
30	A01		A08
31	+12 V		-12 V
32	+5 V	+5V	+5V

D00 - D15	Data bus
A01 - A15	Address bus
GND	Ground
SYSCLK	System clock (16 MHz)
$\overline{\mathrm{DS0}}$,$\overline{\mathrm{DS1}}$	Data strobe lines
R/$\overline{\mathrm{W}}$	Read/Write line
$\overline{\mathrm{DTACK}}$	Data transfer acknowledge line
$\overline{\mathrm{AS}}$	Address strobe line
AM0-AM4	Address modifier code lines
$\overline{\mathrm{IACK}}$	Interrupt acknowledge line
$\overline{\mathrm{IRQ6}}$	Interrupt request line no. 6 (others not shown)

All signal lines are from the 68000 processor.

Table A8.4 Decoder Backplane and Eurocard Connectors.

Port 3.

Pin Number	Row A Signal Mnemonic	Row B Signal Mnemonic
1	GND	GND
2	D00	D08
3	D01	D09
4	D02	D10
5	D03	D11
6	D04	D12
7	D05	D13
8	D06	D14
9	D07	D15
10	GND	GND
11	SYSCLK	
12		FS
13	$\overline{\text{DS1}}$	
14	$\overline{\text{DS0}}$	CLK
15	R/$\overline{\text{W}}$	
16		
17	$\overline{\text{DTACK}}$	AM0
18		AM1
19	AS	AM2
20	CE	AM3
21	STS	$\overline{\text{DSTB}}$
22	$\overline{\text{CS}}$	FP
23		GND
24	AM4	A15
25	A07	A14
26	A06	A13
27	A05	A12
28	A04	A11
29	A03	A10
30	A02	A09
31	A01	A08
32	+5V	+5V

D00 - D15	Data bus from the µP
A00 - A15	Address bus from the µP
GND	Ground
SYSCLK	System clock (16Hz) from the µP
$\overline{\text{DS0}}$,$\overline{\text{DS1}}$	Data strobe lines from the µP
R/$\overline{\text{W}}$	Read/Write line from the µP
$\overline{\text{DTACK}}$	Data transfer acknowledge line µP.
$\overline{\text{AS}}$	Address strobe line from the µP
AM0 - AM4	Address modifier code lines from the µP
CE,STS	Control signal lines to the ADC
$\overline{\text{CS}}$	Control signal line to the DAC
FS	Sampling signal
CLK	clock (8MHz)
$\overline{\text{DSTB}}$	Data strobe signal (local)
FP	Floating point hardware control signal (reserved)

Table A8.5 Decoder Backplane and Eurocard Connectors.
Port 4.

Pin Number	Row A Signal Mnemonic	Row B Signal Mnemonic
1	GND	GND
2		
3		PTM
4	PGND	
5		
6		FS
7		
8	MXO	MX3
9	MX1	MX4
10	MX2	
11		
12	DXO	DX3
13	DX1	DX4
14	DX2	
15		
16	PGA	WDSTB
17	GND	WDSTS
18		
19	GND	
20		
21	GND	GND
22	SH1	SH11
23	SH2	SH12
24	SH3	SH13
25	SH4	SH14
26	SH5	SH15
27	SH6	SH16
28	SH7	SH17
29	SH8	SH18
30	SH9	SH19
31	SH10	SH20
32	+5V	+5V

MXO - MX4	Multiplexer control lines (to ADC/DAC board)
DXO - DX4	Demultiplexer control lines (to ADC/DAC board)
PGA	Programmable gain amplifier control line "
SH1 - SH20	Output sample and hold control lines(to the analogue output board)
WDSTB	Strobe signal for the window det. (reserved)
WDSTS	Status control line for the window det. (to the ADC/DAC board)
PTM	Interrupt signal from the prog. timer (the sampling signal is derived from this signal).
FS	Sampling signal (to the analogue input board)
PGND	Ground from the prog. timer module.

Table A8.6 ADC/DAC Backplane and Eurocard Connectors. Port 7.

	Row A
Pin	Signal
Number	Mnemonic
1	GND
2	D11
3	D10
4	D09
5	D08
6	D07
7	D06
8	D05
9	D04
10	D03
11	D02
12	D01
13	D00
14	N.C
15	N.C
16	N.C
17	N.C.
18	N.C
19	GND
20	CE
21	STS
22	$R/\overline{W}$
23	$\overline{CS}$
24	A1
25	N.C
26	AGND
27	N.C
28	-15V
29	N.C
30	+15V
31	N.C
32	+5V

D00 - D11	Data lines from the µP
CE,STS	Control lines from the decoder (to the ADC)
$R/\overline{W}$	Read/Write lines from the µP
$\overline{CS}$	Control line from the decoder (to the DAC)
A1	Address line from the µP
GND	Digital ground
AGND	Analogue ground

Table A8.7 ADC/DAC Backplane and Eurocard Connectors.

Port 8.

Pin Number	Row A Signal Mnemonic	Row B Signal Mnemonic
1	GND	GND
2	PGA	WDSTS
3	GND	GND
4	S7	S6
5	S5	S4
6	S3	S2
7	S1	MX0
8	MX1	MX2
9	MX3	MX4
10	S8	S9
11	S10	S11
12	S12	S13
13	S14	S15
14	S16	S17
15	S18	S19
16	S20	GND
17	GND	OS7
18	OS6	OS5
19	OS4	OS3
20	OS2	OS1
21	DX0	DX1
22	DX2	DX3
23	DX4	GND
24	OS8	OS9
25	OS10	OS11
26	OS12	OS13
27	OS14	OS15
28	OS16	OS17
29	OS18	OS19
30	OS20	+15V
31	N.C.	N.C.
32	AGND	-15V

PGA	Programmable gain amplifier control line (fr decoder)
S1 - S20	Analogue samples (from analogue input chns)
OS1 - OS20	Analogue samples (to analogue output chs)
MX1 - MX4	Multiplexer control signals (fr decoder)
DX0 - DX4	Demultiplexer control signals (fr decoder)
GND	Digital ground
AGND	Analogue ground
N.C	Not connected
WDSTS	Window det. control line (fr decoder)

Table A8.8 Analogue output Backplane and Eurocard Connectors. Port 10

Pin Number	Row A Signal Mnemonic	Row B Signal Mnemonic
1	GND	GND
2		"
3		"
4		"
5		"
6		"
7		"
8		"
9		OP1
10		OS1
11		OP2
12		OS2
13		GND
14		SH2
15		SH1
16		GND
17		OP3
18		OS3
19		OP4
20		OS4
21		GND
22		GND
23		SH3
24		GND
25		OP5
26		OS5
27		OP6
28	+15	OS6
29		GND
30	-15V	SH6
31		SH5
32		

OP1 - OP6	Analogue output signals (to the EEG machine)
OS1 - OS6	Analogue samples(from the decoder)
SH1 - SH6	Control signals to the out put sample and Holds (from the decoder)
GND	Digital ground.

Appendix A9 - Software Floating Point Arithmetic Routines for the 68000µP

As described in the main text, there are three routines for performing software floating point arithmetic operations in the OAR, and additionally a common routine via which control is returned to the calling program. These routines will now be described in turn.

(a) EXIT Routine

This routine is used to handle 'exceptions', such as overflow, to recover the sign of the result, to put the result in the correct floating point format and to return control to the calling routine. Three exceptions are handled:

(i) Result is zero or too small: If the result of an operation is zero or the number is too small to be represented (under-flow), this routine sets the result to zero.

(ii) Result overflowed: If there is overflow in the arithmetic operation, the result is set to the maximum floating number possible. That is, $(1-2^{-15}) \times 2^{63}$. In practice it is set to $007F7FF, then the mantissa is given the appropriate sign.

(iii) Over- or under-flow detection: In excess 64 representation, the exponent of a valid floating point number is positive. Over- or under-flow causes it to be negative. Thus, the sign of the exponent is used to detect over- or under-flow. Additionally, under-flow generates a carry bit whereass over-flow does not. The EXIT routine determines which of the two has occurred and sets the result to either FPMAX or to zero, which ever is appropriate.

The listing of the EXIT routine in 68000 µP assembly language is given in figure A9.1.

(b) Floating Point Addition Routine (FADD)

This routine is used to perform addition or subtraction. In the case of subtraction, the second operand is negated in the calling routine with a single instruction (e.g NEG.W D2) before entering the addition routine. Before adding two floating point numbers, their exponents must be made equal. This is called alignment, and involves shifting right the mantissa of the smaller operand and incrementing its exponent until it equals that of the larger operand. The aligned numbers are then added. Thus if X1 and X2 are two floating point numbers to be added, where $X1 = M1 \times 2^{E1}$ and $X2 = M2 \times 2^{E2}$, then their sum X, is given by:

$$X = M \times 2^{E}$$

where $M = (M1+M2 \times 2^{E1-E2})$, $E = E_1$ and $X1 > X2$.

That is, the exponent of the result is the exponent of the larger operand and its mantissa is the sum of the mantissas of the two operands properly aligned.

```
*
*
* EXIT ROUTINE
* THIS ROUTINE HANDLES EXCEPTIONS FROM FLOATING POINT ARITHMETIC
* OPERATIONS AND RETURNS CONTROL TO THE CALLING ROUTINE.
*
DFORMAT SWAP.W  D1                              ;PUT RESULT IN CORRECT FP FORMAT
        MOVE.W  D3,D1
        MOVE.L  D1,D3
SIGNS:  SUB.B   #1,D5                           ;RECOVER SIGN OF MANTISSA
        BNE     EXITF
        NEG.W   D3
EXITF:  MOVEM.L (A7)+,D0-D2/D4-D7
        RTS
ZEROF:  CLR.L   D3                              ;RESULT IS ZERO
        BRA     EXITF
UOFLW:  BTST    #15,D6                          ;RESULT UNDER OR OVERFLOW?
        BNE     ZEROF                           ;UNDERFLOW.SET RESULT TO ZERO
OFLW:   MOVE.L  FPMAX,D3                        ;OVERFLOW.SET RESULT TO FPMAX
        BRA     SIGNS
```

Figure A9.1 The EXIT routine
Control is returned to the calling program via this routine.
The routine also handles under/overflow, recovers the sign
and puts the result in the correct format.

The sum of the aligned mantissas, M, may not be properly normalized. It may be too large (mantissa overflow) or it may be too small (mantissa under-flow). The former results when the mantissas of the original operands to be added (M1,M2) have the same signs, and the later when they have opposite signs. Mantissa over-flow and under-flow are corrected by renormalizing the sum and adjusting the result exponent, E, but this may lead to exponent over-flow or under-flow . Exponent over-flow means that the result is too large to be represented and under-flow that it is too small.

Figure A9.2 gives the flow chart for the FADD routines and the corresponding assembly language program is listed in figure A9.3. Referring to figure A9.2, the first operation is to align the operands so that their exponents are equal (boxes 1 to 5). This is done by shifting the mantissa of the operand with the smaller exponent right an amount equal to the difference in the exponents (the shift count). If the two exponents were equal or their difference exceeds 15 (operands too small), no alignment is carried out. In the latter case, the operation is terminated as the smaller operand is effectively zero and the result is simply the larger operand. Next, the aligned numbers are added (box 6) and the exponent of the result set equal to the exponent of the larger operand.

If the sum is not zero, it is then tested for mantissa over-flow or under-flow. The two cases are treated

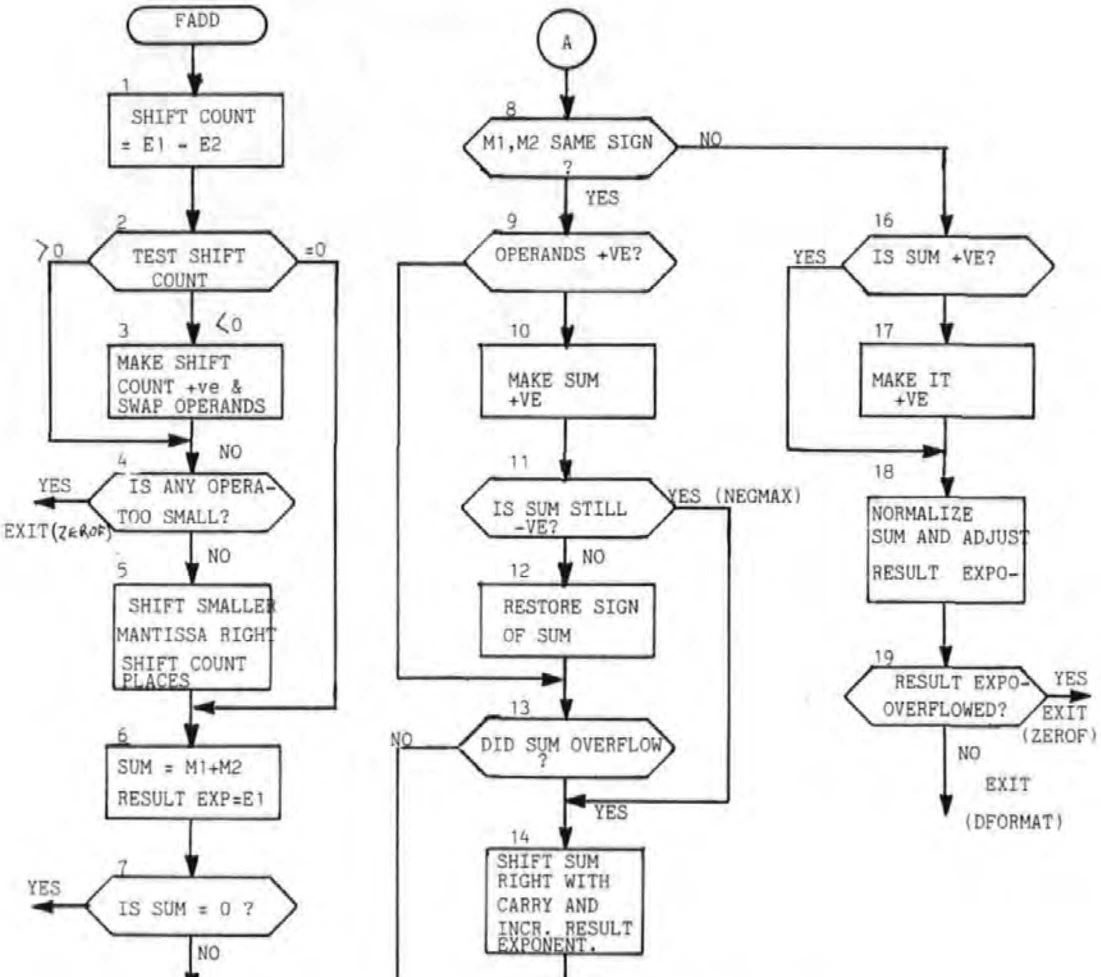


Figure A9.2 Floating point addition of two operands with exponents E1 and E2 mantissaa M1 and M2, respectively.

```
*---------------------------------------------------------------------
* SUBROUTINE FADD
* THIS SUBROUTINE ADDS TWO NORMALIZED FLOATING POINT NUMBERS IN
* D2 AND D3. THE RESULT IS RETURNED IN D3
*
FADD:   MOVEM.L D0-D2/D4-D7,-(A7)
        MOVE.L  D3,D1                   ;MAKE COPIES OF OPERANDS
        MOVE.L  D2,D0                   ;
        SWAP.W  D1                      ;RETRIEVE EXPONENTS (E) OF OPERANDS
        SWAP.W  D0
        MOVE.B  D1,D4
        SUB.B   D0,D4                   ;E2-E1=SHIFT COUNT(SC)
        BEQ     SCZRO                   ;SC IS ZERO
        BPL     SCPOS                   ;SC IS POSITIVE
        NEG.B   D4                      ;SC IS NEG;MAKE POSITIVE
        EXG     D0,D1                   ;SWAP OPERANDS
        EXG     D2,D3
SCPOS:  CMP.B   #15,D4                  ;ANY OPERAND TOO SMALL?
        BPL     EXITF                   ;YES. RESULT = LARGER OPERAND
        ASR.W   D4,D2                   ;NO. SHIFT SMALLER M RIGHT SC PLACES
SCZRO:  MOVE.W  D3,D6                   ;SAVE SIGN OF 2ND OP
        ADD.W   D2,D3                   ;OBTAIN SUM=M2+M1
        ROXR.W  #1,D5                   ;SAVE CARRY FLAG
        TST.W   D3                      ;IS SUM=0?
        BEQ     ZEROF                   ;IF YES,THEN EXIT
        EOR.W   D2,D6                   ;NO. ARE THE TWO OPERNDS SAME SIGN?
        BMI     DFSIGN                  ;OPERNDS ARE OPPOSITE SIGN
        MOVE.W  D5,D4                   ;OPERNDS ARE SAME SIGN
        SUB.W   D5,D5
        TST.W   D2                      ;IS SUM +VE OR -VE?
        BPL     OPRPOS                  ;SUM IS E +VE.
        NEG.W   D3                      ;SUM IS -VE. MAKE SUM +VE
        TST.W   D3                      ;IS SUM STILL -VE?
        BMI     NEGMX                   ;YES. SUM MUST BE NEGMAX
        NEG.W   D3                      ;NO.RECOVER SIGN OF SUM
OPRPOS: EOR.W   D4,D6                   ;DID OVERFLOW OCCUR DURING ADDITION?
        EOR.W   D3,D6                   ;
        BPL     DFORMAT                 ;NO. THEN FORMAT RESULT AND EXIT
        ROXL.W  #1,D4
NEGMX:  ROXR.W  #1,D3                   ;SHIFT SUM RIGHT WITH CARRY AND
        ADDQ.B  #1,D1                   ;INCR EXP: E=E+1
        BMI     OFLW                    ;CHECK THAT E DID NOT OVERLOW
        BRA     DFORMAT                 ;E IS OK. FORMAT RESULT AND EXIT.
DFSIGN: SUB.B   D5,D5
        TST.W   D3                      ;IS SUM POSITIVE?
        BPL     FNORMA                  ;YES. NORMALIZE RESULT
        MOVEQ.L #1,D5                   ;NO. SET FLAG AND MAKE POSITIVE
        NEG.W   D3
FNORMA: MOVEQ.L #15,D0                  ;SET UP REGS FOR NORMALIZTN
        MOVE.L  D0,D2
SHFTLA  ADD.W   D3,D3                   ;NORMALIZE SUM AND
        DBMI    D0,SHFTLA
        SUB.B   D0,D2
        LSR.W   #1,D3                   ;
        SUB.B   D2,D1                   ;ADJUST EXPONENT
        BMI     ZEROF                   ;E UNDERFLOW?
        BRA     DFORMAT                 ;NO.FORMAT RESULT AND EXIT
```

Figure A9.3

The Assembly language program for the floating point Addition.

separately and as mentioned earlier depend on the signs of the operands. Boxes 10 to 15 deal with mantissa over-flow, whilst boxes 16 to 19 deal with mantissa under-flow.

Mantissa over-flow is detected by successively performing an exclusive-OR operation on the sign bits of the two original operands, the sum, M, and the carry bit. The result of this operation is a (1) if there was over-flow and (0) otherwise [1]. Over-flow is corrected by shifting the sum M, right with carry and incrementing the result exponent, E, by one (box 14). If the exponent over-flows, the result is set to FPMAX with the appropriate sign, otherwise the resulting number is now properly normalized. A special form of over-flow may occur when the sum, M, is exactly equal to -1 ($008000 in hexadecimal) or NEGMAX, and this is separately detected. NEGMAX is the maximum negative mantissa and has no positive equivalent. Thus, when negated it still remains negative (boxes 10 and 11). When this condition is detected the sum, M, is normalized as in the ordinary mantissa overflow.

Mantissa under-flow is corrected by successively shifting left the sum, M, and decrementing the exponent, E, until the sum is properly normalized (box 18). The exponent is tested for under-flow at each shift. If exponent under-flow occurs at any stage the operation is terminated and the result set to zero. The sum is made positive (box 17) before normalizing to simplify the operation. The sign is recovered in the EXIT routine where the result exponent, E, and the normalized sum, M, are put into the proper floating point format.

(c) Floating Point Multiply Routine (FMUL)

If X1 and X2 are two FP numbers to be multiplied, where

$$X1 = M1 \times 2^{E1} \quad , \quad X2 = M2 \times 2^{E2}$$

then their product X, is given by:$X = (M1 \times M2)2^{E1+E2} = M \times 2^{E}$ where $M = (M1 \times M2)$, $E = (E1 + E2)$. Thus, the mantissas are multiplied and their exponents added. Since M1 and M2 are both normalized then their product, M, will be in the range, $0.25 \leq M < 1$. Thus the product, M, cannot over-flow but may not be properly normalized (mantissa under-flow).

The flowchart for the floating point multiply routine is given in figure A9.4, and the corresponding assembly language program listing given in figure A9.5. The mantissas of the operands are multiplied and the product tested to see if it is zero (boxes 1 and 2); If it is zero the operation is terminated and the result set to zero. The product, M, is double length (multiplication of two n-bit numbers gives a 2n bit result) and is thus rounded and reduced to single length. The exponents are then added and corrected for excess 64 (box 4). If the product M, is not normalized it is then normalized by a single left shift and the exponent decremented by 1. The sign of the exponent is tested. If it is negative, then there was exponent over-flow or under-flow when the

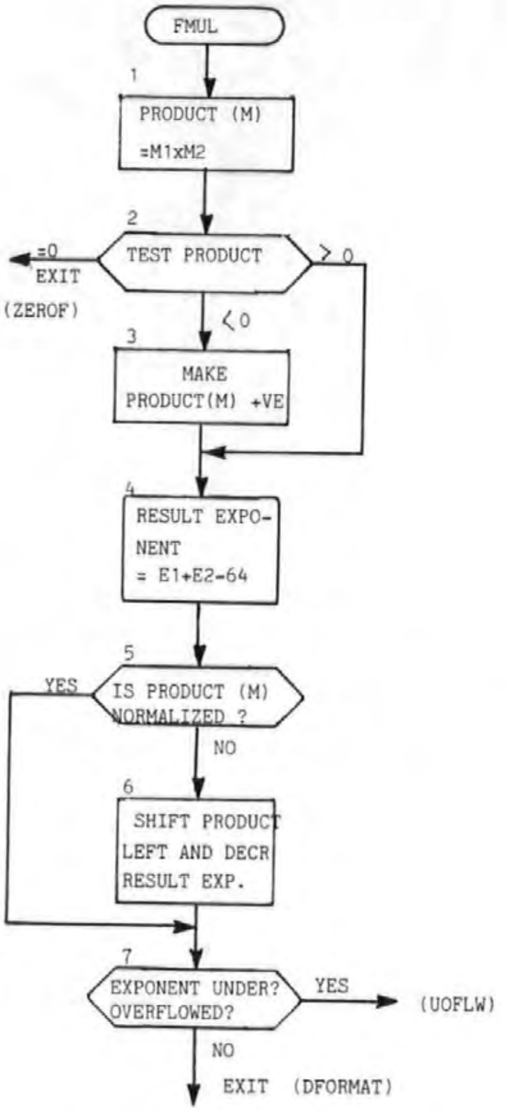


Figure A9.4 Floating point multiplication of two operands with exponents E1 and E2, mantissas M1 and M2, respectively.

```
*-----------------------------------------------------------------------------
*
* SUBROUTINE FMUL
* THIS SUBROUTINE MULTIPLIES TWO NORMALIZED FLOATING POINT NUMBERS
* IN D2 AND D3 AND RETURNS THE RESULT IN D3.
*
FMUL:    MOVEM.L D0-D2/D4-D7,-(A7)
         MOVE.L  D3,D1                     ;MAKE COPIES OF OPERANDS
         MOVE.L  D2,D0
         SWAP.W  D1                        ;RETRIEVE THE EXPS.
         SWAP.W  D0
         SUB.B   D5,D5                     ;CLEAR FLAG
         MULS    D2,D3                     ;OBTAIN PRODUCT M1*M2
         ADD.L   D3,D3                     ;SHIFT DLENGTH PRODUCT LEFT
         ADD.L   #32768,D3                 ;ROUND RESULT
         SWAP.W  D3                        ;CONVERT RESULT TO SINGLE LENGTH
         TST.W   D3                        ;RESULT=0,+VE OR -VE?
         BEQ     ZEROF                     ;RESULT IS =0. EXIT
         BPL     POSMUL                    ;RESULT IS +VE.COMPUTE EXP.
         ADDQ.B  #1,D5                     ;RESULT IS -VE.SET FLAG
         NEG.W   D3                        ;MAKE POSITIVE
POSMUL:  ADD.B   D0,D1                     ;E2+E1=E
         SUB.B   #64,D1                    ;REMOVE XS64
         ROXR.W  #1,D6                     ;SAVE CARRY FLAG
         BTST    #14,D3                    ;IS PRODUCT NORMALIZED?
         BNE     TESTE                     ;YES. TEST EXP.
         ADD.W   D3,D3                     ;NO.NORMALIZE PRODUCT AND
         SUBQ.B  #1,D1                     ;ADDJUST EXPONENT
TESTE:   TST.B   D1                        ;EXP UNDER/OVER FLOWED?
         BMI     UOFLW                     ;YES. BRACH TO UNDER/OVERFLOW TEST
         BRA     DFORMAT                   ;NO. FORMAT DATA AND EXIT.
```

Figure A9.5
The Assembly language program for the floating point multiply

exponents were added. The EXIT routine then determines whether it was under-flow or over-flow that occurred and takes appropriate action. It is to be noted that exponent over-flow or under-flow was not tested for when the exponents were added as subsequent exponent adjustment could correct it. The normalized product, M, and the adjusted exponent give the result for the floating point multiplication, except when the exponent over-flows or under-flows.

(d) Floating Point Divide Routine (FDIV)

In floating point division, the mantissas are divided and the exponents subtracted. Thus, division of X1 by X2, where

$$X1 = M1 \times 2^{E1} \quad , \quad X2 = M2 \times 2^{E2}$$

gives, $X = \frac{M1}{M2} \times 2^{E1-E2} = M \times 2^{E}$

where $M = \frac{M1}{M2}$, $E = E1 - E2$

To ensure that the quotient is within permissible range, M1 must be less than M2. This is readily achieved by shifting M1 one place to the right. After division, the quotient, M, may not be properly normalized. If it is not, it is shifted left one place and the exponent, E, decremented by one and checked for under-flow as before.

Figure A9.6 gives the flowchart for the floating point divide routine. The exponent, E, is obtained from the

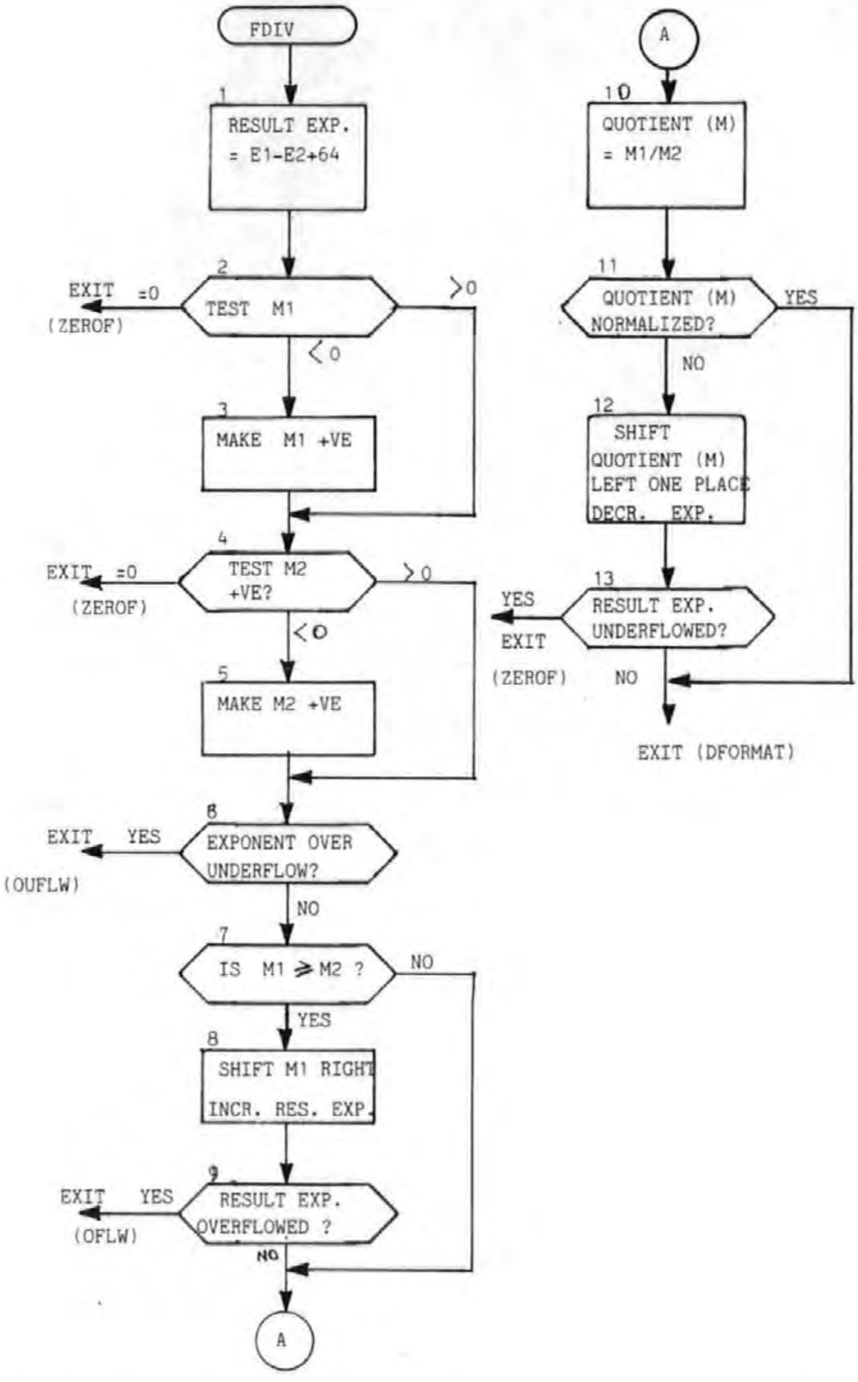


Figure A9.6 Floating point division of two operands with exponents E1 and E2, mantissas M1 and M2, respectively.

```
*
* SUBROUTINE FDIV
* THIS SUBROUTINE DIVIDES TWO NORMALIZED FLOATING POINT NUMBERS.
* ENTER WITH THE DIVIDEND IN D3 AND THE DIVISOR IN D2. THE RESULT
* IN THE CORRECT FLOATING POINT FORMAT IS RETURNED IN D3
*
FDIV:    MOVEM.L D0-D2/D4-D7,-(A7)
         MOVE.L  D3,D1                     ;MAKE COPIES OF OPERANDS
         MOVE.L  D2,D0
         MOVEQ.L #0,D3                     ;RETAIN ONLY M1 IN D3
         MOVE.W  D1,D3
         SWAP.W  D1                        ;RETRIEVE THE EXPS.
         SWAP.W  D0
         ADD.B   #64,D1                    ;ADD BIAS TO E1 (E1=E1+64)
         SUB.B   D0,D1                     ;OBTAIN EXP. E=E1-E2=
         ROXR.W  #1,D6                     ;SAVE CARRY FLAG
         SUB.B   D5,D5                     ;CLEAR SIGN FLAG
         MOVEQ.L #1,D7
         TST.W   D2                        ;IS M2=0,+VE OR -VE?
         BEQ     ZEROF                     ;M2=0,THEN EXIT
         BPL     TESTM1                    ;M2 IS +VE
         EOR.B   D7,D5                     ;M2 IS -VE
         NEG.W   D2                        ;MAKE M2 POSITIVE
TESTM1:  TST.W   D3                        ;IS M1=0,+VE OR -VE?
         BEQ     ZEROF                     ;M1=0,THEN EXIT
         BPL     EUFLOW                    ;M1 IS +VE
         EOR.B   D7,D5                     ;M1 IS -VE
         NEG.W   D3                        ;MAKE M1 POSITIVE
EUFLOW:  TST.B   D1                        ;DID EXP. OVER/UNDERFLOW?
         BMI     UOFLW                     ;IF YES EXIT
         CMP.W   D2,D3                     ;IS M2 .GE. M1?
         BLT     M2GT                      ;YES.THEN IT IS OK
         ASR.W   #1,D3                     ;NO. THEN DIVIDE M1 BY 2 AND
         ADDQ.B  #1,D1                     ;INCR E1 BY 1
         BMI     OFLW
M2GT:    SWAP.W  D3                        ;CONVERT M1 TO DOUBLE LENGTH
         ASR.L   #1,D3
         DIVS    D2,D3                     ;OBTAIN QUOTIENT OF M1/M2
         BTST    #14,D3                    ;IS QUOTIENT NORMALIZED?
         BNE     DFORMAT                   ;YES.THEN GO FORMAT RESULT
         ADD.W   D3,D3                     ;NO. NORMALIZE AND
         SUBQ.B  #1,D1                     ;ADJUST EXPONENT
         BMI     ZEROF                     ;
         BRA     DFORMAT                   ;GO FORMAT RESULT AND EXIT
*---------------------------------------------------------------------*
```

Figure A9.7

The Assembly language program for the floating point division.

exponents of the operands and the bias (ie 64, the so-called excess 64) is added (box 1). The two mantissaa are then tested. If either is zero, the operation is terminated and the result set to zero (boxes 2 to 4). The exponent of the result, E, is then tested for under- and over-flow and treated in the same way as in the multiply. For easy comparison of the mantissas and normalization of the product the mantissas of the operands are always made positive (boxes 3 to 5). The sign is restored in the EXIT routine. Division is performed with the inherent divide instruction of the 68000 µP processor (box 10). The assembly language program for the division routine is listed in figure A9.7.

[1] COPE S.N.
Floating-Point Arithmetic Routines and Macros for an
Intel 8080 Microprocessor. O.U.E.L Report 1123/75, 1975.

Appendix A10 - Floating Point/Fixed Point Decimal Number Conversions

At the testing stage of an algorithm it is desirabe for the microprocessor to communicate results in an easily understandable form so that their validity can be checked. Thus it is useful to be able to convert from hexadecimal floating point representation to decimal. In systems with high level language, this conversion is quite simple. If this facility is not available on the system, the following routines could be used.

(i) Floating Point to Fixed Point Decimal

When denormalized, a floating point number consists of an integer and a fractional part. The fractional part can be expressed as:

$$F = \sum_{i=0}^{n-1} b_i x \ 2^{i-n} \quad , \ b_i = 0 \text{ or } 1 \qquad (A10.1)$$

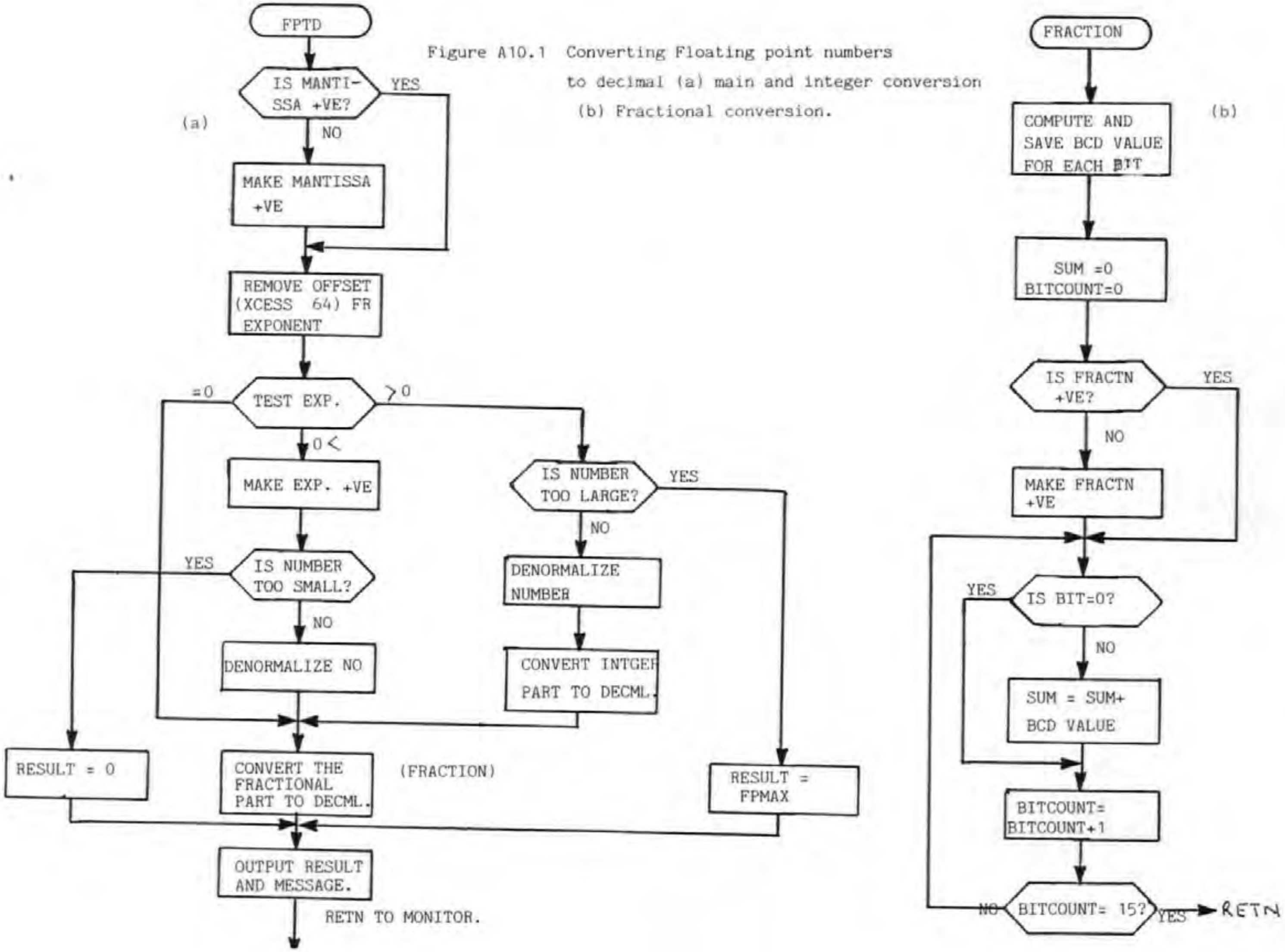
where n is the number of bits in the mantissa less one. The method used converts the fractional and the integer parts separately to decimal and then combines the results as shown in figure A10.1(a).

Conversion of the fractional part to decimal is achieved by a direct realisation of equation (A10.2). It is apparent from equation (A10.1) that the fractional part consists of fixed n binary related quantities. Thus, when the bi's are all (1) F is given given by:

$$F = 0.5 + 0.25 + 0.125 + \ \ldots \ + 0.0000305 = 0.999996 \qquad (A10.2)$$

In conversion, each of the fixed 15 (n=15) quantities in

Figure A10.1 Converting Floating point numbers to decimal (a) main and integer conversion (b) Fractional conversion.



equation (A10.2) are computed and stored in the memory in Binary Coded Decimal (BCD) at initialization. To convert to decimal, the bits of the fractional part of a denormalized number are successively tested starting from bit 0. At each stage if the bit is a (1), the corresponding BCD number is fetched and added to a running sum. When all the bits have been tested, the final sum gives the decimal equivalent of the fractional part. Figure A10.1 (b) gives the flow chart of the fractional conversion.

To convert the integer part to decimal the divide-by-10 method of binary to BCD 1 was used. In this method, the largest power of ten, say m, is found such that the following relationship holds:

$$10^{m} \leqslant B \qquad (A10.3)$$

where B is the maximum binary number to be converted. In the routine, B = 2^{16} = 65536. Thus, m = 4. To convert to decimal, the integer part of the denormalized number is first divided by 10 (expressed in binary) to give a quotient and a remainder. The quotient gives the most significant BCD digit. The remainder is then divided by 10 and the new quotient gives the next most significant BCD digit. This process is continued until the power of ten is zero. The resulting BCD digits give the decimal equivalent of the integer part. The flow chart for the method is given in figure A10.2 .

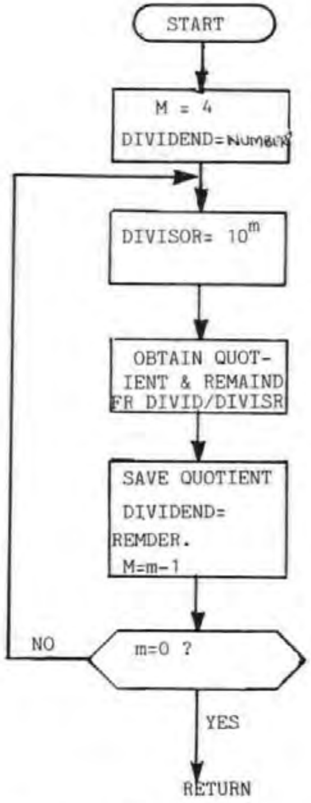


Figure A10.2 Divide by 10 method of binary to decimal conversion

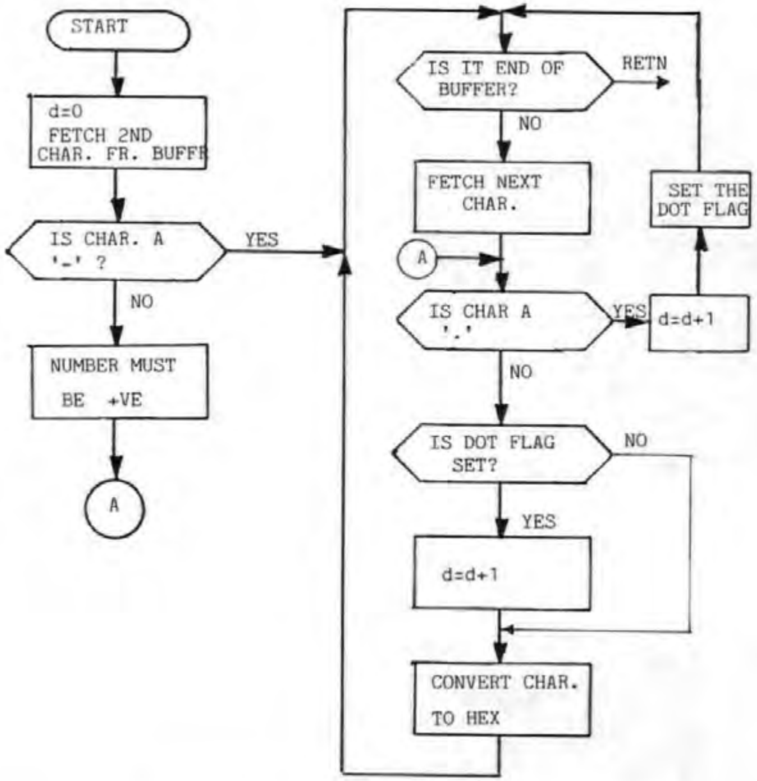


Figure A10.3 Reading and converting characters to hexadecimal

The integer part and the fractional part, all in BCD. together give the decimal equivalent of the floating point number.

(ii) Decimal to Hexadecimal Floating Point

To convert a number from decimal to floating point, express the number as $X = |(sgn)X/10^d|$, d=0,1, ..6. Where (sgn) is the sign of the number, d is the number of digits to the right of the decimal point. e.g. the decimal number -345.67 is expressed as sgn = 1, d = 2 and $X = 34567$.

In the conversion routine, the sign and 'd' of the decimal number are first obtained and saved, the number is then converted to decimal as an unsigned integer. The resulting hexadecimal number is normalized and then divided by 10^d (expressed in hexadecimal floating point) using the floating point divide routine described earlier. The result of this division, with the appropriate sign, gives the desired floating point number.

(iii) Examples of Conversion

In addition to the main conversion routines described above, there are other small routines used to output messages and to determine the type of conversion required. These routines use extensively some of the I/O routines in the VME110 monitor via trap 15 [2,3]. There is also a routine for reading and converting characters entered on the keyboard to hexadecimal. The flowchart for this routine is given in figure A10.3..

When the program is run it prompts the user to enter the number to be converted. Floating point numbers are entered in hexadecimal, prefixed by a dollar sign ($) whereas decimal numbers are entered in fixed point format, prefixed by the ampersand (&). Examples of the use of the routines are given below. The items underlined are those entered by the user and the rest are responses from the routines, apart from the comments which are included for clarity.

Examples of Conversions

In the following examples, the items underlined are those entered in response to prompts.

(i) Floating point to Decimal

When program is run the message below is output on to the terminal:

```
ENTER NUMBER
$404000                    (user enters FP number in HEX)

FP-TO-DECIMAL : $404000 = 0.500000
(result is 0.5 )
```

(ii) Decimal to Floating point

```
ENTER NUMBER
 &0.5
DECIMAL-TO-FP: $404000
```

References for Appendix A10

[1] TOMPKINS, W.J., WEBSTER, J.G. (ed)
Design of Microcomputer-Based Medical Instrumentation.
Prentice-Hall, 1981.

[2] VMEmodule Monoboard Microcomputer User's Manual.
Motorola Semiconductor, 1983.

[3] MC68000 16-Bit Microprocessor User's Manual.
Austin, Texas, Motorola Semiconductor, Inc., 1980.

```
SPACE    EQU      $0BA
STACK    EQU      $DFE
MSGBUF   EQU      $0F00
DBUFINT  EQU      $0900
DBUFINI  EQU      $0B00
FPNUM    EQU      $A60
NTYPE    EQU      $A6A                ;1 FOR FP-TO-DEC,0 FOR DEC-TO-FP.
RESULT   EQU      $A30                ;RESULT RESIDES IN $A28-$A2F
BASE     EQU      $A08
FDIV     EQU      $6FE
*
* MAIN PROGRAM
*
         LEA      STACK,A7
         MOVE.W   #10,BASE
         BSR      VALUES              ;FETCH NUMBR FOR CONVRSN
         TST.B    NTYPE               ;WHAT TYPE OF CONVSN REQUIRED?
         BNE      FPTD                ;
         BSR      FETCHDE             ;GO FETCH DECIMAL NO. AND CONVT TO HEX
         BSR      DTFP                ;CONVERT THE HEX TO FP
         BRA      OUTPUT1             ;GO PRINT RESULT
FPTD     BSR      INITLZE             ;CONVSN TO FP DESIRED.INIT
         BSR      FETCHFP             ;GO FETCH FP NUMBER
         SUB.L    D4,D4
         SUB.L    D5,D5
         MOVE.B   #0,D1               ;SAVE SIGN OF NUMBER
         TST.W    D3
         BPL      CNTNUE
         MOVE.B   #1,D1
         NEG.W    D3
CNTNUE   MOVE.L   D3,D2               ;EXTRACT EXPONENT OF NO.
         SWAP     D2
         SUB.B    #$40,D2
         BEQ      FRACTN              ;EXP IS 0,SO NUMBR IS A FRACTN
         BPL      INTEGR              ;EXP > 0
         NEG.B    D2                  ;EXP < 0 ,MAKE POSITIVE.
         CMP.B    #15,D2              ;AND CHK IF NUMBR IS TOO SMALL
         BLE      FRACTN              ;NOT TOO SMALL,GO CONVERT.
         BRA      OUTPUT2             ;TOO SMALL, SO EXIT
INTEGR   CMP.B    #16,D2              ;IS NUMBR TOO LARGE?
         BGT      MAXINT              ;YES. SET TO MAX INTGR
         ANDI.L   #$FF,D2             ;NO. MASK OFF HI BITS
         MOVE.W   D3,D5               ;MULT MANTISSA BY EXP
         ASL.L    D2,D5               ;(M*2**E)
         MOVE.W   D5,D3               ;PUT FRACTNL PART IN D3
         ANDI.L #$7FFF,D3
         ASL.L    #1,D5               ;AND INTGR PART IN D5
         SWAP     D5
         ANDI.L   #$FFFF,D5
         SUB.L    D2,D2
         BSR      INTEGER             ;CONVERT INTGR TO DEC
FRACTN   AND.W    #15,D2
         BSR      FRACTION            ;CONVERT FRACTN TO DEC
         BRA      OUTPUT2             ;PRINT RESULT AND EXIT
MAXINT   MOVE.L   #65535,D5           ;SET RESULT TO MAX VALUE
         BRA      OUTPUT2
*---------------------------------------------------------------------
```

```
*
* THIS SUBROUTINE ADDS 16 DIGITS BCD NUMBERS
* A4 POINTS TO ONE OF THE OPERANDS,A3 TO THE OTHER AND
* WHERE RESULT IS STORED.
*
BCDADD  MOVEM.L A3-A4,-(A7)
        MOVE.W  #4,CCR              ;RESET FLAGS
        ABCD    -(A4),-(A3)         ;ADD LOWEST BYTES (B0)
        ABCD    -(A4),-(A3)         ;ADD BYTES B1
        ABCD    -(A4),-(A3)         ;ADD BYTES B2
        ABCD    -(A4),-(A3)         ;ADD BYTES B3
        ABCD    -(A4),-(A3)         ;ADD BYTES B4
        ABCD    -(A4),-(A3)         ;ADD BYTES B5
        ABCD    -(A4),-(A3)         ;ADD BYTES B6
        ABCD    -(A4),-(A3)         ;ADD BYTES B7
        MOVEM.L (A7)+,A3-A4
        RTS
*------------------------------------------------------------
* THIS SUB CONVERTS DECIMAL AND HEX NUMBERS TO ASCII
* AND STORES THESE IN THE MSGBUF. THE MOST SIGNIFICANT
* CHARACTER IS STORED FIRST.NUMBER TO BE CONVERTED IS
* IN D0 ON ENTRY.
*
ASCII   MOVEM.L D0-D3,-(A7)
        MOVE.L  #7,D3               ;CHAR COUNT
LOOPY   MOVE.L  #3,D2               ;BIT COUNT
LOOPX   ASL.L   #1,D0               ;PUT CHAR IN D1,
        ROXL.B  #1,D1               ;STARTING WITH THE MSC
        DBF     D2,LOOPX
        AND.B   #$F,D1              ;CONVERT CHAR TO ASCII
        ADD.B   #$30,D1
        CMP.B   #$39,D1             ;IS CHAR A HEX?
        BLE     SAVE                ;NO.
        ADD.B   #7,D1               ;YES. THEN ADD 7
SAVE    MOVE.B  D1,(A6)+            ;SAVE ASCII IN BUFFER
        DBF     D3,LOOPY            ;FETCH NEXT CHAR
        MOVEM.L (A7)+,D0-D3
        RTS
*-----------------------------------------------------------------------
* THIS SUB CONVERTS THE FRACTIONAL PART OF THE FP
* NUMBER TO DECIMAL.NUMBER TO BE CONVERTED IS IN D3
* ON ENTRY ,RESULT IS IN D4 ON EXIT.
*
FRACTION        MOVEM.L D0-D3/A0-A4,-(A7)
        LEA     DBUFINI,A0
        LEA     (A0),A1
        LEA     RESULT,A3
        MOVE.L  #$0F,D0             ;BIT COUNT
        AND.L   #$7FFF,D3           ;MASK OFF HI BITS
        LSL.W   #1,D3               ;DENORMALIZE NUMBER
        LSR.W   D2,D3               ;(M*2**-E)
        SUB.L   D2,D2
        MOVE.L  D2,-4(A3)           ;SET RESULT TO ZERO
        MOVE.L  D2,-8(A3)
NXTBIT  BTST    D2,D3               ;IS BIT N=0?
        BEQ     BITZRO              ;YES, IGNORE
        MOVE.L  D2,D4               ;NO. ADD DECIMAL EQV TO
        LSL.W   #3,D4               ;RESULT
        LEA     8(A0),A4            ;(COMPUTE ADDR OF DEC NO.)
        ADD.W   D4,A4               ;
        BSR     BCDADD          ;BITZRO ADDQ.B  #1,D2           ;FETCH NEXT BIT
        DBF     D0,NXTBIT
        MOVE.L  -8(A3),D4       ;FETCH RESULT
        MOVEM.L (A7)+,D0-D3/A0-A4
        RTS
*
* THIS SUB CONVERTS THE INTEGER PART OF DENORMALIZED FP
* NUMBER TO DECIMAL. ON ENTRY NUMBER IS IN D5 AND ON EXIT
* RESULT IS IN D5.
*
INTEGER MOVEM.L D0-D4/A0-A1,-(A7)
        LEA     DBUFINT,A0
        LEA     (A0),A1
        MOVE.W  #$2710,(A1)+    ;STORE 10**4
        MOVE.W  #$3E8,(A1)+     ;STORE 10**3
        MOVE.W  #$64,(A1)+      ;STORE 10**2
        MOVE.W  #$0A,(A1)+      ;STORE 10**1
        MOVE.W  #01,(A1)+       ;STORE 10**0
        SUB.L   D4,D4
        MOVE.L  #4,D0           ;LOOP COUNT
LOOP    MOVE.W  (A0)+,D1        ;FETCH DIVISOR
        DIVU    D1,D5           ;NUMBER/DIVISOR
        MOVE.W  D5,D2           ;RETRIEVE QUOTIENT
        ASL.L   #4,D4           ;AND SAVE IN D4
        OR.B    D2,D4
        SWAP    D5              ;NUMBER=REMAINDER
        ANDI.L  #$FFFF,D5
        DBF     D0,LOOP         ;REPEAT WITH ANOTHER DIVISOR
        MOVE.L  D4,D5           ;PUT RESULT IN D5 & EXIT
        MOVEM.L (A7)+,D0-D4/A0-A1
        RTS
*
* THIS IS THE INITIALIZATION. IT COMPUTES AND SAVES THE
* DECIMAL EQUIVALENTS OF THE BITS OF THE MANTISSA
*
INITLZE MOVEM.L D0-D1/A0-A3,-(A7)
        LEA     DBUFINI,A0
        LEA     (A0),A1
        MOVE.W  #0,(A1)+        ;SAVE SMALLEST DEC EQV
        MOVE.W  #$1525,(A1)+
        MOVE.W  #$8789,(A1)+
        MOVE.W  #$0625,(A1)+
        MOVE.L  #14,D1          ;DIGIT COUNT
        LEA     (A1),A3         ;POINT
        LEA     -8(A1),A4
LOOPIY  MOVE.L  (A4)+,(A3)+     ;PUT IN THE NEXT DIGIT LOCTN
        MOVE.L  (A4)+,(A3)+     ;AND DOUBLE IT
        BSR     BCDADD          ;
        DBF     D1,LOOPIY
        MOVEM.L (A7)+,D0-D1/A0-A3
        RTS
*
* THIS SUBROUTINE OUTPUTS A MESSAGE REQUESTING NUMBER
* TO BE ENTERED.
*
VALUES  LEA     MSGBUF,A5
        LEA     (A5),A6
        BSR     SPACE           ;OUTPUT A FEW LINE FEEDS
        MOVE.W  #'EN',(A6)+     ;PUT MSG IN BUFFER
        MOVE.W  #'TE',(A6)+
        MOVE.W  #'R ',(A6)+
        MOVE.W  #' N',(A6)+
        MOVE.W  #'UM',(A6)+
        MOVE.W  #'BE',(A6)+
        MOVE.W  #'R ',(A6)+
        TRAP    #15             ;OUTPUT MESSAGE TO TERMINAL
        DC.W    $2
        LEA     (A5),A6         ;READ FP NUMBER FROM CONSOLE
        TRAP    #15
        DC.W    $1
        MOVE.B  #0,NTYPE        ;ESTABLISH TYPE OF CONVERSION
        MOVE.B  (A5)+,D0        ;AND SET FLAG ACCORDINGLY
        CMP.B   #'$',D0         ;IS IT FP-TO-DECIMAL?
        BNE     EXIT1           ;NO.MUST BE DECIMAL-TO-FP
        MOVE.B  #1,NTYPE        ;YES. THEN SET FLAG AND
EXIT1   RTS
*
* THIS SUBROUTINE FETCHES THE FP NUMBER FROM THE BUFFER
* ONE CHARACTER AT A TIME. (THE 3-BYTE FP NUMBER HAS SIX
* CHARACTERS).
*
FETCHFP MOVEM.L D0-D1/A5,-(A7)
        SUB.L   D3,D3
        MOVE.L  #5,D1           ;CHAR COUNT
LOOPFC  MOVE.B  (A5)+,D0        ;FECTH CHAR
        CMP.B   #$39,D0         ;IS CHAR A-F?
        BLE     HEX1            ;NO.MUST BE LESS THAN 10
        SUB.B   #7,D0           ;YES. CONVERT TO HEX
HEX1    AND.B   #$F,D0          ;MASK OFF HI BITS
        LSL.L   #4,D3           ;MAKE ROOM FOR CHAR IN REG
        OR.B    D0,D3           ;PUT CHAR IN REG
        DBF     D1,LOOPFC
        MOVE.L  D3,FPNUM        ;SAVE FP NUMBER
        MOVEM.L (A7)+,D0-D1/A5
        RTS
```

```
*THIS SUB CONVERTS DECIMAL NUMBER TO FLOATING PT.
* NUMBER IS EXPECTED IN THE FOLLOWING FORMAT:
* N=X/(10**E),WHERE E=0,1,..6; X IS IN HEX.
* E.G 163.567 IS EXPRESSED AS: X=163567,E=3.
* ON ENTRY,D0 CONTAINS THE SIGN OF X,D1 E AND D3 X.
* ON EXIT D3 CONTAINS THE FP EQUIVQLENT.
*
DTFP     LEA      DBUFINI,A0        ;SAVE THE POWERS OF 10
         MOVE.W   #$0041,(A0)+
         MOVE.W   #$4000,(A0)+
         MOVE.W   #$0044,(A0)+
         MOVE.W   #$5000,(A0)+
         MOVE.W   #$0047,(A0)+
         MOVE.W   #$6400,(A0)+
         MOVE.W   #$004A,(A0)+
         MOVE.W   #$7D00,(A0)+
         MOVE.W   #$004E,(A0)+
         MOVE.W   #$4E20,(A0)+
         MOVE.W   #$0051,(A0)+
         MOVE.W   #$61A8,(A0)+
         MOVE.W   #$0054,(A0)+
         MOVE.W   #$7A12,(A0)+
         MOVE.W   #$0059,(A0)+
         MOVE.W   #$4C4B,(A0)+
         LEA      DBUFINI,A0
         CMP.B    #0,D1
         BEQ      XEROD
         LSL.W    #2,D1
         MOVE.L   0(A0,D1.W),D2
         BSR      FDIV
ZEROD    TST.B    D0
         BEQ      DONE
         NEG.W    D3
DONE     RTS
*
* NORMALIZE X
*
NORMX    MOVEQ.L #79,D5
         MOVEQ.L #15,D4
         MOVEQ.L #15,D6
         TST.W    D3
         BEQ      ZEROX
         CMP.L    #32768,D3
         BGE      XGE
NORMX1   ADD.W    D3,D3
         DBMI     D4,NORMX1
         SUB.B    D4,D6
         LSR.W    #1,D3
         SUB.B    D6,D5
FORMATX  SWAP     D5
         MOVE.W   D3,D5
         MOVE.L   D5,D3
EXITX    RTS
ZEROX    CLR.L    D3
         BRA      EXITX
XGE      ADDQ.L   #1,D5
         LSR.L    #1,D3
         CMP.L    #32767,D3
         BGT      XGE
         BRA      FORMATX
         RTS
```

```
FETCHDE MOVEM.L D2/D4-D6,-(A7)
        SUB.L   D3,D3
        SUB.L   D5,D5
        SUB.L   D6,D6
        MOVE.W  BASE,D4
        MOVE.B  (A5)+,D0        ;DETERMINE SIGN OF NUMBER
        MOVE.B  #1,D1
        CMP.B   #'-',D0         ;IS IT NEGATIVE?
        BEQ     NXTCHAR         ;YES. LEAVE FLAG SET TO 1
        MOVE.B  #0,D1           ;NO.MUST BE POSITIVE.RESET FLAG.
        BRA     RADIX           ;AND GO SAVE.
NXTCHAR CMP.L   A5,A6           ;ALL CHAR FETCHED AND SAVED?
        BEQ     EXIT3           ;YES. THEN EXIT
        MOVE.B  (A5)+,D0        ;FETCH NXT CHAR
RADIX   CMP.B   #'.',D0         ;CHECK WHETHER IT IS A DECML PT
        BNE     NOTDOT          ;AND SET FLAG ACCORDINLY
        MOVE.B  #1,D6           ;SET DECIMAL PT FLAG
        BRA     NXTCHAR         ;
NOTDOT  TST.B   D6              ;IS DECML PT FLAG SET ?
        BEQ     DOTSET          ;NO. THEN STILL INTEGER
        ADDQ.B  #1,D5           ;INCR POWER OF DIVISOR
DOTSET  AND.L   #$F,D0          ;CONVERT NO. TO HEX
        MOVE.L  D3,D2
        SWAP    D3              ;WORK ON THE HIGHER WORD
        MULU    D4,D3
        SWAP    D3              ;
        TST.W   D3              ;CHECK FOR OVERFLOW
        BNE     EXIT3           ;VALUE IS TOO LARGE
        MULU    D4,D2           ;NOW WORK ON LO WORD
        ADD.L   D2,D3           ;PUT BOTH WORDS TOGETHER
        ADD.L   D0,D3           ;ADD THE NEXT NUMBER
        BVS     EXIT3           ;IF THERE IS OVERFLOW EXIT.
        BRA     NXTCHAR
EXIT3   MOVE.L  D1,D0           ;PUT RESULT IN APPROPRIATE REG
        MOVE.L  D5,D1
        MOVEM.L (A7)+,D2/D4-D6
        RTS
*
* THIS ROUTINE OUTPUTS THE RESULT OF DECIMAL-TO-FP CONVERSION
* TO THE DISPLAY.
*
OUTPUT1 LEA     MSGBUF,A5
        LEA     (A5),A6
        BSR     SPACE
        MOVE.W  #'DE',(A6)+
        MOVE.W  #'CI',(A6)+
        MOVE.W  #'MA',(A6)+
        MOVE.W  #'L-',(A6)+
        MOVE.W  #'TO',(A6)+
        MOVE.W  #'-F',(A6)+
        MOVE.W  #'P:',(A6)+
        MOVE.W  #$D0A,(A6)+     ;PUT LF AND CR
        MOVE.L  D3,D0
        BSR     ASCII
        TRAP    #15             ;OUTPUT MSG
        DC.W    $2
        BSR     SPACE
        TRAP    #15
        DC.W    0
*




*
```

```
*
* THIS PART OF THE ROUTINE PRINTS THE RESULT.
* THE NUMBER TO BE CONVERTED IS PRINTED ALONG WITH THE
* RESULT. HEX NUMBERS TO BE OUTPUT ARE FIRST
* CONVERTED TO ASCII.
*
OUTPUT2 LEA     MSGBUF,A5
        LEA     (A5) ,A6
        BSR     SPACE                   ;OUTPUT A FEW LINE FEEDS
        MOVE.W  #'FP',(A6)+             ;PUT MSG IN BUFFER
        MOVE.W  #'-T',(A6)+
        MOVE.W  #'O-',(A6)+
        MOVE.W  #'DE',(A6)+
        MOVE.W  #'CI',(A6)+
        MOVE.W  #'MA',(A6)+
        MOVE.W  #'L:',(A6)+
        MOVE.W  #$D0A,(A6)+             ;PUT CR & LF
        MOVE.L  FPNUM,D0                ;FETCH FP NUMBER
        BSR     ASCII                   ;CONVERT IT TO ASCII
        MOVE.B  ' =',(A6)+
        TST.B   D1                      ;PUT SIGN OF NUMBER
        BEQ     PLUS
        MOVE.B  #'-',(A6)+
        BRA     ANSWER
PLUS    MOVE.B  #'+',(A6)+
ANSWER  MOVE.L  D5,D0                   ;CONVERT INTGR TO ASCII
        BSR     ASCII
        MOVE.B  #'.',(A6)+
        MOVE.L  D4,D0                   ;CONVERT FRACTN TO ASCII
        BSR     ASCII
        TRAP    #15                     ;PRINT MSG,FPNUM & RESULT
        DC.W    2
        BSR     SPACE
        TRAP    #15                     ;EXIT TO VME BUG
        DC.W    0
*-------------------------------------------------------------
```

Appendix A11 FORTRAN and Minitab Programs for Off-line Removal of OA.

