ADAPTIVE NOISE CANCELLATION OF BRAINSTEM AUDITORY EVOKED POTENTIALS

ADAPTIVE NOISE CANCELLATION OF BRAINSTEM AUDITORY EVOKED POTENTIALS USING SYSTOLIC ARRAYS

by

ROBERT CHARLES SCOTT, B.Eng (Eng Physics)

A Thesis

Submitted to the Faculty of Graduate Studies

in Partial Fulfillment of the Requirements

for the Degree

Master of Engineering

McMaster University

May, 1987

Master of Engineering (1987) McMaster University

(Electrical Engineering) Hamilton, Ontario

TITLE: Adaptive Noise Cancellation of Brainstem Auditory Evoked Potentials Using Systolic Arrays

AUTHOR: Robert C. Scott B.Eng. (Eng. Physics)

SUPERVISOR: Dr. H. deBruin

NUMBER OF PAGES: xi, 184

ABSTRACT

Brainstem Auditory Evoked Potentials (BAEP) contain valuable information about the condition of the neural fibers associated with the auditory pathways. Extraction of this information is a difficult task due to contamination by on-going scalp EEG.

This thesis reviews the current processing techniques and introduces adaptive noise cancellation (ANC) using systolic arrays as an alternative to existing technology. Q-R decomposition theory is reviewed and an explanation of the mechanics of systolic adaptive noise cancellation (SANC) is presented. A modified Given's rotation algorithm is derived resulting in a saving of up to 2/3 in memory requirements.

Real data were collected in the laboratory. Real and simulated data were processed to determine the characteristics and effectiveness of adaptive noise cancellation strategies. Successful ANC of BAEP was

performed on simulated data using a number of signal-to-noise ratios (S/N), data sequence lengths, reference signals and filter parameter values. We conclude that systolic arrays are a very powerful and appropriate technique for the extraction of BAEPs.

Correlation studies indicated that the pre-stimulus EEG signal is inadequately correlated to the primary signal for successful ANC of BAEP in real data. A multi-channel collection scheme is outlined for future collection of Evoked Potential data.

A summary of experimental results is presented to address the problem of data collection and signal processing optimization.

ACKNOWLEDGEMENTS

For his understanding, guidance and helpful comments I wish to express my gratitude to Dr. Hubert deBruin. Without his kind assistance this and other work would not have been possible. Also, my thanks to Gordon Jasechko and Tim Nohara for being such patient subjects.

I would also like to thank McMaster University and the Natural Sciences and Engineering Research Council for their financial assistance.

I am grateful to Miss. Joëlle Cossette for all her time and effort in preparing this thesis.

Finally, to both my parents, who have supported me through all my endeavors. Thank you.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

CHAPTER 1 INTRODUCTION

The bioelectric nature of tissue was first described by Galvani in 1794 and since that time man has investigated many parts of the human body through the use of electrical signals. Electrical potentials are involved in both motor control and sensory perception. Neurons, which propagate the bioelectric signals, are present throughout the body but are most concentrated in the central nervous system (CNS). The neuronal activity which can be measured via potential recordings from the scalp is the Electroencephalograph or EEG. Studies of human EEG have been intensively conducted since Berger's report in 1929.

The EEG is used clinically to evaluate neurological conditions, study sleep patterns, etc. However, due to the non-specific, spontaneous and stochastic nature of the EEG, progress towards understanding specific correlations between the scalp potential and physiological activity has been difficult.

Stimulation of a specific sensory pathway activates an identifiable neuronal population. Using this technique it

is possible to drive a predetermined neurological centre. The resulting electrophysiolgical signals are called Evoked Potentials or EPs. For example the visual evoked potential or VEP elicited by a flash of light can be recorded by electrodes placed strategically on the scalp.

Recording and interpretation of EP data is hampered by the poor signal to noise ratio which exists due to the concurrent, spontaneous EEG Signal to noise ratios in single stimulus brainstem auditory evoked potentials (BAEPs) have been reported to be approximately -20dB (Madhavan, 1985). Our own studies concur with estimates from -15dB to -23dB (bandwidth of 100Hz to 3kHz).

Averaging, first described by Dawson in 1954, is currently used to extract the EP. Under the assumption that the EP is deterministic, that the EEG and EP are uncorrelated and that the EEG is stationary, the signal to noise ratio (power) will improve by a factor of M (M being the number of repetitions of stimulus) and thus the amplitude ratio will improve by $\sqrt{M}$. Obviously there are a number of problems with this technology not to mention the doubtful validity of some of the implicit assumptions. Average EP records do not allow identification of transient phenomena, habituation or real-time monitoring of the nervous system.

Many techniques attempting to improve EP processing have been proposed and investigated. These are more fully discussed in the next chapter. Recent advances in real-time signal processing technology have provided us with powerful new tools which may be adapted to our EP extraction problem. Adaptive noise cancellation schemes using a number of algorithms and data collection montages have been proposed (Madhavan 1984). We propose to use the very recently developed Systolic array architecture in an adaptive noise cancellation scheme to investigate the EP processing problem. Systolic arrays are discussed in detail in Chapter 3 but for now they can best be described as ultra stable, modular, multiply interconnected structures that can be designed to perform channel equalization, adaptive beamforming, adaptive noise cancellation and other signal processing tasks.

We identify two main areas of investigation in order to develop general strategies for EP processing. In Chapter 4 we focus on the collection and analysis of real signals. For purposes of this research we concentrate on the BAEP. In this area we developed the necessary experience an understanding required for successful signal acquisition. Spectral and temporal analyses of the signals allowed us to

develop models and define parameter ranges for subsequent simulations with the systolic noise canceller. Due to the very recent nature of systolic engineering it was necessary for us to develop an understanding of the performance of our noise canceller under the conditions of our problem as indicated by our real data experiments. In Chapter 5 we describe extensive experiments designed to lead us towards optimized parameters and collection schemes for eventual clinical application of systolic noise cancellation processing of evoked potentials.

CHAPTER 2

PHYSIOLOGY and EVOKED POTENTIAL PROCESSING

2.1 Physiology

The Electroencephalogram or EEG is the scalp recording of the electrical activity of the brain. It is well documented that the EEG reflects accurately the underlying Electrocorticogram potential (Nashi, 1985). The Electrocorticogram potential is the sum of spontaneous activity contributed by the entire brain. This activity is transmitted to the scalp by the flow of ionic currents and may be picked up by both needle and surface electrodes.

Instrumentation, electrode placement , size, and montage significantly affect the characteristics of the recorded signal. In Chapter 4 the impact of the choice of electrode montage with respect to our investigation is discussed more fully.

Physiological factors such as alertness and health influence the EEG. It is this correlation between recorded EEG and health factors, especially mental and neurological

conditions which has prompted the search for the hidden message in the EEG. While the EEG has provided us with a useful diagnostic tool in areas such as epilepsy, brain damage, and stroke (Childers, 1977), progress in the development of new diagnostic procedures through analysis of both time and frequency domains has been hampered by the EEG's complex, stochastic and non-specific nature.

Emphasis has shifted from the non-specific EEG to the responses evoked by stimulation of the human sensory system. The Evoked Potential or EP, in contrast to the EEG, is specific, deterministic and somewhat less complex. While it is important to identify these differences it should be noted that the EEG and EP are essentially the same neurological phenomena, the EEG being spontaneous and the EP being specific to the stimulated pathway. EP analysis may be likened in many ways to a crude perturbation technique where the effect of a single input on a system is desired to be known while the system continues to operate.

Under specific stimulation the recorded activity of the brain at the scalp consists, ideally, of two independent, additive components; the spontaneous EEG and the EP response. Of course it is conceivable (and even likely)

that stimulation of a specific pathway may affect the spontaneous EEG and thus change the correlation functions of the EEG and EP response. Under these circumstances a purely additive model will not be completely adequate.

An EP can belong to one of four classes depending upon its proximity to the stimulus:

(1)	Primary	- EP origin in the area of direct stimulation.
(2)	Secondary	- EP origin in an area close to stimulation pathway.
(3)	Non-specific	- EP area not directly related to stimulation pathway.
(4)	Contingent Potentials	- slow change or expectancy wave.

Greater distances, either temporally or spatially, between the response origin and the recording site allow more cerebral processing of the EP thus decreasing the specificity of the recording. In order to minimize the effect of high level processing we chose , for our investigations, the Brainstem Auditory Evoked Potential. The early waves of the BAEP have a very short temporal latency with respect to the stimulus thus providing us with an EP of very specific anatomical origin and relatively well understood physiological significance. The BAEP also exhibits the following characteristics:

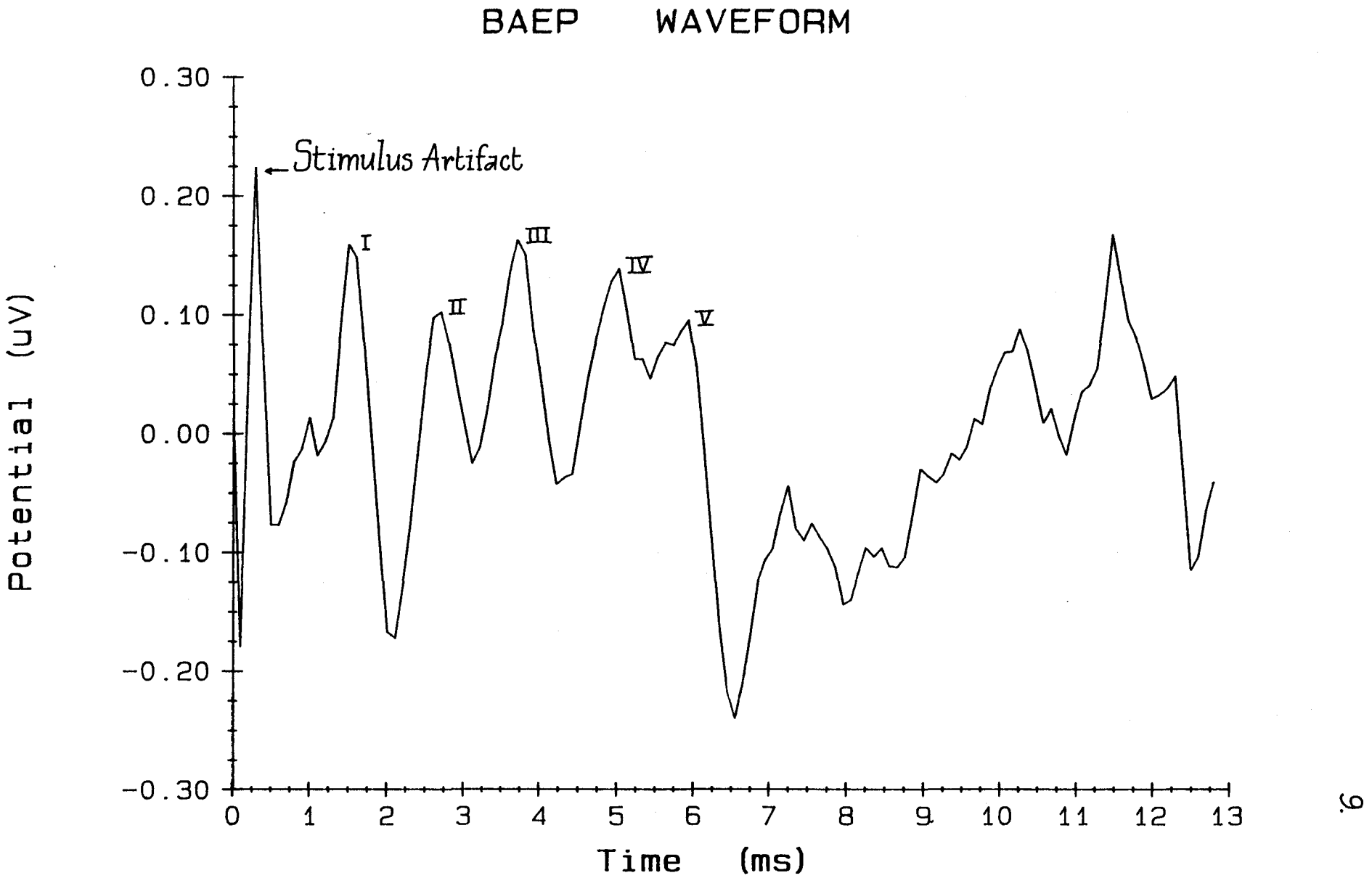
- affected minimally by environmental factors
- exhibits little habituation
- recognizable and consistent shape
- well used clinically and well documented

The BAEP is therefore well suited to our investigation. One such recorded EP is illustrated in figure 2.1.

The EEG is the temporal and spatial summation of the entire neuronal population of the brain whereas the EP represents only a small, specific population. Consequently the signal to noise ratio (the EP is the signal and the EEG is the noise) is almost always less than 0 dB. In the case of the BAEP we have estimated that the S/N ratio can be as poor as -25dB. For visual examination the EPs some form of signal processing must be done to improve the S/N. In section 2.2 we discuss the spectrum of current techniques used in EP processing.

The BAEP is commonly used as one component of a neurological examination. Other EPs used include the Visual Evoked Potential (VEP), the Brainstem Somatosensory EP (BSSEP) and the Somatosensory Cortical EP (SSCEP). Each

FIG. 2.1


BAEP WAVEFORM
Stimulus Artifact
I
II
III
IV
V
0.30
0.20
0.10
0.00
-0.10
-0.20
-0.30
Potential (uV)
0
1
2
3
4
5
6
7
8
9
10
11
12
13
Time (ms)

EP has a different waveform, S/N, spectral density, recording montage and information content. For example the BAEP is a 5-7 peak response with a mean frequency of about 1kHz, appearing within 10 ms after the stimulus. It is used primarily to evaluate the integrity of the auditory pathways.

The short response latency of the BAEP rules out any possible contribution from the cerebral cortex. It is therefore thought that the BAEP is a volume conducted signal and is uncorrelated with the spontaneous EEG. Due to the small amplitude of the BAEP (<0.5 uV) as many as 2000 individual responses must be averaged to assemble a signal which can be visually inspected as in figure 2.1. The latency of the first five peaks is measured and compared to the normal distribution of latencies. The presence of peaks VI & VII is less reliable so little use is made of them clinically. Amplitude measurements are sometimes made but large variation in normals reduces the usefulness of such recordings. Absence or abnormally long latencies of one or more waves may indicate neurological dysfunction.

The accepted anatomical origins and normal latencies of peaks I-V are documented in Table 2.1. We also present the

suspected origins of peaks (VI & VII). Work continues to better understand the precise anatomical relationships involved. Bhuwan and colleagues have reported evidence in a recent study suggesting that peak II may originate from the extramedullary portion of the VIIIth nerve rather than from the medulla (Bhuwan et al.,1982).

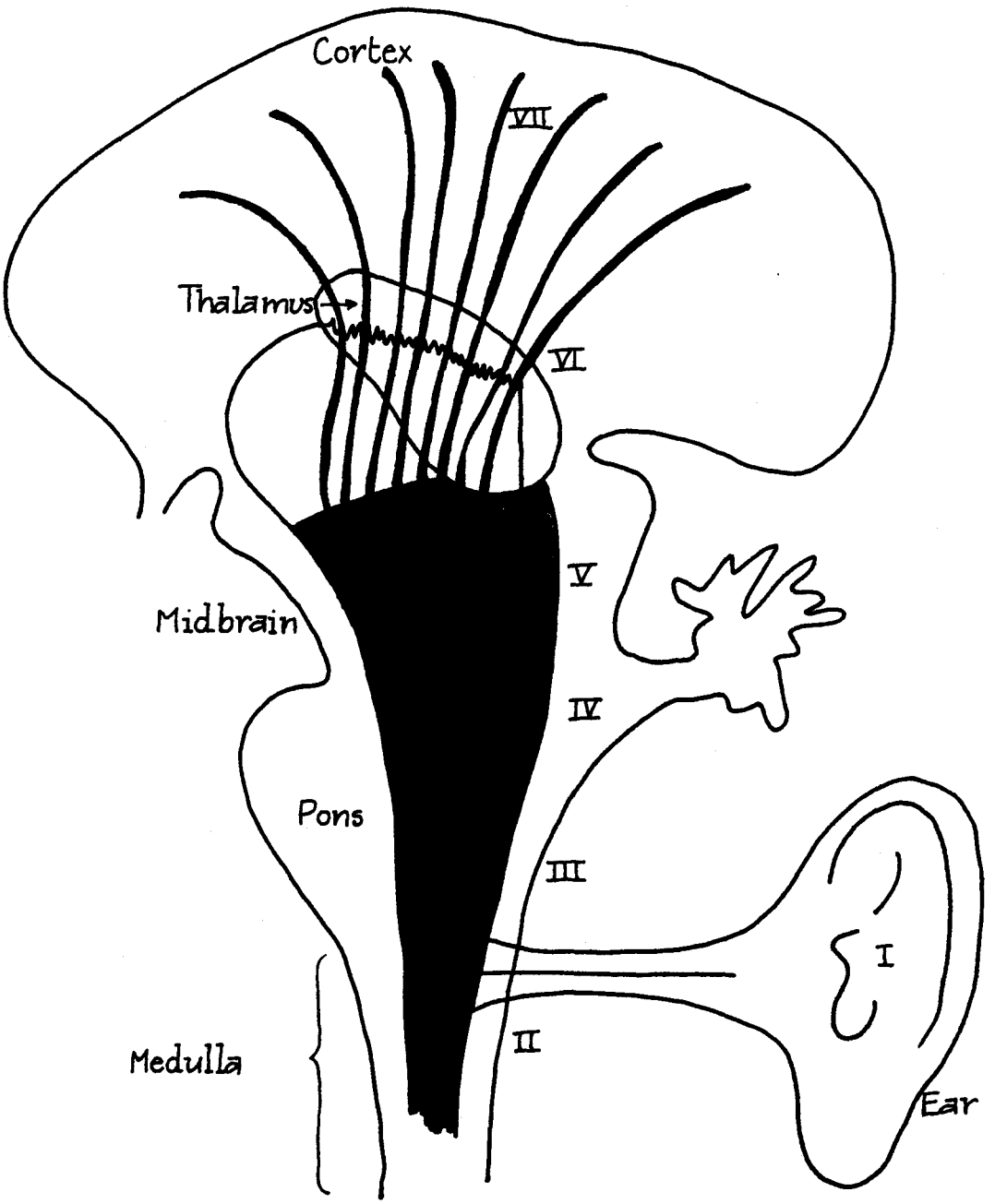
Table 2.1 Latency and Origin of the BAEP Peaks

WAVE	LATENCY (ms)	ORIGIN	%Identified (10/s)
I	1.6-1.7	Eighth Nerve	97
II	2.8	Cochlea Nucleus (Medulla)	96
III	3.8-3.9	Inferior Pons (Superior Olivary Complex)	100
IV	4.8-5.1	Upper Pons	88
V	5.5-5.7	Midbrain	100
VI	7.1	Thalamus ?	
VII	9.1	Thalamo cortical ?	

from Kiloh et al.,1983

The brainstem consists of four regions; the medulla oblongata, the pons, the midbrain and the diencephalon. It is responsible for respiratory, cardiovascular, and gastrointestinal function as well as many of the sterotyped movements of the human body. The auditory pathway begins

fig2.2 Anatomy of the brainstem



neurologically at the organ of Corti located in the inner ear. Nerve impulses are transmitted along the acoustic nerve to the processing centres of the medulla, pons, midbrain, and thalamus as illustrated in figure 2.2. The reader is referred to Webster (1978) for a review of human physiology and anatomy as they relate to EEG and evoked potentials.

Due to its remarkable consistency over time the BAEP has been used in the diagnosis of acoustic neuromas, sub clinical demylination, brainstem gliomas, vascular lesions, and brain death (Kiloh et al., 1981). The monitoring of comatose patients or patients under CNS depressive drug treatment is also possible due to the BAEP invariance during such conditions. Other non-affecting drugs include pentobarbital, ketanine, and halothane (Kiloh et al.). For a detailed description of the clinical uses and potential uses of the BAEP the reader is referred to Chiappa 1983 (Chapter 3).

Exploitation of the EP has been limited by the poor S/N ratio exhibited by virtually all EPs. The use of averaging, while being robust, necessitates repeated stimulation thus obscuring transient phenomena, producing habituation effects, eliminating the possibility of real-

time analysis and generally making the EP recording and analysis inconvenient, tedious and unwieldy.

Potential real-time uses such as intraoperative monitoring, have been difficult to accomplish due to the delay (2-3 minutes in the case of a brain stem) required for averaging. In order to alleviate this problem many research programs have been directed at improving signal processing techniques in the area of clinical neurology. It is these studies to which we now draw our attention.

2.2 Evoked Potential Model

The purpose of evoked potential processing is to obtain an estimate of the buried signal. Dawson used photographic superposition as a first attempt to extract a response (Dawson, 1947). Since that time many new techniques have been investigated. Digital averaging is now the standard approach used to extract EPs in the clinical setting. This involves the arithmetic summation of a large number of digitized, single EP responses.

Not only does digital signal processing provide a convenient and flexible environment for signal acquisition

but it also allows us to bring many powerful analytical tools to bear on the extraction and enhancement problem. These tools include; correlation analysis, spectral estimation, signal averaging, pattern recognition, zero phase filtering, time-series modelling, and adaptive filtering algorithms. All of these techniques have been, and are being used, in an attempt to reduce the number of repetitions required for the extraction of the EP.

The EEG and EP combination can be represented as:

$$\phi_j(i) = \xi_j(i) + \upsilon_j(i) \qquad j=1,2,..N \qquad (2.1)$$

υ .... EEG

ξ .... EP

subscript j stimulus repetition number

i digitally sampled point in record

The response consists of two additive parts which may be characterized by:

EEG: stationary, random process

EP : uncorrelated to the EEG with a probability distribution defining shape, latency from stimulus, and amplitude

The strictly additive or uncorrelated nature of the EP is a universally applied assumption based on the physiological premise that the stimulated neural centre represents an insignificant fraction of the total CNS population. The stationarity of the EEG is less well accepted but has been shown to be valid for short epochs (<12s) (Cohen & Sances, 1977). These statements will be explored more as the various processing techniques are introduced.

2.3 Current EP Processing Techniques

Since Dawson's time a large array of EP processing and extraction techniques have been presented including Woody's correlation method, Weiner filtering, latency corrected averaging, PSR, Maximum likelihood formulations and adaptive noise cancellation (ANC) schemes. Simple averaging, however, still remains the clinically accepted method.

In addition to our model assumptions we require some more restrictive assumptions about the nature of the EEG-EP complex when averaging is used:

EEG (noise) must be uncorrelated from trial to trial.

EP must be deterministic with respect to shape, latency and amplitude.

When these assumptions are perfectly valid the signal to noise ratio (power) will improve by a factor of N where N is the number of stimulus repetitions. The amplitude ratio can then be shown to improve by a factor of $\sqrt{N}$ (Aunon et al., 1981).

We can express the process of averaging via:

$$E[\phi(i)] = E[\xi(i)] + E[\upsilon(i)] \quad (2.2)$$

$$= \xi(i) \quad (2.3)$$

because $E[\upsilon(i)] = 0 \quad (2.4)$

$$E[\xi(i)] = \xi(i) \quad (2.5)$$

and $$E[\upsilon_k(i)\upsilon_l(i)] = \begin{cases} 0 & \text{for } k \neq l \\ \sigma^2 & \text{for } k = l \end{cases} \quad (2.6)$$

In practice it is likely that the EP assumptions are not strictly met. While the EEG seems to be uncorrelated from trial to trial, under certain circumstances the EEG has been shown to exhibit non-stationary characteristics (Cohen & Sances, 1977). However averaging is still the main clinical tool due to its robustness and its conceptual and computational simplicity.

Thus ensemble averaging is a sub-optimal solution due to its unrealistic assumptions and produces the following undesirable effects:

- Inaccurate representation of a single stimulus (SSt) EP
- Long acquisition time (non-real time)
- Habituation of physiological response
- Loss of transient information

In order to address these problems a number of processing techniques have been proposed.

Weiner Filtering was first introduced to the EEG processing field by Walter in 1969. Walter began with the steady-state Weiner filter which will optimally estimate the SSt EP, $\xi(i)$, in the least mean square error sense.

$$H(\omega) = \frac{S_{\xi}(\omega)}{S_{\xi}(\omega) + S_{\upsilon}(\omega)} \qquad (2.7)$$

where: $S_{\xi}(\omega)$ is the power spectrum (Fourier transform of the autocorrelation function) of the SSt EP.

$S_{\upsilon}(\omega)$ is the power spectrum of the noise (EEG).

Since these power spectra are unavailable Walter used the average spectrum of the responses ($\bar{S}_\xi$) and the ensemble averaged spectrum ($S_{\bar{\xi}}$) to estimate the transfer function:

$$\tilde{H}(\omega) = \frac{\frac{N}{N-1} S_{\bar{\xi}} - \frac{1}{N-1} \bar{S}_\xi}{\bar{S}_\xi} \qquad (2.8)$$

Single responses are used with the filter transfer function to produce the signal estimate. An alternate form of equation 2.8 has been developed by Doyle (1977) for use with ensemble averages as inputs. In addition to the assumptions required for averaging, Weiner filtering requires a response of long duration to ensure stationarity of the signal and noise. This requirement was circumvented by deWeerd's 'Time-Varying Filter' design (deWeerd & Kap, 1981). Essentially this consists of estimating the time-varying signal to noise ratio in pre-determined frequency bands. The Weiner filter estimate is then appropriately attenuated.

Results indicate that the performance of time-varying filters is superior to that of time-invariant filters (Yu & McGillem, 1983). Despite the availability of software to perform time-varying filtering, the increased complexity

and inconclusive evidence of improved performance has largely kept this technology out of the clinical setting.

Cross-Correlation techniques designed to accommodate jitter in the EP were introduced by Woody in 1967. A template is used to develop crosscorrelation curves with the individual responses. The alignment of signals is assumed to be at the maximum of the crosscorrelation curve.

Fundamental to this procedure is the assumption that the stimulus to EP latency is variable. Thus, by re-aligning the individual responses a more accurate average EP may be developed. For an accurate correlation there must be a relatively large number of points in each trial. Also the EEG must not contain any components whose shape is similar to those in the EP.

Implicit in this method is the need for a template. The choice of this template has a significant effect upon the success of the extraction. Previously a number of responses (~100 or more) were averaged to develop a template. Poor signal-to-noise ratios and noise components similar to the EP produce significant error in the EP estimate.

Improvements to the cross-correlation technique have been proposed by Aunon & McGillem in 1975. **Latency Corrected Averaging** uses a minimum mean squared error (MMSE) criterion and breaks the EP into components. Thus the assumption of EP shape invariance is replaced by an EP component shape invariance assumption. Significant improvements in results were reported (McGillem et al., 1985). The problem of defining unique EP shapes still remains.

One of the more recent techniques is the **PSR** or predictor, subtracter, restorer filter reported by Kaveh (1978). The PSR operates in a similar fashion to noise cancellation structures. It is based on the ability to model the EEG as an autoregressive time-series and uses the correlation between pre and post stimulus EEG.

The AR model is first developed from the pre-stimulus EEG. These parameters are then used as a prediction filter operating on the post stimulus EEG (EEG & EP). Assuming the pre and post EEG are the same the result is a distorted EP which is then passed through the inverse AR filter to restore the EP. Studies indicate that the PSR performs better than Weiner filtering but has problems associated with the assumption of stationarity.

The **Maximum Likelihood Estimation (MLE)** method also uses ARMA modelling of the EEG. Using this technique Nashi has shown that it is possible to remove the assumptions of constant EP latency, shape and amplitude (Nashi et al., 1987). The method begins with the average of an ensemble of evoked EEG records. A representative shape and amplitude are then obtained for the EP which is then filtered using the characteristic function of the probability density of the post-stimulus latency. State-space equations are obtained from ARMA models of the EP and the best estimate of the EP is obtained by maximization of the Likelihood function.

While this technique has been shown to reduce the number of stimulus repetitions required to produce a recognizable EP, its usefulness is limited by the necessity to begin the process by ensemble averaging.

All of these methods, under certain conditions, improve the S/N of the ensemble or reduce the number of stimuli required for an acceptable EP estimate. However none is successful enough to extract a single EP (especially in the case of BAEP).

Adaptive noise cancellation (ANC) enhancement schemes have been proposed by Madhavan et al. (1984) for use in EP extraction. These schemes involve the use of a reference input to cancel the EEG from the EEG + EP leaving us with T, the single stimulus estimate. Figure 2.3 illustrates the adaptive noise cancellation scheme.

The ANC scheme can be represented mathematically by the primary input:

$$\phi = \xi + \upsilon \tag{2.9}$$

where: ξ is the true EP

υ is the additive EEG noise

ϕ is the single evoked response

and: υ_r is the correlated noise or reference signal

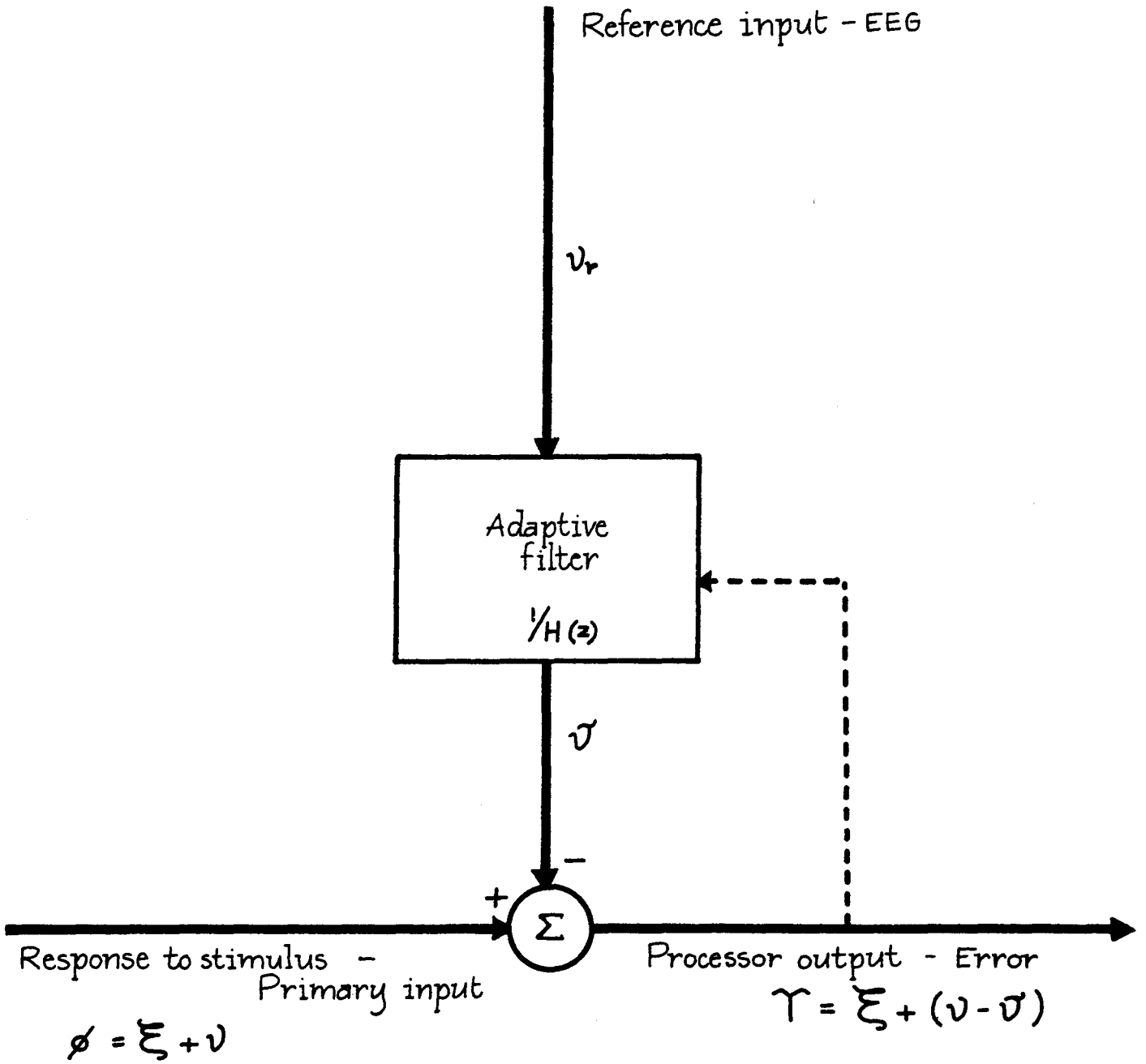
By using ANC schemes we can obtain an estimate of the true EP, T from:

$$T = \xi + (\upsilon - \tilde{\upsilon}) \tag{2.10}$$

where:

$\tilde{\upsilon}$ is the ANC estimate of υ using υ_r as a reference.

fig. 2.3 Adaptive Noise Cancellation Scheme



The use of the LMS algorithm in a parallel channel configuration to extract VEP has been reported but the authors claim to have had little success (Winski & Allison, 1984). This may have been partly a result of the LMS algorithm's poor convergence properties. Madhavan et al. (1984) have employed Kaveh's (1978) idea on correlation using the pre-stimulus EEG as the reference input to the ANC scheme. Their investigation of the BAEP using the least-squares lattice (LSL) algorithm shows a great deal of promise for the ANC technique as applied to the EP extraction problem.

The success of adaptive schemes in noise cancellation warrants further investigation and verification. We propose the use of Systolic Arrays in the adaptive noise cancellation of evoked potentials. These new systolic processing techniques promise to be the most powerful signal processing architectures yet developed. Systolic arrays are ultra stable, modular structures ideally suited to VLSI implementation. Plug-in high speed boards are rapidly becoming available.

In the case of adaptive filtering, systolic arrays are used to implement RLS-QR decomposition via Given's rotations. Due to the very recent nature of systolic technology the

performance of systolic noise cancellation (SNC) schemes is not well documented. The goal of this research was to design noise cancellation experiments that will allow us to understand the characteristics of SNC within the boundaries of EP extraction. We hope to develop the body of knowledge necessary for further investigation into optimized extraction of single stimulus estimated evoked potentials.

The specific areas of interest are:

- the effects of non-stationarity
- the effects of S/N
- data sample size
- choice of reference signal
- comparison of systolic and lattice performance
- the effect of array order

The potential benefits from such a single stimulus extraction technology are enormous. By eliminating problems such as habituation and allowing the recording of transient effects, we can apply the EP to a large number of monitoring and diagnostic problems using a pseudo real-time processing scheme. Intraoperative monitoring of optic chiasm surgery, and recording of signals such as the olfactory evoked potential would then be made possible.

CHAPTER 3

SYSTOLIC ARRAYS

3.1 Introduction

We now review the signal processing theory associated with systolic arrays and adaptive noise cancellation. The following discussion is based on Haykin's book, "Adaptive Filter Theory" (1986) and other cited papers. The reader is invited to refer to these sources for a more detailed presentation of the material.

3.2 Adaptive Processing Historical Development

Adaptive noise cancellation is one of the applications of the filter theory which developed from Karl Friedrich Gauss. Least square estimation was used by Gauss in 1795 to estimate the six parameters associated with the motion of the heavenly bodies. The first studies of minimum mean-squared estimation in stochastic processes were made by Kolmgorov, Krein and Weiner during the late 1930's. Weiner formulated the continuous time linear prediction problem

and, in conjunction with Hopf, derived the integral equation for optimal parameter estimation. In 1947 Levinson formulated the discrete time counter part of the Weiner-Hopf equation resulting in the matrix normal equation:

$$\mathbf{R} \cdot \mathbf{w}_o = \mathbf{p} \tag{3.1}$$

Where $\mathbf{w}_o$ is the optimum tap-weight vector, $\mathbf{R}$ is the correlation matrix of the tap inputs and $\mathbf{p}$ is the cross-correlation vector between the tap inputs and the desired response. A recursive solution to the normal equation was developed by Levinson and rediscovered by Durbin in 1960 for use in modelling time-series data.

The beginnings of adaptive filtering can be traced to the 1950's when Widrow and Hopf developed the least mean squared (LMS) algorithm, a stochastic gradient algorithm. The LMS algorithm iterates each tap-weight in the transversal filter in the direction of the gradient of the squared amplitude of an error signal.

A second class of adaptive filter using state-space formulations (based on Kalman filter theory) was developed by Godard in 1974. The Kalman algorithm had superior

convergence properties when compared to the LMS algorithm.

The third class of filter, closely related to the Kalman filter, is based upon the method of least squares. The Recursive Least Squares (RLS) algorithm, which is superior to the LMS algorithm, is also computationally more complex. This complexity led to the development of faster, more efficient techniques which will be described later.

In 1978 Kung and Leirson introduced the concept of systolic arrays but it was not until 1981 that Gentleman and Kung developed an efficient algorithm for solving the least-squares problem. This algorithm involves a recursive implementation of QR decomposition. The QR decomposition is achieved by applying a series of Givens rotations directly to the data matrix, resulting in an upper triangular orthogonalized matrix. Implementation of this algorithm was done using systolic arrays.

Systolic arrays provide the modular structure necessary for efficient computation. In 1983 McWhirter introduced a modification to the algorithm by eliminating explicit computation of the tap-weight vector. As we will see later in the chapter, successful adaptive noise cancellation does not require explicit knowledge of the tap-weight-vector.

It is this final algorithm, with some minor modifications, which was implemented in our study.

3.3 Least Squares

Two of the filters just described (LMS and Kalman) are based on a statistical approach. The recursive least squares method is, in contrast, deterministic. Since Gauss was the first to develop least squares estimation theory it is probably best to look to him for an explanation. In Sorenson's 1970 paper the basic ingredients of least squares as described by Gauss are explained. We focus on three of these as the essential ingredients.

1) More observations must be taken than the absolute minimum required by the system due to the inherent measurement errors.

2) Parameter estimates must satisfy the observed data in such a way that the difference between the observed values and the values predicted from the estimated parameters are as small as possible.

3) Gauss also referred to the suitable combination of the observations that will produce the most accurate estimates. This alludes to the structure of the least-squares solution (e.g. linear or non-linear filters) and to the choice of performance criterion.

In order to progress towards the implementation of systolic arrays in adaptive noise cancellation of EEG we begin with the introduction of the autoregessive model:

$$y(k) = \sum_{i=1}^{M} a_i y(k-i) + n(k) \qquad (3.2)$$

$$0 = \sum_{i=0}^{M} \alpha_i y(k-i) - n(k) \qquad \alpha_o = 1 \quad \alpha_i = -a_i \qquad (3.3)$$

where: M is the order of the filter and
n(k) is a Gaussian white noise sequence.

We see that the resulting sequence is a weighted sum of past outputs plus a noise term (n(k)). We wish to estimate the parameters a_i (autoregressive coefficients). By feeding a white noise sequence into a filter and minimizing the residual between this noise sequence and the autoregressive process, the coefficients may be determined.

We begin by considering a stochastic process characterized

by {d(i) & u(i), i=1,2,.., N}. Where d(i) is observed at time i as the response to inputs u(i). Thus we have two sets of data. We now use the transversal filter in figure 3.1 to illustrate the derivation of the deterministic normal equation.

To state the linear least squares problem we begin by relating the inputs u(i), the desired response d(i), the parameters of the model w_k and the residual or estimation error e(i) by:

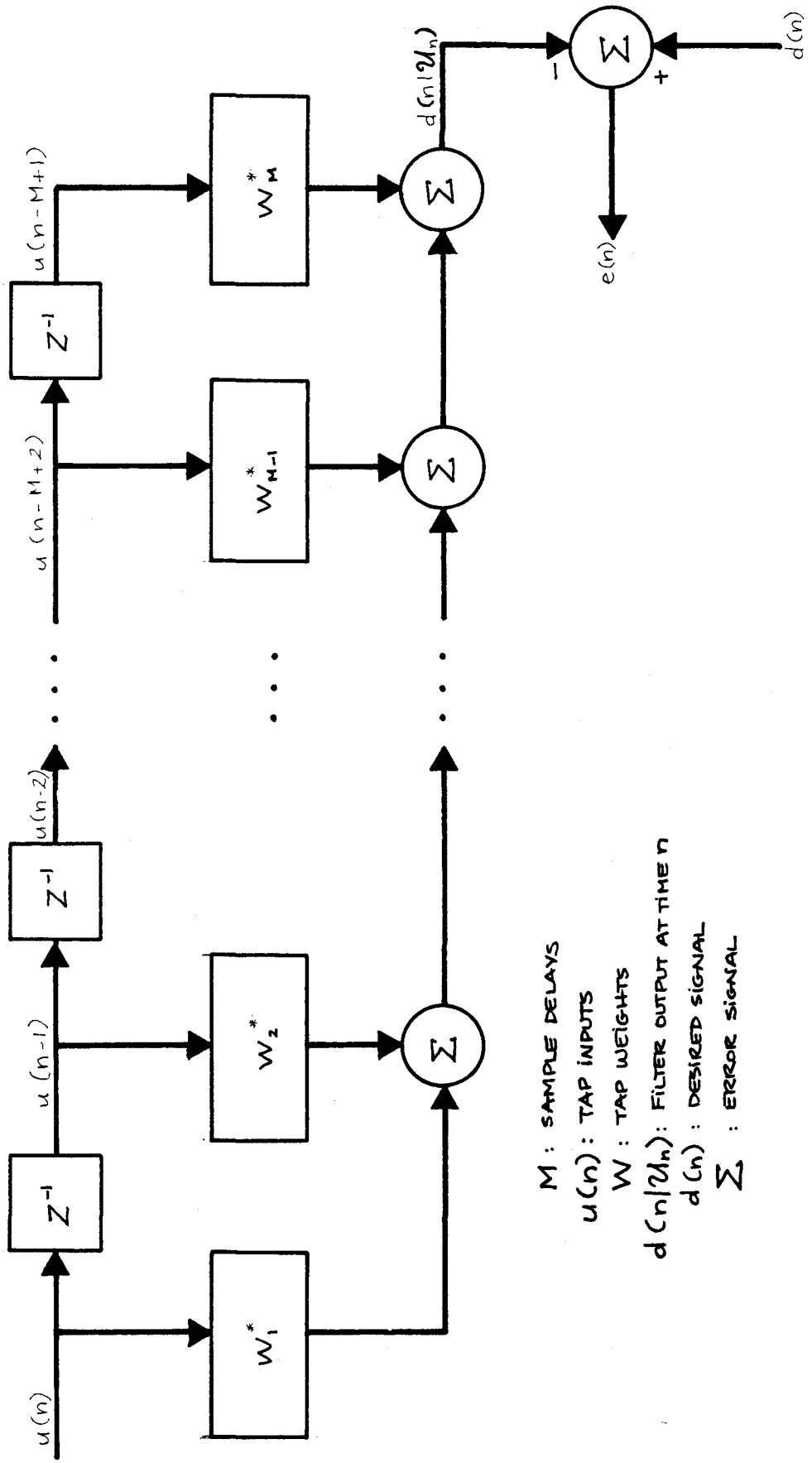
$$e(i) = d(i) - \sum_{k=1}^{M} w_k^* u(i-k+1) \qquad (3.4)$$

In the method of least squares we choose the tap-weights of the transversal filter so as to minimize an index of performance that consists of the sum of error squares:

$$\xi(w_1, w_2, .., w_M) = \sum_{i=i_1}^{i_2} |e(i)|^2 \qquad (3.5)$$

Where M is the number of tap-weights in the filter, i_1 and i_2 define the index limits between which the error minimization occurs. The values assigned to the limits depend upon the type of data windowing employed. By expressing the minimization function in terms of the data matrix ('A'), the desired output vector ('b') and the tap-

fig. 3.1 Transversal Filter
(HAYKIN, 1986)



weight vector ('w'), then differentiating with respect to the tap-weight vector it is relatively straight forward to arrive at the so-called deterministic normal equation (Haykin, 1986):

$$A^H A\hat{w} = A^H b \qquad (3.6)$$

Solution of this equation (as in the Yule-Walker spectral estimation algorithm) can be done by LU decomposition or, preferably, Singular Value Decomposition (to find the psuedo-inverse). However, in terms of adaptive processing eqn. 3.6 is unsuitable because of its non-recursive , numerically unstable and computationally costly nature. We now look at various recursive algorithms for solving the least-squares problem.

3.4 Recursive Solutions to the Least Squares Problem

The choice of algorithm for the solution to the least squares problem depends upon the documented characteristics such as stability, speed, and complexity. One may also wish to consider compatibility with existing system models. If the phenomena being investigated can be modelled by a modular lattice structure one may wish to implement a

lattice type filter in the hope of gaining new insight into the nature of the phenomena. Very often a compromise between desirable characteristics (e.g. speed vs complexity) may have to be struck.

3.4.1 Recursive Least Squares (RLS)

The earliest and conceptually simplest least squares algorithm is the recursive least squares (RLS) which is developed from a transversal filter structure. Central to the functioning of the RLS algorithm is an algebraic result known as the matrix inversion lemma.

Given: $$\mathbf{A} = \mathbf{B}^{-1} + \mathbf{C}\mathbf{D}^{-1}\mathbf{C}^{H} \tag{3.7}$$

A,B are positive definite MxM matrices

D is a positive definite NxN matrix

C is an MxN matrix

We have: $$\mathbf{A}^{-1} = \mathbf{B} - \mathbf{B}\mathbf{C}(\mathbf{D}+\mathbf{C}^{H}\mathbf{B}\mathbf{C})^{-1}\mathbf{C}^{H}\mathbf{B} \tag{3.8}$$

From these equations it is easy to see that the RLS algorithm is computationally costly. In order to increase

speed, related algorithms using properties of serialized data such as the fast RLS and fast transversal filter (FTF), have been developed. The reader is directed to Haykin (1986) Chapter 8 for details.

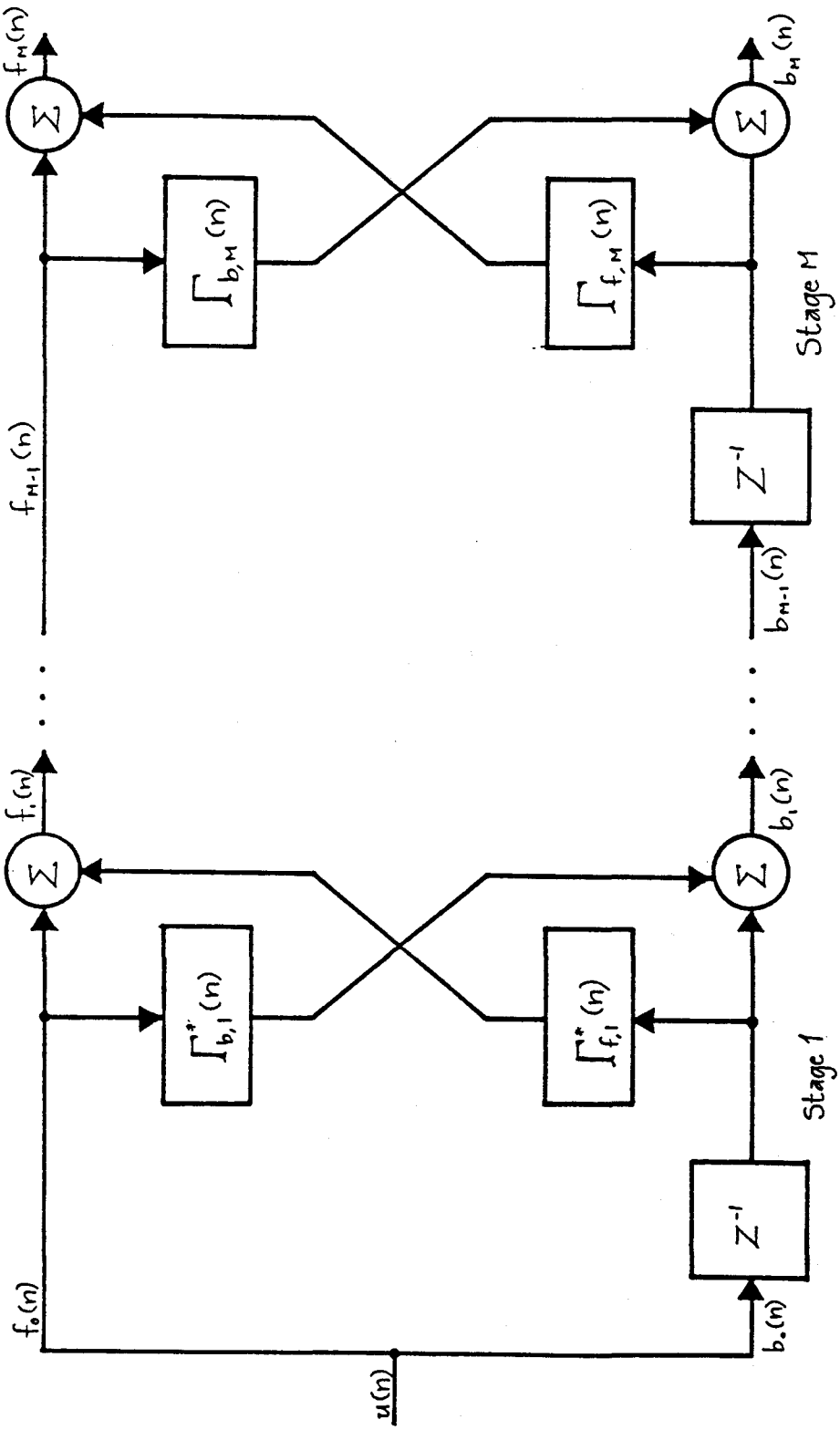
3.4.2 Least Squares Lattice

Until now all the algorithms discussed i.e. LMS, Kalman, RLS, fast RLS, FTF, have been based on a transversal filter structure. We now look at the so called least squares lattice algorithm, which is a multistage lattice predictor. It will be useful to discuss this algorithm in greater detail since it is used in experiments to compare the performance of the lattice with the systolic implementation.

The exact LSL algorithm (so called because it provides a exact solution to the least squares problem at each stage) involves both time and updates and produces a set of so called reflection coefficients. The lattice structure is illustrated in figure 3.2. The tap-weights are related to reflection coefficients. We are therefore no longer presented with the instantaneous values of the transversal filter coefficients. The reflection coefficients are

fig 3.2 Lattice Filter

(HAYKIN, 1986)



f : FORWARD A POSTERIORI PREDICTION ERROR

b : BACKWARD A POSTERIORI PREDICTION ERROR

Γ : REFLECTION COEFFICIENT

orthogonal to each other, a result of the decoupled nature of each lattice stage. A modification of the LSL algorithm, the normalized LSL, ensures that the value of the reflection coefficients lies between -1 and +1. In Appendix I a computer program is presented for performing adaptive noise cancellation using the Normalized Least Squares Lattice algorithm.

3.4.3 QR Decomposition

QR decomposition is a technique for finding the optimum weight vector in the least-squares estimation problem. 'Q' is a unitary matrix which operates directly on the data matrix 'A', transforming it into an upper triangular matrix 'R'. In this form back substitution can be performed to determine the optimum tap-weight vector 'w'. The estimation error can then be computed from 'w'. A more detailed look at QR decomposition is presented in section 3.6.3.

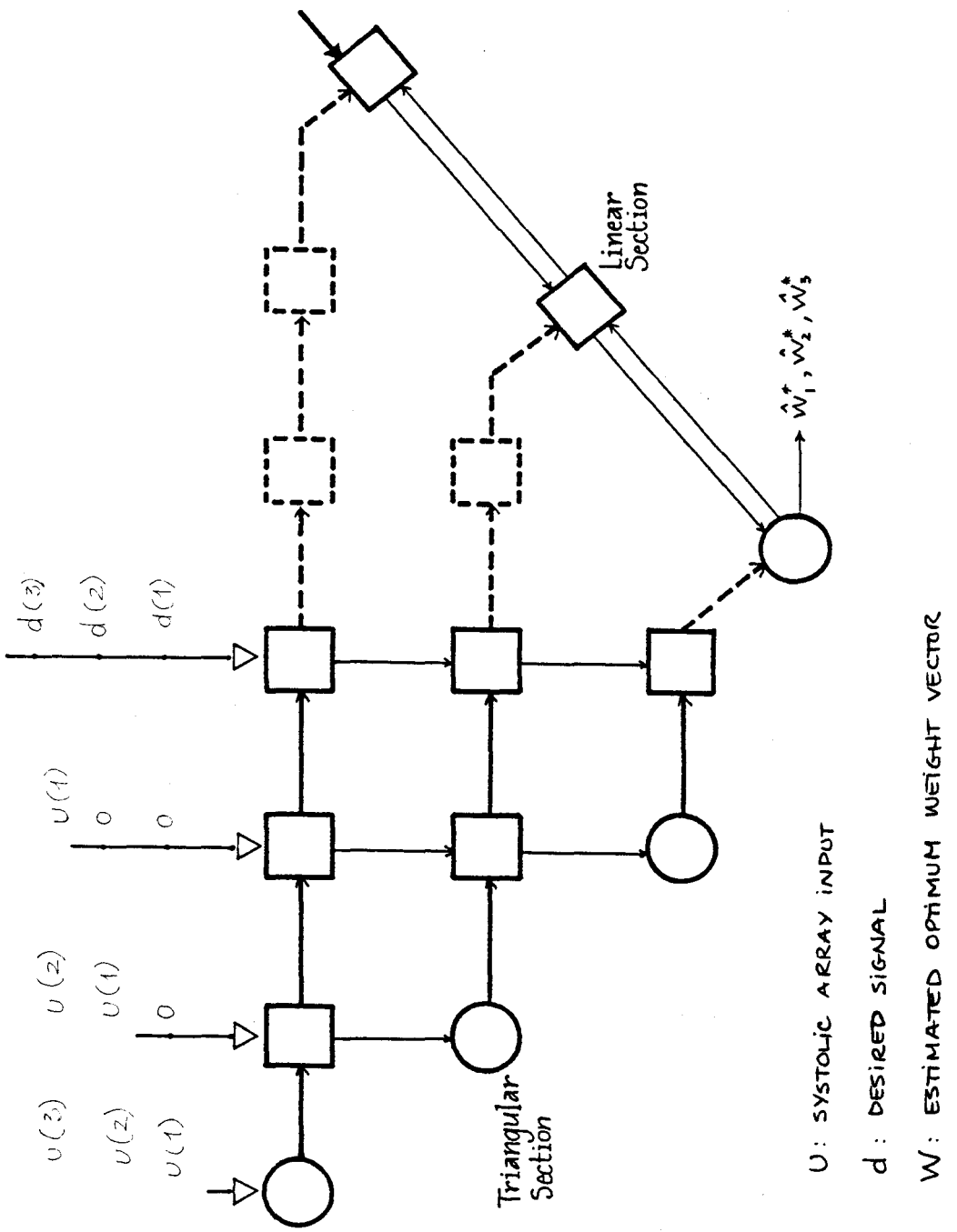
The QR algorithm is implemented in the final section of this chapter using systolic arrays. In contrast to all other adaptive algorithms, systolic arrays process the data iteratively in an open-loop configuration. In its original

form the algorithm consists of two stages. The first stage, a triangular section, involves an orthogonal triangularization process achieved by a special form of QR decomposition using Givens rotation. Other orthogonalization methods such as the Gram-Schmit used in the lattice configuration have been used in a systolic architecture. Triangularization is performed directly upon the data matrix (avoiding any calculations involving auto and cross-correlation matrices) in a recursive fashion. The second stage is a linear section used to perform backward substitution in order to extract the tap-weight vector. Figure 3.3 illustrates the 2 stage systolic adaptive filter.

For purposes of ANC it is not necessary to calculate the tap-weight vector explicitly (see next section). A second implementation of systolic arrays (Systolic II) achieved by McWhirter (1983) avoids this explicit calculation (see section 3.6). This second implementation is simpler to program, computationally more efficient and guarantees numerical stability.

fig 3.3 Systolic Array I

(HAYKIN, 1986)



3.5 Adaptive Noise Cancellation (ANC)

Prior to the development and implementation of our systolic adaptive noise canceller (SANC) it is necessary to briefly consider the requirements of a noise cancellation system.

ANC refers to the estimation of a signal corrupted by additive noise. To perform ANC we require a primary input which is composed of both noise and signal and a reference input which is correlated, in some way, to the noise in the primary signal (Widrow et al., 1975). The reference input is adaptively filtered and subtracted from the primary input to obtain a signal estimate.

In contrast to conventional filters which require apriori knowledge of the system in order to operate, an adaptive filter changes its parameters in response to some minimum error criterion. We are thus able to deal with a completely unknown system.

The schematics of the ANC process has been illustrated in figure 2.3. A signal 's' is transmitted in some channel which adds an uncorrelated noise n_p. We define the entire primary signal as :

$$z = s + n_p \qquad (3.9)$$

A second channel (reference) carries a noise signal n_R which is also uncorrelated to 's' but is correlated to n_p. The purpose of the ANC is to determine a set of coefficients which will transform n_R into an approximation of n_p i.e. $\hat{n}_p$. The additive nature of the noise and the correlation relationships outlined above are all that is required for ANC. At the output of the filter we find:

$$e = s + (n_p - \hat{n}_p) \tag{3.10}$$

where: $(n_p - \hat{n}_p)$ is small and therefore

$$e \cong s \tag{3.11}$$

Notice that it is not necessary to know, explicitly, the value of the tap-weights determined by the system. We therefore use McWhirter's modified systolic array (Systolic Implementation II) since only the error sequence output by the systolic array is required.

3.6 QR Decomposition via Givens Rotation Using Systolic Array II

We now proceed to describe the theory and implementation of the SANC.

3.6.1 Systolic Arrays In Signal Processing

Systolic arrays have found a wide variety of uses in modern real-time signal processing problems. Tasks which can be reduced to common sets of basic operations are well suited to the modular repetitive structure of the systolic array. The algebraic operations performed by systolic arrays include:

- Deconvolution, DFT & FFT
- Matrix and vector multiplications
- Adaptive beamforming
- Singular Value Decomposition (SVD)
- Adaptive Spectral Estimation (via SVD)
- Generalized SVD
- Eigenvalue decomposition
- Matrix Triangularization

from Speiser & Whitehouse, 1983

Obviously there is a great deal of interest in the applications and properties of systolic arrays. Kung presents a comprehensive list of such properties in his 1981 paper. For application techniques and VLSI considerations the reader is invited to read H.T. Kung's "Why Systolic Architectures?" (1982).

3.6.2. Givens Rotation and Triangularization

By matrix triangularization we mean the computation of **M** and **R** in the equation:

$$\mathbf{M} \cdot X = \begin{bmatrix} R \\ O \end{bmatrix} \tag{3.12}$$

If we restrict **M** to be an unitary matrix **Q**, triangularization becomes the key step in solving the least squares problem. We begin with a partially triangularized matrix as shown below.

```
x x x x                    x x x x
  x x x   ----->             x x x
    x x                        x x
x x x x                        x
```

Triangularization can be achieved by replacing the fourth row by some linear combination of 1st, 2nd and 3rd rows. The result is an upper triangular matrix shown on the right side. Obviously this is a repetitive process and is well suited to systolic implementation. The triangularization of the data matrix in the least-squares problem resulting in the desired QR configuration is achieved via Givens rotation. Givens' original work was published in 1958. In the context of QR decomposition and SANC we have a data matrix (nxM)of the following form:

$$\mathbf{A} = \begin{array}{c} \overbrace{\qquad\qquad}^{M} \\ \left.\begin{array}{ccccc} \backslash & \times & \times & \times & \times \\ o & \backslash & \times & \times & \times \\ o & o & \backslash & \times & \times \\ o & o & o & \backslash & \times \\ o & o & o & o & \backslash \end{array}\right\} M \\ \left.\begin{array}{c} \mathbf{0's} \end{array}\right\} n-M-1 \\ \left.\begin{array}{ccccc} \times & \times & \times & \times & \times \end{array}\right\} 1 \end{array}$$

M is the number of tap-weights and n is the data sample index. We wish to annihilate the m^{th} element in row n (last row) by rotation with row m of matrix **A**, m=1,2...M. The Givens matrix **G**, is defined as an nxn unitary matrix with elements g_{ik}:

$$g_{ik} = \begin{cases} \cos\phi, & i,k=m \\ \sin\phi e^{j\theta}, & i=m,k=n \\ -\sin\phi e^{-j\theta}, & i=n,k=m \\ \cos\phi & i,k=n \\ 1, & i,k=1,2,..,n-1,\text{excluding } m \\ 0, & \text{otherwise} \end{cases} \qquad (3.13)$$

By way of example we choose M=3, n=5, m=2 where **G** premultiplies the data matrix **A**:

$$\mathbf{G_2} \times \mathbf{A}$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & \cos\phi & 0 & 0 & \sin\phi e^{j\theta} \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & -\sin\phi e^{-j\theta} & 0 & 0 & \cos\phi \end{bmatrix} \times \begin{bmatrix} \times & \times & \times \\ 0 & y_{mm} & \times \\ 0 & 0^{*} & \times \\ 0 & 0 & 0 \\ 0 & y_{nm} & \times \end{bmatrix}$$

* has just been rotated, y_{nm} is the target for annihilation and y_{mm} is the rotation victim.

To eliminate the y_{nm} we solve the following simultaneous equations:

$$\cos\phi y_{mm} + \sin\phi e^{j\theta} y_{nm} = y'_{mm} \tag{3.14}$$

$$-\sin\phi e^{-j\theta} y_{nm} + \cos\phi y_{mm} = y'_{nm} = 0 \tag{3.15}$$

Application of the sequence of Givens rotations $G_M..G_2$, G_1 will result in a completely triangularized data matrix. It is the implementation of these equations which allows us to define the internal and boundary cell function. Discussion of implementation procedures is reserved until we present the recursive QR decomposition algorithm.

3.6.3 Recursive QR Decomposition via Givens Rotation without Explicit Tap-Weight Vector

The following derivation, adapted from McWhirter (1983) and Haykin (1986), outlines the procedure for the solution of the least squares problem in a recursive manner using Givens rotation to perform QR decomposition. From McWhirter's work it is possible to do this without explicit computation of the tap-weight vector. Since only the residual error is required to perform ANC therefore,

McWhirter's Systolic II implementation may be used for the greatest computational efficiency and stability. Much of this derivation is based upon Haykin's (Chapter 10) and McWhirter's work. Readers wishing to investigate more theory are directed to Haykin's (1986) comprehensive book.

Definitions:

$$\xi(n) = \sum_{i=1}^{n} \lambda^{n-i} |e_i|^2 \qquad (3.16)$$

Where n is the number of data points, $\xi(n)$ is the exponentially weighted index of performance and λ is the exponential weighting factor.

$$e_i = d_i - \mathbf{w}^H(n)\mathbf{u}_i \qquad (3.17)$$

e_i is the error, d_i the desired output, $\mathbf{w}^H(n)$ the weight vector and $\mathbf{u}_i$ the input vector defined as follows:

$$\mathbf{w}^H(n) = [w_1, w_{n-1}, \ldots, w_{n-M+1}] \qquad (3.18)$$

$$\mathbf{u}^H(i) = [u_i, u_{i-1}, \ldots, u_{i-M+1}] \qquad (3.19)$$

The error vector $\boldsymbol{\varepsilon}(n)$:

$$\boldsymbol{\varepsilon}^H(n) = [e_1, e_2, \ldots, e_n] \qquad (3.20)$$

or

$$\boldsymbol{\varepsilon}(n) = \mathbf{b}(n) - \mathbf{A}(n)\mathbf{w}(n) \qquad (3.21)$$

Where the desired response $\mathbf{b}(n)$ is:

$$\mathbf{b}^H(n) = [d_1, d_2, \ldots, d_n] \qquad (3.22)$$

and the data matrix, **A**, is:

$$\mathbf{A}^H(n) = [\mathbf{u}(1),\mathbf{u}(2),\ldots,\mathbf{u}(M),\ldots,\mathbf{u}(n)] \qquad (3.23)$$

$$= \begin{bmatrix} u(1) & 0 & 0 & 0 & 0 \\ u(2) & u(1) & 0 & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ u(M) & \cdot & \cdot & \cdot & u(1) \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ u(n) & u(n-1) & \cdot & \cdot & u(n-M+1) \end{bmatrix}$$

From 3.16 and 3.20 we can rewrite the performance index as:

$$\xi(n) = \varepsilon^H(n)\Lambda(n)\varepsilon(n) \qquad (3.24)$$

where: $$\Lambda(n) = \text{diag}\,[\lambda^{n-1},\lambda^{n-2},\ldots,\lambda] \qquad (3.25)$$

or in terms of the euclidean norm:

$$\xi(n) = ||\Lambda(n)\varepsilon(n)||^{1/2}{}^2 \qquad (3.26)$$

For minimization purposes this is equivalent to:

$$\xi(n) = ||\Lambda(n)\varepsilon(n)|| \qquad (3.27)$$

Part I: Generate Least Squares expression for ξ_{min}.

Let $$\mathbf{Q}(n)\Lambda(n)\mathbf{A}(n) = \begin{bmatrix}\mathbf{R}(n)\\ \mathbf{0}\end{bmatrix} \qquad (3.28)$$

so that **R** is MxN, A is NxM, Λ is also NxN, **w** is Mx1 **b** is Nx1 where M is the array order (tap-weights) and N is the number of data points.

We define $\mathbf{Q}$ to be unitary i.e.

$$\mathbf{Q}^T = \mathbf{Q}^{-1} \tag{3.29}$$

Further:

$$\mathbf{Q}(n) = \begin{bmatrix} \mathbf{P}(n) \\ \mathbf{S}(n) \end{bmatrix} \tag{3.30}$$

Where $\mathbf{P}(n)$ is MxN and $\mathbf{S}(n)$ is (N-M)xN

Since the norm of a vector is unaffected when premultiplied by a unitary matrix we have:

$$\xi(n) = ||\mathbf{Q}(n)\mathbf{\Lambda}(n)\ \boldsymbol{\varepsilon}(n)|| \tag{3.31}$$

by substituting eqn. 3.22:

$$\xi(n) = ||\mathbf{Q}(n)\mathbf{\Lambda}(n)\ [\mathbf{b}(n) - \mathbf{A}(n)\mathbf{w}(n)]|| \tag{3.32}$$

$$= ||\mathbf{Q}(n)\mathbf{\Lambda}(n)\mathbf{b}(n) - \mathbf{Q}(n)\mathbf{\Lambda}(n)\mathbf{A}(n)\mathbf{w}(n)|| \tag{3.33}$$

Substituting eqns. 29 and 30 we get:

$$\xi(n) = || \begin{bmatrix} \mathbf{P}(n) \\ \mathbf{S}(n) \end{bmatrix} \mathbf{\Lambda}(n)\mathbf{b}(n) - \begin{bmatrix} \mathbf{R}(n) \\ \mathbf{0} \end{bmatrix} \mathbf{\Lambda}(n)\mathbf{A}(n)\mathbf{w}(n)|| \tag{3.34}$$

$$= || \begin{bmatrix} \mathbf{U}(n) \\ \mathbf{V}(n) \end{bmatrix} - \mathbf{R}(n)\mathbf{w}(n) || \tag{3.35}$$

where:

$$\begin{bmatrix} \mathbf{U}(n) \\ \mathbf{V}(n) \end{bmatrix} = \mathbf{Q}(n)\mathbf{\Lambda}(n)\mathbf{b}(n) \tag{3.36}$$

and:

$$\mathbf{U}(n) = \mathbf{P}(n)\mathbf{\Lambda}(n)\mathbf{b}(n) \tag{3.37}$$

$$\mathbf{V}(n) = \mathbf{S}(n)\mathbf{\Lambda}(n)\mathbf{b}(n) \tag{3.38}$$

$\mathbf{U}$ is Mx1 and V is (N-M)x1.

From eqn. 3.35 we obtain the least squares solution via:

$$\mathbf{R}(n)\mathbf{w}(n) = 0 \tag{3.39}$$

Therefore $\xi_{min}(n) = ||\mathbf{V}(n)||$ (3.40)

when eqn. 3.39 is satisfied.

Part II. We now wish to develop a more precise expression for $\boldsymbol{\varepsilon}$, the error vector.

From eqn. 3.22

$$\mathbf{\Lambda}(n)\boldsymbol{\varepsilon}(n) = \mathbf{\Lambda}(n)\mathbf{b}(n) - \mathbf{\Lambda}(n)\mathbf{A}(n)\mathbf{w}(n) \tag{3.41}$$

and from eqn. 3.28:

$$\mathbf{\Lambda}(n)\mathbf{A}(n) = \mathbf{Q}^T(n)\begin{bmatrix}\underline{R}(n)\\ \mathbf{0}\end{bmatrix} \tag{3.42}$$

thus

$$\mathbf{\Lambda}(n)\boldsymbol{\varepsilon}(n) = \mathbf{\Lambda}(n)\mathbf{b}(n) - \mathbf{Q}^T(n)\begin{bmatrix}\underline{R}(n)\\ \mathbf{0}\end{bmatrix}\mathbf{w}(n) \tag{3.43}$$

subs. eqn. 3.30

$$= \mathbf{\Lambda}(n)\mathbf{b}(n) - \mathbf{P}^T(n)\mathbf{R}(n)\mathbf{w}(n) \tag{3.44}$$

subs. eqn. 3.38 (the least squares requirement)

$$= \mathbf{\Lambda}(n)\mathbf{b}(n) - \mathbf{P}^T(n)\mathbf{U}(n) \tag{3.45}$$

now from eqn 3.36 :

$$\mathbf{\Lambda}(n)\mathbf{b}(n) = \mathbf{Q}^T(n)\begin{bmatrix}\underline{U}(n)\\ \underline{V}(n)\end{bmatrix} \tag{3.46}$$

$$= \mathbf{P}^T(n)\mathbf{U}(n) + \mathbf{S}^T(n)\mathbf{V}(n) \tag{3.47}$$

Lastly, from eqns. 3.45 and 3.47 we have

$$\mathbf{\Lambda}(n)\mathbf{e}(n) = \mathbf{S}^T(n)\mathbf{V}(n) \tag{3.48}$$

Part III. We can now generate a recursive form for the triangularization of **R**.