This appendix gives FORTRAN source codes and minitab macros used in the off-line analysis. Apart from a few instructions and graphics routines which are machine dependent (see later), the FORTRAN codes conform to the ANSI FORTRAN standard [1,2]. In these and other programs described in later appendices, the programs have been developed in stages, with the intermediate stages providing data for later stages. This approach was adopted to reduce the computing overheads. Of course, a command-input file can be easily written to combine the appropriate programs so that the final result is obtained in one run.

Several programs are listed here. Each main program has several dependent subprograms, most of which are located in the same file as the main program. Those not in the same file as the main program are referred to as external subprograms or subroutines, and have in most cases been developed as standard library of programs for general use. The purpose of each subprogram is given as comments just before its listing. The purpose of the main programs are described briefly here.

1. Program WFM.FTN

This program together with the subprogram RDATA is used to read data from a binary file, and written into a user-speci fied file. The resulting data file normally provides the input data for subsequent analysis.

2. Program MTBPLOT.FTN

This program together with the associated internal and external subprograms (NGRAPH, ACF and CCF; the last two are contained in a file called CORREL.FTN) is used to read six columns of data from user-specified data file, plot the data directly, or their ACfs , or the crosscorrelation functions between the data columns. Upto 8 data files may be obtained.

3. Program ONECHN

This program is used to study one parameter models. The program together with the external subroutines RDATA, NGRAPH and ACF reads data from a binary file, corrects the EEG for artefact using the four one-parameter models described in chapter 2, and plots the corrected EEG and their ACFs.

4. Program MTBDATA.FTN

This program is similar to WFM.FTN. The main difference between them is that it generates also the data required for model 4E (appendix A3); this program was used in the preliminary investigation of models.

MINITAB Programs

1. Program OLS.MTB

This program together with the associated macro ESTMATE.MTB is used to study the models defined in section 2.2 using the OLS method. The program uses the statistical package called MINITAB, which allowed extensive investigation to be carried out relatively easily. To use the program, the data to be analized is first written into a file 'MTB.DATA' and then the program is run.

2. Program DIFF.MTB

This program is the same as program OLS.MTB, the only difference being that the data is differenced prior to processing.

3. Program OADIFF.MTB

This program is used to analyze the data. It allows OLS method to be performed and computes the corrected EEGs and the OA estimates as well as the differences between the OAs. The processed data are written into files which serve as inputs to further stages of analysis, e.g plotting of the data.

4. Program OADIFF1.MTB

This program is similar to OADIFF.MTB. However, whereas OADIFF.MTB allows OLS method to be performed on several models OADIFF1.MTB allows OLS to be performed on six models. These were models that were used in the preliminary analysis of models.

Index to programs

References for Appendix A11

[1] ANSI subcommitte X3J3
Clarification of FORTRAN standards.-Second Report.
COMM. ACM., 14, 1971, pp 628-642.

[2] ASA Committee X3 FORTRAN VS BASIC FORTRAN.
Comm. ACM., 7, 1964 , pp591-625.

Notes

The main differences between the FORTRAN codes given below and the standards above are in input-output codes. E.G the FORTRAN given here uses OPEN$A and CLOSE$A which are commands specific for the PRIME computers. Also the graphics routines are system dependent.

PROGRAM WFM.FTN

```
C
C THIS PROGRAM, TOGETHER WITH THE SUBPROGRAM 'RDATA'
C IS USED TO READ DATA FROM A BINARY FILE AND WRITE
C TO A DATA FILE FOR FURTHER PROCESSING.
C
      COMMON VL(1024),VR(1024),HL(1024),HR(1024)
      INTEGER*2 FNAME(20)
      COMMON /VDATA/EEG1(1024),EEG2(1024),FNAME
      COMMON /VEEGF/EEGF(1024)
      DATA L2/0/
      NPT=1024
      N=1024
      IPLOT=1
      WRITE(1,10)
10    FORMAT('ENTER BATNO')
      READ(1,*)L1
      L4=0
      CALL RDATA(L1,L2,L3,L4)
C
C READ DATA AND REMOVE MEAN
C
      CALL MEAN(N,VL,VLM)
      CALL MEAN(N,VR,VRM)
      CALL MEAN(N,HL,HLM)
      CALL MEAN(N,HR,HRM)
      CALL MEAN(N,EEG1,EEGM)
      WRITE(6,40)(VL(I),VR(I),HL(I),HR(I),EEG1(I),I=1,1024)
   40 FORMAT(5(F10.5,2X))
      CALL EXIT
      END
C
C------------------------------------------------
C
      SUBROUTINE MEAN(NPTS,DATA,RMEAN)
      DIMENSION DATA(NPTS)
      RMEAN=0.0
      DO 20 I=1,NPTS
20    RMEAN=RMEAN+DATA(I)
      RMEAN=RMEAN/FLOAT(NPTS)
      DO 30 I=1,NPTS
30    DATA(I)=DATA(I)-RMEAN
      RETURN
      END
```

PROGRAM MTBPLOT.FTN

```
C
C THIS PROGRAM (AND THE INTERNAL SUBPROGRAMS) IS USED
C TO READ SIX COLUMNS OF DATA, PLOT THE DATA
C DIRECTLY OR THEIR ACFS OR THE CROSS CORRELATION
C FUNCTIONS BETWEEN THE DATA COLUMNS.
C
      COMMON /XVAL/X1(1024),X2(1025)
      COMMON /MTDATA/ C0(1024),C1(1024),C2(1024),C3(1024)
     +,C4(1024),C5(1024)
      INTEGER FNAME(20)
      N=1024
      FREQ=125.0
      NCORL=513
      DO 45 I=1,N
      IM1=I-1
      I2=I-NCORL
      X1(I)=FLOAT(IM1)/FREQ
45    X2(I)=FLOAT(I2)/FREQ
      X2(1025)=512.0/FREQ
      CALL DEVICE(IDEV)
      WRITE(1,5)
5     FORMAT('ENTER NUMBER OF FILES TO PROCESS')
      READ(1,*)NFILES
      DO 50 NTIMES=1,NFILES
      CALL GETNAM(FNAME)
      CALL MTBRD(FNAME)
C
C SELECT HOW DATA SHOULD BE PLOTTED
C
      WRITE(1,10)
10    FORMAT('ENTER 0 FOR TIME 1 FOR ACF 2 FOR CCF')
      READ(1,*)IST
      IF(IST .EQ. 0)CALL AMPL(C0,C1,C2,C3,C4,C5,IST,N)
      IF(IST .EQ. 1)CALL ACFN(C0,C1,C2,C3,C4,C5,IST,N)
      IF(IST .EQ. 2)CALL CCFN(C0,C1,C2,C3,C4,C5,IST,N)
50    CONTINUE
      CALL DEVEND
      CALL EXIT
      END
C
C-----------------------------------------------------
C
      SUBROUTINE AMPL(C0,C1,C2,C3,C4,C5,IST,N)
C
C THIS SUBPROGRAM TOGETHER WITH SUBPROGRAM NGRAPH
C IS USED TO PLOT DATA DIRECTLY
C
      DIMENSION C0(1024),C1(1024),C3(1024),C4(1024)
     +,C5(1024)
      COMMON /XVAL/X1(1024),X2(1025)
      NCORL=513
      CALL NGRAPH(N,X1,C0,0,IST)
      CALL NGRAPH(N,X1,C1,1,IST)
      CALL NGRAPH(N,X1,C2,2,IST)
      CALL NGRAPH(N,X1,C3,3,IST)
      CALL NGRAPH(N,X1,C4,4,IST)
      CALL NGRAPH(N,X1,C5,5,IST)
      RETURN
      END
C
C-----------------------------------------------
C
      SUBROUTINE ACFN(C0,C1,C2,C3,C4,C5,IST,N)
C
C THIS SUBPROGRAM TOGETHER WITH SUBPROGRAMS ACF
C AND NGRAPH ARE USED TO COMPUTE THE ACFS OF THE DATA,
C AND TO PLOT THESE
C
      DIMENSION C0(1024),C1(1024),C2(1024),C3(1024)
     +,C4(1024),C5(1024)
      DIMENSION ACFC0(1024),ACFC1(1024),ACFC2(1024)
     +,ACFC3(1024),ACFC4(1024),ACFC5(1024)
      COMMON /XVAL/X1(1024),X2(1025)
      NCORL=1024
      N=512
      CALL ACF(NCORL,C0,ACFC0)
      CALL ACF(NCORL,C1,ACFC1)
      CALL ACF(NCORL,C2,ACFC2)
      CALL ACF(NCORL,C3,ACFC3)
      CALL ACF(NCORL,C4,ACFC4)
      CALL ACF(NCORL,C5,ACFC5)
      CALL NGRAPH(N,X1,ACFC0,0,IST)
      CALL NGRAPH(N,X1,ACFC1,1,IST)
      CALL NGRAPH(N,X1,ACFC2,2,IST)
      CALL NGRAPH(N,X1,ACFC3,3,IST)
      CALL NGRAPH(N,X1,ACFC4,4,IST)
      CALL NGRAPH(N,X1,ACFC5,5,IST)
      RETURN
      END
C
C---------------------------------------------------
C
      SUBROUTINE CCFN(C0,C1,C2,C3,C4,C5,IST,N)
C
C THIS SUBPROGRAM AND SUBPROGRAMS XCF AND NGRAPH
C ARE USED TO COMPUTE THE CCF BETWEEN THE DATA
C COLUMNS, AND TO PLOT THESE.
C
      DIMENSION C0(1024),C1(1024),C2(1024),C3(1024)
     +,C4(1024),C5(1024)
      DIMENSION XCFC0(1025),XCFC1(1025),XCFC2(1025)
     +,XCFC3(1025),XCFC4(1025),XCFC5(1025)
      COMMON /XVAL/X1(1024),X2(1025)
      NCORL=513
      CALL XCF(NCORL,C0,C1,XCFC0)
      CALL XCF(NCORL,C0,C2,XCFC1)
      CALL XCF(NCORL,C0,C3,XCFC2)
      CALL XCF(NCORL,C1,C2,XCFC3)
      CALL XCF(NCORL,C1,C3,XCFC4)
      CALL XCF(NCORL,C2,C3,XCFC5)
      CALL NGRAPH(N,X2,XCFC0,0,IST)
      CALL NGRAPH(N,X2,XCFC1,1,IST)
      CALL NGRAPH(N,X2,XCFC2,2,IST)
      CALL NGRAPH(N,X2,XCFC3,3,IST)
      CALL NGRAPH(N,X2,XCFC4,4,IST)
      CALL NGRAPH(N,X2,XCFC5,5,IST)
      WRITE(6,10)(XCFC0(I),XCFC1(I),XCFC2(I)
     +,XCFC3(I),XCFC4(I),XCFC5(I),I=513,523)
10    FORMAT(6(F10.5,2X))
      RETURN
      END
C
C---------------------------------------------
C
      SUBROUTINE GETNAM(FNAME)
C
C THIS SUBPROGRAM REQUESTS FOR AND READS FROM
C THE TERMINAL THE NAME OF FILE CONTAINING THE DATA
C
      INTEGER*2 FNAME(20),TBUFF(20)
      DATA TBUFF /'NONAMEGIVEN                              '/
      WRITE(1,8)
8     FORMAT('GIVE NAME OF FILE TO BE PROCESSED')
      READ(1,10)FNAME
10    FORMAT(20A2)
      IF(FNAME(1) .EQ. '  ')GO TO 100
      DO 20 I=1,20
20    TBUFF(I)=FNAME(I)
      RETURN
100   DO 120 I=1,20
120   FNAME(I)=TBUFF(I)
      RETURN
      END
C
C---------------------------------------------------
C
      SUBROUTINE MTBRD(FNAME)
C
C THIS SUBPROGRAM READS THE DATA FROM THE FILE
C GIVEN IN SUBPROGRAM GETNAM
C
$INSERT SYSCOM|A$KEYS
      COMMON /MTDATA/C0(1024),C1(1024),C2(1024),C3(1024)
     +,C4(1024),C5(1024)
      INTEGER*2 FNAME(20),RKEY
      LOGICAL OPEN,CLOSE
      RKEY=3
      NLEN=40
      OPEN=OPEN$A(A$READ,FNAME,NLEN,RKEY)
      IF(.NOT. OPEN)GO TO 1700
      READ(7,*)(C0(I),C1(I),C2(I),C3(I),C4(I),C5(I),I=1,1024)
      CLOSE=CLOS$A(RKEY)
      IF(CLOSE)RETURN
      WRITE(1,30)FNAME
30    FORMAT('*** CANT CLOSE FILE',20A2,'***')
      STOP 1
1700  WRITE(1,35)FNAME
35    FORMAT('*** CANT OPEN FILE ',20A2,'***')
      STOP 2
      END
```

PROGRAM ONECHN.FTN

```
C
C THIS PROGRAM AND THE ASSOCIATED SUBPROGRAMS
C ARE USED TO STUDY ONE PARAMETER MODELS
C
      DIMENSION EEGCVL(1024),EEGCVR(1024),EEGCHL(1024)
      DIMENSION EEGCHR(1024)
      INTEGER*2 FNAME
      COMMON VL(1024),VR(1024),HL(1024),HR(1024)
      COMMON /VDATA/EEG1(1024),EEG2(1024),FNAME(20)
      COMMON /XVAL/X(1024)
      DATA L2/0/
      NPT=1024
      N=1024
      DO 45 I=1,1024
      IM1=I-1
45    X(I)=FLOAT(IM1)/125.0
      CALL DEVICE(IDEV)
      WRITE(1,2)
2     FORMAT('ENTER NO OF RECORDS TO BE PROCESSED')
      READ(1,*)NTIMES
      DO 5 M=1,NTIMES
      WRITE(1,10)
10    FORMAT('ENTER BATNO')
      READ(1,*)L1
      L4=0
C
C READ DATA AND REMOVE THE MEAN FROM EACH DATA COLUMN
C
      CALL RDATA(L1,L2,L3,L4)
      CALL MEAN(N,VL,VLM)
      CALL MEAN(N,VR,VRM)
      CALL MEAN(N,HL,HLM)
      CALL MEAN(N,HR,HRM)
      CALL MEAN(N,EEG1,EEGM)
C
C REMOVE OA FROM EEG USING EACH OF THE FOUR
C SINGLE PARAMETER MODELS IN TURN, COMPUTE
C THE ACFS OF TH EEGCS, AND PLOT BOTH EEGCS AND ACFS
C
      CALL ONECHN(VL,EEGCVL,N)
      CALL ONECHN(VR,EEGCVR,N)
      CALL ONECHN(HL,EEGCHL,N)
      CALL ONECHN(HR,EEGCHR,N)
      CALL VMPLOT(EEGCVL,EEGCVR,EEGCHL,EEGCHR)
5     CONTINUE
      CALL DEVEND
      CALL EXIT
      END
C
C--------------------------------------------------
C
      SUBROUTINE MEAN(NPTS,DATA,RMEAN)
C
C SUBROUTINE USED TO COMPUTE AND REMOVE MEAN FROM
C AN ARRAY. THE MEAN-CORRECTED DATA OVERWRITES THE
C OLD DATA.
C
      DIMENSION DATA(NPTS)
      RMEAN=0.0
      DO 20 I=1,NPTS
20    RMEAN=RMEAN+DATA(I)
      RMEAN=RMEAN/FLOAT(NPTS)
      DO 30 I=1,NPTS
30    DATA(I)=DATA(I)-RMEAN
      RETURN
      END
C
C--------------------------------------------------
C
      SUBROUTINE ONECHN(X,DOUT,N)
      DIMENSION X(1024),DOUT(1024)
      INTEGER*2 FNAME
      COMMON /VDATA/EEG1(1024),EEG2(1024),FNAME(20)
      A=0.0
      B=0.0
      D=0.0
C
C     OBTAIN THE COVARIANCE OF THE MEASURED DATA AND THE
C     PARAMETER ESTIMATE, AK.
C
      DO 10 I=1,N
      A=A+EEG1(I)*X(I)
10    B=B+X(I)*X(I)
      AK=A/B
```

```
C
C       REMOVE EMA ESTIMATES FROM THE EEG
C
      DO 20 I=1,N
20    DOUT(I)=EEG1(I)-X(I)*AK
      RETURN
      END
C
C------------------------------------------------------
C
      SUBROUTINE VMPLOT(D1,D2,D3,D4)
C
C THIS SUBPROGRAM, TOGETHER WITH SUBPROGRAM ACF,
C ALLOWS THE ACFS OF THE DATA IN ARRAYS D1,D2,D3,D4
C TO BE SEPARATELY COMPUTED.
C
      DIMENSION D1(1024),D2(1024),D3(1024),D4(1024)
      DIMENSION D5(1024),D6(1024)
      DIMENSION ACF1(1024),ACF2(1024),ACF3(1024)
      DIMENSION ACF4(1024)
      NCORL=1024
      CALL ACF(NCORL,D1,ACF1)
      CALL ACF(NCORL,D2,ACF2)
      CALL ACF(NCORL,D3,ACF3)
      CALL ACF(NCORL,D4,ACF4)
      CALL RGRAPH(D1,D2,D3,D4,0)
      CALL RGRAPH(ACF1,ACF2,ACF3,ACF4,1)
      RETURN
      END
C
C------------------------------------------------------------
C
      SUBROUTINE RGRAPH(D1,D2,D3,D4,IST)
C
C THIS SUBPROGRAM TOGETHER WITH NGRAPH ALLOWS FOUR
C GRAPHS TO BE PLOTTED ON ONE PAGE.
C
      DIMENSION D1(1024),D2(1024),D3(1024),D4(1024),D5(1024)
      DIMENSION D6(1024)
      COMMON /XVAL/X(1024)
      N=1024
      CALL PICCLE
      CALL NGRAPH(N,X,D1,0,IST)
      CALL NGRAPH(N,X,D2,1,IST)
      CALL NGRAPH(N,X,D3,2,IST)
      CALL NGRAPH(N,X,D4,3,IST)
      RETURN
      END
```

PROGRAM MTBDATA.FTN

```
C
C THIS PROGRAM, TOGETHER WITH THE SUBPROGRAM 'RDATA'
C IS USED TO READ DATA FROM A BINARY FILE AND WRITE
C TO A DATA FILE FOR FURTHER PROCESSING.
C
      DIMENSION TEMP(1024)
      COMMON VL(1024),VR(1024),HL(1024),HR(1024)
      INTEGER*2 FNAME(20)
      COMMON /VDATA/EEG1(1024),EEG2(1024),FNAME
      COMMON /VEEGF/EEGF(1024)
      DATA L2/0/
      NPT=1024
      N=1024
      IPLOT=1
      WRITE(1,10)
10    FORMAT('ENTER BATNO')
      READ(1,*)L1
      L4=0
      CALL RDATA(L1,L2,L3,L4)
C
C READ DATA AND REMOVE MEAN
C
      CALL MEAN(N,VL,VLM)
      CALL MEAN(N,VR,VRM)
      CALL MEAN(N,HL,HLM)
      CALL MEAN(N,HR,HRM)
      CALL MEAN(N,EEG1,EEGM)
C
C OBTAIN THE DELAY TERM FOR MODEL 4E
C
      DO 50 I=2,1023
50    TEMP(I)=VL(I-1)+VL(I+1)
      TEMP(1)=VL(2)
      TEMP(1024)=VL(1023)
C
C WRITE ALL DATA TO A FILE
C
      WRITE(6,40)(VL(I),VR(I),HL(I),HR(I)
     +,EEG1(I),TEMP(I),I=1,1024)
   40 FORMAT(6(F10.5,2X))
      CALL EXIT
      END
C
C--------------------------------------------
C
      SUBROUTINE MEAN(NPTS,DATA,RMEAN)
      DIMENSION DATA(NPTS)
      RMEAN=0.0
      DO 20 I=1,NPTS
20    RMEAN=RMEAN+DATA(I)
      RMEAN=RMEAN/FLOAT(NPTS)
      DO 30 I=1,NPTS
30    DATA(I)=DATA(I)-RMEAN
      RETURN
      END
      SUBROUTINE RDATA(L1,L2,L3,L4)
C
C THIS SUBROUTINE READS DATA FROM A GIVEN BINARY FILE AND
C CONVERTS IT TO REAL FORMAT
C
      REAL MEEG1,MEEG2
      COMMON VL(1024),VR(1024),HL(1024),HR(1024)
      INTEGER*2 FNAME(20)
      INTEGER BATNO,INP(1024)
      COMMON /VDATA/EEG1(1024),EEG2(1024),FNAME
      LOGICAL CSE
      DATA N,CSE/1024,.FALSE./
      BATNO=L1
      IF(L4 .GT. 0)GO TO 25
      CALL GETNAM(FNAME)
      WRITE(1,8)FNAME
8     FORMAT('DATA FILE ',20A2,///)
C CHECK WHETHER THE WHOLE FILE IS TO BE READ
      IF(L2 .EQ. 0)GO TO 25
10    WRITE(1,15)
15    FORMAT('ENTER BATCH NO. OR (-1) TO QUIT')
      READ(1,*)BATNO
      IF(BATNO .LT. 0 .OR. BATNO .GT. 191)GO TO 299
25    WRITE(1,20)BATNO
20    FORMAT('BATNO=',I5)
      L=BATNO
C               READ THE DATA AND CONVERT TO REAL FORMAT.
      CALL DATIN(L,INP,FNAME,SF1,SF2,SAMRAT,CSE)
      DO 30 I=1,N
   30 VL(I)=FLOAT(INP(I))*SF1
      L=L+1
      CALL DATIN(L,INP,FNAME,SF1,SF2,SAMRAT,CSE)
      DO 40 I=1,N
   40 VR(I)=FLOAT(INP(I))*SF1
      L=L+1
      CALL DATIN(L,INP,FNAME,SF1,SF2,SAMRAT,CSE)
      DO 50 I=1,N
   50 HL(I)=FLOAT(INP(I))*SF1
      L=L+1
      CALL DATIN(L,INP,FNAME,SF1,SF2,SAMRAT,CSE)
      DO 60 I=1,N
   60 HR(I)=FLOAT(INP(I))*SF1
      L=L+1
      CALL DATIN(L,INP,FNAME,SF1,SF2,SAMRAT,CSE)
      DO 70 I=1,N
   70 EEG1(I)=FLOAT(INP(I))*SF2
      L=L+1
      CALL DATIN(L,INP,FNAME,SF1,SF2,SAMRAT,CSE)
      DO 80 I=1,N
   80 EEG2(I)=FLOAT(INP(I))*SF2
      WRITE(1,90)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,CSE)
      RETURN
299   L3=0
      IF(BATNO .EQ. -1)GO TO 240
      WRITE(1,300)
300   FORMAT('ILLEGAL BATCH NO.-NO SHOULD BE BETWEEN (0) AND (191)')
      GO TO 10
      END
C
C---------------------------------------------
C
      SUBROUTINE GETNAM(NAME)
      INTEGER*2 NAME(20),TBUFF(20)
      DATA TBUFF /'NONAMEGIVEN                             '/
6     WRITE(1,8)
8     FORMAT('GIVE NAME OF FILE TO BE PROCESSED')
      READ(1,10)NAME
10    FORMAT(20A2)
      IF(NAME(1) .EQ. '  ')GO TO 100
      DO 20 I=1,20
20    TBUFF(I)=NAME(I)
      RETURN
100   DO 120 I=1,20
120   NAME(I)=TBUFF(I)
      RETURN
      END
C
C---------------------------------------------
C
```

```
C
      SUBROUTINE DATIN(IBATNO,IDATA,RNAME,SF1,SF2,SAMRAT,CSE)
$INSERT SYSCOM|A$KEYS
      INTEGER*2 IDATA(1024),ONAME(20),RNAME(20),TITLE(36),NME(6)
      INTEGER*2 RWKEY,NLEN,NLEN2,PRIMNO,NBAT,IBATNO,IAA1(4)
      LOGICAL OPEN,NEOPEN,CSE
      DATA NEOPEN /.FALSE./
      RWKEY=1
      PRIMNO=1
      NLEN=40
      NLEN2=NLEN/2
      IF(NEOPEN)GO TO 1300
      IF(CSE)RETURN
  900 OPEN=OPEN$A(RWKEY,RNAME,NLEN,PRIMNO)
      IF(.NOT. OPEN)GO TO 1700
      DO 1000 IC=1,NLEN2
 1000 ONAME(IC)=RNAME(IC)
      NEOPEN=.TRUE.
      READ(5,3000,END=1800,ERR=1900)NME,SF1,SF2
      READ(5,3010,END=1800,ERR=1900)MAXBAT
      READ(5,3020,END=1800,ERR=1900)TITLE
      DO 234 IL=1,4
  234 IAA1(IL)=TITLE(IL+20)
      DECODE (8,236,IAA1)SAMRAT
  236 FORMAT(F8.4)
 1100 READ(5,END=1800,ERR=1900)NBAT
      READ(5,END=1800,ERR=1900)(IDATA(I),I=1,1024)
      IF(NBAT .NE. IBATNO)GO TO 1100
      NBAT=NBAT+1
      RETURN
 1200 OPEN=CLOS$A(PRIMNO)
      IF(.NOT. OPEN)GO TO 1600
      NEOPEN=.FALSE.
      RETURN
 1300 IF(CSE)GO TO 1200
      DO 1400 IC=1,NLEN2
 1400 IF(ONAME(IC) .NE. RNAME(IC))GO TO 1500
      IF(IBATNO .GE. NBAT)GO TO 1100
 1500 OPEN=CLOS$A(PRIMNO)
      IF(OPEN)GO TO 900
 1600 WRITE(1,1610)ONAME
 1610 FORMAT('*** CANT CLOSE FILE ',20A2,' ***')
      STOP 1
 1700 WRITE(1,1710)RNAME
 1710 FORMAT('*** CANT OPEN FILE ',20A2,' ***')
      STOP 2
 1800 WRITE(1,1810)RNAME
 1810 FORMAT('*** END OF FILE ',20A2,' ***')
      STOP 3
 1900 WRITE(1,1910)IBATNO,RNAME
 1910 FORMAT('*** ERROR TRYING TO READ BATCH ',I5,' FROM FILE ',20A2,' ***
     +')
      STOP 4
 3000 FORMAT(6A2,2F8.6)
 3010 FORMAT(I4)
 3020 FORMAT(36A2)
      END
```

SUBPROGRAM NGRAPH.FTN

```
C
C ROUTINE TO PLOT A GRAPH ON 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
      SUBROUTINE NGRAPH(N,X,Y,ING,IST)
      DIMENSION X(N),Y(N)
      INTEGER*2 ILAB1(7),ILAB2(2),ILAB3(2)
      DATA ILAB1,ILAB2,ILAB3/'amplitude (uV)','acf ','ccf '/
      X1=0.
      IF(MOD(ING,2) .EQ. 1)X1=110.
      Y1=57.
      IF(ING .LT. 2)Y1=115.
      IF(ING .GT. 3)Y1=0.
      X2=X1+100.
      Y2=Y1+70.
      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+20.0
      CALL MOVTO2(AX1,AY1)
      CALL CHAHOL('time (sec)*.')
      CALL CHAANG(90.)
      CALL MOVTO2(AX2,AY2)
      IF(IST .EQ. 0)CALL CHAARR(ILAB1,7,2)
      IF(IST .EQ. 1)CALL CHAARR(ILAB2,2,2)
      IF(IST .EQ. 2)CALL CHAARR(ILAB3,2,2)
      CALL CHAANG(0.)
      RETURN
      END
C
C-----------------------------------------------------
C
      SUBROUTINE DEVICE(IDEV)
C
C THIS SUBROUTINE NOMINATES THE GRAPHICS DEVICE TO
C BE USED FOR PLOTTING GRAPHS.
C
      WRITE(1,10)
10    FORMAT('ENTER 1 TOI VIEW 2 FOR CC906 3 FOR CC81')
      READ(1,*)IDEV
      IF(IDEV .EQ. 1)CALL T4010
      IF(IDEV .EQ. 2)CALL CC906
      IF(IDEV .EQ. 3)CALL CC81
      RETURN
      END
```

```
             SUBPROGRAM CORREL.FTN
C
C     THIS PROGRAM CONTAINS THREE SUBPROGRAMS:
C        ACF - COMPUTES A 513 POINT ACF FROM A 1024-POINT DATA
C              (FOR POSITIVE LAGS ONLY)
C        SMEAN - TAKES OFF THE D.C OR MEAN VALUE FROM THE DATA
C                BEFORE ACF OR XCF IS CALCULATED
C        XCF - COMPUTES A 1025-POINT XCF FROM TWO 1024-POINT
C              DATA (513-POINT XCF IS CALCULATED FOR BOTH
C                    POSITIVE AND NEGATIVE LAGS)
C
      SUBROUTINE ACF(NCORL,DATA,AUTOCF)
C THIS SUBROUTINE COMPUTES AND RETURNS TO THE MAIN PROGRAM
C THE AUTOCORRELATION FUNCTION OF AN NPOINT-DATA
      DIMENSION DATA(1024),AUTOCF(NCORL)
      DOUBLE PRECISION STORE
      INTEGER*2 Z1
      COMMON /ICOV/ICV0
      NPOINT=1024
      CALL SMEAN(NPOINT,DATA,AMEAN)
      DO 25 I=1,NCORL
      II=I-1
      STORE=0.
      Z1=NPOINT-II
      DO 20 J=1,Z1
   20 STORE=STORE+DATA(J)*DATA(J+II)
   25 AUTOCF(I)=STORE
      IF(ICV0 .EQ.  1)RETURN
      STORE=ABS(AUTOCF(1))
      DO 30 I=1,NCORL
   30 AUTOCF(I)=AUTOCF(I)/STORE
      RETURN
      END
C
C--------------------------------------------------
C
      SUBROUTINE SMEAN(NPTS,DATA,RMEAN)
C THIS SUBROUTINE COMPUTES AND REMOVES THE MEAN
C OR DC VALUE FROM AN NPOINT-DATA
      DIMENSION DATA(NPTS)
      DOUBLE PRECISION RMEAN
      RMEAN=0.
      DO 20 I=1,NPTS
   20 RMEAN=RMEAN+DATA(I)
      RMEAN=RMEAN/FLOAT(NPTS)
      DO 30 I=1,NPTS
   30 DATA(I)=DATA(I)-RMEAN
      RETURN
      END
C
C--------------------------------------------------
C
      SUBROUTINE XCF(NCORL,DATA1,DATA2,CCF)
C THIS SUBROUTINE COMPUTES THE CROSS CORRELATION FUNCTION
C OF TWO NPOINT DATA
      DIMENSION DATA1(1024),DATA2(1024)
      DIMENSION CCF(1025)
      DOUBLE PRECISION SUM1,SUM2,A,B,BIG
      COMMON /ICOV/ICV0
      NPOINT=1024
      NCORL1=NCORL+1
      NPT=NPOINT+1
      CALL SMEAN(NPOINT,DATA1,AMEAN1)
      CALL SMEAN(NPOINT,DATA2,AMEAN2)
      A=0.
      B=0.
      DO 20 I=1,NCORL
      I1=I-1
      I2=I+NCORL-1
      I3=NCORL1-I
      SUM1=0.
      SUM2=0.
      J=NPOINT-I1
      DO 10 K=1,J
      SUM1=SUM1+DATA1(K)*DATA2(K+I1)
10    SUM2=SUM2+DATA2(K)*DATA1(K+I1)
      CCF(I2)=SUM1
20    CCF(I3)=SUM2
      IF(ICV0 .EQ. 1)GO TO 40
      DO 30 L=1,NPOINT
      A=A+DATA1(L)*DATA1(L)
      B=B+DATA2(L)*DATA2(L)
30    CONTINUE
      BIG=SQRT(A*B)
      DO 25 I=1,NPT
25    CCF(I)=CCF(I)/BIG
      RETURN
40    DO 50 I=1,NPT
50    CCF(I)=CCF(I)/NPOINT
      RETURN
      END
```

ROGRAM OLS.MTB

```
#PROGRAM USED TO ANALYZE THE PERFORMANCE OF
#OF MODELS
#
#READ DATA AND REMOVE THE MEAN VALUES
#
READ 'MTB.DATA' C1-C5
MEAN C1 K1
MEAN C2 K2
MEAN C3 K3
MEAN C4 K4
MEAN C5 K5
SUBTRACT K1 C1 C1
SUBTRACT K2 C2 C2
SUBTRACT K3 C3 C3
SUBTRACT K4 C4 C4
SUBTRACT K5 C5 C5
NAME C1='VL',C2='VR',C3='HL',C4='HR',C5='EEG'
NAME C6='VLD',C7='VRD',C8='HLD',C9='HRD',C10='EEGD'
NAME C80='K1',C81='K2',C82='K3',C83='K4'
NAME C84='TK1',C85='TK2',C86='TK3',C87='TK4'
NAME C88='VAR(E)',C89='RKSQ',C90='CK',C91='FK'
NAME C92='MODEL'
LET C11='HL'-'HR'
LET C12='HL'*'HR'
LET C13='VL'*'VL'
LET C14='VL'-'VR'
#
#PERFORM OLS ANALYSIS ON DATA USING THE MACRO 'ESTIMATE.MTB'
#
EXEC 'ESTIMATE.MTB' 1
STOP
```

PROGRAM DIFF.MTB

```
#
#PROGRAM USED TO ANALYZE THE PERFORMANCE OF MODELS
# FOR THE OLS WITH DIFFERENCED DATA METHOD
#
READ 'MTB.DATA' C1-C5
MEAN C1 K1
MEAN C2 K2
MEAN C3 K3
MEAN C4 K4
MEAN C5 K5
SUBTRACT K1 C1 C1
SUBTRACT K2 C2 C2
SUBTRACT K3 C3 C3
SUBTRACT K4 C4 C4
SUBTRACT K5 C5 C5
NAME C6='VL',C7='VR',C8='HL',C9='HR',C10='EEG'
NAME C80='K1',C81='K2',C82='K3',C83='K4'
NAME C84='TK1',C85='TK2',C86='TK3',C87='TK4'
NAME C88='VAR(E)',C89='RKSQ',C90='CK',C91='FK'
NAME C92='MODEL'
#
#DIFFERENCE THE DATA
#
DIFF C1,'VL'
DIFF C2,'VR'
DIFF C3,'HL'
DIFF C4,'HR'
DIFF C5,'EEG'
LET C11='HL'-'HR'
LET C12='HL'*'HR'
LET C13='VL'*'VL'
LET C14='VL'-'VR'
#
#PERFORM OLS ANALYSIS ON THE DIFFERENCED DATA
# USING THE MACRO 'ESTIMATE.MTB'
#
#
EXEC 'ESTIMATE.MTB' 1
STOP
```

PROGRAM OADIFF.MTB

```
#
#PROGRAM USED TO PERFORM PICTORIAL COMPARISON
#OF MODELS.
#
#READ DATA AND REMOVE THE MEAN VALUES.
#
READ 'OADIFF.DATA' C1-C5
MEAN C1 K1
MEAN C2 K2
MEAN C3 K3
MEAN C4 K4
MEAN C5 K5
SUBTRACT K1 C1 C1
SUBTRACT K2 C2 C2
SUBTRACT K3 C3 C3
SUBTRACT K4 C4 C4
SUBTRACT K5 C5 C5
NAME C1='VL',C2='VR',C3='HL',C4='HR',C5='EEG'
LET C11='HL'-'HR'
LET C12='HL'*'HR'
LET C13='VL'*'VL'
LET C14='VL'-'VR'
#
#PERFORM OLS ANALYSIS ON THE DATA USING THE
# VARIOUS MODELS
#
BRIEF 2
REGRESS 'EEG' 1 'VL' C19;
NOCONSTANT;
COEFF C21;
XPXINV M1.
LET C41=C21(1)*'VL'
BRIEF 2
REGRESS 'EEG' 1 'VR' C19;
NOCONSTANT;
COEFF C22;
XPXINV M1.
LET C42=C22(1)*'VR'
BRIEF 2
REGRESS 'EEG' 1 'HL' C19;
NOCONSTANT;
COEFF C23;
XPXINV M1.
LET C43=C23(1)*'HL'
BRIEF 2
REGRESS 'EEG' 1 'HR' C19;
NOCONSTANT;
COEFF C24;
XPXINV M1.
LET C44=C24(1)*'HR'
BRIEF 2
REGRESS 'EEG' 2 'VL','HL' C19;
NOCONSTANT;
COEFF C25;
XPXINV M1.
LET C45=C25(1)*'VL'+C25(2)*'HL'
BRIEF 2
REGRESS 'EEG' 2 'VL','HR' C19;
NOCONSTANT;
COEFF C26;
XPXINV M1.
LET C46=C26(1)*'VL'+C26(2)*'HR'
BRIEF 2
REGRESS 'EEG' 2 'VR','HL' C19;
NOCONSTANT;
COEFF C27;
XPXINV M1.
LET C47=C27(1)*'VR'+C27(2)*'HL'
BRIEF 2
REGRESS 'EEG' 2 'VR','HR' C19;
NOCONSTANT;
COEFF C28;
XPXINV M1.
LET C48=C28(1)*'VR'+C28(2)*'HR'
BRIEF 2
REGRESS 'EEG' 2 'VL','VR' C19;
NOCONSTANT;
COEFF C29;
XPXINV M1.
LET C49=C29(1)*'VL'+C29(2)*'VR'
BRIEF 2
REGRESS 'EEG' 2 'HL','HR' C19;
NOCONSTANT;
COEFF C30;
```

```
XPXINV M1.
LET C50=C30(1)*'HL'+C30(2)*'HR'
BRIEF 2
REGRESS 'EEG' 2 'VL',C11 C19;
NOCONSTANT;
COEFF C31;
XPXINV M1.
LET C51=C31(1)*'VL'+C31(2)*C11
BRIEF 2
REGRESS 'EEG' 3 'VL','HL','HR' C19;
NOCONSTANT;
COEFF C32;
XPXINV M1.
LET C52=C32(1)*'VL'+C32(2)*'HL'+C32(3)*'HR'
BRIEF 2
REGRESS 'EEG' 3 'VL','VR','HL' C19;
NOCONSTANT;
COEFF C33;
XPXINV M1.
LET C53=C33(1)*'VL'+C33(2)*'VR'+C33(3)*'HL'
BRIEF 2
REGRESS 'EEG' 3 'VL','VR','HR' C19;
NOCONSTANT;
COEFF C34;
XPXINV M1.
LET C54=C34(1)*'VL'+C34(2)*'VR'+C34(3)*'HR'
BRIEF 2
REGRESS 'EEG' 3 'VR','HL','HR' C19;
NOCONSTANT;
COEFF C35;
XPXINV M1.
LET C55=C35(1)*'VR'+C35(2)*'HL'+C35(3)*'HR'
BRIEF 2
REGRESS 'EEG' 4 'VL','VR','HL','HR' C19;
NOCONSTANT;
COEFF C36;
XPXINV M1.
LET C56=C36(1)*'VL'+C36(2)*'VR'+C36(3)*'HL'+C36(4)*'HR'
BRIEF 2
REGRESS 'EEG' 4 'VL',C13,'HL','HR' C19;
NOCONSTANT;
COEFF C37;
XPXINV M1.
LET C57=C37(1)*'VL'+C37(2)*C13+C37(3)*'HL'+C37(4)*'HR'
BRIEF 2
REGRESS 'EEG' 4 'VL','HR',C14,C11 C19;
NOCONSTANT;
COEFF C38;
XPXINV M1.
LET C58=C38(1)*'VL'+C38(2)*'HR'+C38(3)*C14+C38(4)*C11
BRIEF 2
REGRESS 'EEG' 4 'VL',C12,'HL','HR' C19;
NOCONSTANT;
COEFF C39;
XPXINV M1.
LET C59=C39(1)*'VL'+C39(2)*C12+C39(3)*'HL'+C39(4)*'HR'
#
#OBTAIN THE OA DIFFERENCES
#
LET C61=C45-C41
LET C62=C52-C45
LET C63=C56-C52
LET C64=C58-C52
LET C65=C59-C52
LET C71=C48-C42
LET C72=C55-C48
LET C73=C56-C55
LET C74=C58-C55
LET C75=C59-C55
#
#OBTAIN THE CORRECTED EEGS
#
LET C81='EEG'-C41
LET C82='EEG'-C45
LET C83='EEG'-C52
LET C84='EEG'-C56
LET C85='EEG'-C58
LET C86='EEG'-C59
LET C87='EEG'-C42
LET C88='EEG'-C48
LET C89='EEG'-C55
LET C92='EEG'-C42
LET C93='EEG'-C43
LET C94='EEG'-C44
```

```
#
#SAVE BOTH THE OA DIFFERENCES AND THE EEGS FOR FURTHER
#PROCESSING
#
WRITE 'OADIFFOLD.DATA' C41,C61-C65
WRITE 'OAOLD.DATA' C41,C45,C52,C56,C58,C59
WRITE 'OADIFFNEW.DATA' C42,C71-C75
WRITE 'OANEW.DATA' C42,C48,C55,C56,C58,C59
WRITE 'EEGCOLD.DATA' C81-C86
WRITE 'EEGCNEW.DATA' C87-C89,C84-C86
STOP
```

PROGRAM OADIFF .MTB.

```
#
#PROGRAM USED FOR THE PRELIMINARY COMPARISON OF
#MODELS, BY THE PICTORIAL METHOD.
#
#READ DATA AND THE MEAN VALUES
#
READ 'OA.DATA' C1-C6
MEAN C1 K1
MEAN C2 K2
MEAN C3 K3
MEAN C4 K4
MEAN C5 K5
SUBTRACT K1 C1 C1
SUBTRACT K2 C2 C2
SUBTRACT K3 C3 C3
SUBTRACT K4 C4 C4
SUBTRACT K5 C5 C5
NAME C1='VL',C2='VR',C3='HL',C4='HR',C5='EEG'
LET C11='HL'-'HR'
LET C12='HL'*'HR'
LET C13='VL'*'VL'
LET C14='VL'-'VR'
#
# PERFORM OLS ANALYSIS ON DATA USING MODELS 2A, 3A, 4A
# 4B, 4D, AND 4E, AND FOR EACH MODEL OBTAIN THE OA
# ESTIMATE
#
BRIEF 2
REGRESS 'EEG' 2 'VL','HL' C19;
NOCONSTANT;
COEFF C25;
XPXINV M1.
LET C45=C25(1)*'VL'+C25(2)*'HL'
BRIEF 2
REGRESS 'EEG' 3 'VL','HL','HR' C19;
NOCONSTANT;
COEFF C32;
XPXINV M1.
LET C52=C32(1)*'VL'+C32(2)*'HL'+C32(3)*'HR'
BRIEF 2
REGRESS 'EEG' 4 'VL','VR','HL','HR' C19;
NOCONSTANT;
COEFF C36;
XPXINV M1.
LET C56=C36(1)*'VL'+C36(2)*'VR'+C36(3)*'HL'+C36(4)*'HR'
BRIEF 2
REGRESS 'EEG' 4 'VL',C13,'HL','HR' C19;
NOCONSTANT;
COEFF C37;
XPXINV M1.
LET C57=C37(1)*'VL'+C37(2)*C13+C37(3)*'HL'+C37(4)*'HR'
BRIEF 2
REGRESS 'EEG' 4 'VL',C12,'HL','HR' C19;
NOCONSTANT;
COEFF C39;
XPXINV M1.
LET C59=C39(1)*'VL'+C39(2)*C12+C39(3)*'HL'+C39(4)*'HR'
BRIEF 2
REGRESS 'EEG' 4 'VL',C6,'HL','HR' C19;
NOCONSTANT;
COEFF C40.
LET C60=C40(1)*'VL'+C40(2)*C6+C40(3)*'HL'+C40(4)*'HR'
#
#OBTAIN OA DIFFERENCES AND SAVE BOTH OA DIFFERENCES
#AND OAS FOR FURTHER ANALYSIS.
#
LET C62=C52-C45
LET C63=C56-C52
LET C64=C57-C52
LET C65=C59-C52
LET C66=C60-C52
WRITE 'OADIFFOLD.DATA' C62-C66,C62
WRITE 'OAOLD.DATA' C45,C52,C56,C57,C59,C60
STOP
```

PROGRAM OLSAR.MTB

```
#
#PROGRAM IS USED TO STUDY THE OLS WITH RAW DATA, OLS WITH DIFFERECED DATA,
#AND AR(1) ESTIMATION METHODS USING MODELS 3D,4A,
#AND 4D, AND ALLOWS COMPARISON OF THE OFF-LINE
#AND ON-LINE RESULST
#
#READ DATA AND REMOVE MEAN VALUES.
#
READ 'MTB.DATA' C1-C5
MEAN C1 K1
MEAN C2 K2
MEAN C3 K3
MEAN C4 K4
MEAN C5 K5
SUBTRACT K1 C1 C1
SUBTRACT K2 C2 C2
SUBTRACT K3 C3 C3
SUBTRACT K4 C4 C4
SUBTRACT K5 C5 C5
NAME C1='VL',C2='VR',C3='HL',C4='HR',C5='EEG'
NAME C6='VLL',C7='VRL',C8='HLL',C9='HRL',C10='EEGL'
NAME C15='VLST',C16='VRST',C17='HLST',C18='HRST'
NAME C19='EEGST',C80='VLD'
NAME C11='VRD',C12='HLD',C13='HRD',C14='EEGD'
#
#DIFFERENCE THE DATA
#
DIFF 'VL','VLD'
DIFF 'VR','VRD'
DIFF 'HL','HLD'
DIFF 'HR','HRD'
DIFF 'EEG','EEGD'
LET C81='HL'*'HR'
LET C82='HLD'*'HRD'
#
#PERFORM OLS WITH RAW AND WITH DIFFERENCED DATA, AND AR(1) ANALYSIS ON DATA
#USING MODEL 3D, AND OBTAIN ESTIMATES OF OAS AND EEGCS.
#
BRIEF 2
REGRESS 'EEG' 3 'VR','HL','HR' C95;
NOCONSTANT;
MSE K1;
COEFF C24;
XPXINV M1.
LET C51=SQRT(K1)*C95
LET C54='EEG'-C51
BRIEF 2
REGRESS 'EEGD' 3 'VRD','HLD','HRD' C95;
NOCONSTANT;
MSE K1;
COEFF C24;
XPXINV M1.
LET C64=C24(1)*'VR'+C24(2)*'HL'+C24(3)*'HR'
LET C61='EEG'-C64
LET C22=C51
ACF 1 C22,C97
LET C31='VR'
LET C32='HL'
LET C33='HR'
EXEC 'GLS3.MTB' 3
LET C71=C22
LET C74='EEG'-C71
#
#REPEAT ABOVE OPERATIONS USING MODEL 4A
#
BRIEF 2
REGRESS 'EEG' 4 'VL','VR','HL','HR' C95;
NOCONSTANT;
MSE K1;
COEFF C24;
XPXINV M1.
LET C52=SQRT(K1)*C95
LET C55='EEG'-C52
BRIEF 2
REGRESS 'EEGD' 4 'VLD','VRD','HLD','HRD' C95;
NOCONSTANT;
MSE K1;
COEFF C24;
XPXINV M1.
LET C65=C24(1)*'VL'+C24(2)*'VR'+C24(3)*'HL'+C24(4)*'HR'
LET C62='EEG'-C65
LET C22=C52
ACF 1 C22,C97
LET C31='VL'
LET C31='VL'
LET C32='VR'
LET C33='HL'
LET C34='HR'
EXEC 'GLS3.MTB' 3
LET C72=C22
LET C75='EEG'-C72
#
#REPEAT ABOVE OPERATIONS USING MODEL 4D
#
BRIEF 2
REGRESS 'EEG' 4 'VR',C81,'HL','HR' C95;
NOCONSTANT;
MSE K1;
COEFF C24;
XPXINV M1.
LET C53=SQRT(K1)*C95
LET C56='EEG'-C53
BRIEF 2
REGRESS 'EEGD' 4 'VRD',C82,'HLD','HRD' C95;
NOCONSTANT;
MSE K1;
COEFF C24;
XPXINV M1.
LET C66=C24(1)*'VR'+C24(2)*C81+C24(3)*'HL'+C24(4)*'HR'
LET C63='EEG'-C66
LET C22=C53
ACF 1 C22,C97
LET C31='VR'
LET C32=C81
LET C33='HL'
LET C34='HR'
EXEC 'GLS3.MTB' 3
LET C73=C22
LET C76='EEG'-C73
#
#SAVE PROCESSED DATA FOR LATER USE.
#
WRITE 'OLS3D4A4D.DATA' C54,C51,C55,C52,C56,C53
WRITE 'DIF3D4A4D.DATA' C64,C61,C65,C62,C66,C63
WRITE 'ARS3D4A4D.DATA' C74,C71,C75,C72,C76,C73
STOP
```

MACRO ESTIMATE.MTB

```
#PROGRAM PERFORMS OLS ANALYSIS ON DATA FOR
#SEVERAL MODELS.
#
SSQ 'EEG' K20
BRIEF 2
REGRESS 'EEG' 1 'VL' C19;
NOCONSTANT;
MSE K1;
COEFF C21;
XPXINV M1.
LET K31=1023.0*K1
LET C88(1)=K31/1022.0
DIAGONAL M1,C60
LET C41=C88(1)*C60
BRIEF 2
REGRESS 'EEG' 1 'VR' C19;
NOCONSTANT;
MSE K2;
COEFF C22;
XPXINV M1.
LET K32=1023.0*K2
LET C88(2)=K32/1022.0
DIAGONAL M1,C60
LET C42=C88(2)*C60
BRIEF 2
REGRESS 'EEG' 1 'HL' C19;
NOCONSTANT;
MSE K3;
COEFF C23;
XPXINV M1.
LET K33=1023.0*K3
LET C88(3)=K33/1022.0
DIAGONAL M1,C60
LET C43=C88(3)*C60
REGRESS 'EEG' 1 'HR' C19;
NOCONSTANT;
MSE K4;
COEFF C24;
XPXINV M1.
LET K34=1023.0*K4
LET C88(4)=K34/1022.0
DIAGONAL M1,C60
LET C44=C88(4)*C60
BRIEF 2
REGRESS 'EEG' 2 'VL','HL' C19;
NOCONSTANT;
MSE K5;
COEFF C25;
XPXINV M1.
LET K35=1022.0*K5
LET C88(5)=K35/1021.0
DIAGONAL M1,C60
LET C45=C88(5)*C60
BRIEF 2
REGRESS 'EEG' 2 'VL','HR' C19;
NOCONSTANT;
MSE K6;
COEFF C26;
XPXINV M1.
LET K36=1022.0*K6
LET C88(6)=K36/1021.0
DIAGONAL M1,C60
LET C46=C88(6)*C60
BRIEF 2
REGRESS 'EEG' 2 'VR','HL' C19;
NOCONSTANT;
MSE K7;
COEFF C27;
XPXINV M1.
LET K37=1022.0*K7
LET C88(7)=K37/1021.0
DIAGONAL M1,C60
LET C47=C88(7)*C60
BRIEF 2
REGRESS 'EEG' 2 'VR','HR' C19;
NOCONSTANT;
MSE K8;
COEFF C28;
XPXINV M1.
LET K38=1022.0*K8
LET C88(8)=K38/1021.0
DIAGONAL M1,C60
LET C48=C88(8)*C60
BRIEF 2
```

```
REGRESS 'EEG' 2 'VL','VR' C19;
NOCONSTANT;
MSE K9;
COEFF C29;
XPXINV M1.
LET K39=1022.0*K9
LET C88(9)=K39/1021.0
DIAGONAL M1,C60
LET C49=C88(9)*C60
BRIEF 2
REGRESS 'EEG' 2 'HL','HR' C19;
NOCONSTANT;
MSE K10;
COEFF C30;
XPXINV M1.
LET K40=1022.0*K10
LET C88(10)=K40/1021.0
DIAGONAL M1,C60
LET C50=C88(10)*C60
BRIEF 2
REGRESS 'EEG' 2 'VL',C11 C19;
NOCONSTANT;
MSE K11;
COEFF C31;
XPXINV M1.
LET K41=1022.0*K11
LET C88(11)=K41/1021.0
DIAGONAL M1,C60
LET C51=C88(11)*C60
BRIEF 2
REGRESS 'EEG' 3 'VL','HL','HR' C19;
NOCONSTANT;
MSE K12;
COEFF C32;
XPXINV M1.
LET K42=1021.0*K12
LET C88(12)=K42/1020.0
DIAGONAL M1,C60
LET C52=C88(12)*C60
BRIEF 2
REGRESS 'EEG' 3 'VL','VR','HL' C19;
NOCONSTANT;
MSE K13;
COEFF C33;
XPXINV M1.
LET K43=1021.0*K13
LET C88(13)=K43/1020.0
DIAGONAL M1,C60
LET C53=C88(13)*C60
REGRESS 'EEG' 3 'VL','VR','HR' C19;
NOCONSTANT;
MSE K14;
COEFF C34;
XPXINV M1.
LET K44=1021.0*K14
LET C88(14)=K44/1020.0
DIAGONAL M1,C60
LET C54=C88(14)*C60
BRIEF 2
REGRESS 'EEG' 3 'VR','HL','HR' C19;
NOCONSTANT;
MSE K15;
COEFF C35;
XPXINV M1.
LET K45=1021.0*K15
LET C88(15)=K45/1020.0
DIAGONAL M1,C60
LET C55=C88(15)*C60
BRIEF 2
REGRESS 'EEG' 4 'VL','VR','HL','HR' C19;
NOCONSTANT;
MSE K16;
COEFF C36;
XPXINV M1.
LET K46=1020.0*K16
LET C88(16)=K46/1019.0
DIAGONAL M1,C60
PRINT C36,C60
PRINT K16
LET C56=C88(16)*C60
BRIEF 2
REGRESS 'EEG' 4 'VL',C13,'HL','HR' C19;
NOCONSTANT;
MSE K17;
```

```
COEFF C37;
XPXINV M1.
LET K47=1020.0*K17
LET C88(17)=K47/1019.0
DIAGONAL M1,C60
LET C57=C88(17)*C60
BRIEF 2
REGRESS 'EEG' 4 'VL','HR',C14,C11 C19;
NOCONSTANT;
MSE K18;
COEFF C38;
XPXINV M1.
LET K48=1020.0*K18
LET C88(18)=K48/1019.0
DIAGONAL M1,C60
LET C58=C88(18)*C60
BRIEF 2
REGRESS 'EEG' 4 'VL',C12,'HL','HR' C19;
NOCONSTANT;
MSE K19;
COEFF C39;
XPXINV M1.
LET K49=1020.0*K19
LET C88(19)=K49/1019.0
DIAGONAL M1,C60
LET C59=C88(19)*C60
GENERATE 1,STEP 1,TO 19,PUT IN C92
#
#OBTAIN THE STANDARD ERRORS OF THE PARAMETER
#ESTIMATES FOR THE VARIOUS MODELS.
#
LET C84(1)=C21(1)/SQRT(C41(1))
LET C84(2)=C22(1)/SQRT(C42(1))
LET C84(3)=C23(1)/SQRT(C43(1))
LET C84(4)=C24(1)/SQRT(C44(1))
LET C84(5)=C25(1)/SQRT(C45(1))
LET C85(5)=C25(2)/SQRT(C45(2))
LET C84(6)=C26(1)/SQRT(C46(1))
LET C85(6)=C26(2)/SQRT(C46(2))
LET C84(7)=C27(1)/SQRT(C47(1))
LET C85(7)=C27(2)/SQRT(C47(2))
LET C84(8)=C28(1)/SQRT(C48(1))
LET C85(8)=C28(2)/SQRT(C48(2))
LET C84(9)=C29(1)/SQRT(C49(1))
LET C85(9)=C29(2)/SQRT(C49(2))
LET C84(10)=C30(1)/SQRT(C50(1))
LET C85(10)=C30(2)/SQRT(C50(2))
LET C84(11)=C31(1)/SQRT(C51(1))
LET C85(11)=C31(2)/SQRT(C51(2))
LET C84(12)=C32(1)/SQRT(C52(1))
LET C85(12)=C32(2)/SQRT(C52(2))
LET C86(12)=C32(3)/SQRT(C52(3))
LET C84(13)=C33(1)/SQRT(C53(1))
LET C85(13)=C33(2)/SQRT(C53(2))
LET C86(13)=C33(3)/SQRT(C53(3))
LET C84(14)=C34(1)/SQRT(C54(1))
LET C85(14)=C34(2)/SQRT(C54(2))
LET C86(14)=C34(3)/SQRT(C54(3))
LET C84(15)=C35(1)/SQRT(C55(1))
LET C85(15)=C35(2)/SQRT(C55(2))
LET C86(15)=C35(3)/SQRT(C55(3))
LET C84(16)=C36(1)/SQRT(C56(1))
LET C85(16)=C36(2)/SQRT(C56(2))
LET C86(16)=C36(3)/SQRT(C56(3))
LET C87(16)=C36(4)/SQRT(C56(4))
LET C84(17)=C37(1)/SQRT(C57(1))
LET C85(17)=C37(2)/SQRT(C57(2))
LET C86(17)=C37(3)/SQRT(C57(3))
LET C87(17)=C37(4)/SQRT(C57(4))
LET C84(18)=C38(1)/SQRT(C58(1))
LET C85(18)=C38(2)/SQRT(C58(2))
LET C86(18)=C38(3)/SQRT(C58(3))
LET C87(18)=C38(4)/SQRT(C58(4))
LET C84(19)=C39(1)/SQRT(C59(1))
LET C85(19)=C39(2)/SQRT(C59(2))
LET C86(19)=C39(3)/SQRT(C59(3))
LET C87(19)=C39(4)/SQRT(C59(4))
#
#OBTAIN AND PRINT THE PARAMETER ESTIMATES OF THE VARIOUS
#MODELS.
#
LET C80(1)=C21(1)
LET C80(2)=C22(1)
LET C80(3)=C23(1)
LET C80(4)=C24(1)
LET C80(5)=C25(1)
LET C81(5)=C25(2)
LET C80(6)=C26(1)
LET C81(6)=C26(2)
LET C80(7)=C27(1)
LET C81(7)=C27(2)
LET C80(8)=C28(1)
LET C81(8)=C28(2)
LET C80(9)=C29(1)
LET C81(9)=C29(2)
LET C80(10)=C30(1)
LET C81(10)=C30(2)
LET C80(11)=C31(1)
LET C81(11)=C31(2)
LET C80(12)=C32(1)
LET C81(12)=C32(2)
LET C82(12)=C32(3)
LET C80(13)=C33(1)
LET C81(13)=C33(2)
LET C82(13)=C33(3)
LET C80(14)=C34(1)
LET C81(14)=C34(2)
LET C82(14)=C34(3)
LET C80(15)=C35(1)
LET C81(15)=C35(2)
LET C82(15)=C35(3)
LET C80(16)=C36(1)
LET C81(16)=C36(2)
LET C82(16)=C36(3)
LET C83(16)=C36(4)
LET C80(17)=C37(1)
LET C81(17)=C37(2)
LET C82(17)=C37(3)
LET C83(17)=C37(4)
LET C80(18)=C38(1)
LET C81(18)=C38(2)
LET C82(18)=C38(3)
LET C83(18)=C38(4)
LET C80(19)=C39(1)
LET C81(19)=C39(2)
LET C82(19)=C39(3)
LET C83(19)=C39(4)
PRINT C92,C80-C87
#
#OBTAIN AND PRINT THE TEST STATISTICS FOR THE VARIOUS
#MODELS
#
LET C89(1)=1.0-K31/K20
LET C89(2)=1.0-K32/K20
LET C89(3)=1.0-K33/K20
LET C89(4)=1.0-K34/K20
LET K70=1023.0/1022.0
LET C89(5)=1.0-K70*K35/K20
LET C89(6)=1.0-K70*K36/K20
LET C89(7)=1.0-K70*K37/K20
LET C89(8)=1.0-K70*K38/K20
LET C89(9)=1.0-K70*K39/K20
LET C89(10)=1.0-K70*K40/K20
LET C89(11)=1.0-K70*K41/K20
LET K70=1023.0/1021.0
LET C89(12)=1.0-K70*K42/K20
LET C89(13)=1.0-K70*K43/K20
LET C89(14)=1.0-K70*K44/K20
LET C89(15)=1.0-K70*K45/K20
LET C89(16)=1.0-K70*K46/K20
LET C89(17)=1.0-K70*K47/K20
LET C89(18)=1.0-K70*K48/K20
LET C89(19)=1.0-K70*K49/K20
LET K70=1022.0
LET C90(1)=(K31/C88(16))-K70
LET C90(2)=(K32/C88(16))-K70
LET C90(3)=(K33/C88(16))-K70
LET C90(4)=(K34/C88(16))-K70
LET K70=1020.0
LET C90(5)=(K35/C88(16))-K70
LET C90(6)=(K36/C88(16))-K70
LET C90(7)=(K37/C88(16))-K70
LET C90(8)=(K38/C88(16))-K70
LET C90(9)=(K39/C88(16))-K70
LET C90(10)=(K40/C88(16))-K70
LET C90(11)=(K41/C88(16))-K70
LET K70=1018.0
LET C90(12)=(K42/C88(16))-K70
LET C90(12)=(K42/C88(16))-K70
LET C90(13)=(K43/C88(16))-K70
LET C90(14)=(K44/C88(16))-K70
LET C90(15)=(K45/C88(16))-K70
LET K70=1016.0
LET C90(16)=(K46/C88(16))-K70
LET C90(17)=(K47/C88(16))-K70
LET C90(18)=(K48/C88(16))-K70
LET C90(19)=(K49/C88(16))-K70
LET K70=1021.0
LET C91(5)=K70*(K31-K35)/K35
LET C91(6)=K70*(K31-K36)/K36
LET C91(7)=K70*(K32-K37)/K37
LET C91(8)=K70*(K32-K38)/K38
LET C91(9)=K70*(K31-K39)/K39
LET C91(10)=K70*(K33-K40)/K40
LET C91(11)=K70*(K31-K41)/K41
LET K70=1020.0
LET C91(12)=K70*(K35-K42)/K42
LET C91(13)=K70*(K35-K43)/K43
LET C91(14)=K70*(K36-K44)/K44
LET C91(15)=K70*(K37-K45)/K45
LET K70=1019.0
LET C91(16)=K70*(K42-K46)/K46
LET C91(17)=K70*(K42-K47)/K47
LET C91(18)=K70*(K36-K48)/K48
LET C91(19)=K70*(K42-K49)/K49
PRINT C92,C88-C91
END
```