We have, from eqn. 3.28:

$$\mathbf{Q}(n)\mathbf{\Lambda}(n)\mathbf{A}(n) = \begin{bmatrix}\underline{\mathbf{R}(n)}\\ \mathbf{0}\end{bmatrix} \tag{3.49}$$

Let:

$$\mathbf{Q}(n-1)\mathbf{\Lambda}(n-1)\mathbf{A}(n-1) = \begin{bmatrix}\underline{\mathbf{R}(n-1)}\\ \mathbf{0}\end{bmatrix} \tag{3.50}$$

and

$$\overline{\mathbf{Q}(n-1)} = \begin{bmatrix}\underline{\mathbf{Q}(n-1)} & \underline{0}\\ \underline{0} & 1\end{bmatrix} \tag{3.51}$$

thus $\overline{\mathbf{Q}(n-1)}$ is the augmented NxN unitary matrix.

$$\overline{\mathbf{Q}(n-1)}\mathbf{\Lambda}(n)\mathbf{A}(n) = \mathbf{Q}(n-1)\begin{bmatrix}\lambda\mathbf{\Lambda}(n-1)\mathbf{A}(n-1)\\ \hline \mathbf{a}^{T}(n)\end{bmatrix} \tag{3.52}$$

$$\text{from eqn. 3.50} = \begin{bmatrix}\underline{\lambda\mathbf{R}(n-1)}\\ \underline{\mathbf{0}}\\ \mathbf{a}^{T}(n)\end{bmatrix} \tag{3.53}$$

Therefore

$$\mathbf{Q}(n) = \hat{\mathbf{Q}}(n)\,\overline{\mathbf{Q}(n-1)} \tag{3.54}$$

and

$$\hat{\mathbf{Q}}(n) \bullet \begin{bmatrix}\underline{\lambda\mathbf{R}(n-1)}\\ \underline{\mathbf{0}}\\ \mathbf{a}^{T}(n)\end{bmatrix} = \begin{bmatrix}\underline{\mathbf{R}(n)}\\ \underline{\mathbf{0}}\\ \underline{0}\end{bmatrix} \tag{3.55}$$

$$\hat{\mathbf{Q}}(n-1)\mathbf{\Lambda}(n)\mathbf{A}(n) = \begin{bmatrix}\underline{\mathbf{R}(n)}\\ \mathbf{0}\end{bmatrix} \tag{3.56}$$

thus $\hat{\mathbf{Q}}(n)$ performs the Given's rotation and triangularizes **R**.

In terms of $\mathbf{Q}(n)$ (from eqn. 3.36):

$$\begin{bmatrix}\underline{U}(n)\\ \underline{V}(n)\end{bmatrix} = \mathbf{Q}(n)\mathbf{\Lambda}(n)\mathbf{b}(n) \tag{3.57}$$

subs. eqn. 3.54

$$= \hat{\mathbf{Q}}(n)\overline{\mathbf{Q}(n-1)}\mathbf{\Lambda}(n)\mathbf{b}(n) \tag{3.58}$$

subs. eqn. 3.51

$$= \hat{\mathbf{Q}}(n)\left[\begin{array}{c|c}\underline{Q}(n-1) & \underline{0}\\ \hline \underline{0} & 1\end{array}\right]\begin{bmatrix}\lambda\underline{\Lambda}(n-1)\underline{b}(n-1)\\ d(n)\end{bmatrix} \tag{3.59}$$

$$= \hat{\mathbf{Q}}(n)\begin{bmatrix}\lambda\underline{Q}(n-1)\underline{\Lambda}(n-1)\underline{b}(n-1)\\ d(n)\end{bmatrix} \tag{3.60}$$

from eqn. 3.56

$$\begin{bmatrix}\underline{U}(n)\\ \underline{V}(n)\end{bmatrix} = \hat{\mathbf{Q}}(n)\begin{bmatrix}\lambda\underline{U}(n-1)\\ \lambda\underline{V}(n-1)\\ \hline d(n)\end{bmatrix} \tag{3.61}$$

Part IV: From the results in parts II and III we can now isolate the n^{th} element (e_n) in the error vector ($\varepsilon(n)$).

From eqn. 3.30

$$\mathbf{Q}(n) = \begin{bmatrix}\underline{P}(n)\\ \underline{S}(n)\end{bmatrix} \tag{3.62}$$

From eqn. 3.54

$$\mathbf{Q}(n) = \hat{\mathbf{Q}}(n)\,\overline{\mathbf{Q}(n-1)} \tag{3.63}$$

and

$$\hat{\mathbf{Q}}(n) = \left[\begin{array}{c|c|c}\underline{\tau}(n) & \underline{0} & \underline{\kappa}(n)\\ \hline \underline{0} & \underline{I}(n) & \underline{0}\\ \hline \underline{\sigma}^T(n) & \underline{0} & \gamma(n)\end{array}\right] \tag{3.64}$$

where $\mathbf{I}$ is the (N-M-1)x(N-M-1) identity matrix, κ is the Mx1 column vector of sine factors, τ is the MxM diagonal

matrix of cosine factors, σ^T 1xM row vector of negative sine factors. Section V of this exercise explains the components of $\hat{Q}(n)$ in greater detail.

Combining eqns. 3.62, 3.63, and 3.64 we get :

and
$$Q(n) = \begin{bmatrix} \tau(n) & 0 & \kappa(n) \\ 0^T & I(n) & 0 \\ \sigma^T(n) & 0 & \gamma(n) \end{bmatrix} \bullet \begin{bmatrix} P(n-1) & 0 \\ S(n-1) & 0 \\ 0 & 1 \end{bmatrix} \tag{3.65}$$

$$= \begin{bmatrix} \tau(n)P(n-1) & \kappa(n) \\ \sigma^T(n)P(n-1) & 0 \\ & \gamma(n) \end{bmatrix} \tag{3.66}$$

Lastly, the portion of Q(n) in which we are interested (from eqn. 3.48 $\Lambda(n)e(n) = S^T(n)V(n)$) is:

$$S^T(n) = \begin{bmatrix} S^T(n-1) \\ 0 \end{bmatrix} \bullet \begin{bmatrix} P^T(n-1)s(n) \\ \gamma(n) \end{bmatrix} \tag{3.67}$$

Now from eqn. 3.61 $V(n)$:

$$\begin{bmatrix} U(n) \\ V(n) \end{bmatrix} = \hat{Q}(n) \bullet \begin{bmatrix} \lambda U(n-1) \\ \lambda V(n-1) \\ d(n) \end{bmatrix} \tag{3.68}$$

subs. eqn 3.64

$$= \begin{bmatrix} \tau(n) & 0 & \kappa(n) \\ 0^T & I(n) & 0 \\ \sigma^T(n) & 0 & \gamma(n) \end{bmatrix} \bullet \begin{bmatrix} \lambda U(n-1) \\ \lambda V(n-1) \\ d(n) \end{bmatrix} \tag{3.69}$$

$$V(n) = \begin{bmatrix} \lambda V(n-1) \\ \hline \lambda s^T(n)U(n-1) + \lambda(n)d(n) \end{bmatrix} \tag{3.70}$$

$$= \begin{bmatrix} \lambda \underline{V}(n-1) \\ \alpha(n) \end{bmatrix} \tag{3.71}$$

where $\alpha(n) = \lambda \underline{s}^T(n)\underline{U}(n-1) + \gamma(n)d(n)$ (3.72)

Finally..., combining eqns. 3.48, 3.67, and 3.72 we get an expression for the n^{th} element of the error vector:

$$\Lambda(n)\varepsilon(n) = \begin{bmatrix} \underline{S}^T(n-1) & \underline{P}^T(n-1)\underline{s}(n) \\ \underline{0} & \gamma(n) \end{bmatrix} \cdot \begin{bmatrix} \lambda \underline{V}(n-1) \\ \alpha(n) \end{bmatrix} \tag{3.73}$$

$$= \lambda \begin{bmatrix} \underline{S}^T(n-1)\underline{V}(n-1) \\ \underline{0} \end{bmatrix} + \alpha(n) \begin{bmatrix} \underline{P}^T(n-1)\underline{s}(n) \\ \gamma(n) \end{bmatrix} \tag{3.74}$$

And now the n^{th} element can be extracted:

$$e_n = \alpha(n)\gamma(n) \tag{3.75}$$

Part V: An explanation of $\hat{Q}(n)$.

From eqn. 3.64

$$\hat{Q}(n) = \begin{bmatrix} \underline{\tau}(n) & \underline{0} & \underline{\kappa}(n) \\ \underline{0}^T & \underline{I}(n) & \underline{0} \\ \bar{\sigma}^T(n) & \underline{0}^T & \bar{\gamma}(n) \end{bmatrix} \tag{3.76}$$

which is the product of a series of Given's rotations $[G_M \cdot \ldots \cdot G_2 \cdot G_1]$ (see section 3.6.2).

If the appropriate Given's matrices are constructed for M=4 from eqn. 3.13 , the final product of matrices would be:

$$\hat{\mathbf{Q}}(n) = \begin{bmatrix} c_1 & 0 & 0 & 0 & 0 & . & 0 & s_1 \\ 0 & c_2 & 0 & 0 & 0 & . & 0 & s_2 \\ 0 & 0 & c_3 & 0 & 0 & . & 0 & s_3 \\ 0 & 0 & 0 & c_4 & 0 & . & 0 & s_4 \\ 0 & 0 & 0 & 0 & 1 & . & 0 & . \\ 0 & 0 & . & . & 0 & 1 & 0 & . \\ 0 & 0 & 0 & 0 & . & 1 & . & . \\ 0 & 0 & 0 & 0 & . & . & . & . \\ -s_1 & -s_2 & -s_3 & -s_4 & 0 & . & . & \pi c_i \end{bmatrix} \tag{3.77}$$

$$\tau(n) \equiv \text{diag}[c_1,.....,c_M] \tag{3.78}$$

$$\kappa^T(n) \equiv [s_1,.....,s_M] \tag{3.79}$$

$$\sigma^T(n) \equiv [-s_1,.....,-s_M] \tag{3.80}$$

$$\gamma(n) \equiv \pi c_i \tag{3.81}$$

The vector $\alpha(n)$ is given by eqn. 3.72 and can be compared to eqn 3.15:

$$\alpha(n) = \lambda s^T(n)U(n-1) + \gamma(n)d(n) \tag{3.82}$$

$$y'_{mm} = \sin\phi e^{j\theta} y_{nm} + \cos\phi y_{mm} \tag{3.83}$$

By looking at figure 3.5 it can be seen that $\alpha(n)$ is generated as the array rotates y_{mm} with the incoming data y_{nm} ($U(n-1)$).

3.6.4 Implementation of Systolic Array II as a Noise Canceller (SANC)

We now solve the simultaneous equations resulting from the Givens rotation in order to implement them in our triangular systolic array. They are (for complex data):

$$-\sin\phi e^{-j\theta} y_{nm} + \cos\phi y_{mm} = y'_{nm} \tag{3.84}$$

$$\cos\phi y_{mm} + \sin\phi e^{j\theta} y_{nm} = y'_{mm} \tag{3.85}$$

let:

$$c \equiv \cos\phi$$

$$s \equiv \sin\phi$$

$$p \equiv \sin\phi e^{j\theta}$$

and:

$$R \equiv y_{mm}$$

$$R' \equiv y'_{mm}$$

$$U_{in} \equiv y_{nm}$$

$$U_{out} \equiv y'_{nm}$$

therefore

$$-p^{*}R + cU_{in} = U_{out} \tag{3.87}$$

$$cR + pU_{in} = R' \tag{3.86}$$

We require the systolic array to perform three types of tasks:

1) Boundary Cell : Solution of the Givens equations where $U_{out}=0$ for generation of the rotation parameters.

2) Internal Cell : Propagation of rotation parameters through the input data.

3) Final Cell : Application of the 'product of cosines' parameter to generate the least squares residual.

Boundary Cell

$$cR + pU_{in} = R' \tag{3.88}$$

$$-p^{*}R + cU_{in} = U_{out} = 0 \tag{3.89}$$

$$c^{2} + |p|^{2} = 1 \tag{3.90}$$

We must solve for the rotation parameters (c,p) and update the cell contents (R').

A) $U_{in} = 0, \quad R \neq 0 \quad \rightarrow \quad R' = R, \; c=1, \; p=(0,0)$

B) $U_{in} \neq 0, \quad R = 0 \quad \rightarrow \quad R' = U_{in}, \; c=0, \; p=(1,0)$

C) $U_{in} = 0, \quad R = 0 \quad \rightarrow$ use either A or B

D) $U_{in} \neq 0, \quad R \neq 0$

$$p = U_{in} / \sqrt{(R)^2 + (U_{in})^2} \tag{3.91}$$

$$c = p^* R / U_{in} \tag{3.91a}$$

$$= R / \sqrt{(R)^2 + (U_{in})^2} \tag{3.91b}$$

$$= +\sqrt{1 - |p|^2} \tag{3.91c}$$

$$R' = cR + pU_{in} \tag{3.92}$$

Calculation of 'p' before 'c' allows us to develop a memory saving algorithm for implementation of the Givens rotation. If we inspect 3.91b and 3.92 it is obvious that both 'R' and 'c' are always positive. The sign (real data) or phase

(complex data) information is contained in the sine rotation parameter 'p'.

At first glance it would seem necessary to maintain three matrices one for each of 'c', 'p' and 'R'. However we can reduce memory requirements by 1/3 by storing the upper triangular matrix containing 'p' in the the lower half of the 'c' matrix and by using the following transform to extract 'p':

$$p(x,y) = c(M+3-x,\ M+2-y) \tag{3.93}$$

We now have to maintain two matrices one MxM the other (M+2)x(M+1). Use of this transform results in no significant increase in computation time.

For further reductions in memory requirements we look at eqn. 3.91c and notice that it is unnecessary to maintain a matrix of 'c' values because they can be calculated, when necessary, from 'p' alone. Notice that this would be impossible if eqns. 3.91a or 3.91b were used because the value of U_{in} is not stored. Also if c were the basic rotation parameter (i.e. 'p' calculated from 'c', 'R' and U_{in}) it would be impossible to formulate an equivalent of eqn 3.91c since the sign or phase information would be

unavailable. We now replace the upper half of the 'c','p' matrix by the upper triangular matrix of 'R' and the result is a total reduction in memory requirement of 2/3. The remaining matrix has dimensions (M+2)x(M+1). The cost in terms of computation is that we must now calculate 'c' explicitly during an internal cell computation resulting in an increase in processing time of about 30%.

The choice of equation for the computation of 'c' in the boundary cell will determine which input parameter ('R' or U_{in}) will have to be checked for zero value to ensure numerical stability of the algorithm. We use eqn. 3.91b because it involves no additional square root calculation after use of eqn. 3.90. Figure 3.4 illustrates the flow charting for boundary cell computations.

Internal Cell

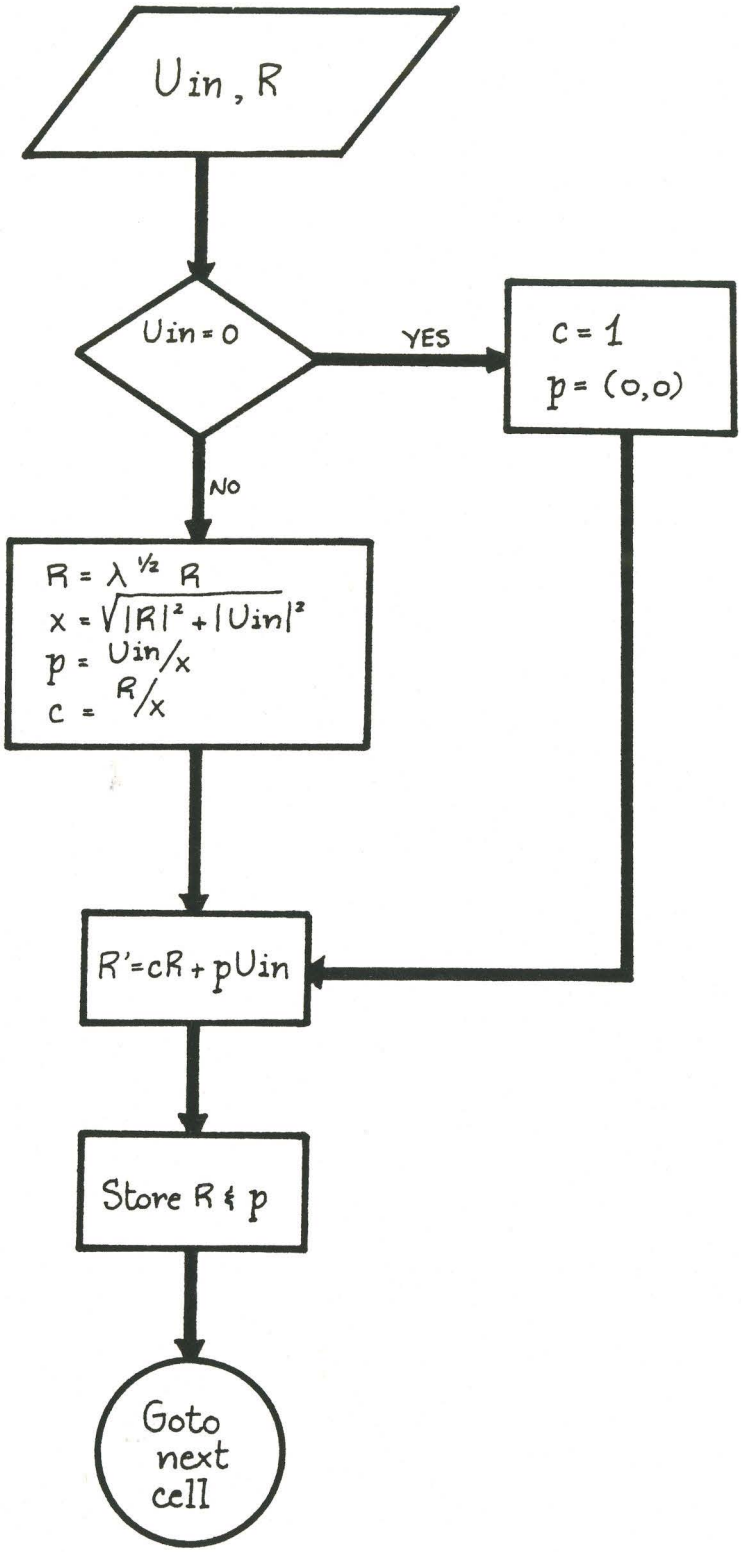
$$c = \sqrt{1 - |p|^2} \tag{3.94}$$

$$R' = cR + pU_{in} \tag{3.95}$$

$$U_{out} = -p^{*}R + cU_{in} \tag{3.96}$$

Equation 3.93 is only necessary when using the minimum memory algorithm. The increase in processing time of 30%

fig 3.4 Flow chart for Boundary Cell Computation



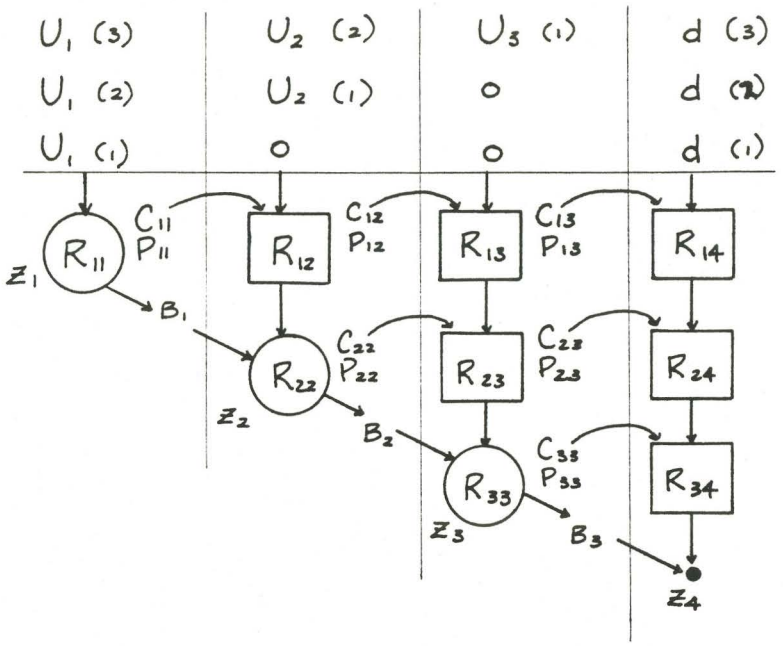
is due to the large arrays being processed and thus the high internal to boundary cell ratio. Using the minimum memory algorithm we now have three rather than two computations to perform.

Final Cell

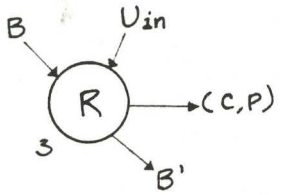
$$U_{out} = \gamma^{1/2} U_{in} \tag{3.97}$$

Figure 3.5 illustrates the flow of data in the systolic array (real data example). The first row of the systolic array (i.e. the top row of cells) turns each arriving row of U into a row with a zero in its first entry. Results are output to the second row where the next element from the left is eliminated (in the boundary cell). While triangularizing the data matrix, the contents of the cell 'R' are updated and the rotation parameters sent to the right (for use in the internal cells). Because of the temporal skew imposed upon the incoming data the rotation parameters determined at the boundary cell reach the internal cell at the correct time. This arrangement ensures that as each row of data flows through the array it interacts with the previously stored triangular matrix R(n-1) and undergoes the sequence of Givens rotations described in the previous section.

fig. 3.5 Data Flow through the Systolic Array

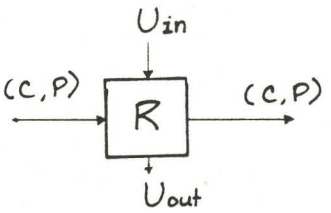


Boundary cell :



See fig 3.4

Internal cell :

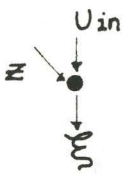


$R \leftarrow \lambda^{1/2} R$

$U_{out} = CU_{in} - PR$

$R' = PU_{in} + CR$

Final cell :



Error = $\xi = z \cdot U_{in}$

The parameter $\gamma(n)$ is updated at each boundary cell as the cosine parameters are computed and passed to temporary storage. The extra delay is a direct result of the temporal skew imposed upon the incoming data. For programming ease the matrix is worked from top to bottom and left to right. We begin by calculating R_{14} from U_{in}, s_{13} and c_{13} (the Givens parameters from the time before). Working down the columns each U_{out} from the upper cell becomes the U_{in} to the cell below and is rotated with contents of the cell (R) and c and s from the cell to left. After each calculation c and s are moved to the right by incrementing the appropriate index.

γ is propagated via two vectors ζ and β. At each boundary cell we perform:

$$\zeta(n) = \beta(n-1) \tag{3.98}$$

$$\text{and} \quad \beta(n) = \zeta(n) \cdot c(n) \tag{3.99}$$

The final cell calculation is in place of a boundary cell calculation at the bottom right of the array. Since this requires γ, $\zeta(M+1)$ must be updated from $\beta(M)$ prior to computation of the residual.

A program listing of the SANC (minimum memory algorithm) is presented in Appendix II.

CHAPTER 4

REAL DATA ACQUISITION

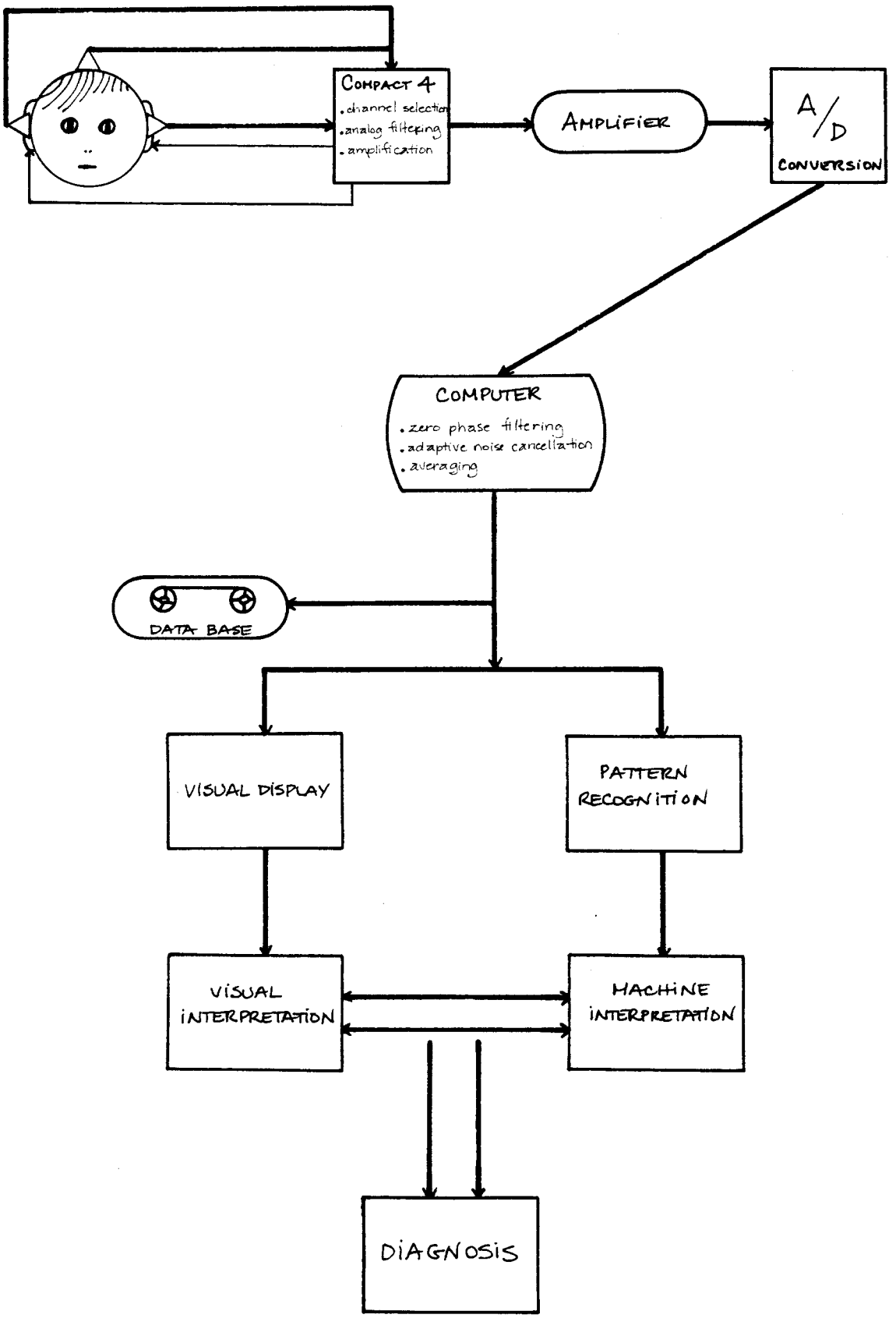
4.1 Introduction

The ultimate goal of our study in evoked potential processing is to develop a clinically useful system. Real signal collection and analysis is fundamental to the attainment of this goal. Use of real signals has two advantages:

i) We are able to evaluate the appropriateness of proposed signal processing techniques. We are also better able to characterize the signal of interest (e.g. band-width) and estimate the range of values for parameters (e.g. S/N ratio) in simulation studies.

ii) Evaluation of the success of the scheme should be done using real data. Use of real data should follow simulation studies when we are able to distinguish signal processing, data acquisition or physiologically-based system problems.

fig. 4.1 Evoked Potential Acquisition System



A final system for EP processing might look like the diagram in figure 4.1 . The stimulation unit consists of visual, auditory or electrical hardware with programmable intensity, stimulation rates, etc. Collection begins at the electrodes and the signal is fed to the pre-amplifier. Further amplification and analog filtering may be included prior to digitization of the signal. A/D conversion allows the implementation of much more advanced processing such as adaptive noise cancellation and zero-phase digital filtering. The choice of processing and acquisition parameters will depend upon the final use of the data (i.e. visual interpretation by a physician or machine classification). Pattern recognition schemes have already been tested and developed for BAEP and VEP (Madhavan et al., 1984). As attempts are made to standardize tests and test results, the necessity for access to a central data base will become more important.

4.2 Data Acquisition

4.2.1 Protocol for BAEP Signal Acquisition

The importance of collecting good, clean data cannot be over emphasized. Much unnecessary processing and can be avoided by taking appropriate care to acquire data which

has as little noise as possible. In physiological studies the effect of environmental influences on the subject must always be considered. Experience has determined that best results can be obtained when the following guidelines are applied:

1) The subject should be resting horizontally with adequate neck and head support. A stretcher provides a mobile and comfortable surface.

2) Lights should be dimmed and where possible, turned off. Environmental noise (e.g. computer) should be minimized. If possible sleep should be induced. A significant reduction in noise of a muscular origin will be observed if the patient does fall asleep.

3) Plenty of time should be reserved for the test to ensure that the subject remains as relaxed as possible throughout the session.

4) The subject should be exposed to the stimulus by performing a dummy run of about 100 stimulations.

5) Once the acquisition has begun interruptions should be entirely avoided.

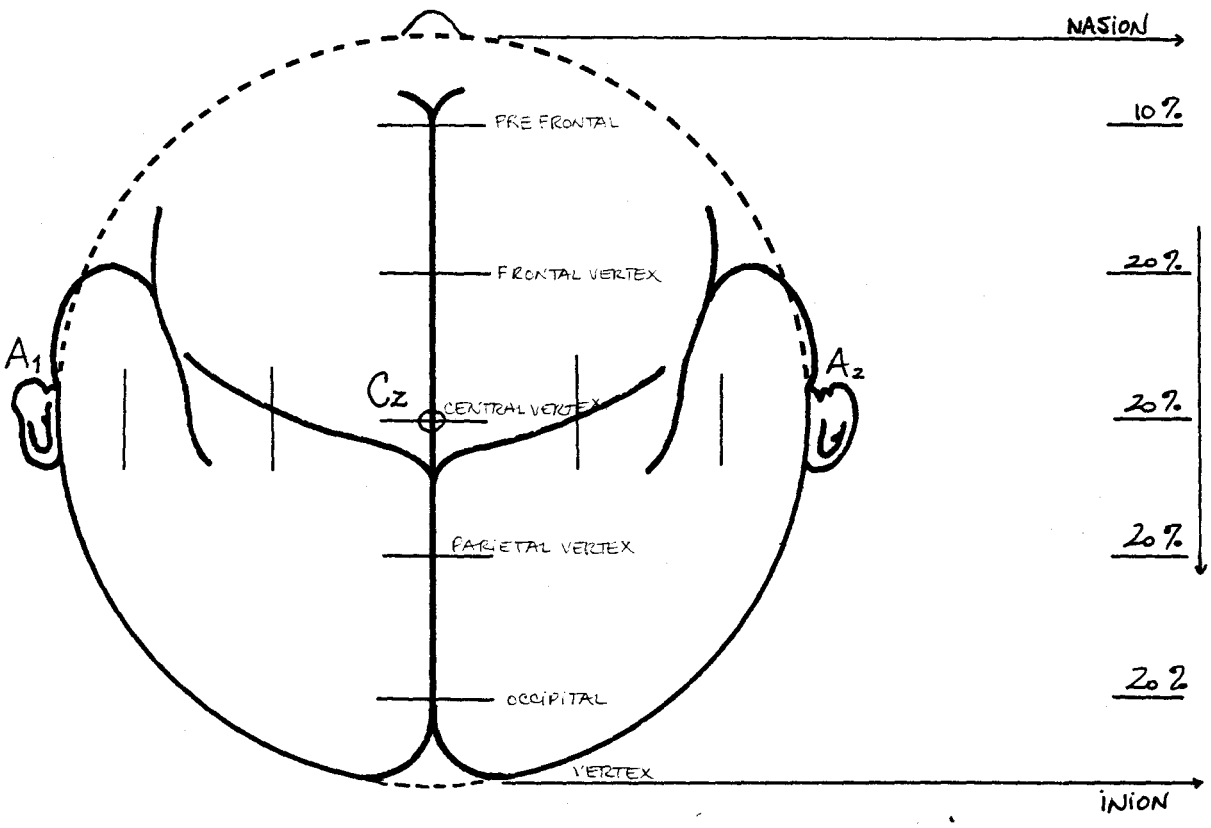
The front end of our acquisition system was a Nicolet Compact-4 (C-4) which is used in clinics for EP extraction via averaging. Headphones were provided for stimulation of the subject's auditory pathway. Stimulation rates of 3.0 per second were used with a rarefaction pulse of 200 usec duration. Rarefaction was used to improve the resolution of waves IV and V (Chiappa, 1983). Clinical tests use a rate of 9.8 per second but we were limited by the disk-writing speed of the PDP-11 used to store the individual EEG records. A total of 2000 records were stored requiring about 2005 to 2020 stimulations (a small percentage were rejected due to large noise spikes). Stimulation level was 60-70 dB above hearing threshold. Table 4.1 details the protocol used in our acquisition and briefly describes the effect of each parameter. Further details of parameters and their influence may be found in Chiappa (1983).

Table 4.1

Parameter	Value	Description
Click Intensity	60 -70 dB above threshold	As intensity increases absolute latency decreases.
Stimulus Rate	3.0 per sec	As the rate increases the percentage of subjects with identifiable waves decreases.
Band-Pass Filtering	150Hz-3kHz	Phase distortion will result from tight analog filtering.
Repetitions	2000	More repetitions improve signal to noise ratio but worsen the habituation.

BAEP are recorded from the vertex (C_z) which in the centre of the scalp as described by the International 10-20 system (Chiappa, 1983) in figure 4.2. The stimulus is monaural and the reference electrode was placed on the mastoid contralateral to the stimulus side (A_1 or A_2). The remaining mastoid (ipsilateral) was used as ground. After the scalp was cleaned with an abrasive paste, gold disk surface electrodes were fastened using collodion. The scalp electrode interface was formed by squeezing an electrolytic paste between the electrode and scalp. Absolute (with respect to the stimulus) and relative

fig. 4.2. Electrode Montage (10-20 system)



latency of the peaks in the BAEP were used to characterize the evoked potential signal. Amplitude information was not considered in this study due to its inherent variability.

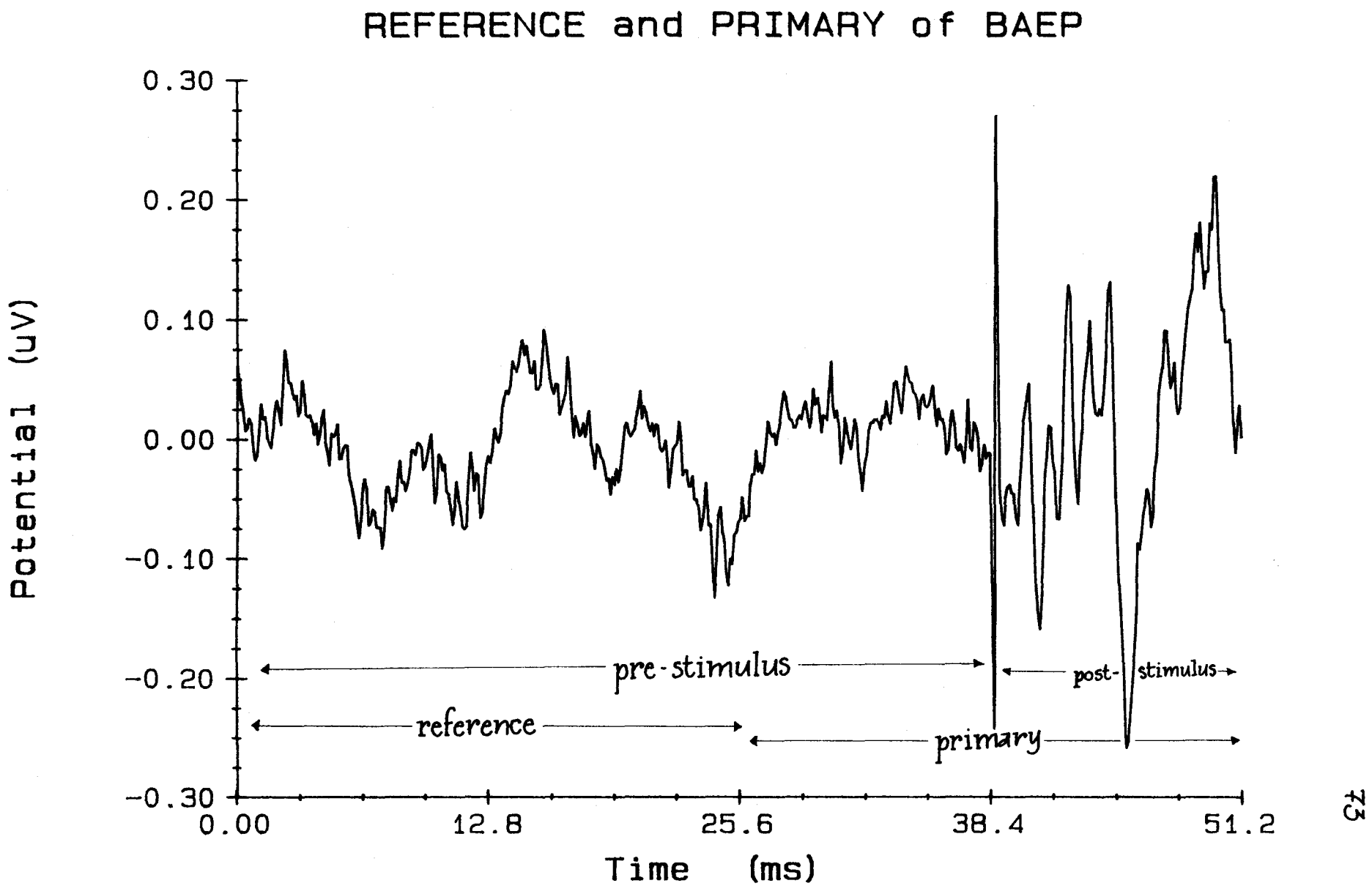
4.2.2 Computer Acquisition

Output from the C-4 was sent to an amplifier for maximum resolution in the A/D process. An LPS-11A Laboratory Interface system was used to perform 12-bit A/D conversion. Dummy runs were used to calibrate the amplifier.

A Fortran program written for a PDP-11/34 was used to control an assembly language data acquisition program. Both programs are listed in Appendix III. The data acquisition program was designed to collect EEG continuously at 10kHz and store the samples in a ring buffer. When a stimulus is elicited from the Compact-4 it causes an interrupt. An additional 128 samples are collected and the last 512 points are unwrapped from the ring buffer is at point 385. Then the EEG sample was written to disk as a single record. In order to validate the collection, the 2000 records were averaged via the PDP and the BAEP was compared to the print out from the C-4.

Figure 4.3 illustrates the result of signal collection and

FIG. 4.3


REFERENCE and PRIMARY of BAEP
0.30
0.20
0.10
0.00
-0.10
-0.20
-0.30
Potential (uV)
0.00
12.8
25.6
38.4
51.2
Time (ms)
pre-stimulus
post-stimulus
reference
primary

averaging using the PDP. The collected wave is composed of two segments, the reference and the primary. These labels are based on the roles of each segment in the ANC scheme described in the previous chapter. Each segment is 25.6 ms in duration and consists of 256 points. For the most part we will be dealing with the primary section because it contains the evoked potential. The stimulus is given 12.8 ms from the beginning of the primary signal. The first 384 points of the signal is referred to as the pre-stimulus signal (reference plus pre-stimulus primary signal) and the second half of the primary signal the post-stimulus signal. The BAEP which is approximately 10 ms in duration, can be seen in the last 12.8 ms of the signal. The stimulus artifact can be identified as the spike at point 129 in the primary or point 385 in the entire record. Wave latencies are measured in milliseconds from the stimulus point.

A total of 22 sessions were conducted using 4 different subjects all in their 20's. All latencies were within normal limits. Primary averaged BAEP signals from two subjects are illustrated in figure 4.4 . The wave forms exhibit similar characteristics with slight differences in peak resolution and latency. The differences between sessions for a single patient were remarkably small as indicated in table 4.2 .

FIG. 4.4

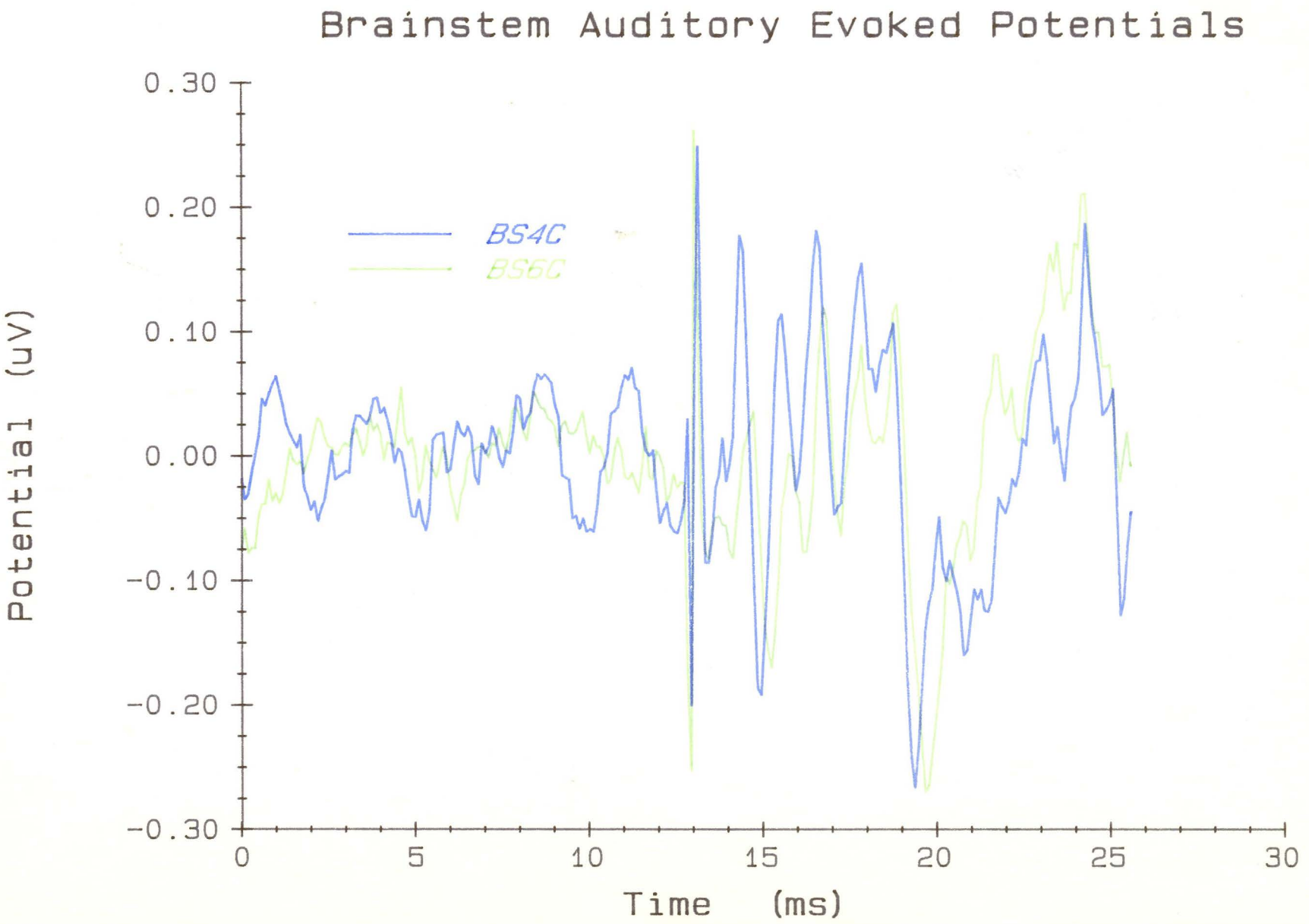

Brainstem Auditory Evoked Potentials
0.30
0.20
0.10
0.00
-0.10
-0.20
-0.30
Potential (uV)
BS4C
BS6C
0
5
10
15
20
25
30
Time (ms)

Table 4.2

Wave # (latency ms)	Session # I	II	III
I	1.72	1.72	1.82
II	2.88	2.84	2.88
III	3.84	3.84	3.86
IV	5.12	5.04	5.08
V	5.76	5.78	5.76

4.3 Real Data Analysis

4.3.1 EEG and BAEP Spectra

We began our analysis by computing the spectrum of the averaged EP record. Figure 4.5 illustrates two such spectra from the averaged EPs of figure 4.4 . For each 512 sample EEG segment the 128 point spectra was estimated using Welch periodograms. The windows had 256 points (128 data samples and 128 0's) and used 4th order optimum Blackman-Harris windows (50% overlap). Appendix IV contains a listing of the spectral estimation program.

Even though the EPs are from two different subjects they exhibit similar characteristics. Three principal components are noticeable. The lowest frequency (approx. 250 Hz.) peak

FIG. 4.5a

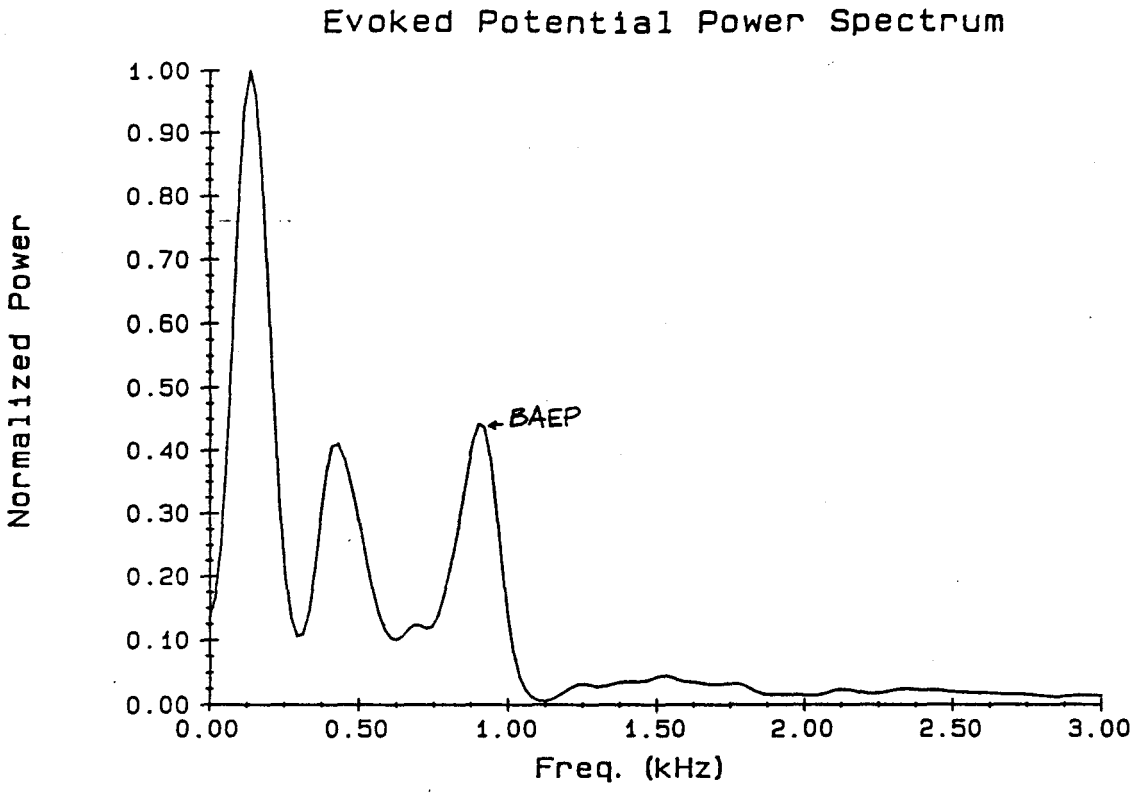
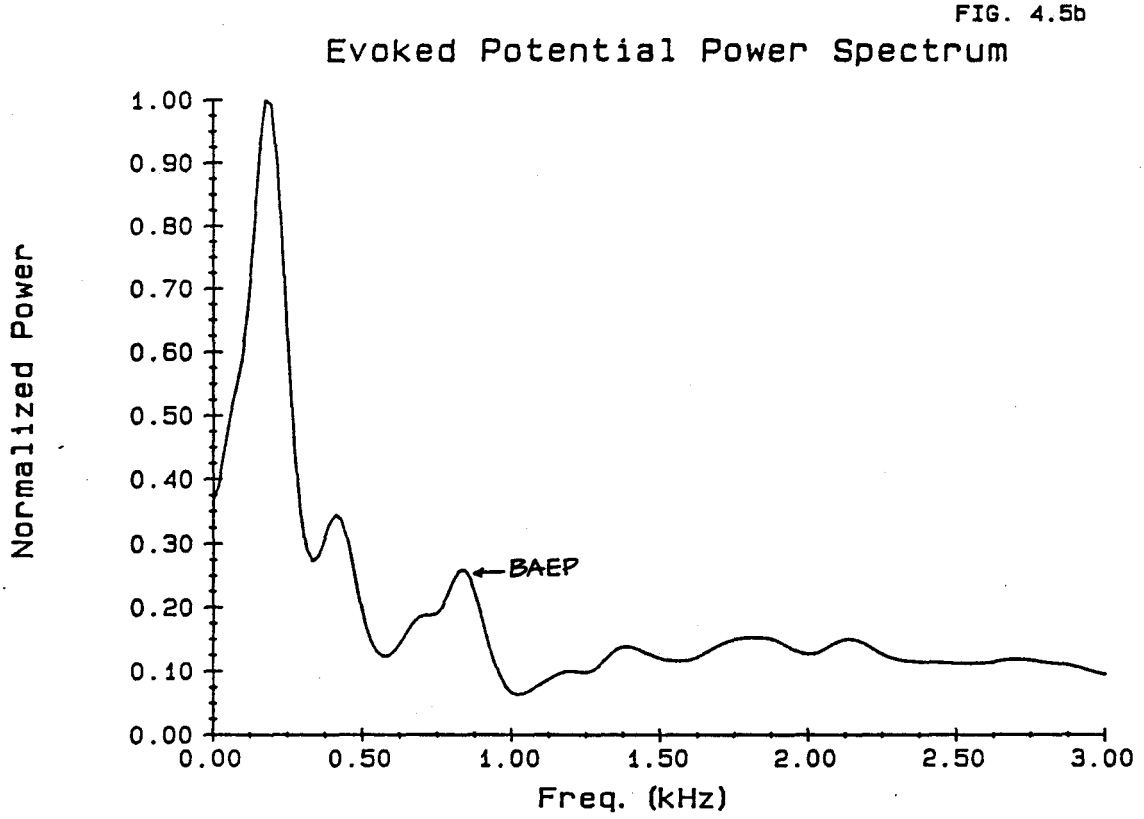


FIG. 4.5b



represents the power in the plateau upon which the five peaks of the averaged EP sit. Generally speaking, the plateau gives wave III its traditionally high amplitude and wave V its lower amplitude. Beyond the plateau the elusive waves VI and VII are sometimes identifiable.

At about 450 Hz. a second peak is present in both spectra. This phenomenon is reported by Boston (1981) and is attributed to the lower frequency content of Wave V. However if we examine figure 4.4 we can see that the pre-stimulus averaged EEG exhibits a dominant wave. This wave has a frequency of about 450 Hz. We therefore conclude that the 2nd spike in the averaged EEG record is at least partially due to a periodic component in the EEG. If this were synchronized to the BAEP it would represent a limiting factor in the averaging of BAEP data since time-locked noise components will not decrease in amplitude as the number of repetitions is increased.

The component at about 1kHz is the fundamental BAEP power. A notch filter (600-1100 Hz) removed the BAEP wave from the times-series record of figure 4.4 . Above 1100 Hz we expect to find power from the stimulus artifact, from high frequency noise such as instrumentation noise and from residual EEG power. There is still a small amount of power

noticeable above 3kHz and below 150 Hz which is due to limitations of analog filtering and use of a 2nd order Butterworth configuration in the C-4 hardware.

While the two records of figure 4.4 are very similar, resolution of the EP spectral component (and the related peaks in the time-series record) is obviously superior in the record labelled BS4C. We can attribute this to:

1) Physiological factors mentioned earlier. The subject for BS4C fell asleep at the beginning of the test and remained asleep. The subject for BS6C fell asleep only briefly and became anxious towards the end of the test producing a larger number of muscle-twitch spikes and more high frequency EEG.

2) The subject for BS4C produce consistently good results. This is due to the natural variation of BAEP recordings in the population.

4.3.2 Zero-Phase Digital Filtering

With our knowledge of the BAEP spectrum we can now proceed to investigate the effect of zero-phase band pass filtering

on the EP waveform. Band-pass filtering affords us a relatively simple and fast technique for improving the SNR and consequently the readability of the BAEP. By using zero-phase filters we can avoid phase distortion of the signal which would directly affect the latency of the BAEP.

We wish to resolve the individual peaks for easy latency computation (possibly using an automated system) without excessively distorting the overall shape of the BAEP. Loss of BAEP shape may remove information in the signal (such as amplitude) used by physicians in diagnosis. Figure 4.6 illustrates the same eight BAEPs filtered with eight different 97th order zero-phase filters.

The first plot shows the unfiltered BAEP with the large biphasic stimulus artifact at time zero. The sharp edges are caused by the low sample rate (which exceeds the Nyquist criterion but not sufficiently for a good temporal resolution). Peaks I-IV are quite clearly resolved but wave V is less clearly visible. There is evidently a small but noticeable amount of high frequency contamination (as we saw in the spectral estimate). By digital filtering from 150 to 3kHz in figure 4.6b we have reduced the negative going component of the stimulus artifact and increased the resolution of all waves especially wave I.

FIG. 4.6a

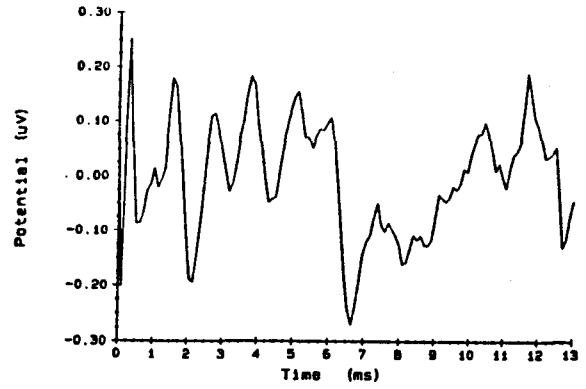


FIG. 4.6b

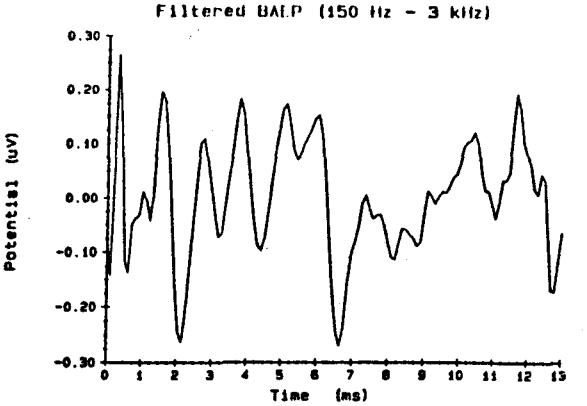


FIG. 4.6c

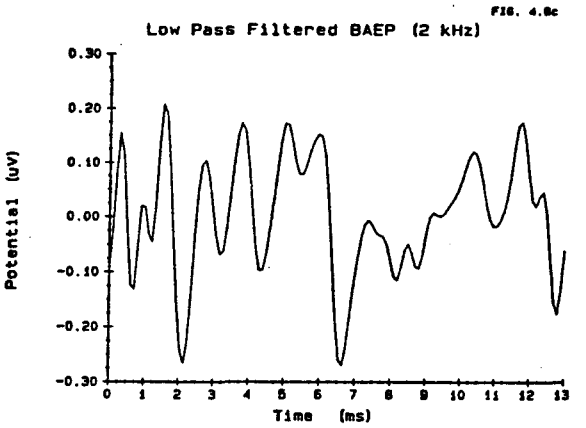


FIG. 4.6d

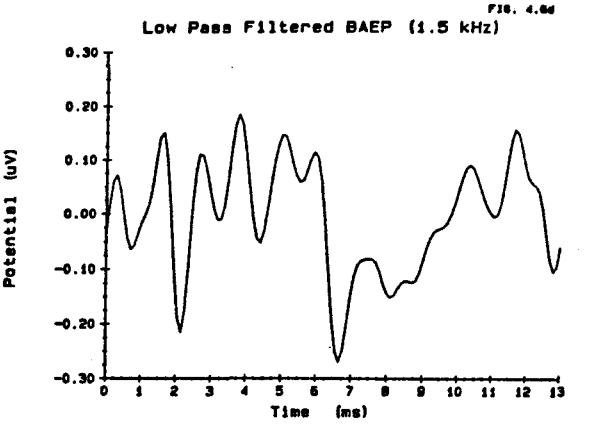


FIG. 4.6e

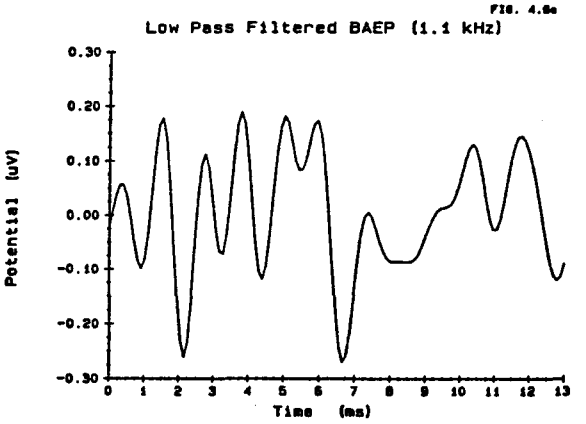


FIG. 4.6f

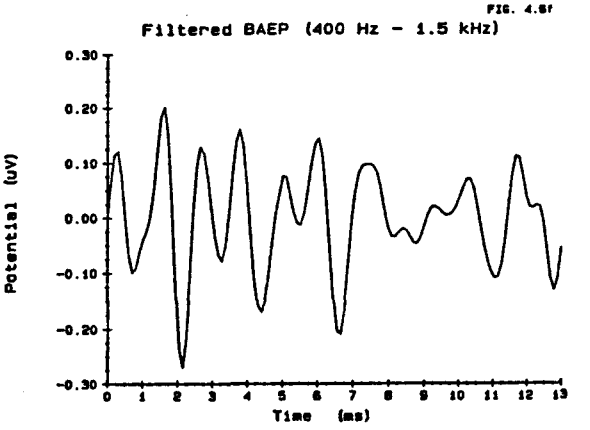


FIG. 4.6g

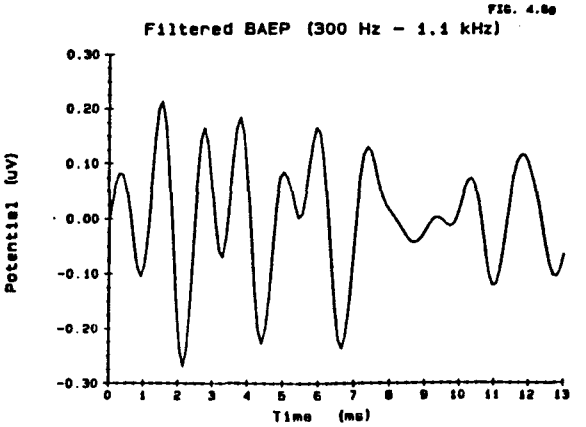
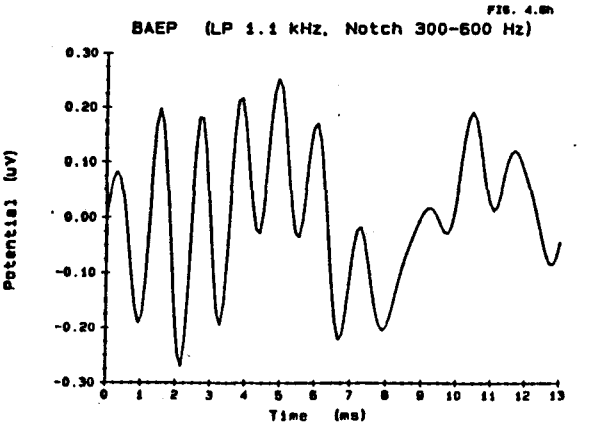


FIG. 4.6h



Figures 4.6 c,d,e were low-pass filtered at 2, 1.5 and 1.1 kHz respectively There is a progressive reduction in high frequency contamination and the rounding of the peaks progresses as the filtering becomes more severe. However, due to the high quality of the original signal, there is little improvement in readability except for the removal of the small pre-wave I artifact. By eliminating more of the low-frequency components (figure's 4.6 f,g) the characteristic plateau shape was removed from the BAEP. While this will not affect our latency measurements significantly it will certainly cause distress to investigators looking at amplitude or shape information.

In the final diagram a filter specially suited to the BS4C trace has been designed. A low pass filter removes all high frequency contamination and a notch filter removes the previously mention 450 Hz component. The resulting wave is easily read and retains the traditional plateau shape. This technique of spectral analysis followed by specialized filter design may be of use in some cases but is undesirable in the single stimulus case since we cannot have apriori knowledge about the frequency characteristics of the individual BAEP.

Before the simulations were begun a BAEP-like signal was

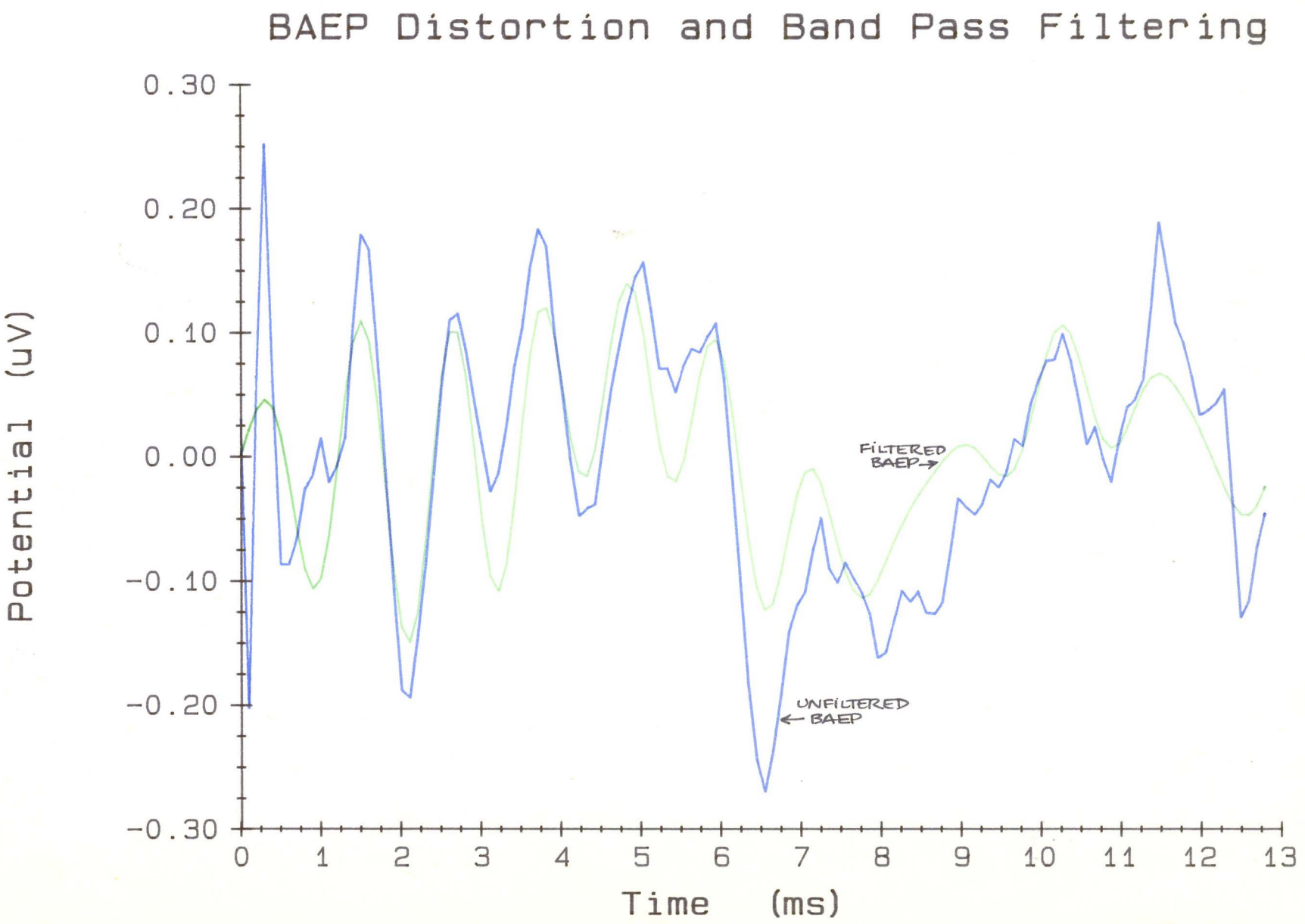
required for use in our additive noise scenario for ANC. To avoid possible complications arising from broad-spectrum signals and for ease of readability we have obtained the signal in figure 4.7. The signal trace is low-pass filtered (1100 Hz) BS4C signal. Notice that the signal is now very simple, resembling a sinusoid of about 1kHz. When compared to the original signal it can be seen that the peaks are no longer perfectly aligned. This is not due to phase distortion but rather to our removal of frequencies in the BAEP.

4.3.3 Signal to Noise Ratio

The purpose of our spectral investigation and band-pass filtering experiments has been to develop techniques for improving SNR of the extracted EP either prior to or after adaptive noise cancellation. BAEP have amplitudes of less than 1 uV in comparison to the EEG which may be as large as 10's of uVs. By limiting the spectrum to 150-3000 Hz we eliminate much of the low frequency power of the EEG (alpha rhythms etc).

An estimate of the SNR of a single BAEP was made by applying the assumptions of stationarity and fixed latency

FIG. 4.7


BAEP Distortion and Band Pass Filtering
Potential (uV)
0.30
0.20
0.10
0.00
-0.10
-0.20
-0.30
0
1
2
3
4
5
6
7
8
9
10
11
12
13
Time (ms)
FILTERED BAEP
UNFILTERED BAEP

From figures such as 4.5 the amplitude ratio of the averaged EP can be estimated to be 2:1. This is an SNR of 6dB. If we now work back assuming that the amplitude ratio improves as the square root of the number of repetitions we see that the SNR of a single EP would be -27dB (amplitude ratio of 1:22). This result is in agreement with the result of other researchers who estimate the averaged EP SNR to be 2.5:1 (8dB) (Fridman et al., 1982).

The assumptions outlined in chapter 2 result in the best possible improvement of SNR. If the assumptions are not strictly true (i.e. jitter of the EP or non-stationary of the EEG) we would expect a larger SNR for the single EP. Work by Coopela (1978) has challenged the long accepted assumption that the BAEP is time-locked to the stimulus. The study concludes that the BAEP exhibits a much larger variation in absolute latency than previously thought. From the perspective of averaging this is a very serious problem. Not only will the averaging process operate sub-optimally but also the resulting averaged EP will be very different from the individual EP. We are therefore faced with the very real possibility that the individual EPs may contain a great deal more information than the currently employed averaged signal. For ANC a variable time-delay between the stimulus and BAEP poses no problem whatsoever.

With the effect of jitter , the non-stationarity of the EEG and our experiments indicating the presence of a time locked EEG component at about 450Hz, we can expect the SNR of a single BAEP to be better than -25dB. To cover a appropriate range the simulations will use single stimulus BAEP SNRs of 0 to -25dB.

4.3.4 ARMA Modelling of EEG

EEG has been successfully modelled as an autoregressive process by a number of authors (Kaveh et al., 1978 & Rauner et al., 1983). The autoregressive process may be represented by:

$$y_t = \frac{1}{\phi_r(z^{-1}) \cdot U_t} \qquad (4.1)$$

where: U_t is the gaussian white noise input
r is the model order

More discussion of the AR process is presented Chapter 5.

The order required to model EEG is reported to be between 8 and 10 (Madhavan, 1985). Modelling of EEG recorded in the laboratory using a Box and Jenkins package modified to run on the PDP-11 was successful using 8th order. The Akaike

information criterion along with inspection of the residual were used to determine adequacy of the model. Table 4.3 shows a representative list of the autoregressive coefficients produced in the laboratory.

Table 4.3

Coefficient	Value
ϕ_1	-0.8921
ϕ_2	0.1762
ϕ_3	-0.2142
ϕ_4	0.0823
ϕ_5	-0.0771
ϕ_6	-0.0136
ϕ_7	-0.1565
ϕ_8	0.2643

In figure 4.8 the collected EEG and its associated modelled AR8 time-series are shown. The signals are similar in appearance and have similar spectra. In all cases tested the EEG records indicated a non-stationary nature by failing the run-time test. However since the EEGs were modelled successfully we may assume that the non-stationarity is not severe. Cohen's 1977 paper reviews the idea of stationarity and presents a technique for determining the stationarity of EEG signals (Cohen & Sances, 1977). Further work on the modelling of non-stationary signals can be found in Greiner's work (1983).