MARCO GLS2.MTB

```
LAG C22,C20
BRIEF 2
REGRESS C22 1 C20;
NOCONSTANT;
COEFF C23.
LAG 'EEG','EEGL'
LAG C31,'VRL'
LAG C32,'HRL'
LET 'EEGST'='EEG'-C23(1)*'EEGL'
LET 'VRST'=C31-C23(1)*'VRL'
LET 'HRST'=C32-C23(1)*'HRL'
BRIEF 2
REGRESS 'EEGST' 2 'VRST','HRST' C95;
NOCONSTANT;
MSE K1;
COEFF C24;
XPXINV M1.
LET C96=SQRT(K1)*C95
ACF 1 C96,C97
LET K2=C97(1)
LET K3=2.0*(1.0-K2)
LET K32=1022.0*K1
LET K33=K32/1020.0
DIAGONAL M1,C41
LET C42=K33*C41
LET C43(1)=C24(1)/SQRT(C42(1))
LET C43(2)=C24(2)/SQRT(C42(2))
LET C22='EEG'-(C24(1)*C31+C24(2)*C32)
END
```

MACRO GLS3.MTB

```
LAG C22,C20
BRIEF 2
REGRESS C22 1 C20;
NOCONSTANT;
COEFF C23.
LAG 'EEG','EEGL'
LAG C31,'VRL'
LAG C32,'HLL'
LAG C33,'HRL'
LET 'EEGST'='EEG'-C23(1)*'EEGL'
LET 'VRST'=C31-C23(1)*'VRL'
LET 'HLST'=C32-C23(1)*'HLL'
LET 'HRST'=C33-C23(1)*'HRL'
BRIEF 2
REGRESS 'EEGST' 3 'VRST','HLST','HRST' C95;
NOCONSTANT;
MSE K1;
COEFF C24;
XPXINV M1.
LET C96=SQRT(K1)*C95
ACF 1 C96,C97
LET K2=C97(1)
LET K3=2.0*(1.0-K2)
LET K32=1020.0*K1
LET K33=K32/1019.0
DIAGONAL M1,C41
LET C42=K33*C41
LET C43(1)=C24(1)/SQRT(C42(1))
LET C43(2)=C24(2)/SQRT(C42(2))
LET C43(3)=C24(3)/SQRT(C42(3))
LET C22='EEG'-(C24(1)*C31+C24(2)*C32+C24(3)*C33)
END
```

MACRO GLS4.MTB

```
LAG C22,C20
BRIEF 2
REGRESS C22 1 C20;
NOCONSTANT;
COEFF C23.
LAG 'EEG','EEGL'
LAG C31,'VLL'
LAG C32,'VRL'
LAG C33,'HLL'
LAG C34,'HRL'
LET 'EEGST'='EEG'-C23(1)*'EEGL'
LET 'VLST'=C31-C23(1)*'VLL'
LET 'VRST'=C32-C23(1)*'VRL'
LET 'HLST'=C33-C23(1)*'HLL'
LET 'HRST'=C34-C23(1)*'HRL'
BRIEF 2
REGRESS 'EEGST' 4 'VLST','VRST','HLST','HRST' C95;
NOCONSTANT;
MSE K1;
COEFF C24;
XPXINV M1.
LET C96=SQRT(K1)*C95
ACF 1 C96,C97
LET K2=C97(1)
LET K3=2.0*(1.0-K2)
PRINT K2,K3
LET K32=1019.0*K1
LET K33=K32/1018.0
DIAGONAL M1,C41
LET C42=K33*C41
LET C43(1)=C24(1)/SQRT(C42(1))
LET C43(2)=C24(2)/SQRT(C42(2))
LET C43(3)=C24(3)/SQRT(C42(3))
LET C43(4)=C24(4)/SQRT(C42(4))
LET C22='EEG'-(C24(1)*C31+C24(2)*C32+C24(3)*C33+C24(4)*C34)
END
```

Appendix A12 FORTRAN Programs for On-line Removal of OA

This appendix gives the relevant programs used for on-line analysis. There is one main program, ONLCRTN.FTN,which contains several subroutines, and a set of library of subprograms for the three on-line algorithms described in chapter 4, viz, Recursive least squares, square root and UD algorithms. The library of subprograms are in a common file called ONLALG.FTN. The main program, ONLCRTN.FTN, also makes use of a number of other subprograms, some of which were given in appendix A11; these are: RDATA (used to read data from a binary file - see the appendix A11; NGRAPH and NGRFS which are used to plot various graphs (e.g parameter estimates, corrected EEG).

As discussed in the last appendix, some of the codes in the programs are system dependent. These are mostly, commands used to open and close files (e.g. OPEN$A) and the graphic routines which are mostly calls to a graphic package on the PRIME system called GINO-F. Apart from these system dependent codes, the programs are standard FORTRAN IV codes.

Index to Program

```
                PROGRAM  ONLCRTN.FTN
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-SPECIFIED ON-LINE ALGORITHMS
C (THESE ARE RLS,SQRT AND UD FILTERS).
C
      INTEGER*2 FNAME(20)
      COMMON VL(1024),VR(1024),HL(1024),HR(1024)
      COMMON /VDATA/EEG1(1024),EEG2(1024),FNAME
      COMMON /XVAL/X(1024)
      COMMON /INTLV/ALPHA,GAMMA,THETA0
      DATA L2/0/
      NPT=1024
      N=1024
      DO 45 I=1,N
      IM1=I-1
45    X(I)=FLOAT(IM1)/125.0
C
C SPECIFY GAMMA AND THE INITIAL VALUES FOR
C P (OR S OR U).
C
      WRITE(1,16)
      WRITE(1,18)
      WRITE(1,20)
16    FORMAT('ENTER ALPHA -INITIAL VALUE FOR PS')
18    FORMAT('         GAMMA -THE FORGETTING FACTOR')
20    FORMAT('  AND THETA(0) - THE INITIAL VALUE FOR THETA')
      READ(1,*)ALPHA,GAMMA,THETA0
      CALL DEVICE(IDEV)
C
C SPECIFY AND THEN READ THE EEG RECORD TO PROCESS
C
      WRITE(1,10)
10    FORMAT('ENTER BATNO')
      READ(1,*)L1
      L4=0
      CALL RDATA(L1,L2,L3,L4)
      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)
C
C SELECT ON-LINE ALGORITHM TO USE
C
      WRITE(1,30)
30    FORMAT('ENTER 0 FOR RLS 1 FOR SQRT 2 FOR UDU')
      READ(1,*)ITYPE
      CALL ADAPTV(NPT,N,L1,ITYPE,IPLOT)
5     CONTINUE
100   CALL DEVEND
      CALL EXIT
      END
C
C---------------------------------------------------
C
      SUBROUTINE ADAPTV(NPT,N,L1,ITYPE,IPLOT)
      DIMENSION EEGC3(1024),EEGCA(1024),EEGCD(1024)
      DIMENSION THETA3(4096),THETAA(4096),THETAD(4096)
      DIMENSION RSS3(1024),RSSA(1024),RSSD(1024)
      DIMENSION RSQ3(1024),RSQA(1024),RSQD(1024)
      DIMENSION OA3(1024),OAA(1024),OAD(1024)
      DATA NPAR3,NPAR4/3,4/
      DATA ID3,IDA,IDD/3,4,5/
      CALL XYVAL(THETA3,EEGC3,RSS3,RSQ3,OA3,NPAR3,ID3,ITYPE)
      CALL XYVAL(THETAA,EEGCA,RSSA,RSQA,OAA,NPAR4,IDA,ITYPE)
      CALL XYVAL(THETAD,EEGCD,RSSD,RSQD,OAD,NPAR4,IDD,ITYPE)
      CALL OAGRAF(OA3,EEGC3,OAA,EEGCA,OAD,EEGCD,N)
      CALL THGRAF(THETA3,THETAA,THETAD,N)
      CALL SSGRAF(RSS3,RSSA,RSSD,RSQ3,RSQA,RSQD,N)
      RETURN
      END
```

```
C
C-----------------------------------------------------------
C
      SUBROUTINE OAGRAF(D0,D1,D2,D3,D4,D5,N)
      DIMENSION D0(1024),D1(1024),D2(1024),D3(1024)
      DIMENSION D4(1024),D5(1024)
      COMMON /XVAL/X(1024)
      CALL PICCLE
      CALL NGRAPH(N,X,D0,0,0)
      CALL NGRAPH(N,X,D1,1,0)
      CALL NGRAPH(N,X,D2,2,0)
      CALL NGRAPH(N,X,D3,3,0)
      CALL NGRAPH(N,X,D4,4,0)
      CALL NGRAPH(N,X,D5,5,0)
      RETURN
      END
C
C----------------------------------------------------------
C
      SUBROUTINE THGRAF(D0,D1,D2,N)
      DIMENSION D0(4096),D1(4096),D2(4096)
      COMMON /XVAL/X(1024)
      ISCALE=3
      CALL PICCLE
      CALL NGRFS1(N,3,X,D0,0,3,ISCALE)
      CALL NGRFS1(N,4,X,D1,2,3,ISCALE)
      CALL NGRFS1(N,4,X,D2,4,3,ISCALE)
      RETURN
      END
C
C----------------------------------------------------------
C
      SUBROUTINE SSGRAF(D0,D1,D2,D3,D4,D5,N)
      DIMENSION D0(1024),D1(1024),D2(1024),D3(1024)
      DIMENSION D4(1024),D5(1024)
      DIMENSION DD0(4096),DD1(4096)
      COMMON /XVAL/X(1024)
      I1=2*N
      DO 10 I=1,N
      DD0(I)=D0(I)
      DD0(I+N)=D1(I)
      DD0(I+I1)=D2(I)
      DD1(I)=D3(I)
      DD1(I+N)=D4(I)
10    DD1(I+I1)=D5(I)
      CALL PICCLE
      CALL NGRFS1(N,3,X,DD0,0,6,ISCALE)
      CALL NGRFS1(N,3,X,DD1,2,7,ISCALE)
      RETURN
      END
C
C----------------------------------------------------------
C
      SUBROUTINE XYVAL(PARM,DOUT,RSS,RSQ,OA,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(4096),DOUT(1024),RSS(1024),RSQ(1024),OA(1024)
      COMMON VL(1024),VR(1024),HL(1024),HR(1024)
      N=1024
      IF(ID .LT. 5)GO TO 110
      DO 150 I=1,N
150   VL(I)=HR(I)*HL(I)
110   CONTINUE
      CALL ONLSUB(PARM,DOUT,RSS,RSQ,OA,NPAR,ITYPE)
      RETURN
      END
C
C-------------------------------------------------------
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(4),PS(4,4),THETA(4),SST(1024),S(10),U(10)
      DIMENSION PARM(4096),DOUT(1024),RSS(1024),RSQ(1024),OA(1024)
      INTEGER*2 FNAME(20)
      COMMON /VDATA/EEG1(1024),EEG2(1024),FNAME
      COMMON /INTLV/ ALPHA,GAMMA,THETA0
      COMMON /INTLV/ ALPHA,GAMMA,THETA0
      DATA J1,J2,J3/1024,2048,3072/
      N=1024
      NPT=N
      DO 40 I=1,10
      S(I)=0.0
      U(I)=0.0
40    CONTINUE
      DO 50 I=1,4
      DO 50 J=1,4
      PS(I,J)=0.0
50    CONTINUE
      WRITE(1,12)
12    FORMAT('ENTER NUMBER OF CYCLES')
      READ(1,*)NTIMES
C     INITIALIZE 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(1,111)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)
      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
      RETURN
      END
C
C-----------------------------------------------------
C
      SUBROUTINE XVAL1(XN,NPAR,I1)
C PRIME THE VECTOR X WITH EOG DATA
C
      DIMENSION XN(4)
      COMMON VL(1024),VR(1024),HL(1024),HR(1024)
      XN(1)=VR(I1)
      XN(2)=HR(I1)
      XN(3)=HL(I1)
      XN(4)=VL(I1)
      RETURN
      END
```

SUBPROGRAM ONLALG.FTN

```
      SUBROUTINE RLSFLT(Y,X,P,THETA,NPAR,E,GAMMA,I1)
C
C THIS SUBROUTINE IS BASED ON BIERMAN'S CODING OF THE KALMAN
C FILTERING ALGORITHM. IT COMPUTES THE PARAMETER ESTIMATES,
C THE OA ESTIMATES AND THE CORRECTED EEG USING RECURSIVE LEAST
C SQUARES
C
C INPUTS:
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    SAMPLE NUMBER
C        P     THE COVARIANCE MATRIX
C
C OUTPUTS:
C        THETA UPDATED PARAMETER ESTIMATE VECTOR
C        E     THE CORRECTED EEG SAMPLE
C        P     UPDATED COVARIANCE MATRIX
C
      DIMENSION X(4),P(4,4),THETA(4),V(4),G(4)
      PERR=Y
      SF=1.0/GAMMA
      DELTA=GAMMA
      DO 10 I=1,NPAR
      V(I)=0.0
      DO 5 J=1,NPAR
5     V(I)=V(I)+P(I,J)*X(J)
      PERR=PERR-X(I)*THETA(I)
10    DELTA=DELTA+X(I)*V(I)
      DO 20 I=1,NPAR
      G(I)=V(I)/DELTA
      THETA(I)=THETA(I)+G(I)*PERR
      DO 20 J=I,NPAR
      P(I,J)=(P(I,J)-G(I)*V(J))*SF
20    P(J,I)=P(I,J)
      E=Y
C
C REMOVE THE EMA ESTIMATE FROM THE EEG
C
      DO 30 I=1,NPAR
30    E=E-X(I)*THETA(I)
      RETURN
      END
C
C--------------------------------------------------
C
      SUBROUTINE MEAN(NPTS,DATA,RMEAN)
      DIMENSION DATA(NPTS)
      RMEAN=0.0
      DO 20 I=1,NPTS
20    RMEAN=RMEAN+DATA(I)
      RMEAN=RMEAN/FLOAT(NPTS)
      DO 30 I=1,NPTS
30    DATA(I)=DATA(I)-RMEAN
      RETURN
      END
C
C--------------------------------------------------
C
      SUBROUTINE SQRTFL(Y,X,SQRTP,PAR,NPAR,E,GAMMA)
C
C     CLARKE"S CODING OF PETERKA S ALGORITHM
C INPUTS:
C     Y,X EEG SAMPLE AND EOG DATA VECTOR
C     PAR VECTOR OF PREVIOUS PARAMETER ESTIMATES
C     NPAR  NUMBER OF PARAMETER ESTIMATES
C     SQRTP SQUARE ROOT OF P
C     GAMMA  FORGETING FACTOR
C
C OUTPUTS:
C     E    CORRECTED EEG SAMPLE
C     THETA UPDATED PARAMETER ESTIMATES
C     SQRTP UPDATED SQUARE ROOT OF P
C     G    KALMAN GAIN VECTOR
C
C     G  KALMAN GAIN VECTOR
C     THETA  VECTOR OF PARAMETERS TO BE ESTIMATED
C     NPAR  NUMBER OF PARAMETERS TO BE ESTIMATED
C     FORGT SQUARE ROOT OF THE FORGETTING FACTOR  GAMMA
C
      DIMENSION X(4),SQRTP(10),G(4),PAR(4)
      PERR=Y
      DO 5 I=1,NPAR
      DO 5 I=1,NPAR
5     PERR=PERR-PAR(I)*X(I)
      FORGT=SQRT(GAMMA)
      SIG=FORGT
      SIGSQ=FORGT*FORGT
      IJ=0
      JI=0
      DO 10 J=1,NPAR
      J1=J-1
      FJ=0.0
      DO 20 I=1,J
      JI=JI+1
20    FJ=FJ+SQRTP(JI)*X(I)
      A=SIG/FORGT
      B=FJ/SIGSQ
      SIGSQ=SIGSQ+FJ*FJ
      SIG=SQRT(SIGSQ)
      A=A/SIG
      G(J)=SQRTP(JI)*FJ
      SQRTP(JI)=A*SQRTP(JI)
      IF(J1 .EQ. 0)GO TO 31
      DO 30 I=1,J1
      IJ=IJ+1
      SQP=SQRTP(IJ)
      SQRTP(IJ)=A*(SQP-B*G(I))
      G(I)=G(I)+SQP*FJ
30    CONTINUE
31    CONTINUE
      IJ=IJ+1
10    CONTINUE
      DO 40 I=1,NPAR
      PAR(I)=PAR(I)+G(I)*PERR/SIGSQ
40    CONTINUE
      E=Y
      DO 50 I=1,NPAR
50    E=E-X(I)*PAR(I)
      RETURN
      END
C
C-----------------------------------------------------
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 OUTPUTS:
C        U  UPDATED UPPER TRIANGULAR MATRIX
C        B  THE UNWEIGHTED KALMAN GAIN
C        THETA  UPDATED PARAMETER ESTIMATES
C        E  CORRECTED EEG
C
      DIMENSION X(4),U(10),THETA(4),B(4),V(4)
      SF=1.0/GAMMA
      PERR=Y
      DO 2 J=1,NPAR
2     PERR=PERR-X(J)*THETA(J)
      M=1
      V(1)=X(1)
      DO 10 J=2,NPAR
      V(J)=X(J)
      J1=J-1
      DO 5 K=1,J1
      M=M+1
5     V(J)=V(J)+U(M)*X(K)
      M=M+1
10    B(J)=U(M)*V(J)
      B(1)=U(1)*X(1)
      ALPHA=GAMMA+B(1)*V(1)
      DELTA=1.0/ALPHA
      U(1)=U(1)*DELTA
      M=1
      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
20    B(K)=B(K)+B(J)*BETA
      M=M+1
25    U(M)=U(M)*BETA1*DELTA*SF
      PERR=PERR/ALPHA
      DO 30 J=1;NPAR
30    THETA(J)=THETA(J)+B(J)*PERR
      E=Y
      DO 35 I=1,NPAR
35    E=E-X(I)*THETA(I)
      RETURN
      END
```

SUBPROGRAM NGRFS.FTN

```
      SUBROUTINE NGRFS1(N,NGRF,X,DIN,ING,IST,ISCALE)
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 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
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*2 ILAB0(4),ILAB1(2),ILAB2(2),ILAB3(9),ILAB4(5)
      INTEGER*2 ILAB5(5),ILAB6(8),ILAB7(7)
      DATA ILAB0,ILAB1,ILAB2/'AMP(UV) ','ACF ','CCF '/
      DATA ILAB3,ILAB4/'estmted parameters','ERROR NORM'/
      DATA ILAB5/'parameters'/
      DATA ILAB6,ILAB7/'sample variance','mult cor coeff'/
C
C DETERMINE THE PART OF THE PAGE TO PLOT GRAPH
C
      X1=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
      CALL MAXVAL(BBEG,BEND,DIN,N,NGRF)
      IF(X(1) .GE. 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 THESE
C
60    CALL AXIPOS(IOR,X3,Y3,90.,1)
      CALL AXIPOS(IOR,X3,Y3,50.,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.)
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. 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 NPLOT(DIN,N,Y,I)
70    CALL GRACUR(X,Y,N)
      RETURN
      END
C
C-----------------------------------------------------
C
      SUBROUTINE NPLOT(DIN,N,Y,IST)
C
C THIS SUBROUTINE ALLOWS NCURVES TO BE PLOTTED
C ON ONE GRAPH
C
      DIMENSION Y(1025),DIN(10240)
      I1=(IST-1)*N
      DO 10 I=1,N
10    Y(I)=DIN(I1+I)
      RETURN
      END
C
C-----------------------------------------------------
C
      SUBROUTINE MAXVAL(BBEG,BEND,DIN,N,NGRF)
      DIMENSION DIN(10240)
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(1,20)BBEG,BEND
20    FORMAT('BBEG,BEND',2F12.4)
      RETURN
      END
```

Appendix A13 Spectral Analysis Programs

This appendix includes FORTRAN source codes for the two spectral analysis methods described in chapter 3. There are two main programs used in the spectral analysis, WELCH.FTN' and CORLTN.FTN. Each main program has several dependent subprograms located in the same file, and external library of subprograms. The purpose of the main program and subprograms is given as comments at the start of each program listing.

The data to be analysed is contained in a file, arranged in six columns or arrays, representing for example, VR, HR, EEG, EEGC, VL, HL, in raw forms or as preprocessed data.

For the Welch method, when the program is run it requests for the name of the data file and reads the data, the name of the graphics device to use, the window length and the window function to use, then for each of the first four data read the spectra is computed and plotted. Cross spectral analysis follows the same approach, except that the quantities computed and plotted are, spectra of the first (or the second) and third data columns, coherence, cross spectra, and gain functions between the two data columns.

For the correlation method, the procedure is essentially the same. In this case however, after the data to be analysed is read, the autocorrelation functions of each data is first computed and this is then used to compute the spectral estimates in the subroutine ASPEC after windowing The transform algorithm used here is due to Jenkins and Watts [1].

The minitab macro FREQPLOT.MTB reads the raw data (VL,VR HL, HR, EEG) from a data file' MTB.DATA,removes the mean from the samples, corrects the EEG for OA using OLS and differencing methods, and writes the mean-corrected and preprocessed data into separate data files,FREQ1.DATA and FREQ2.DATA. These files normally contain the data used by the programs WELCH.FTN and CORLTN.FTN.

Index to Programs

[1] JENKINS and WATTS

Spectral analysis and its Applications. Holden day 1968.

PROGRAM WELCH.FTN

```
C
C THIS PROGRAM IS USED TO STUDY THE POWER
C AND CROSS POWER SPECTRA OF THE EOG, THE EEG
C AND THE CORRECTED EEG
C THE COHERENCE SPECTRUM
C BETWEEN THE EOG AND THE EEG, AND THE SYSTEM GAIN
C
      COMMON /XVAL/X1(257)
      COMMON /MTDATA/ C0(1024),C1(1024),C2(1024),C3(1024)
     +,C4(1024),C5(1024)
      INTEGER*2 FNAME(20)
      N=1024
      FREQ=125.0
      DO 10 I=1,257
10    X1(I)=(FREQ*FLOAT(I))/FLOAT(N)
      CALL DEVICE(IDEV)
      CALL GETNAM(FNAME)
      CALL MTBRD(FNAME)
      WRITE(1,15)
15    FORMAT('ENTER 1 FOR DIFERENCED DATA 0 OTHERWISE')
      READ(1,*)IDIFF
      WRITE(1,20)
20    FORMAT('ENTER 0 FOR AUTOSPEC, 1 FOR CSPEC')
      READ(1,*)ISPEC
      IF(ISPEC .EQ. 0)CALL ASPEC(C0,C1,C2,C3,C4,C5,IDIFF)
      IF(ISPEC .EQ. 1)CALL CSPEC(C0,C1,C2,IDIFF)
      CALL DEVEND
      CALL EXIT
      END
C
C
C---------------------------------------------------------
C
      SUBROUTINE ASPEC(C0,C1,C2,C3,C4,C5,IDIFF)
C
C THIS SUB. CONTROLS THE COMPUTING OF THE
C SPECTRAL ESTIMATES OF THE DATA IN THE
C FIRST FOUR COLUMNS.
C
      DIMENSION W(513)
      DIMENSION C0(1024),C1(1024),C2(1024),C3(1024),
     +C4(1024),C5(1024),SUM(257)
      CALL VWNDOW(W,E,LSEG)
      NSEG=(2048/LSEG)-1
      CALL PSD(C0,SUM,W,E,LSEG,NSEG,0,IDIFF)
      CALL PSD(C1,SUM,W,E,LSEG,NSEG,1,IDIFF)
      CALL PSD(C2,SUM,W,E,LSEG,NSEG,2,IDIFF)
      CALL PSD(C3,SUM,W,E,LSEG,NSEG,3,IDIFF)
      RETURN
      END
C
C-----------------------------------------------------------
C
      SUBROUTINE PSD(DIN,SUM,W,E,LSEG,NSEG,ING,IDIFF)
C
C THIS SUB COMPUTES THE FINAL SPECTRAL ESTIMATES FOR
C EACH DATA COLUMN BY AVERAGING OVER K PERIODOGRAMS
C
      DIMENSION SUM(257),DIN(1024),DOUT(513),W(513)
      DIMENSION Y(257)
      COMMON /XVAL/ X1(257)
      NPT=65
      N=1024
      DO 5 IZ=1,NPT
5     SUM(IZ)=0.0
      DO 20 KSEG=1,NSEG
      CALL PSD1(DIN,KSEG,NPT,DOUT,W,LSEG)
      DO 30 IZ=1,NPT
30    SUM(IZ)=SUM(IZ)+DOUT(IZ)
20    CONTINUE
C
C COMPUTE LOG OF SPECTRA AND CORRECT FOR DIFFERENCING
C IF NECESSARY
C
      PI=4.0*ATAN(1.0)
      FREQ=125.0
      DO 40 K3=1,NPT
      IF(SUM(K3) .LE. 0.0)SUM(K3)=0.01
      XX=X1(K3)/FREQ
      XY=4.0*SIN(PI*XX)*SIN(PI*XX)
      Y(K3)=ALOG10(SUM(K3))-ALOG10(NSEG*E)
40    IF(IDIFF .EQ. 1)Y(K3)=Y(K3)-ALOG10(XY)
      CALL MINVAL(Y,NPT)
      CALL FGRAPH(NPT,X1,Y,ING,0)
      RETURN
      END
```

```
C----------------------------------------------------------
C
C THIS SUBROUTINE COMPUTES THE PERIODOGRAM
C FOR THE Kth SEGMENT AFTER WINDOWING AND
C ZERO PADDING
C
      SUBROUTINE PSD1(DIN,KSEG,NPT,DOUT,W,LSEG)
      DIMENSION DIN(1024),DOUT(513),W(513)
      COMPLEX X(1024)
      ND=LSEG/2
      N=1024
      KM1=KSEG-1
      DO 10 I=1,LSEG
      I1=I+KM1*ND
10    X(I)=CMPLX(DIN(I1)*W(I),0.0)
C
C ZERO PAD THE WINDOWED DATA
C
      LP1=LSEG+1
      DO 20 I=LP1,N
20    X(I)=CMPLX(0.0,0.0)
      CALL FFT(X,N,0)
      DO 30 K=1,NPT
30    DOUT(K)=CABS(X(K))*CABS(X(K))
      RETURN
      END
C
C
C-------------------------------------------------
C
      SUBROUTINE CSPEC(C0,C1,C2,IDIFF)
      DIMENSION C0(1024),C1(1024),C2(1024),W(513)
      CALL VWNDOW(W,E,LSEG)
      NSEG=(2048/LSEG)-1
      CALL CSD(C0,C2,W,E,LSEG,NSEG,0,IDIFF)
      CALL PICCLE
      CALL CSD(C1,C2,W,E,LSEG,NSEG,1,IDIFF)
      RETURN
      END
C
C-------------------------------------------------
C
      SUBROUTINE CSD(C1,C2,W,E,LSEG,NSEG,ING,IDIFF)
C
C THIS SUBROUTINE COMPUTES THE COHERENCY SPECTRUM
C AND THE SYSTEM GAIN
C
      DIMENSION C1(1024),C2(1024),W(513),XSPEC(257),
     +YSPEC(257),COHSQ(257),GAIN(257),SQ(257)
      COMPLEX XYSPEC(257)
      COMMON /XVAL/ X1(257)
      NPT=65
      DO 5 K=1,NPT
      COHSQ(K)=0.0
5     GAIN(K)=0.0
      CALL PSD(C1,XSPEC,W,E,LSEG,NSEG,0,IDIFF)
      CALL PSD(C2,YSPEC,W,E,LSEG,NSEG,1,IDIFF)
      CALL CSD1(C1,C2,W,E,LSEG,NSEG,XYSPEC)
      PI=4.0*ATAN(1.0)
      FREQ=125.0
      DO 10 K=1,NPT
      SQ(K)=CABS(XYSPEC(K))
      SS=XSPEC(K)*YSPEC(K)
      IF(SS .GT. 0.0)COHSQ(K)=SQ(K)*SQ(K)/SS
      IF(XSPEC(K) .GT. 0.0)GAIN(K)=10.0*ALOG10(SQ(K)/XSPEC(K))
      IF(SQ(K) .GT. 0.0)SQ(K)=10.0*ALOG10(SQ(K))
      XX=X1(K)/FREQ
      XY=4.0*SIN(PI*XX)*SIN(PI*XX)
      IF(IDIFF .EQ. 1)SQ(K)=SQ(K)-ALOG10(XY)
10    CONTINUE
      CALL FGRAPH(NPT,X1,COHSQ,2,2)
      CALL FGRAPH(NPT,X1,GAIN,3,3)
      CALL FGRAPH(NPT,X1,SQ,4,1)
      RETURN
      END
```

```
C
C------------------------------------------------------
C
      SUBROUTINE CSD1(C1,C2,W,E,LSEG,NSEG,XYSPEC)
C
C THIS SUB. COMPUTES THE CROSS- PERIODOGRAM
C FOR THE KTH SEGMENT AFTER WINDOWING AND
C ZERO PADDING
C
      DIMENSION C1(1024),C2(1024),W(513)
      COMPLEX DOUT(257),XYSPEC(257)
      NPT=65
      N=1024
      DO 5 IZ=1,NPT
5     XYSPEC(IZ)=CMPLX(0.0,0.0)
      DO 20 KSEG=1,NSEG
      CALL CSD2(C1,C2,KSEG,DOUT,W,LSEG)
      DO 30 IZ=1,NPT
30    XYSPEC(IZ)=XYSPEC(IZ)+DOUT(IZ)
20    CONTINUE
      RETURN
      END
C
C-------------------------------------------------
C
      SUBROUTINE CSD2(C1,C2,KSEG,DOUT,W,LSEG)
C
C THIS SUB COMPUTES THE UNSCALED CROSS
C SPECTRUM FOR THE KTH SEGMENT AFTER WINDOWING
C AND ZERO PADDING.
C
      DIMENSION C1(1024),C2(1024),W(513)
      COMPLEX X(1024),Y(1024),DOUT(257)
      NPT=65
      N=1024
      ND=LSEG/2
      KM1=KSEG-1
      DO 10 I=1,LSEG
      I1=I+KM1*ND
      X(I)=CMPLX(C1(I1)*W(I),0.0)
10    Y(I)=CMPLX(C2(I)*W(I),0.0)
      LP1=LSEG+1
      DO 20 I=LP1,N
      X(I)=CMPLX(0,0,0.0)
20    Y(I)=CMPLX(0.0,0.0)
      CALL FFT(X,N,0)
      CALL FFT(Y,N,0)
      DO 30 K=1,NPT
30    DOUT(K)=X(K)*CONJG(Y(K))
      RETURN
      END
C
C-----------------------------------------------------
C
      SUBROUTINE VWNDOW(W,E,LSEG)
C
C THIS SUB ALLOWS THE SELECTION OF THE WINDOW
C TO BE USED IN THE COMPUTATION OF THE
C SPECTRAL ESTIMATES
C
      DIMENSION W(513)
      WRITE(1,10)
10    FORMAT('ENTER SEGMENT LENGTH')
      READ(1,*)LSEG
      IF(LSEG .GT. 513)GO TO 99
      WRITE(1,20)
20    FORMAT('ENTER 0 FOR HANNING 1 FOR BARTLET 2 FOR PARZEN')
      READ(1,*)IW
      IF(IW .EQ. 0)CALL HANNG(W,E,LSEG)
      IF(IW .EQ. 1)CALL BARTLT(W,E,LSEG)
      IF(IW .EQ. 2)CALL PARZEN(W,E,LSEG)
      RETURN
99    WRITE(1,100)
100   FORMAT('SEGMENT LENGTH TOO LONG')
      STOP
      END
```

```
C
C-----------------------------------------
C
      SUBROUTINE BARTLT(W,E,LSEG)
C
C
C GIVEN THE SEGMENT LENGTH, LSEG, THIS SUB. COMPUTES
C THE COEFFICIENTS FOR THE BARTLET WINDOW FUNCTION
C
      DIMENSION W(513)
      LSEG2=LSEG/2
      LSEG=2*LSEG2+1
      E=0.0
      DO 10 K=1,LSEG
      K1=K-1
      X1=FLOAT(K1-LSEG2)
      W(K)=1.0-2.0*ABS(X1)/FLOAT(LSEG-1)
      E=E+W(K)*W(K)
10    CONTINUE
      RETURN
      END
C
C-----------------------------------------
C
      SUBROUTINE HANNG(W,E,LSEG)
C
C GIVEN THE SEGMENT LENGTH, LSEG, THIS SUB COMPUTES
C THE COEFFICIENTS FOR THE HANNING WINDOW FUNCTION
C
      DIMENSION W(513)
      E=0.0
      LSEG2=LSEG/2
      LSEG=2*LSEG2+1
      PI=4.0*ATAN(1.0)
      DO 10 K=1,LSEG
      K1=K-1
      X1=2.0*FLOAT(K1-LSEG2)
      T2=PI*X1/FLOAT(LSEG-1)
      W(K)=0.5+0.5*COS(T2)
      E=E+W(K)*W(K)
10    CONTINUE
      RETURN
      END
C
C-----------------------------------------------------
C
      SUBROUTINE PARZEN(W,E,LSEG)
C
C GIVEN THE SEGMENT LENGTH, LSEG, THIS SUB COMPUTES
C THE COEFFICIENTS FOR THE PARZEN WINDOW FUNCTION
C
      DIMENSION W(513)
      LSEG2=LSEG/2
      LSEG4=LSEG/4
      LSEG=2*LSEG2+1
      LSEG21=LSEG2+1
      LSEG41=LSEG4+1
      DO 10 K=1,LSEG2
      T1=FLOAT(2*K)/FLOAT(LSEG-1)
      T2=1.0-T1
      IF(K .LT. LSEG41)W(LSEG21+K)=1.0-(6.0*T1*T1*T2)
      IF(K .GE. LSEG41)W(LSEG21+K)=2.0*T2*T2*T2
      W(LSEG21-K)=W(LSEG21+K)
10    CONTINUE
      W(LSEG21)=1.0
      E=0.0
      DO 20 K=1,LSEG
      E=E+W(K)*W(K)
20    CONTINUE
      RETURN
      END
```

PROGRAM CORLTN.FTN

```
C
C THIS PROGRAM IS USED TO STUDY THE POWER
C SPECTRA OF THE EOG, THE EEG AND THE
C CORRECTED EEG BY THE CORRELATION METHOD
C
      COMMON /XVAL/X1(257)
      COMMON /MTDATA/ C0(1024),C1(1024),C2(1024),C3(1024)
     +,C4(1024),C5(1024)
      INTEGER FNAME(20)
      N=1024
      FREQ=125.0
      NF=N-1
      MAXM=511
      DO 10 I=1,257
10    X1(I)=FREQ*FLOAT(I)/(2.0*FLOAT(NF))
      CALL DEVICE(IDEV)
      CALL GETNAM(FNAME)
      CALL MTBRD(FNAME)
      WRITE(1,30)
30    FORMAT('ENTER 1 FOR DIFFERENCED DATA 0 OTHERWISE')
      READ(1,*)IDIFF
      CALL ACFN(C0,C1,C2,C3,C4,C5,IDIFF,NF,MAXM)
      CALL DEVEND
      CALL EXIT
      END
C
C-----------------------------------------------------
C
      SUBROUTINE ACFN(C0,C1,C2,C3,C4,C5,IDIFF,NF,MAXM)
C
C THIS SUB IS USED TO CONTROL THE COMPUTATION AND
C PLOTTING OF THE SPECTRA
C
      DIMENSION C0(1024),C1(1024),C2(1024),C3(1024)
     +,C4(1024),C5(1024)
      DIMENSION SPEC0(1024),SPEC1(1024),SPEC2(1024)
     +,SPEC3(1024),SPEC4(1024),SPEC5(1024)
      CALL SPECOM(C0,SPEC0,MAXM,NF,0,IDIFF)
      CALL SPECOM(C1,SPEC1,MAXM,NF,1,IDIFF)
      CALL SPECOM(C2,SPEC2,MAXM,NF,2,IDIFF)
      CALL SPECOM(C3,SPEC3,MAXM,NF,3,IDIFF)
      CALL SPECOM(C4,SPEC4,MAXM,NF,4,IDIFF)
      CALL SPECOM(C5,SPEC5,MAXM,NF,5,IDIFF)
      RETURN
      END
C
C-------------------------------------------------------
C
      SUBROUTINE SPECOM(DIN,SPEC,MAXM,NF,ING,IDIFF)
      DIMENSION DIN(1024),ACFD(1024),SPEC(1024),Y(257)
      COMMON /XVAL/X1(257)
      PI=4.0*ATAN(1.0)
      FREQ=125.0
      NCORL=513
      NPT=129
      CALL ACF(NCORL,DIN,ACFD)
      CALL ASPEC(ACFD,MAXM,NF,SPEC,M)
      DO 10 K=1,NPT
      IF(SPEC(K) .LE. 0.0)SPEC(K)=0.1
      XX=X1(K)/FREQ
      XY=4.0*SIN(PI*XX)*SIN(PI*XX)
      Y(K)=ALOG10(SPEC(K))
10     IF(IDIFF .EQ. 1)Y(K)=Y(K)-ALOG10(XY)
      CALL MINVAL(Y,NPT)
      CALL FGRAPH(NPT,X1,Y,ING,0)
      RETURN
      END
```

```
C
C-------------------------------------------------------
C
      SUBROUTINE ASPEC(COV,MAXM,NF,SPEC,M)
C
C THIS ROUTINE COMPUTES THE SMOOTHED AUTOSPECTRUM OF THE
C DATA WHOSE COVARIANCE IS GIVEN IN 'COV'
C
C    INPUTS: COV       AUTOCOVARIANCE OF SERIES TO BE TRANSFORMED
C           MAXM       MAXIMUM NUMBER OF LAGS IN COV
C          NF         NUMBER OF FREQ. POINTS DESIRED
C
C     OUTPUT:         SPEC  SMOOTHED AUTOSPECTRAL ESTIMATES
C                 M      TRUNCATION POINT FOR SPECTRAL WINDOW
C
      DIMENSION COV(513),SPEC(1024),W(513)
      PI=4.0*ATAN(1.0)
      CALL VWNDOW(M,W)
      MP2=M+2
C
C     CALCULATE SMOOTHED AUTOSPECTRAL ESTIMATES
C
      WRITE(1,11)M,NF
11    FORMAT('ASPECM,NF:',2I6)
      DO 2 I=1,NF
      IM1=I-1
      U1=FLOAT(IM1)/FLOAT(NF)
      V0=0.0
      V1=0.0
      C=COS(PI*U1)
      DO 1 K=2,M
      K1=MP2-K
      V2=2.0*C*V1-V0+COV(K1)*W(K1)
      V0=V1
1     V1=V2
2     SPEC(I)=2.0*(COV(1)+2.0*(V1*C-V0))
      RETURN
      END
C
C-------------------------------------------------------
C
      SUBROUTINE TUKEY(M,W)
      DIMENSION W(513)
      PI=4.0*ATAN(1.0)
      DO 10 K=1,M
      K1=K-1
      T1=PI*FLOAT(K1)/FLOAT(M)
      T2=COS(T1)
10    W(K)=0.5*(1.0+T2)
      RETURN
      END
C
C-------------------------------------------------------
C
      SUBROUTINE BARTLT(M,W)
      DIMENSION W(513)
      WRITE(1,2)M
2     FORMAT('BARTLET,M=',I5)
      DO 10 K=1,M
      K1=K-1
      T1=FLOAT(K1)/FLOAT(M)
10    W(K)=1.0-T1
      RETURN
      END
C
C-------------------------------------------------------
C
      SUBROUTINE PARZEN(M,W)
      DIMENSION W(513)
      M2=M/2
      M3=M2+1
      DO 10 K=1,M2
      K1=K-1
      T1=FLOAT(K1)/FLOAT(M)
      T2=T1*T1
      T3=T1*T2
      W(K)=1.0-6.0*T2+6.0*T3
10    CONTINUE
      DO 20 K=M3,M
      K1=K-1
      T1=1.0-FLOAT(K1)/FLOAT(M)
20    W(K)=T1*T1*T1*2.0
      RETURN
      END
```

```
C
C-----------------------------------------------------------
C
      SUBROUTINE CSPEC(EV,OD,NF,M,COSPEC,QSPEC,SQ,MAXM)
      DIMENSION EV(513),OD(513),COSPEC(1024)
      DIMENSION QSPEC(1024),SQ(1024),W(513)
C
C THIS SUBROUTINE COMPUTES THE SMOOTHED CROSS SPECTRAL
C ESTIMATES FROM THE EVN AND ODD LAG CROSS CORRELATION ESTIMATES
C
C INPUTS:
C       EV    EVEN LAG CROSS CORRELATION ESTIMATES
C       OD    ODD  "  "
C       M,MAXM,NF
C
C OUTPUTS:
C      COSPEC
C      QSPEC   QUADRATURE COMPONENTS OF THE SPECTRAL ESTIMATES
C      SQ      SQUARED MAGNITUDE SPECTRAL ESTIMATES
C
      CALL VWNDOW(M,W)
      PI=4.0*ATAN(1.0)
      MP2=M+2
      DO 2 I=1,NF
      IM1=I-1
      V0=0.0
      V1=0.0
      Z0=0.0
      Z1=0.0
      C=COS((PI*IM1)/NF)
      SN=SIN((PI*IM1)/NF)
      DO 1 K=2,M
      K1=MP2-K
      V2=2.0*C*V1-V0+EV(K1)*W(K1)
      Z2=2.0*C*Z1-Z0+OD(K1)*W(K1)
      V0=V1
      V1=V2
      Z0=Z1
1     Z1=Z2
      COSPEC(I)=2.0*(EV(1)+2.0*(V1*C-V0))
      QSPEC(I)=4.0*Z1*SN
2     SQ(I)=COSPEC(I)*COSPEC(I)+QSPEC(I)*QSPEC(I)
      RETURN
      END
C
C-------------------------------------------------------------------
C
      SUBROUTINE MLTCOR(COV,VR,HR,EEG,MAXM,ICOV)
C
C THIS PROGRAM CALCULATES AUTOC AND CROSS CAVARIANCES
C OF THE EOGS AND EEG.
C
C     INPUTS: VR,HR
C             EEG1        (    "       "        )
C             ICOV      =0  PROGRAM COMPUTES COVARIANCES
C                       =1  PROGRAM COMPUTES CORRELATIONS
C     OUTPUT: COV           COVARIANCES
C             COR           CORRELATIONS
      DIMENSION X(1024,3),COV(513,3,3),COR(513,3,3)
      DIMENSION VR(1024),HR(1024),EEG(1024)
      DOUBLE PRECISION SUM
      INTEGER*2 FNAME(20)
      N=1024
      NCORL=MAXM+2
      NS=3
      DO 10 I=1,N
      X(I,1)=VR(I)
      X(I,2)=HR(I)
      X(I,3)=EEG(I)
10    CONTINUE
      WRITE(1,110)
110   FORMAT('XVAL OK')
      DO 30 L=1,NS
      DO 30 J=1,NS
      DO 30 K=1,NCORL
      K1=K-1
      SUM=0.0
      NZ1=N-K1
      DO 20 I1=1,NZ1
20    SUM=SUM+X(I1,J)*X(I1+K1,L)
30    COV(K,J,L)=SUM/FLOAT(N)
      IF(ICOV .EQ. 0)RETURN
      DO 50 L=1,NS
      DO 50 J=1,NS
      DO 50 K=1,NCORL
      DO 50 K=1,NCORL
50    COR(K,J,L)=COV(K,J,L)/SQRT(COV(1,J,J)*COV(1,L,L))
      RETURN
      END
C
C-----------------------------------------------------------------
C
      SUBROUTINE VWNDOW(M,W)
      DIMENSION W(513)
      WRITE(1,10)
10    FORMAT('ENTER M')
      READ(1,*)M
      IF(M .GE. 513)GO TO 99
      WRITE(1,20)
20    FORMAT('ENTER 0 FOR TUKEY,1 FOR BARTLET,2 FOR PARZEN')
      READ(1,*)IW
      IF(IW .EQ. 0)CALL TUKEY(M,W)
      IF(IW .EQ. 1)CALL BARTLT(M,W)
      IF(IW .EQ. 2)CALL PARZEN(M,W)
      RETURN
99    WRITE(1,100)
100   FORMAT('M IS TOO LARGE')
      STOP
      END
```

SUBPRO SUBROUTINE FFT.FTN

```
      SUBROUTINE FFT(X,N,INV)
C THIS SUBPROGRAM IMPLEMENTS THE FFT ALGORITHM TO
C COMPUTE THE DISCRETE FOURIER COEFFICIENTS OF A
C DATA SEQUENCE OF N POINTS.
C
C CALLING SEQUENCE FROM THE MAIN PROGRAM:
C CALL FFT(X,N,INV)
C          N      NUMBER OF DATA POINTS
C          X      COMPLEX ARRAY CONTAINING THE DATA
C                 SEQUENCE. THE DFT COEFFS. ARE RETURNED IN
C                 THE SAME ARRAY.
C          INV    FLAG FOR DIRECTION OF TRANSFORM
C                 INV=0 FOR FORWARD TRANSFORM
C                 INV=1 FOR INVERSE TRANSFORM
C
      COMPLEX X(N),W,T,CMPLX
C
C     CALCULATE THE NO. OF ITERATIONS(LOG.N TO BASE 2)
C
      ITER=0
      IREM=N
   10 IREM=IREM/2
      IF(IREM .EQ. 0)GO TO 20
      ITER=ITER+1
      GO TO 10
   20 CONTINUE
      SIGN=-1.0
      IF(INV .EQ. 1)SIGN=1.0
      NXP2=N
      DO 50 IT=1,ITER
C
C     COMPUTATION FOR EACH ITERATION
C     NXP:NUMBER OF POINTS IN PARTITION
C     MSP2:NXP/2
C
      NXP=NXP2
      NXP2=NXP/2
      WPWR=3.141592/FLOAT(NXP2)
      DO 40 M=1,NXP2
C
C     CALCULATE THE MULTIPLIER
C
      ARG=FLOAT(M-1)*WPWR
      W=CMPLX(COS(ARG),SIGN*SIN(ARG))
      DO 40 MXP=NXP,N,NXP
C
C     COMPUTATION FOR EACH PARTITION
C
      J1=MXP-NXP+M
      J2= J1+NXP2
      T=X(J1)-X(J2)
      X(J1)=X(J1)+X(J2)
   40 X(J2)=T*W
   50 CONTINUE
C
C     UNSCRAMBLE THE BIT-REVERSED DFT COEFFS.
C
      N2=N/2
      N1=N-1
      J=1
      DO 65 I=1,N1
      IF(I .GE. J)GO TO 55
      T=X(J)
      X(J)=X(I)
      X(I)=T
   55 K=N2
   60 IF(K .GE. J)GO TO 65
      J=J-K
      K=K/2
      GO TO 60
   65 J=J+K
      IF(INV .EQ. 1)GO TO 75
      DO 70 I=1,N
   70 X(I)=X(I)/FLOAT(N)
   75 CONTINUE
      RETURN
      END
```

SUBPROGRAM FGRAPH.FTN

```
C
C THIS ROUTINE PLOTS  A GRAPH ON  ONE  OF SIX
C PARTS OF A PAGE.   THUS  UPTO SIX GRAPGS  C AN BE
C PLOTTED ON ONE PAGE  IF CALLED  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
      SUBROUTINE FGRAPH(N,X,Y,ING,IST)
      DIMENSION X(257),Y(257)
      INTEGER*2 ILAB1(7),ILAB2(7)
      INTEGER*2 ILAB3(5),ILAB4(2)
      DATA ILAB1,ILAB2/'Log power(PSD)','Log power(CSD)'/
      DATA ILAB3,ILAB4/'coherence ','Gain'/
      X1=0.
      IF(MOD(ING,2) .EQ. 1)X1=110.0
      Y1=57.0
      IF(ING .LT. 2)Y1=115.
      IF(ING .GT. 3)Y1=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=X2-60.0
      AX2=X1+10.0
      AY1=Y1+5.0
      AY2=Y1+20.0
      CALL MOVTO2(AX1,AY1)
      CALL CHAHOL('Freq(Hz)*.')
      CALL CHAANG(90.0)
      CALL MOVTO2(AX2,AY2)
      IF(IST .EQ. 0)CALL CHAARR(ILAB1,7,2)
      IF(IST .EQ. 1)CALL CHAARR(ILAB2,7,2)
      IF(IST .EQ. 2)CALL CHAARR(ILAB3,5,2)
      IF(IST .EQ. 3)CALL CHAARR(ILAB4,2,2)
      CALL CHAANG(0.0)
      RETURN
      END
C
C-------------------------------------------------------
C
      SUBROUTINE DEVICE(IDEV)
C
C THIS SUBROUTINE NOMINATES THE  GRAPHICS  DEVICE
C TO  BE  USED  FOR  PLOTTING GRAPHS
C
      WRITE(1,10)
10    FORMAT('ENTER 1 TO VIEW, 2 FOR CC906, 3 FOR CC81')
      READ(1,*)IDEV
      IF(IDEV .EQ. 1)CALL T4010
      IF(IDEV .EQ. 2)CALL CC906
      IF(IDEV .EQ. 3)CALL CC81
      RETURN
      END
C
C------------------------------------------------------
C
      SUBROUTINE MINVAL(DIN,NPT)
C
C SUBROUTINE TO NORMALIZE THE SPECTRAL ESTIMATES
C
      DIMENSION DIN(257)
      PMIN=DIN(1)
      DO 10 I=2,NPT
      IF(DIN(I) .LT. PMIN)PMIN=DIN(I)
10    CONTINUE
      DO 20 I=1,NPT
      DIN(I)=DIN(I)-PMIN
20    CONTINUE
      RETURN
      END
```

```
C
C--------------------------------------------------------
C
      SUBROUTINE GETNAM(FNAME)
C
C THIS SUBROUTINE REQUESTS FOR AND READS THE NAME
C OF THE DATA FILE TO BE PROCESSED
C
      INTEGER*2 FNAME(20),TBUFF(20)
      DATA TBUFF /'NONAMEGIVEN                             '/
      WRITE(1,8)
8     FORMAT('GIVE NAME OF FILE TO BE PROCESSED')
      READ(1,10)FNAME
10    FORMAT(20A2)
      IF(FNAME(1) .EQ. '  ')GO TO 100
      DO 20 I=1,20
20    TBUFF(I)=FNAME(I)
      RETURN
100   DO 120 I=1,20
120   FNAME(I)=TBUFF(I)
      RETURN
      END
C
C------------------------------------------------------
C
      SUBROUTINE MTBRD(FNAME)
C
C THIS SUBROUTINE READS THE DATA FROM THE SPECIFIED
C DATA FILE
C
$INSERT SYSCOM|A$KEYS
      COMMON /MTDATA/C0(1024),C1(1024),C2(1024),C3(1024)
     +,C4(1024),C5(1024)
      INTEGER*2 FNAME(20),RKEY
      LOGICAL OPEN,CLOSE
      RKEY=3
      NLEN=40
      OPEN=OPEN$A(A$READ,FNAME,NLEN,RKEY)
      IF(.NOT. OPEN)GO TO 1700
      READ(7,*)(C0(I),C1(I),C2(I),C3(I),C4(I),C5(I),I=1,1024)
      CLOSE=CLOS$A(RKEY)
      IF(CLOSE)RETURN
      WRITE(1,30)FNAME
30    FORMAT('*** CANT CLOSE FILE',20A2,'***')
      STOP 1
1700  WRITE(1,35)FNAME
35    FORMAT('*** CANT OPEN FILE ',20A2,'***')
      STOP 2
      END
```

PROGRAM FREQPLOT.MTB

```
#PROGRAM USED TO PREPROCESS DATA FOR THE
#SPECTRAL ANALYSIS
#
#COMPUTE AND REMOVE MEAN FROM DATA
READ 'MTB.DATA' C1-C5
MEAN C1 K1
MEAN C2 K2
MEAN C3 K3
MEAN C4 K4
MEAN C5 K5
SUBTRACT K1 C1 C1
SUBTRACT K2 C2 C2
SUBTRACT K3 C3 C3
SUBTRACT K4 C4 C4
SUBTRACT K5 C5 C5
NAME C1='VL',C2='VR',C3='HL',C4='HR',C5='EEG'
NAME C11='VRD',C12='HLD',C13='HRD',C14='EEGD'
#
#DIFFERENCE THE DATA
#
DIFF 'VR','VRD'
DIFF 'HL','HLD'
DIFF 'HR','HRD'
DIFF 'EEG','EEGD'
#
#PERFORM OLS FOR BOTH RAW AND DIFFERENCED DATA
#
BRIEF 2
REGRESS 'EEG' 3 'VR','HL','HR' C96;
NOCONSTANT;
MSE K1;
COEFF C24;
XPXINV M1.
LET C61=C24(1)*'VR'+C24(2)*'HL'+C24(3)*'HR'
LET C81='EEG'-C61
PRINT C24
BRIEF 2
REGRESS 'EEGD' 3 'VRD','HLD','HRD' C95;
NOCONSTANT;
MSE K1;
COEFF C24;
XPXINV M1.
LET C62=C24(1)*'VR'+C24(2)*'HL'+C24(3)*'HR'
LET C82='EEG'-C62
PRINT C24
LET 'VRD'(1)=0.0
LET 'HRD'(1)=0.0
LET 'EEGD'(1)=0.0
LET C82(1)=0.0
LET C95(1)=0.0
#
#SAVE PREPROCESSED DATA FOR USE IN SPECTRAL ANALYSIS
#
WRITE 'FREQTEST1.DATA' 'VR','HR','EEG',C81,'VR','HR'
WRITE 'FREQTEST2.DATA' 'VRD','HRD','EEGD',C82,C95,'HRD'
STOP
```

Appendix A14 Digital Filtering Programs

A main program, EOGFILTER, is used to filter the EOG noise. The data to be anlyzed is contained in a data file, arranged in six columns representing VL, VR, HL, HR, EEG1, EEG2. When program is run, it requests for the name of the data file, the name of the file containing the filter length and coefficients. The coefficients are then used to filter out the EOG noise, and the filtered and unfiltered data are written to another file for further processing (eg plotting).

```
        PROGRAM  EOGFILTER.FTN
C
C THIS PROGRAM IS USED TO REALISE THE EOG FILTR
C
      DIMENSION VR(1024),HR(1024)
      COMMON /MTDATA/ C0(1024),C1(1024),C2(1024),C3(1024)
     +,C4(1024)
      COMMON /HBDATA/HB(40)
      INTEGER*2 FNAME(20),FCOEF(20)
      ITYPE1=1
      ITYPE2=2
      NPT=1024
C
C GET NAMES OF DATA FILE, COEFF. FILE, AND READ COEFFS.
C AND DATA
C
      CALL GETNAM(FNAME,ITYPE1)
      CALL MTBRD(FNAME,ITYPE1,N)
      CALL GETNAM(FNAME,ITYPE2)
      CALL MTBRD(FNAME,ITYPE2,N)
C
C SAVE VR AND HR, THEN FILTER THE EOGS
C
      DO 5 I=1,NPT
      VR(I)=C0(I)
5     HR(I)=C1(I)
      CALL FILTR(N)
      WRITE(6,10)(VR(I),C0(I),HR(I),C1(I),C2(I),C3(I),
     +I=1,NPT)
10    FORMAT(6(F12.5,2X))
      STOP
      END
C
C-----------------------------------------------------
C
      SUBROUTINE GETNAM(FNAME,ITYPE)
      INTEGER*2 FNAME(20),TBUFF(20)
      DATA TBUFF /'NONAMEGIVEN                             '/
      IF(ITYPE .EQ. 1)WRITE(1,8)
      IF(ITYPE .EQ. 2) WRITE(1,12)
8     FORMAT('GIVE NAME OF FILE TO BE PROCESSED')
12    FORMAT('GIVE NAME OF FILE FOR COEFF')
      READ(1,10)FNAME
10    FORMAT(20A2)
      IF(FNAME(1) .EQ. '  ')GO TO 100
      DO 20 I=1,20
20    TBUFF(I)=FNAME(I)
      RETURN
100   DO 120 I=1,20
120   FNAME(I)=TBUFF(I)
      RETURN
      END
C
C-----------------------------------------------------
C
      SUBROUTINE MTBRD(FNAME,ITYPE,N)
$INSERT SYSCOM|A$KEYS
      COMMON /MTDATA/C0(1024),C1(1024),C2(1024),C3(1024)
     +,C4(1024)
      COMMON /HBDATA/HB(40)
      INTEGER*2 FNAME(20),RKEY
      LOGICAL OPEN,CLOSE
      RKEY=3
      NLEN=40
      OPEN=OPEN$A(A$READ,FNAME,NLEN,RKEY)
      IF(.NOT. OPEN)GO TO 1700
      IF(ITYPE .EQ. 2)READ(7,*)N
      K=(N+1)/2
      IF(ITYPE .EQ. 2)READ(7,*)(HB(I),I=1,K)
      IF(ITYPE .EQ. 1)READ(7,*)(C0(I),C1(I),C2(I),
     +C3(I),C4(I),I=1,1024)
      CLOSE=CLOS$A(RKEY)
      IF(CLOSE)RETURN
      WRITE(1,30)FNAME
30    FORMAT('*** CANT CLOSE FILE',20A2,'***')
      STOP 1
1700  WRITE(1,35)FNAME
35    FORMAT('*** CANT OPEN FILE ',20A2,'***')
      STOP 2
      END
```

```
C
C-----------------------------------------------------
C
      SUBROUTINE FILTR(N)
      COMMON /MTDATA/ VR(1024),HR(1024),C2(1024),C3(1024)
     +,C4(1024)
      COMMON /HBDATA/HB(40)
      CALL FILTER(VR,HB,N)
      CALL FILTER(HR,HB,N)
      RETURN
      END
C
C--------------------------------------------------------
C
C     SUBROUTINE TO CALL FILTER ALGORITHM
C     GIVEN AN ARRAY OF DATA TO BE FILTERED (1024 SAMPLES)
C     A FILTERED ARRAY OF THE SAME SIZE WITH
C     ZERO PHASE SHIFT IS RETURNED
C
      SUBROUTINE FILTER(DAT,H,N)
      DIMENSION DAT(1024),DATEX(1064),H(40)
      N2=2*N
      NS=(N+1)/2
C
C     SET UP EXTENDED DATA ARRAY
C
      DO 10 I=1,1024
   10 DATEX(I)=DAT(I)
      M=1024+NS-1
      DO 20 I=1025,M
   20 DATEX(I)=0.
C
C     INITIALISE FILTER
C
      CALL FILT(1,X,Y,H,N,N2,NS)
C
C     CALL FILTER ALGORITHM FOR EACH DATA POINT
C     REMOVE DELAY AND STORE RESULTING 1024 SAMPLES
C
      DO 30 I=1,M
      CALL FILT(0,DATEX(I),Y,H,N,N2,NS)
      IF(I.LT.NS) GO TO 30
      IEX=I-NS+1
      DAT(IEX)=Y
   30 CONTINUE
      RETURN
      END
C
C-----------------------------------------------------------
C
C     FILTER ALGORITHM BASED ON RABINER
C     UTILISING IMPULSE RESPONSE SYMMETRY
C
      SUBROUTINE FILT(INIT,X,Y,H,N,N2,NS)
      DIMENSION H(40),XSAV(100)
      IF(INIT.EQ.0) GO TO 50
      DO 10 I=1,N2
   10 XSAV(I)=0.
      IPT=N+1
      RETURN
   50 XSAV(IPT)=X
      XSAV(IPT-N)=X
      Y=0.
      M=NS-1
      DO 60 I=1,M
   60 Y=Y+H(I)*(XSAV(IPT-I+1)+XSAV(IPT-N+I))
      Y=Y+H(NS)*XSAV(IPT-NS+1)
      IPT=IPT+1
      IF(IPT.GT.N2) IPT=N+1
      RETURN
      END
```

Appendix A15 68000 μP Assembly language Program for the OAR System

There are essentially two different OAR system softwares, namely, one using the UD algorithm and another the fixed parameter algorithm. Both share a number of subroutines, since most of their operations are similar. Infact, the fixed parameter algorithm is obtained by a trivial modification to the full on-line algorithm that uses the UD algorithm.

The UD algorithm consists of a main program called UDMAIN.ASM and several subroutines; one for acquiring data, OADATA; one for outputing data, OUTPUT, another for updating the UD elements, UDFLT etc. These programs have been described in detail in the main text (chapter 6).

The fixed parameter algorithm consists of only two specific routines, viz, FXDMAIN (the main program, which is very similar to UDMAIN),and FXDCRTN analogous to UDCRTN. As examination of the program listing will show (p A149) the fixed parameter algorithm has only a few lines of codes specific to it. However, the algorithm makes use of a number of subroutines that belong to the UD on-line algorithm. These include OADATA for acquiring data, the interrupt service routine, the output routines and the floating point arithmetic routines.

OAR System Default Values

As stated in the main text, when the OAR system programs are run, they request for certain system constants, viz, the number of parameter estimates, NPAR, the number of EEG channels to correct for OA, NEEG, etc. The following are the ranges within which the constants should be. The lower limits are used as defaults.

$$
\begin{aligned}
2 &\leq \mathrm{NPAR} \leq 4 \\
5 &\leq \mathrm{NCHN} \leq 20 \\
1 &\leq \mathrm{NEEG} \leq (\mathrm{NCHN-NEOG}) \\
&\ \mathrm{NEOG} = 4 \\
1 &\leq \mathrm{EEGC} \leq \mathrm{NEEG} \\
1 &\leq \mathrm{EEGM} \leq (\mathrm{NCHN-EEGC}) \\
0 &\leq \mathrm{UEOG} \leq (\mathrm{NCHN-EEGC-EEGM}) \\
0 &= \mathrm{UPAR} \leq (\mathrm{NCHN-EEGC-EEGM-NEOG})
\end{aligned}
$$

where NPAR is the number of parameter estimates, which effectively determines the models to be used (see model definitions in chapter 6); NCHN the number of channels to acquire data from (NCHN is determined internally from the values of NEEG and NEOG). NEOG is the number of EOGs (this is fixed at 4); NEEG is the number of EEG channels from which OA is to be removed; EEGC is the number of corrected EEGs to output to the EEG machine; EEGM is the number of measured EEGs to output to the EEG machine; UEOG is the number of EOGs to output to the EEG machines; UPAR is the number of parameter estimates to output to the EEG machine. The philosophy behind the default values is that the number of EEG channels to correct should not exceed the total number of channels less the EOGs nor should the total number of variables to output to the EEG machine exceed the total number of channels available.

Changing the Sampling frequency

To change the sampling frequency, all that needs to be done is to change the programmable timer module's counts. The instruction 'MOVE.W $7C31,D3' at the present gives a sampling frequency of 128Hz. The following table gives the values to use to give different sampling frequencies.

COUNT	FREQ
$7C31	128Hz
$9F31	100Hz
$A731	95Hz

The instruction given above is at location $F04018 in ROM, and so cannot be changed. It is therefore necessary to first move the OAR software into RAM, alter it and then run it from RAM.

Memory Map

1. Data Memory

Address	Mnemonic	description of data
980-FF	THETA	Ocular artefact parameters
A00-27	U	Upper unit triangular matrix
A28-37	B	Unscaled Kalman gain vector
A38-47	V	a local vector
A48-A57	X	EOG data vector
A58-97	Y	EEG data vector
A98-AA7	XBAR	EOG data duplicate
AA8-E7	YBAR	EEG data duplicate
AE8-F7	XMEAN	Mean vector for EOG

Address	Mnemonic	description
AF8-B37	YMEAN	Mean vector for EEG
B38-BEB	ALPHA	used in UD update
B3C-B3F	BETA	"
B40-B43	DELTA	"
B44-47	GAMMA	forgetting factor
B48-4B	NETA	(1-GAMMA)
Constants		
B4C	NPAR	number of parameters
B4D	NPAR-1	
B4E	NPAR-2	
B4F	NCHN	no of EEG and EOG channels
B50	NEEG	no of EEG channels to correct for OA
B51	NEOG	no of EOG channels to acquire
B52	EEGM	no of raw EEG to output
B53	EEGC	no of corrected EEGs to output
B54	EOG	number of EOGs to output
B56	PAR	number of parameters to o tput
B56	DVALID	Data valid flag
I/O devices		
B60-63	PTM	programmable timer
B64-67	FPSEL	FP devices (reserved)
B68-6B	"	"
B6c-B6F	ADCSTC	ADC address
B70-73	DACSTC	DAC address
B74-77	MUX	multiplexer
B78-7B	DMUX	demultiplexer
B7C-7F	PGA	Prog. gain amplifier.
B80-B83	WDSTS	Window detector status
B84-B87	SHLD	output sample and holds
MISCELLANEOUS		
B90-93	ADCMIN	min FP value
B94-97	FPMAX	max FP value
B98-9B	FPONE	floating point ONE.
B9C-9F	DGAMMA	Default value for gamma (407F80)
BA0-A3	DTHETA0	default value for THETA
BA4-A7	DU0	default value for U(0)
BA8-BAB	NEGMAX	minus 1 for FP
BAC	XCESS64	
BB0-B1	OFFST1	off set
BB4-BB5	OFFST2	"
BB6-B7	OFFST3	"
BB8-BB9	DACMAX	max DAC value
USER locations		
C00-03	UTHETA0	user defined $\hat{\theta}(0)$
C04-07	UU0	user defined U(0)
C08-C0B	UGAMMA	usaer specified GAMMA
C0C	UNPAR	user spccified PAR

COD	UEEGM	user specified no of EEGs
COE	UEEGC	user specified EEGC for output
COF	UNCHN	" number of channels (EOG and EEG)
C10	UNEEG	" number of EEGs for correction
C11	UEOG	" number of EOGs

Note: Some of the user specified parameters above are not requested for at run time, but may be specified by physically changing or entering the desired values at the appropriate locations.

---

2. Program Memory.

Address	Program name	Purpose
F04000-053	UDMAIN	main program for on-line OA removal
F04054-079	FXDMAIN	main prog for fixed para method
F0407A-0AE	FXDCRTN	removal control routine
F040B4-373	OAUSER	Fetches user specified constants
F04394-603	OAINIT	initialization routine
F04610-637	EXIT	Exit routine for FP arithmetic.
F04638-6B7	FADD	Floating point addition routine
F046B8-6FD	FMUL	Floating point multiply
F046FE-763	FDIV	Floating point divide routine
F04784-7E9	OADATA	Data acquisition routine
F047EA-7FB	INTERRUPT	Interrupt service routine
F047FC-897	DSAVE	Saves acquired data
F048B8-8F9	UDCRTN	removes OA using UD algorithm
F048FA-A09	UDFLT	updates elements of UD filter
F04A0A-A37	PERR	Computes prediction error
F04A38-A6F	ESTMATE	updates parameter estimates
F04A90-AA5	SCALE	scales parameters for output
F04AA6-B1D	OUTPUT	outputs processed data
F04B1E-B65	FLTINT	floating point to integer convsn
F04B66-B83	DACOUT	converts data to analogue
F04BA4-BED	ERROR	outputs error message when too many paramet. are specified.

```
*
* ROUTINE TO OUTPUT A WARNING WHEN TOO MANY EEG CHANNELS OR
* PARAMETERS ARE SPECIFIED BY THE USER
*

ERROR         LEA       $F00,A5           ; INIT REG WITH BUFFER
              LEA       (A5),A6
              MOVE.W    #'TO',(A6)+       ; PUT MESSAGE IN BUFFER
              MOVE.W    #'O ',(A6)+
              MOVE.W    #'MA',(A6)+
              MOVE.W    #'NY',(A6)+
              MOVE.W    #' C',(A6)+
              MOVE.W    #'HN',(A6)+
              MOVE.W    #'S ',(A6)+
              MOVE.W    #'OR',(A6)+
              MOVE.W    #' P',(A6)+
              MOVE.W    #'AR',(A6)+
              MOVE.W    #'AM',(A6)+
              MOVE.W    #'ET',(A6)+
              MOVE.w    #'ER',(A6)+
              MOVE.W    #'S ',(A6)+
              TRAP      #15                ;OUTPUT MESSAGE.
              DC.W        2
              BSR         SPACE            ;PUT SPACES FOR CLARITY.
              TRAP        #15
              DC.W        0
```

```
*
* UDMAIN.ASM
*   THIS IS THE MAIN PROG FOR THE REMOVAL OF OA BY
*   THE UD FACTORIZATION METHOD.
*

START   LEA      USTACK,A7
        MOVE.W   #$2700,SR                   ;DISABLE INTERRUPTS
        BSR      USER                        ;FETCH USER SPECFD CONSTANTS
        BSR      INIT                        ;INIT SYSTEM
        MOVE.L   #ROM+INTSVC,AVECTR          ;INIT INTERPT VECTR
        MOVE.W   #$7C31,D3                   ;INIT PTM FOR 8MS INTRPT
        MOVEA.L  PTM,A0                      ;FETCH CR#13 ADDR
        MOVE.B   #01,2(A0)                   ;ADDRESS CR#1
        MOVE.B   #01,(A0)                    ;PRESET LATCHES
        MOVEP.W  D3,4(A0)                    ;INIT TIMER LATCHES
        MOVE.B   #$C6,(A0)                   ;ENABLE TIMER
        MOVE.W   #$2500,SR                   ;ENABLE INTERPT
*
*ENDLESSLY LOOP AROUND UNTIL DATA IS VALID
*
        LEA      DVALID,A5                   ;
DMOVE   TST.B    (A5)                        ;IS DATA FLAG SET?
        BEQ      DMOVE                       ;NO. GO TEST AGAIN.
        BSR      OADATA                      ;YES. GO ACQUIRE DATA,
        BSR      UDCRTN                      ;UPDATE UD ELEMNTS & REMOVE OA,
        BSR      OUTPUT                      ;AND OUTPUT DATA TO M/C
        LEA      DVALID,A5                   ;CLEAR DATA FLAG AND
        MOVE.B   #0,(A5)                     ;WAIT UNTIL DATA FLAG IS
        BRA      DMOVE                       ;AGAIN SET.




*-------------------------------------------------------------------
* FXDMAIN - THIS IS THE MAIN PROG FOR THE FIXED PARAMETER ALGORITHM
*-------------------------------------------------------------------
        MOVE.B   #0,DVALID
        LEA      USTACK,A7
DMOVE   LEA      DVALID,A5
        TST.B    (A5)                        ;IS DATA FLAG SET?
        BEQ      DMOVE                       ;NO. THEN GO CHECK AGAIN
        BSR      OADATA                      ;YES. GO ACQUIRE DATA
        BSR      FXDCRTN                     ;THEN GO REMOVE OA
        BSR      OUTPUT                      ;OUTPUT PROCESSED DATA
        LEA      DVALID,A5                   ;CLEAR DATA FLAG
        MOVE.B   #0,(A5)                     ; AND WAIT UNTIL DATA FLAG
        BRA      DMOVE                       ;IS AGAIN SET
*-------------------------------------------------------------------
*
* FXDCRTN  THIS SUBROUTINE CONTROLS THE REMOVAL OF OCULAR ARTEFACTS
* BY THE FIXED PARAMETER ALGORITHM
*
FXDCRTN LEA      XBASE,A1
        MOVE.L   4(A1),D2                    ;FETCH X(3)  (HR)
        MOVE.L   8(A1),D3                    ;FETCH X(4)  (HL)
        BSR      FMUL                        ; COMPUTE HR*HL
        MOVE.L   D3,12(A1)                   ;SAVE AS X(2)  (VL = HR*HL)
        SUB.L    D0,D0
        MOVE.B   NEEG,D0                     ;INIT REGS.
        SUBQ.B   #1,D0                       ;
        SUB.L    D1,D1
        LEA      YBASE,A3
EEGCRT  MOVE.L   (A3),D4                     ;FETCH EEG SAMPLE
        BSR      PERR                        ;REMOVE OA FROM SAMPLE
        MOVE.L   D3,(A3)+                    ;SAVE CORRECTED EEG SAMPLE
        ADD.B    #16,D1                      ;UPDATE ESTIMATES POINTER
        DBF      D0,EEGCRT                   ;ALL EEG CHS CORRECTED?
        RTS
```

```
*
*
* OAUSER.ASM
* PROG REQUESTS FOR AND OBTAINS USER SPECIFIED CONSTANTS
*
*-------------------------------------------------------------
*
* SUBR SPACE - PUTS SPACE BETWEEN REQUESTS FOR CLARITY
*
SPACE   MOVEM.L A5-A6,-(A7)
        LEA     MSGBUF,A5                ;INIT REGS
        LEA     (A5),A6
        MOVE.W  #$'DOA',(A6)+            ;OUTPUT FIVE SPACES
        MOVE.W  #$'DOA',(A6)+
        MOVE.W  #$'DOA',(A6)+
        MOVE.W  #$'DOA',(A6)+
        MOVE.W  #$'DOA',(A6)+
        TRAP    #15
        DC.W    #2
        MOVEM.L (A7)+,A5-A6
        RTS
*----------------------------------------------------------
* SUBR FECTCH.ASM - FETCHES THE NUMBER GIVEN BY THE USER
*
FETCH   MOVEM.L D0/D2/D4/A5,-(A7)
        SUB.L   D3,D3
        MOVE.W  BASE,D4
NXTNUM  CMPA.L  A5,A6
        BEQ     EXITO
        MOVE.B  (A5)+,D0
        AND.L   #$F,D0
        MOVE.L D3,D2
        SWAP    D3
        MULU    D4,D3
        SWAP    D3
        TST.W   D3
        BNE     EXITO
        MULU    D4,D2
        ADD.L   D2,D3
        ADD.L   D0,D3
        BVS     EXITO
        BRA     NXTNUM                   ;GO FETCH NEXT NUMBER
EXITO   MOVEM.L (A7)+,D0/D2/D4/A5
        RTS
*-------------------------------------------------------------
* THIS IS THE ENTRY POINT FOR THE ROUTINE 'USER.ASM'.
* REQUESTS FOR THE NUMBER OF EEG CHNS TO BE CORRECTED.
*
USER    LEA     MSGBUF,A5                ;INIT BUFFER POINTERS
        LEA     (A5),A6
        MOVE.W  #10,BASE
        BSR     SPACE                    ;GO OUTPUT A SPACE
        MOVE.W  #'EN',(A6)+              ;REQUESTS FOR NO OF EEG
        MOVE.W  #'TE',(A6)+              ;CHNS TO BE CORRECTED.
        MOVE.W  #'R ',(A6)+
        MOVE.W  #'NO',(A6)+
        MOVE.W  #' O',(A6)+
        MOVE.W  #'F ',(A6)+
        MOVE.W  #'EE',(A6)+
        MOVE.W  #'G ',(A6)+
        MOVE.W  #'CH',(A6)+
        MOVE.W  #'NS',(A6)+
        MOVE.W  #' T',(A6)+
        MOVE.W  #'O ',(A6)+
        MOVE.W  #'BE',(A6)+
        MOVE.W  #' C',(A6)+
        MOVE.W  #'OR',(A6)+
        MOVE.W  #'RE',(A6)+
        MOVE.W  #'CT',(A6)+
        MOVE.W  #'ED',(A6)+
        TRAP    #15
        DC.W    #2
        LEA     (A5),A6                 ;GO FETCH THE NUMBER GIVEN
        TRAP    #15
        DC.W    #1
        BSR     FETCH
        MOVE.B  D3,UNEEG                ;INIT UNEEG WITH NUMBER
        ADDQ.B  #4,D3                   ;ALSO INIT UNCHN AS
        MOVE.B  D3,UNCHN                ;UNCHN = UNEEG+4
*
* REQUEST FOR THE NUMBER OF PARAMETERS TO USE IN MODEL
*
        BSR     SPACE
        LEA     (A5),A6
        MOVE.W  #'EN',(A6)+
        MOVE.W  #'TE',(A6)+
        MOVE.W  #'R ',(A6)+
        MOVE.W  #'NO',(A6)+
        MOVE.W  #' O',(A6)+
        MOVE.W  #'F ',(A6)+
        MOVE.W  #'PA',(A6)+
        MOVE.W  #'RA',(A6)+
        MOVE.W  #'ME',(A6)+
        MOVE.W  #'TE',(A6)+
        MOVE.W  #'RS',(A6)+
        MOVE.W  #' T',(A6)+
        MOVE.W  #'O ',(A6)+
        MOVE.W  #'US',(A6)+
        MOVE.W  #'E ',(A6)+
        TRAP    #15
        DC.W    #2
        LEA     (A5),A6
        TRAP    #15                     ;GO FETCH THE NUMBER GIVEN
        DC.W    #1
        BSR     FETCH
        MOVE.B  D3,UNPAR                ;INIT NPAR WITH NUMBER
*
* REQUEST FOR THE NUMBER OF CORRECTED EEG CHNS TO OUTPUT
* (EEGC)
        BSR     SPACE
        LEA     (A5),A6
        MOVE.W  #'EN',(A6)+
        MOVE.W  #'TE',(A6)+
        MOVE.W  #'R ',(A6)+
        MOVE.W  #'NO',(A6)+
        MOVE.W  #' O',(A6)+
        MOVE.W  #'F ',(A6)+
        MOVE.W  #'EE',(A6)+
        MOVE.W  #'GC',(A6)+
        MOVE.W  #' T',(A6)+
        MOVE.W  #'O ',(A6)+
        MOVE.W  #'BE',(A6)+
        MOVE.W  #' O',(A6)+
        MOVE.W  #'UT',(A6)+
        MOVE.W  #'PU',(A6)+
        MOVE.W  #'T ',(A6)+
        TRAP    #15
        DC.W    #2
        LEA     (A5),A6
        TRAP    #15                    ;GO FETCH THE NUMBER GIVEN
        DC.W    #1
        BSR     FETCH
        MOVE.B  D3,UEEGC               ;INIT EEGC WITH NUMBER
*
* REQUEST FOR THE NUMBER OF RAW EEGS TO OUTPUT
*
        BSR     SPACE
        LEA     (A5),A6
        MOVE.W  #'EN',(A6)+
        MOVE.W  #'TE',(A6)+
        MOVE.W  #'R ',(A6)+
        MOVE.W  #'NO',(A6)+
        MOVE.W  #' O',(A6)+
        MOVE.W  #'F ',(A6)+
        MOVE.W  #'EE',(A6)+
        MOVE.W  #'GS',(A6)+
        MOVE.W  #' T',(A6)+
        MOVE.W  #'O ',(A6)+
        MOVE.W  #'BE',(A6)+
        MOVE.W  #' O',(A6)+
        MOVE.W  #'UT',(A6)+
        MOVE.W  #'PU',(A6)+
        MOVE.W  #'T ',(A6)+
        TRAP    #15
        DC.W    #2
        LEA     (A5),A6
        TRAP    #15                    ;GO FETCH THE NUMBER GIVEN
        DC.W    #1
        BSR     FETCH
        MOVE.B  D3,UEEGM               ;INIT EEGM WITH NUMBER GIVEN
*
* REQUEST FOR THE NUMBER OF EOG CHANNELS TO OUTPUT
*
        BSR     SPACE
        LEA     (A5),A6
        MOVE.W  #'EN',(A6)+
        MOVE.W  #'TE',(A6)+
        MOVE.W  #'R ',(A6)+
        MOVE.W  #'NO',(A6)+
        MOVE.W  #' O',(A6)+
        MOVE.W  #'F ',(A6)+
        MOVE.W  #'EO',(A6)+
        MOVE.W  #'GS',(A6)+
        MOVE.W  #' T',(A6)+
        MOVE.W  #'O ',(A6)+
        MOVE.W  #'BE',(A6)+
        MOVE.W  #' O',(A6)+
        MOVE.W  #'UT',(A6)+
        MOVE.W  #'PU',(A6)+
        MOVE.W  #'T ',(A6)+
        TRAP    #15
        DC.W    #2
        LEA     (A5),A6
        TRAP    #15                     ;GO FETCH THE NUMBER GIVEN
        DC.W    #1
        BSR     FETCH
        MOVE.B  D3,UEOG                 ;INIT EOG WITH NUMBER
*
* REQUEST FOR THE NUMBER OF PARAMETERS  TO OUTPUT
*
        BSR     SPACE
        LEA     (A5),A6
        MOVE.W  #'EN',(A6)+
        MOVE.W  #'TE',(A6)+
        MOVE.W  #'R ',(A6)+
        MOVE.W  #'NO',(A6)+
        MOVE.W  #' O',(A6)+
        MOVE.W  #'F ',(A6)+
        MOVE.W  #'PA',(A6)+
        MOVE.W  #'RA',(A6)+
        MOVE.W  #'ME',(A6)+
        MOVE.W  #'TE',(A6)+
        MOVE.W  #'RS',(A6)
        MOVE.W  #' T',(A6)+
        MOVE.W  #'O ',(A6)+
        MOVE.W  #'BE',(A6)+
        MOVE.W  #' O',(A6)+
        MOVE.W  #'UT',(A6)+
        MOVE.W  #'PU',(A6)+
        MOVE.W  #'T ',(A6)+
        TRAP    #15
        DC.W    #2
        LEA     (A5),A6
        TRAP    #15                     ;GO FETCH THE NUMBER GIVEN
        DC.W    #1
        BSR     FETCH
        MOVE.B  D3,UPAR                 ;INIT PAR WITH NUMBER
*
* DISPLAY 'OA REMOVAL IN PROGRESS' MESSAGE
*
        BSR     SPACE
        LEA     (A5),A6
        MOVE.W  #'OA',(A6)+
        MOVE.W  #'  ',(A6)+
        MOVE.W  #'RE',(A6)+
        MOVE.W  #'MO',(A6)+
        MOVE.W  #'VA',(A6)+
        MOVE.W  #'L ',(A6)+
        MOVE.W  #'IN',(A6)+
        MOVE.W  #' P',(A6)+
        MOVE.W  #'RO',(A6)+
        MOVE.W  #'GR',(A6)+
        MOVE.W  #'ES',(A6)+
        MOVE.W  #'S ',(A6)+
        TRAP    #15
        DC.W    #2
        RTS
```

```
*
*-------------------------------------------------------------------------
*  SUBR INIT.ASM
*  INITIALIZES POINTERS, AND SYSTEM CONSTANTS WITH USER SPECIFIED
*  CONSTANTS IF THESE ARE VALID OTHERWISE DEFAULT VALUES ARE USED.
*
INIT     MOVE .W #$FE,PTM                 ;INITIALIZE TIMER ADDR.
         MOVE.W  #$8011,PTM+2
         MOVE.W  #$FF,D0                  ;
         MOVE.W  D0,ALU                   ;INIT ALU ADDR.
         MOVE.W  #$FF40,ALU+2             ;PTMMSB ADDR
         MOVE.W  D0,MPY
         MOVE.W  #$FF50,MPY+2             ;
         MOVE.W  D0,ADCSTC
         MOVE.W  #$FF60,ADCSTC+2          ;
         MOVE.W  D0,DACSTC                ;
         MOVE.W  #$FF68,DACTSTC+2
         MOVE.W  D0,MUX
         MOVE.W  #$FF70,MUX+2
         MOVE.W  D0,DMUX
         MOVE.W  #$FF72,DMUX+2            ;
         MOVE.W  D0,PGA
         MOVE.W  #$FF74,PGA+2
         MOVE.W  D0,WDSTS
         MOVE.W  #$FF76,WDSTS+2
         MOVE.W  D0,SHLD
         MOVE.W  #$FF78,SHLD+2
         MOVE.W  #$4C,ADCMIN
         MOVE.W  #$C000,ADCMIN+2
         MOVE.W  #$7F,FPMAX               ;STORE FPMAX
         MOVE.W  #$7FFF,FPMAX+2
         MOVE.W  #$41,FPONE               ;STORE FP ONE
         MOVE.W  #$4000,FPONE+2
         MOVE.W  #$40,DGAMMA              ;STORE GAMMA DEFAULT
         MOVE.W  #$7F80,DGAMMA+2
         MOVE.W  #$3D,DTHETA0             ;STORE THETA DEFAULT
         MOVE.W  #$6666,DTHETA0+2
         MOVE.W  #$3C,DU0                 ;STORE  U(0) DEFAULT
         MOVE.W  #$6666,DU0+2
         MOVE.W  #$41,NEGMAX              ;STORE NEGMAX
         MOVE.W  #$C000,NEGMAX+2
         MOVE.W  #$40,XS64                ;STORE XS64
         MOVE.W  #$8000,OFFST1            ;STORE OFFST1
         MOVE.W  #$800,OFFST2             ;STORE OFFST2
         MOVE.W  #$50,OFFST3              ;STORE OFFST3
         MOVE.W  #$FFF,DACMAX             ;STORE MAX DAC VALUE
         LEA     GAMMA,A0                 ;FETCH USER SPEC GAMMA,CHECK ITS
         MOVE.L  UGAMMA,D0                ;VALIDITY,STORE IF VALID OTHERWISE
         BGT     GDFLT                    ;DEFAULT
         MOVE.L  DGAMMA,D0
GDFLT    MOVE.L  D0,(A0)+
         MOVE.L  D0,D2                    ;COMPUTE 1-GAMMA
         NEG.W   D2
         MOVE.L  FPONE,D3
         BSR     FADD
         MOVE.L  D3,(A0)+
         MOVE.B  UNPAR,D0                 ;FETCH UNPAR,CHECK VALIDITY,STORE
         CMP.B   #1,D0                    ;IF OK, OTHERWISE DEFAULT TO 2
         BLE     NPDFLT
         CMP.B   #4,D0
```

```
        BLE     NPSAV
NPDFLT  MOVE.B  #2,D0
NPSAV   MOVE.B  D0,NPAR
        SUB.B   #1,D0                   ;COMPUTE AND SAVE NPAR-1
        MOVE.B  D0,NPAR+1
        SUB.B   #1,D0
        MOVE.B  D0,NPAR+2               ;COMPUTE AND SAVE NPAR-2
        MOVE.B  UNCHN,D0                ;FECTH USER SPEC NCHN CHECK
        CMP.B   #5,D0                   ;VALIDITY,DEFAULT TO 5 IF NOT
        BLT     NCHDFLT
        CMP.B   #20,D0
        BLE     NCHSAV
NCHDFLT MOVE.B  #5,D0                   ;
NCHSAV  MOVE.B  D0,NCHN
        MOVE.B  UNEEG,D1                ;FETCH UNEEG, CHECK ITS VALIDITY
        BLE     NEDFLT
        SUB.B   #4,D0
        CMP.B   D1,D0                   ;IS NEEG .LE. (NCHN-NEOG) ?
        BGE     NESAV                   ;YES. THEN SAVE.
NEDFLT  MOVE.B  #1,D1                   ;OTHERWISE DEFAULT TO 1.
NESAV   MOVE.B  D1,NEEG
        MOVE.B  #4,NEOG                 ;SET NUMBER OF EOGS TO 4.
        MOVE.B  UEEGC,D0                ;FETCH UEEGC,CHECK VALIDITY AND SAVE
        BLT     ECDFLT
        CMP.B   NEEG,D0
        BLE     ECSAV
ECDFLT  MOVE.B  #1,D0
ECSAV   MOVE.B  D0,EEGC
        MOVE.B  NCHN,D3
        MOVE.B  UEEGM,D0                ;FETCH UEEGC,CHECK VALIDITY AND SAVE
        BLE     EMDFLT
        SUB.B   EEGC,D3
        CMP.B   D3,D0                   ;IS EEGM .LE. (NCHN-EEGC) ?
        BLE     EMSAV
EMDFLT  MOVE.B  #1,D0
EMSAV   MOVE.B  D0,EEGM
        MOVE.B  UEOG,D0                 ;FETCH USER SPECIFIED EOG CHNS FOR
        BLT     EODFLT                  ;OUTPUT, CHECK VALIDITY
        SUB.B   EEGM,D3
        CMP.B   D3,D0
        BLE     EODSAV
EODFLT  MOVE.B  #0,D0
EMSAV   MOVE.B  D0,EOG
        MOVE.B  UPAR,D0                 ;FETCH USER SPECIFD NO OF PARAMETERS
        BLT     UPDFLT                  ;FOR OUTPUT
        SUB.B   EOG,D3
        CMP.B   D3,D0
        BLE     UPSAV
UPDFLT  MOVE.B  #0,D0
UPSAV   MOVE.B  D0,PAR
        MOVE.L  UTHETA0,D0              ;FETCH THETA(0),CHECK VALIDITY
        TST.W   D0                      ;SAVE IF OK
        BLE     THDFLT
        MOVE.L D0,D3
        SWAP.W  D3
        CMP.B   #64,D3
        BHI     THDFLT
        LSL.W   #1,D3                   ;CHECK THAT THETA IS NORMALIZED
        EORI.W  #16384,D1
        BPL     THSAV
        BPL      THSAV
THDFLT  MOVE.L   DTHETA0,D0
THSAV   MOVE.L   #31,D3
        LEA      PBASE,A0
INITHE  MOVE.L   D0,(A0)+                 ;INITIALIZE ALL 64 THETA'S WITH
        DBF      D3,INITHE                ;THETA(0)
        MOVE.L   #0,D0                    ;ZERO ALL U,X,Y
        MOVE.L   #78,D1
        LEA      (A0),A1
ZROUXY  MOVE.L   D0,(A1)+
        DBF      D1,ZROUXY
        MOVE.L   UU0,D0                   ;FETCH U(0),CHECK VALIDITY AND SAVE
        TST.W    D0
        BGT   U0SAV
        MOVE.L   DU0,D0                   ;DEFAULT TO SYSTEM U(0)
U0SAV   SUB.L    D1,D1                    ;INITIALIZE ALL U(I,I) WITH U(0)
        MOVE.B   NPAR+1,D1                ;USE VECTOR SUBSCRIPTING
        MOVE.L   #1,D2                    ;I=1
UINIT   MOVE.L   #4,D4
        MOVE.L   D2,D3                    ;IK=I
        ADDQ.B   #1,D3                    ;IK=I+1
        MULU     D2,D3                    ;I(I+1)=IK
        LSR.W    #1,D3                    ;IK=I(I+1)/2
        SUB.B    #1,D3                    ;IK=IK-1
        MULU     D4,D3                    ;IK=IK*4
        MOVE.L   D0,0(A0,D3.W)            ;STORE U(0) AT U(IK)
        ADDQ.B   #1,D2                    ;I=I+1
        DBF      D1,UINIT
        MOVEA.L PGA,A5                    ;SELECT PGA HI GAIN
        MOVE.W   #1,(A5)
        MOVE.B   #0,DVALID                ;INIT DATA FLAG
        RTS
*
```

```
*
* DATA ACQUISITION ROUTINE
*
OADATA  MOVEM.L D0-D7/A0-A6,-(A7)       ;SAVE REGS.
        LEA     BUFFER,A0               ;INITIALIZE ADDR REGS.
        MOVEA.L MUX,A1
        MOVEA.L ADCSTC,A2
        MOVEA.L PGA,A3
        MOVEA.L WDSTS,A4
        MOVE.W  MPT,D6                  ;FETCH CURRENT BUFFER POINTER
        MOVE.B  #0,D7
        MOVE.L  #4,D1                   ;INITIALIZE COUNTERS
        MOVE.L  #1,D0                   ;INIT CH COUNTER
        SUB.L   D2,D2
        MOVE.B  NCHN,D2
        SUB.L   #1,D2
        MOVE.W  D0,(A1)                 ;SELECT CH 1 OF MUX
        MOVE.W  (A4),D4                 ;READ WINDOW STATUS
        MOVE.W  D4,(A3)
        MOVE.W  D0,(A2)                 ;START ADC
NXTCH   MOVE.W  4(A2),D3                ;END OF CONVERSION?
        ANDI.W  #1,D3
        BHI     NXTCH
        MOVE.W  D7,(A1)                 ;GROUND MUX TO REDUCE XTALK
        MOVE.W  #01,(A3)                ;RESET PGA TO HIGH GAIN
        MOVE.W  D4,D5                   ;SAVE WINDOW STATUS
        ADDQ.W  #1,D0                   ;SELECT NEXT CH OF MUX
        MOVE.W  D0,(A1)
        MOVE.W  (A4),D4                 ;READ WINDOW STATUS AND USE TO
        MOVE.W  D4,(A3)                 ;SELECT GAIN OF PGA
        MOVE.W  (A2),D3                 ;READ 12 BIT SAMPLE
        ASL.W   D1,D3                   ;SHIFT LEFT 4 PLACES
        MOVE.W  D0,(A2)                 ;START ADC OF NEXT CH SAMPLE
        BSR     DSAVE                   ;SAVE CONVERTED SAMPLE(PRESENT CH)
        DBF     D2,NXTCH                ;ALL CH CONVERTED? IF NO CNTINUE
        MOVE.W  D6,MPT                  ;SAVE POINTER
        MOVEM.L (A7)+,D0-D7/A0-A6
        RTS
*-------------------------------------------------------------------
*
* INTRRUPT SERVICE ROUTINE
*
        TAS     DVALID          ;WAS DATA FLAG CLEARED?
        BNE     ERROR1          ;THEN PROG TOO SLOW- REPORT.
        MOVEA.L PTM,A0          ;CLEAR INTERRUPT FLAG
        MOVEP.W 2(A0),D0
        RTE                     ;RETURN FROM INTRPT
*---------------------------------------------------------------------------
*
* DSAVE- SUBROUTINE TO SAVE THE SAMPLE FROM THE ACQUISITION ROUTINE  .THE MEAN
* IS REMOVED BEFORE SAVING SAMPLE IN THE WORK AREA.
*
DSAVE   MOVEM.L D0-D2/D7/A1,-(A7)       ;SAVE REGS.
        MOVE.W  XS64,D1                 ;
        BTST    #0,D5                   ;WAS HIGHER GAIN SELECTED?
        BNE     HGAIN                   ;YES. NO ADJUSTMENT NEEDED
        ADDQ.B  #1,D1                   ;(INC EXPONENT)
BNFULL  SUB.B   D5,D5                   ;SET UP REGS FOR NORMALIZATION
        MOVE.L  #11,D0                  ;ADC MAX EXP.
        SUB.W   OFFST1,D3               ;REMOVE OFFSET FROM SAMP
        TST.W    D3                      ;IS SAMPLE ZERO?
        BEQ      ZSAMPLE                 ;YES. SET FLOATING PT NO TO ZERO
        BPL      SPLNORM                 ;SAMPLE IS POSITIVE. NORMALIZE
        MOVE.W   #1,D5                   ;SAMPLE IS NEG.
        NEG.W    D3                      ;MAKE POSITIVE
        TST.W    D3                      ;IS IT STILL NEG?
        BMI      SADCMIN                 ;YES. MUST BE ADCMIN
SPLNORM ADD.W    D3,D3                   ;NORMALIZE SAMPLE
        DBMI     D0,SPLNORM
        ADD.B    D0,D1                   ;ADJUST EXP. AND MANTISSA
        LSR.W    #1,D3
        SWAP.W   D1                      ;PUT SAMPLE INTO CORRECT
        MOVE.W   D3,D1                   ;FP FORMAT
        MOVE.L   D1,D3
        SUBQ.B   #1,D5                   ;RECOVER SIGN OF SAMPLE
        BNE      RMVMEAN
        NEG.W    D3
RMVMEAN MOVE.L   NETA,D2                 ;REMOVE MEAN FROM SAMPLE
        MOVE.L   D3,D5
        BSR      FMUL                    ;(1-GAMMA)*X(m+1)
RMVM    MOVE.L   D3,D0
        LEA      XMEAN,A0
        LEA      XBASE,A1
        MOVE.L   0(A0,D7.W),D3           ;FETCH X_(m)
        MOVE.L   GAMMA,D2
        BSR      FMUL                    ;GAMMA*X_(m)
        MOVE.L   D0,D2
        BSR      FADD                    ;(1-GAMMA)*X(m+1)+GAMMA*X_(m+1)
SPLSAV  MOVE.L   D3,0(A1,D7.W)           ;SAVE AS X(m+1)
        NEG.W    D3                      ;REMOVE MEAN FROM SAMPLE
        MOVE.L   D3,D2
        MOVE.L   D5,D3
        BSR      FADD
        MOVE.L   D3,0(A1,D7.W)           ;SAVE MEAN-CORRECTED SAMPLE
        MOVE.L   D3,80(A1,D7.W)          ;
        ADDQ.B   #4,D7                   ;UPDATE POINTER
        MOVEM.L  (A7)+,D0-D2/D7/A1       ;RECOVER REGS.
        RTS
SADCMIN LEA      ADCMIN,A5
        MOVE.L   (A5),D3
        BRA.S    RMVMEAN
ZSAMPLE CLR.L    D3
        BRA      RMVM
```

```
*-----------------------------------------------------------------------
*
* OACRTN THIS SUBROUTINE CONTROLS THE CRRECTION OF OCULAR ARTEFACTS
*
UDCRTN  LEA     XBASE,A1
        MOVE.L  4(A1),D2                ;FETCH X(3) (HR)
        MOVE.L  8(A1),D3                ;FETCH X(4) (HL)
        BSR     FMUL                    ; COMPUTE HR*HL
        MOVE.L  D3,12(A1)               ;SAVE AS X(2) (VL = HR*HL)
        BSR     UDUFLT                  ;UPDATE UD ELEMENTS
        SUB.L   D0,D0
        MOVE.B  NEEG,D0                 ;INIT CNTRS  AND REMOVE
        SUBQ.B  #1,D0                   ;OAS FROM THE EEG SIGNALS
        SUB.L   D1,D1
        LEA     YBASE,A3
EEGCRT  MOVE.L  (A3),D4                 ;FETCH EEG SAMPLE
        BSR     PERR                    ;OBTAIN ITS PERR Y-THETA*X
        BSR     ESTMATE                 ;UPDATE COEFFICIENTS
        MOVE.L  (A3),D4
        BSR     PERR                    ;REMOVE OA FROM EEG AND
        MOVE.L  D3,(A3)+                ;SAVE CORRECTED EEG SAMPLE
        ADD.B   #16,D1                  ;UPDATE ESTIMATES POINTER
        DBF     D0,EEGCRT               ;ALL EEG CHS CORRECTED?
        RTS
*-------------------------------------------------------------------
*
*UDUFLT- THIS ROUTINE UPDATES THE ELEMENTS OF THE UDU' FILTER
*
UDUFLT  LEA     UBASE,A0
        LEA     XBASE,A1
        LEA     BBASE,A2
        LEA     (A0),A3                 ;MAKE COPIES OF ADDR PTRS
        LEA     (A1),A6
        LEA     (A2),A5
        SUB.L   D0,D0
        SUB.L   D1,D1
        SUB.L   D5,D5
        MOVE.B  NPAR+2,D5
        MOVE.B  D5,D0
        MOVE.L  $20(A5),D3              ;FETCH X(1)
        MOVE.L  D3,$10(A5)              ;V(1)=X(1)
        MOVE.L  (A0)+,D2                ;FETCH U(1),M=M+1
        BSR     FMUL                    ;U(1)*X(1)
        MOVE.L  D3,(A5)+                ;=B(1)
JLOOP   MOVE.L  D5,D1                   ;INIT K COUNTER
        SUB.B   D0,D1                   ;J1=J-1
        MOVE.L  $20(A5),D4              ;FETCH X(J)
KLOOP   MOVE.L  (A1)+,D3                ;FETCH X(K);K=K+1
        MOVE.L  (A0)+,D2                ;FETCH U(M);M=M+1
        BSR     FMUL                    ;U(M)*X(K)
        MOVE.L  D4,D2                   ;FETCH V(J)
        BSR     FADD                    ;V(J)+U(M)*X(K)
        MOVE.L  D3,D4                   ;=TEMP (V(J))
        DBF     D1,KLOOP                ;K GT. J1?
        MOVE.L  D4,$10(A5)              ;YES. V(J)=V(J)+U(M)*X(K)
        MOVE.L  (A0)+,D2                ;FETCH U(M);M=M+1
        BSR     FMUL                    ;U(M)*V(J)
        MOVE.L  D3,(A5)+                ;B(J)=U(M)*V(J);J=J+1
        LEA     (A6),A1                 ;RESET K COUNTER;K=1
        DBF     D0,JLOOP




*
```

```
*
* UPDATE B AND U VECTORS
        LEA     (A2),A6                 ;INIT ADDR REGS.
        LEA     (A2),A5
        LEA     DELTA,A0
        LEA     ALPHA,A1
        MOVE.L  $10(A5),D2              ;FETCH V(1)
        MOVE.L  (A5)+,D3                ;FETCH B(1);J=2
        BSR     FMUL                    ;B(1)*V(1)
        MOVE.L  GAMMA,D2                ;FETCH GAMMA
        MOVE.L  D2,D4                   ;SAVE IN TEMP
        BSR     FADD                    ;GAMMA+B(1)*V(1)
        MOVE.L  D3,(A1)                 ;SAVE AS ALPHA
        MOVE.L  FPONE,D2
        EXG     D2,D3
        BSR     FDIV                    ;1/ALPHA
        MOVE.L  D3,(A0)                 ;SAVE AS DELTA
        MOVE.L  (A3),D2                 ;FETCH U(1)
        BSR     FMUL                    ;U(1)*GAMMA*DELTA
        MOVE.L  D3,(A3)+                ;=U(1)
        SUB.L   D1,D1
        MOVE.L  D5,D0
JJLOOP  MOVE.B  D5,D1                   ;
        SUB.B   D0,D1                   ;J1=J-1
        MOVE.L  (A1),D4                 ;SET BETA1=ALPHA
        MOVE.L  $10(A5),D3              ;FETCH  V(J)
        MOVE.L  (A5),D2                 ;FETCH B(J)
        BSR     FMUL                    ;B(J)*V(J)
        MOVE.L  (A1),D2                 ;FETCH ALPHA
        BSR     FADD                    ;ALPHA+B(J)*V(J)
        MOVE.L  D3,(A1)                 ;=ALPHA
        MOVE.L  (A0),D3                 ;FETCH DELTA
        MOVE.L  $10(A5),D2              ;FETCH V(J)
        BSR     FMUL                    ;V(J)*DELTA
        NEG.W   D3
        MOVE.L  D3,D6                   ;P=-V(J)*DELTA
        MOVE.L  FPONE,D3
        MOVE.L  (A1),D2                 ;FETCH ALPHA
        BSR     FDIV                    ;1/ALPHA
        MOVE.L  D3,(A0)                 ;=DELTA
KKLOOP  MOVE.L  (A3),D7                 ;BETA=U(M)
        MOVE.L  D6,D3
        MOVE.L  (A2),D2                 ;FETCH B(K)
        BSR     FMUL                    ;B(K)*P
        MOVE.L  D7,D2
        BSR     FADD                    ;BETA+B(K)*P
        MOVE.L  D3,(A3)+                ;=U(M)   ,M=M+1
        MOVE.L  D7,D3                   ;FETCH BETA
        MOVE.L  (A5),D2                 ;FETCH B(J)
        BSR     FMUL                    ;B(J)*BETA
        MOVE.L  (A2),D2                 ;FETCH B(K)
        BSR     FADD                    ;B(K)+B(J)*BETA
        MOVE.L  D3,(A2)+                ;=B(K)  ,K=K+1
        DBF     D1,KKLOOP                       ;K GT. J1 ?
        MOVE.L  (A0),D3                 ;YES. FETCH DELTA
        MOVE.L  D4,D2                   ;FETCH BETA1
        BSR     FMUL                    ;BETA1*DELTA
        MOVE.L  (A3),D2
        BSR     FMUL                    ;U(M)*BETA1*DELTA
        MOVE.L  GAMMA,D2
        MOVE.L  GAMMA,D2
        BSR     FDIV
        MOVE.L  D3,(A3)+                ;=U(M),M=M+1
        LEA     (A6),A2                 ;RESET K COUNTER;K=1
        MOVE.L  (A5)+,D2                ;J=J+1
        DBF     D0,JJLOOP               ;J GT. NPAR?
        RTS
*---------------------------------------------------------------------
*
* PERR - ROUTINE TO COMPUTE PERR AND THE CORRECTED EEG SAMPLE
* PERR=Y-THETA*X
*
PERR    MOVE.L  D0,-(A7)
        LEA     XBASE,A1
        LEA     PBASE,A6
        SUB.L   D0,D0
        MOVE.B  NPAR+1,D0
PERLOOP MOVE.L  (A1)+,D2                ;FETCH X(I)
        MOVE.L  0(A6,D1.W),D3           ;FETCH THETA(I)
        BSR     FMUL                    ;X(I)*THETA(I)
        NEG.W   D3                      ;NEGATE PRODUCT
        MOVE.L  D4,D2                   ;FETCH Y
        BSR     FADD                    ;Y-X(I)*THETA(I)
        MOVE.L  D3,D4
        ADDQ.L  #4,A6                   ;I=I+1
        DBF     D0,PERLOOP              ;I GT. NPAR?
        MOVE.L  (A7)+,D0
        RTS
*-----------------------------------------------------------------------
*
* ESTMATE - ROUTINE TO UPDATE THE COEFF. ESTIMATES FOR EACH EEG CH
*
ESTMATE MOVE.L  D0,-(A7)
        LEA     BBASE,A2
        LEA     PBASE,A6
        SUB.L   D0,D0
        MOVE.B  NPAR+1,D0
        MOVE.L  DELTA,D2
        BSR     FMUL                    ;PERR=PERR*DELTA
        MOVE.L  D3,D4                   ;PUT IN TEMP
ESTLOOP MOVE.L  (A2)+,D2                ;FETCH B(I);I=I+1
        BSR     FMUL                    ;B(I)*PERR
        MOVE.L  0(A6,D1.W),D2           ;FETCH THETA(I)
        BSR     FADD                    ;THETA(I)+B(I)*PERR
        MOVE.L  D3,0(A6,D1.W)           ;=THETA(I)
        MOVE.L  D4,D3                   ;FETCH PERR
        ADDQ.L  #4,A6                   ;I=I+1
        DBF     D0,ESTLOOP              ;I GT. NPAR?
        MOVE.L  (A7)+,D0
        RTS
*-------------------------------------------------------------------------
```

```
*-----------------------------------------------------------------------
*
* SUBR OADOUT.ASM - CONSISTS OF A CONTROL SUB 'OUTPUT' AND
* THREE AUX SUBS, AND USED TO OUTPUT PROCESSED DATA TO EEG M/C.
*------------------------------------------------------------------------
* SUBR SCALE- USED TO SCALE PARAMETERS BY 256 BEFORE OUTPUTING TO
* BRING IT WITHIN THE DAC RANGE. ALSO CALLS SUBRS 'FLTINT' AND 'DACOUT'
*
SCALE   MOVE.L  D3,D1
        SWAP    D1
        ADDQ.B  #8,D1
        SWAP    D1
        MOVE.L  D1,D3
*
* CONVERT SAMPLE TO INTEGER AND OUTPUT TO EEG M/C
*
CALLS   BSR     FLTINT
        BSR     DACOUT
        ADDQ.B  #1,D0
        RTS
*------------------------------------------------------------------
* SUBR OUTPUT - CONTROLS THE OUTPUT SUBS.
*
OUTPUT  MOVEA.L DACSTC,A4
        MOVEA.L DMUX,A5
        MOVE.L  #1,D0
*OUTPUT THETA, IF SPECIFIED
        SUB.L   D4,D4
        MOVE.B  PAR,D4                  ;FETCH NO OF PAR FOR O/P
        SUB.B   #1,D4                   ;WAS ANY SPECIFIED?
        BLT     OUTEOG                  ;NO. THEN GO CHECK EOGS
        LEA.L   PBASE,A0                ;YES. THEN GOUTPUT THETAS
LPAR    MOVE.L  (A0)+,D3
        BSR     SCALE
        DBF     D4,LPAR
*
* OUPUT RAW EOGS, IF SPECFIED BY USER
*
OUTEOG  SUB.L   D4,D4
        MOVE.B  EOG,D4                  ;REPEAT ABOVE FOR EOGS
        SUB.L   #1,D4
        BLT     OUTEEG
        LEA     XBASE,(A0)
LXBASE  MOVE.L  (A0)+,D3
        BSR     CALLS
        DBF     D4,LXBASE
*
* OUTPUT RAW EEG, IF SPECIFIED BY USER
*
OUTEEG  SUB.L   D4,D4                   ;REPEAT ABOVE FOR EEG (RAW)
        MOVE.B  EEGM,D4
        SUB.L   #1,D4
        BLT     OUEEGC
        LEA     YBASE,80(A0)
LYBAS1  MOVE.L  (A0)+,D3
        BSR     CALLS
        DBF     D4,LYBAS1
OUEEGC  SUB.L   D4,D4
        MOVE.B  EEGC,D4                 ;REPEAT ABOVE FOR EEGC
        SUB.L   #1,D4
        BLT     OEXIT
        LEA     YBASE,(A0)
LYBAS2  MOVE.L  (A0)+,D3
        BSR     CALLS
        DBF     D4,LYBAS2
OEXIT   RTS

*------------------------------------------------------------------------
*
*FLTINT - SUBROUTINE TO CONVERT SAMPLE FROM FLOATING PT TO 12 BIT INTEGER.
*
FLTINT  MOVEM.L D1-D2/D4,-(A7)          ;
        MOVE.L  D3,D4                   ;MAKE COPIES OF SAMPLE
        MOVE.L  D3,D2
        TST.W   D3                      ;IS SAMPLE=0?
        BEQ     ADOFFST                 ;YES. ADD OFFSET AND EXIT
        MOVE.W  XS64,D1
        ADD.B   #15,D1                  ;ADD REGISTER LENGTH
        SWAP.W  D2                      ;OBTAIN EXPONENT OF SAMPLE
        SUB.B   D2,D1
        CMP.B   #4,D1                   ;IS SAMPLE TOO LARGE?
        BLT     MAXVAL                  ;YES. SET TO MAX VALUE
        CMP.B   #15,D1                  ;IS SAMPLE TOO SMALL?
        BGE     ZERO1                   ;YES
        ASR.W   D1,D3
ADOFFST ADD.W   OFFST2,D3               ;ADD OFFSET TO NUMBER
EXITFTI MOVEM.L (A7)+,D1-D2/D4
        RTS
MAXVAL  MOVE.W  #0,D3                   ;RESET TO MAX NEG VALUE
        TST.W   D4                      ;WHAT IS SIGN OF SAMPLE?
        BMI     EXITFTI                 ;-VE. EXIT ROUTINE
        MOVE.W  DACMAX,D3               ;+VE.SET TO MAX +VE VALUE
        BRA     EXITFTI
ZERO1   CLR.W   D3                      ;
        BRA     ADOFFST
*---------------------------------------------------------------------
*
* DACOUT - ROUTINE TO OUTPUT A 12 BIT SAMPLE TO THE DAC/MUX/S+H/EEG M/C
*
DACOUT  MOVEM.L D2/D5,-(A7)             ;SAVE REGS ON STACK, AND THEN
        MOVE.W  #21,D5                  ;INIT THEM.
        SUB.L   D2,D2
        MOVE.W  D2,6(A5)                ;SEND HOLD COMMAND TO ALL S+HS
        MOVE.W  D5,(A5)                 ;AND DISABLE DEMUX CHANNELS.
        MOVE.W  D3,(A4)                 ;START DAC OF SAMPLE.
        MOVE.W  D0,(A5)                 ;SELECT DEMUX CHANNEL.
        MOVE.W  D0,6(A5)                ;SEND SAMPLE COMMAND TO A S+H.
        MOVEM.L (A7)+,D2/D5             ;RETRIEVE REGS FR STACK AND
        RTS                             ;EXIT
```

Appendix A16 Costing of the OAR system Hardware

	Price (£)
MVE110-1 Microcomputer + Memory	1182.10
Instrumentation amplifiers	282.25
Multiplexers/Demultiplxers	64.25
ADC/DAC	71.88
Operational amplifiers	46.88
Comparators	2.50
Sample and Holds	167.40
Integrated Circuits	40.00
Instrument Racks + Accesories	107.44
Connectors (DIN and 43-way types)	114.39
IC sockets (DIL)	7.76
IC sockets (wire wrap)	20.00
Wrapping Wires	21.00
Capacitors	30.47
Resistors+potentiometers	28.44
Diodes	1.80
Floating point chip set	760.00
Double euroboards +printed circuit boards	80.00
Total	3028.56

Notes

1. The FP chip set is not used at the present
2. The cost above does not include some items that were drawn from the department store but were not paid for.

Contents

THE REMOVAL OF OCULAR ARTEFACTS FROM THE ELECTROENCEPHALOGRAM: A SURVEY

Barrie W Jervis[1], Emmanuel C Ifeachor[2], Elaine M Allen[3], E L Morris[4]

[1] Department of Electrical and Electronic Engineering
Sheffield City Polytechnic
Pond Street
Sheffield S1 1WB

[2] Department of Communication Engineering
Plymouth Polytechnic
Drake Circus
Plymouth PL4 8AA

[3] Department of Neurosciences
Freedom Fields Hospital
Plymouth
Devon

[4] Department of Engineering Science
University of Exeter
Exeter

Running title: Removal of Ocular Artefacts from the EEG. A survey.

Send proofs to: Dr B W Jervis
Department of Electrical and Electronic Engineering
Sheffield City Polytechnic
Pond Street
SHEFFIELD
S1 1WB

Keywords: EEG, Ocular artefacts, eye movement, eye movement artefact.