The autoregressive nature of EEG will allow the ANC to

FIG. 4.8a

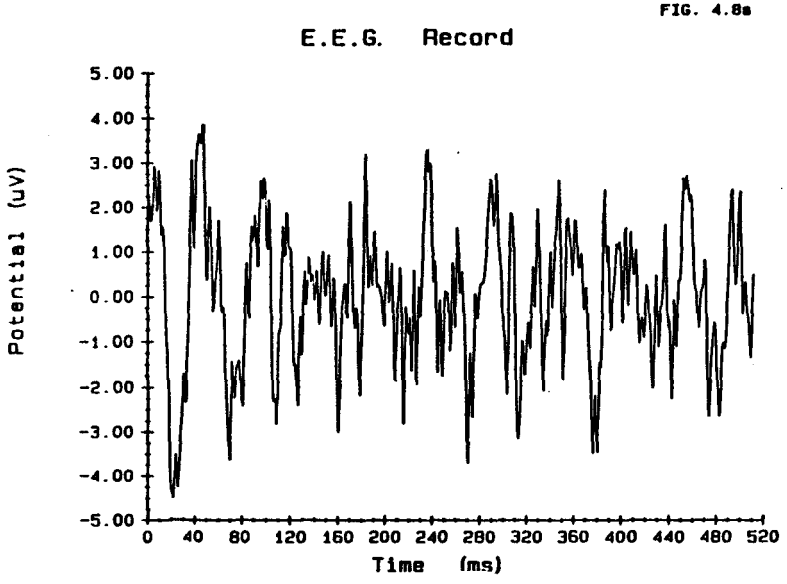


FIG. 4.8b

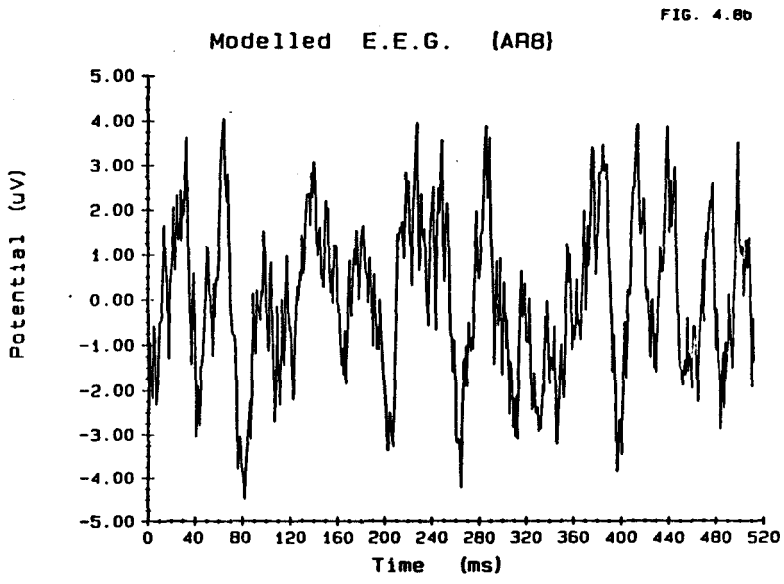


FIG. 4.8c

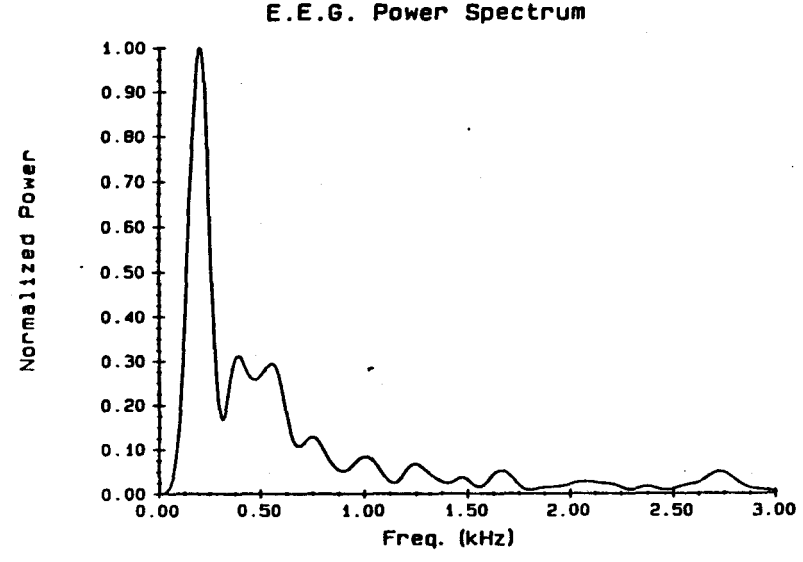
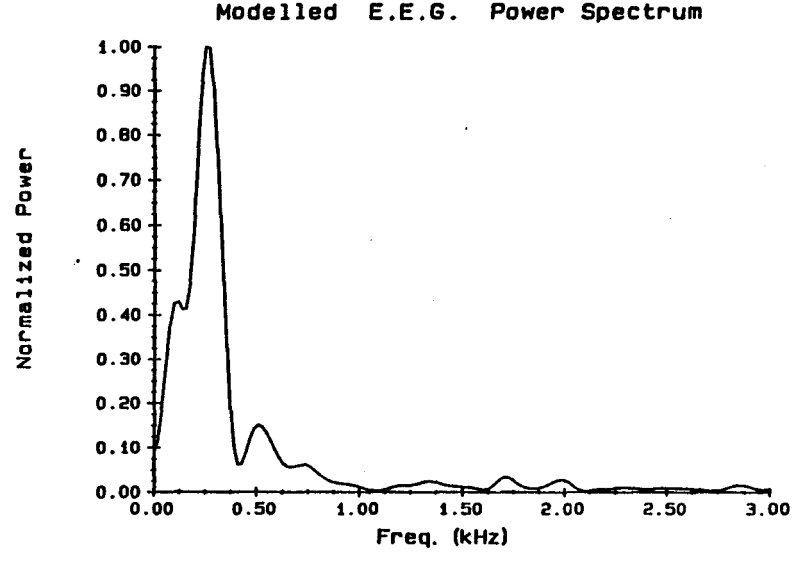


FIG. 4.8d



estimate the signal parameters. We should therefore be able to predict or estimate array orders for simulations and real data experiments.

4.4 Summary of Real Data Analysis

For the purposes of simulation the following have been determined :

1) The averaged EP may be tightly band-passed filtered (600-1100 Hz) and still retain its essential components.
2) EEG may be modelled as an autoregressive process of order 8.
3) The EEG exhibits a slightly non-stationary character.
4) The worst case SNR should be -25dB. Due to invalid assumptions we expect the SNR to be greater than this (perhaps up to -15dB).

We now begin studies of systolic adaptive noise cancellation process applied to BAEP based on the theory of chapter 3 and the signal characteristics of the BAEP and EEG discussed in this chapter.

CHAPTER 5

ADAPTIVE NOISE CANCELLATION USING SYSTOLIC ARRAYS

5.1 Introduction

In order to successfully apply adaptive noise cancellation techniques to our physiological data appropriate primary and reference signals (as discussed in Chapter 4) must be chosen. As a first approximation we look to the results of Kaveh (1978) and Madhavan (1985) which indicate that a satisfactory degree of correlation may exist between EEG prior to stimulation and EEG during the evoked response. One of the primary objectives in this study is to evaluate the appropriateness of this choices of reference and primary signals. Other possible choice of reference exist such as as multi-channel recording, which would provide a temporally parallel noise signal.

Before experimenting with real data it was necessary to examine the characteristics of the systolic adaptive noise canceller (SANC). Using parameter values (e.g. SNR) and information from Chapter 4 relevant to the BAEP extraction

problem experiments can be performed to determine the optimum filter parameters (memory factor (λ), order etc.). We began with general simulations and worked towards a close approximation of the real data scenario. Once sufficient knowledge was acquired the real BAEP data taken in the laboratory was processed and the results interpreted.

The major advantage of the ANC procedure when applied to BAEP processing is that many of the assumptions necessary for justification of other less powerful signal processing techniques can be dropped . Variable latency of the BAEP with respect to the stimulus is no longer a problem. Non-stationarity of the EEG as reported in Chapter 4 can be taken into account by the memory factor λ. The only conditions required are:

1) The BAEP and EEG must be additive in nature.

2) The BAEP and EEG are uncorrelated. (The BAEP may be the sum of individual components. It may be stationary or non-stationary, linear or non-linear and will not affect validity of the processing at all.

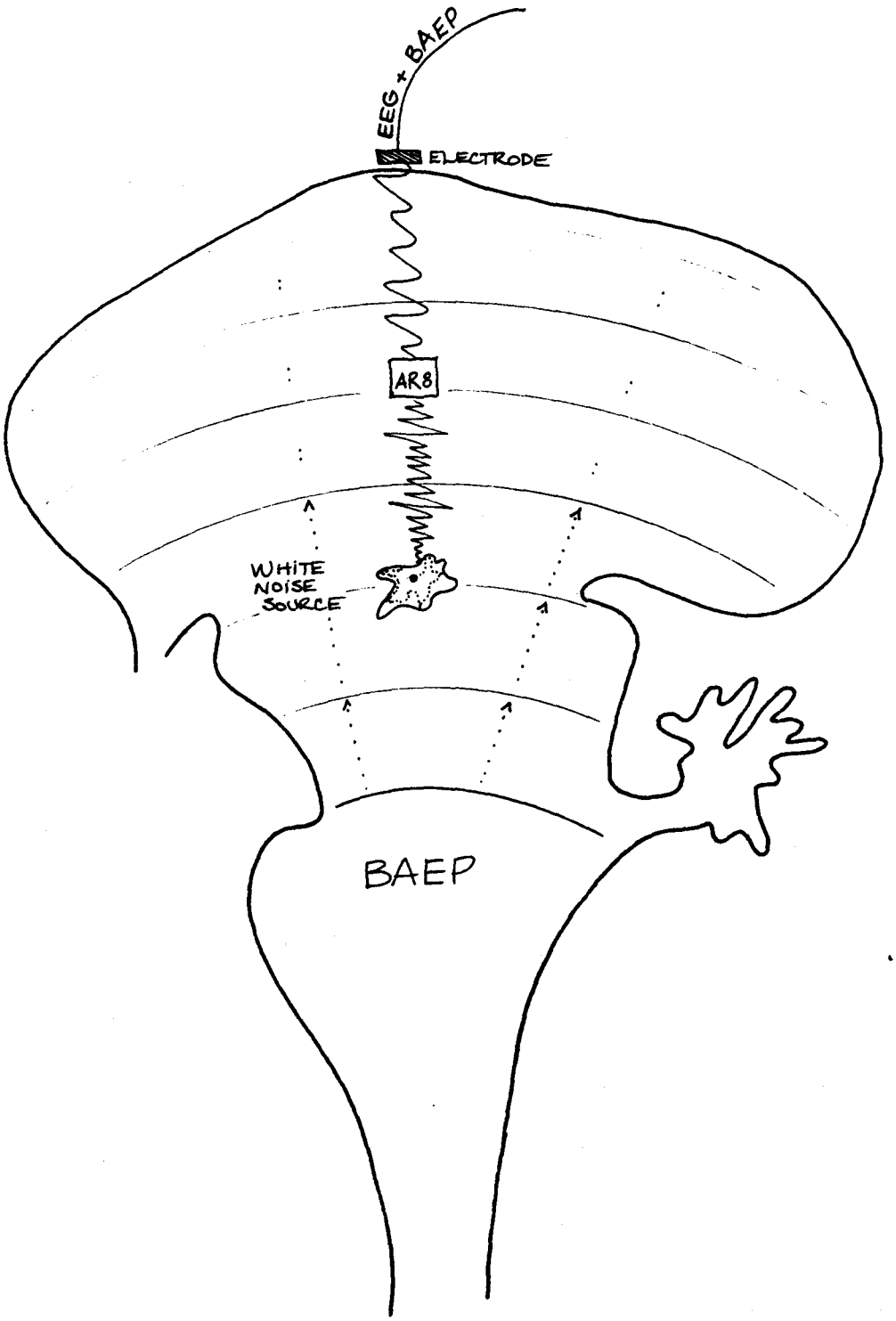
3) It must be possible to find a reference signal which is correlated to the EEG and uncorrelated to the BAEP of the primary signal.

In figure 5.1, cortex potential is modelled as a sum of the BAEP and a white noise which has passed through a spatial autoregressive type filter. The BAEP, due to its very short latency, is thought to be volume conducted signal (Madhavan, 1985). With this model it can be expected that the additive requirement will be met. There are assumptions made about the structure of the BAEP. Current research suggests that the BAEP may be the superposition of several components each one contributing one of the peaks (Hunt, 1985).

The primary concern with our model lies with the choice of a reference signal. Adequate correlation must exist between the reference and primary EEG. The use of pre-stimulus EEG as a reference is based on the results of Kaveh (1978) and Madhavan (1985) who have reported some success.

Correlation between EEG and certain BAEP components has been reported in studies (Rogers, 1980). This is not surprising since the neural source of the BAEP may be

fig. 5.1. PHYSIOLOGICAL MODEL OF EEG & BAEP AT THE SCALP



represented in normal spontaneous EEG during the normal functioning of the cerebrum. It is almost certain, therefore, that auditory stimulation affects the EEG inducing correlation between the EEG and BAEP. However, since the neuronal population responding to the stimulus is such a small percentage of the total cerebral population , we assume that the correlation is minimal. Roger (1980) has reported three factors affecting correlation:

1) Frequency of the EEG.

2) Contralateral or Ipsilateral measurement montage.

3) Latency of the BAEP

Roger reports correlation between BAEP components of 100-150 ms latency and alpha-type EEG. In our studies the components are much earlier (< 10ms) and the frequency of the EEG used is greater than 150 Hz (alpha rhythms are between 4-8 Hz). Correlation studies involving EEG and BAEP in the bandwidth and latency would be of great value to future studies. Further ANC studies using visual and somatosensory (EPs which have longer latencies and much lower frequencies) will be much more sensitive to correlation effects.

5.2 Simulation Techniques

The report of our experiments begins with the detailing of the procedure used in the simulations. As a first approximation to the real case we use a band-pass filtered averaged BAEP signal (fig. 5.2) added to white Gaussian noise. The reference generated is an autoregressive signal of order 2. The coefficients for this signal are taken from Madhavan (1985) and the resulting equation is:

$$u(i) = 0.4\ u(i-1) - 0.043\ u(i-2) + w(i) \qquad (5.1)$$

where 'w' is a gaussian white noise sequence.

Therefore: $\phi_1 = 0.4$

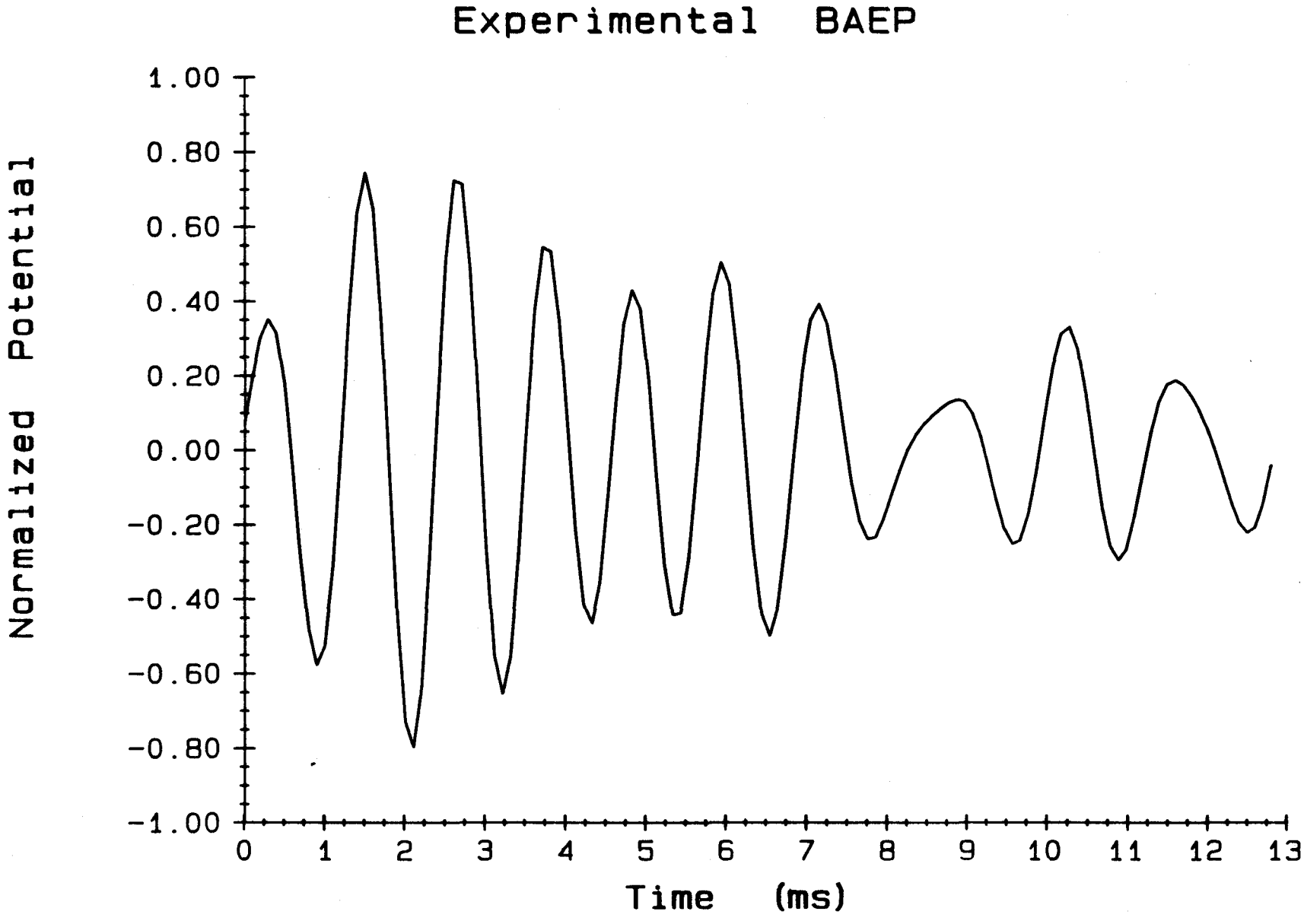
$\phi_2 = -0.043$

which has poles at : $0.2 \pm 0.055i$

with a magnitude of: 0.207 on the unit circle

The choice of a very simple scenario eliminates some of the complexities associated with real EEGs. We diverge now to discuss the implications of the filter structure of the systolic array and its relation to the simulation design.

FIG. 5.2


Experimental BAEP
Normalized Potential
1.00
0.80
0.60
0.40
0.20
0.00
-0.20
-0.40
-0.60
-0.80
-1.00
0
1
2
3
4
5
6
7
8
9
10
11
12
13
Time (ms)

The systolic array (and most other filter structures) have finite impulse responses (FIR). In other words there is no feedback capability. The other class of filters, the infinite impulse response (IIR), do have feedback and therefore tend to be less stable than their FIR counterparts.

The **generation** of an AR process is an IIR process (a quick look at equation 5.1 will confirm this), in other words feedback is required. The output of the filter is returned to the input side for further processing. Hence the infinite impulse name is derived from the result that a single impulse is filtered and a fractional part is returned to the input only to be returned fractionally yet again.

In our SANC scheme the primary signal is white and the reference (AR2) must be **synthesized** into its original white source signal. Since generation of an AR process is IIR in nature it follows that synthesis of an AR process is FIR. This can be seen by rearranging equation 5.1 and noting that we are inputting the u(i)s and looking for w(i).

$$u(i) - 0.4\,u(i-1) + 0.043\,u(i-2) = w(i) \qquad (5.2)$$

By using such primary and reference signals the systolic array develops the least squares approximation to the ϕs of equation 5.2 as its tap weights. There are a finite number of weights and therefore a finite filter order is expected to be optimal (in a theoretical sense). Therefore our choice of simulation suits our filter structure. It should be noted that the opposite may be said of a moving average signal in that generation is FIR and synthesis is IIR.

In almost all cases we are presented with the problem of synthesizing a signal (generation is a trivial problem). This may explain why AR sequences are so much favoured over MA signal in modern signal processing literature.

Before returning to our description of the simulation procedure it should be noted that in the real case both inputs to the SANC are autoregressive in nature (pre and post-stimulus EEG are the reference and primary respectively). In this case the FIR filter will be attempting to perform an IIR function. To gain an understanding of the processing involved we imagine the transfer function of an IIR process as the reciprocal of a polynomial. If this polynomial is represented as a numerator rather than a denominator the result is a

transfer function with an infinite number of coefficients. Therefore, under such conditions we would expect the filter to use a very large number of tap weights.

The sequence of figures which follow illustrates the protocol used to examine the SANC process. First a Gaussian white noise sequence is generated (fig 5.3). Then the sequence is passed through the AR filter (fig 5.4). The resulting AR sequence is identified by its order and pole magnitude (i.e. AR2207). The band-passed BAEP is added at some SNR to the white noise to create the primary signal (fig 5.5). Now the 128 point BAEP is buried in the 256 point white noise sequence starting at point 129. In this case 0dB is used for simplicity. The systolic array then processes the signal and produces an approximation to the averaged BAEP (fig 5.6).

For a quantitative measure of performance we use zero-mean, unit variance signals and compare the input BAEP and the output error signal (fig 5.7). The mathematical formulation used to calculate the performance is given in equation 5.3. Equation 5.3 is basically a measure of the percentage fit of two signals. We therefore refer to the performance as percentage fit (Γ). Thus the best fit is

FIG. 5.3

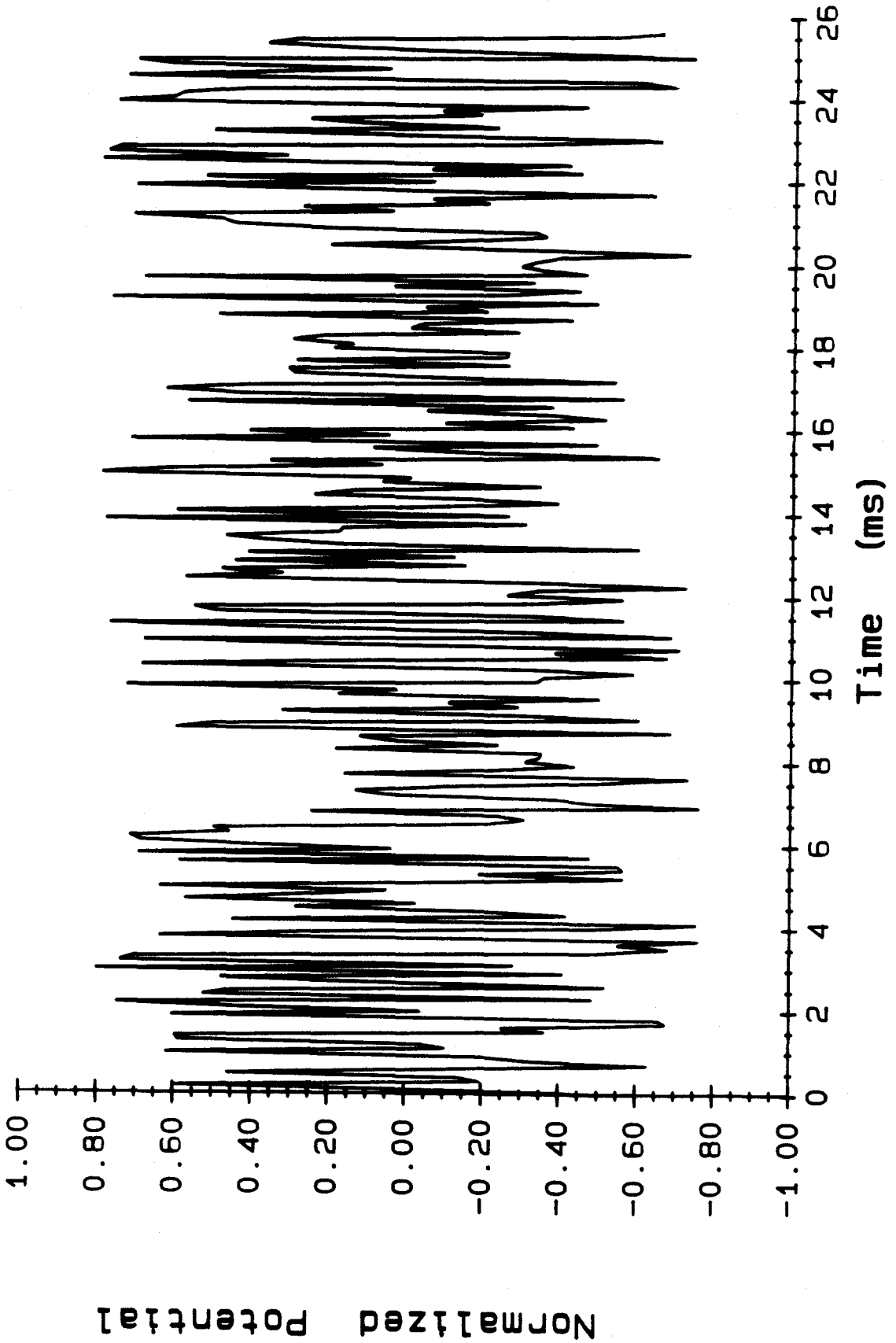

White Noise
Normalized Potential
1.00
0.80
0.60
0.40
0.20
0.00
-0.20
-0.40
-0.60
-0.80
-1.00
0
2
4
6
8
10
12
14
16
18
20
22
24
26
Time (ms)

FIG. 5.4

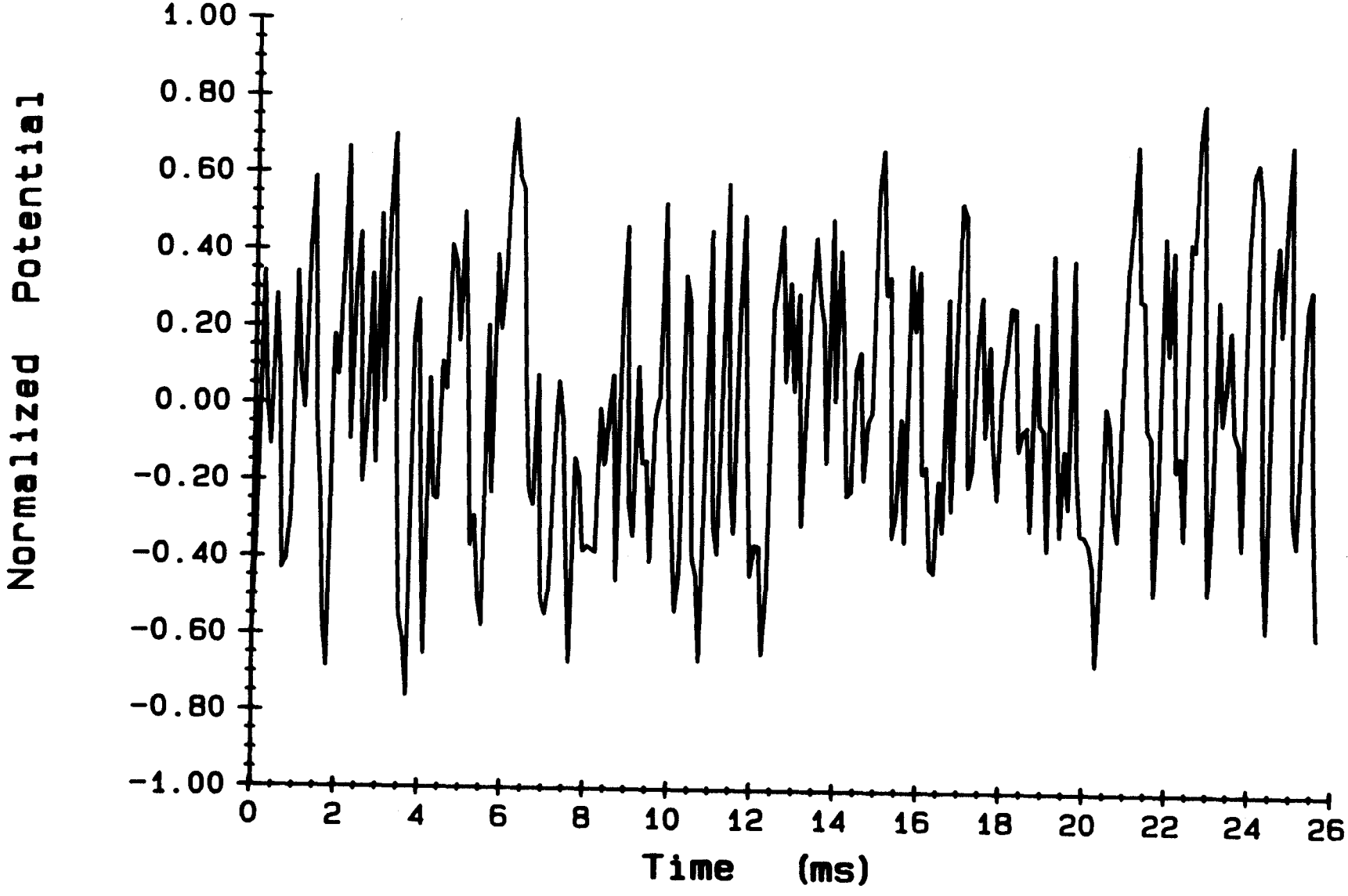

Reference Signal (AR2207)
Normalized Potential
1.00
0.80
0.60
0.40
0.20
0.00
-0.20
-0.40
-0.60
-0.80
-1.00
0
2
4
6
8
10
12
14
16
18
20
22
24
26
Time (ms)

FIG. 5.5

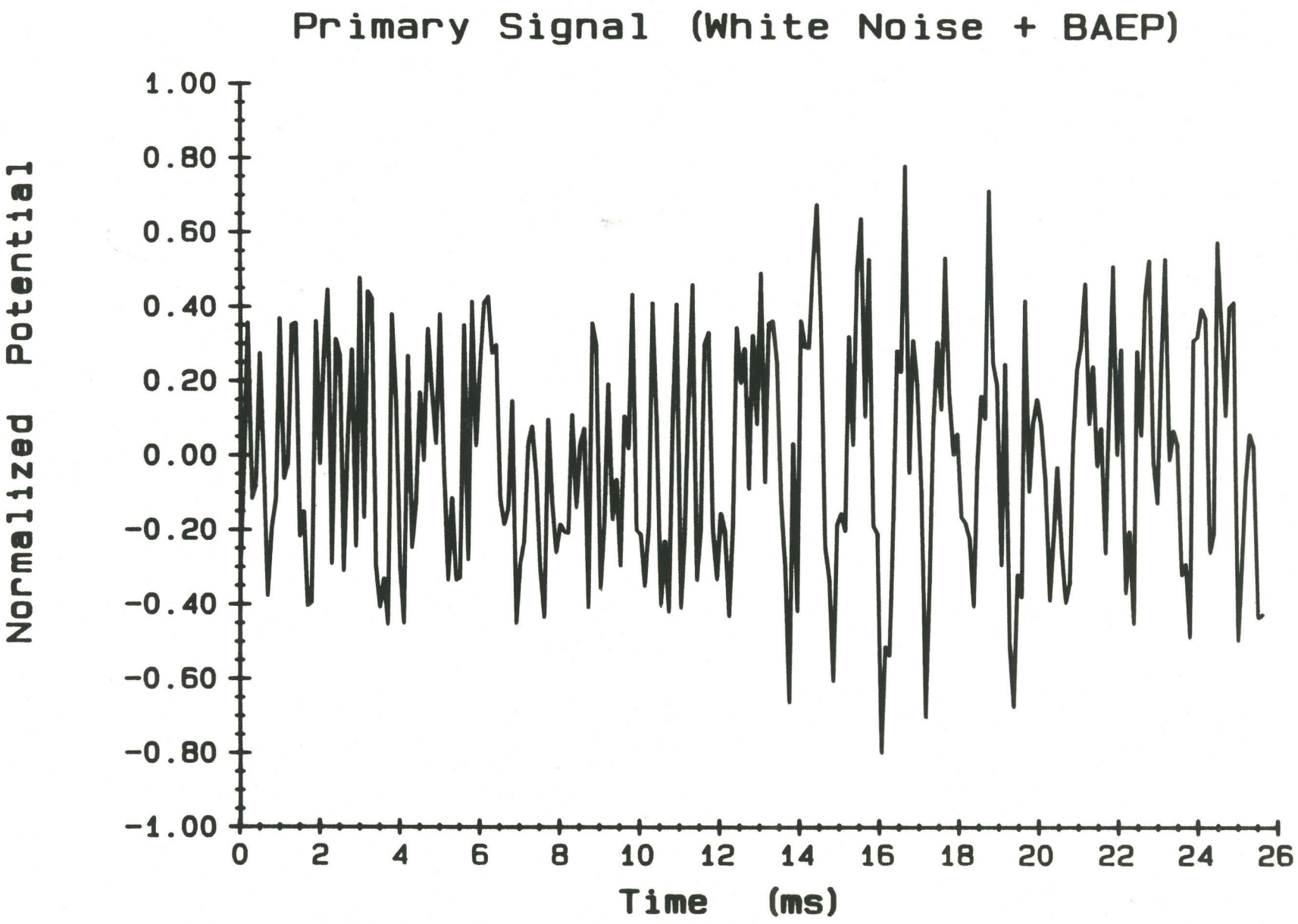

Primary Signal (White Noise + BAEP)
Normalized Potential
1.00
0.80
0.60
0.40
0.20
0.00
-0.20
-0.40
-0.60
-0.80
-1.00
0
2
4
6
8
10
12
14
16
18
20
22
24
26
Time (ms)

FIG. 5.6

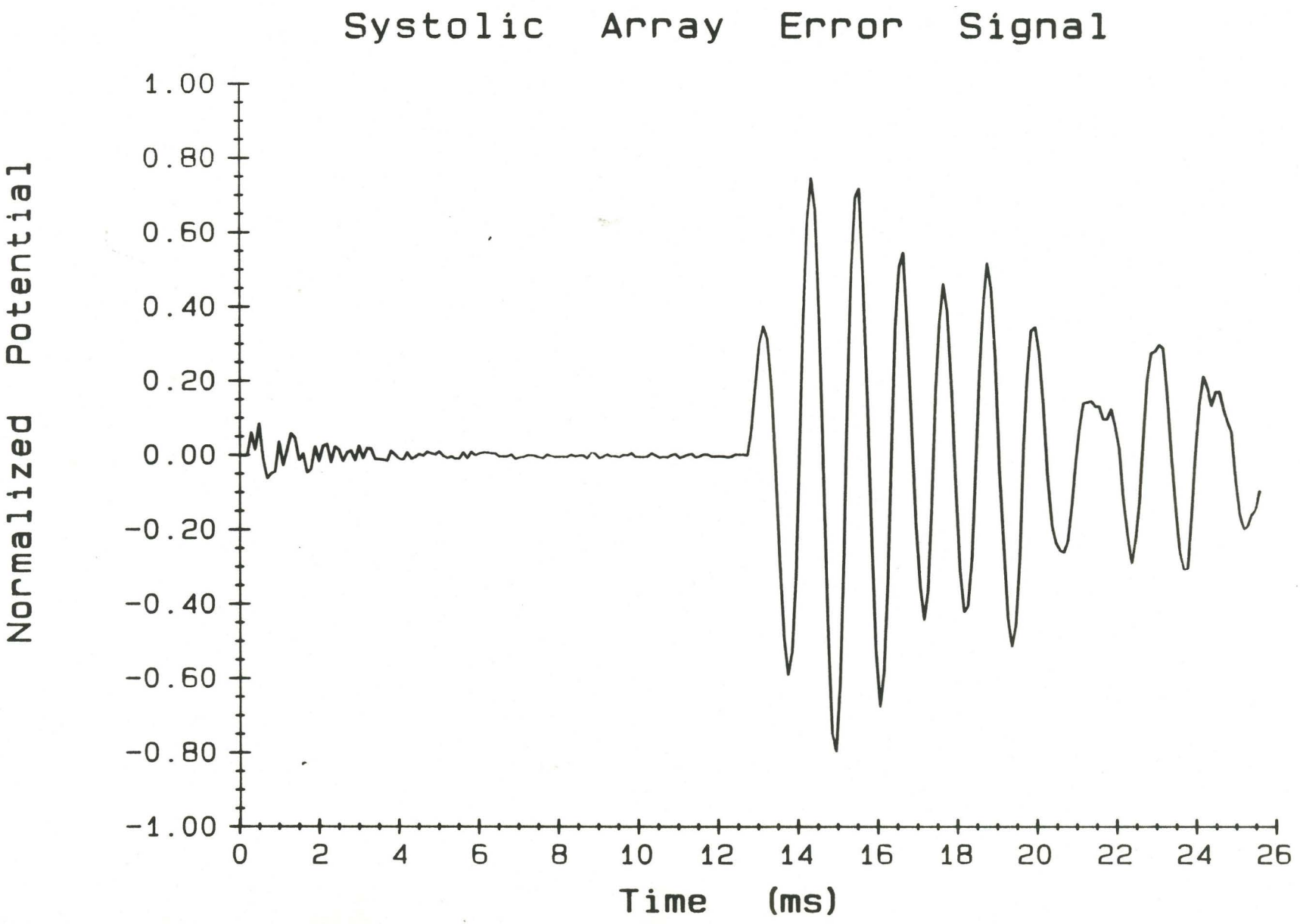
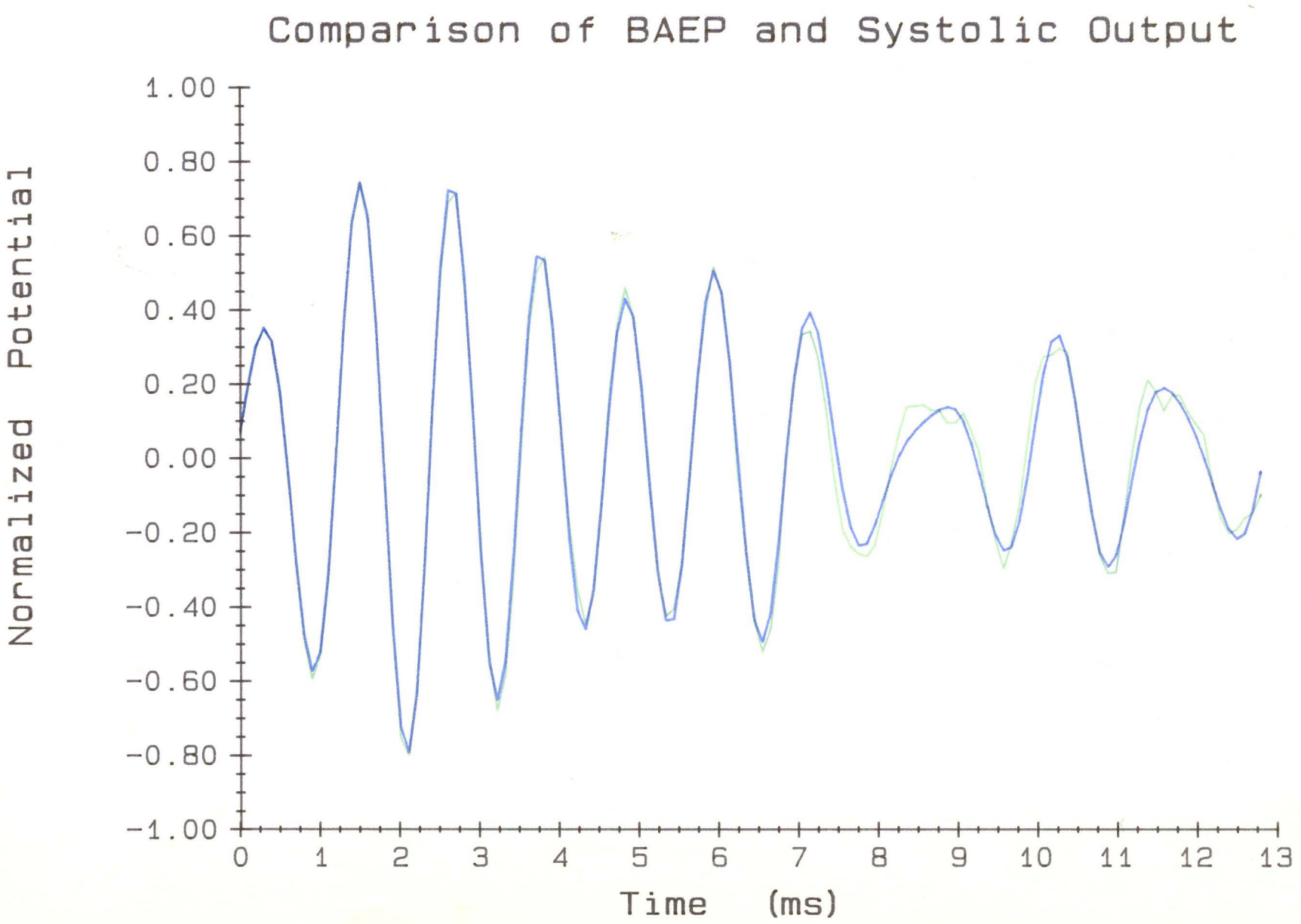

Systolic Array Error Signal
Normalized Potential
1.00
0.80
0.60
0.40
0.20
0.00
-0.20
-0.40
-0.60
-0.80
-1.00
0
2
4
6
8
10
12
14
16
18
20
22
24
26
Time (ms)

FIG. 5.7


Comparison of BAEP and Systolic Output
Normalized Potential
1.00
0.80
0.60
0.40
0.20
0.00
-0.20
-0.40
-0.60
-0.80
-1.00
0
1
2
3
4
5
6
7
8
9
10
11
12
13
Time (ms)

100% when the error signal from the ANC is equal to the input BAEP.

$$\Gamma\ (\%\ \text{fit}) = \left[1 - \frac{\sum_{i=1}^{N} (s_i - \hat{s}_i)^2}{\sum_{i=1}^{N} (s_i)^2}\right] \times 100 \qquad (5.3)$$

where: $\hat{s}$ is the systolic array error signal

s is the input BAEP

s_i is the ith point in the sequence

N is the number of points in the BAEP (usually 128).