1 Ocular Artefacts in the Electroencephalogram.

Electroencephalogram (EEG) signals are well known to be seriously obscured by the superimposition of large electrical potentials associated with eye movements (EM) and/or blinks. The source of these potentials is the corneo-retinal dipole and short-circuiting of it by the eyelid (Hillyard, 1974). Thus EM results in dipole motion and scalp potential changes, modified by any associated eyelid movement. Blinks represent the case of temporary closure of the eyelids and the corresponding artefact is due to the motion of the eyelid over the cornea (Matsuo, et al, 1975; Barry and Jones, 1965). These EM, eyelid and blink artefacts are referred to collectively as ocular artefacts (OA). It is necessary to be able to remove the OA from the measured EEG so that the true EEG record can be studied. For example, it has been found that OAs are associated closely with the CNV response of co-operative normal subjects (Weerts and Lang, 1973; Straumanis et al, 1969; Wasman et al, 1970; Hillyard, 1974; Low et al, 1966). Because the Contingent Negative Variation (CNV) of Huntington's Chorea (HC) patients is of diagnostic value (Jervis et al, 1984) and is frequently completely obliterated by OA we have devoted a considerable effort to identify the best means to remove the OA from the EEG (Ifeachor et al, Papers A and B in preparation). In this paper we review the causes of OAs and methods proposed for removing them from the EEG and include our conclusions on the effectiveness of different OA removal methods (Jervis et al, 1980). We use the term "Electro-oculogram (EOG)" to refer to the electrical potential measured between two electrodes placed close to the eyes and due to OAs.

2 The Causes of Ocular Artefacts

The human eye contains an electric dipole with a positive cornea and negative retina. When the eyes move the electric dipole changes orientation so that the associated electric potential on the scalp, EMA, changes. There are several types of ocular movement (Shackell, 1967; Young and Sheena, 1975) of which the more relevant to EEG work are described here.

Blinks, Figure 1, are characterised by a brief artefact potential of between 0.2 and 0.4 s in duration and occur at intervals of 1-10 s. Barry and Jones (1965) and Matsuo et al (1975) showed that the blink potential was attributable to the eyelid moving over the cornea and not to eyeball movement as was previously thought. When the eyelid moves over the cornea it shorts the positive cornea to the EOG electrode. This shorting effect is removed when the eyelid is again raised. It was also shown (Corby and Kopel, 1972; Overton and Shagass, 1969) that the scalp distribution of blinks was different from that of normal EM, and therefore probably had a separate origin.

Saccadic eye movements, Figure 2, are rapid conjugate movements of speeds between 100 and 500 degree s^{-1}. Normal everyday movements of the eye from one fixation point to another come under this category. These include the "jump and pause" fixation movements performed when reading or scanning a visual field.

During vertical eye movements with the eyes open, a brief potential remarkably similar to blink potential is sometimes observed in the EOG (Barry and Jones, 1965; Ford, 1959). This artefact was called rider artefact by Ford and was later demonstrated (Barry and Jones, 1965) to be due to eyelid movement. Rider artefact is reported to occur mostly during vertical eye movementsor blink reflex. We have, however, observed it in both voluntary vertical and horizontal EOG records, Figure 2.

Smooth pursuit movements involve tracking smoothly visual targets travelling at about 1-30 deg s^{-1}. Smooth compensatory EMs are used to compensate for the motion of the head or trunk during tracking.

Optokinetic nystagmic EM is elicited by a visual field which contains repeated patterns. The potential change produced in optokinetic nystagmus follows a characteristic saw tooth pattern. The eye fixates on a part of the field and tracks it as in pursuit movement. Then there is a return saccade, and the process is repeated.

In convergence EM the eyeballs move in opposite directions. This EM occurs when the eyes focus from far to near objects.

Miniature or fixation EMs include a number of movements that are generally less than 1 degree in amplitude and occur when the eyes are supposedly stationary, as for example during a fixation on a stationary target. The three main miniature EMs are flicks, drifts and tremor. Flicks, or micro-saccades, are small EMs lasting for about 30 ms and which are performed to correct a drift and redirect the eyes to the fixation target. Drift is a slow involuntary EM during fixation that makes the eye wander away from a target for short periods of time before correction by flicks. Tremors are rapid oscillatory EMs which may be superimposed on flicks and drifts. Flicks and drifts from both eyes are partially correlated while tremor is a disconjugate EM. The existence of miniature EM means there is always a measureable EOG signal and this is important because it means the signal is "persistently exciting" which is a necessary requirement for our on-line recursive least squares ocular artefact removal technique (Ifeachor et al, Paper B, in preparation).

Eye flutter is a rapid eyelid movement that tends to occur when the eyes are closed or nearly closed. Flutter frequency is usually in the range 3-7 Hz but occasionally in the range of 8-10 Hz and sometimes as high as 14 Hz.

3 The Measurement of Ocular Movements

Optical methods of detecting EM have been developed (Young and Sheena 1975). In most cases they consist of a light source, optical detectors, and a lens system. The limiting disadvantages of optical methods includes that they cannot be used on unco-operative subjects such as children and some patient categories, that they are bulky, and that they may not give an output when the eyes are closed.

Eye and eyelid movements can be detected from the resulting electro-oculograms (EOGs), (Hillyard, 1974; Shackell, 1967; Quilter et al, 1977). The advantages of the EOG method are that all types of eye movements are detected including voluntary and involuntary, vertical, horizontal, and diagonal movements. EOGs are easy to record from skin electrodes placed around the orbit, some of which are often already placed for EEG recording. The method can be used whether the eyes are open or closed. It is also simple and can be set up quickly with minimal calibration. Whether the EEG ever contributes to the EOG has not been determined.

The main disadvantages of the EOG method are that (i) the relative contributions of movement of the eyeball in the orbit and the eyelid over the cornea cannot be distinguished (Hillyard, 1974), (ii) there is a substantial discrepancy between upward shifts of gaze with the eyes open compared with the downward shift (10-30%). Overton and Shagass (1969) have suggested this is related to a difference in eyelid displacement between upward and downward movements, (iii) the measured potential may include electromyographic (EMG) signals from orbital and temporal muscles although these can be filtered out using low pass filters, (iv) slow rolling eye movements occurring during drowsiness may be impossible to distinguish from "psychogalvanic sway" which are slow drifts of potential related to changes in skin impedance, (v) physical movements of the recording electrode can occur especially if placed over a pulsating artery, or if the electrode is poorly applied. The use of a suitable filter may remove mains interference usually associated with high impedance electrodes.

There is an EOG magnitude variability between individuals and also a variation within an individual, depending on background luminance (leading to the use of the EOG as a measurement of dark adaption) and also with the level of anxiety and state of alertness.

The disadvantages of the above method have led to the technique of Impedance Oculography (Geddes et al, 1973; Sullivan and Welfmen, 1963). Impedance oculography is based on the principle that the impedance across the eye changes as the eye moves. A similar electrode placement method as used for the EOG is employed but in addition a low level high frequency current is injected into the electrodes. Movement of the eye changes the current distribution and produces a modulating signal which is proportional to the change in impedance between the electrodes. This impedance change can be extracted from the output of a demodulator. Although the technique is supposed to eliminate some of the disadvantages of the EOG technique it has seldom been used.

4 Electrode Placements in Electro-oculography

Figure 3 shows some EOG electrode placements which have been used. The placements are determined by the need to obtain maximum information regarding an ocular movement and to use a minimum number of electrodes to record and identify the horizontal and vertical components of ocular movement. Electrodes are placed across and close to the eyes in the horizontal and vertical planes at the upper and lower orbital ridges (above and below the eyes) and the outer and inner canthi. These may be referred to as the vertical and horizontal EOG channels respectively. It has already been pointed out that to obtain an acceptably good estimate of OA more than one EOG signal should be used in the model (Girton and Kamiya, 1973) and our findings are in agreement (Jervis et al, 1980; Jervis et al in preparation; Ifeachor et al, Paper A and Paper B in preparation). It is known that the OA at any point on the scalp is a function of the EOGs of both eyes (Girton and Kamiya, 1973; Henderson et al, 1975). This rules out placement (a) in Figure 3. The arrangement of Figure 3b has the advantage of using few electrodes and minimises cross-coupling between the horizontal and vertical channels which can be a problem in multiple electrode placement (Young and Sheena, 1975; Ifeachor et al, Paper A in preparation). Unfortunately, the method combines the horizontal EOGs of the two eyes. As has been noted (Quilter et al, 1977) and confirmed by our work (Ifeachor et al, Paper A in preparation) there are significant differences between the wave shapes of the left and right EOGs especially during horizontal eye movement. Thus electrode placements that keep the EOGs of the left and right eyes separate would be a better choice. For the same reasons the placements of (d), (e) and (f) if Figure 3 are also unsuitable. Electrode placements (g), (h) and (i) are designed to overcome some of the above disadvantages. Electrode placement (j) is sensitive to VEM, HEM and blinks, but the EOG recorded by this method does not distinguish between VEM and HEM. It has been pointed out (Connolly and Kleiman, 1978) that this distinction may be unnecessary if all that is required is the ability to measure the potential change due to OA. This idea of single channel recording may be extended to reduce the contamination of the EOG by the EEG by using the electrode placement (k). In our work (Jervis et al, 1980; Jervis et al, in preparation; Ifeachor et al, Paper A and Paper B in preparation) we found the best correction was obtained using one vertical channel and some combination of the two horizontal channels. However, we did not investigate the placements of Figure 3 (j) and (k).

5 Methods for the Removal and Control of Ocular Artefacts

5.1 Rejection Methods. OA is controlled in the rejection methods by discarding data thought to contain significant OA. This can lead to an unacceptable loss of data. In some Event Related Potential (ERP) studies sections of the record containing OA also contain useful information and deletion of these sections could make the data unrepresentative (Verleger et al, 1982, Gratton et al, 1983). A common technique has been to monitor one EOG channel and to consider that when the level exceeded some fixed threshold the corresponding EEG channels contained unacceptable levels of artefact (John et al, 1977). The threshold is subjectively determined prior to the experiment as that level of EOG for which no vestige of the EOG can be seen in the EEG. We have shown that visual inspection of an EEG cannot be used to determine whether all the artefact has been removed (Jervis et al, 1980; in preparation) and so this method leads to the acceptance of EEG records with relatively low levels of artefact as being artefact free.

Gevin et al (1977) described a method for on-line computer rejection of artefact contaminated EEG segments in the frequency domain. The method is based on the assumption that artefacts due to body movement, eye movement, and muscle action occupy known frequency bands. If the activity in any of these bands increased significantly above the pre-experimental value, the corresponding EEG segments were regarded as contaminated by artefact and were rejected. There are several disadvantages associated with this method. It cannot be used in studies involving unco-operative subjects, the assumption that the artefacts are confined to specific frequency bands may be wrong, and EEG segments containing genuine slow-wave activities can be rejected (Barlow and Rémond, 1981; Govin et al, 1977). The authors found that nearly a third of the data rejected by this method would not have been rejected by the clinician. They also found that their method could not be applied where EMs were continuous as it resulted in the rejection of the entire records. Gotman et al (1973) also used a spectral technique. They observed that the spectra of EEGs containing OA were often different in shape to those containing pathological slow waves. They developed an algorithm to distinguish between the two kinds of waves. When an OA was detected, features extracted from the corresponding EEG were assigned a lower weight. However, the method will result in a lower weight being assigned to genuine pathological slow waves because of the occasional similarity of the two spectra (Gotman et al, 1973; Gotman et al, 1975).

5.2 Eye Fixation Methods. To reduce the amount of data lost by the rejection method the subjects are often asked to fixate their eyes on a target and sometimes to avoid blinking or moving their eyes at critical times during the experiment. However, Rowland (1968) claimed that as many as 50% of the subjects could not fixate successfully. They described a more controlled way of reducing OA by fixation. The subject was required to detect a square wave displayed on a visual display unit (VDU). This resulted in a reduction of EMAs but not of blink artefacts. The method could also cause micro-saccades as the subject searched the VDU for the square wave. Papakostopoulos et al (1973) found that by asking the subject to focus on his own eyes in a mirror directly in front of him the amount of rejected data was significantly reduced.

5.3 EOG Subtraction Methods

These methods probably offer the best means of removing OA from EEGs. All of the methods are based on the principle that the measured EEG is a linear combination of the background EEG and the OA.

(a) Off-line methods

In discrete form

$$y(i) = \theta_1 x_1(i) + \theta_2 x_2(i) + \ldots + \theta_n x_n(i) + e(i)$$

$$= \sum_{j=1}^{n} \theta_j x_j(i) + e(i) \qquad i = 1, 2, \ldots. m \qquad (1)$$

where y(i) and x(i) are the samples of the measured EEG and the EOGs respectively and e(i) is the true background EEG which may be regarded as an error term. The θ_j are constants of proportionality called the ocular artefact parameters, and n is the number of parameters in the model. If the θ_j can be estimated then $\hat{e}(i)$ may be estimated as

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

where the $\hat{\theta}_j$ are the estimates of the θ_j and $\hat{e}(i)$ is the estimate of e(i). The problem then is one of estimating the $\hat{\theta}_j$.

It is usually assumed that the background EEG is uncorrelated noise with zero expectation. We, however, have found that there is a significant amount of correlation in the background EEG and have taken steps to allow for it (Ifeachor et al, Paper A in preparation). It is also generally assumed that the EOG and the EEG are not correlated. In some methods it is assumed that the OA parameters are time-invariant (Barlow and Rémond, 1981; Girton and Kamiya, 1973; Hillyard and Galambos, 1970).

More recently Hillyard (1974) concluded that the time-invariant assumption was not always correct due to the effect of eyelid position, and that is in agreement with our findings (Ifeachor et al, Paper A, in preparation).

Hillyard and Galambos (1970) tried to remove OA associated with CNVs. They first assumed that the artefacts for upward and downward EMs were equal and opposite but soon realised that this was not true. They observed that an upward EM produced a potential change that was 10-30% smaller than for a downward EM. They showed that as much as 23% of the measured CNV could be due to EMA, thereby emphasising the importance of being able to remove EMA.

Verleger et al (1982) pointed out the lack of quantitative assessment of the reliability and validity of ocular artefact removal techniques. This is the problem to which we have been addressing ourselves for some time (Jervis et al 1980; Jervis et al, in preparation; Ifeachor et al, Paper A in preparation). Verleger et al used the electrode placement of Figure 3a and, as we have shown, more than one EOG channel is necessary for OA removal (Jervis et al, in preparation; Ifeachor et al, Paper A). Verleger et al obtained the estimated θ_j of equation (2) by applying the method of least squares to their one EOG channel model.

Gratton et al (1983) recognised that most of the existing correction methods do not distinguish between EM and blink artefacts, and that the $\hat{\theta}_j$ were usually regarded as constants prior to and during the experiments. They also pointed out that there are differences between voluntary and involuntary EMs which may lead to differences in the θ_j's obtained before and during the experiments. Thus they identified and obtained separate parameter estimates for blinks and EMs. They showed that the ERP obtained from the corrected EEG resembled the ERP obtained from insignificantly contaminated realisations. The method is time consuming and would be very expensive in computing time for multi-channel situations: it is not suitable for use on-line.

Whitton et al (1978) have decribed an off-line EOG subtraction method in the frequency domain. Their method suffers from several disadvantages which include the use of only one EOG signal and the assumption of time invariance of θ_j for each subject. The method is also time-consuming and not suitable for multi-channel use or on-line applications.

Quilter et al (1977) have described an off-line OA removal procedure which involves cross-correlating the EOG channels with the EEG channel(s). In this method the correction parameters θ_j are obtained directly from the data. The values of θ_j are different for different data. Only two EOG channels were used. The electrode placement used is shown in Figure 3(i). This arrangement keeps the left and right EOGs separated, so their contributions to OA in the EEG can be accounted for separately. It is noteworthy that these authors pointed out that the EOGs from the two eyes are different, particularly during HEM.

Quilter et al's technique has been extended by Jervis et al (1980; in preparation) to three and four parameter models. Their electrode placement was as in Figure 3 (g). Jervis et al introduced an autocorrelation technique to test quantitatively the effectiveness of OA removal. They performed experiments in which a number of subjects carried out nearly periodic VEMs, HEMs, and diagonal EMs. This produced rectangular wave contamination of the EEG. Assuming that the OA was not completely removed from the corrected EEG, the latter would contain a remanent rectangular wave. The amplitude of this was usually too small to be detected by eye in the background EEG, but when the corrected EEG was auto-correlated it frequently manifested itself as a nearly periodic triangular wave. This provided a sensitive test for remanent artefact and also allowed the authors to establish the best OA removal technique. Thus, they found that the analgue correction method of McCallum and Walter (1968) was inferior to the correlation technique (Jervis et al, 1980; in preparation). They also showed that a four-channel correction model involving both the vertical and the horizontal channels of both eyes was the most effective and that while the use of the vertical channel due to only one eye might be necessary, the EOGs for the HEMs of both eyes were essential. The suggested reason was that during HEMs the dipoles of the two eyes tended to oppose each other. This is not true of VEMs in which the dipoles of both eyes move in conjugation. Both Jervis et al (1980; in preparation) and Quilter et al (1977) suggested that the use of both vertical EOGs could lead to over-correction in some cases. This suggestion has been confirmed by our later work (Ifeachor et al, Paper A).

Fortgens and DeBruin (1983) used the Least Squares approach to obtain the θ_j for a four EOG channel model. They proved theoretically that four EOG channels were necessary for complete correction of the contaminated EEG. However, they assumed that four uncorrelated EOG channels existed. We have since shown that there are not four uncorrelated EOG channels (Ifeachor et al, Paper A in preparation) since for example VEMs produce horizontal as well as vertical EOGs.

The most important correction techniques described so far in which θ_j is not assumed invariant have been the correlation technique of

Quilter et al (1977), and the Least Squares approach taken by, for example, Fortgens and De Bruin (1983). Until now these techniques have been regarded as quite distinct. However, it is worth noting that we have shown that both lead to exactly the same expressions for the θ_j (Ifeachor et al, Paper A in preparation). The two methods simply provide different ways of visualising the determination of θ_j.

(b) On-line methods

In applications requiring real-time analysis, the delay involved when off-line methods are employed is unacceptable. The trend in EEG signal processing is clearly towards real time processing (Harris, 1983; Barlow, 1979) and it is then necessary to remove the artefact on-line.

McCallum and Walter (1968) described an analogue on-line EMA removal technique. One end of a 24 kΩ potentiometer was connected to a frontal EEG electrode and the other end to linked earlobes (see Figure 4). The central terminal of the potentiometer was adjusted by trial and error until visual inspection of the corrected EEG showed no evidence of EMA. Disadvantages of the method include that manual setting up is required, which is very time consuming particularly on multi-channel recordings, and is also subjective and inaccurate. Also Jervis et al, (1980; in preparation) showed that visual inspection of corrected EEGs does not reveal the level of residual artefact, and it is assumed that the correction coefficient is fixed throughout the experiment and is the same for VEMs, HEMs and blinks. We (Ifeachor et al, Paper A and Paper B in preparation) have shown that these assumptions are untrue. Also Jervis et al (in preparation) have shown that this analogue technique may actually be unsuitable for removing artefacts due to VEM, and for all subjects tested there was a degree of coupling between the EOG and EEG when the EOG changed rapidly. This was probably due to the RC network formed by the potentiometer resistance and the electro-chemical capacitance of the electrode-skin interface. This effect was reduced when a larger value potentiometer (1 MΩ) was used. The presence of this RC time constant will result in distortion of the EOG and hence false artefact correction. Wasman et al (1970) reported that their attempt to use this method was unsuccessful as it resulted in a positive CNV.

Girton and Kamiya (1973) also used a potentiometer adjustment method (see Figure 5). Their technique incorporated three modifications to the above method. The EOGs used were measured round the eyes; the use of differential amplifiers improved the signal-to-noise ratio and the reliability of the system; both the vertical and horizontal components of the EOGs were included. However, this system can be adjusted for only one setting and therefore cannot correct for both blinks and EMAs. Also the horizontal EOG was measured across both eyes.

Barlow and Rémond (1981) introduced a further improvement in that the same EOG signals were used for correcting EMA for each of the EEG channels used. There was also a facility for independently switching the vertical and horizontal EOGs in or out of the circuit, and for changing the polarity of the EOG. The method suffers the disadvantages of those due to McCallum and Walter (1968) and to Girton and Kamiya (1973). Further disadvantages of these methods are that they are unsuitable for use with unco-operative subjects.

We have discussed the disadvantages of presently available on-line and off-line OA correction procedures. A common factor was that the correction parameter was regarded as fixed for a particular subject and length of data.

We have developed an on-line OA removal technique which continuously updates the parameters of the OA removal model based upon recursive least squares (RLS) techniques which we are now in the process of implementing on-line in a microprocessor-based instrument.

6 Quantitative Assessment of Ocular Artefact Removal

Verleger et al (1983) attempted to establish the reliability or otherwise of subject-specific parameter estimates since other workers found that the parameters were subject-specific. Their implied conclusion was that the subject-specific estimates were unreliable. However, the assumption was made that the parameter estimate θ_j for any individual would be the same for both a comparison and a concept-learning task. Different ocular movements may, however, be elicited in the two tasks which would invalidate the assumption. This would be particularly so with the 67 subjects aged 10 to 13 years used in their experiment, and of whom 25 were mentally retarded. The wide variation in the estimates for each task and between tasks is probably largely due to genuine factors and perhaps to the inadequacy of the model used. Their one-parameter model neither corrected for HEM nor catered for the differential contribution of blinks and the effects of lid position.

Gratton et al (1983) assessed the constancy of the parameter estimates over different trials for the same subjects in and between experimental sessions. Their results showed that for a given subject the parameter estimate was not constant during or between experimental sessions and that there are inter-subject differences in the parameter estimates. They also showed that the variation in the subject specific parameter estimates between sessions is larger than during a given session, which they pointed out may be due to not placing the electrodes at exactly the same position in the sessions. This variation emphasises the need for correcting not only on an individual basis but also adaptively.

Verleger et al (1982) investigated the validity of their correction procedure using the criteria that visual inspection of the corrected EEG or ERP derived from it should show little trace of EOG and the ERP should not be attenuated, that the procedure should give substantial benefits over the rejection method in terms of data loss, and that the covariance between the corrected ERP and the EOG should be lower than that between the ERP derived from the rejection method and the EOG. The covariance criterion showed that the correction procedure was significantly superior to the rejection method. This was to be expected since the correction criterion is the minimisation of the covariance between the measured EOG and EEG. Gratton et al (1983) assessed the validity of their correction procedure by determining how different the ERP from the corrected EEG is to the true ERP. Here the true ERP was taken to be that obtained from trials with insignificant artefact selected by the rejection method. To obtain a quantitative assessment they computed a deviation index which was the sum of squares of the deviations of the corrected ERP from the true ERP. They concluded that in general the corrected ERP resembled the true ERP more than the uncorrected one, that when the EEG contained little OA the correction procedure slightly increased the deviation index, and that the correction procedure reduced the variance of the ERP. The validity of the method rests upon a common assumption that the ERP from trials selected by the rejection method is a good estimate of the ERP, but as the authors point out this assumption may sometimes be incorrect. For example, if the number of trials used to obtain the estimate of the true ERP is small, the noisy EEG would affect the average. Furthermore, rejection methods inevitably accept trials with low level artefact which could affect the result. Such effects may explain why the deviation index never becomes zero. Examining their results in some detail suggests that they are not all reliable. For example, for subject 2 in their Table III the indices in two sessions before correction were 28.74 and 6.04 respectively, the smaller value indicating that there was little EMA in the trials. After correction the indices became 19.82 and 5.63 respectively. One would have expected the two indices to tend to similar values. The difference suggests that the correction procedure may not have removed all the artefacts.

The authors have carried out further extensive investigations of the effectiveness of ocular artefact removal methods and these will be described elsewhere (Jervis et al, in preparation; Ifeachor et al, Papers A and B in preparation).

7 Other Practical Problems of Removing EMA from the EEG

A disadvantage of using the EOG in removing OA is that it may contain other artefacts not related to ocular movement. These artefacts may include electro-myographic (EMG) signals resulting from muscle action, contamination of the EOG by the frontal EEG signals, or electric interference picked-up by the electrodes. Because in the EOG subtraction method a fraction of one or more of the EOG signals is subtracted from the measured EEG, any artefacts in the EOG would be introduced into the EEG as "secondary artefacts". The technique usually used to deal with this problem is to filter out the EOG artefacts which are mostly high frequency activities (Ford, 1959; Reiman et al, 1974; Shackell, 1967; Weerts and Lang, 1973). Whitton et al (1978) in their spectral analysis of the EOG and the EEG have found that there is little shared activity between the two signals above 8 Hz. For this reason they employed low pass filtering to remove any high frequency activities in the EOG above this range. To avoid the introduction of phase distortion they filtered in both directions of time. We have studied this problem and are in agreement with these authors about the high frequency nature of the secondary artefact and that low pass filtering is probably the best way to deal with the problem. However, the simple filter used by Whitton et al (1978) would introduce amplitude distortion of the EOG signals, and their double direction filtering approach, apart from not completely removing phase distortion because of end effects, cannot be used for on-line correction. A better approach is to use an efficient zero phase low pass digital Finite Impulse Response filter (Hamer et al, in preparation).

The extent of the effect of secondary artefact has not been settled. Verleger et al (1982) stated that the effect of EEG contamination was by and large unnoticeable. Quilter et al (1977) reported a small increase in noise in the EEG signals after correction which was only significant in a few cases. Shakell (1967) stated that in most cases the EEG contamination was less than the equivalent of half a degree of eye movement, thus making the problem insignificant. In some subjects with relatively large EEG contaminated EOGs, no apparent relationship was found between the scalp EEG and the EOG contamination. We (Ifeachor et al, Paper B in preparation) have studied the variation of the estimates of the model parameters $\hat{\theta}_j$ with time and found that the variance of $\hat{\theta}_j$ for those EOG channels which contained relatively more noise were larger than for less noisy EOGs. We interpreted this as indicating that secondary artefacts do affect the correction process.

It is usually assumed that the background EEG is not correlated with the EOG (eg Quilter et al (1977); Fortgens and De Bruin (1983); Gratton et al (1983)). We have found that the background EEG is significantly correlated (Ifeachor et al, Paper A in preparation), and have had to take this into account by representing the residuals as an autoregressive series. This technique overcomes the effect of the correlation.

The removal of OAs by the EOG subtraction technique has to be considered carefully when OAs may be present together with the ERPs or with pathological slow waves of interest. Since the fraction of the EOG subtracted is obtained from the degree of correlation between the EOG and its component in the EEG, the presence of ERPs or slow waves of similar shape to the OA can lead to the subtraction of a fraction which depends on the ERP or slow wave as well as the EOG. This results in over-correction. For example, it is very difficult to distinguish between the electrical potential contributed by eye movement and the true CNV potential (Weerts et al, 1973; Straumanis et al, 1969; Hillyard and Galambos, 1970; Wasman et al, 1970). Muras and Binnie (1970) used judicious electrode placement to distinguish between OA and slow waves. It will probably be necessary to distinguish between the various ERP components, slow waves, and OAs by examining the propagation of the electric potentials over the scalp by using a number of scalp electrodes. This is an area we hope to investigate in the near future.

Another problem lies in the possibility of overloading the amplifier and analogue-to-digital conversion system. Quilter et al (1977) found this to be their most significant problem. Overloading causes erroneous parameter estimates and hence a reduction in the effectiveness of the converter. The common approach is to set the dynamic range of the analogue-to-digital converter (ADC) such that overloading is infrequent. This often involves a compromise between utilising most of the dynamic range of the converter and avoiding overload. Verleger et al (1982) reported a data loss of about 4% because of overloading. We have introduced a method of dynamically changing the range of the ADC which should alleviate the overload problem.

SUMMARY

This paper constitutes a review of ocular artefacts in the human electroencephalogram (EEG). The causes of the artefacts are explained. Methods for their removal from the EEG are reviewed and the effectiveness of the different methods is discussed.

Besides covering the existing literature, the text is interspersed with comments derived from our own observations. It is shown that analogue subtraction techniques are inferior in many aspects to techniques based upon parameter estimation using the Least Squares Technique (to which correlation correction belongs). It is shown that the different types of ocular artefact ie eye movement artefact (EMA), and blink artefact can never both be simultaneously completely removed, although overall satisfactory OA removal can be achieved. Models used in removing OA from the EEG are discussed. Reference is made to ways of assessing the effectiveness of the different models.

The correlation of the background EEG, normally assumed to be random, is mentioned together with autogressive modelling techniques which reduce the effects of correlation. In a look ahead we also mention our recursive least squares approach for on-line realtime removal of OAs which is to be implemented in a microprocessor-based system.

Sommaire

Ce mémoire constitue un compte-rendu des artefacts oculaires (OA) apparaissant dans l'électro encephalogramme (EEG) humain. Leurs causes sont expliquées. Des méthodes pour leur suppression sont passées en revue et leur efficacité discutée.

Outre couvrant la documentation actuelle, le texte est parsemé de commentaires dérivant de nos propres observations. Il est montré que les techniques de soustraction analogue sont inférieures, par de nombreux aspects, a celles basées sur l'estimation paramétrique utilisant la technique des moindres carrés (auquelle la correction correlative appartient). Il est montré que les différents types d'OA, celui du mouvement de l'oeil (EMA) et celui du clignotement ne peuvent être simultanément totalement supprimés, bien que la suppression entièrement satisfaisante de l'un des duex peut êntre achevée. Las modèles utilisés pour la suppression de l'OA de l'EEG sont discutés. Une référence est faite aux moyens d'estimation de l'efficacité des différents modèles.

La corrélation de l'EEG de base, normalement assurée d'être fortuite, est mentionnée avac les techniques de différentciation et celles de modèle autoregressif qui en réduisent les effets.

En prévision, nous mentionnons également notre approche récidiviste des moindres carrés pour une suppression des OAs à connection directe qui doit être réalisée sur un système à microprocesseur.

nt

R^2
.42
.41
.43
.43
.42
.43
.43
.42
.44
.56
.58
.56

References

barlow J S

Computerized clinical Electroencephalography in perspective, IEEE Trans. Biomed. Engr., 1979, 26, pp 377-390.

Barlow J S and Rémond A

"Eye Movement Artefact Nulling in EEGs by Multichannel on-line EOG Subtraction". Electroenceph and Clin. Neurophysiol., 1981, 55, pp 418-423.

Barry W and Jones G M

"Influence of Eyelid Movement upon Electro-oculographic Recording of Vertical Eye Movements". Aerospace Medicine, 1965, Vol. 36, pp 855-858.

Connolly J F and Kleiman K M

"A Single Channel Method for Recording Vertical and Lateral Eye Movements". Electroenceph and Clin. Neurophysiol., 1978, 45, pp 128-129.

Corby J C and Kopel B S

"Differential Contributions of Blinks and Vertical Eye Movements as Artefacts in EEG Recording" Psychophysiology, 1972, Vol 9, pp 640-644.

Ford A

"Significance of Terminal Transients in Electro-oculographic Recordings". Archives of Ophthalmology, 1959, Vol. 61, pp 899-906.

Fortgens, C and De Bruin M P

"Removal of Eye Movement and ECG Artefacts from the Non-cephalic Reference EEG". Electroenceph. Clin. Neurophysiol, 1983, Vol. 56, pp 90-96.

Geddes L A, Mcgrady J D and Hoff H E

"The Impedance Nystagmogram - A Record of the Level of Anesthesia in the Horse" South Western Veterinarian, 1965, Vol. 19, pp 23-25.

Gevin A S, Yeager C L, Zeitlin G M, Ancoli S and Dedon M F
"On-line Computer Rejection of EEG Artefacts". Electroenceph and Clin. Neurophysiol., 1977, 42, pp 267-274.

Girton D G and Kamiya J
"A Simple On-line Technique for Removing Eye Movement Artefacts from the EEG". Electroenceph and Clin. Neurophysiol., 1973, 34, pp 212-216.

Gotman J, Skuce D R, Thompson C J, Gloor P, Ives J R and Ray W F
"Clinical Applications of Spectral Analysis and Extraction of Features from Electroencephalograms with slow waves in Adult Patients". Electroenceph and Clin. Neurophysiol., 1973, 35, pp 225-235.

Gotman J, Gloor P and Ray W F
"A Quantitative Comparison of Traditional Reading of the EEG and Interpretation of Computer Extracted Features in Patients with Supertentorial Brain Lesions". Electroenceph and Clin. Neurophysiol., 1975, 38, pp 623-639.

Gratton G, Coles M G H, and Douchin E
"A New Method for Off-line Removal of Ocular Artefacts". Electroenceph and Clin. Neurophysiol., 1983, 55, pp 468-484.

Hamer C F, Ifeachor E C and Jervis B W
"Digital Filtering of Physiological Signals with Minimal Distortion". In preparation.

Harris M A
"Brain Waves appear on TV in Real-time". Electronics, Vol 24, 1983, pp 47-48.

Henderson C J, Butler S R and Glass A
"The Localisation of Equivalent Dipoles of EEG Sources by the Application of Electrical Field Theory", J Electroenceph. and Clin. Neurophysiol, VoL 39, 1975, pp 117-130.

Hillyard J A and Galambos R
"Eye Movement Artefact in the CNV". Electroenceph and Clin. Neurophysiol., 1970, 24, pp 173-182. See also section 4.3.

Hillyard S A
"Methodological Issues in CNV Research". Bioelectric Recording Techniques Part B, Electroencephalography and Human Brain Potentials; Ed R F Thompson and M M Patterson, Academic Press, 1974, pp 259-280.

Ifeachor E C, Jervis B W, Allen E M, Morris E L, Wright D E and Hudson N R
"Investigation and Comparison of Some Models for Removing Ocular Artefacts from EEG Signals", Paper A, in preparation.

Ifeachor E C, Jervis B W, Allen E M, Morris E L, Wright D E, and Hudson N R.
"A New On-line Method for Removing Ocular Artefacts from EEG Signals", Paper B, in preparation.

Jervis B W, Nichols M J, Allen E M, Hudson N R and Johnson T E.
"The Quantitative Assessment of Electroencephalograms corrected for Eye Movement Artefact", First European Signal Processing Conference, Lausanne, Switzerland, September 1980.

Jervis B W, Nichols M J, Johnson T E, Allen E M and Hudson N R
"A Fundamental Investigation of the Composition of Auditory Evoked Potentials", IEEE Trans. Biomed. Eng., Vol BME-30, No. 1, January, 1983, pp 43-50.

Jervis B W, Allen E W, Johnson T E, Nichols M J and Hudson N R
"The Application of Pattern Recognition Techniques to the Contingent Negative Variation for the Differentiation of Subject Categories", IEEE Trans. Biomed. Eng., Vol. BME-31, No 4, April 1984, pp 342-349.

Jervis B W, Nichols M J, Allen E M, Hudson N R and Johnson T E
"Comparison of Two Methods of Removing Eye Movement Artefact from the EEG", in preparation.

John E R, Karmel B Z, Corning W C, Easton P, Brown D, Ahu H, John M, Harmony T, Prichep L, Toro A, Gerson I, Bartlett F, Thatcher R, Kaye H, Valdes P and Schwartz E.
"Neurometrics", Science, June 24th, 1977, Vol. 196, pp 1393-1410.

Low M D, Borda R P, Frost J D, and Kellaway P
"Surface Negative Slow Potential Shift Associated with Conditioning in Man", Neurology, 1966, 16, pp 771-782.

Matsuo F, Peters J F and Reilly E L
"Electrical Phenomena Associated with Movements of the Eyelid". Electroenceph and Clin. Neurophysiol., 1975, 38, pp 507-511.

McCallum and Walter W H
"The Effects of Attention and Distraction on the Contingent Negative Variation in Normal and Neurotic Subjects". Electroenceph and Clin. Neurophysiol., 1968, 25, pp 319-328.

Muras J S and Binnie C D
"The Recognition of Frontal Slow Activity in the Presence of Eye Movements". The Proc. of the Electrophysiological Technol. Assoc., 1970, Vol. 17, No 2, pp 131-143.

Overton and Shagass C
"Distribution of Eye Movement and Eye Blink Potentials over the Scalp", Electroenceph and Clin. Neurophysiol., 1969, 27, p 546.

Papakostopoulos D, Winter A and Newton P
"The Control of Eye Potential Artefacts in Multi Channel CNV Recordings", Electroenceph and Clin. Neurophysiol., 1973, 35, p 419.

Quilter P M, MacGillivray and Wadbrook D G
"The Removal of Eye Movement Artefact from EEG Signals using Correlation Techniques", Random Signal Analysis, IEE Conference Publication, No 159, 1977, pp 93-100.

Reiman V, Korth M and Keidel W D
"Korrelations Analyse Von EEG Und Augenbevengungen", Vision Res., Vol. 14, 1974, pp 959-963.

Rowland V
"Cortical Steady Potential (Direct Current Potential) in Reinforcement and Learning", in Stellar E, and Sprague J M (Eds), Progress in Physiological Psychology, Vol. 2, Academic Press, New York, 1968, pp 1-77.

Shackell B
"Eye Movement Recording by Electro-oculography". A Manual of Psychophysiological Methods, Ed. P H Venables and I Martin, North Holland Pub. Co., 1967, pp 301-334.

Straumanis J J, Shagass C and Overton D A
"Problems Associated with Applications of the Contingent Negative Variation to Psychiatric Research". Journal of Nervous and Mental Disease, 1969, 148, pp 170-179.

Sullivan G and Welfmen G
"The Impedance Oculogram", Journal of Applied Physiology, 1963, Vol. 18, pp 215-216.

Verleger R, Gasser T and Mocke J
"Correction of EOG Artefacts in Event-Related Potentials of the EEG: Aspects of Reliability and Validity". Psychophysiology, 1982, 19, pp 474-480.

Wasman M, Morehead S, Lee H, and Rowlands V
"Interaction of Electro-ocular Potentials with the Contingent Negative Variation", Psychophysiology, 1970, Vol 7, pp 103-111.

Weerts T C and Lang P J
"The Effects of Eye Fixation and Stimulus and Response Location on the Contingent Negative Variation (CNV)", Biological Psychology, 1, 1973, pp 1-19.

Whitton J, Leu F and Noldofshy H
"A Spectral Method for Removing Eye Movement Artefacts from the EEG", Electroenceph and Clin. europhysiol., 1978, 44, pp 735-741

Young L R and Sheena D
"Survey of Eye Movement Recording Methods". Behaviour Research Methods and Instrumentation, 1975, Vol 7(5), pp 397-429.

Figure Captions

1 Blink Potentials

2 EOGs Showing Saccadic Eye Movements and Rider Artefact (vertical eye movement)

3 EOG Electrode Placements

4 EMA Correction Method of McCallum and Walter

5 EMA Correction Method of Girton and Kamiya

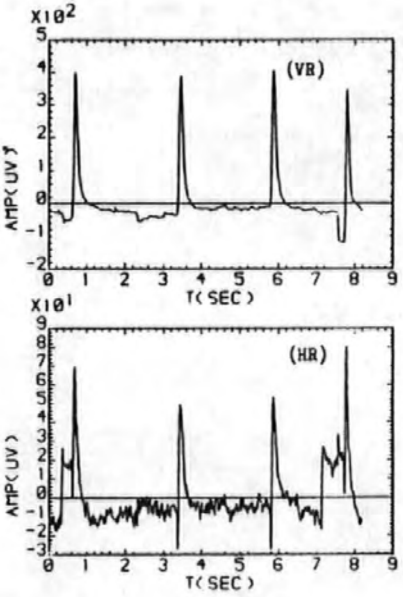
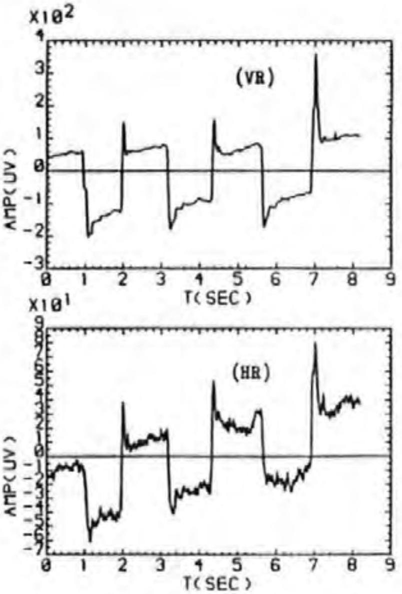

X10^2
5
4
3
2
1
0
-1
-2
(VR)
AMP(UV)
0 1 2 3 4 5 6 7 8 9
T(SEC)
X10^1
9
8
7
6
5
4
3
2
1
0
-1
-2
-3
(HR)
AMP(UV)
0 1 2 3 4 5 6 7 8 9
T(SEC)

Fig 2

n

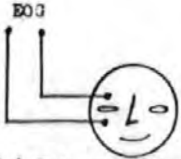
ed

ithms

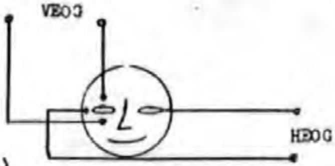


tor,

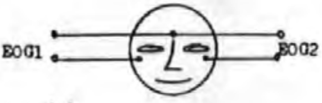
tes



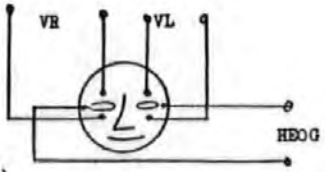
(a) This method is sensitive to VEM and blink but not to HEM. does not take the effect of the two eyes into account



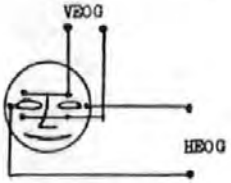
(b)



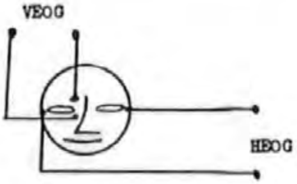
(c)



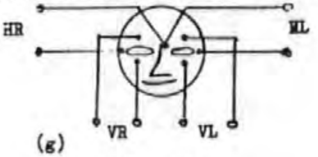
(d)



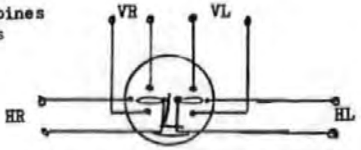
(e) To minimize coupling between the horozontal and vertical atis this method combines the two vertical and th e two horizontal EOGs



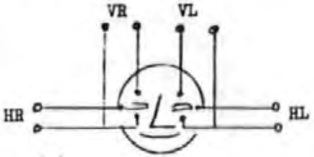
(f)



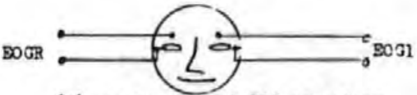
(g)



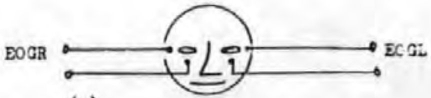
(h)



(i)



(j) single channel EOG recording



(k)

Fig 4

e

.

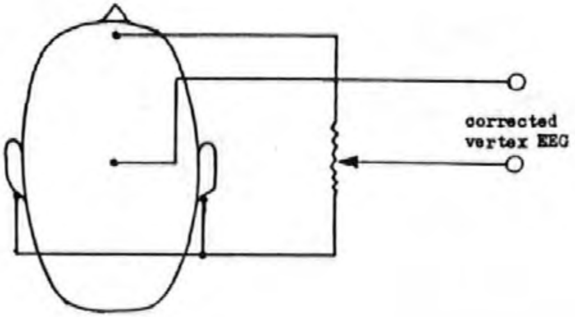
he

s,

ates

ffline

both



n

plots

nston

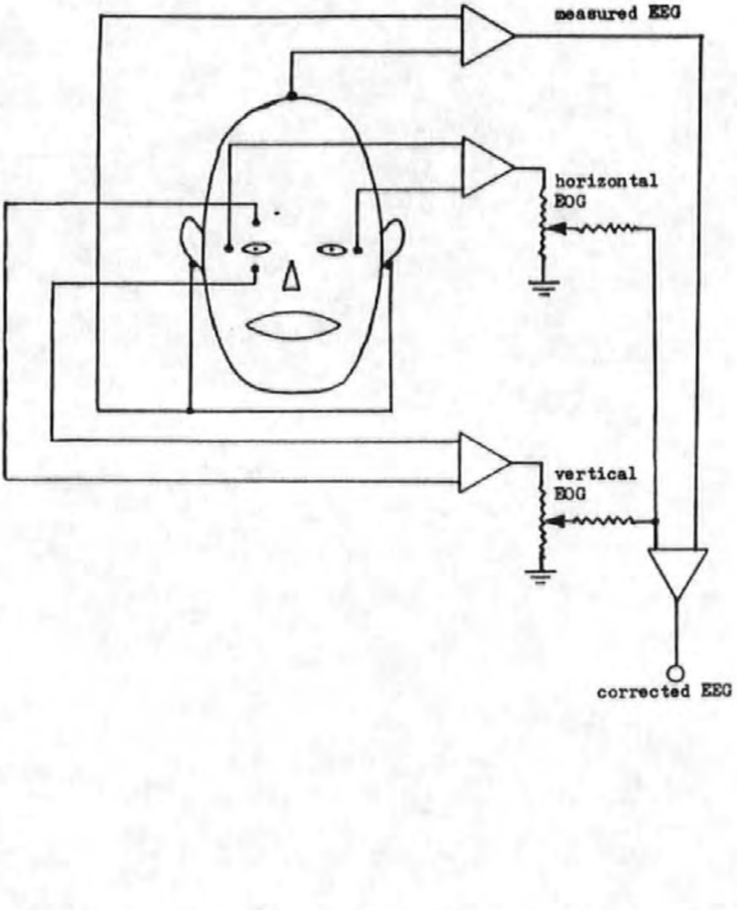
were

the

8a)

8b)

f


measured EEG
horizontal
EOG
vertical
EOG
corrected EEG

DIGITAL FILTERING OF PHYSIOLOGICAL SIGNALS WITH MINIMAL DISTORTION

C.F.Hamer, E.C.Ifeachor, B.W.Jervis.*

Department of Communication Engineering,
Plymouth Polytechnic,
Drake Circus,
Plymouth.
PL4 8AA

*Department of Electrical and Electronic Engineering,
Sheffield City Polytechnic,
Pond street,
Sheffield.
S1 1WB

DIGITAL FILTERING OF PHYSIOLOGICAL SIGNALS WITH MINIMAL DISTORTION

ABSTRACT

The passband amplitude characteristics of digital filters used in physiological applications is generally poor, greater emphasis being placed on precise phase linearity. Following a brief review of the digital filtering techniques commonly employed and their associated errors, a method for designing linear phase filters with excellent amplitude characteristics is discussed. The design and implementation of such a filter suitable for online and offline processing of the electrooculogram is then presented.

KEYWORDS:

Digital filtering, minimal distortion, electrooculogram

1. INTRODUCTION

Digital filtering has become an important feature in the processing of physiological signals such as electroencephalogram (EEG) and electrooculogram (EOG) for diagnostic aid or to reduce signal contamination,(Verburg et al 1974, Whitton et al 1978). The desire for exact phase linearity in the passband which results in zero phase distortion of the wanted signal has limited the choice of digital filters in these applications. Less attention has been paid to the amplitude characteristic in the passband than we feel is justified. In fact any deviation from unity transmission in the passband introduces distortion into the filtered signals. In addition to the poor passband amplitude characteristics of filters which have been used in this field, the technique of phase cancellation by time reversal sometimes employed (see for example Coppola 1979) will be shown to be unreliable.

In connection with ongoing research into the problem of ocular artefact (OA) in the EEG a requirement arose to filter EOG signals to remove secondary artefact unrelated to ocular movement, such as external electrical interference, muscle artefact and cross coupling from the frontal EEG. Due to the subsequent application for the EOG it was important that the filter introduced the minimum possible distortion into the true EOG. The filter was required for offline and online processing, the online filter being part of a larger process and hence time critical. After a detailed survey of the possible design and implementation techniques the excellent methods of McClellan et al (1973) and Rabiner (1977) were selected as the best approach. Due to the advent of advanced 16 bit microprocessors and the efficiency of Rabiner's algorithm a purely software implementation was possible for both requirements.

In the next section a brief review of digital filter types is given, the problems associated with using IIR filters are outlined and the advantages of McClellan's design are defined. Following this the design and implementation of a filter suitable for reducing noise in the EOG is discussed and its effect on typical data is illustrated.

It should be noted here that although filtering of the EOG is discussed, the methods employed are equally suitable for other physiological applications.

2. DIGITAL FILTERING METHODS

Digital filters may be loosely split into two types, finite impulse response (FIR) and infinite impulse response (IIR) with recursion relationships as given by eqns. 1 and 2 respectively.

$$y(n)=\sum_{m=0}^{m=N-1} h(m)x(n-m) \quad \text{-------------} \quad (1)$$

$$y(n)=\sum_{m=0}^{m=k-1} a(m)y(n-m) + \sum_{m=0}^{m=l-1} h(m)x(n-m) \quad \text{-------} \quad (2)$$

where: $y(n)$= the output time sequence

$x(n)$= the input time sequence

$h(m)$= filter coefficients

$a(m)$= filter coefficients

N = number of coefficients for FIR filter

k,l = numbers of coefficients for IIR filter

Exact phase linearity may only be obtained from an IIR filter if one or both of the following criteria are met:

1. All the poles of the transfer function lie on the unit circle in the Z plane.
2. The filter is unrealiseable in realtime.

Clearly, the first criterion considered in isolation leads to an unstable filter since the Z plane poles must reside inside the unit circle for stability. However, Lynn(1977) has used this criterion to obtain stable linear phase IIR filters with integer coefficients by placing zeros equispaced around the unit circle in the Z plane and cancelling selected zeros with corresponding poles to form passbands. Due to the rapid processing speed of these filters, they represent an economical solution to many online physiological filtering requirements. However, the amplitude of the frequency response of an rth order lowpass filter using Lynn's design is given by equation 3.

$$H(e^{j\omega}) = \left[\frac{\sin\left(\frac{\omega NT}{2}\right)}{\sin\left(\frac{\omega T}{2}\right)}\right]^{r} \qquad \text{-------------} (3)$$

From this it can be seen that the passband is defined by the width of the main lobe which is in turn set by the sampling period (T) and the parameter N. If the filter is normalised to unity gain, unity transmission only occurs at DC, the deviation from unity increasing rapidly with frequency over any frequency band defined within the main lobe. To achieve greater attenuation in the stopband requires increasing the order (r) and hence the deviation in the passband. Recently Lynn(1983) has described another class of integer coefficient IIR filters offering improved frequency amplitude characteristics which he has named transversal resonator digital filters (trdf). The synthesis of a filter to approximate a desired frequency response by this method firstly involves the conversion of the frequency specification to an equivalent impulse response for which Lynn uses an N point FFT routine. The trdf is then designed to approximate this impulse response in the time domain. The procedure does not involve any minimisation of the deviation between the specified amplitude of the frequency response and that obtained by interpolation between the N frequency samples taken. No account is taken of this deviation in the paper and all of the ripple evident in the design example of bandpass filters for EEG analysis is attributed to impulse response quantisation. It is difficult to predict the deviation from the desired response at the design stage using this method. Consequently, as the author indicated in his conclusions, for applications not involving tight constraints on the amplitude of the frequency response these filters provide an excellent solution but where the deviation from the desired response is to be closely controlled, they do not represent the best solution. The EOG filter, as discussed in section 3, falls into the latter category.

Others have employed the second criterion to achieve linear phase by applying the same filter twice, once in the positive time direction and once in the negative time direction. (Coppola 1979, Whitton et al 1978, Verburg et al 1974). Consider a digital filter with impulse response h(n) and transfer function H(Z). Figure 1. shows the two methods of using this filter to form a time reversal (TR) filter with new impulse response $h_{TR}(n)$ and transfer function $H_{TR}(Z)$. Now if X(Z) is the Z transform of the sequence x(n), then the Z transform of the time reversed sequence x(-n) is $X(Z^{-1})$. Thus for method 1:

$$H_{TR}(e^{j\omega}) = H(e^{j\omega})H(e^{-j\omega}) = \left|H(e^{j\omega})\right|^2$$

and for method 2:

$$H_{TR}(e^{j\omega}) = H(e^{j\omega})+H(e^{-j\omega}) = 2\left|H(e^{j\omega})\right|\cos\phi(\omega)$$

Method 1 produces a superior result and is the method generally adopted because the amplitude characteristic is not scaled by the cosine of the original phase function. To demonstrate the phase cancellation, consider a simple non linear phase FIR filter with N=3. Figure 2 shows the original and resulting impulse responses when this filter is used to realise a TR filter by method 1. Clearly, the TR filter introduces zero phase distortion since $h_{TR}(n)$ is symmetrical about the time of application of the impulse (n=0). In order to produce this exact symmetry all of the sequence h(n) must be present prior to reversing the time axis. When using IIR filters, the impulse response is theoretically infinite and therefore h(n) cannot be completely present prior to the time reversal. In practice the response is truncated to prevent machine underflow. This introduces errors in the symmetry of $h_{TR}(n)$ and some phase distortion results. This effect is exaggerated in figure 3 where the response of the FIR filter is truncated at values less than or equal to unity. The asymmetry of the resulting response implies phase distortion and indicates that the amplitude response will not be as expected. It is therefore important to ensure that h(n) for the IIR filter used has decayed to such a value that errors in symmetry resulting from its truncation are negligible prior to reversing the time axis. It should be noted that serious errors can result from truncation of the response at the end of the data array on the first filter application. This causes distortion of samples towards the end of the filtered data array, the amount of distortion being dependant upon the rate of decay of the IIR filter's time response. Another drawback of this method is its inapplicability to realtime processing.

Exact phase linearity is obtained from FIR filters by ensuring at the design stage that the filter coefficients and hence the impulse response are symmetrical about a given point (sample position (N-1)/2). There are basically three well known techniques for designing FIR filters (Rabiner and Gold 1975).

1. The WINDOW method.
2. The FREQUENCY SAMPLING method.
3. The OPTIMAL (minimax error) method.

Of these the window method is the commonest in this field probably due to the simplicity of the procedure. Although adequate filter responses may be achieved by using 'optimal' window functions such as the Kaiser window, the frequency response obtained is not optimal in any rigid sense. The pass and stopband edge frequencies cannot be specified at the design stage since the design process involves convolution in the frequency domain which 'smears' any transition frequencies into transition bands. Ripples introduced into the response by the same process satisfy no criteria for minimising the deviation from the desired response. The frequency sampling method produces superior results to the window method but involves complex linear programming and is not recommended since the optimal method of McClellan et al (1973) produces even better results using a Fortran design program (Rabiner and Gold 1975).

As shown by McClellan et al (1973), the optimal design is formulated as a Chebyshev approximation problem ensuring that the peak error of approximation is minimised over the entire range of the approximation. Given the band edge frequencies, the number of coefficients and the required function in each band (eg. pass, stop), the Remez multiple exchange algorithm (Cheney 1966, Remez 1957) is employed to obtain the filter coefficients which optimise the deviation in each band in a MINIMAX sense. The band edge frequencies remain unchanged by the design process. Also, a weighting function on the deviation in each band allows the optimised deviation between different bands to be traded relative to one another. This method was hence selected as the best design procedure for the EOG filter.

.935
.942
866
899

3. THE EOG FILTER

Ocular movement related signals in the EOG are low frequency activities typically below 8 Hz (see for example Whitton et al 1978). Thus the EOG filter was designed to satisfy the specifications given in figure 4. The number of multiplications allowed so as to give sufficient time for all the computations following the filter was set at a maximum of 19 between samples. This implies a maximum of 37 coefficients in a linear phase FIR filter.

Using McClellan's design routine, the absolute value of the deviation in each band is not specified by the designer but is obtained via the optimisation procedure. Hence the specified deviations may be met either by iteration of the program inputs alone or by calculation of the initial inputs using equations available in Rabiner and Gold (1975) followed by iteration. A form of the second approach was adopted in the EOG filter design. Since all of the filter specifications except for fp were essentially variable, equation 4 was used to obtain the range of possible solutions shown in figure 5 by calculating Δf for the maximum allowable deviations and a range of numbers of filter coefficients, $N \leqslant 37$.

$$\Delta f = \frac{N-1}{2f(\delta_1,\delta_2)}\left[\sqrt{1+\frac{4f(\delta_1,\delta_2)D_\infty(\delta_1,\delta_2)}{(N-1)^2}}-1\right] \quad ----(4)$$

where $fs = \Delta f + fp$

fs = normalised stopband edge frequency fp = normalised passband edge frequency

Δf = transition width

δ_1 = passband deviation

δ_2 = stopband deviation

This defined the curved lower boundary in figure 5, the top boundary being set by fs=20Hz and the side boundary by N=37. N was implicitly odd to avoid a non integer number of samples delay through the filter. From this the program inputs marked (x) were used with a range of weighting functions to obtain the curves shown in figure 6 from the design program. It can be seen from figure 6 that there exists a point on the curve for N=29 where the deviation in each band is the same as for N=31. A similar result occurs between N=31 and N=33. These are extraripple solutions (Rabiner and Gold 1975) and are particularly efficient. The solution selected on the basis of these results was the predicted extraripple solution for N=29 which was obtained with a weighting function of 8,1 giving pass and stopband deviations of -51.49dB and -33.43dB respectively. The frequency response of this filter is plotted in figure 7 which shows it to be a scaled extraripple solution (Rabiner and Gold 1975). It should be noted that this filter not only has a precisely linear phase characteristic but also has a well defined passband in which the deviation from unity transmission is minimal in a MINIMAX sense. The filter coefficients are listed in table 1. For comparison the response of a third order integer coefficient filter with N=5 (equation 3) which meets the required stopband specification is shown dotted in figure 7. The response is markedly inferior to the optimal design, maximum deviation in the passband being +1.156dB as opposed to -51.49 dB. In order to achieve a comparable passband deviation from a Kaiser window designed filter would require 37 coefficients rather than the 29 required by the optimal design.

Direct form realisation of a FIR filter requires calculation of the recursion relationship, equation (1). This involves N additions, N multiplications and N-1 shifts per output sample. Use of a moving pointer eliminates the N-1 shifts but then requires a check on the index of every sample to ensure that the storage array is not exceeded. An algorithm for direct form realisation of FIR filters was suggested by Rabiner (1977). The incoming data sample is stored twice hence generating an array of 2N samples for an N point filter. In this way use of a moving pointer is permitted and all of the data samples required to calculate an output sample may be simply addressed relative to the pointer without checking that the index lies within the bounds of the array during the calculation. A slight modification of Rabiners algorithm to take account of impulse response symmetry for a linear phase filter reduces the required number of multiplications to (N+1)/2 and leads to a very efficient software realisation.

Figure 8 shows filtered and unfiltered data for typical 1024 sample records of EEG and EOG taken at a sampling frequency of 128Hz and filtered using a Fortran implementation of the optimal EOG filter based upon the preceding discussion. It is clear from this figure that although the noise in the EOG is considerably reduced,the true EOG remains visibly undistorted. The filtered version of the corresponding EEG record very clearly shows the contamination from the EOG . The filtered EEG amplitude scale has been automatically adjusted from that of the unfiltered EEG by the computer during printing, this also applies to figure 9. Figure 9 shows estimates of relative power spectral density for the records of figure 8. These were obtained by taking the autocorrelation function of the data, windowing with a Turkey window and computing the Fourier transform of the result. The spectral power below 10Hz is seen to be the same before and after filtering whereas there is clearly a reduction in signal power above 10Hz. Studies indicate that an equivalent fixed point routine for the EOG filter in 68000 µp assembly language will require approximately 0.27ms to filter and return a data sample at a clock frequency of 8MHz. Also, assuming EOG samples quantised to 12 bits and utilising 32 bit internal working there is negligible predicted difference in frequency response due to the fixed point computations.

AJS108

move-

H_R

1.000

H_R

1.000

4. CONCLUSION

In conclusion, the design and software realisation of a filter suitable for reducing noise in the EOG without corrupting the ocular information has been presented. It is significant that a fixed point software routine may realise this filter in realtime without detrement to the frequency response. This has been made possible by the efficient algorithm employed and the advent of advanced 16 bit microprocessors. It is hoped that this paper will stimulate greater interest in the methods presented, hence leading to the use of better filters for physiological applications.

TABLE 1

Filter coefficients for EOG lowpass filter.

H(0) = -0.11191774E-01 = H(28)

H(1) = -0.38390798E-02 = H(27)

H(2) = 0.73034801E-02 = H(26)

H(3) = 0.11693427E-01 = H(25)

H(4) = 0.18245798E-01 = H(24)

H(5) = 0.78111943E-02 = H(23)

H(6) = -0.75105876E-02 = H(22)

H(7) = -0.31717509E-01 = H(21)

H(8) = -0.41073106E-01 = H(20)

H(9) = -0.31341068E-01 = H(19)

H(10) = 0.13298627E-01 = H(18)

H(11) = 0.79011545E-01 = H(17)

H(12) = 0.15616089E-00 = H(16)

H(13) = 0.21316704E-00 = H(15)

H(14) = 0.23729908E-00

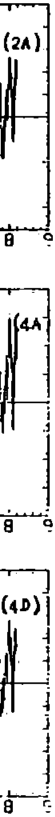


FIGURE LEGEND

FIGURE 1.

Two methods of cancelling phase distortion by time reversal.

$H(e^{j\omega}) = H(Z)\Big|_{Z=e^{j\omega}} = \left|H(e^{j\omega})\right| e^{j\phi(\omega)}$

For method 1: $H_{TR}(Z) = H(Z)H(Z^{-1})$

For method 2: $H_{TR}(Z) = H(Z)+H(Z^{-1})$

FIGURE 2.

Illustration of impulse response symmetry produced by time reversal.

(a) h(n) = positive time impulse response.

(b) h(-n) = negative time impulse response.

(c) $h_{TR}(n)$ = overall time reversal impulse response, h(n)*h(-n)

FIGURE 3.

Effect of truncation on resulting impulse response symmetry.

(a) h(n) = positive time impulse response with h(2) truncated.

(b) h(-n) = negative time impulse response.

(c) $h_{TR}(n)$ = overall time reversal impulse response, h(n)*h(-n)

FIGURE 4.

EOG filter design specifications.

passband deviation δ_1=0.003 (-50dB)

stopband deviation δ_2=0.03 (-30dB)

passband edge frequency,fp=0.078 (10 Hz)

stopband edge frequency,fs≤0.156 (20 Hz)

number of coefficients,N≤37

sampling frequency,Fs=128 Hz

transition width,Δf≤0.078

FIGURE 5.

Range of possible solutions. (shown unshaded)

fs = stopband edge frequency

N = number of filter coefficients.

x = selected initial program inputs.

FIGURE 6.

Deviations returned by design program for fp=0.078 (10Hz), fs=0.148 (19Hz).

δ_1 = passband deviation

δ_2 = stopband deviation

N = number of filter coefficients

FIGURE 7.

Frequency response of optimal FIR filter for EOG with response of a simple integer coefficient filter shown dotted. (a)complete response, (b)passband response.

FIGURE 8.

Typical data for subject repeatedly blinking. (a) unfiltered EEG, (b)filtered EEG, (c)unfiltered EOG, (d)filtered EOG.

FIGURE 9.

Estimated relative power spectral densities corresponding to figure 8. (a)unfiltered EEG, (b)filtered EEG, (c)unfilterd EOG, (d)filtered EOG.

REFERENCES

Cheney E.W.(1966) "Introduction to approximation theory." New York. McGraw Hill.

Coppola R.(1979) "Isolating low frequency activity in EEG spectrum analysis". Electroenceph. and Clin. Neurophysiol. 46 pp224-226.

Lynn P.A.(1977) "Online digital filters for biological signals: some fast designs for a small computer." Med. & Biol. Eng. & Comput. 15 pp534-540.

Lynn P.A. (1983) "Transversal resonator digital filters : fast and flexible online processors for biological signals" Med.& Biol.Eng.& Comput. 21,718-730

McClellan J.H., Parks T.W., Rabiner L.R.(1973) "A computer program for designing Optimum FIR Linear Phase Digital Filters." IEEE Trans. on Audio and Electroacoustics. Vol.Au-21, N0.6. pp506-525

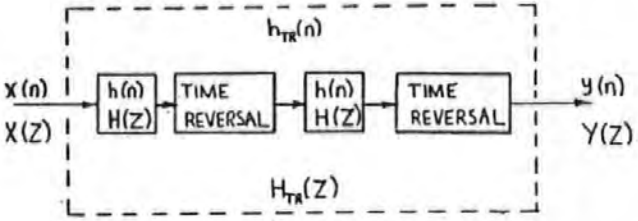
Rabiner L.R.(1977) "A simplified computational algorithm for implementing FIR Digital Filters." IEEE Trans. on Acoustics, Speech and Signal Processing. Vol.ASSP-25 No.3, pp259-261.

Rabiner L.R. and Gold B.(1975) "The Theory and Application of Digital Signal Processing." Prentice Hall .

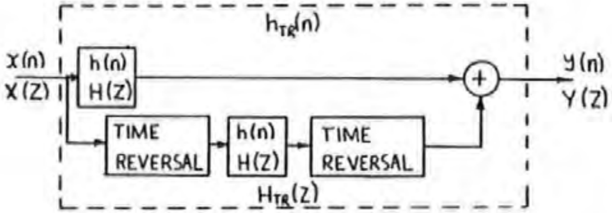
Remez E.Ya.(1957) "General computational methods of Tchebyscheff approximation." Kiev, (Atomic energy translation 4491, pp1-85)

Verburg J. and Strackee J.(1974) "Phaseless Recursive Filtering Applied to Chestwall Displacements and Velocities, using Accelerometers." Med. & Biol. Eng. Vol.12 No.4 pp483-488.

Whitton J., Lue F. and Noldofsky H.(1978) "A Spectral Method for Removing Eye Movement Artefacts from the EEG." Electroenceph. & Clin. Neurophysiol. 44, pp735-741.



method. 1.



method 2.

FIGURE 2

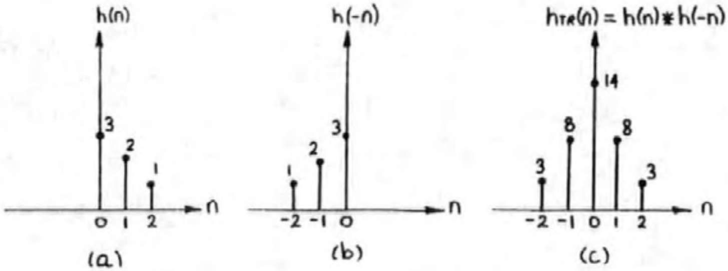
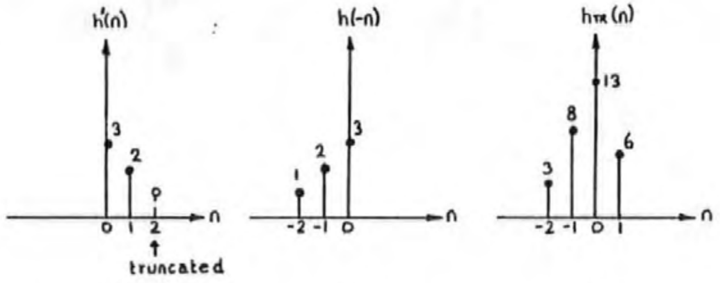


FIGURE 3.



(a) (b) (c)

FIGURE 4

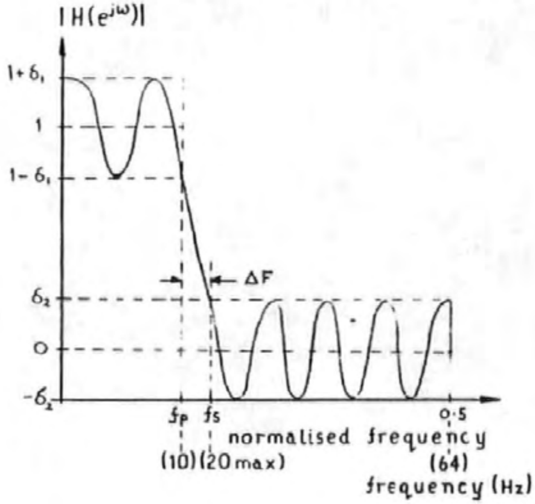

$|H(e^{j\omega})|$
$1+\delta_1$
1
$1-\delta_1$
ΔF
δ_2
0
$-\delta_2$
f_p
f_s
0·5
normalised frequency
(10)
(20 max)
(64)
frequency (Hz)

FIGURE 5

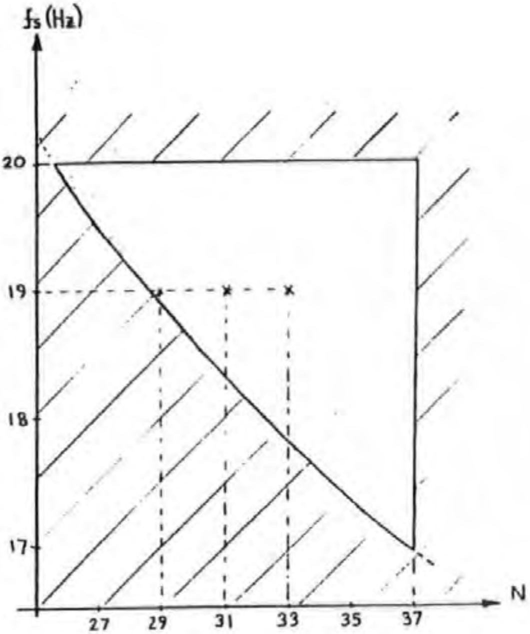


FIGURE 6.

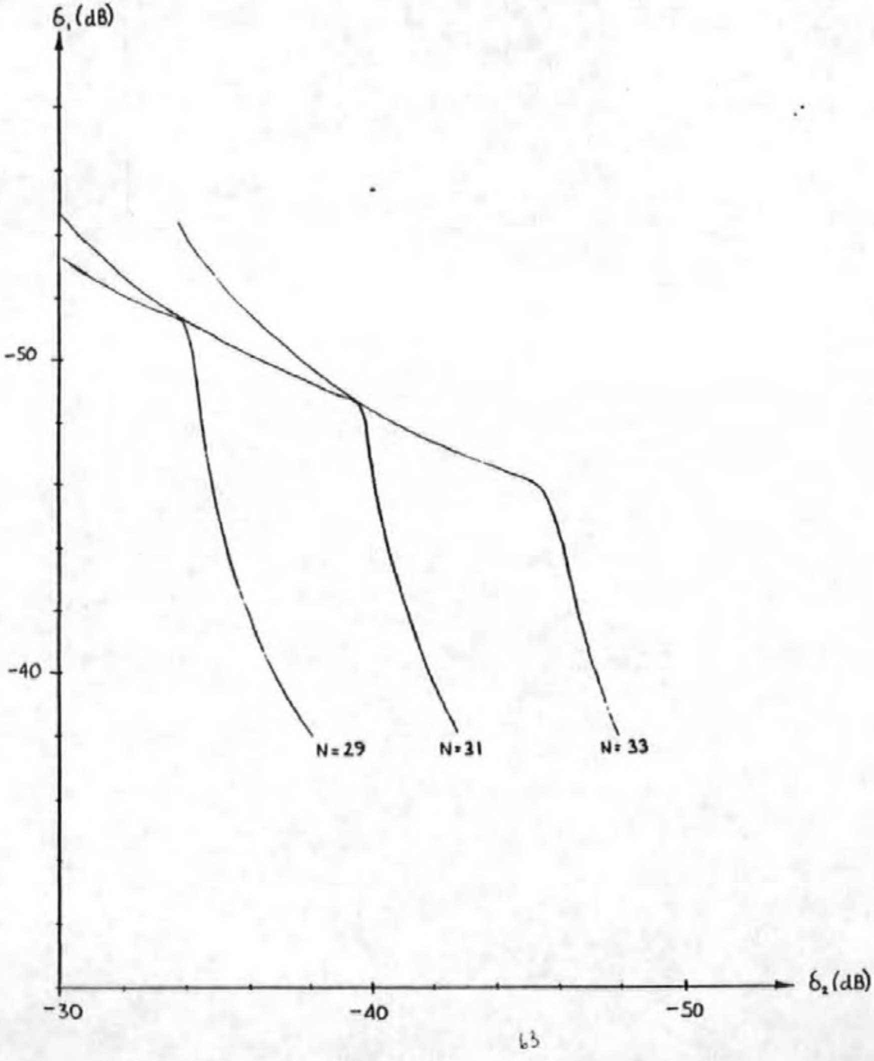


FIGURE 7.

ol.,

tion,

y and

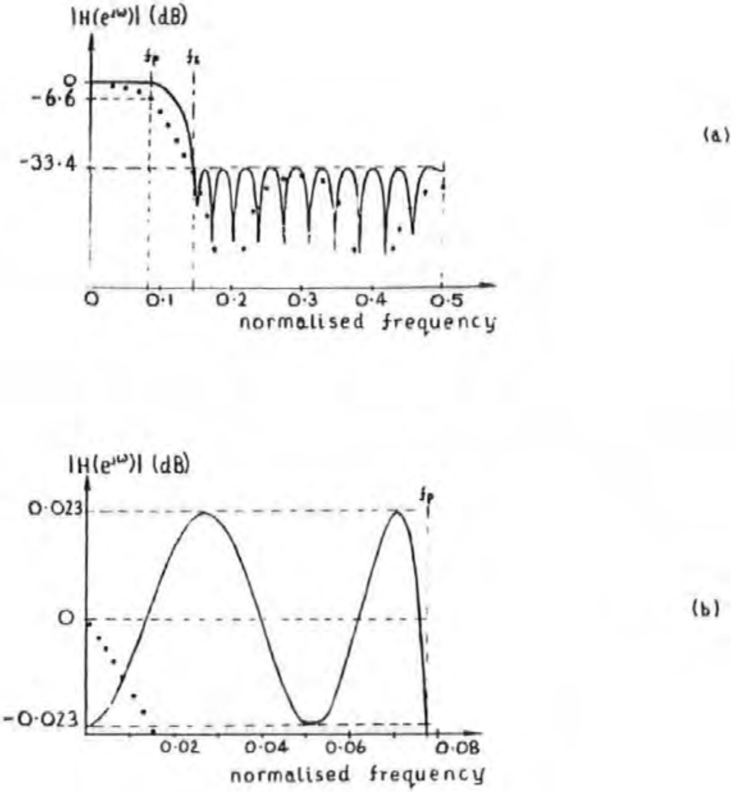
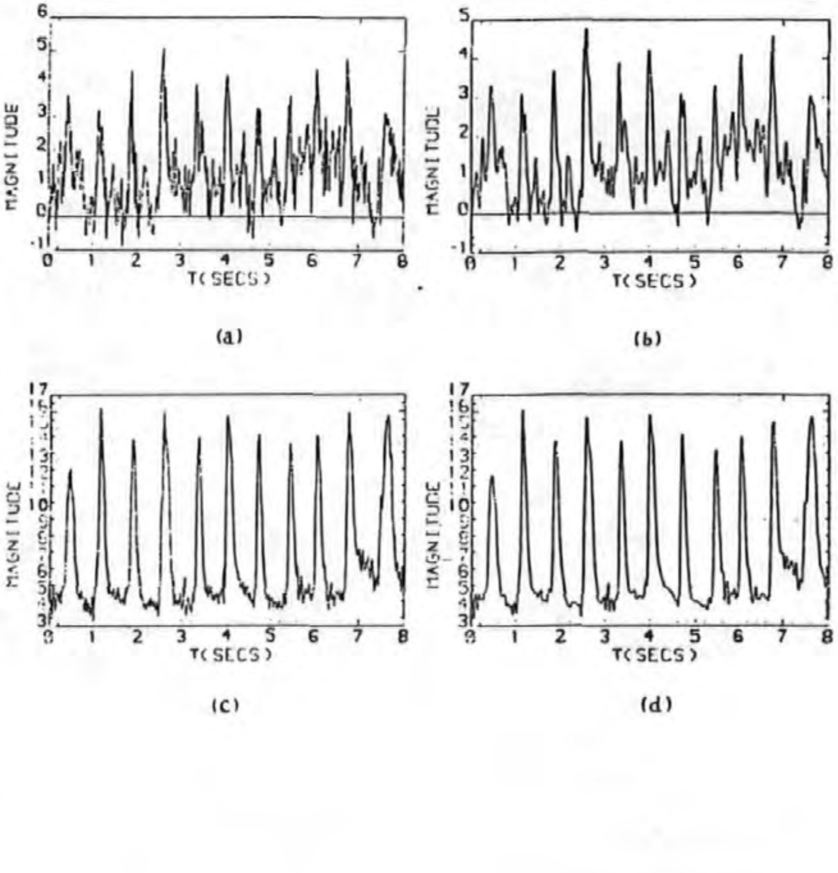


FIGURE 8



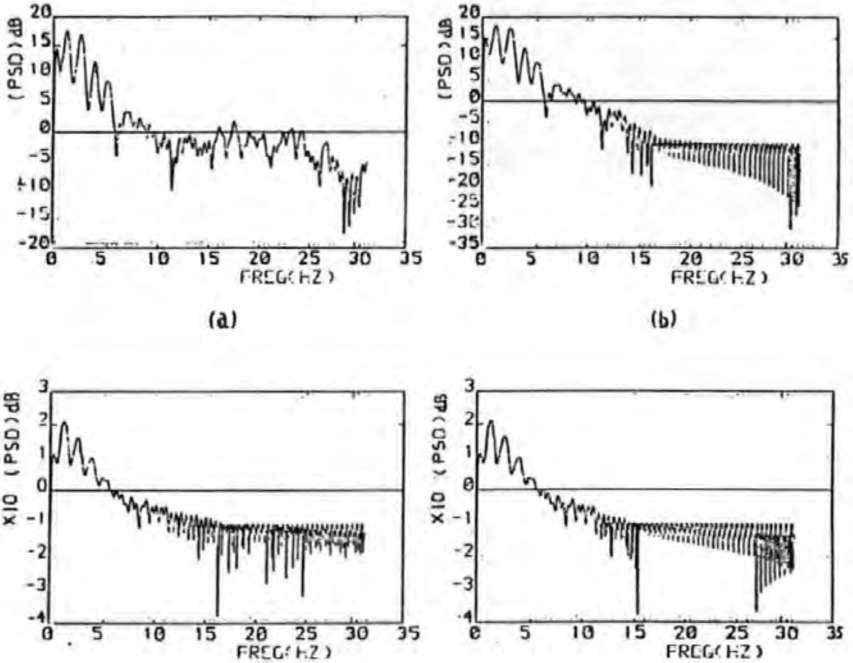
(a)

(b)

(c)

(d)

FIGURE 9



(a) (b)

(c) (d)

INVESTIGATION AND COMPARISON OF SOME MODELS FOR REMOVING OCULAR ARTEFACTS FROM EEG SIGNALS

Emmanuel C. Ifeachor[1], Barrie W. Jervis[2], Elaine M. Allen[3], Earl L. Morris[4], David E. Wright[5] and Nigel R. Hudson[3]

Department of Communication Engineering,
Plymouth Polytechnic,
Drake Circus,
Plymouth, PL 4 8AA.

[1] This author is grateful to Devon County Council for financial support.

[2] Department of Electrical and Electronic Engineering,
Sheffield City Polytechnic,
Pond Street,
Sheffield S1 1WB.

[3] Department of Neurosciences,
Freedom Fields Hospital,
Plymouth.

[4] Department of Engineering Science,
University of Exeter,
Exeter.

[5] Department of Mathematics, Statistics and Computing,
Plymouth Polytechnic,
Drake Circus,
Plymouth.

Comparison of Models for Removing Ocular Artefacts...

Keywords: EEG,Ocular artefacts,Models,Autocorrelation,
Collinearity.

Reprint requests should be sent to:

Dr. B.W. Jervis,
Department of Electrical and Electronic Engineering,
Sheffield City Polytechnic,
Pond Street,
Sheffield S1 1WB.

INVESTIGATION AND COMPARISON OF SOME MODELS FOR REMOVING OCULAR ARTEFACTS FROM EEG SIGNALS

Eye or lid movement produces a change in the electrostatic field associated with the corneo-retinal dipole. This field is propagated with decreasing strength across the scalp. Thus a scalp signal may contain both cerebral signals and ocular artefacts (OA) potentials due to eye and/or eye lid movement. The OA in the Electroencephalogram (EEG) should be removed or at least controlled because it may reduce the clinical usefulness of diagnostic signals and hamper the automatic analysis of the signals (e.g. Mc Callum and Walter 1968;Hillyard 1974;Gotman et al 1975;Quilter et al 1977;Gratton et al 1983).

Several partial solutions have been proposed for removing the OA from the EEG. These can be divided into three categories: Eye Fixation,Rejection and Electrooculogram (EOG) subtraction methods. The eye fixation method does not give satisfactory control of the OA in all cases, but may be used to supplement the other two. In rejection methods, sections of an EEG record where the level of OA exceeds some predefined level are deleted. This may lead to an unacceptably high wastage of data in,for example, studies involving children and uncooperative adults where eye movement (EM) could be uncontrollable. Verleger et al (1982) found that 62% of the data could be lost through rejection and that their EOG subtraction method was superior to the rejection method. OA contaminated records may also contain useful information, and deletion of these sections would both lose data and make it unrepresentative (Gratton et al 1983).

The EOG subtraction method,in our view, offers the most promising way of dealing with the problem of OA in the EEG.

However, the various techniques reported so far do not completely solve the problem; thus, new approaches are continually being developed (e.g Quilter et al 1977; Barlow and Rémond 1981;Gratton et al 1983;Fortgens and De Bruin 1983). The EOG subtraction methods are based on the principle that the OA is additive to the background EEG. Thus in discrete form we have:

$$\begin{aligned} EEG(i) &= OA(i) + e(i) \\ &= \theta EOG(i) + e(i) \end{aligned} \quad (1)$$

where EEG(i) and e(i) are the measured and the true EEGs respectively. θ ,the correction coefficient or OA parameter , is the fraction of the EOG that is measured at the EEG electrode as artefact and EOG(i) is the measured EOG.Thus if the coefficient θ can be estimated then the 'true' EEG can be recovered by subtracting $\hat{\theta}$EOG(i) from EEG(i) in (1). The various EOG subtraction techniques differ primarily in the way θ is estimated, in the number of EOG signals that is used in (1) and the way these are measured (compare for example,Hillyard 1974;Girton and Kamiya 1973;Quilter et al 1977;Verleger et al 1982; Jervis et al 1980).

Verleger et al (1982) suggested that differences in the value of θ between individuals could be due to random errors and not to genuine factors. There are however,differences between subjects which would lead to genuine differences in their OA parameters. These include inter-subject differences in anatomical structures (e.g differences in size and shape of face and head) which would lead to different electrode separations. Inter-subject differences in the values of θ have been reported (e.g. Girton and Kamiya 1973) ,although in some cases they were small (e.g. Gratton et al 1983) and in others they were only significant for blinks (e.g. Corby and Kopel 1972).

Some workers accept that the coefficients should be obtained for each individual,but consider the subject-specific coefficients time-invariant (e.g Girton and Kamiya 1973;Barlow and Rémond 1981). It has been established that blinks and vertical eye movement,for example, require different correction coefficients (Overton and Shagass 1969;Corby and Kopel 1972;Gratton et al 1983). Even when only one type of ocular movement is predominant the position of the eye lids could significantly alter the coefficients (Hillyard 1974). Thus subject-specific coefficients should not be assumed time-invariant. In these approaches, estimates of θ were obtained subjectively by having the subjects move their eyes voluntarily prior to the experiment. However,Gratton et al (1983) pointed out that there may be differences between voluntary and involuntary ocular movements which may lead to differences in the correction coefficients obtained before and during the experiment.

Some recent EOG subtraction techniques have attempted to take into account the factors discussed above (Quilter et al 1977; Jervis et al 1980;in preparation; Gratton et al 1983;Portgens and De Bruin 1983). In these techniques θ is obtained directly from the data. Gratton et al (1983) described a method where eye movement and blink artefacts were separately detected and corrected.This represents some improvements over most other OA removal methods, but it is time consuming in a multichannel situation and where many EOG signals are used, and may only be suitable for off-line use.

It is important to determine whether a model adequately describes the EOG-OA relationships. In particular, is the assumption of a linear relationship between the EOGs and the

OA reasonable? Which and how many EOG signals should be used to obtain optimum removal of OA? Girton and Kamiya (1973) and Quilter et al (1977) pointed out that to obtain a reasonably good estimate of OA, more than one EOG signal should be used. This view is confirmed by our results. Jervis et al (1980; in preparation) extended Quilter et al's approach to three and four parameter models (utilising three and four EOGs respectively)and found that the three parameter model was better than the 2 parameter model.They also found that the four parameter model was best. Fortgens and De Bruin (1983) reached similar conclusions . However, there should be no need to use all the EOGs in a model because often they are highly correlated, and our results confirmed this.

There remained a need to carry out a systematic comparison of the various models to establish their relative effectiveness in removing OA from the EEG. This paper extends the work of Jervis et al (1980; in preparation) to find the most effective model for implementation on-line. A number of models were studied and their effectiveness of correction compared.

A common assumption in the least squares method,to which the EOG subtraction method belongs, is that the error terms e(i) (which represent the background EEG) are uncorrelated. We found that the assumption is not valid and give a simple method to reduce the effects of correlation on the parameter estimates.

METHODS AND MATERIAL

1 Apparatus,subjects and Data Acquisition

The data used in this investigation was obtained in a previous study (see Jervis et al,1980,in preparation). Separate vertical and horizontal EOG signals were recorded for each eye from silver-silver chloride electrodes placed above and below the eye, and from similar electrodes at the outer canthus and the nasion. This electrode placement gave maximum information about ocular movements and allowed the horizontal and vertical components to be measured separately. The EEG signal was derived from silver-silver chloride electrodes placed at the vertex referred to linked earlobes. These signals were fed into an 8-channel Electroencephalograph to produce a paper chart record and into a data acquisition system (DAS). In the DAS, the signals were first high pass filtered (cut off frequency, 0.16 Hz) to minimize drift, low pass filtered (cut off frequency, 30 Hz) to filter out mains frequency and to reduce aliasing, and then digitized.

Volunteers with no known eye defects were asked to make periodic eye movements in the vertical,horizontal and diagonal directions between pairs of light emitting diodes located on a wooden screen. This corresponded to about 20° of eye movement. The Subjects were also asked to blink at regular intervals. This ensured that the performance of the models were evaluated under a given type of OA.

Several 8-sec lengths of EEG records (each record consisted of 1024 samples from each of the four EOG and the EEG channels) were acquired for each subject and for each type of ocular movement (vertical eye movement (VEM),horizontal eye movement (HEM), diagonal eye movement (DEM) and blink) and stored on a disk under the control of a PdP 8 minicomputer. The acquired data was later transferred to a main frame computer for analysis.

2 Data Preprocessing

The mean of the data samples of each channel was computed and removed from each sample. Any low frequency trend was removed by differencing. Differencing also helped to reduce the effects of serial correlation of the background EEG on the estimates of θ. This gave two sets of data for each EEG record viz:

(a) Data segments with the mean removed (raw data).

(b) Data segments from (a) and also differenced (differenced data).

3 Some Properties of the Measured Data

A preliminary analysis of the data segments of each record was carried out to identify those properties of the data that could affect the results of OA removal. Three properties of interest were, the relationship between the EEG and the ocular artefacts, the linear dependencies between the EOGs, and the EOG artefacts. Scatter diagrams were used to study the relationships between the ocular artefacts and the EEG, and between the EOGs themselves. To assess the extent of the linear dependencies (or collinearity) between the EOGs we examined their correlation matrices and the associated eigenvalues. Strong collinearity is accompanied by one or more very small eigenvalues and can lead to difficulties in computation and statistical analysis (Silvey,1969;Johnston 1972;Chatterjee and Price 1977).

The EOG may contain other artefacts that are not related to ocular movement, such as electromyographic (EMG) signals resulting from muscle action and contamination of the EOG by the frontal EEG signals. Because in the EOG subtraction method we merely subtract fractions of one or more EOG signals from the measured EEG, any artefacts in the EOGs would be introduced into the corrected EEG as 'secondary artefacts'. We studied this problem by examining plots of the preprocessed data.

4 Description of Models and the removal algorithm

(a) Models used in the investigation

The form of the models used is given in equation (2)

$$\mathrm{EEG}(i) = \theta_1 V_L(i) + \theta_2 V_R(i) + \theta_3 H_L(i) + \theta_4 H_R(i) + e(i) \qquad i = 1,..M \qquad (2)$$

where EEG(i),e(i) are the ith samples of the measured and background EEG respectively. $V_L(i),H_L(i)$ are the vertical and horizontal left EOGs respectively. $V_R(i),H_R(i)$ are the corresponding right EOGs and M is the number of data points for each variable. θ_1 to θ_4 ,the correction coefficients, are the fractions of the corresponding EOGs that reach the EEG electrode as artefact.

The following are the variations of (2) which were used in the investigation:

Model name	EOG's used in (2)	
2A	V_L, H_L	$(\theta_2 = \theta_4 = 0)$
2B	V_L, H_R	$(\theta_2 = \theta_3 = 0)$
2C	V_R, H_L	$(\theta_1 = \theta_4 = 0)$
2D	V_R, H_R	$(\theta_1 = \theta_3 = 0)$
2E	V_L, V_R	$(\theta_3 = \theta_4 = 0)$
2F	H_L, H_R	$(\theta_1 = \theta_2 = 0)$
2G	$V_L, (H_L-H_R)$	$(\theta 3 = \theta_4 = 0; V_R = (H_L-H_R))$
3A	V_L, H_L, H_R	$(\theta_2 = 0)$
3B	V_L, V_R, H_L	$(\theta_4 = 0)$
3C	V_L, V_R, H_R	$(O_3=0)$
3D	V_R, H_L, H_P	$(O_1=0)$
4A	V_L, V_R, H_L, H_R	
4B	$V_L, (V_L x V_L), H_L, H_R)$	$(V_R = (V_L x V_L))$
4C	$V_L, (V_L-V_R), H_L, H_R$	$(V_R = (V_L-V_R))$
4D	$V_L, (H_L x H_R), H_L, H_R$	$(V_R = (H_L x H_R))$

Model 2A is an approximation of the model used by Quilter et al (1977).Model 2G simulates what appears to be a popular two-parameter model (e.g Barlow and Rémond 1981).It has the advantage of having reduced coupling between the vertical and the horizontal

EOG signals,but combines the EOGs of both eyes which may be different (Quilter et al,1977). Models 3A and 4A were used by Jervis et al (1980;in preparation). The latter is also an approximation of the model used by Fortgens and De Bruin (1983). Model 4B attempts to correct for possible non-linearities which may be present. Model 4C attempts to minimize the effects of dependencies between the EOGs (due to coupling and conjugate eye movement) ,while model 4D attempts to compensate for possible non-linear interaction between the horizontal EOGs. It has been suggested that the ocular dipoles tend to oppose each other in their effects during HEM (Jervis et al;in preparation). Other models were included to ensure that all 'possible' models were compared.

Additionally, four single parameter models, each containing one of the four EOG signal, were used to correct for ocular artefact. The aim here was to verify whether or not single parameter models could satisfactorily remove ocular artefacts from EEG signals.

(b) OA Removal Algorithm -the Ordinary Least Squares Method

Assuming that e(i) is an uncorrelated sequence with zero mean and constant variance then optimum estimates of the coefficients θ can be obtained, using the ordinary least squares (OLS) method, by minimizing the sum of squares of e(i) in (2). This leads to equation 3 (see the appendix):

$$\hat{\theta} = (X^T X)^{-1} X^T Y \qquad (3)$$

where $\hat{\theta}$,the parameter estimate (estimate of the correction coefficients),is given by: $\hat{\theta} = \left[\hat{\theta}_1 \ \ \hat{\theta}_2 \ \ \hat{\theta}_3 \ \ \hat{\theta}_4\right]^T$

X,the EOG matrix, and Y a vector of EEG samples, are respectively:

$$X = \begin{bmatrix} V_L(1) & V_R(1) & H_L(1) & H_R(1) \\ V_L(2) & V_R(2) & H_L(2) & H_R(2) \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ V_L(M) & V_R(M) & H_L(M) & H_R(M) \end{bmatrix} \qquad Y = \begin{bmatrix} EEG(1) \\ EEG(2) \\ \cdot \\ \cdot \\ \cdot \\ EEG(M) \end{bmatrix}$$

Thus $\hat{\theta}$ can be obtained using any suitable matrix inversion technique Having obtained $\hat{\theta}$, estimates of the OA and hence the background EEG can be obtained as:

$$\hat{E} = Y - X\hat{\theta} \qquad (4)$$

where $\hat{E} = \begin{bmatrix} e(1) & e(2) & \ldots & e(M) \end{bmatrix}^T$

The estimates of the error terms or the background EEG, $\hat{E}$, are referred to as the residuals. Implicit in the OLS method is the assumption that the EOGs are not perfectly collinear. If they are then the inverse of the matrix $(X^T X)$ will not exist and the parameter estimates $\hat{\theta}$, cannot be obtained.

It is worth noting that all the EOG subtraction methods that appear in the literature under different names are essentially least squares methods and thus subject to the above assumptions. For example,we show in the appendix that the correlation technique method used by Quilter et al (1977) and Jervis et al (1980,in preparation) is in fact an OLS method.

5 Criteria used to Compare the Performance of the Models

Quantitative and pictorial criteria were used in the comparison of models. They were intended to be complementary.

(a) Minimization of the Sum of Squares of the EEG

As discussed above, each model minimizes the sum of squares of the error term . Thus minimum sum of squares of the estimates of the error terms (or residual sum of squares) can be used as a criterion for comparing the models. To take into account the decrease of the residual sum of squares with the number of parameters in a model,we used instead the residual variance given by (5) (Weisberg 1980).

$$S_p^2 = Rss_p/(M-p-1) \qquad (5)$$

where Rss_p and p are respectively,the residual sum of squares and the number of parameters in the model. Models with small residual variances are considered good models. (e.g. Weisberg,1980).

(b) Mallows' Statistic, C_p.

A model is a subset of a larger model if the latter contains all the variables in the former plus one or more variables. Subset models yield biased estimates except when the variables omitted are unimportant. Thus a subset model is to be preferred if the bias of its estimates is small. A criterion that can be used to choose between models is Mallow's statistic, C (Mallow 1973, Hocking 1976,Weisberg 1980).

$$C_p = \left(Rss_p/S_K^2\right) + 2p - M \qquad (6)$$

where S_K^2 is the residual variance from the full model,here taken to be model 4A. A good model should have a C_p value close to p,which in our case should be no more than about 4 as no model has more than four parameters. If models have C_p values that differ by only small amounts, they may have comparable performance. Therefore,we used a somewhat more relaxed criterion to ensure that all 'good' models were included. Thus a model was 'good' if its C_p value was about ten or less. After studying several records a pattern emerged and the 'good' models were then subjected to further analysis.

(c) OA Estimates and their differences as measures of performance

A complementary method used to study the performance of the models was to compare plots of OA estimates from the various models and also the differences between these estimates. The difference between the OA estimates of any two models is equivalent to the difference between the corresponding corrected EEGs. This approach may be described as a pictorial form of 'F-Test' and it enabled us to judge visually the effects of adding extra terms in the model. To relate the differences to EM, plots of the autocorrelation functions (ACF) of the differences may also be obtained.

6 Autocorrelation in the Background EEG and Durbin-Watson Statistic

A crucial assumption in the least squares method is that the error terms, e(i),are uncorrelated .If this assumption is violated then the parameter estimates obtained by the OLS method are innefficient as they no longer have minimum variance, and tests of significance based on the sum of squares are no longer strictly valid. (e.g. Cochrane and Orcutt 1949;Johnston 1972;Chatterjee and Price 1977).

The Durbin-Watson statistic,d, (Durbin and Watson 1950;1951), defined in (7), can be used to test for the existence of autocorrelation in the error terms. Since in general the error terms are not directly available, the test is applied to the residuals (the estimates of the error terms).

$$d = \frac{\sum_{i=2}^{M} \left[e(i) - e(i-1) \right]^2}{\sum_{i=2}^{M} e^2(i)} \qquad (7)$$

d is close to zero when the residuals are highly positively correlated and to 2 when they are not. Tables giving the upper and lower bounds (dL,dU) for d for different percentage points are available for testing the significance of d (e.g Johnston,1972). If d is significant then some remedial action should be taken.

Correcting for Autocorrelation

A significant value for d might be due to model inadequacy or autocorrelation in the errors. If the model is inadequate due to the omission of some important variables, or a linear model is specified where in fact the relationship between the dependent and independent variables is nonlinear, then the residuals from such a model will be correlated. The simple solution in this case is to use the correct model.

However, if the error terms are correlated then the OLS method will always give correlated residuals and this can be corrected by transforming the data. A way of doing this is to represent the residuals as an autoregressive (AR) series.
A form commonly used is:

$$e(i) = \sum_{j=1}^{n} \phi_j \, e(i-j) + a(i) \qquad (8)$$

where n is the order of the AR model, ϕ the autoregressive parameters, and a(i) an uncorrelated sequence.
If n is set to unity then we have the first order AR series which is equivalent to transforming the data in the following way (see the appendix):

$$\bar{y}(i) = \bar{x}^T(i)\,\theta + a(i) \qquad (9a)$$

where $\bar{y}(i) = y(i) - \phi y(i-1); \; \bar{x}^T(i) = x^T(i) - \phi\, x^T(i-1)$ (9b)

$y(i) \equiv$ the measured EEG; $x^T(i) \equiv \left[V_L(i) \;\; V_R(i) \;\; H_L(i) \;\; H_R(i)\right]$, the measured EOGs.

An estimate of θ can be obtained iteratively using the following algorithm (e.g. Cochrane and Orcutt,1949;Chatterjee and Price,1977).

Step 1. Obtain OLS estimates of θ and hence the residuals from

$\hat{e}(i) = y(i) - x^T(i)\,\hat{\theta} \quad i=1,2..,M$

Step 2. Use OLS to estimate ϕ from : $e(i) = \phi e(i-j) + a(i)$

Step 3. Obtain a new set of variables ($\bar{x}^T(i)$ and $\bar{y}(i)$) as in (9b).

Step 4. Use OLS to obtain $\hat{\theta}$, an estimate of θ from (8a) and hence a new residual $a(i)=\bar{y}(i) - \bar{x}^T(i)\hat{\theta}$. Then go back to step 2 and continue until the estimate of ϕ converges.

When ϕ converges, $\hat{\theta}$ is then the estimate of the parameters and is used in (2) to obtain estimates of the corrected EEG. If e(i) is uncorrelated then $\phi = 0$ and both the OLS and the iterative methods lead to the same results. It is to be noted that differencing used in the data preprocessing is a special case of the first order AR model,obtained by setting both n and ϕ to unity in (8).

In the investigation we used all three estimation methods, viz. OLS, differencing and iterative methods.

RESULTS.

1. Properties of the Measured Data

(a) The relationships between the ocular artefacts and EOGs

Typical plots of the scatter diagrams of the EOGs and the EEG are given in figure 1 for a horizontal eye movement record. Examination of these and similar plots for other ocular movements showed that there was sometimes a linear relationship between the EOGs and the OAs, but that this was not true in all cases.

(b) Collinearity between the EOGs

(i) The correlation coefficients between all pairs of the EOG variables were estimated for several EEG records and a representative selection of the correlation matrices and their eigenvalues are given in table I for different ocular movements. Examination of the table showed that in each case there was high correlation between some of the EOGs, and at least two of the eigenvalues were very small. In fact, only the first two eigenvalues in each case accounted for nearly 99% of the total variability $((\lambda_1+\lambda_2) \times 100\%/(\sum_{i=1}^{4}\lambda_i))$, suggesting strong collinearity and that about two EOGs were redundant.

For the differenced data ,the correlation coefficients were much smaller than for the raw data and no eigenvalue was too small. For example, after differencing the eigenvalues for the record given in table I(a) were :$\lambda1=2.630$,$\lambda2=0.647$,$\lambda3=0.561$, $\lambda4=0.162$. This suggested that differencing could be used to reduce the effects of collinearity.

For all types of ocular movement except lateral eye movement, the vertical EOGs were more strongly correlated than the horizontal EOGs. The correlation coefficients,for the differenced data, were typically 0.6 for the horizontal EOGs for HEM and 0.75 for the vertical EOGs for blink or VEM.

(ii) Scatter plots of the EOGs against one another showed linear relationship between the EOGs, especially between the two vertical EOGs and between the horizontal EOGs .

(c) EOG Artefacts

Several EOG and EEG waveforms for both raw and differenced data, were studied visually and were always contaminated by 'noise'. This may have been due to EEG contamination or electromyographic signals etc. It was found that for VEM and blink records the EOG level relative to noise was smaller for the horizontal channels than for the vertical channels.The opposite was true for HEM.

In most vertical eye movement (VEM) and occasionally in horizontal eye movement records rider artefacts could be seen. These are transient potentials,similar in characteristics to blinks, that are superimposed on the normal EOG,especially during vertical eye movements with the eyes open (Barry and Jones 1965;

high correlation was not the omission of a variable. Similar results were obtained for all types of ocular movement.

A study of the residuals (or corrected EEGs) and their ACFs suggested that the error terms e(i) behaved like an autoregressiv series (see for example, Box and Jenkins 1970).The residuals behaved like 'random walks' and their ACFs decayed slowly to zero Figure 2(i) and 2(ii) give typical examples for a HEM record. Figure 2(iii) to 2(vi) give similar plots but for the differencin and the iterative methods, and these show that all systematic effects have been removed and the ACFs decreased rapidly to zero. This was true for all types of ocular movements and was taken as an indication that the autoregressive representation for e(i) was reasonable. Table III(b) gives results for the test statistics fo the differencing method and showed that the d statistics were nearer 2 than those for the OLS method. Table III(b) also showed that only models 2D,3C,3D,4C and 4D satisfied our C_p criterion of about ten or less and, in this instance, are the 'good' models. Similar conclusions can be drawn from the values of the variance, but C_p values are easier to interpret.

Table IV gives a summary of the results based on the C_p values for both the OLS and the differencing methods for the various ocular movements for each subject. From the table it is seen that the following models performed well in the OLS and differencing methods, respectively: (3D,4C and 4D) and (3D,4C). We note that these models contained (H_L,H_R) and one or the other but not both vertical EOGs,except model 4C which contained V_R indirectly. It appears that there is no need for more than one vertical EOG signal in a model. It is to be noted that models containing the EOGs of the right eye performed better than those containing the EOGs of the left eye. (compare for example, Matsuo et al 1975).

2 Single Parameter Models

The four single parameter models were used to correct several EEG records for OA, and in many cases the corrected EEGs still contained a vestige of the EOGs. It was found that for blinks and vertical eye movements, the models that used vertical EOGs performed better than those using horizontal EOGs and vice versa for lateral eye movements.

The mean of the absolute values of the parameter estimates and their standard deviations for two single parameter models (using V_R and H_R respectively) are given in table II for two subjects. The values of the standard deviations indicated that for VEM,blink and DEM, the single parameter model which used V_R gave more reliable estimates than the model that used H_R. The opposite was true for lateral eye movement. It is seen that the estimates for the OLS and the differencing methods differed,but only by small amounts if the standard deviations are taken into account. Similar results were obtained for models using V_L and H_L,respectively.

3 Model Comparison

(a) Quantitative comparison of models

Several EEG records were corrected for OA with the multi-parameter models defined earlier using the OLS method. In each case and for each model the statistics d,S_p and C_p were computed. Table III(a) lists the models and these statistics for a typical case. Examination of table III(a) showed that the values of the d statistics were very low for all the models, suggesting that the residuals were highly correlated, and test of significance was hardly necessary. As all models were affected, including those that contained all the EOGs, this suggested that the cause of the lateral eye movement. For vertical eye movement and blink it was found that models using more than two EOGs did not yield significant improvement, especially if one of the EOGs already included was a vertical one.

The corrected EEGs corresponding to the OA estimates given in figure 3 are shown in figure 5. Some vestige of the EOG can be seen in these waveforms particularly in that of the single parameter model. It is note worthy that the differences between these EEGs are not as easily seen as they are from figure 4, illustrating the sensitivity of our pictorial criterion. A common feature in all these models was that rider artefact, when it occurred, was hardly ever completely removed. Peaks were often observed in the corrected EEG at points corresponding to where the rider artefacts occurred in the raw data.

Similar results to those given above were obtained for the differencing and iterative methods.In these cases,however, the improvements were more pronounced. However,model 4D did not perform as well as in the OLS method. Figure 6 gives the OA estimates for model 3D for the three methods and the differences between these estimates. The figure showed that the differencing and iterative methods differed only in the size of their OA estimates,whereas the OLS method gave results that were dissimilar to either methods.

DISCUSSION

(a) Autocorrelation in the EEG

Our results showed that the background EEG is highly correlated and thus violates one of the assumptions of the least squares method. Thus optimum estimates of the correction coefficient cannot be obtained. Although we have reduced this problem by modelling the EEG as a first order autoregressive (AR) models 2A and 2D or 3A and 3D).

The iterative method was applied to some of the 'best models'. It was found that convergence of the parameter estimates was obtained after about two to three iterations. Results very similar to those of the differencing method were obtained. For comparison we present results for model 3D using the three methods in table V. Inspection of the table showed that the results for the differencing and iterative methods were in general similar,but differed from those for the OLS method. Compare for example, the values for O_3 in table V(c) for the three methods. This is to be expected as ϕ was nearly the same for the differencing and iterative methods so the methods were nearly equivalent.

(b) Pictorial Comparison of Models.

Figures 3 and 4 show,respectively, the OA estimates and the differences between these estimates for a horizontal eye movement record corrected by the OLS method. Figure 3 showed that not all the OA estimates were the same in amplitude and shapes, suggesting that the models corrected the EEG to different extents. It was also found that the OA for models 4A and 4C in general contained much higher level of noise than any other model. This was probably because the EOGs used in these models contained more secondary artefacts. Refering to figure 4,we note that all but figures 4(iv) and 4(v) contained EOG related signal in the form of a rectangular wave and only little noise. Figure 4(iv) and (v), associated with models that use all four EOGs (models 4A and 4C),contained significant levels of noise. This suggested that hardly any improvement was achieved by using all four EOGs in a model. Figure 4(vi) shows that only small improvements was achieved by including square terms (in this case, H_L x H_R) in the model. These results hold for series,higher order AR models may give better estimates (see for example, Isakson 1981). However, there are drawbacks in using AR models. Firstly,the data points have only a limited accuracy, so that differencing leads to a reduction in the accuracy of the data. Secondly, these operations are equivalent to high pass filtering and therefore,the correction coefficients are effectively determined only by the changes in the data. This in itself is not a bad thing since the detection and removal of changes in potential due to ocular movement and not eye fixation is our goal in OA removal.

(b) Collinearity

Our investigations have shown that for any ocular movement at least a pair of EOGs were highly correlated. The correlation is probably because in most cases both eyes move in unison and there is no isolation between the vertical and horizontal measuring sites. It is possible therefore, that for some ocular artefacts (e.g blink and vertical eye movement) a single parameter model is sufficiently sensitive for satisfactory correction. However, when the best possible correction is required or when there is less correlation between the EOGs as in lateral and random eye movements more EOGs are required. However, when additional EOGs are used computational difficulties due to collinearity may arise (e.g Johnston 1972). If the EOGs were strongly collinear, as in our case,it is not possible to estimate the exact contribution of each EOG signal and the parameter estimates are less reliable (Farrar 1967;Johnston 1972;Silvey 1969).

(c) Comparison and choice of Models.

A good model should include at least a horizontal and a vertical EOG signal. On the basis of our results it appears that these should be the horizontal and the vertical right EOGs. As mentioned earlier, there is no need to use both vertical EOGs because they are nearly always highly correlated and may introduce significant secondary artefacts into the EEG. Account should also be taken of the various observed effects in the EEG and EOGs, such as the opposing effects of the eye dipoles during lateral movements (Jervis et al;in preparation), and the slight curvatures in the scatter diagrams. We chose model 4D (which used $V_L, H_L x H_R,$ H_L, H_R) for implementation on-line, but with hind-sight thought that a model that used ($V_R, H_L x H_R, H_L, H_R$) would be a better choice. A brief investigation confirmed this to be so. However, if the concept of "fewer is better" is used then model 3D (which uses V_R, H_L, H_R) should be used.

Jervis et al (1980;in preparation) compared a number of models which included model 4A (which uses all the EOGs) and some of its subsets, using a quantitative test based on the autocorrelation functions of the corrected EEGs. They found that model 4A was the best. However,the presence of strong serial correlation in the EEG and the large secondary artefacts associated with this model could significantly affect the ACF of the corrected EEG and make the quantitative results unreliable. Fortgens and De Bruin (1983) reached similar conclusions as Jervis et al (1980; in preparation) by considering 2- and 4-parameter models. They reported that at least four EOGs were required for adequate removal of OA, especialy in the frontal regions. They also showed theoretically that four EOG signals were necessary, but we have shown that the EOGs are correlated so that four EOGs are unnecessary.

The OA models used in this investigation probably do not remove completely OA which is due to rider artefacts. For the rectangular EOGs used, rider artefacts tended to occur at twice the frequency of the EOGs and appeared as peaks in the corrected

EEG . Examination of a number of corrected EEGs from records that contained rider artefact suggested that this may be the source of the second harmonic reported by Jervis et al (in preparation). The inability of the OA models to deal with multiple artefacts is mainly because all data points are given equal weight in the estimation of O and only simultaneous changes in the EOGs and the EEG signals are taken into account. Some form of dynamic model that does not give equal weight to all M data points would appear to be more appropriate. Models encorporating some of these ideas have been simulated on a main frame computer and will be the theme of a future paper in which attention will be focused on on-line removal of OA.

SUMMARY

This paper compares the performance of several models for removing ocular artefacts from EEG signals using quantitative and pictorial criteria. These include most of the models reported in the literature and some new ones. It was shown that for most ocular movements there were at least a pair of EOGs that were strongly correlated. Thus there is no need to use all four EOGs (one vertical and one horizontal for each eye). Using the criteria it was found that regardless of the type of ocular movement,there was always a smaller model that has a performance that is comparable or better than the full model. Overall the model using the vertical right EOG (V_R), the two horizontal EOGs (H_L and H_R) and their product (H_L x H_R) gave the best performance. The model using the vertical right EOG and the two horizontal EOGs was found to be the best three-parameter model. This updates the conclusions of Jervis et al (1980;in preparation) and Portgens and De Bruin (1983).

The common assumption,implicit in all EOG subtraction methods, that the error terms (which represents the background EEG) are uncorrelated was shown to be invalid. The effects of this invalid assumption on the ocular artefact parameter estimates were reduced by modelling the error terms as a first order autoregressive series.

ETUDE ET COMPARAISON DE QUELQUES MODELES SERVANT A SOUSTRAIRE LES ARTEFACTS OCULAIRES DES SIGNAUX EEG

RESUME

Cet exposé compare le fonctionnement de plusieurs modèles servant à supprimer les artefacts oculaires dans les signaux EEG en utilisant des critères quantitatifs et imagés. Ceux-in comprennent la plupart des modèles mentionnés dans la documentation plus quelques uns de nouveaux. Il a été démontré que pour la plupart des mouvements oculaires, il y avait au moins deux EOG qui avaient une corrélation étroite. Il n'est donc pas nécessaire d'utiliser les quatre EOGs (un vertical et un horizontal pour chaque œil), En utilisant les critères, on a constaté que, quel que soit le type de mouvement oculaire, il y avait toujours un modèle plus petit dont le fonctionnement était comparable ou supérieur au modèle grandeur nature. Dans l'ensemble, le modèle utilisant le EOG vertical droit EOG (V_R) les deux EOGs horizontaux (H_L èt H_R) et leur produit ($H_L x H_R$) fournissait le meilleur résultat. Le modèle utilisant le EOG vertical droit et les deux EOGs horizontaux s'avéra être le modèle aux trois paramètres le meilleur. Ceci met à jour les conclusions de Jervis et al (1980; en préparation) et Fortgens et De Bruin (1983).

L'hypothèse commune, implicite dans toutes les méthodes de soustraction EOG, selon laquelle les termes d'erreur (qui représente EEG de fond ne sont pas liés, s'est révélée être non valable. Les effets de cette fausse hypothèse sur les évaluations du paramètre artefact oculaire sont réduits par le modelage des termes d'erreur comme une série autoregressif de premier ordre.

REFERENCES

Barry,W. and Jones,G.M. Influence of eyelid movement upon electroculographic recording of vertical eye movements. Aerospace med.,1965,36:855-858.

Barlow,J.S and Remond,A. Eye movement artefact nulling in EEGs by multichannel on-line EOG subtraction. Electroenceph. clin. Neurophysiol.,1981,52:418-423.

Box G.E.P and Jenkins G.M. Time Series Analysis forecasting and control Holden-Day,1970

Chatterjee S. and Price B. Regression Analysis By Example. John Wiley & Sons,Inc.,1977.

Corby,J.C. and Kopel,B.S. Differential contributions of blinks and vertical eye movements as artifacts in EEG recordings. Psychophysiology.1972,9:640-644.

Cochrane,D. and Orcutt,G.H. Application of Least squares regression to relationships containing autocorrelated error terms. J. Amer. Stat. Assoc. 1949,44:32-61.

Durbin,J. and Watson,G.S. Testing for serial correlation in least squares regression I. Biometrika,1950,37:409-428 and 1951, 38:159-178.

Farrar,D.E and Glauber,R.R. Multicollinearity in regression analysis: the problem revisited. The review of Econ. Stat.,1967,49:92-107

Fortgens,C. and De Bruin M.P. Removal of eye movement and ECG artifacts from the non cephalic reference EEG. Electroenceph. clin. Neurophysiol.,1983,56:90-96.

Gotman,J.,Gloor,P. and Ray,W.F A quatitative comparison of traditional readings of the EEG and interpretation of computer extracted features in patients with supertentorial brain lessions. Electoenceph. clin. neurophysiol.,1975,38:623-639.

Girton,D.G. and Kamiya,J. A simple on-line technique for removing eye movement artefacts from the EEG. Electroenceph. clin. Neurophysiol. 1973,34:212-216.

Gratton,G.,Coles,M.G.H. and Donchin,H. A new method for offline removal of ocular artefacts. Electroenceph. clin. neurophysiol. ,1983,55,468-484.

Hillyard,S.A Methodological issues in CNV research. In Bioelectric recording Techniques part B, Electroencephalography and Human Brain Potentials, (ed) R.F Thompson and M.M. Patterson, Academic Press,1974,259-280.

Hocking, R.R. The analysis and selection of variables in linear regression, Biometrics,1976,32:1-49.

Isaksson,A., Wennberg,A and Zetterberg L.H Computer analysis of EEG signals with parametric models. Proc. IEEE,1982,69:451-461.

Jervis B.W.,Nichols,M.J.,Allen,E.,Hudson,N.G. and Johnson,T.E. The Quantitative assesment of electroencephalograms corrected for eye movement artefacts. First European signal processing conference,Lausane,september,1980.

Jervis B.W.,Nichols,M.J.,Allen,E.,Hudson,N.G. and Johnson,T.E. Comparison of two methods for removing eye movement artefact from the EEG signals.(in preparation)

Johnston J. Econometric Methods. 2nd ed,McGraw-Hill,New York,1972.

Mallows, C.L. Some comments on Cp,Technometrics,1973,15:661-675

Matsuo,F.,Peters,J.F. and Reilly,E.C. Electrical phenomena associated with movements of the eyelid. Electroenceph. clin. Neurophysiol.,1975,38:507-511.

McCullum N.C,Walter W.G. The effects of attention and distraction on the contingent negative variation in normal and neurotic subjects. Electroenceph. clin. Neurophysiol.,25:319-328,1968

Overton and Shagass,C. Distribution of eye movement and eye blink potentials over the scalp. Electroenceph. clin. Neurophysiol., 1969,27:546P.

Quilter,P.M.,Macgillvray,B.B. and Wadbrook,D.G. The removal of eye movement artefact from EEG signals using correlation techniques. Random signals analysis,IEE conference publication, no 159,1977:93-100.

Ryan, T.A.,Joiner,B.F. and Ryan,B.F. Minitab reference manual. Pennsysvannia State Univ.,USA,1982.

Silvey S.D. Multicollinearity and Imprecise Estimation. J. Royal Statist. Soc.,series B,1969,31:539-552

Verleger,R.,Gasser,T. and Mocks,J. Correction of EOG artifacts in event-related potentials of the EEG: Aspects of reliability and validity. Psychophysiology,1982,19:472-480.

Weisberg S. Applied Linear Regression.John Wiley & Sons,Inc,1980.

Appendix

A. The least squares and the correlation techniques methods

The form of the models used in the investigation was given in equation (2) and can be written more compactly in matrix form as:

$$Y = X\Theta + E \qquad \text{(A1)}$$

where Y,X,Θ and E are given by:

$$Y = \begin{bmatrix} EEG(1) \\ EEG(2) \\ . \\ . \\ EEG(m) \end{bmatrix} \qquad X = \begin{bmatrix} V_L(1) & V_R(1) & H_L(1) & H_R(1) \\ V_L(2) & V_R(2) & H_L(2) & H_R(2) \\ . & . & . & . \\ . & . & . & . \\ V_L(m) & V_R(m) & H_L(m) & H_R(m) \end{bmatrix}$$

$$\Theta = \begin{bmatrix} \Theta_1 & \Theta_2 & \Theta_3 & \Theta_4 \end{bmatrix}^T \qquad E = \begin{bmatrix} e(1) & e(2) & \ldots & e(m) \end{bmatrix}^T$$

The suffix T indicates transposition.