The choice of an absolute index rather than an index which measures the improvement of SNR was required so that the effects of SNR in the primary signal could be studied.

The % fit of figure 5.7 is 98.23 with unity lambda and a filter order of 3. Illustrated in figure 5.8 is the extraction of the same signal with the SNR at -25dB. Note that this is the SNR determined in Chapter 4 to be the worst case scenario. The extraction is obviously less successful but is nonetheless impressive with Γ = 95.74%.

In order to approach the real scenario more closely the

FIG. 5.8

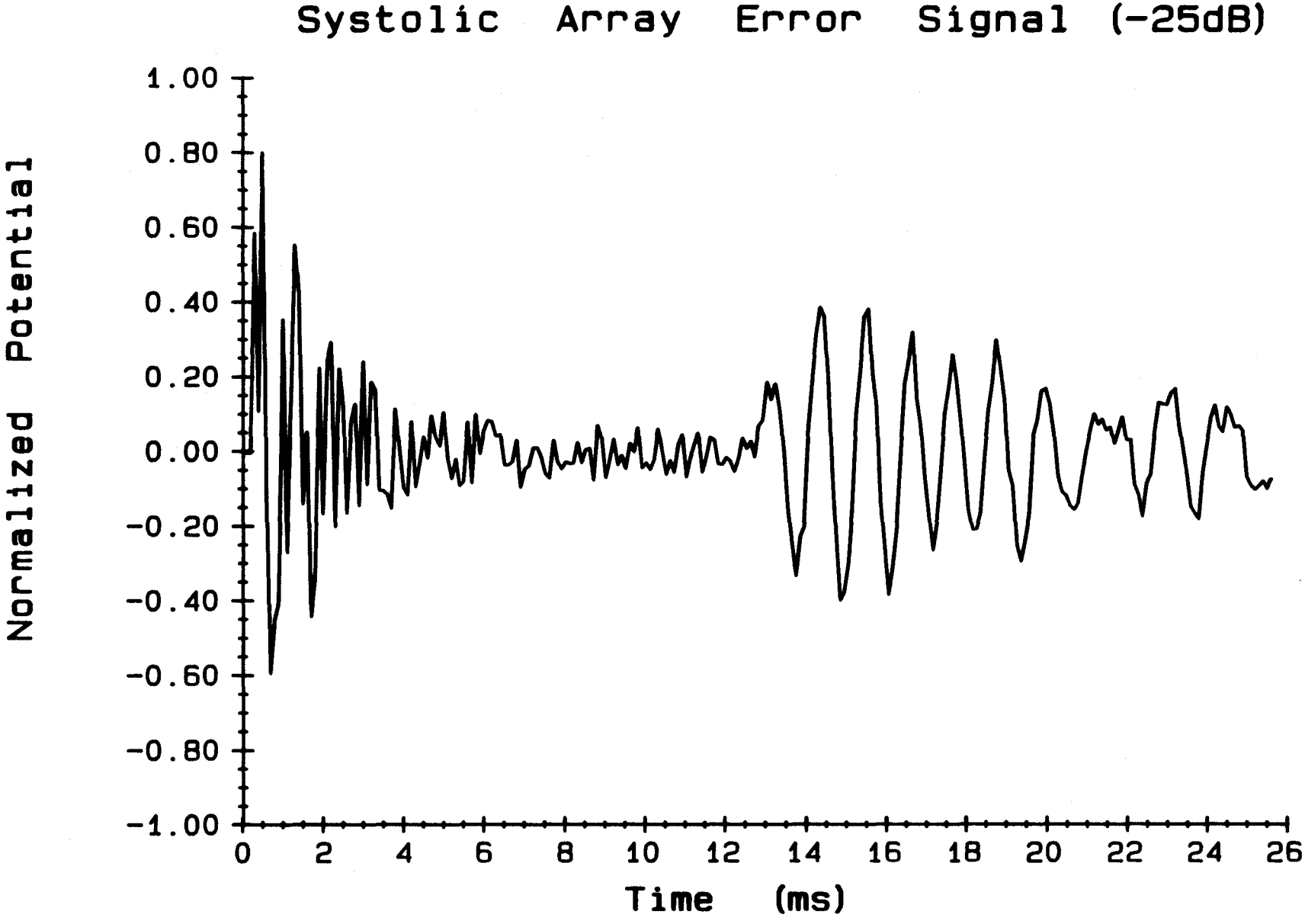

Systolic Array Error Signal (-25dB)
Normalized Potential
1.00
0.80
0.60
0.40
0.20
0.00
-0.20
-0.40
-0.60
-0.80
-1.00
0
2
4
6
8
10
12
14
16
18
20
22
24
26
Time (ms)

FIG. 5.9

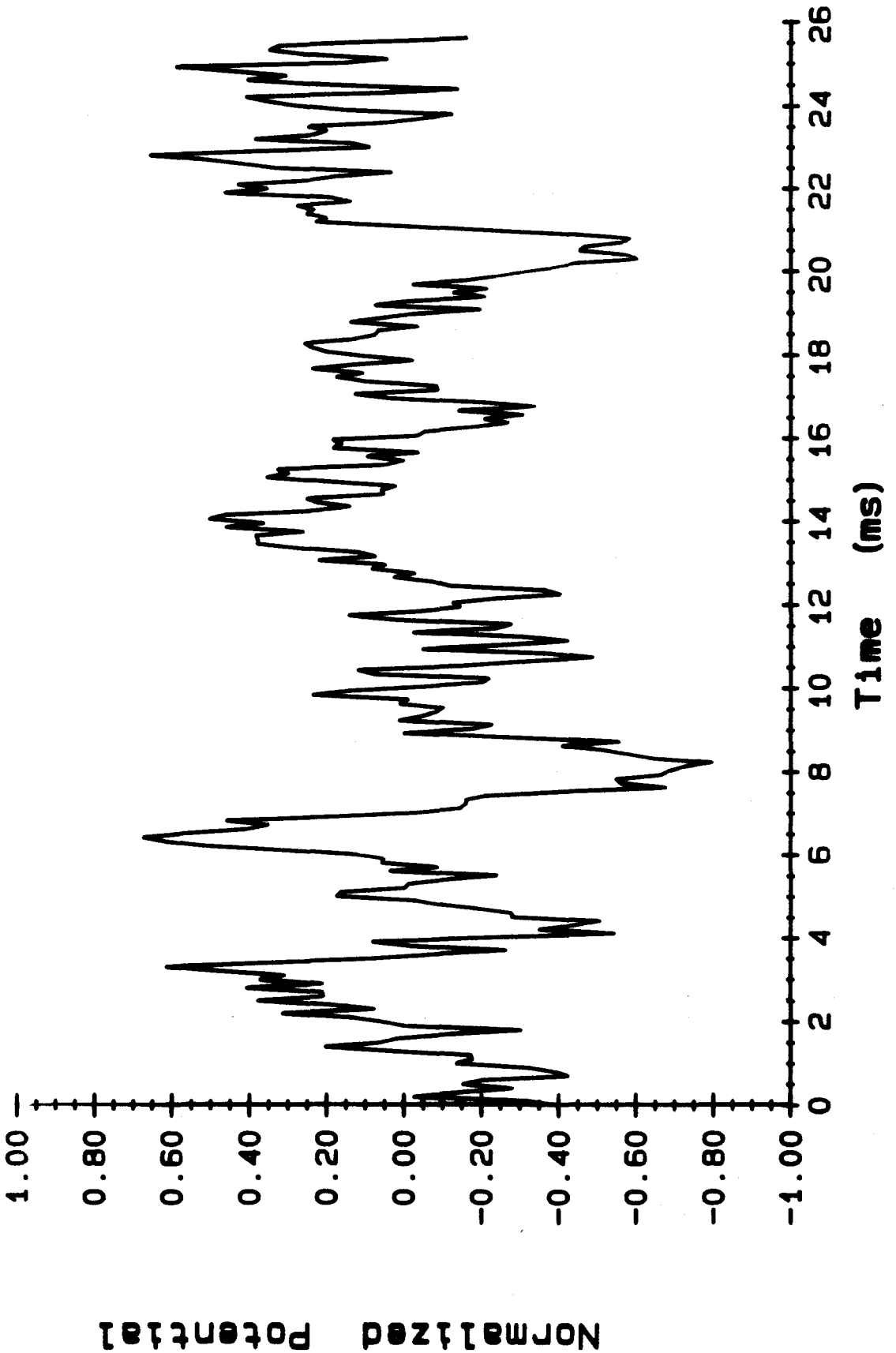

Reference Signal (Modelled EEG : AR8)
Normalized Potential
1.00
0.80
0.60
0.40
0.20
0.00
-0.20
-0.40
-0.60
-0.80
-1.00
0
2
4
6
8
10
12
14
16
18
20
22
24
26
Time (ms)

fig. 5.10 Recovery of BAEP from AR8 signal at -25 dB

FIG. 5.10a

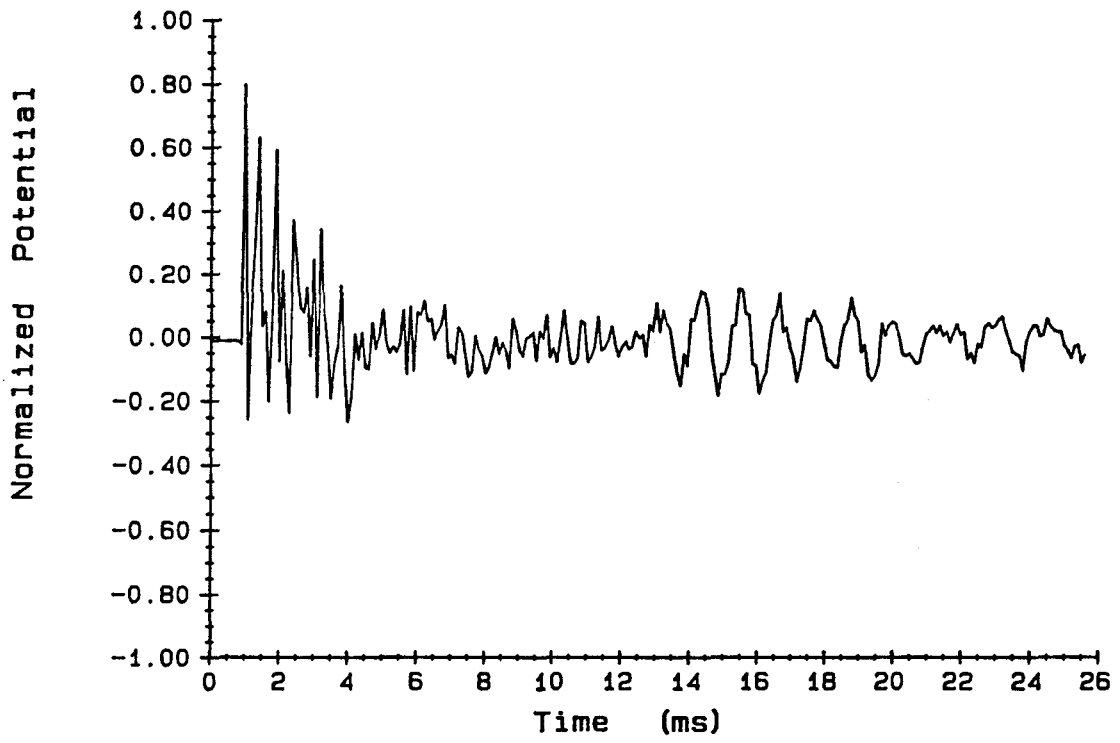
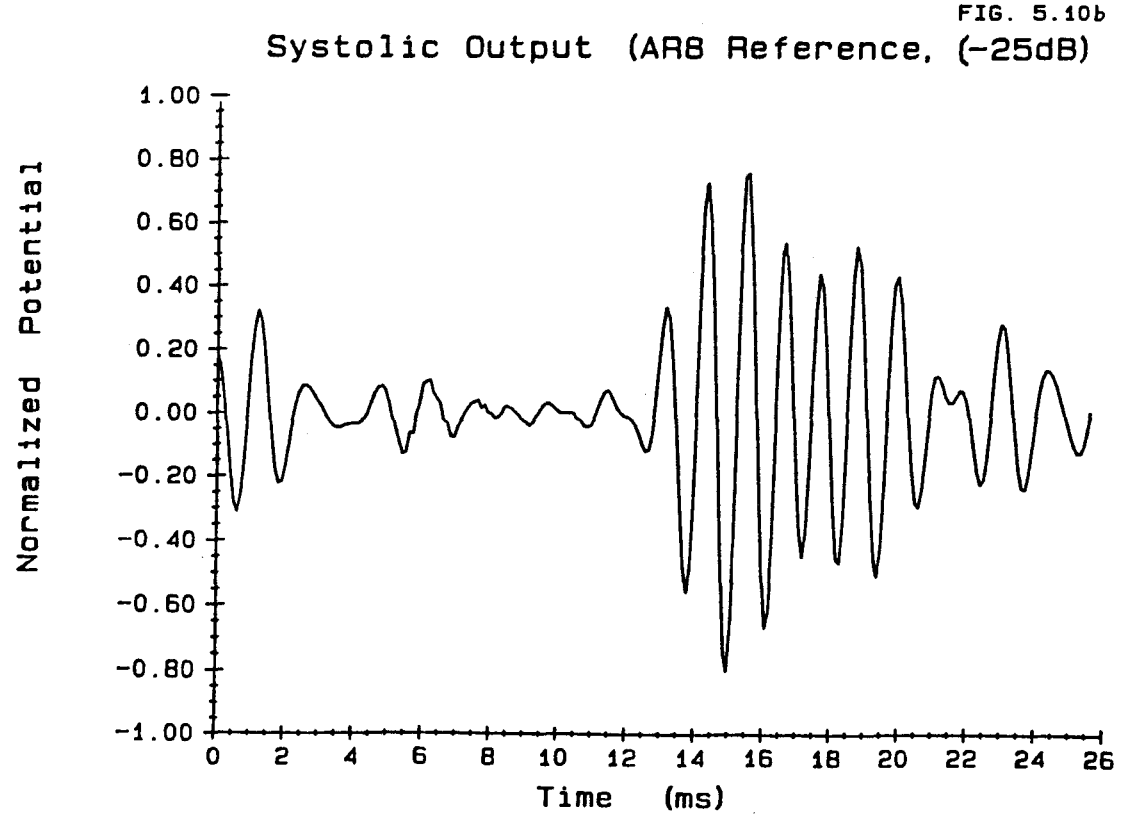


FIG. 5.10b



coefficients of the AR8 (modelled EEG) process are used to generate a second reference signal (fig 5.9). The extraction is shown in figure 5.10a (SNR=-25dB). Γ for this experiment was 84.27% . From this experiment it can be seen that the poorer the SNR and the larger the order of the reference signal the worse the extraction. However results are encouraging and it will be shown that by correct application of filter order, appropriate λ and use of band pass filtering the Γ may be increased to 97.83% (fig. 10b).

5.3 Experimental Results and Discussion

5.3.1 Introduction

In the following series of experiments the performance of the systolic array is investigated as the following parameters are changed:

1) Pole magnitude.
2) Precision of calculation.
3) Number of points in the data sequence.
4) Order of the reference signal.
5) Signal to noise ratio.

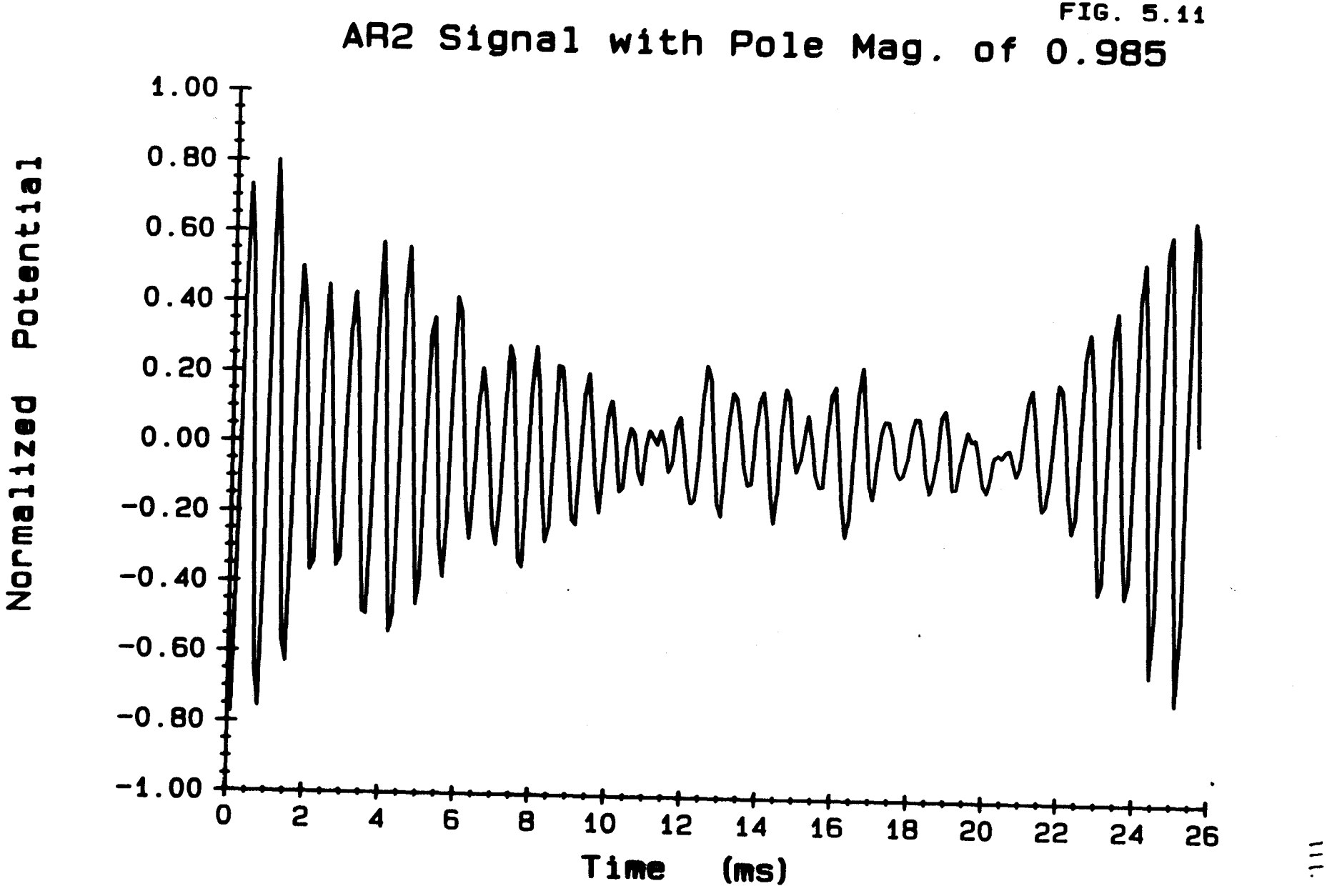
The effect of forcing the systolic array to process a reciprocal polynomial is also examined. In other words the roles of the white noise and autoregressive sequences are switched forcing the systolic array to operate in a generation mode. Finally the real data collected in Chapter 4 is processed and correlation analysis is used to interpret the results.

5.3.2 Synthesis Scenario (White Primary)

Pole position within the unit circle has a profound affect upon the nature of the auto-regressive process. As the pole moves towards the edge of the unit circle the response of the system to an impulse becomes longer. In the limiting case where the magnitude is unity the autoregressive process becomes a rotating exponential and the impulse response is infinite in duration. Outside the unit circle the signal is non-stationary and in the case of real data, poles may move around becoming first stationary and then non-stationary. This phenomena is alpha-stationarity.

Figure 5.11 illustrates an AR2 process with a pole at 0.985 (AR2985). Notice that, in comparison to AR2207, the

FIG. 5.11


AR2 Signal with Pole Mag. of 0.985
Normalized Potential
1.00
0.80
0.60
0.40
0.20
0.00
-0.20
-0.40
-0.60
-0.80
-1.00
0
2
4
6
8
10
12
14
16
18
20
22
24
26
Time (ms)

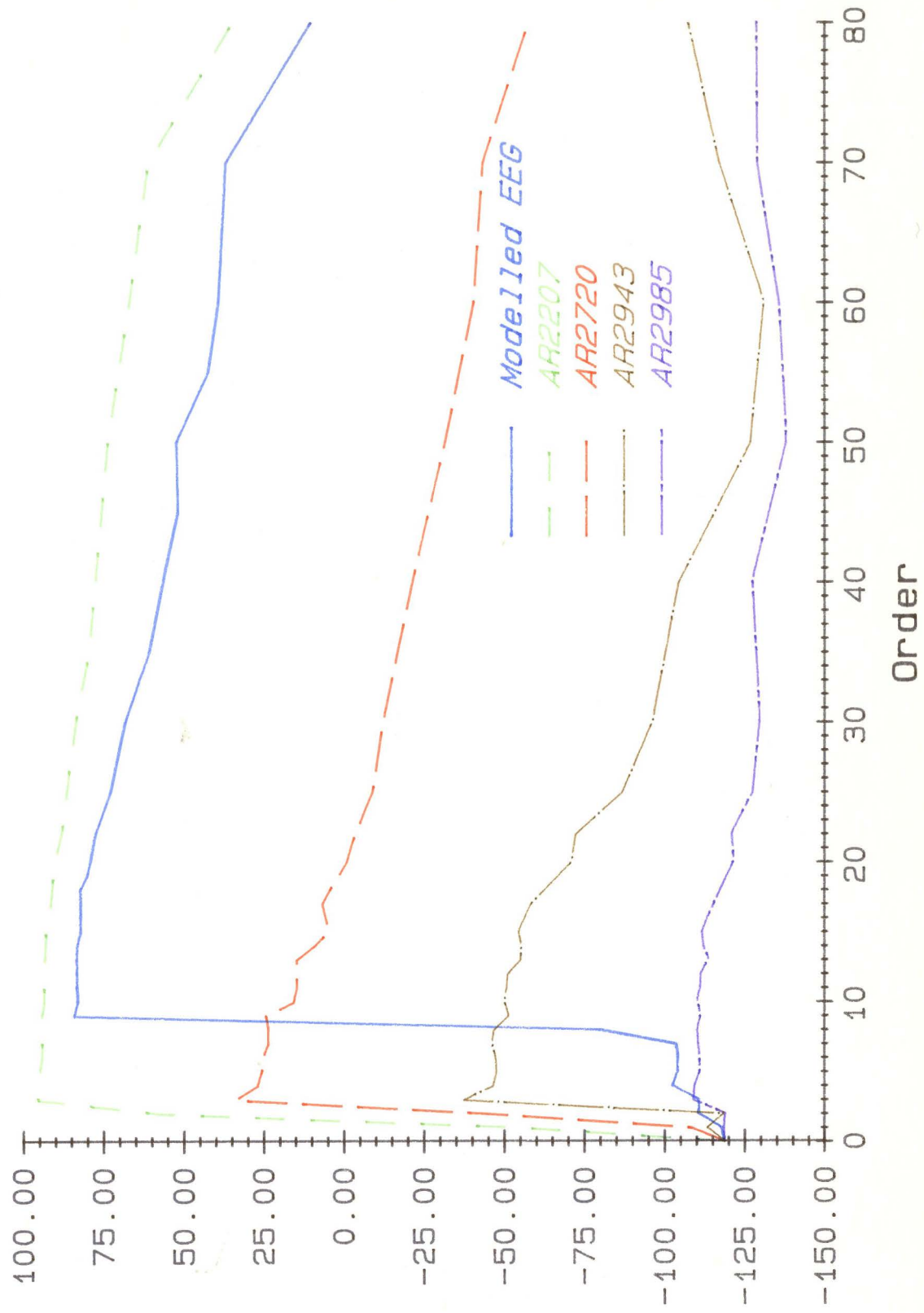
signal is much less like its white noise input and resembles much more closely a sinusoid. In figure 5.12 the extraction performance of the systolic array as a function of filter order is compared for a variety of signals (AR8 model presented in Chapter 4 plus signals results from different pole positions). The extraction (SNR -25dB, λ = 1.0) is most successful when the pole is near the centre of the unit circle. Almost no cancellation takes place for AR2985. Fortunately the noise cancellation of the model EEG is quite successful despite one of its poles being quite close to the unit circle (0.895). This is probably due to the distribution of the three remaining pairs of symmetrically placed poles.

The memory factor lambda (λ) and the filter order (Ω) are two fundamental parameters that are of interest when running the systolic array. Results relating to the optimum order (measured by eqn. 5.3) from figure 5.12 and other experiments are summarized in table 5.1.

Table 5.1

Primary	Reference	Ω_{opt}
white	white (AR0)	1
white	AR1	2
white	AR2	3
white	AR8	9

FIG. 5.12


Adaptive Noise Cancellation of BAEP
Extraction
100.00
75.00
50.00
25.00
0.00
-25.00
-50.00
-75.00
-100.00
-125.00
-150.00
0
10
20
30
40
50
60
70
80
Order
Modelled EEG
AR2207
AR2720
AR2943
AR2985

The results of table 5.1 can be summarized as:

$$\Omega_{opt} = \text{AR order} + 1 \qquad (5.3)$$

This is valid for the synthesis scenario only and can be justified in terms of the prediction error filter. Prediction filters produce the least squares estimate for the next value in a sequence. They use one extra tap to subtract this prediction from the next input thus producing a prediction error. In the systolic array the extra tap-weight required comes from the final column which carries the primary signal and produces the error signal. Since synthesis of the AR signal through the triangular section requires the same number of tap-weights as coefficients in the AR process the total order of the filter is therefore the AR order plus one. Experiments (in addition to those shown in figures 5.12 and 5.13) indicate that the optimum order, Ω_{opt}, **is independent of both the SNR and the lambda chosen** within the ranges tested (0 to -100dB and 1.1 to 0.75 respectively).

Lambda, the weighting function described by McWhirter (1983), may be used to compensate for non-stationarities resulting from finite data sequence representation of real

signals. The autoregressive signals generated for these simulations are examples of such finite sequences. By finite representation of infinite signals non-stationarities are introduced. The differences in extraction rate of the four AR2 sequences tested may be explained in terms of sampling. Since the AR2985 has poles near the unit circle its impulse response is very long compared to AR2207. By using the same number of points to represent each signal there will be a relatively poorer representation of the AR2985 signal (due to the longer length of its impulse responses). This can be compensated for by using $\lambda<1$. In fig. 5.13 (AR2985 signal from fig 5.12) the effect of lambda on extraction is profound. The use of $\lambda = 0.94$ increases the extraction, Γ, to 87.3%. The order and lambda optimization problems are clearly decoupled. Table 5.2 lists λ_{opt} for each AR2 signal.

Table 5.2

Signal	λ_{opt}
AR2207	1.000
AR2743	0.993
AR2943	0.983
AR2985	0.945

The interpretation of the role of lambda may be thought of in terms of a memory factor. By weighting the past values

FIG. 5.13

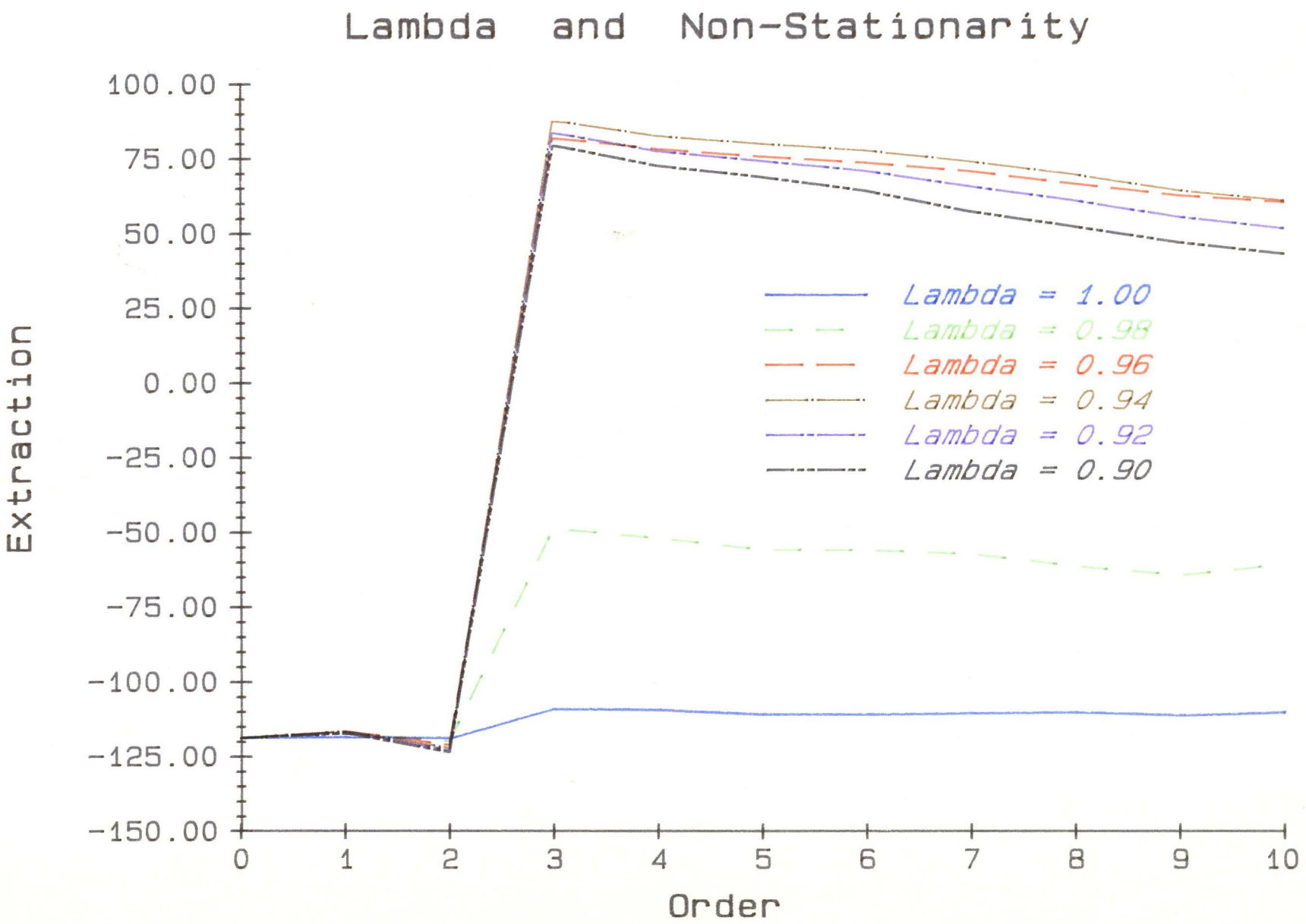
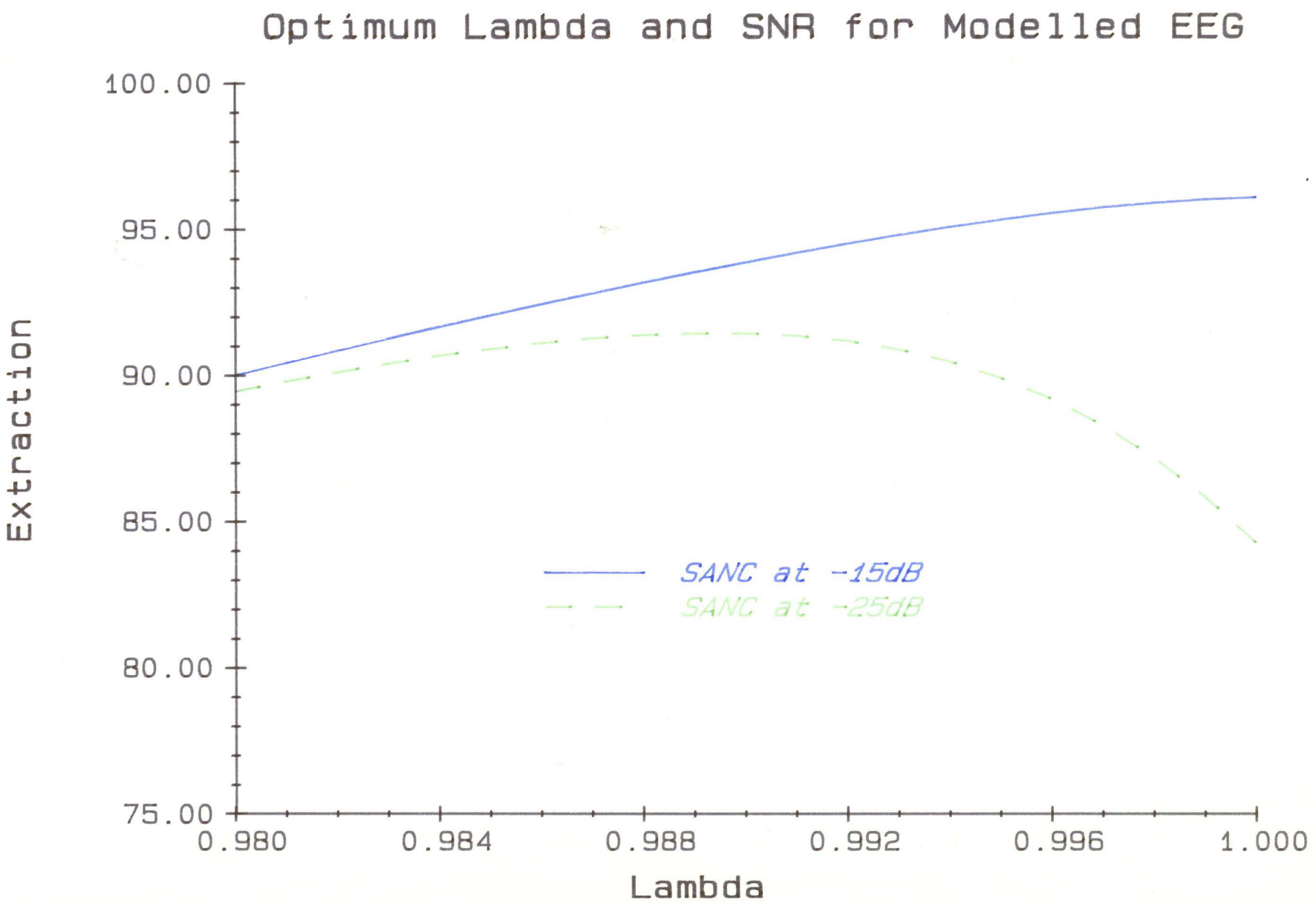

Lambda and Non-Stationarity
Extraction
100.00
75.00
50.00
25.00
0.00
-25.00
-50.00
-75.00
-100.00
-125.00
-150.00
0
1
2
3
4
5
6
7
8
9
10
Order
Lambda = 1.00
Lambda = 0.98
Lambda = 0.96
Lambda = 0.94
Lambda = 0.92
Lambda = 0.90

FIG. 5.14

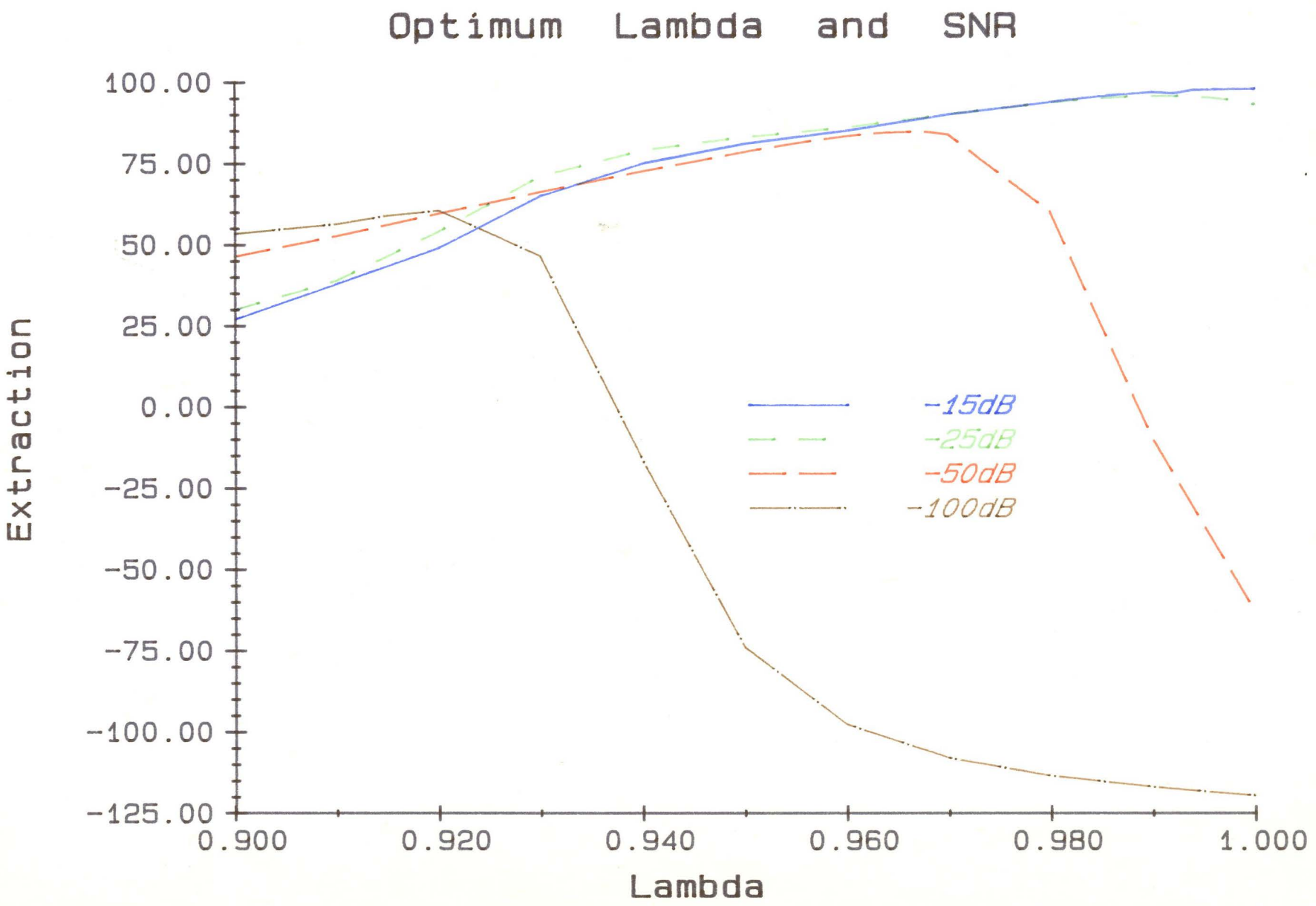

Optimum Lambda and SNR for Modelled EEG
100.00
95.00
90.00
85.00
80.00
75.00
Extraction
SANC at -15dB
SANC at -25dB
0.980
0.984
0.988
0.992
0.996
1.000
Lambda

of the input less and less the systolic array damps out the long impulse responses and increases the importance of the new information resulting from more recent impulses.

The choice of lambda is also affected by the SNR of the primary signal. The poorer the SNR the more λ_{opt} will deviate from unity. This effect is quite small when using the systolic array as can be seen in figure 5.14 (filtered at Ω_{opt}). Using the AR8 signal with an SNR of -15dB and -25dB the optimal lambdas, λ_{opt}, are 1.000 and 0.990 respectively. The plot of -25dB is markedly asymmetric with respect to lambda. Extraction suffers less for an underestimated lambda than an overestimated one. The effect of SNR is smaller when optimal lambda and order are used. In this case the difference between Rs for a change in SNR of 10dB is only 4% when the optimal lambdas and orders are used. Due to our AR8 model being reasonably stationary the functioning of lambda is not critical since use of λ=1 in the -25dB case would only result in a decrease in extraction of 5%.

A larger range of SNRs is explored in figure 5.15 (AR8 at Ω_{opt}). As the SNR decreases the extraction falls and the optimum lambda deviates further from unity. Even though the extraction at -100 dB reaches maximum of only 58.2%

FIG. 5.15


Optimum Lambda and SNR
100.00
75.00
50.00
25.00
0.00
-25.00
-50.00
-75.00
-100.00
-125.00
Extraction
0.900
0.920
0.940
0.960
0.980
1.000
Lambda
-15dB
-25dB
-50dB
-100dB

the amplitude ratio at this SNR is 1:100,000.

The explanation of the interaction between SNR and lambda (as SNR drops λ_{opt} drops) may be a result of undesirable effects associated with the use of $\lambda < 1$. The lower λ_{opt} the shorter the memory of the array and the quicker the adaptation and the more likely the array may begin to cancel the desired signal (BAEP). The array is more likely to cancel the BAEP at larger SNR because at this noise level the BAEP constitutes a large portion of the primary signal. **Thus the optimum λ will be a compromise between the cancellation of the noise (aided by a lower lambda) and the cancellation of the buried signal (also aided by a lower lambda).** However the cancellation of the buried signal will be aggravated by a larger SNR thus at higher SNR the λ_{opt} will tend to be closer to unity. Obviously the correlation between the reference and the components for the primary will play a large role in the final determination of λ_{opt}.

Comparisons of the performance of the systolic array and the exact least-squares lattice (LSL) are made in figures 5.16 and 5.17 (Ω_{opt} of 3). In figure 5.16 the AR2207 signal is used at -15dB and -25dB. **The systolic array performance is superior under all conditions tested. This** remark applies, without exception, to all experiments where

FIG. 5.16

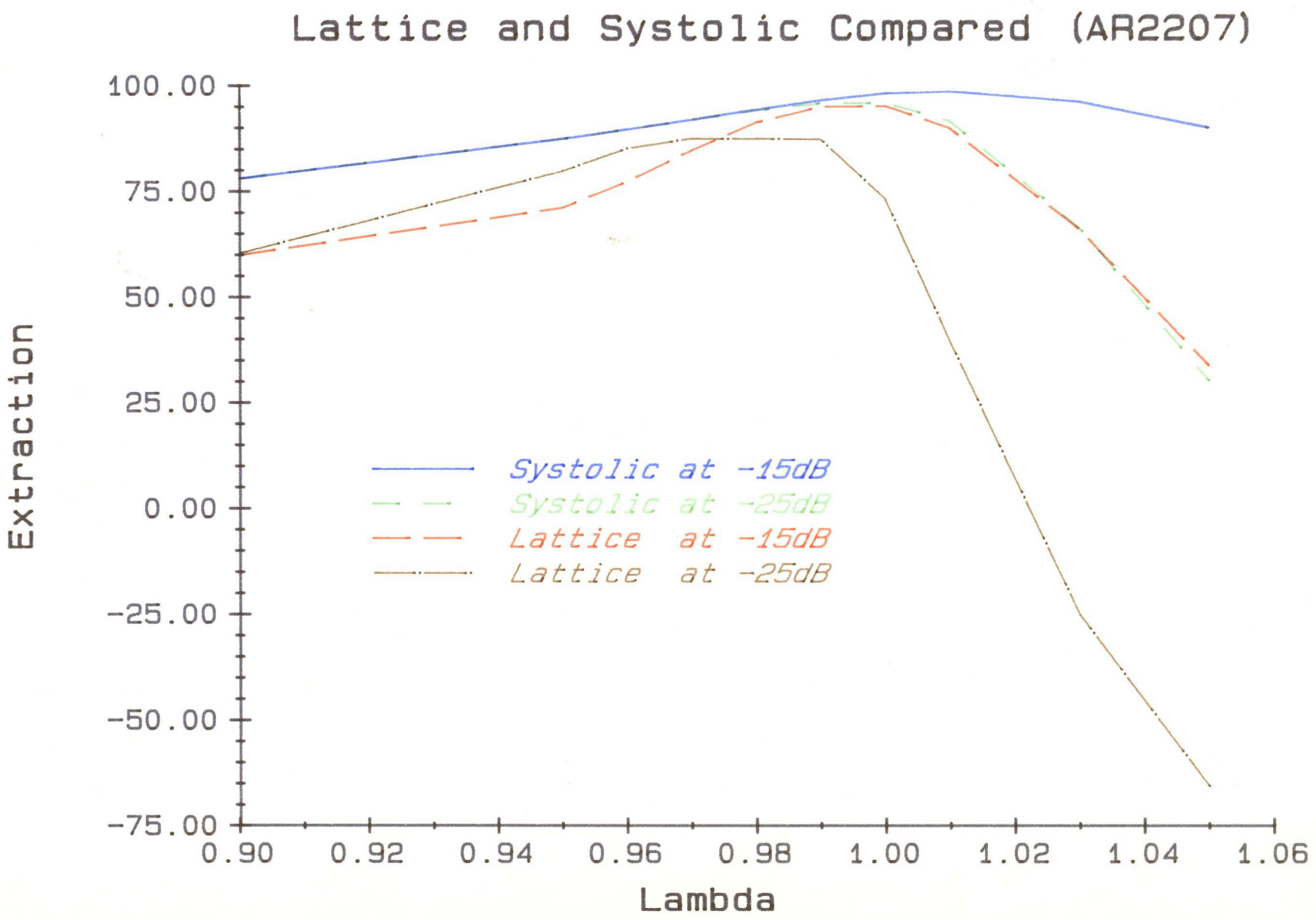
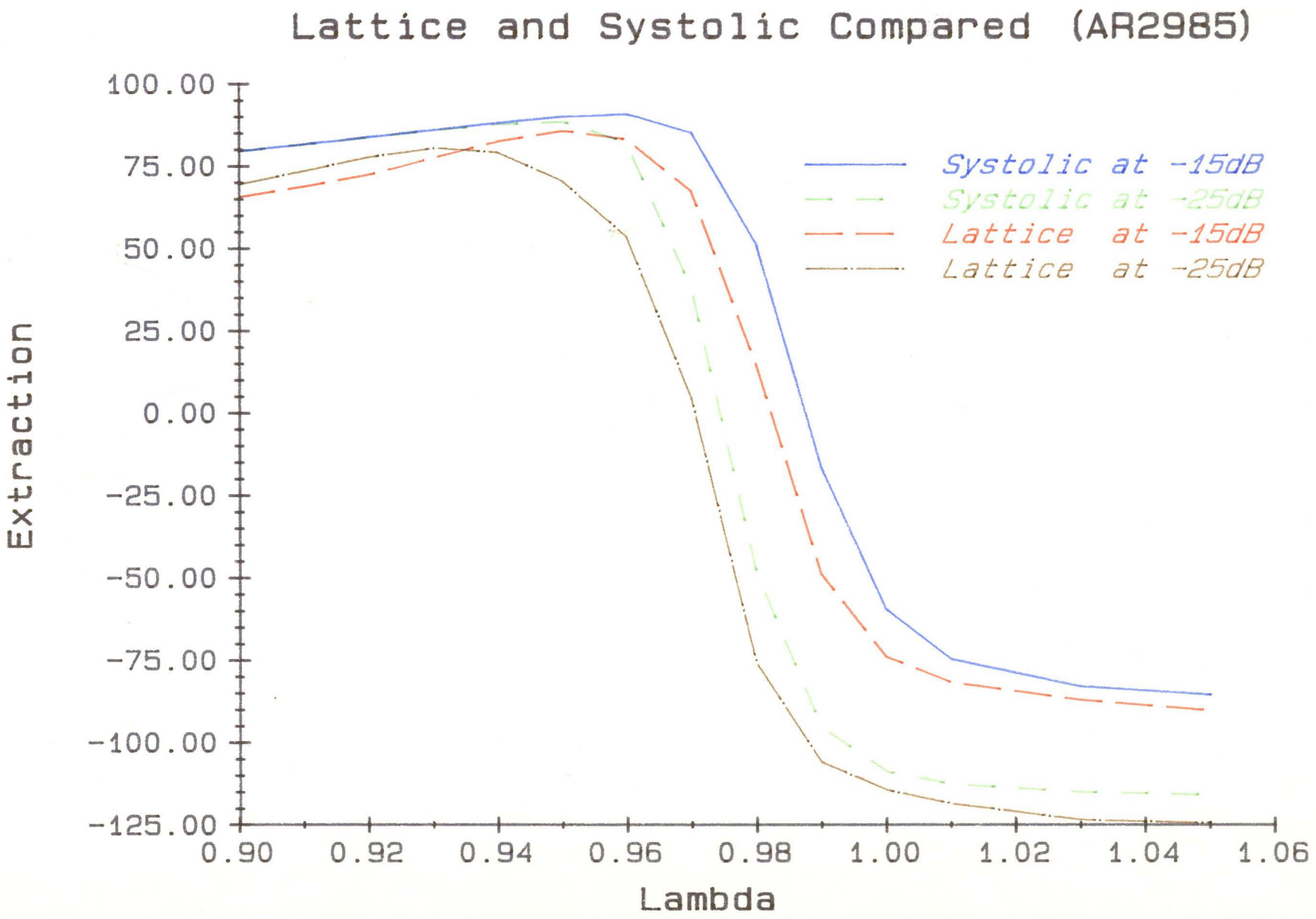

Lattice and Systolic Compared (AR2207)
100.00
75.00
50.00
25.00
0.00
-25.00
-50.00
-75.00
Extraction
0.90
0.92
0.94
0.96
0.98
1.00
1.02
1.04
1.06
Lambda
Systolic at -15dB
Systolic at -25dB
Lattice at -15dB
Lattice at -25dB

FIG. 5.17


Lattice and Systolic Compared (AR2985)
Systolic at -15dB
Systolic at -25dB
Lattice at -15dB
Lattice at -25dB
Extraction
100.00
75.00
50.00
25.00
0.00
-25.00
-50.00
-75.00
-100.00
-125.00
0.90
0.92
0.94
0.96
0.98
1.00
1.02
1.04
1.06
Lambda

comparisons were made. Also the sensitivity of the LSL algorithm to SNR in terms of extraction and λ_{opt} is greater than for the systolic array. The optimum lambda for the systolic array is 1.0 for both SNRs while it is 0.97 for the LSL at -25dB. The apparent convergence of the systolic array curve for -25dB and the lattice curve for -15dB represents a statistical coincidence and was not present in other experiments.

The use of lambda greater than one was pursued due to a report by Madhavan (1985) that better results may be achieved with this unconventional use of lambda. This was not confirmed by the work presented here and all optimal lambdas were found to be less than unity. The discrepancy may be due to the relative extraction index, or some other processing procedure used by Madhavan.

Figure 5.17 illustrates the striking effectiveness of lambda when used to compensate for finite sampling induced non-stationarity. Table 5.3 lists the results of fig. 5.17.

Table 5.3

	λ_{opt}		Max. Extraction	
	LSL	Sys	LSL	Sys
-15dB	0.95	0.96	85.63	90.88
-25dB	0.93	0.95	80.46	88.54

The price paid for the superior performance of the systolic array is the increase in computing power required at larger filter orders. This is because the triangular structure of the array results in an N^2 relationship between the number of calculations and the filter order N. Processing time for the lattice and systolic array are approximately equal up to about 30th order. Beyond this, the memory mapping of the PDP-11 becomes cumbersome and the systolic array takes significantly longer. Results will vary according to the specific machine. Test results comparing the run time of the systolic array (standard memory algorithm, see chap. 3) and the exact lattice algorithm processing a 256 point sequence on the McMaster Senior Sciences Vax are listed in table 5.4.

Table 5.4

Order	Systolic (sec)	Lattice (sec)
30	5	5
50	11	9
75	17	13
100	30	17
150	85	21
200	240	35
300	627	49

5.3.3 Ancillary Experiments

Word length effects were investigated with double precision arithmetic. Table 5.5 shows that the systolic array is numerically very stable. Under single precision on the PDP-11 five decimal places are carried while double precision allows 17 decimal digits. The increase in accuracy using double precision is insignificant.

Table 5.5

Order	Single	Double	|Error|
20	78.97052	78.97028	0.00024
35	60.65286	60.65312	0.00026
50	42.86654	42.86683	0.00029

Pre-filtering of the reference signal results in complete lack of cancellation of the corresponding frequencies in the primary signal. Therefore to achieve maximum noise cancellation band-pass filtering should be reserved until after the adaptive processing has been completed.

Frequency content of the BAEP used as a test signal in the primary sequence had no significant effect on the extraction. Consequently further experiments were done using the full band-width averaged BAEP from figure 2.1.

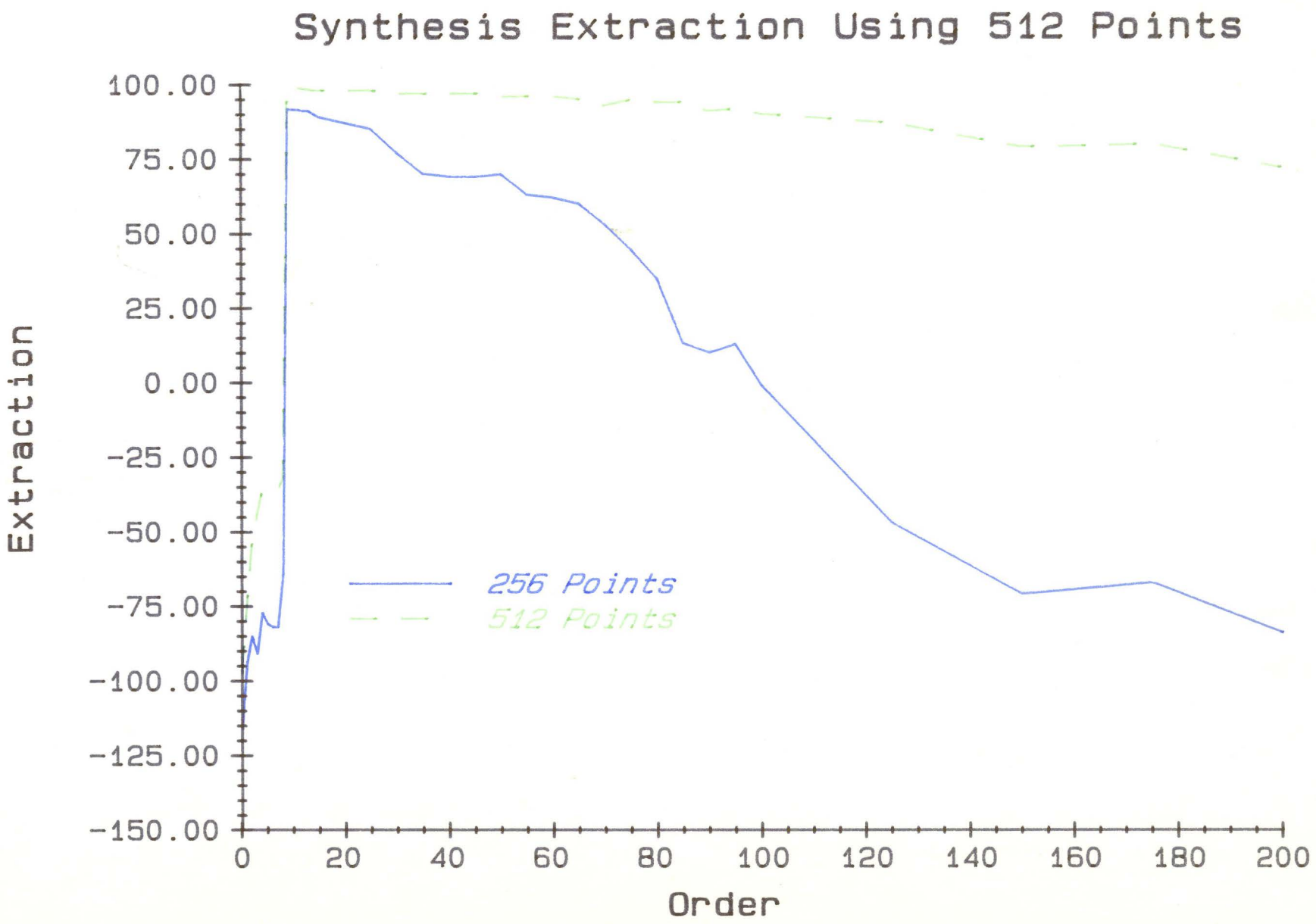
The use of a 256 point signal was based upon our data collection setup. Experiments examining the effect of increasing the number of data samples are presented in figure 5.18 (AR8 with Ω_{opt} of 9). By using 512 points it was possible to increase the maximum extraction percentage and decrease the loss of extraction when $\Omega > \Omega_{opt}$. **This indicates that there is a limitation to the maximum extraction possible imposed by the number of data points in the sequence.** The objective of changing the associated filter parameters (λ,Ω) is, therefore, to achieve this extraction maximum.

Results of tests to find the optimum lambda for sequences of 512 points reinforce our theory about the function of lambda for simulated data. With longer data sequences the true signal is better represented. Thus from our hypothesis we would expect to find a λ_{opt} nearer to unity for the 512 compared to the 256 point sequence. The results of table 5.6 (filtered at Ω_{opt}) agree with this prediction.

Table 5.6

Reference signal	λ_{opt} 256 pnts	512 pnts
AR2207	1.00	1.00
AR2985	0.95	0.98
AR8 (modelled EEG)	0.99	1.00

FIG. 5.18

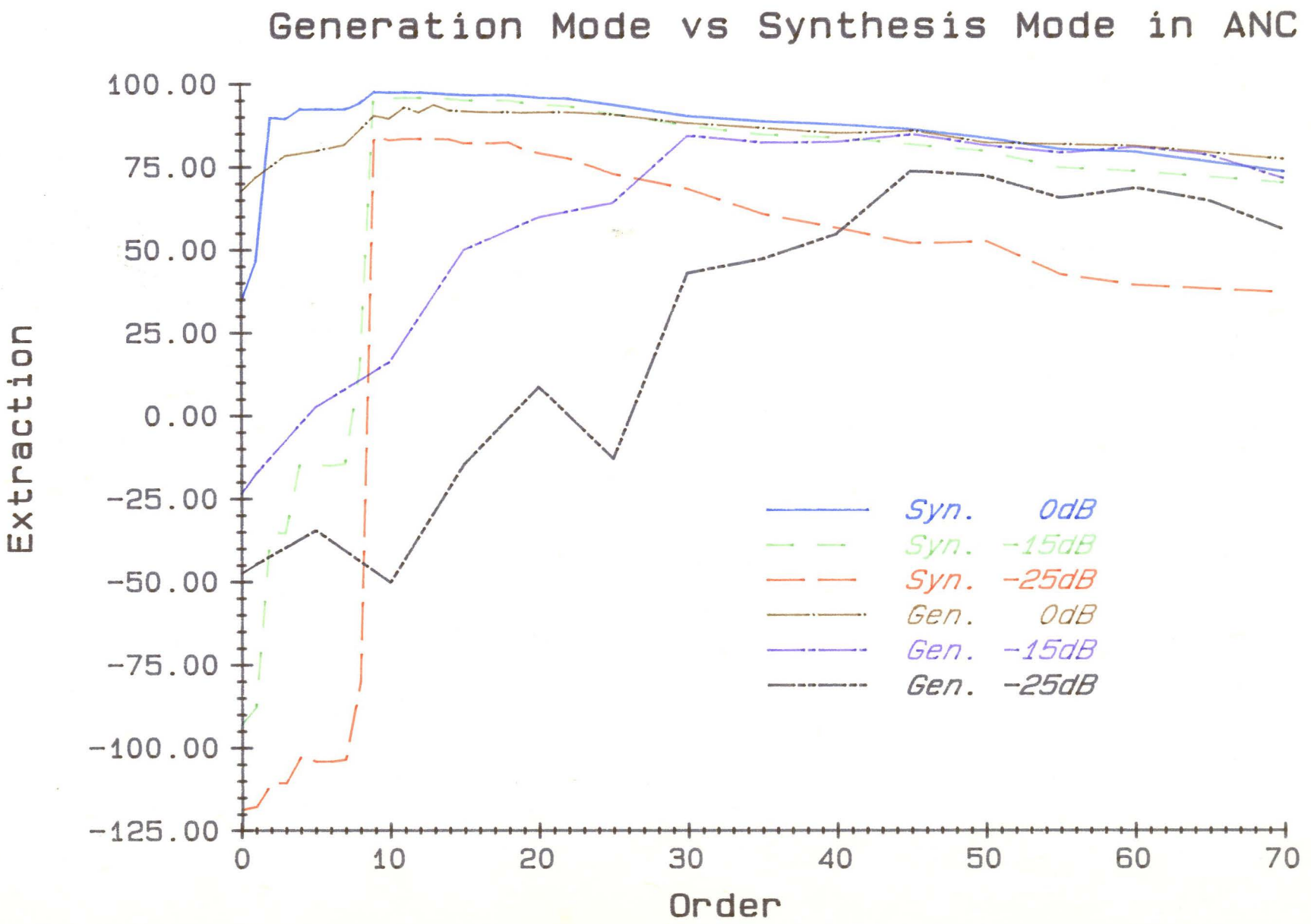

Synthesis Extraction Using 512 Points
Extraction
100.00
75.00
50.00
25.00
0.00
-25.00
-50.00
-75.00
-100.00
-125.00
-150.00
0
20
40
60
80
100
120
140
160
180
200
Order
256 Points
512 Points

5.3.4 Generation Scenario (Coloured Primary)

We now investigate the performance of the SANC in an IIR mode. Using a coloured primary noise and a white reference noise the systolic array is forced to generate an auto-regressive process from the white noise source. The AR expression is now represented in terms of its inverse. A large number of tap weights may be required to adequately represent an AR process of a relatively small number of coefficients.

Figure 5.19 (λ = 1) illustrates the SANC performance with the modelled EEG as the primary noise compared to the previous results using the modelled EEG as the reference input. Notice the slower rise in the extraction rate as a function of order compared to the synthesis plots. Also there is no sharp transition to an optimum order, rather there is a gradual non-monotonic increase in extraction followed by a decrease similar to the synthesis case. The optimum order is now a function of the SNR and the poorest SNR requires the largest order for maximum extraction. The higher extraction at zeroth order is due to the more restricted frequency content of the primary noise (AR8).

FIG. 5.19


Generation Mode vs Synthesis Mode in ANC
Extraction
100.00
75.00
50.00
25.00
0.00
-25.00
-50.00
-75.00
-100.00
-125.00
0
10
20
30
40
50
60
70
Order
Syn. 0dB
Syn. -15dB
Syn. -25dB
Gen. 0dB
Gen. -15dB
Gen. -25dB

These findings are quite easily understood from the stand point of the representation of a polynomial by its inverse. The functional relationship between SNR and Γ_{opt} is due to the necessity for a more accurate representation of the AR process in order to successfully extract the BAEP at lower SNRs. The slow upward curve of the lines is due to the relatively large number of tap weights required and the theoretical need for an infinite number of coefficients for complete representation.