Ordinary least squares (OLS) estimates of Θ are obtained by minimizing the sum of squares of the error terms,J:

$$J = \sum_{i=1}^{m} e^2(i) = E^T E$$
$$= (Y-X\Theta)^T (Y-X\Theta)$$

Differentiating J with respect to Θ and simplifying gives :

$$X^T Y = (X^T X)\Theta$$

or if we write these out explicitly:

$$\begin{bmatrix} \sum_i EEE(i)V_L(i) \\ \sum_i EEG(i)V_R(i) \\ \sum_i EEG(i)H_L(i) \\ \sum_i EEG(i)H_R(i) \end{bmatrix} = \begin{bmatrix} \sum_i V_L^2(i) & \sum_i V_L(i)V_R(i) & \sum_i V_L(i)H_L(i) & \sum_i V_L(i)H_R(i) \\ \sum_i V_L(i)V_R(i) & \sum_i V_R^2(i) & \sum_i V_R(i)H_L(i) & \sum_i V_R(i)H_R(i) \\ \sum_i V_L(i)H_L(i) & \sum_i V_R(i)H_L(i) & \sum_i H_L^2(i) & \sum_i H_L(i)H_R(i) \\ \sum_i V_L(i)H_R(i) & \sum_i V_R(i)H_R(i) & \sum_i H_L(i)H_R(i) & \sum_i H_R^2(i) \end{bmatrix} \begin{bmatrix} \Theta_1 \\ \Theta_2 \\ \Theta_3 \\ \Theta_4 \end{bmatrix} \quad \ldots\ldots \text{(A2)}$$

If we define the elements of $(X^T Y)$ and $(X^T X)$ as follows:

$$M_{VL} = \sum_i EEG(i)V_L(i) \quad ; \; P_{VL} = \sum_i V_L^2(i)$$
$$M_{VR} = \sum_i EEG(i)V_R(i) \quad ; \; P_{VR} = \sum_i V_R^2(i)$$
$$M_{HL} = \sum_i EEG(i)H_L(i) \quad ; \; P_{HL} = \sum_i H_L^2(i)$$
$$M_{HR} = \sum_i EEG(i)H_R(i) \quad ; \; P_{HR} = \sum_i H_R^2(i)$$

$$A = \sum_i H_R(i)H_L(i) \; ; \; B = \sum_i V_R(i)V_L(i) \; ; \; C = \sum_i H_R(i)V_L(i)$$

$$D = \sum_i H_L(i)V_R(i) \; ; \; C_{CL} = \sum_i H_L(i)V_L(i) \; ; \; C_{CR} = \sum_i H_R(i)V_R(i)$$

and substituting into (A2) we obtain equation (A3) which is identical to the results of Jervis et al (1980,in preparation).

$$\begin{bmatrix} M_{VL} \\ M_{VR} \\ M_{HL} \\ M_{HR} \end{bmatrix} = \begin{bmatrix} P_{VL} & B & C_{CL} & C \\ B & P_{VR} & D & C_{CR} \\ C_{CL} & D & P_{HL} & A \\ C & C_{CR} & A & P_{HR} \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \end{bmatrix} \qquad (A3)$$

Identical results to those of Quilter et al (1977) can be obtained by considering only a 2-parameter model,e.g:

$$EEG(i) = \theta_1 V_L(i) + \theta_2 H_L(i) + e(i) \qquad (A4)$$

Proceeding as before leads to the following normal equation:

$$\begin{bmatrix} \sum_i EEG(i)V_L(i) \\ \sum_i EEG(i)H_L(i) \end{bmatrix} = \begin{bmatrix} \sum_i V_L^2(i) & \sum_i V_L(i)H_L(i) \\ \sum_i V_L(i)H_L(i) & \sum_i H_L^2(i) \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} \qquad (A5)$$

If we now define the elements of the vectors and matix as :

$$C_V = \sum_i EEG(i)V_L(i) \; ; \; C_H = \sum_i EEG(i)H_L(i) \; ; \; C_C = \sum_i V_L(i)H_L(i)$$

$P_V = \sum_i V_L^2(i)$; $P_H = \sum_i H_L^2(i)$ we obtain:

$$\begin{bmatrix} C_V \\ C_H \end{bmatrix} = \begin{bmatrix} P_V & C_C \\ C_C & P_H \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} \qquad (A6)$$

which if multiplied out is identical to equations (8) and (9) of Quilter et al (1977). Thus we see that the leasts quares approach is simply a more formal way of deriving the normal equations than the so called correlation techniques.

The correlation techniques approach is instructive in the sense that it reveals the need to make certain assumptions about the error terms which is not apparent in the formal least squares method. This can be illustrated as follows. If we multiply equation (A4) by $V_L(i)$ and $H_L(i)$ separately, and sum each result over all M data points (using a slightly more refined method than Quilter et al,1977) we have:

$$\begin{bmatrix}\sum_i \mathrm{EEG}(i)V_L(i)\\ \sum_i \mathrm{EEG}(i)H_L(i)\end{bmatrix} = \begin{bmatrix}\sum_i V_L^2(i) & \sum_i V_L(i)H_L(i)\\ \sum_i V_L(i)H_L(i) & \sum_i H_L^2(i)\end{bmatrix}\begin{bmatrix}\theta_1\\ \theta_2\end{bmatrix} + \begin{bmatrix}\sum_i e(i)V_L(i)\\ \sum_i e(i)H_L(i)\end{bmatrix} \qquad \text{(A7)}$$

Notice the similarity between (A2),(A5) and (A7). Thus (A7) may be written more compactly as:

$$X^T Y = (X^T X)\theta + X^T E \qquad \text{(A8)}$$

It is seen that to obtain θ, it is necessary that $X^T E = 0$. That is, that the EOG and the EEG (or error terms) are uncorrelated. This is the explicit assumption that is made in correlation techniques and is met if the error terms,e(i) are a random sequence. We have seen this assumption is untenable in the OA problem.

B Autocorrelation in the Error terms

Equation (2) in the main text may be written as in (B1)

$$y(i) = x^T(i)\theta + e(i) \qquad i=1,2 \ldots M \qquad \text{(B1)}$$

where $y(i) \equiv \mathrm{EEG}(i)$, the ith sample of the EEG

$x^T(i) \equiv \begin{bmatrix}V_L & V_R & H_L & H_R\end{bmatrix}$ a matrix of the ith samples of the EOGs

θ and e(i) are,respectively, the ocular artefact parameter and the the error terms. Also, $y(i-1) = x^T(i-1)\theta + e(i-1)$

so that $e(i-1) = y(i-1) - X^T(i-1)\hat{\theta}$ (B2)

Now if the error term is represented as a first order autoregressive series then :

$$e(i) = \phi e(i-1) + a(i) \qquad \text{(B3)}$$

Thus (B1) becomes:

$$y(i) = x^T(i) + \phi e(i-1) + a(i) \tag{B4}$$

substituting (B2) into (B4) and rearranging gives

$$y(i) - \phi y(i-1) = \left[x^T(i) - \phi x^T(i-1)\right]\theta + a(i)$$

or $$\bar{y}(i) = \theta\bar{x}^T(i) + a(i) \tag{B5}$$

where $\bar{y}(i) = y(i) - y(i-1)$; $\bar{x}^T(i) = x^T(i) - x^T(i-1)$

Thus, representing the error terms as in (B3) is equivalent to transforming the data as in (B5).

FIGURE LEGEND

Figure 1. Scatter diagrams of the EEG and EOGs for a horizontal eye movement record (AJS36). The symbol * on the diagrams indicates a single point, the numbers give counts of points that fall on the same spot and the symbol + is used where the count exceeds 9. (i) EEG vs vertical right EOG (ii) EEG vs horizontal right EOG.

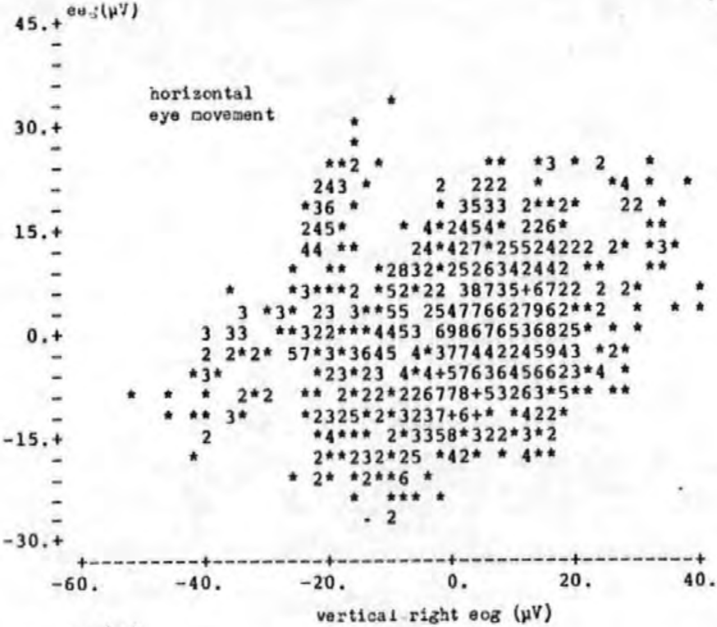
Figure 2. Comparison of the residuals and their autocorrelation functions (ACF) for a horizontal eye movement record (JSS48) for the three estimation methods. (i), (iii) and (v) residuals for the OLS, differencing and iterative methods, respectively. (ii),(iv) and (vi) the corresponding ACFs.

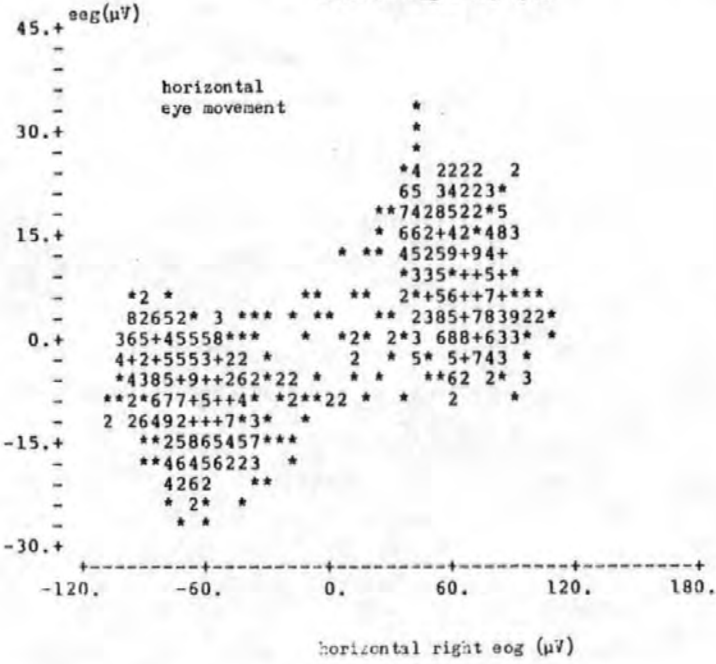
Figure 3. Estimates of the ocular artefacts for a HEM record (AJS36) using (i) model 1A (ii) model 2A (iii)model 3A (iv) model 4A (v) model 4C (vi) model 4D.

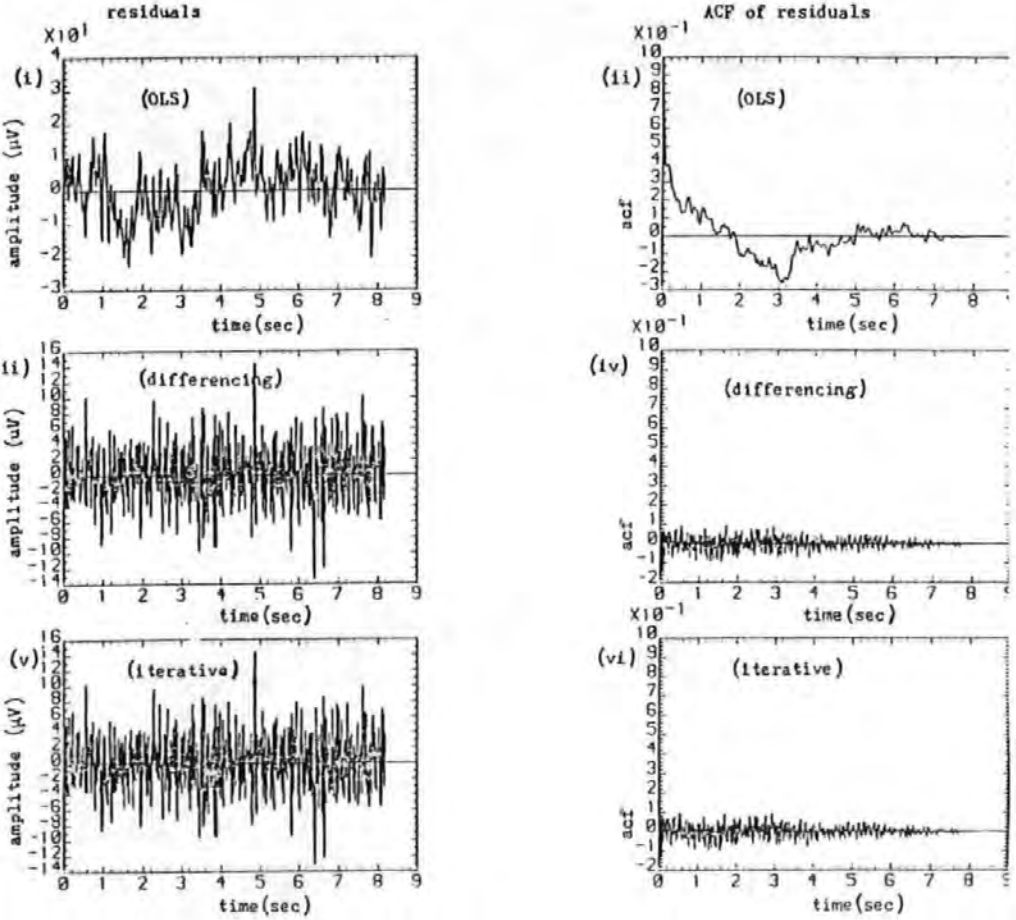
Figure 4. Differences between the (i) raw EEG and the corrected EEG using model 1A (ii) OA estimates of models 2A and 1A (iii) OA estimates of models 3A and 2A (iv) OA estimates of models 4A and 3A (v) OA estimates of models 4C and 3A (vi) OA estimates of models 4D and 3A.

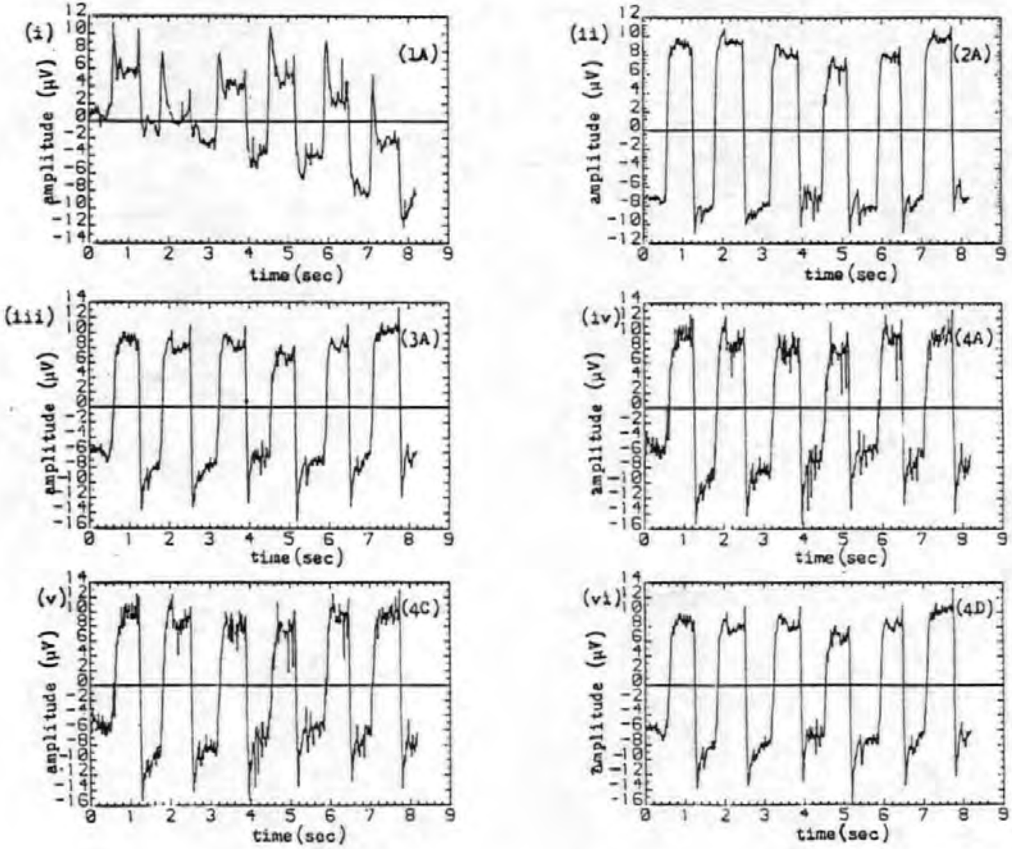
Figure 5. Waveforms of the corrected EEG corresponding to the OA estimates of figure 3.

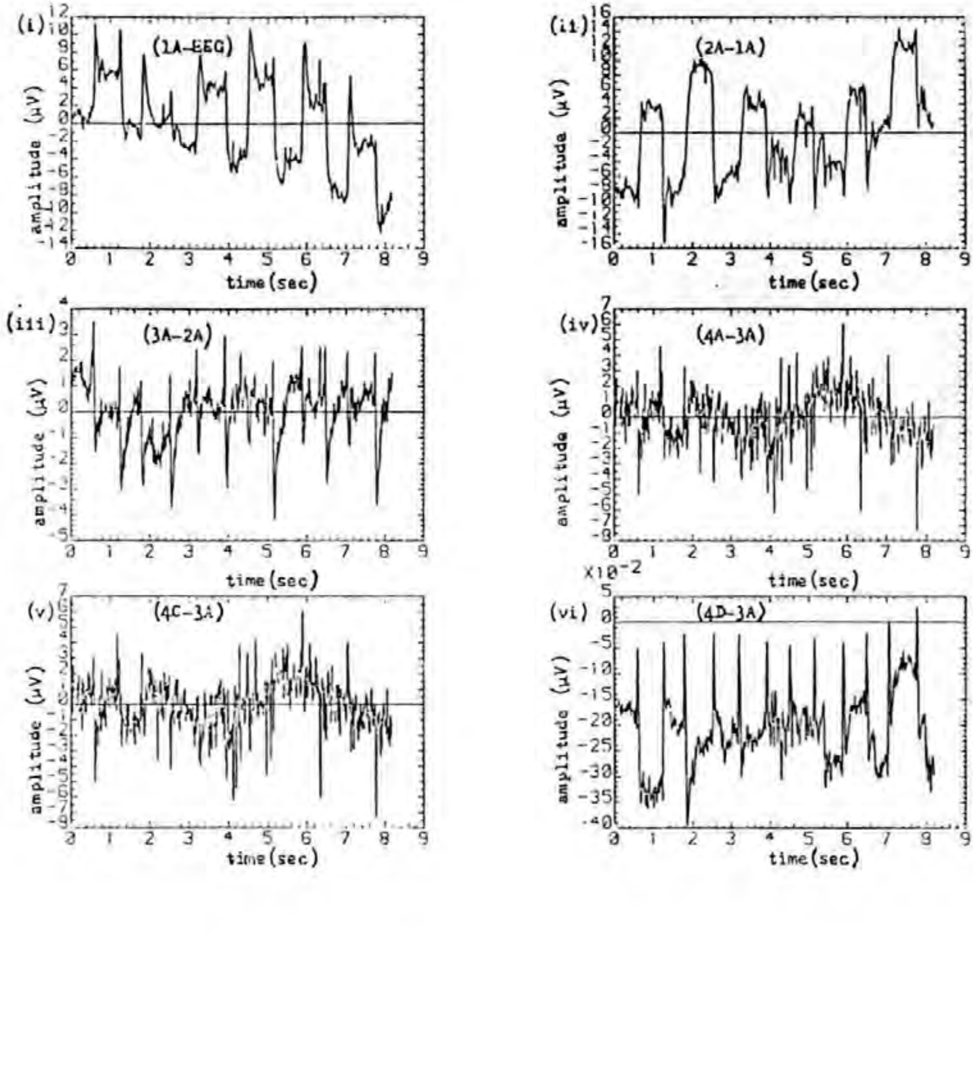
Figure 6. Comparison of the OA estimates for a blink record (AJS114) using the three estimation methods with model 3D. (i),(iii) and (v) OA estimates for the OLS ,differencing and the iterative methods. (ii),(iv) and (vi) Differences between the OA estimates for OLS and differencing methods,OA estimates of OLS and iterative methods, and the OA estimates of differencing and Iterative methods.

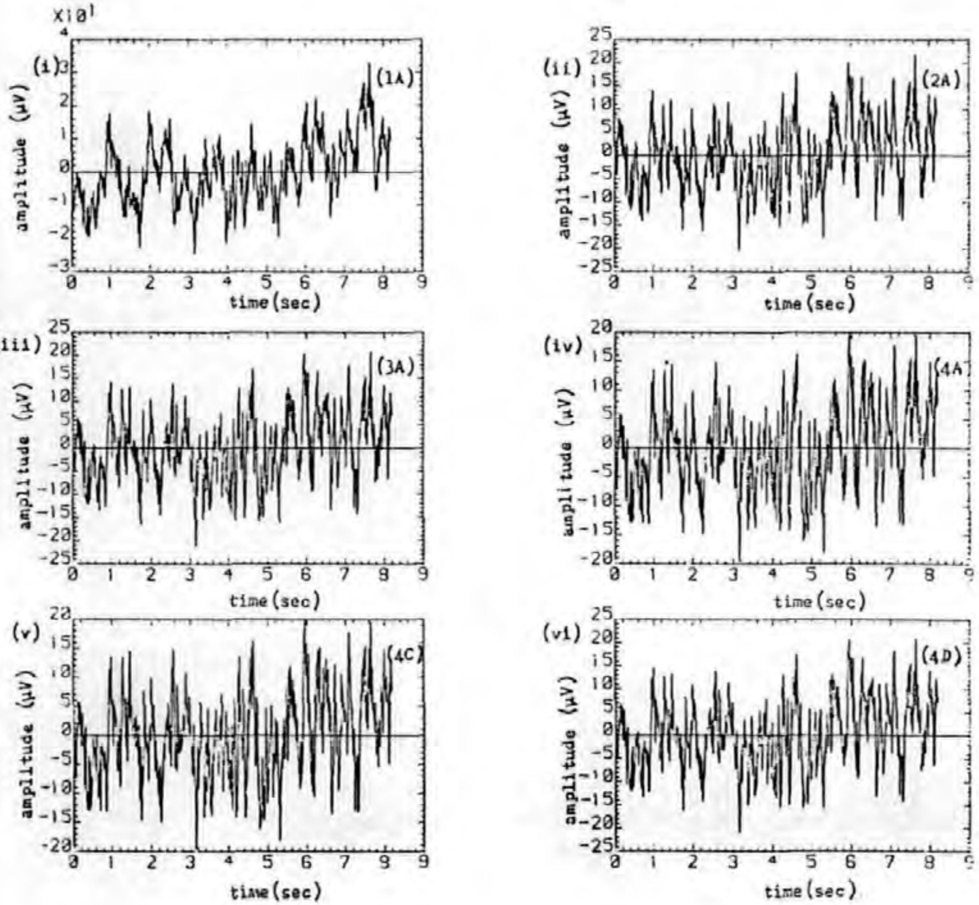

eog(µV)
horizontal
eye movement
45.
30.
15.
0.
-15.
-30.
-60.
-40.
-20.
0.
20.
40.
vertical right eog (µV)

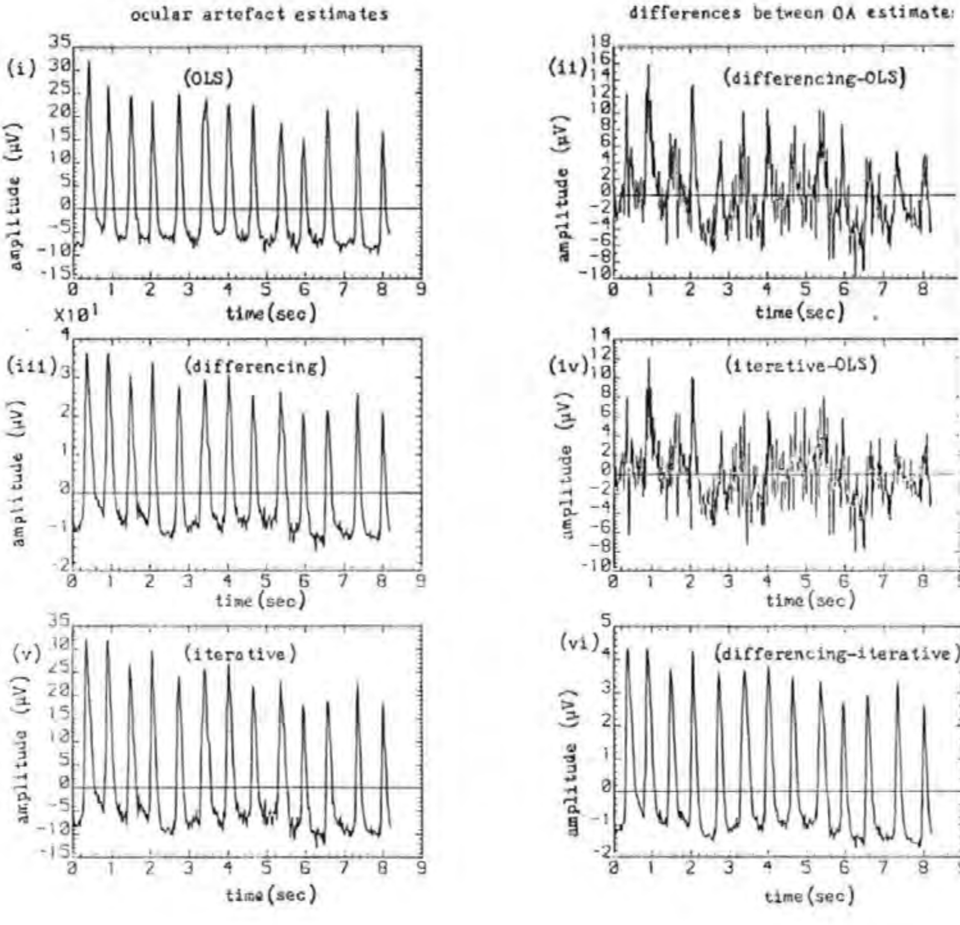

eog(µV)
horizontal
eye movement
45.
30.
15.
0.
-15.
-30.
-120.
-60.
0.
60.
120.
180.
horizontal right eog (µV)


residuals
ACF of residuals
X10^1
(i)
(OLS)
amplitude (μV)
time(sec)
X10^-1
(ii)
(OLS)
acf
time(sec)
(iii)
(differencing)
amplitude (uV)
time(sec)
(iv)
(differencing)
acf
time(sec)
(v)
(iterative)
amplitude (μV)
time(sec)
(vi)
(iterative)
acf
time(sec)


(i)
(1A)
(ii)
(2A)
(iii)
(3A)
(iv)
(4A)
(v)
(4C)
(vi)
(4D)
amplitude (μV)
time (sec)
12 10 8 6 4 2 0 -2 -4 -6 -8 -10 -12 -14
14 12 10 8 6 4 2 0 -2 -4 -6 -8 -10 -12 -14 -16
0 1 2 3 4 5 6 7 8 9


(i)
(1A-EEG)
amplitude (μV)
12
10
8
6
4
2
0
-2
-4
-6
-8
-10
-12
-14
time(sec)
0 1 2 3 4 5 6 7 8 9
(ii)
(2A-1A)
amplitude (μV)
16
14
12
10
8
6
4
2
0
-2
-4
-6
-8
-10
-12
-14
-16
time(sec)
0 1 2 3 4 5 6 7 8 9
(iii)
(3A-2A)
amplitude (μV)
4
3
2
1
0
-1
-2
-3
-4
-5
time(sec)
0 1 2 3 4 5 6 7 8 9
(iv)
(4A-3A)
amplitude (μV)
7
6
5
4
3
2
1
0
-1
-2
-3
-4
-5
-6
-7
-8
time(sec)
0 1 2 3 4 5 6 7 8 9
(v)
(4C-3A)
amplitude (μV)
7
6
5
4
3
2
1
0
-1
-2
-3
-4
-5
-6
-7
-8
time(sec)
0 1 2 3 4 5 6 7 8 9
(vi)
(4D-3A)
X10^-2
amplitude (μV)
5
0
-5
-10
-15
-20
-25
-30
-35
-40
time(sec)
0 1 2 3 4 5 6 7 8 9


X10¹
(i)
(1A)
amplitude (μV)
4
3
2
1
0
-1
-2
-3
time (sec)
0 1 2 3 4 5 6 7 8 9
(ii)
(2A)
amplitude (μV)
25 20 15 10 5 0 -5 -10 -15 -20 -25
time (sec)
0 1 2 3 4 5 6 7 8 9
(iii)
(3A)
amplitude (μV)
25 20 15 10 5 0 -5 -10 -15 -20 -25
time (sec)
0 1 2 3 4 5 6 7 8 9
(iv)
(4A)
amplitude (μV)
20 15 10 5 0 -5 -10 -15 -20
time (sec)
0 1 2 3 4 5 6 7 8 9
(v)
(4C)
amplitude (μV)
20 15 10 5 0 -5 -10 -15 -20
time (sec)
0 1 2 3 4 5 6 7 8 9
(vi)
(4D)
amplitude (μV)
25 20 15 10 5 0 -5 -10 -15 -20 -25
time (sec)
0 1 2 3 4 5 6 7 8 9



ocular artefact estimates
differences between OA estimate
(i)
(OLS)
(ii)
(differencing-OLS)
(iii)
(differencing)
X10¹
(iv)
(iterative-OLS)
(v)
(iterative)
(vi)
(differencing-iterative)
amplitude (µV)
time(sec)
0 1 2 3 4 5 6 7 8 9
35 30 25 20 15 10 5 0 -5 -10 -15
18 16 14 12 10 8 6 4 2 0 -2 -4 -6 -8 -10
4 3 2 1 0 -1 -2
14 12 10 8 6 4 2 0 -2 -4 -6 -8 -10
5 4 3 2 1 0 -1 -2

Table I

Correlation Matrices and their eigenvalues for the EOGs for (a) a vertical eye movement Record (JSS36) (b) a blink record (AJS108 (c) a horizontal eye movement Record (AJS36) (d) a diagonal eye movement record (AJS54)

	V_L	V_R	H_L	H_R
V_L	1.000			
V_R	0.990	1.000		
H_L	0.917	0.926	1.000	
H_R	0.894	0.907	0.978	1.000

Eigenvalues: $\lambda 1=3.806, \lambda 2=0.164$

$\lambda 3=0.020, \lambda 4=0.010$

(a) $\sum_{i=1}^{4}\frac{1}{\lambda_i}=156.36$

	V_L	V_R	H_L	H_R
	1.000			
	0.987	1.000		
	0.956	0.960	1.000	
	0.942	0.943	0.943	1.000

Eigenvalues: $\lambda 1=3.876, \lambda 2=0.078$

$\lambda 3=0.033, \lambda 4=0.013$

(b) $\sum_{i=1}^{4}\frac{1}{\lambda_i}=120.30$

	V_L	V_R	H_L	H_R
V_L	1.000			
V_R	0.853	1.000		
H_L	-0.662	-0.258	1.000	
H_R	0.799	0.430	-0.957	1.000

Eigenvalues: $\lambda 1=3.012, \lambda 2=0.933$

$\lambda 3=0.037, \lambda 4=0.018$

(c) $\sum_{i=1}^{4}\frac{1}{\lambda_i}=83.99$

	V_L	V_R	H_L	H_R
	1.000			
	0.994	1.000		
	0.955	0.949	1.000	
	-0.761	-0.733	-0.849	1.000

Eigenvalues: $\lambda 1=3.628, \lambda 2=0.330$

$\lambda 3=0.036, \lambda 4=0.006$

(d) $\sum_{i=1}^{4}\frac{1}{\lambda_i}=197.75$

Table II

Mean parameter estimates ($\hat{\theta}$) and their standard deviations (SD) for records corrected for OA with single parameter models that used V_R and H_R for the OLS and differencing methods.

Subject		Vertical EM				Blink			
		OLS		Differencing		OLS		Differencing	
		(V_R)	(H_R)	(V_R)	(H_R)	(V_R)	(H_R)	(V_R)	(H_R)
	$\hat{\theta}$	0.16	0.52	0.14	0.33	0.07	0.23	0.09	0.31
AJS	SD	0.03	0.12	0.01	0.06	0.01	0.06	0.01	0.06
	$\hat{\theta}$	0.18	0.76	0.15	0.34	0.10	0.35	0.11	0.40
JSS	SD	0.04	0.38	0.02	0.08	0.01	0.21	0.01	0.12

Subject		Horizontal EM				Diagonal EM			
		OLS		Differencing		OLS		Differencing	
		(V_R)	(H_R)	(V_R)	(H_R)	(V_R)	(H_R)	(V_R)	(H_R)
	$\hat{\theta}$	0.20	0.13	0.28	0.16	0.16	0.24	0.15	0.18
AJS	SD	0.14	0.022	0.1	0.037	0.07	0.06	0.05	0.11
	$\hat{\theta}$	0.38	0.13	0.22	0.22	0.18	0.29	0.18	0.36
JSS	SD	0.07	0.04	0.08	0.12	0.08	0.10	0.11	0.24

Table III

Test statistics for the multiparameter models for a blink record (AJS114) (a) OLS method (b) Differencing method.

table III(a)

Model	S_p	Cp	d
1A	47.49	67.15	0.35
2A	45.58	23.33	0.35
2B	47.50	68.32	0.34
2C	45.56	23.81	0.34
2D	44.97	10.30	0.37
2E	45.55	23.62	0.34
2F	48.95	101.62	0.33
2G	47.03	57.49	0.37
3A	46.78	52.82	0.36
3B	45.55	24.77	0.34
3C	44.86	8.90	0.38
3D	44.57	2.34	0.39
4A	44.56	*	0.39
4B	46.78	53.80	0.36
4C	44.56	3.00	0.39
4D	46.74	52.85	0.36

table III(b)

Model	S_p	Cp	d
1A	16.45	166.33	1.59
2A	15.47	95.53	1.60
2B	14.45	22.96	1.52
2C	15.48	97.21	1.58
2D	14.30	11.94	1.51
2E	15.48	97.47	1.59
2F	14.43	21.04	1.51
2G	15.46	95.90	1.51
3A	14.41	20.97	1.51
3B	15.50	99.20	1.59
3C	14.27	10.61	1.51
3D	14.16	2.56	1.49
4A	14.15	*	1.49
4B	14.42	22.87	1.51
4C	14.15	3.00	1.49
4D	14.02	-6.18	1.49

* not applicable

Table IV

Summary of results based on the Cp criterion: the table shows the number of times each model had Cp values of about ten or less for each ocular movement and for two subjects (AJS and JSS).

table IV(a)

MODEL	VEM		Blink		HEM		DEM		
	AJS	JSS	AJS	JSS	AJS	JSS	AJS	JSS	TOTAL
2A	0	0	0	0	0	1	1	2	4
2B	0	0	0	0	0	0	2	2	4
2C	0.	0	0	1	0	0	4	1	6
2D	0	0	2	0	2	0	2	3	9
2E	0	0	0	0	0	0	3	0	3
2F	0	0	0	1	0	1	1	0	3
2G	1	0	0	0	0	0	1	1	3
3A	1	2	0	3	0	3	4	5	16
3B	2	2	1	2	1	0	4	1	16
3C	1	0	2	1	1	0	4	5	14
3D	1	2	3	3	3	0	5	4	21
4B	1	1	0	0	0	4	4	5	15
4C	5	5	3	5	3	4	8	7	40
4D	2	3	3	3	3	3	4	5	26

table IV(b)

MODEL	VEM		Blink		HEM		DEM		
	AJS	JSS	AJS	JSS	AJS	JSS	AJS	JSS	TOTAL
2A	0	0	0	0	0	0	0	0	0
2B	0	0	0	0	0	1	0	0	1
2C	0	0	0	0	0	0	2	1	3
2D	5	0	3	0	2	1	3	1	15
2E	0	0	0	0	0	0	1	0	1
2F	0	0	1	1	0	0	0	2	4
2G	0	1	0	0	0	0	1	1	3
3A	0	0	1	1	0	1	1	3	7
3B	0	0	0	0	1	1	2	2	6
3C	3	0	2	0	3	1	3	1	13
3D	5	5	3	4	3	4	8	4	36
4B	0	1	0	0	0	1	1	3	6
4C	5	5	3	5	3	4	8	7	40
4D	0	0	2	1	0	2	2	3	10

Table V

Comparison of parameter estimates,Durbin-Watson statistics and the autoregressive parameter ϕ for the OLS,differencing and iterative methods using model 3D.

(a) VEM record (JSS36), (b) HEM record (AJS36)

(c) Blink record (AJS114),(d) VEM record (AJS0).

	Parameter Estimates			T-ratio				
	$\hat{\theta}_1$	$\hat{\theta}_2$	$\hat{\theta}_3$	$T\theta_1$	$T\theta_2$	$T\theta_3$	d	ϕ
OLS	0.121	-0.231	0.609	19.36	-4.47	14.61	0.17	0
Diff.	0.129	-0.181	0.331	9.69	-5.19	9.17	1.81	1
Iter.	0.140	-0.165	0.355	11.47	-4.74	9.94	1.75	0.935
(a)								
OLS	0.091	-0.177	-0.108	4.69	-15.41	-7.18	0.27	0
Diff	0.330	-0.044	0.102	9.99	-2.46	4.47	1.65	1
Iter.	0.306	-0.046	0.077	9.39	-2.64	3.50	1.60	0.942
(b)								
OLS	0.107	0.094	-0.174	10.07	3.16	-4.85	0.39	0
Diff.	0.0515	-0.104	0.304	4.52	-3.37	9.83	1.49	1
Iter.	0.047	-0.105	0.274	4.24	3.46	8.79	1.41	0.866
(c)								
OLS	0.168	-0.115	0.0664	27.68	-3.56	2.43	0.22	0
Diff.	0.129	-0.011	0.105	15.20	-0.44	3.94	1.59	1
Iter.	0.132	-0.017	0.097	17.02	-0.66	3.70	1.52	0.899
(d)								

$T\theta_j = \dfrac{\hat{\theta}_j}{\delta\theta}$ where δ_θ is the standard error for $\hat{\theta}_j$, j=1,2,3.

A NEW ON-LINE METHOD FOR REMOVING OCULAR ARTEFACTS FROM EEG SIGNALS

Emmanuel C. Ifeachor, Barrie W. Jervis[1], Earl L. Morris[2],
Elaine Allen[3] and Nigel R. Hudson[3].

Department of Communication Engineering,
Plymouth Polytechnic,
Drake Circus,
Plymouth, PL4 8AA.

1 Department of Electrical and Electronic Engineering,
Sheffield City Polytechnic,
Pond Street,
Sheffield S1 1WB

2 Department of Engineering Science,
University of Exeter,
Exeter.

3 Department of Neurosciences,
Freedom Fields Hospital,
Plymouth, PL4 7JJ

A NEW ON-LINE METHOD FOR REMOVING OCULAR ARTEFACTS FROM EEG SIGNALS

Abstract.

A method is described for on-line removal of ocular artefacts from the human electroencephalograms (EEG). It uses numerically stable algorithms based on the efficient recursive least squares . The method is shown to give similar results to its off-line equivalents. Compared to the present on-line methods our approach is superior,requiring no subjective manual adjustment and it processes all signals digitally. At present, the ocular artefact removal algorithms are carried out off-line, but can be carried out in real-time on a microprocessor.

Keywords: EEG,EOG,ocular artefacts,recursive least squares,
on-line.

A NEW ON-LINE METHOD FOR REMOVING OCULAR ARTEFACTS FROM EEG SIGNALS

1 Introduction

Movement of the eye or eyelid produces a contaminating potential in the Electroencephalogram (EEG) signals known as ocular artefact (OA). The extent of the contamination decreases with scalp distance from the eyes. In both research and routine clinical neurophysiological recordings, the presence of OA reduces the clinical usefulness of valuable signals in diagnosis.
For example, an event-related potential known as the Contingent Negative Variation (CNV) which may be of value in the pre-symptomatic diagnosis and monitoring of Huntingtons Chorea, is often contaminated by ocular artefacts which may be similar in shape to the CNV. (e.g Hillyard and Galombos 1970, Wasman et al 1970,Jervis et al, 1984). In other applications OA hampers the automatic analysis of the EEG signals (e.g. Barlow 1979,Quilter et al 1977, Gotman et al 1975).

Several methods have been proposed for removing the OA from the EEG, but the electrooculogram (EOG) subtraction method is the most promising. In this approach, skin electrodes are placed at sites near the eyes to measure the change in potential due to movement of the eye and/or lid. This potential is called the EOG. To remove the OA, appropriate fractions of one or more EOG signals are subtracted from the measured EEG (e.g. Quilter et al,1977;Jervis et al; in preparation).

The EOG subtraction method can be carried out either on-line (that is as the data is being obtained),(e.g Barlow and Rémond 1981) or off-line at some later time (e.g. Quilter et al 1977,Ifeachor et al, in preparation). The main advantage of off-line methods is that more sophisticated removal techniques can be employed. Previous on-line OA removal methods have been simple and somewhat inefficient. (compare for example, Girton and Kamiya 1973 & Portgens and De Bruin 1983). Nevertheless, the trend in EEG signal processing is clearly towards real-time or near real-time processing (e.g. Harris 1983, Barlow 1979,Barlow et al 1981), so that the delay involved when off-line methods are employed is unacceptable. We agree with Barlow and Rémond (1981) who concluded that the best solution was to remove the artefact at the outset. Also a dedicated on-line OA removal system will eliminate the computing costs associated with off-line methods.

Previously published on-line methods required manual calibration; setting them up was thus time consuming, especially for multichannel recording. Also their calibration required subjects to move their eyes or blink repetitively. This ruled out their use with infants and uncooperative adults. Additionally,the method of assessing whether the OA was removed was subjective and the whole procedure required the operator to be familiar with the method. Furthermore, these methods were unable to deal with multiple artefacts,so that if a different ocular artefact occured after calibration, it was not removed properly (e.g. Weerts et al,1970).

In this paper a method is presented which is suitable for on-line removal of OA with a performance that is at least as good as its off-line line equivalents. It is based on the well known recursive least squares method.

2 Present On-line Ocular artefact Removal methods.

Only three on-line removal methods have been reported in detail in the literature viz McCallum and Walter (1968),Girton and Kamiya (1973) and Barlow and Rémond (1981). All these methods employed analogue techniques to remove OA from the EEG signals.

McCallum and Walter (1968) described one of the first on-line methods for reducing OA in the CNV recordings which used simple potentiometer arrangement. The vertex EEG signal was referred to the centre terminal of the potentiometer. The other terminals of the potentiometer were connected to a frontal EEG electrode and linked earlobes. The corrector was calibrated,prior to recording, by adjusting the centre terminal whilst the subject moved his eyes repetitively until there was no trace of OA in the EEG. The device was then left at this setting during recording. As Girton and Kamiya (1973) pointed out, this method could not deal satifactorily with lateral eye movement artefacts as these were not taken into account, and there was the additional difficulty of interpreting the corrected EEG. Jervis et al (in preparation) have used this method and found it unsatisfactory. Wasman et al (1970) reported that their attempt to use this method was unsuccessful as it resulted in "positive CNV".

The on-line method described by Girton and Kamiya (1973),like the McCallum and Walter method, used a potentiometer arrangement and required manual calibration. Modifications were made to cater for lateral eye movement and to improve the signal quality. However, a number of workers have used the method and found it unwieldy and inefficient in removing ocular artefacts (e.g Gotman et al 1975,Whitton et al 1978).

Barlow and Rémond (1981) proposed a method that was similar to that of Girton and Kamiya (1973). The essential differences between the two were that Barlow and Rémond's method required fewer amplifiers when used in multichannel recording and had facilities such as those for changing the sign of the EOGs used to remove the OA.

3 The Least Squares estimation of ocular artefact parameters.

All the EOG subtraction methods,both off-line and on-line, are essentially forms of the least squares method (Ifeachor et al, in preparation). They are based on the assumption that the OA is linearly related to the EOGs. Thus:

$$y(i) = \theta_1 x_1(i) + \theta_2 x_2(i) + \theta_3 x_3(i) \ldots \theta_n x_n(i) + e(i) \qquad i=1,2,..m \quad (1)$$

where y(i) and $x_j(i)$ are the samples of the measured EEG and the EOGs respectively. e(i),the error term, represents the background EEG.The θ_j are the ocular artefact parameters and n is the number of parameters in the model. The problem in ocular artefact removal is,given the $x_j(i)$ and y(i) above to obtain estimates of θ_1 to θ_n which are then used to obtain an estimate of e(i) in (1). θ_j can be estimated either off-line or on-line.

3.1 Off-line Estimation of the ocular artefact parameters,θ.

Assuming that the error terms,e(i), are uncorrelated random variables with zero mean and constant variance, optimum estimates of the parameters can be obtained by minimizing the sum of squares of the error terms,$J=\sum_{i=1}^{m} e^2(i)$. This leads to the ordinary least squares (OLS) estimates of equation (2) which is the basis of the offline OA removal methods.

$$\hat{\theta}_m = (X_m^T X_m)^{-1} X_m^T Y_m \qquad (2)$$

where $Y_m, \hat{\theta}_m, X_m$ are respectively,vectors of measured EEG samples, parameter estimates and a matrix of EOG samples given by:

$$Y_m = \begin{bmatrix} y(1) \\ y(2) \\ . \\ . \\ y(m) \end{bmatrix} \qquad \hat{\theta}_m = \begin{bmatrix} \hat{\theta}_1 \\ \hat{\theta}_2 \\ . \\ . \\ \hat{\theta}_n \end{bmatrix} \qquad X_m = \begin{bmatrix} x_1(1) & x_2(1) & \ldots & x_n(1) \\ x_1(2) & x_2(2) & \ldots & x_n(2) \\ . & . & \ldots & . \\ . & . & \ldots & . \\ x_1(m) & x_2(m) & \ldots & x_n(m) \end{bmatrix}$$

The suffix m indicates that each matrix was obtained using all m data points (Ifeachor et al, in preparation).

3.2 On-line Algorithm for removing Ocular Artefacts

As may be apparent, the computation of $\hat{\theta}(m)$ in (2) requires the use of all samples available. Thus the amount of data storage and computation increase with m. Furthermore, the time consuming inverse matrix has to be computed each time θ is to be estimated. Clearly this approach is not suitable for real-time or on-line estimation . In practice when continuous data are being acquired, and we wish to improve our parameter estimates using the new data, recursive methods are preferred. With a recursive algorithm the estimates can be updated for each new set of data acquired without repeatedly solving the time consuming matrix equation of (2) directly.

A suitable recursive least squares (RLS) algorithm, which also tracks slowly varying parameters is obtained by exponentially weighting the data to gradually remove the effects of old data on the estimates. Thus we minimize :

$$J = \sum_{i=1}^{m} \Upsilon^{m-i} e^2(i) \qquad 0 < \Upsilon < 1$$

Υ is referred to as the forgetting factor. The weighting scheme reduces tp that of OLS when Υ=1. Typically Υ is between 0.98 and 1. Smaller values assign too much weight to the more recent data which leads to wildly fluctuating estimates. Minimization of J with respect to the θ's leads to the following sets of equations (see for example, Peterka 1975, Morris and Abaza (1976)):

$$\hat{\theta}(m+1) = \hat{\theta}(m) + P(m+1)\ x(m+1)z(m+1) \qquad (3a)$$

$$P(m+1) = \frac{1}{\gamma}\left[P(m) - \frac{P(m)x(m+1)x^T(m+1)P(m+1)}{a(m+1)}\right] \qquad (3b)$$

where,

$$a(m+1) = \gamma + x^T(m+1)P(m)x(m+1)$$

$$z(m+1) = y(m+1) - x^T(m+1)\hat{\theta}(m)$$

$$x^T(m+1) = \left[x_1(m+1) \quad x_2(m+1) \quad \dots \quad x_n(m+1)\right]$$

All that is required to start the algorithm even when only the first set of samples have been acquired is to assign initial values to P(0) and $\hat{\theta}(0)$. A common practice is to select arbitrary values for $\hat{\theta}(0)$ and then set $P(0)=\alpha I$,where I is a unit matrix and α is a positiv constant (Clarke 1981b;Eykoff 1974). This approach can be shown to converge to the true values for P and $\hat{\theta}$. (e.g. Eykoff 1974).

Factorization Algorithms

The RLS method is very efficient and involves exactly the same number of arithmetic operations between samples, as $\hat{\theta}(m+1)$ and P(m+1) in (3) have fixed dimensions. This is an important requirement for efficient on-line realization. There are however, two main problems that may be encountered when the RLS algorithm is implemented directly. Firstly, if the signal is not 'persistently exciting' as will be the case when there is no ocular movement, the matrix P would blow up due to scaling by . However, because of miniature eye movements and other activities that are normally picked up in the EOG channels, this problem may not be so serious in OA removal. Secondly, for succesful estimation, the matrix P is required to be positive semidefinite. If this condition is not met, then P does not exist and the algorithm becomes unstable. Because of differencing of terms in equation (3b) positive definiteness of P cannot be guaranteed. (see for example, Bierman,1976; Peterka,1975). This problem is worse in multiparameter models,especially if the variables (BOGs in our case) are linearly dependent (Peterka,1975) and when the algorithm is implemented on a small system with finite word length. When the algorithm has iterated for a long time the two terms in the bracket in (3b) are very nearly equal and subtraction of such terms in a finite word length system may lead to errors and a negative definite P matrix (Clarke 1981b,Bierman 1976).

An elegant way of solving the problems of 'blow-up' and numerical instability is to replace the RLS algorithm with one that is numerically better conditioned. Such algorithms factorize the matrix P(m+1) so that the differencing of terms in (3b) is avoided. We shall briefly describe two of these algorithms which we have adopted in our OA removal algorithm.

(a) The Square root Algorithm

In the square root method, the matrix P is factored as:

$$P(m+1) = S(m+1)S^{T}(m+1) \qquad (4)$$

where S(m+1), an upper triangular matrix and $S^{T}(m+1)$ its transpose, is a square root of P(m+1). Thus if S(m+1) instead of P(m+1) is updated, the positive definiteness of P(m+1) is guarranteed since the product of two square roots is always positive. By rearranging equation (3b), S(m+1) can be shown to be (Peterka,1975):

$$S(m+1) = \frac{1}{\sqrt{\Upsilon}} S(m)H(m) \qquad (5)$$

where H(m) is an upper triangular matrix. Peterka (1975) has given a recursive algorithm for updating S(m+1).

(b) U-D factorization Algorithm.

The U-D factorization is an alternative method of updating the covariance matrix P(m+1) in a numerically stable form (e.g Bierman,1976). It is in fact a square root- free arrangement of the conventional square root algorithm and thus shares the same properties as the latter. In this method P(m+1) is factored as:

$$P(m+1) = U(m+1)D(m+1)U^{T}(m+1) \qquad (6)$$

where U(m+1) is a unit upper triangular matrix, $U_T(m+1)$ its transpose and D(m+1) a diagonal matrix. Thus instead of updating P (dropping the suffices for convenience) its factors U and D are updated. Bierman (1976) has given an algorithm for updating U and D recursively for the Kalman filter, which uses the variance of the error term,e(i), but not Υ. This algorithm has been trivially modified for our OA problem to encorporate Υ instead.

Compared to the square root filter, the U-D method is more efficient in terms of storage and computation. Additionally, for an n-parameter model, the square root filter requires the evaluation of n square roots. Also in the U-D method the elements of D give directly the diagonal elements of P which can be used for statistical analysis. For these reasons, we prefer the U-D method.

4 Experimental data

The data used in the investigation were obtained in a previous study into the problem of ocular artefacts in the EEG (Jervis et al 1980;in preparation). A pair of skin electrodes were placed above and below each eye and others at the nasion and the outer canthi. One channel of EEG was obtained from the vertex referred to linked ear lobes. Subjects with no known eye defects were asked to make periodic eye movements in the vertical and horizontal planes between pairs of light emitting diodes corresponding to 20° of angular displacement. The EOG and EEG signals were fed to an electroencephalograph to produce paper record and to a data acquisition system where they were filtered, digitized and stored on a disc. The data was later transferred to a main frame computer for analysis. Figure 1 gives typical waveforms of the EOGs and the contaminated EEG for a horizontal eye movement.

5 Models

The models given in (7) below were used in the investigation. Two of these models (3D and 4D) had already been found to give the best ocular artefact removal in a previous investigation (Ifeachor et al, in preparation), where statistical and pictorial methods were successfully used to compare ocular artefact models, and the third is included to show the effects of EOG 'noise' and correlation among the EOGs on the parameter estimates. These models, in that paper, were referred to as 3D,4A and 4D.

Model name	Model description	
3D	$y(i) = \theta_2 V_R(i) + \theta_3 H_L(i) + \theta_4 H_R(i) + e(i)$	(7a)
4A	$y(i) = \theta_1 V_L(i) + \theta_2 V_R(i) + \theta_3 H_L(i) + \theta_4 H_R(i) + e(i)$	(7b)
4D	$y(i) = \theta_1 V_L(i) + \theta_2 H_L(i) x\, H_R(i) + \theta_3 H_L(i) + \theta_4 H_R(i) + e(i)$	(7c)

where y(i),e(i) are,respectively the ith samples of the measured scalp signal and background EEG. $V_L(i)$ and $H_L(i)$ are,respectively, the ith samples of the vertical and horizontal left eye EOGs. $V_R(i)$ and $H_R(i)$ are the corresponding right eye EOGs. The θ_j have the usual meaning.

6 Results and Discussion

To verify that the recursive algorithms described in section 3.2 when used on-line will give the same results as the off-line (or batch) method, several EEG records contaminated by ocular artefacts were corrected on a main frame computer using both methods.

(a) Comparison of the recursive On-line and the Off-line methods.

Figures 2 and 3 give the ocular artefact estimates and the corrected EEGs for these models for the batch and the U-D methods, respectively. Ignoring the initial transients, both the OA estimates and the corrected EEG for the two methods are similar. Table 1 compares the parameter estimates for the batch or conventional offline method with those for the UD algorithm after 1024 iterations. An examination of the table shows that the parameter estimates for both methods are essentially the same,an indication that they are equivalent. Similar results to figures 2 and 3 were obtained for other types of ocular movement (blinks,vertical eye movement, and diagonal eye movement).

(b) Convergence of the Parameter estimates.

To study the behaviour of the recursive algorithm between samples and its convergence properties, we obtained and studied plots of the variation with time of the parameter estimates,the sample variance,S^2 and the multiple correlation coefficient, R^2 (e.g Johnston 1972). The sample variance and multiple correlation coefficient were obtained with the following approximate recursive equations (see the appendix):

$$S^2(m+1) = SS(m+1) / asL(m+1) - n \qquad ; asL(m+1) > n \qquad (8a)$$

$$R^2(m+1) = SS(m+1) / SS_T(m+1) \qquad (8b)$$

where SS(m+1) and $SS_T(m+1)$ are,respectively, the sum of squares of the residuals and the total sum of squares and given by:

$$SS(m+1) = \Upsilon\, SS(m) + e(m+1)e(m+1) \tag{9a}$$

$$SS_T(m+1) = \Upsilon\, SS_T(m) + y^T(m+1)y(m+1) \tag{9b}$$

n is the number of parameters in the model and asL(m+1) is given by

$$asL(m+1) = 1 + \Upsilon asL(m+1) = \sum_{j=0}^{m} \Upsilon^j$$

Figures 4(i) to 4(iii) illustrate the variation of the parameter estimates with time for the U-D algorithm. It is seen that there were initial transients as a result of guessed initial values for U,D and $\hat{\theta}$, after which they appeared to converge to some value about which they varied slowly. We note from the figures that the two parameter estimates $\hat{\theta}_1$ and $\hat{\theta}_2$ in figure 4(ii) were opposite in sign, approximately equal in magnitude and varied in unison. This was because the corresponding EOGs were positively correlated (correlation coefficient, 0.853), so that the effects of including one EOG was largely offset by also including the other. The same was true of $\hat{\theta}_3$ and $\hat{\theta}_4$, but in this case the EOGs were negatively correlated (correlation coefficient, -0.957). The large variation in $\hat{\theta}_1$ and $\hat{\theta}_2$ was probably because the corresponding EOGs contained a relatively large amount of noise unrelated to ocular movement (see figure 1). These effects suggested that it is undesirable to include all the EOGs in a model and this agrees with our previous findings (Ifeachor et al, in preparation).

The variation with time of estimates of the residual variance S^2 and the multiple correlation coefficient, R^2, are depicted in figures 5(i) and (ii). It is seen that both S^2 and R^2 increased with time, rapidly at first and then slowly. This is an indication that these quantities were converging.

To simulate longer lengths of data and to verify whether the parameter estimates have converged, the operation was recycled for the UD algorithm. That is, at the end of the data the algorithm was resumed at the begining of the data without reassigning starting values for U,D and $\hat{\theta}$. Table 2 compares the final values of the parameter estimates ,the sample variance and the multiple correlation coefficient for the first,second and third cycles with those of the batch method. It is evident that the quantities have converged by the end of the first cycle. It is note-worthy that the final values of both the variance and the multiple correlation coefficients are reasonably close to those of the batch method. Demonstrating that equations (8a) and (8b) are good estimates of these quantities. The traces on the right hand side of figures 4 and 5 give the parameter estimates,the sample variance and the multiple correlation coefficient for the second cycle. They confirm that the algorithm had converged. The estimates of the ocular artefact and the background EEG in this case were very similar to the results for the batch method.

Results very similar to those of the UD algorithm were obtained for the ordinary recursive least squares and the square root algorithms The notable difference between them was that they always produced different initial transients due to the starting values for P(0) (or S(0) for the square root algorithm) and $\hat{\theta}(0)$.

7 Conclusions

The investigation showed that the powerful recursive algorithms give similar results to the off-line methods and can therefore be used to remove ocular artefacts from the EEGs. It is thus possible to correct for OA on-line with algorithms that have performance at least as good as the offline methods. By using a forgetting factor, Υ, of less than 1 the recursive methods would yield better estimates as they are able to track slowly varying parameters. We found that a suitable value for γ is between 0.99 and 0.999. These efficient algorithms can be implemented on a microprocessor-based instrument. In such a system a variety of models could be implemented and the desired model selected by the user. Compared to the present on-line methods, our approach is superior. It does not require any manual adjustment and all the signal processing required for correction is carried out digitally. The removal algorithms are at the present carried out off line,but can be carried out in real time.

REFERENCES

BARLOW,J.S. and RÉMOND,A.(1981) Eye movement artefact nulling in EEGs by multichannel on-line EOG subtraction. Electroenceph. and clin. neurophysiol.,55,418-423

BARLOW,J.S.,CREUTZFELDT,O.D.,MICHAEL,D.,HOUCHIN,J. and EPELBAUM,H. (1981) Automatic Adaptive segmentation of clinical EEGs. Electroenceph. clin. Neurophysiol.,51,512-525.

BARLOW,J.S. (1979) Computerrized Clinical Electroencephalography in perspective, IEEE Trans. Biomed. Engr.,26,377-390

BIERMAN,G.J. (1976) Measurement updating using the U-D factorization, Automatica,12,375-382.

CLARKE,D.W. (1981a). Introduction to self-tuning controllers,in HARRIS,C.J. and BILLINGS,S.A. (Eds.) Self-Tuning and Adaptive Control. Peter Peregrinus,36-71.

CLARKE,D.W. (1981b) Implementation of sef-tuning controllers, ibid, 144-165.

CORBY,J.C and KOPEL,B.S. (1972) Differential contributions of blinks and vertical eye movements as artifacts in EEG recordings. Psychophysiology, 9,640-644.

EYKHOFF,P. (1974) System Identification. John Wiley.

FORTGENS,C. and DE BRUIN M.P. (1983) Removal of eye movement and ECG artifacts from the non-cephalic reference EEG. Electroenceph. clin. Neurophysiol.,56,90-96

GOTMAN,J.,GLOOR,P. and RAY,W.F. (1975) A quatitative comparison of traditional reading of the EEG and interpretation of computer extracted features in patients with supertentorial brain lessions. Electoenceph. clin. Neurophysiol.,38,623-639.

GIRTON,D.G. and KAMIYA,J. (1973) A simple on-line technique for removing eye movement artifacts from the EEG. Electoenceph. Clin. Neurophysiol.,34,212-216.

IFEACHOR,E.C.,JERVIS,B.W.,ALLEN,E.,MORRIS,E.L.,WRIGHT,D. and HUDSON, N.R. (in preparation) Investigation and comparison of some models for removing ocular artefacts from EEG signals.

HARRIS,M.A. (1983) Brain Waves appear on T.V in real-time. Electronics,24,47-48.

HILLYARD,J.A. and GALAMBOS,R. (1970) Eye movement artefact in the CNV. Elctroenceph. Clin. Neurophysiol.,24,173-182.

JERVIS,B.W.,NICHOLS,M.J.,ALLEN,E.,HUDSON,N.R. and JOHNSON,T.E. (1980) The quantitative assessment of electoencephalograms corrected for eye movement artefacts. First European conference on signal processing,Lausanne.

JERVIS,B.W.,NICHOLS,M.J.,ALLEN,E.,HUDSON,N.R. and JOHNSON,T.E. (in preparation) Comparison of two methods for removing eye movement artefact from the EEG.

JERVIS,B.W.,NICHOLS,M.J.,ALLEN,E.,JOHNSON,T.E.,and HUDSON,N.R.(1984) Pattern recognition techniques applicable to the Contingent Negative Variation for the differentiation of subjects categories. IEEE Trans.,Biomed. Engr.,31,342-349.

JOHNSTON,J (1972) Econometric methods. McGraw-Hill.

McCALLUM,W.C. and WALTER,W.G. (1968) The effects of attention and distraction on the contingent negative variation in normal and neurotic subjects. Electroenceph. Clin. Neurophysiol., 25,319-329.

MORRIS,E.L. and ABAZA,B.A. (1976) Adaptive control of steam turbine Proc. I.E.E.,123,549-553.

OVERTON and SHAGASS,C. (1969) Distribution of eye movement and blink potentials over the scalp. Electroenceph. clin. Neurophysiol. 27,546P

PETERKA,C. (1975) A square root filter for real time multivariate regression. Kybernetika,11,53-67.

QUILTER,P.M.,MACGILLIVRAY,B.B and WADBROOK,D.G. (1977) The removal of eye movement artefact from EEG signals using correlation techniques. Random signal analysis,IEE conference publication,NO, 159,93-100.

THORNTON,C.L.,BIERMAN,G.J. (1978) Filtering and error analysis via the UDU covariance factorization. I.E.E.E. Trans. automatic control,5,901-907.

WASMAN,M.,MOREHEAD,S.,LEE,H. and ROWLAND,V. (1970) Interaction of electroocular potentials with the contingent negative variation. Psychophysiology,7,103-111.

WHITTON,J.L.,LUE,F.,MOLDOFSKY,H., (1978) A spectral method for removing eye movement artifacts from the EEG. Electroenceph. clin. Neurophysiol.,44,735-741.

WEERTS,T.C. and LANG, P.J. (1973) The effects of eye fixation and stimulus and response location on the contingent negative variation. Biological Psychology,1,1-19.

Appendix.

Recursive estimation of the residual variance and the multiple correlation coeffcient.

The residual variance, S^2 and the multiple correlation coefficient, 'R^2 for the batch method may be defined as in (A1) (e.g. Johnston 1972):

$$S^2 = SS/M-n \tag{A1a}$$

$$R^2 = SS/SS_T \tag{A1b}$$

where M and n are the number of samples and the number of parameters in the model. SS, the sum of squares of the residuals and SS_T, the total sum of squares are, respectively, given by:

$$SS = \sum_{i=1}^{m} e^2(i) \; ; \; SS_T = \sum_{i=1}^{m} y^2(i)$$

y(i), e(i) are, respectively, the ith samples of the measured and corrected EEG. In the recursive algorithms it is desirable to obtain S^2 and R^2 recursively. We justify here the recursive forms of S^2 and R^2 used in the investigation.

From section 6, the sum of squares at each sample point may be written as:

$$\begin{aligned} SS(m) &= \Upsilon SS(m-1) + e^T(m)e(m) \\ &= \Upsilon SS(m-1) + e^2(m) \end{aligned} \tag{A2}$$

since e(m) is a scalar. Similarly,

$$SS(m-1) = \Upsilon SS(m-2) + e^2(m-1)$$

Thus on substituting into (A2) SS(m) becomes:

$$SS(m) = \Upsilon^2 SS(m-2) + \Upsilon e^2(m-1) + e^2(m)$$

By successive substitution for SS(m-2), SS(m-3) etc and simplifying SS(m) becomes

$$SS(m) = \sum_{i=1}^{m} \Upsilon^{m-i} e^2(i) \tag{A3}$$

which is the same as the expression for the sum of squares for the exponentially weighting scheme. Thus (A2) converges to the required form. In view of (A2) we can write

$$SS(m+1) = \Upsilon\, SS(m) + e^2(m+1) \qquad (A4)$$

By a similar argument, the total sum of squares given by (A5) also converges to the true total sum of squares.

$$SS_T(m+1) = \Upsilon\, SS_T(m) + y^2(m+1) \qquad (A5).$$

In the exponential weighting scheme, the number of samples that effectively contribute to the estimation of O(m+1) at each sample point is called the Asymptotic Sample Lenght (ASL) (e.g Clarke 1981a) and it is given by

$$ASL = \sum_{i=0}^{\infty} \Upsilon^i = \frac{1}{1-\Upsilon} \qquad (A6)$$

Thus m in (A1a) may be replaced by ASL. A recursive form for ASL is given in (A7):

$$asL(m+1) = 1 + \Upsilon\, asL(m) \qquad (A7)$$

By a similar argument to that for the sum of squares, asl(m+1) can be shown to converge to the true ASL,that is:

$$asL(m+1) = 1 + \Upsilon\, asL(m) = \sum_{i=0}^{\infty} \Upsilon^i \qquad (A8).$$

To compute S^2 and R^2 recursively all that is now required is to use equations (A2),(A5) and (A8) as indicated in equation (8) in the main text.

Table 1 Comparison of the parameter estimates for the batch and the UD algorithm. The estimates for the UD algorithm are the values at the end of the iterations.

	Model	$\hat{\theta}_1$	$\hat{\theta}_2$	$\hat{\theta}_3$	$\hat{\theta}_4$
Batch	3D	*	0.091	-0.178	-0.108
	4A	-0.218	0.312	-0.169	-0.053
	4D	0.035	0.000	-0.169	-0.090
UD	3D	*	0.091	-0.177	-0.108
	4A	-0.216	0.310	-0.169	-0.053
	4D	0.035	0.00	-0.160	-0.090

* not applicable

Table 2 Final values of the estimates of the parameters ($\hat{\theta}$), the sample variance (S^2), and the multiple correlation coefficient (R^2) after 1,2 and 3 cycles.

		Parameter estimates					
No of cycles	Model	$\hat{\theta}_1$	$\hat{\theta}_2$	$\hat{\theta}_3$	$\hat{\theta}_4$	S^2	R^2
1	3D 4A 4D	* -0.216 0.035	0.091 0.310 0.000	-0.177 -0.169 -0.160	-0.108 -0.053 -0.090	49.22 47.75 49.87	0.42 0.41 0.43
2	3D 4A 4D	* -0.217 0.035	0.090 0.311 0.000	-0.176 -0.169 -0.160	-0.106 -0.053 -0.090	49.88 48.20 50.36	0.43 0.42 0.43
3	3D 4A 4D	* -0.218 0.035	0.090 0.311 0.000	-0.176 -0.169 -0.160	-0.107 -0.053 -0.090	49.95 48.37 50.53	0.43 0.42 0.44
Batch method	3D 4A 4D	* -.218 0.035	0.091 0.312 0.000	-0.178 -0.169 -0.160	-0.108 -0.053 -0.090	50.34 49.02 51.22	0.56 0.58 0.56

* not applicable

FIGURE LEGEND

Figure 1. Measured EOG and EEG signals for a horizontal eye movement. (AJS36). (i),(ii) vertical and horizontal EOGs of the left eye (V_L and H_L). (iii) the vertex EEG. (iv),(v) vertical and horizontal EOGs of the right eye (V_R and H_R).

Figure 2. Estimates of the ocular artefacts and the background EEG by the conventional batch method . (i),(ii) and (iii) Ocular artefact estimates using models 3D,4A and 4D respectively; (iv),(v) and (vi) the corresponding estimates of the background EEGs.

Figure 3. Estimates of the ocular artefacts and the background EEG by the UD recursive algorithm. (α= 0.05,γ=1.0, $\hat{\beta}$ = 0.1). Other details are the same as for figure 2.

Figure 4. The evolution of the parameter estimates. (i), (ii), and (iii) estimates for models 3D, 4A and 4D, respectively, for the first cycle. (iv), (v) and (vi) the corresponding estimates for the second cycle.

Figure 5. The evolution of the sample variance and the multiple correlation coefficient. (i) and (ii) estimates of the sample variance and multiple correlation coeficient, respectively, for the first cycle. (iii) and (iv) the corresponding estimates for the second cycle.

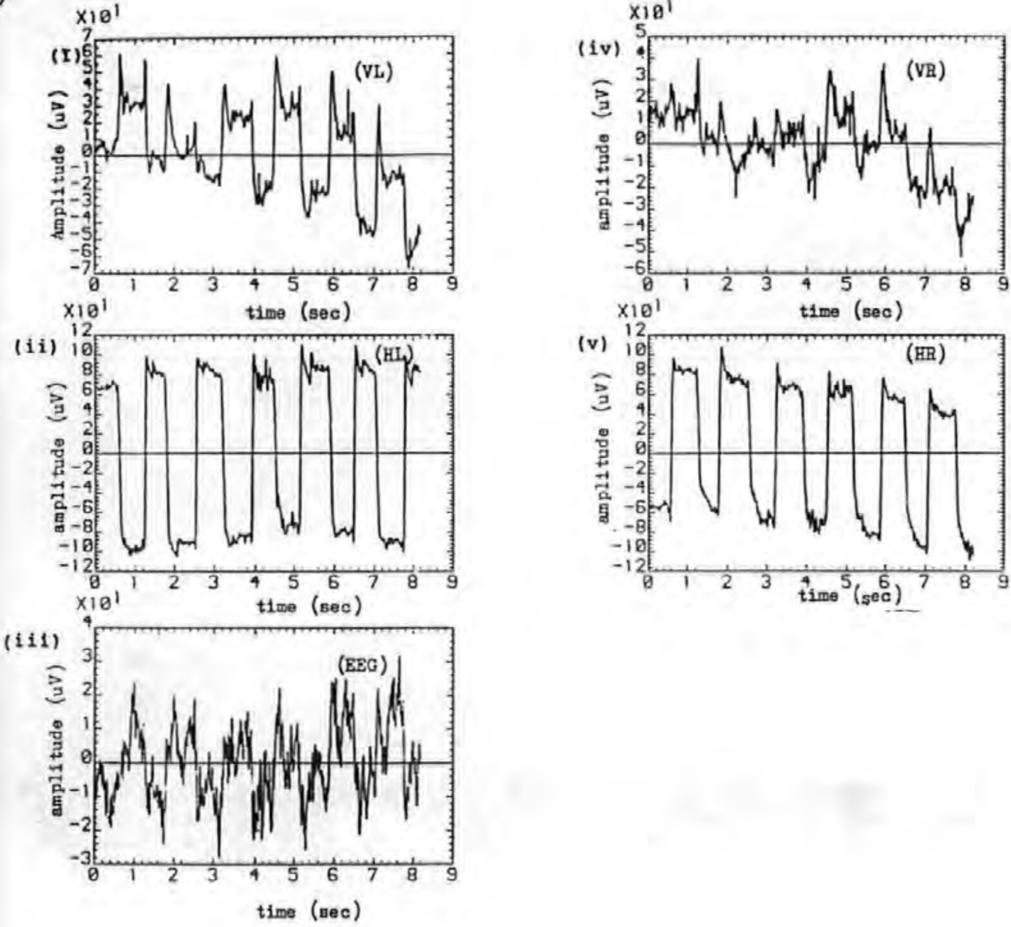
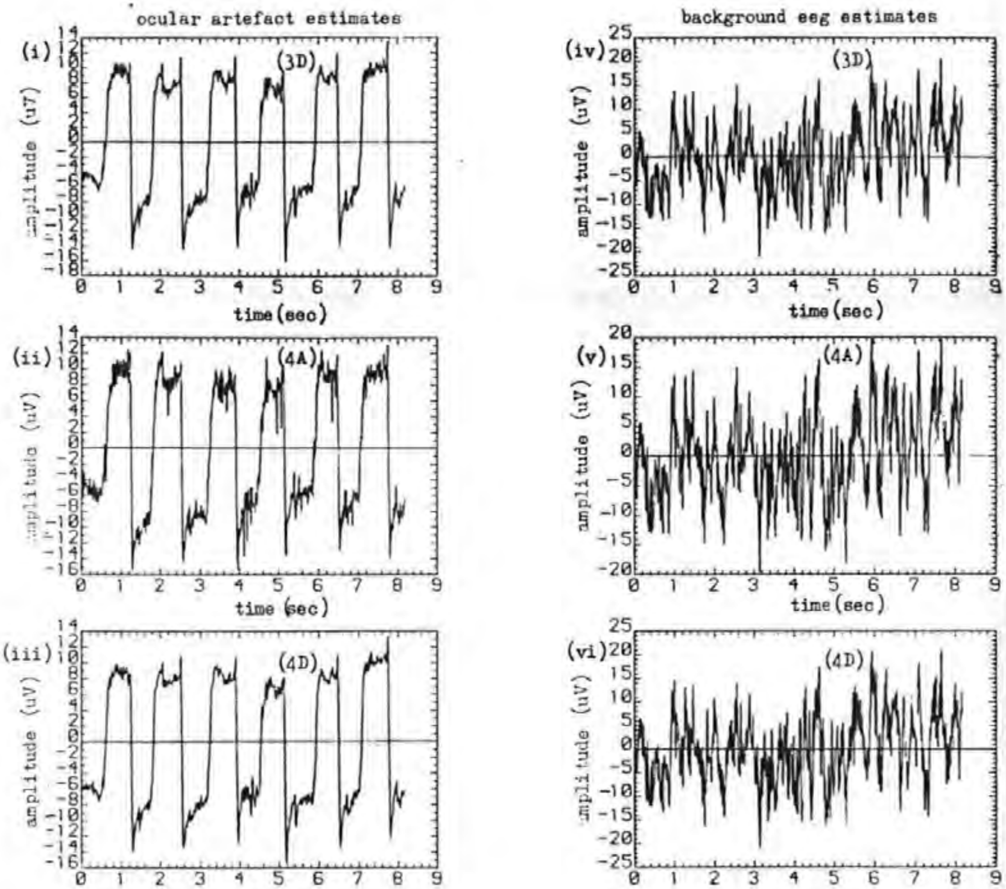

(I)
(VL)
X10¹
Amplitude (uV)
time (sec)
(ii)
(HL)
X10¹
amplitude (uV)
time (sec)
(iii)
(EEG)
X10¹
amplitude (uV)
time (sec)
(iv)
(VR)
X10¹
amplitude (uV)
time (sec)
(v)
(HR)
X10¹
amplitude (uV)
time (sec)

Figure 1



ocular artefact estimates
background eeg estimates
(i)
(3D)
amplitude (uV)
time (sec)
(ii)
(4A)
(iii)
(4D)
(iv)
(v)
(vi)
0 1 2 3 4 5 6 7 8 9
14 12 10 8 6 4 2 0 -2 -4 -6 -8 -10 -12 -14 -16 -18
25 20 15 10 5 0 -5 -10 -15 -20 -25

Figure 2

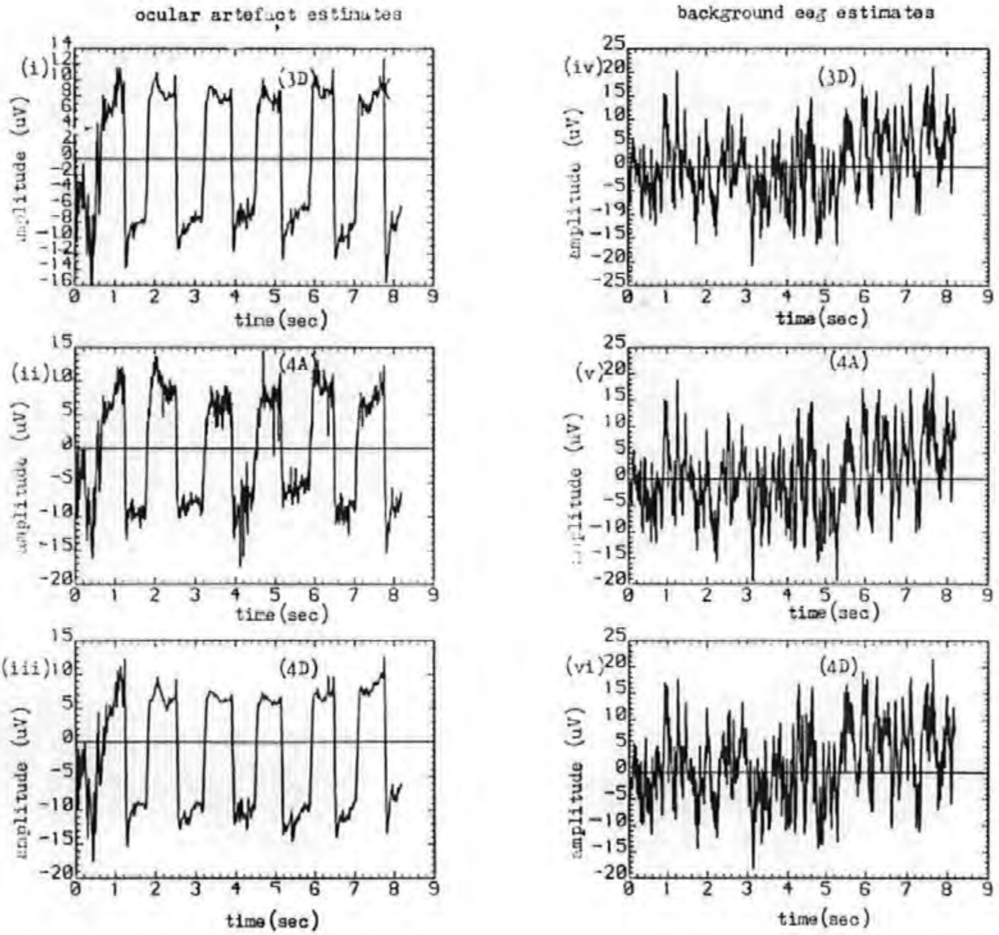

ocular artefact estimates
background eeg estimates
(i)
(3D)
amplitude (uV)
14
12
10
8
6
4
2
0
-2
-4
-6
-8
-10
-12
-14
-16
0 1 2 3 4 5 6 7 8 9
time(sec)
(ii)
(4A)
amplitude (uV)
15
10
5
0
-5
-10
-15
-20
0 1 2 3 4 5 6 7 8 9
time(sec)
(iii)
(4D)
amplitude (uV)
15
10
5
0
-5
-10
-15
-20
0 1 2 3 4 5 6 7 8 9
time(sec)
(iv)
(3D)
amplitude (uV)
25
20
15
10
5
0
-5
-10
-15
-20
-25
0 1 2 3 4 5 6 7 8 9
time(sec)
(v)
(4A)
amplitude (uV)
25
20
15
10
5
0
-5
-10
-15
-20
0 1 2 3 4 5 6 7 8 9
time(sec)
(vi)
(4D)
amplitude (uV)
25
20
15
10
5
0
-5
-10
-15
-20
0 1 2 3 4 5 6 7 8 9
time(sec)

Figure 3

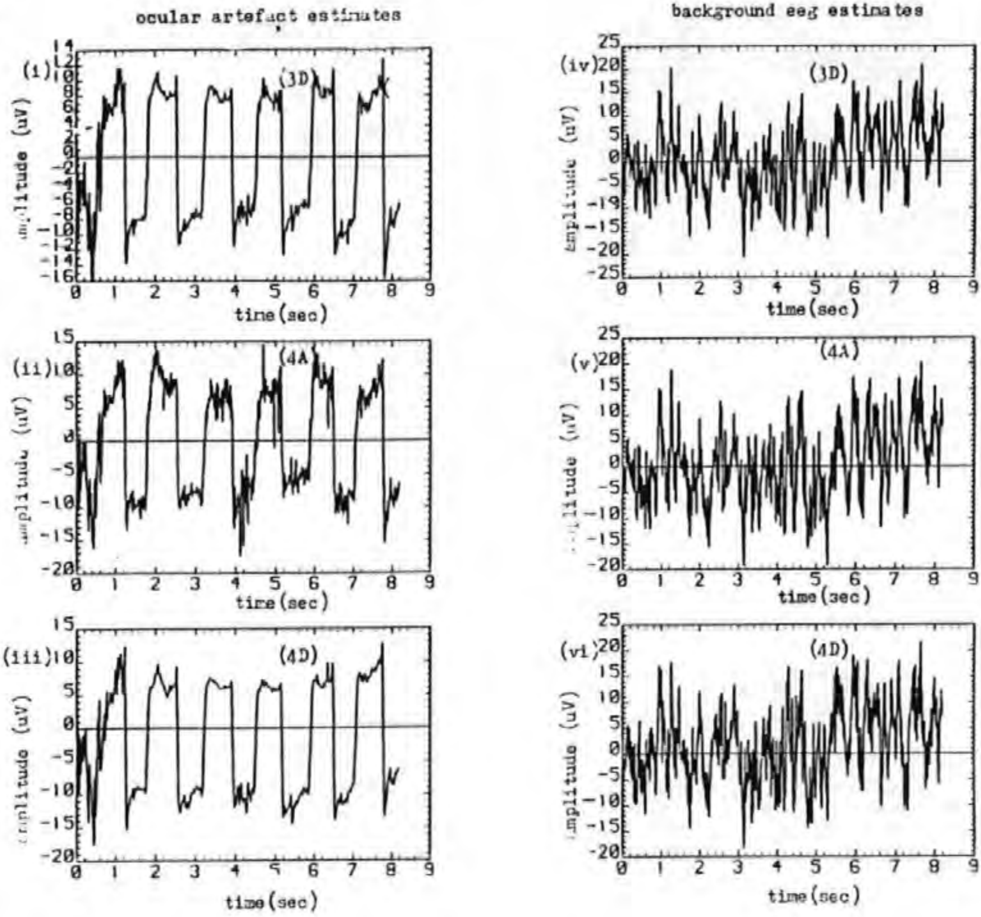

ocular artefact estimates
background eeg estimates
(i)
(3D)
(ii)
(4A)
(iii)
(4D)
(iv)
(3D)
(v)
(4A)
(vi)
(4D)
amplitude (uV)
time (sec)
14
12
10
8
6
4
2
0
-2
-4
-6
-8
-10
-12
-14
-16
15
5
-5
-15
-20
25
20
-25
0
1
2
3
4
5
6
7
8
9

Figure 3

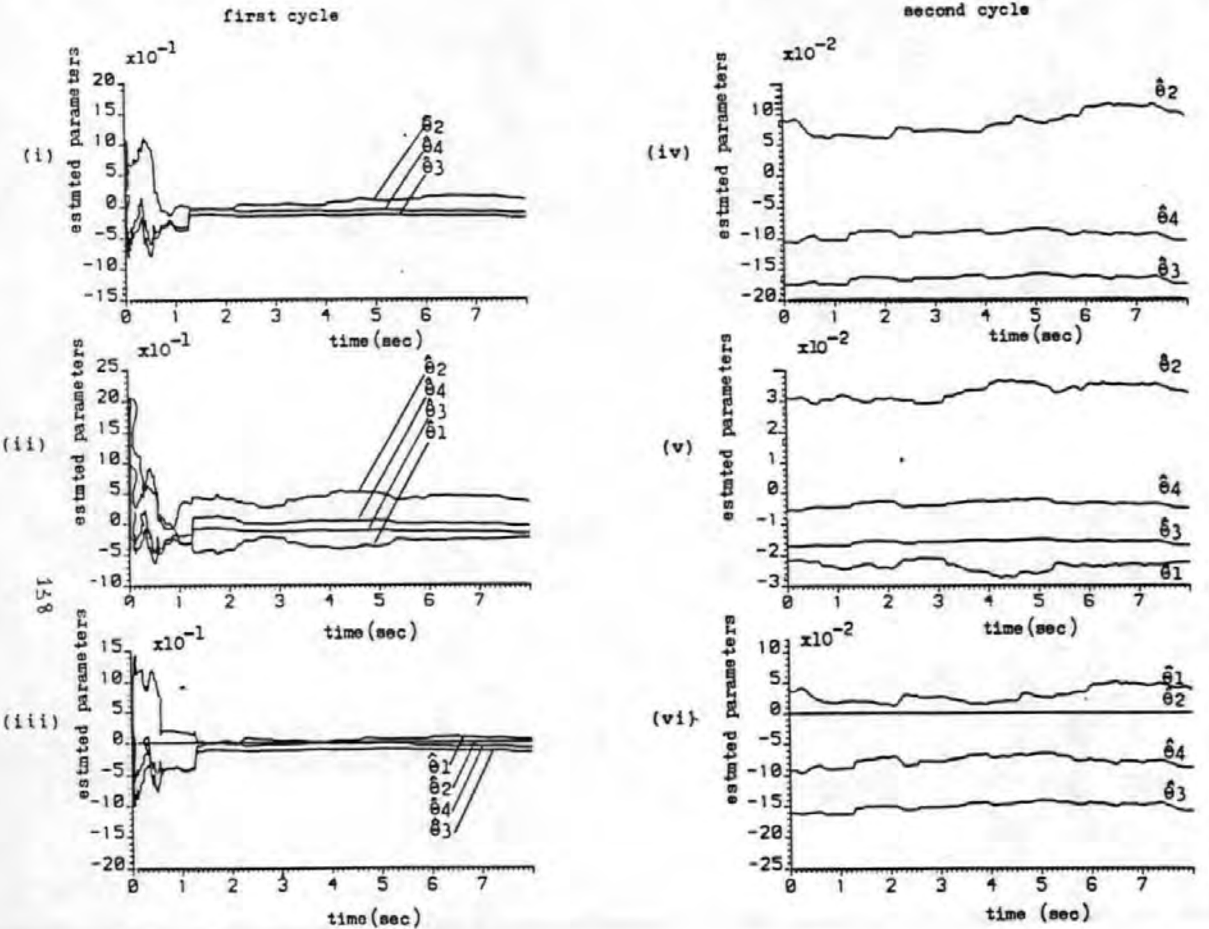

first cycle
second cycle
(i)
(ii)
(iii)
(iv)
(v)
(vi)
x10^-1
x10^-2
estmted parameters
time(sec)
θ̂1
θ̂2
θ̂3
θ̂4
0 1 2 3 4 5 6 7

Figure 4

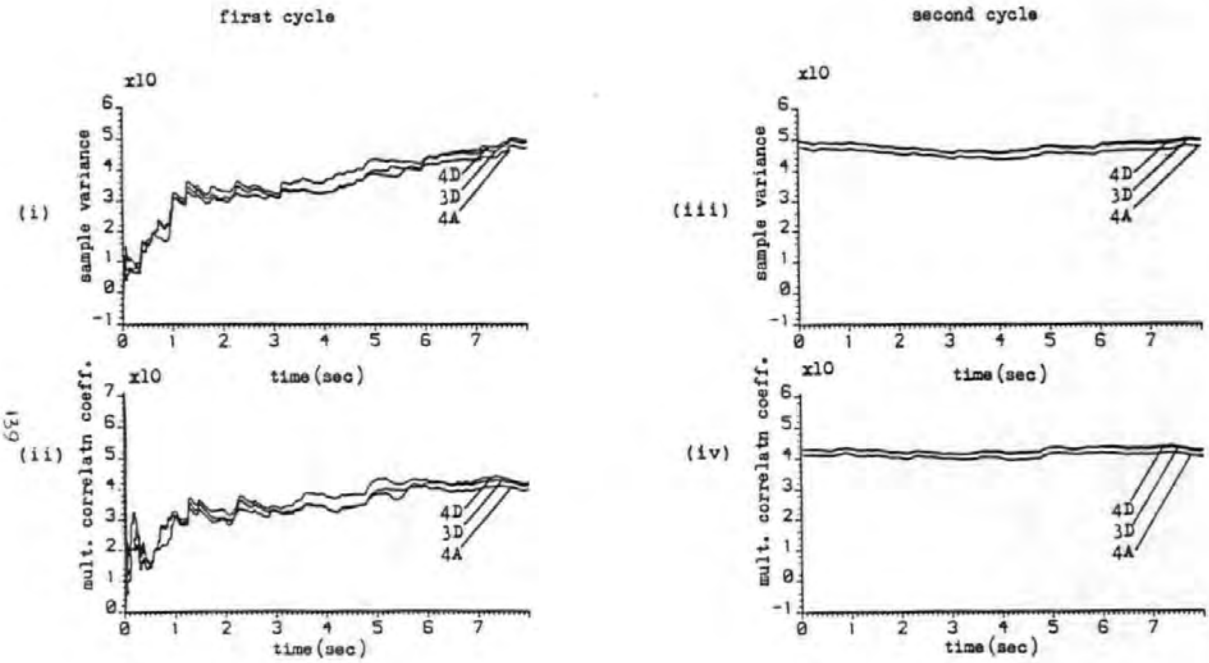

first cycle
second cycle
(i)
(ii)
(iii)
(iv)
x10
sample variance
mult. correlatn coeff.
time (sec)
6
5
4
3
2
1
0
-1
7
0
1
2
3
4
5
6
7
4D
3D
4A

Figure 5

SOFTWARE FLOATING POINT ARITHMETIC ROUTINES FOR THE MOTOROLA 68000 MICROPROCESSOR.

1.0 Introduction

Fixed point and floating point are the two most common types of arithmetic in a digital system (Flores 1963,Rabiner and Gold 1975). Fixed point representation has the advantage of being fast, but limits the range of numbers that can be represented. To prevent results of arithmetic operations going outside the number range, the operands have to be scaled before and/or after each arithmetic operation. To overcome the problem of scaling associated with fixed point approach floating point reprentation is often used. Floating point scheme greatly increases the dynamic range of the numbers and it is preferred in applications where the magnitudes of numbers vary widely (Flores 1963).

In some control and signal processing applications, it is often desired to implement on small computer systems algorithms which have been developed on large systems, such as a main frame computer. On the large systems floating point representation is invariably used. In these cases the designers of the algorithms have to face the problem of number representation and make the difficult decision of whether to use floating point or fixed point representation. Such was the case in an ongoing research project in which it was desired to implement a real-time algorithm on a single card processor. In this case, the decision was in favour of floating point representation. As speed was vital in our application, hardware floating point was considered the best approach, but we found that hardware floating point devices available at the time were both expensive and too slow. Therefore, the system for which the algorithm was intended was scaled down so that sofware floating point could be used until fast floating point devices became available.

We describe in this paper binary floating point routines which were used to perform the basic arithmetic operations of add/subtract,multiply and divide in 68000 microprocessor assembly language. The floating point number representation and the data format we have adopted will be described in section 2 and in section 3 we describe the floating point arithmetic routines. In a companion paper, we shall describe routines for converting from floating point (in Hexadecimal) to fixed point (in decimal) which we found useful.

2.0 Binary floating point representation

A floating point number X is represented as the product of two signed numbers,the mantissa M and the exponent, E:

$$X = M \; . \; 2^{E} \qquad (1)$$

where 2 is the base of the binary system.

The exponent primarily determines the range of the numbers that can be represented, whereas the mantissa determines the accuracy of the numbers. In our system, the exponent and the mantissa are represented respectively, by 8 and 16 bits. This choice is a suitable compromise between speed and accuracy. Similar choice of exponent and mantisa lengths have been used by Clarke et al (1975) in a control application and were found to be adequate. The range of the floating point numbers is the difference between the largest and the smallest numbers that can be represented,which in our case is:
(0.5×2^{-64}) to $(1-2^{-15}) \times 2^{63}$,or 2.7105×10^{-19} to 0.9223×10^{19} .
Of the 16 bits used to represent the mantissa, 1 bit is the sign bit and the least significant bit may be of a doubtful accuracy due to rounding effects. Thus the accuracy of the floating point numbers is

1 in 2^{14} (0.61×10^{4}),that is about 4 decimal digits.

2.1 Floating point data format.

The format for the floating point number (or word) is given in figure 1. The mantissa is expressed as a 16 bit fractional two's complement value with the binary point assumed to the right of the sign bit.. The exponent is 8 bits long in excess 64 form. The exponent normally has a range of $-64 \leq E \leq 63$,but by adding a fixed constant to the exponent (in this case 64) such that it is always positive the range becomes: $0 \leq E \leq 127$. This ensures that when the mantissa is zero the exponent is also zero,giving an all zero floating point number. Additionally, this form of representation has the advantage of making the detection of over- and under-flow simple (e.g Cope 1975). However,when performing arithmetic operations the exponent has to be restored to its unbiased form.

All numbers going into or out of the floating point routines must be normalized. A normalized floating point number is one in which the sign bit and the bit to the right of the binary point (bits 8 and M14) of the mantissa are different. In our system, a normalized mantissa satisfies the condition : $0.5 \leq |M| < 1$. If the result of an arithmetic operation produces a mantissa that is outside this range, it is said to have overflowed (mantissa overflow) if the result is equal to or greater than unity, and underflowed (mantissa underflow) if it is less than 0.5.

The floating point numbers in memory are allocated four consecutive bytes of memory (2 words), although only the first three are used as shown in figure 1. This is to simplify reading and writing to the memory of the 68000 μp. It also allows the number of bits in a floating point word (mantissa + exponent) to be increased easily, if desired.

3.0 Floating point arithmetic routines.

The floating point arithmetic routines consist of three routines for the basic arithmetic operations : add/subtract,multiply and divide. Exit from these routines is made via a common routine, EXIT.

On entry into any of the arithmetic routines, registers D3 and D2 contain,respectively, the first and second operands. In the case of division, D3 and D2 contain respectively,the dividend and the divisor. On exit, D3 contains the result. Apart from D3,none of the registers are modified on exit by the floating point routines.

The 68000 μP has an abundance of registers and these have been used freely to improve the execution speed. For the same reason, the use of some otherwise efficient instructions have been avoided. Arithmetic operations can be chained with the result of the previous operation left in D3 to be used as the first operand of the next operation.

3.1 EXIT routine.

This routine is used to handle "exceptions" such as overflow, to recover the sign of the result ,to put the result in the correct floating point format and to return control to the calling routine. Three exceptions are handled:

(a) Result is zero or too small.

If the result of an operation is zero or the number is too small to be represented (under flow) this routine sets the result to zero.

(b) Result overflowed.

If there is overflow in the arithmetic operation, the result is set to the maximum floating point number possible. That is, $(1-2^{-15}) \times 2^{63}$. In practice it is set to $007F7FF, then the mantissa is given the appropriate sign.

(c) Over/under flow detection.

In excess 64 representation, the exponent of a valid floating point number is positive. Over-/under-flow causes it to be negative. Thus the sign of the exponent is used to detect over/under flow. Additionally, underflow generates a carry bit whereas over flow does not. The EXIT routine determines which of the two has occured and sets the result to either FPMAX or to zero, which ever is appropriate.

Figure 2 gives a listing of the EXIT routine, in 68000 up assembly language.

3.2 Floating point Addition routine (FADD).

This routine is used to perform addition or subtraction. In the case of subtraction the second operand is negated in the calling routine with a single instruction (eg NEG.W D2) before entering the addition routine. Before adding two floating point numbers, their exponents must be made equal. This is called alignment, and involves shifting right the mantissa of the smaller operand and incrementing its exponent until it equals that of the larger operand. The aligned numbers are then added. Thus if X1 and X2 are two floating point numbers to be added,where $X1 = M1 \times 2^{E1}$ and $X2 = M2 \times 2^{E2}$,then their sum X,is given by

$$X = (M1 + M2 \times 2^{E1-E2}) \times 2^{E1} = M \times 2^{E}$$

where $M = (M1+M2 \times 2^{E1-E2})$, $E = (E1 - E2)$ and X1   X2.
That is, the exponent of the result is the exponent of the larger operand and its mantissa is the sum of the mantissas of the two operands properly aligned.

The sum of the aligned mantissas ,M, may not be properly normalized. It may be too large (mantissa overflow) or it may be too small (mantissa underflow). The former results when the mantissas of the original operands to be added (M1,M2) have the same signs, and the later when they have opposite signs. Mantissa overflow and underflow are corrected by renormalizing the sum and adjusting the result exponent,E,but this may lead to exponent overflow or underflow of the exponent. Exponent overflow means that the result is too large to be represented and underflow that it is too small.

Figure 3 gives the flow chart for the FADD routine and the corresponding assembly program is given in figure 4 .The first operation is to align the operands so that their exponents are equal (boxes 1 to 5). This is done by shifting the mantissa of the operands with the smaller exponent right an amount equal to the difference in the exponents (the shift count). If the two exponents were equal or their difference exceeds 15 (operand too small),no alignment is carried out. In the later case the operation is terminated as the smaller operand is effectively zero and the result is simply the larger operand. Next the aligned numbers are added (box 6) and the exponent of the result set equal to the exponent of the larger operand.

If the sum is not zero, it is then tested for mantissa overflow or underflow. The two cases are treated separately and as mentioned earlier depend on the signs of the operands. Boxes 10 to 15 deal with mantissa overflow, whilst boxes 16 to 19 deal with mantissa underflow.

Mantissa overflow is detected by successively performing exclusive-OR operation on the sign bits of the two original operands, the sum ,M, and the carry bit. The result of this operation is a (1) if there was overflow and (0) otherwise (Cope 1975). Overflow is corrected by shifting the sum,M, right with carry and incrementing the result exponent,E by one (box 14). If the exponent overflows the result is set to FPMAX with the appropriate sign, otherwise the resulting number is now properly normalised. A special form of overflow may occur when the sum,M, is exactly -1 ($8000 in hexadecimal) or NEGMAX, and this is separately detected. NEGMAX is the maximum negative mantissa and has no positive equivalent. Thus when negated it still remains negative (boxes 10 and 11). When this condition is detected the sum,M, is normalised as in the ordinary mantissa overflow.

Mantissa underflow is corrected by successively shifting left the sum ,M,and decrementing the exponent,E, until the sum is properly normalized (box 18). The exponent is tested for underflow at each shift. If exponent underflow occurs at any stage the operation is terminated and the result set to zero. The sum is made positive (box 17) before normalizing to simplify the operation. The sign is recovered in the EXIT routine where the result exponent,E and the normalized sum,M are put into the proper floating point format.

3.3 Floating point Multiply routine (FMUL)

If X1 and X2 are two FP numbers to be multiplied,where

$X1 = M1 \times 2^{E1}$

$X2 = M2 \cdot 2^{E2}$

then their product X, is given by : $X = (M1 \times M2)\ 2^{E1+E2} = M \cdot 2^{E}$ where $M = (M1 \times M2)$, $E = E1 + E2$. Thus the mantissas are multiplied and their exponents added. Since M1 and M2 are both normalized then their product,M, will be in the range: $0.25 \leq M < 1$. Thus the product,M cannot overflow but may not be properly normalized (mantissa underflow).

The flowchart for the floating point multiply routine is given in figure 5 and figure 6 gives the assembly language program. The mantissas of the operands are multiplied and the product tested to see if it is zero (boxes 1 and 2). If it is zero the operation is terminated and the result set to zero. The product,M, is double length (multiplication of two n-bit numbers gives a 2n bit result) and is thus rounded and reduced to single length. The exponents are then added and corrected for excess 64 (box 4). If the product M, is not normalised it is then normalized by a single left shift and the exponent decremented by 1. The sign of the exponent is tested. If it is negative, then there was exponent overflow or underflow when the exponents were added. The EXIT routine then determines whether it was underflow or overflow that occurred and takes appropriate action. It is to be noted that exponent overflow or underflow was not tested for when the exponents were added as subsequent exponent adjustment could correct it. The normalized product,M and the adjusted exponent give the result for the floating point multiplication, except when the exponent overflows or underflows.

3.4 Floating Point Divide Routine (FDIV).

In floating point division the mantissas are divided and the exponent subtacted .Thus, division of X1 by X2 where

$$X1 = M1 \times 2^{E1}$$

$$X2 = M2 \times 2^{E2}$$

gives, $X = \frac{M1}{M2} \times 2^{E1-E2} = M \times 2^{E}$

where $M = \frac{M1}{M2}$,E = E1-E2.

To ensure that the quotient is within permissible range M1 must be less than M2.This is readily achieved by shifting M1 one place to the right. After division, the quotient,M may not be properly normalized. If it is not, it is shifted left one place and the exponent E, decremented by one and checked for underflow as before.

Figure 7 gives the flowchart for the floating point divide routine. The exponent,E is obtained from the exponents of the operands and the bias is added (box 1). The two mantissas are then tested. If either is a zero, the operation is terminated and the result set to zero (box 2 to 4). The exponent of the result,E is then tested for under- and over flow and treated in the same way as in the multiply.

For easy comparison of the mantissas and normalization of the product the mantissas of the operands are always made positive (boxes 3 and 5). The sign is restored in the EXIT routine. Division is performed with the inherent divide instruction of the 68000 up processor (box 10). Figure 5 gives a listing of the assembly language program for the floating floating point divide routine.

3.5 Execution times of the floating point routines.

Estimates of the execution times of the floating point routines are given below. These times are only approximate as the execution times depend on the data.

Routine	Execution time (µS)
Addition	70
Multiplication	70
Division	85

RFERENCES

FLORES,I. The logic of computer arithmetic. Prentice Hall,1963.

COPE,S.N. Floating-point arithmetic routines and macros for an Intel 8080 microprocessor. O.U.E.L. report 1123/75. ,1975.

CLARKE,D.W., COPE,S.N. and GAWTHROP,P.J. Feasibility study of the application of microprocessors to self-tuning controllers O.U.E.L. Report 1137/75,1975.

RABINER,L.R. and GOLD,B. Theory and application of digital signal processing. Prentice-Hall,1975.

MC68000 16-Bit Microprocessor User's Manual. Austin,TX., Motorola Semiconductor,Inc.,1980.

e7	e6	...	e0	S	m14	m13	...	m0

exponent (8bits)   mantissa (16bits)

(a)

Address	XX
Addrass+1	Exponent
Address+2	High byte of mantissa
Address+3	LOw byte of mantissa

(b)

Figure 1   (a) Floating point data format (b) Floating point memory allocation.

```
*
*
* EXIT ROUTINE
* THIS ROUTINE HANDLES EXCEPTIONS FROM FLOATING POINT ARITHMETIC
* OPERATIONS AND RETURNS CONTROL TO THE CALLING ROUTINE.
*
DFORMAT SWAP.W  D1                        ;PUT RESULT IN CORRECT FP FORMAT
        MOVE.W  D3,D1
        MOVE.L  D1,D3
SIGNS:  SUB.B   #1,D5                     ;RECOVER SIGN OF MANTISSA
        BNE     EXITF
        NEG.W   D3
EXITF:  MOVEM.L (A7)+,D0-D2/D4-D7
        RTS
ZEROF:  CLR.L   D3                        ;RESULT IS ZERO
        BRA     EXITF
UOFLW:  BTST    #15,D6                    ;RESULT UNDER OR OVERFLOW?
        BNE     ZEROF                     ;UNDERFLOW.SET RESULT TO ZERO
OFLW:   MOVE.L  FPMAX,D3                  ;OVERFLOW.SET RESULT TO FPMAX
        BRA     SIGNS
```

Figure 2 The EXIT routine
Control is returned to the calling program via this routine.
The routine also handles under/overflow, recovers the sign
and puts the result in the correct format.

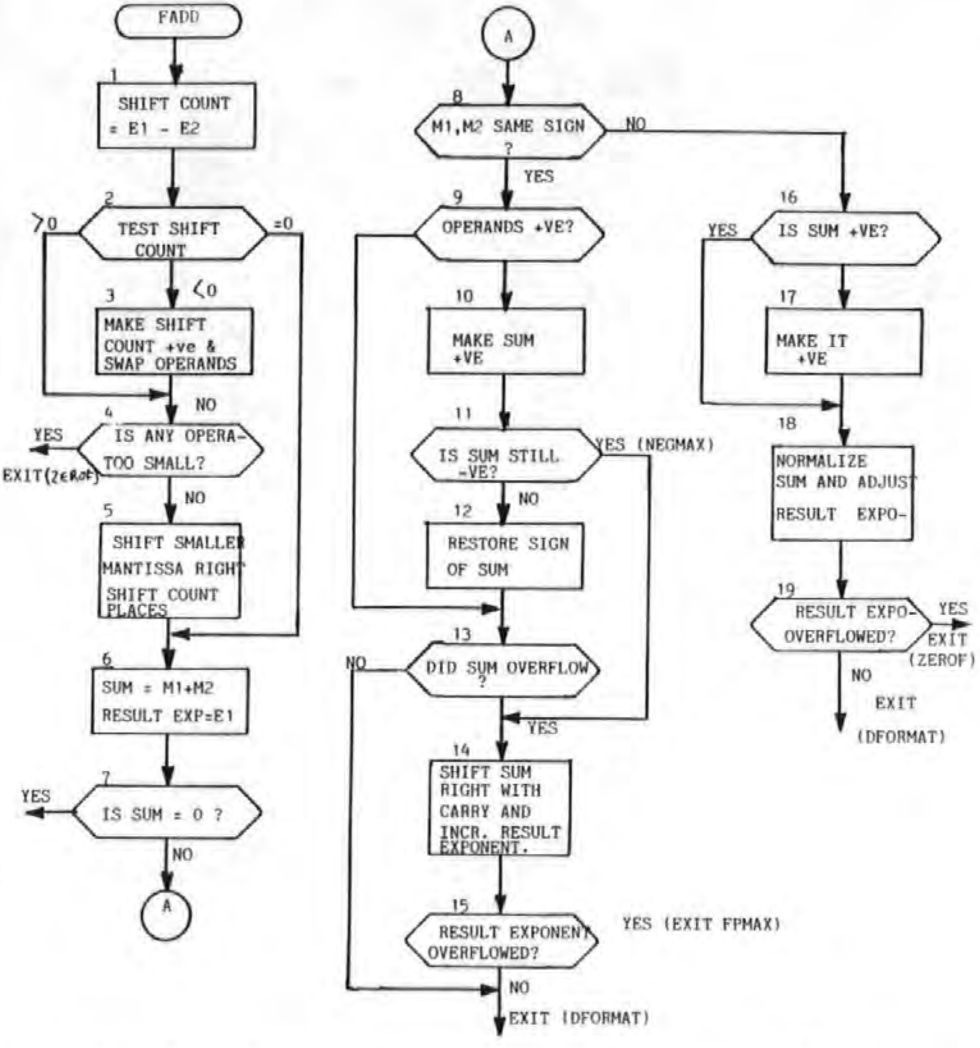


Figure 3 Floating point addition of two operands with exponents E1 and E2 mantissaa M1 and M2, respectively.

```
*----------------------------------------------------------------------
* SUBROUTINE FADD
* THIS SUBROUTINE ADDS TWO NORMALIZED FLOATING POINT NUMBERS IN
* D2 AND D3. THE RESULT IS RETURNED IN D3
*
FADD:    MOVEM.L D0-D2/D4-D7,-(A7)
         MOVE.L  D3,D1                    ;MAKE COPIES OF OPERANDS
         MOVE.L  D2,D0                    ;
         SWAP.W  D1                       ;RETRIEVE EXPONENTS (E) OF OPERANDS
         SWAP.W  D0
         MOVE.B  D1,D4
         SUB.B   D0,D4                    ;E2-E1=SHIFT COUNT(SC)
         BEQ     SCZRO                    ;SC IS ZERO
         BPL     SCPOS                    ;SC IS POSITIVE
         NEG.B   D4                       ;SC IS NEG;MAKE POSITIVE
         EXG     D0,D1                    ;SWAP OPERANDS
         EXG     D2,D3
SCPOS:   CMP.B   #15,D4                   ;ANY OPERAND TOO SMALL?
         BPL     EXITF                    ;YES. RESULT = LARGER OPERAND
         ASR.W   D4,D2                    ;NO. SHIFT SMALLER M RIGHT SC PLACES
SCZRO:   MOVE.W  D3,D6                    ;SAVE SIGN OF 2ND OP
         ADD.W   D2,D3                    ;OBTAIN SUM=M2+M1
         ROXR.W  #1,D5                    ;SAVE CARRY FLAG
         TST.W   D3                       ;IS SUM=0?
         BEQ     ZEROF                    ;IF YES,THEN EXIT
         EOR.W   D2,D6                    ;NO. ARE THE TWO OPERNDS SAME SIGN?
         BMI     DFSIGN                   ;OPERNDS ARE OPPOSITE SIGN
         MOVE.W  D5,D4                    ;OPERNDS ARE SAME SIGN
         SUB.W   D5,D5
         TST.W   D2                       ;IS SUM +VE OR -VE?
         BPL     OPRPOS                   ;SUM IS E +VE.
         NEG.W   D3                       ;SUM IS -VE. MAKE SUM +VE
         TST.W   D3                       ;IS SUM STILL -VE?
         BMI     NEGMX                    ;YES. SUM MUST BE NEGMAX
         NEG.W   D3                       ;NO.RECOVER SIGN OF SUM
OPRPOS:  EOR.W   D4,D6                    ;DID OVERFLOW OCCUR DURING ADDITION?
         EOR.W   D3,D6                    ;
         BPL     DFORMAT                  ;NO. THEN FORMAT RESULT AND EXIT
         ROXL.W  #1,D4
NEGMX:   ROXR.W  #1,D3                    ;SHIFT SUM RIGHT WITH CARRY AND
         ADDQ.B  #1,D1                    ;INCR EXP: E=E+1
         BMI     OFLW                     ;CHECK THAT E DID NOT OVERLOW
         BRA     DFORMAT                  ;E IS OK. FORMAT RESULT AND EXIT.
DFSIGN:  SUB.B   D5,D5
         TST.W   D3                       ;IS SUM POSITIVE?
         BPL     FNORMA                   ;YES. NORMALIZE RESULT
         MOVEQ.L #1,D5                    ;NO. SET FLAG AND MAKE POSITIVE
         NEG.W   D3
FNORMA:  MOVEQ.L #15,D0                   ;SET UP REGS FOR NORMALIZTN
         MOVE.L  D0,D2
SHFTLA   ADD.W   D3,D3                    ;NORMALIZE SUM AND
         DBMI    D0,SHFTLA
         SUB.B   D0,D2
         LSR.W   #1,D3                    ;
         SUB.B   D2,D1                    ;ADJUST EXPONENT
         BMI     ZEROF                    ;E UNDERFLOW?
         BRA     DFORMAT                  ;NO.FORMAT RESULT AND EXIT
```

Figure 4

The Assembly language program for the floating point Addition.

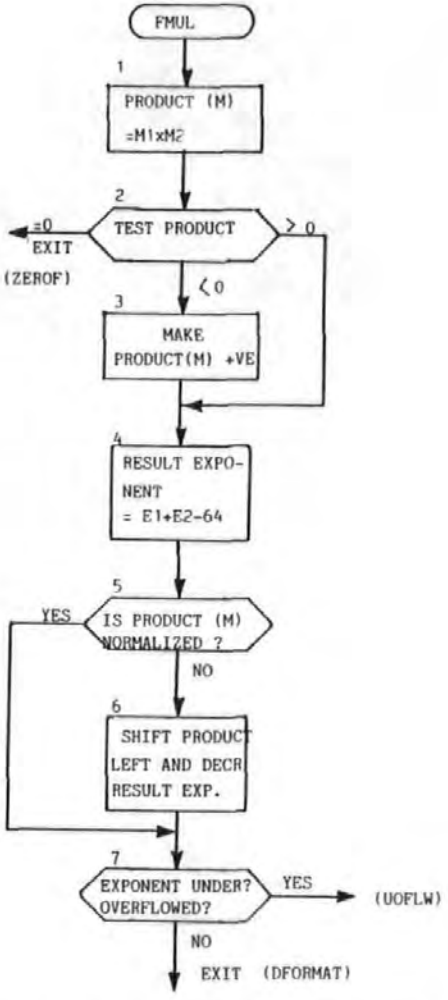


Figure 5 Floating point multiplication of two operands with exponents E1 and E2, mantissas M1 and M2, respectively.

```
*---------------------------------------------------------------------
*
* SUBROUTINE FMUL
* THIS SUBROUTINE MULTIPLIES TWO NORMALIZED FLOATING POINT NUMBERS
* IN D2 AND D3 AND RETURNS THE RESULT IN D3.
*
FMUL:   MOVEM.L D0-D2/D4-D7,-(A7)
        MOVE.L  D3,D1                   ;MAKE COPIES OF OPERANDS
        MOVE.L  D2,D0
        SWAP.W  D1                      ;RETRIEVE THE EXPS.
        SWAP.W  D0
        SUB.B   D5,D5                   ;CLEAR FLAG
        MULS    D2,D3                   ;OBTAIN PRODUCT M1*M2
        ADD.L   D3,D3                   ;SHIFT DLENGTH PRODUCT LEFT
        ADD.L   #32768,D3               ;ROUND RESULT
        SWAP.W  D3                      ;CONVERT RESULT TO SINGLE LENGTH
        TST.W   D3                      ;RESULT=0,+VE OR -VE?
        BEQ     ZEROF                   ;RESULT IS =0. EXIT
        BPL     POSMUL                  ;RESULT IS +VE.COMPUTE EXP.
        ADDQ.B  #1,D5                   ;RESULT IS -VE.SET FLAG
        NEG.W   D3                      ;MAKE POSITIVE
POSMUL: ADD.B   D0,D1                   ;E2+E1=E
        SUB.B   #64,D1                  ;REMOVE XS64
        ROXR.W  #1,D6                   ;SAVE CARRY FLAG
        BTST    #14,D3                  ;IS PRODUCT NORMALIZED?
        BNE     TESTE                   ;YES. TEST EXP.
        ADD.W   D3,D3                   ;NO.NORMALIZE PRODUCT AND
        SUBQ.B  #1,D1                   ;ADDJUST EXPONENT
TESTE:  TST.B   D1                      ;EXP UNDER/OVER FLOWED?
        BMI     UOFLW                   ;YES. BRACH TO UNDER/OVERFLOW TEST
        BRA     DFORMAT                 ;NO. FORMAT DATA AND EXIT.
```

Figure 6
The Assembly language program for the floating point multiply

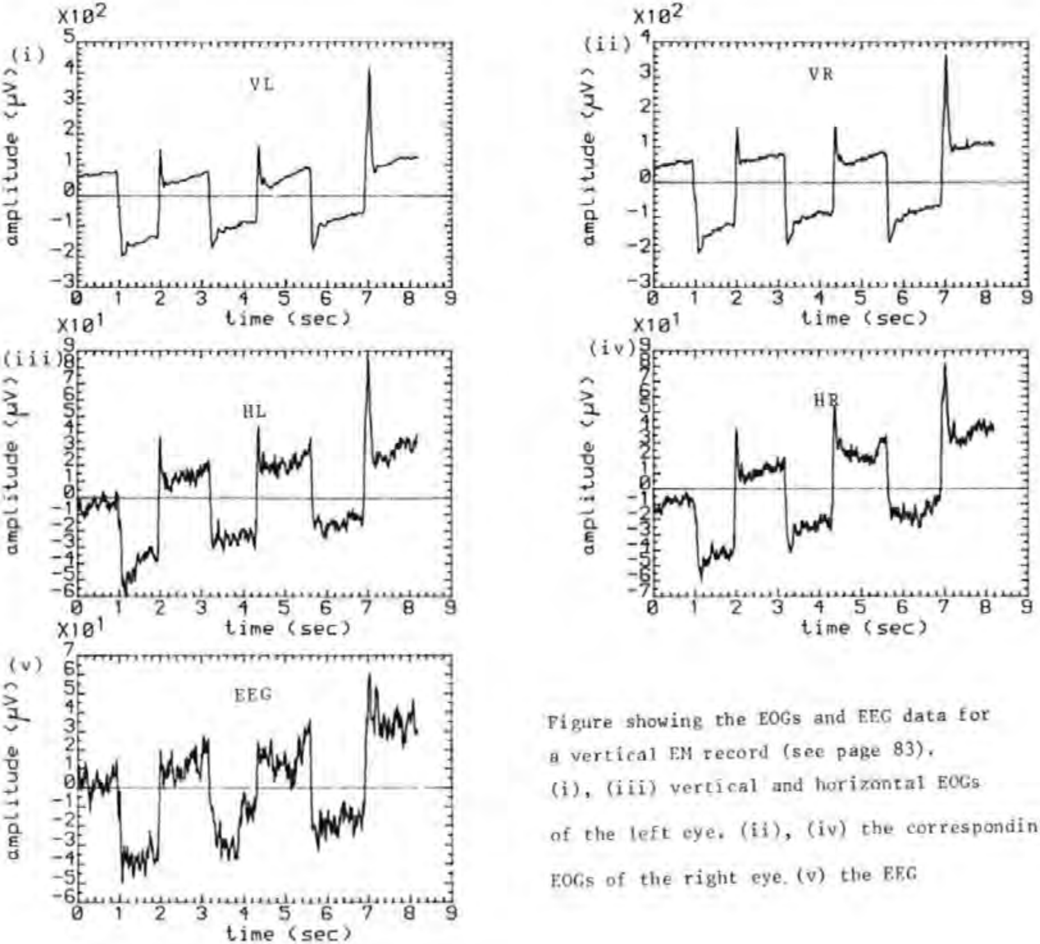


Figure showing the EOGs and EEG data for a vertical EM record (see page 83). (i), (iii) vertical and horizontal EOGs of the left eye. (ii), (iv) the corresponding EOGs of the right eye. (v) the EEG

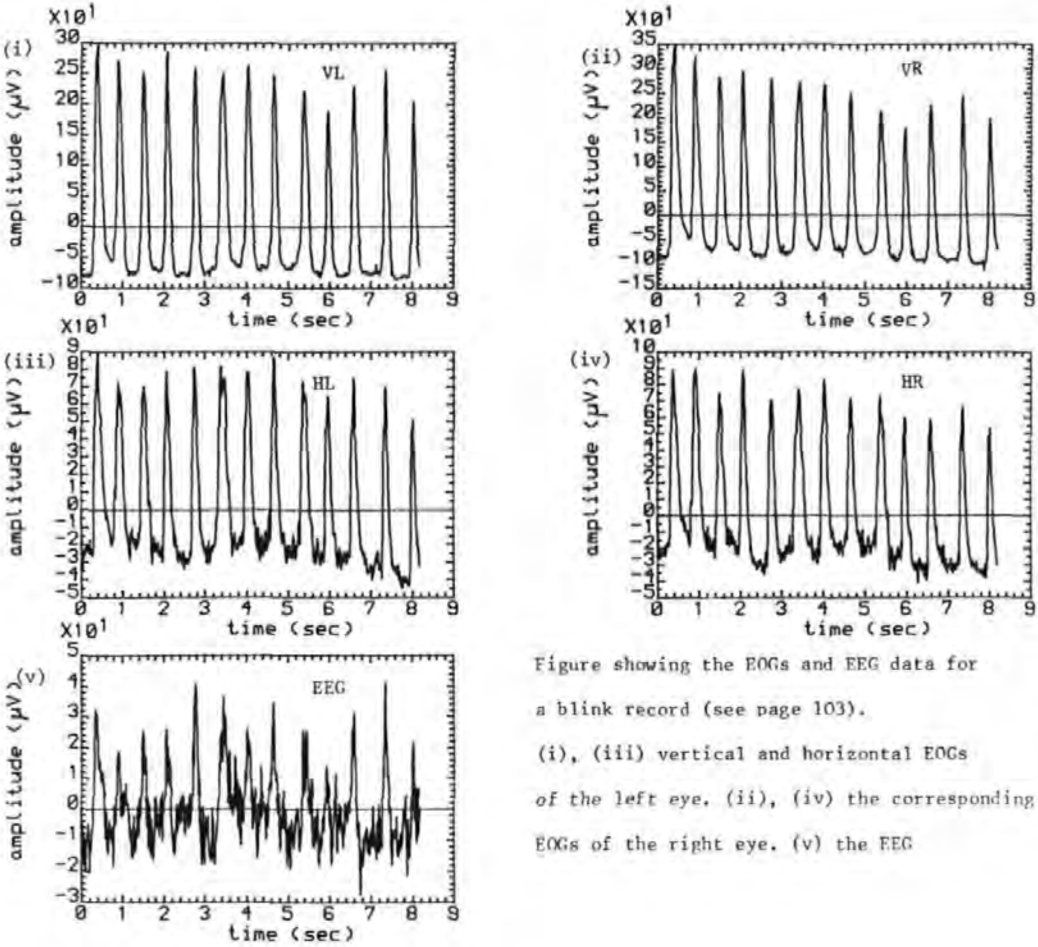


Figure showing the EOGs and EEG data for a blink record (see page 103). (i), (iii) vertical and horizontal EOGs of the left eye. (ii), (iv) the corresponding EOGs of the right eye. (v) the EEG