In figure 5.20 (256 points, SNR = -25dB, λ = 1) the results of using different orders of AR signals as the primary noise are compared. When attempting to use the SANC in a generation format it is important to have a large number of data samples to represent the sequence. Optimum orders of 50-70 were used and thus the number of data points in the sequence has a much more profound effect upon the extraction curve than when a 9th order filter is required. Figure 5.21 (AR8 with λ = 1 and SNR = -25dB) illustrates the differences between the use of 256 and 512 point sequences used in the generation mode. The extraction curve is a function of the SNR, the auto-regressive order and the number of points in the data sample. The curve will rise until the limitation based on the number of data points in the sample is reached (as described in the

FIG. 5.20

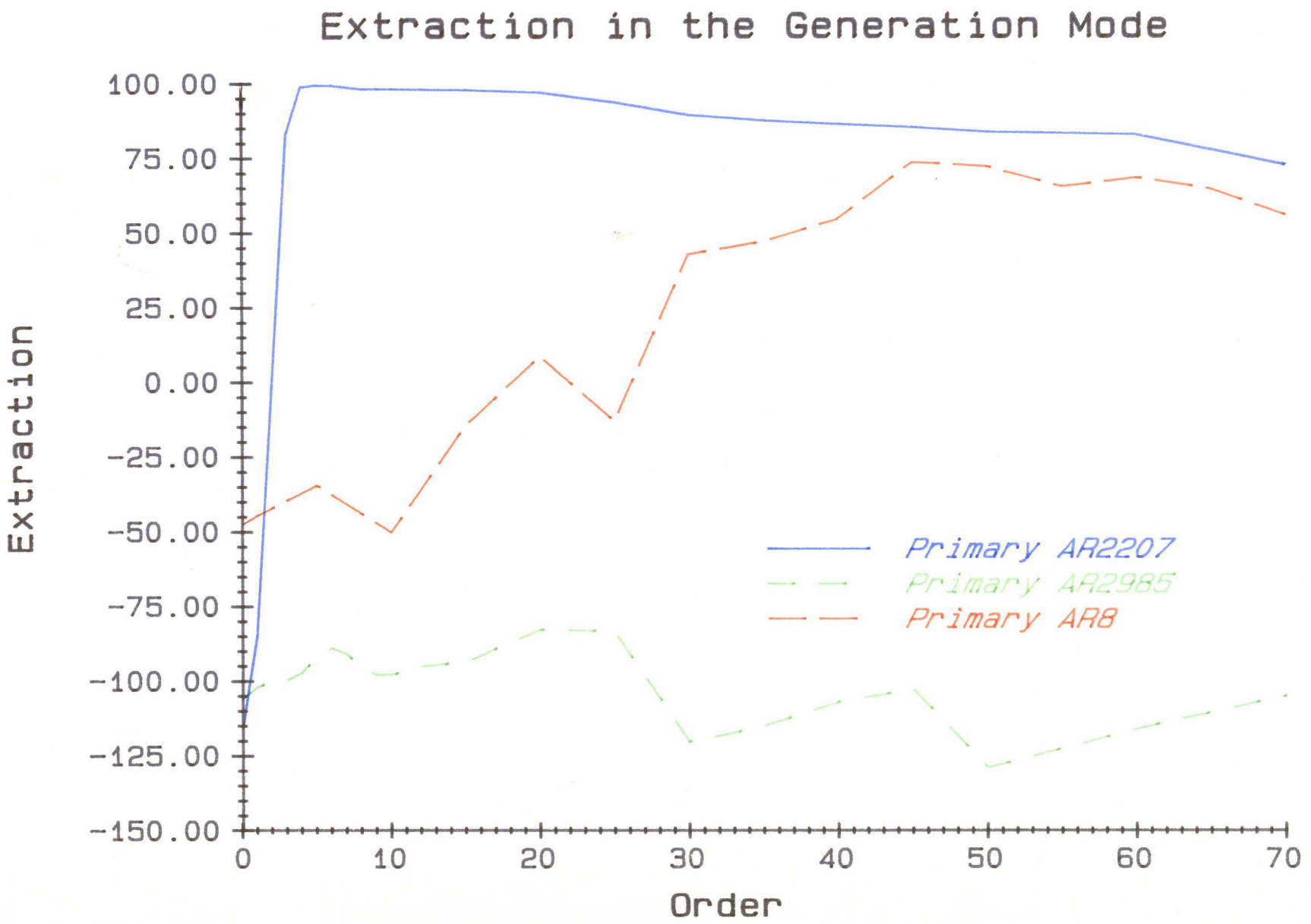
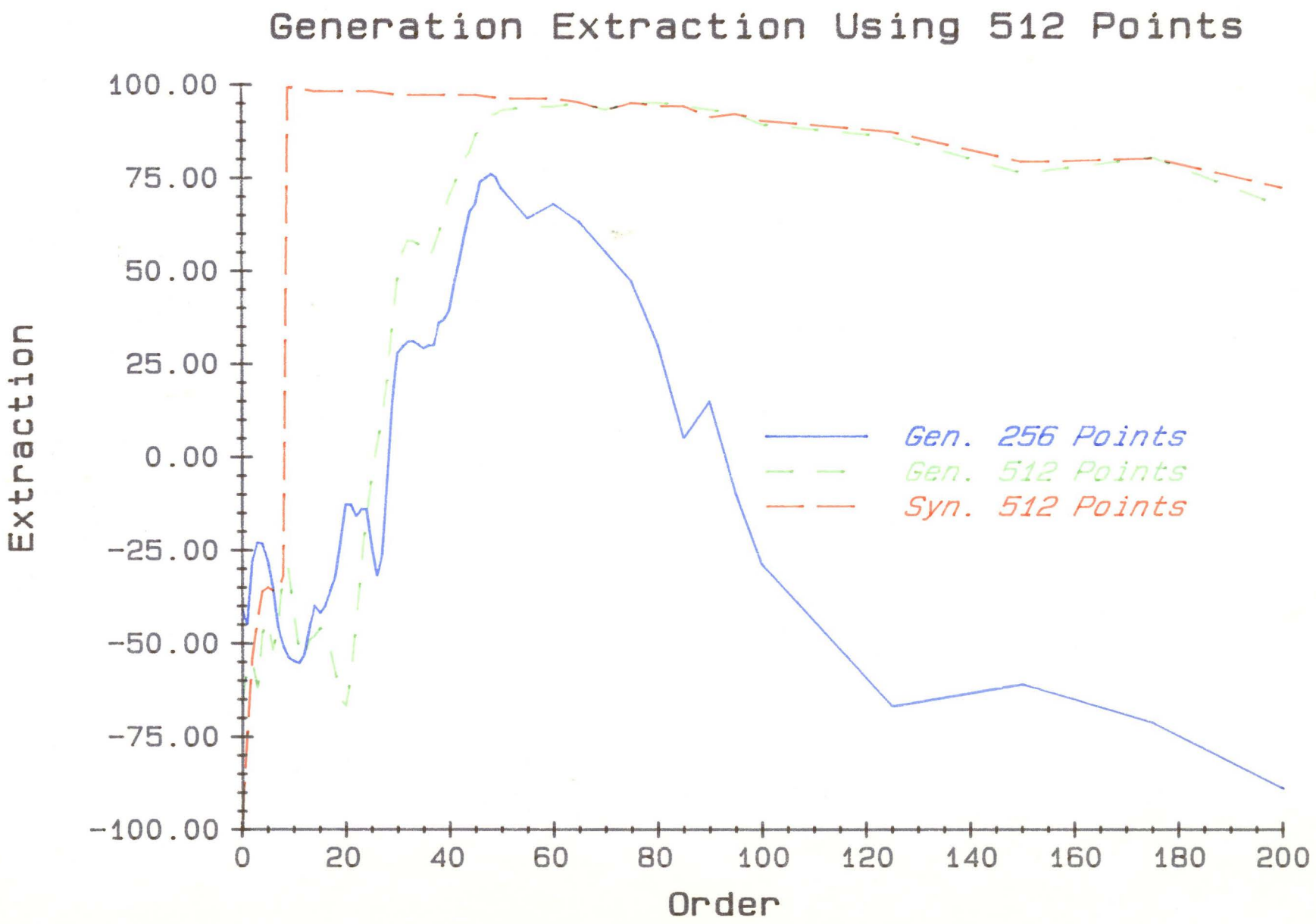

Extraction in the Generation Mode
Extraction
100.00
75.00
50.00
25.00
0.00
-25.00
-50.00
-75.00
-100.00
-125.00
-150.00
0
10
20
30
40
50
60
70
Order
Primary AR2207
Primary AR2985
Primary AR8

FIG. 5.21


Generation Extraction Using 512 Points
Extraction
100.00
75.00
50.00
25.00
0.00
-25.00
-50.00
-75.00
-100.00
0
20
40
60
80
100
120
140
160
180
200
Order
Gen. 256 Points
Gen. 512 Points
Syn. 512 Points

synthesis procedure). The maximum of the extraction curve is at the intersection between the rising curve due to the large number of tap weights and the falling curve due to the data sample limitation. The faster the curve rises the better the extraction.

From the results in this section, the use of SANC in a generation format has a number of disadvantages:

1) The maximum extraction is less successful than in the synthesis format.
2) The optimum order is larger thus requiring more processing time.
3) The optimum order is less distinct.
4) The optimum order is a function of the SNR and AR order.
5) The extraction curve is non-monotonic.

For reliable results orders of 50-70 must be used.

Experiments using coloured noise sequences as inputs to both the primary and reference channels result in curves which display the same characteristics as listed above. In this mode the SANC is representing the ratio of polynomials which also requires an infinite number of coefficients for

complete theoretical representation. It is therefore not surprising to find the similarities between this format and the generation format.

The results of this section provide a great deal of incentive to develop new adaptive filter structures (either IIR or a combination of FIR) in order to improve both processing time and characteristics.

5.3.5 Adaptive Noise Cancellation of Physiological Data

The data collection as described in Chapter 4 was designed to exploit correlations between EEG prior to and after stimulation (figure 4.3). From earlier discussions it is obvious that the average BAEP from a set of 2000 records may not be a good template for use in evaluation of the extraction procedure applied to a single stimulus BAEP. In order to provide a good template, and to eliminate possible correlations between BAEP and EEG we recorded several series of EEG without stimulation. The 512 point records were then split into reference and primary signals. As in the simulations, the averaged BAEP was added to the primary noise (at 0dB). Thus we are testing, specifically, the correlation of the pre-stimulus EEG and post-stimulus EEG.

(After successful tests the SNR could first be reduced to -25dB and then real BAEPs in EEG could be tested)

Results are shown in figure 5.22. If the correlation between the pre-stimulus EEG and post-stimulus EEG were adequate we would expect curves similar to those in figure 5.20. However the inability of the SANC to produce results comparable to those in the simulation indicate that the correlation is **not sufficient.** Tests were done using a number of records from two different subjects. **All results indicated a lack of appropriate correlation.**

The reason for the difference in results can be seen by comparing the format of the collected data and the simulated data. The simulated data can be viewed as multiple channel data. The white noise is passed through an autoregressive filter producing a reference which is collected in a parallel channel format. In contrast the real data was collected in a single channel format where the reference signal was the pre-stimulus EEG.

To investigate this difference an AR8 sequence of 512 points was generated using the modelled EEG coefficients. The simulated sequence was then treated in the same manner as the real EEG just described. Results in figure 5.23

FIG. 5.22

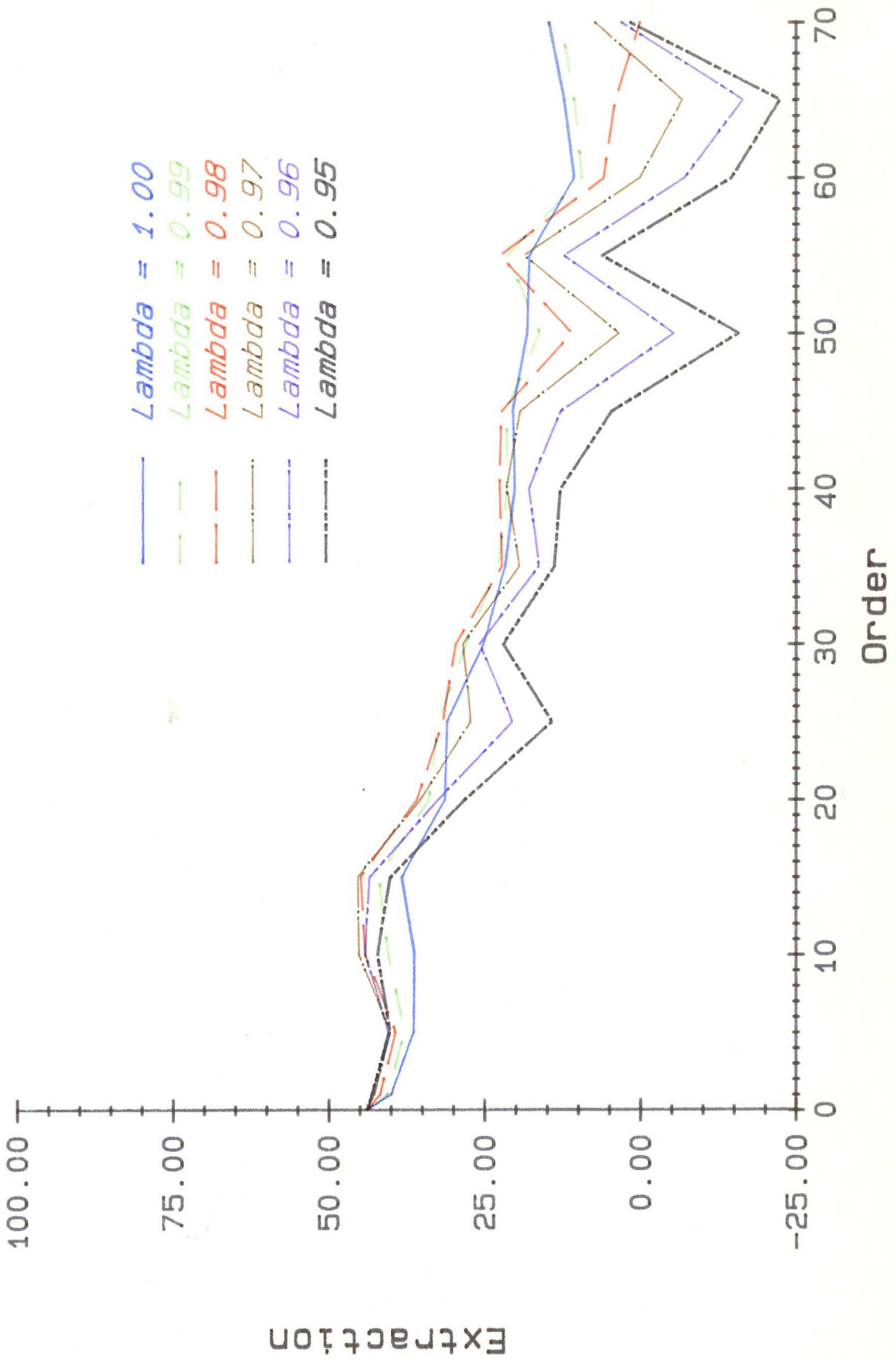
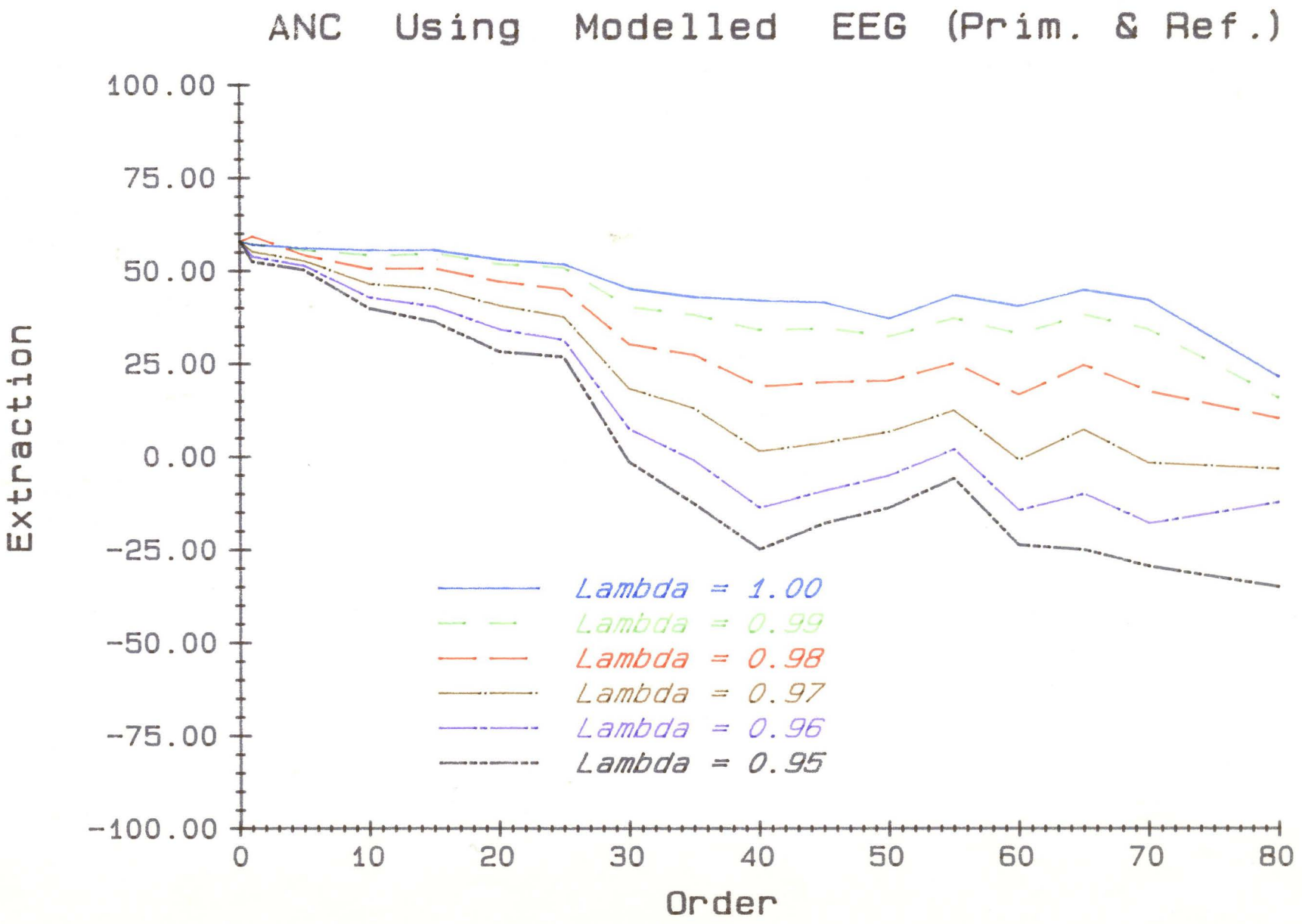

ANC Using Real EEG
Lambda = 1.00
Lambda = 0.99
Lambda = 0.98
Lambda = 0.97
Lambda = 0.96
Lambda = 0.95
Extraction
100.00
75.00
50.00
25.00
0.00
-25.00
0
10
20
30
40
50
60
70
Order

FIG. 5.23


ANC Using Modelled EEG (Prim. & Ref.)
100.00
75.00
50.00
25.00
0.00
-25.00
-50.00
-75.00
-100.00
Extraction
0
10
20
30
40
50
60
70
80
Order
Lambda = 1.00
Lambda = 0.99
Lambda = 0.98
Lambda = 0.97
Lambda = 0.96
Lambda = 0.95

support the finding that single channel collection format produces an inadequate reference signal for adaptive noise cancellation of BAEP.

In figure 5.24 the cross correlation of the parallel channel and single channel reference and primary signal. Figure 5.24a and 5.24b are the cross correlations between the white noise and the generated AR2207 and modelled EEG (AR8) sequences respectively (parallel). Notice the predominance of the spike at lag zero. In figures 5.24c and 5.24d the correlation is illustrated between the reference and primary of the single channel modelled EEG and the real EEG respectively (sequential). The striking similarity between figs. 5.24c and 5.24d is the result of the high quality autoregressive model of the EEG signal. Figures 5.24a and 5.24b contrast with 5.24c and 5.24d supporting our interpretation of figures 5.21 and 5.22 in terms of lack of appropriate correlation.

FIG. 5.24a

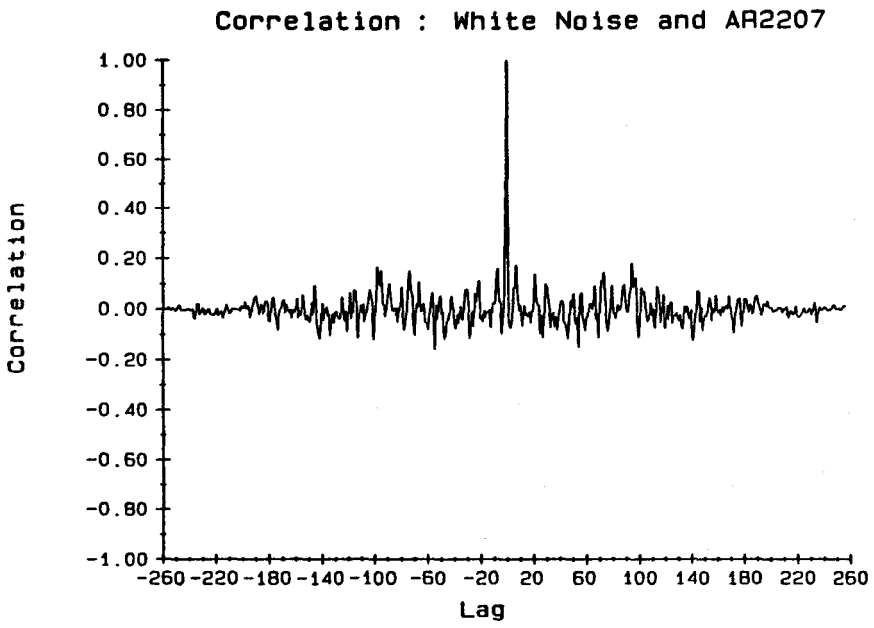


FIG. 5.24b

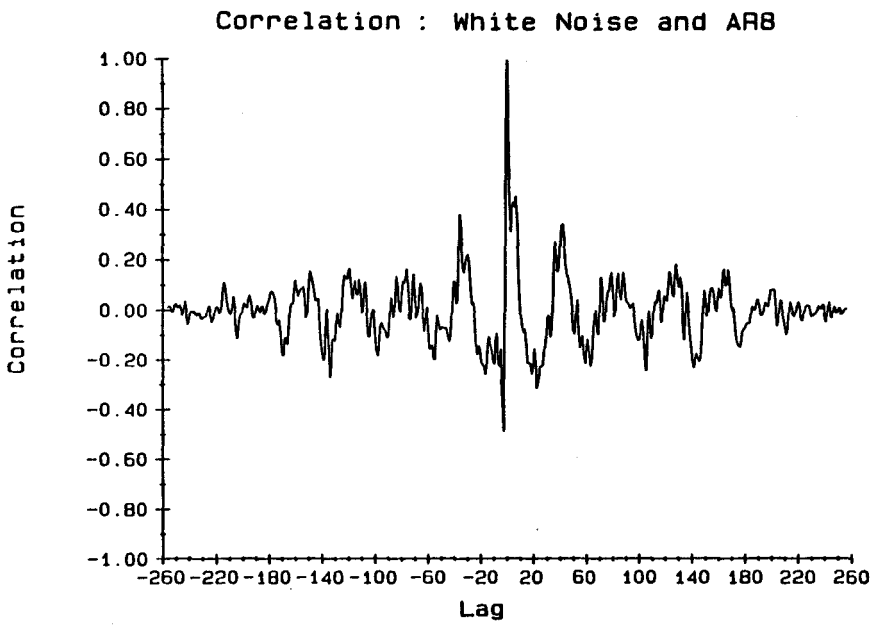


FIG. 5.24c

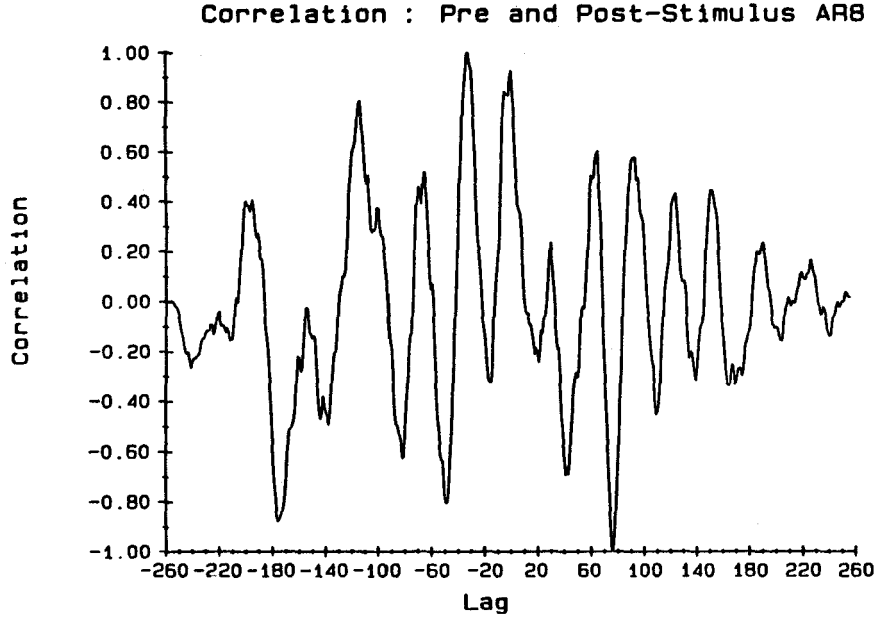


FIG. 5.24d

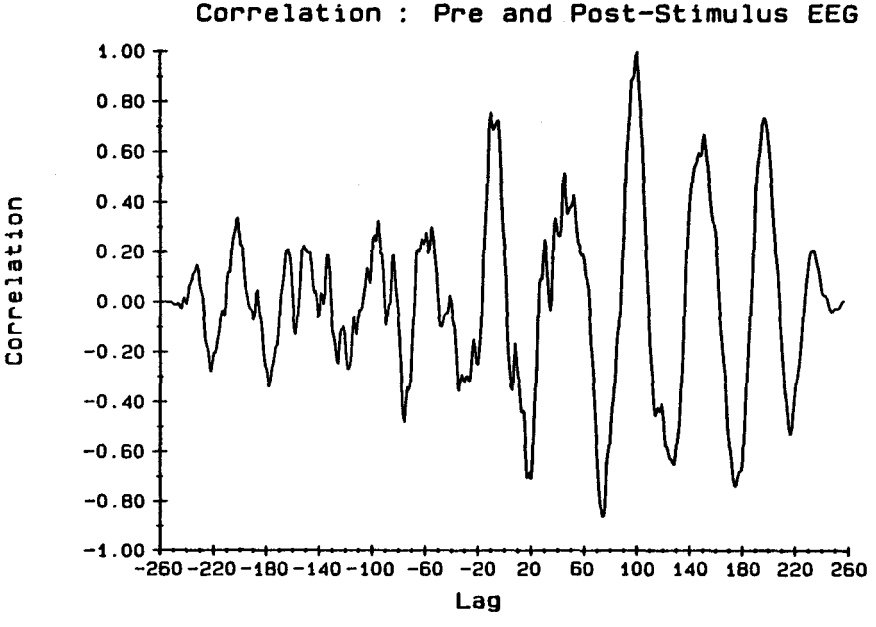


fig. 5.24 Correlation plots

5.4 Summary of Experimental Results

We now present a list of observations and conclusions determined from the results in this chapter.

1. Optimization of order and lambda are decoupled procedures in the synthesis scenario.

2. Order optimization depends upon the nature of filtering format. For FIR it is the order of the AR sequence plus one. For IIR using AR8 inputs the optimum order is between 50-70.

3. The number of data points sampled significantly effects the extraction of the desired signal. We recommend the use of 512-point sequences when dealing with EEG signals.

4. The value of the optimum lambda depends upon the accuracy of the representation of the real signal by the digitally sampled sequence. Poor representation results from fewer points and pole magnitudes approaching unity. Lambda may be used to vastly improve extraction of signals from inadequately sampled noise sequences.

5. Lambdas greater than unity were not shown to be useful.

6. Systolic arrays outperformed lattice structures under all conditions tested.

7. Systolic arrays are numerically very stable.

8. Extraction of simulated signals at -25dB was easily achieved with excellent results.

9. Frequency content of the buried signal was found to be irrelevant to the success of the extraction process.

10. Reference and primary signals from single channel recordings were shown to be inappropriately correlated for adaptive noise cancellation purposes.

CHAPTER 6

CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE WORK

This investigation began with the objective of applying systolic arrays to adaptively cancel EEG noise in order to extract single stimulus BAEP.

EEG and BAEP data were collected using a single channel montage. The data suggested that EEG could be modelled using an autoregressive sequence of 8th order. Signal to noise ratios of -25dB were extrapolated from data using assumptions of constant latency and uncorrelated sample to sample EEG. The validity of these assumptions was discussed and tests using SNRs from -25dB to -15dB were used to simulate real conditions.

Experiments in Chapter 5 were designed to examine the feasibility of applying systolic arrays to the problem of extracting the BAEP from a single EEG record. We began by implementing the synthesis problem (white primary noise and coloured reference signal). Extractions of signals with

SNR as low as -25 to -50dB were successful. Optimization of lambda and the filter order proved to be decoupled problems. Γ_{opt} was found to be the AR order plus one and λ_{opt} was a function of the pole magnitude and number of data points in the sample. We recommend that 512 point sequences be used for the representation of EEG signals in order to minimize the effects of sampling induced non-stationarity.

Tests comparing the performance of the systolic array and the exact least square lattice showed the systolic array to be the superior processing architecture. Processing time for the two implementations was similar up to 30th order at which point the systolic array began to require substantially longer computing times.

Implementation of the generation problem resulted in the use of much larger filter orders (50-70). The overall performance of the ANC was much poorer than for the synthesis format. In order to process EEG data more efficiently the following possibilities are suggested:

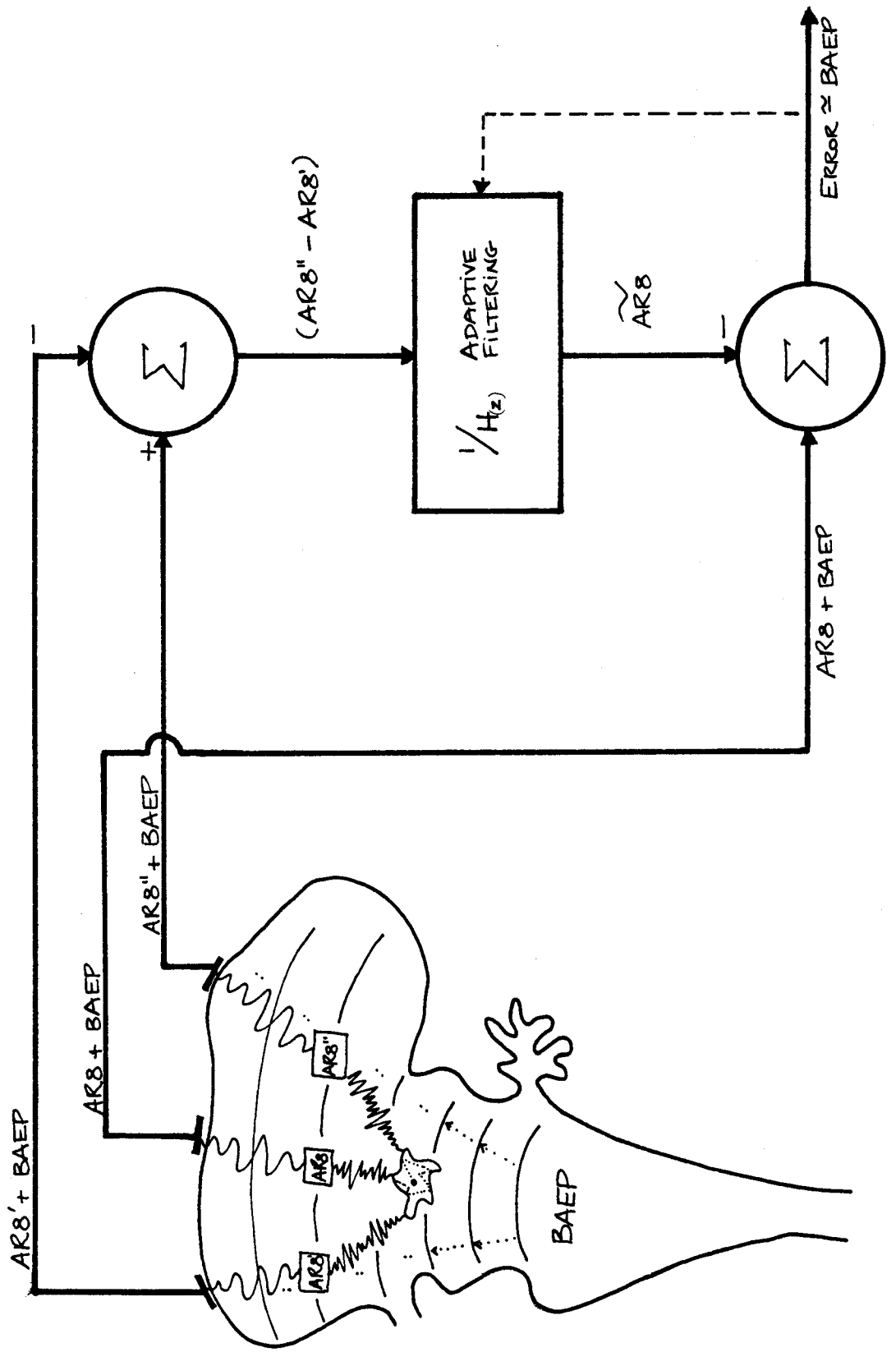
1) Development of an IIR systolic filter. The reader is referred to Landau (1984) for further information.

2) Use of a high order systolic array with a long (512) point data sequence for input. This should be implemented on a high speed machine with a target processing time of less than 5 seconds.

Results using real data were disappointing due to a lack of adequate correlation between temporally separated EEG reference and primary signal. Analysis of the simulation format and correlation data indicate that multi-channel data would be better suited to ANC processing. The model in figure 6.1 shows how a multi-channel collection montage would be comparable to the simulated data format in Chapter 5.

We postulate the existence of a white noise generator in the cerebrum. At the surface there are two correlated EEG signals produced by special filtering of the source signal. We are then faced with the problem of contamination of the reference EEG by BAEP due to the volume conducted nature of the BAEP. Correlation studies are necessary to determine the best location for the electrodes to maximize EEG correlation and minimize cross-talk. Cross-talk resistant adaptive noise cancellation (CRANC) techniques are

fig. 6.1 MULTICHANNEL MODEL FOR ADAPTIVE NOISE CANCELLATION OF BRAINSTEM AUDITORY EVOKED POTENTIALS



available (Ferrara & Widrow, 1981 and Harrison et al., 1986) and investigation using CRANC are continuing here at McMaster.

We recommend that studies in this area be pursued as vigorously as possible. In this thesis we have identified the systolic array as a powerful and flexible signal processing technique. We also believe that through the use of a multi-channel data collection scheme the extraction of a BAEP from a single EEG record will soon be a reality. Much work remains to be done in terms of optimization of processing parameters and electrode montages.

There is a huge potential for the application of this technique to the extraction of other evoked potentials such as the visual evoked potential (VEP) and the brainstem somatosensory evoked potential (BSSEP).

The rewards of success in this area will be the availability of new information for successful diagnosis of neurological disorders and in the discovery of presently unknown neurological phenomena.

APPENDIX I

```
C       F I L E N A M E  :  L S L A N C . F O R
C
C
C       AUTHORS : ROBERT C. SCOTT & MADHAVAN (PRIMARY)
C
C
C       DATE  1ST OCTOBER 1985
C
C
C            ***      LSLANC.FOR      ***
C
C     ************************************************
C     * ADAPTIVE NOISE CANCELLATION USING NORMALIZED *
C     *                L A T T I C E                 *
C     ************************************************
C
C    INPUT :-  SIMULATION
C
C            PRIMARY : BS#C##___.SPN - (SIGNAL+NOISE) DATA FILE
C            REFERENCE:REFAR#___.NSE -    (NOISE) DATA FILE
C
C    OUTPUT:-
C            ESTIMATED SIGNAL :    BS####___.L##
C
C
C
C    INPUT : - RAW ADATAPTATION
C
C          PRIMARY:  B#REF#.EVK   SIGNAL PLUS NOISE
C          REFERENCE B#REF#.EEG   NOISE FILE
C
C
C    OUTPUT :  ESTIMATED SIGNAL
C
C           B#REF#.LNC
C
C
C
      DIMENSION XN(257).XSN(257).ECOVN(2).ECOVSN(2)
      DIMENSION RFCOFN(257.2).FPEN(257.2).BPEN(257.2)
      DIMENSION RGCOF(257.2).SNOR(257.2).SOUT(257)
      DIMENSION RFINI(257).RGINI(257)
      DIMENSION IFILE(7).JFILE(7).KFILE(7)
      DIMENSION IPARAM(7)
C
C
C     ************************************************************
C     FUNCTION STATEMENTS
      UPD1(A.B.C)=A*SQRT(1-B*B)*SQRT(1-C*C)+B*C
      UPD2(D.E.F)=(D-E*F)/SQRT(1-E*E)/SQRT(1-F*F)
C
      AUTOD='N'
      AUTOL='N'
      WRITE(7.300)
300   FORMAT(/' INPUT RAW OR SIMULATION (R/S)')
301   FORMAT(A1)
      READ(5.301)C3
6001  IF(C3.EQ.'S')GOTO 302
C
      DATA IFILE/'DK'.'1:'.'B#'.'##'.'##'.'.E'.'VK'/
```

```
      DATA JFILE/'DK'.'1:'.'B#'.'##'.'##'.'.E'.'EG'/
      DATA KFILE/'DK'.'1:'.'B#'.'##'.'##'.'.L'.'NC'/
C
      WRITE(7.303)
303   FORMAT(' ENTER DATA BASE # (B#) FOR INPUT FILE')
      READ(5.304)IFILE(3)
304   FORMAT(A2)
      WRITE(7.305)
305   FORMAT(' ENTER REC# FOR FILE (##) MOST SIG. PAIR FIRST')
      READ(5.304)IFILE(4)
      WRITE(7.305)
      READ(5.304)IFILE(5)
      JFILE(3)=IFILE(3)
      KFILE(3)=IFILE(3)
      JFILE(4)=IFILE(4)
      KFILE(4)=IFILE(4)
      JFILE(5)=IFILE(5)
      KFILE(5)=IFILE(5)
      WRITE(7.311)(IFILE(J).J=1.7)
311   FORMAT(' '.7A2)
      GOTO 306
302   CONTINUE
      IFILE(1)='DK'
      JFILE(1)=IFILE(1)
      KFILE(1)=IFILE(1)
      IFILE(2)='1:'
      JFILE(2)=IFILE(2)
      KFILE(2)=IFILE(2)
      IFILE(3)='BS'
      JFILE(3)='RE'
      IFILE(6)='.S'
      IFILE(7)='PN'
      JFILE(4)='FA'
      JFILE(6)='.N'
      JFILE(7)='SE'
      KFILE(6)='.L'
C
C
      IF(AUTOL.EQ.'Y')GOTO 7003
      WRITE(7.322)
322   FORMAT(' ENTER ORDER # (<128) (-- TO END AUTO)')
      READ(5.304)KFILE(5)
      IF(KFILE(5).EQ.'--')GOTO 6999
      IF(AUTOD.EQ.'Y')GOTO 2001
      WRITE(7.320)
320   FORMAT(' ENTER DATA BASE NUMBER OF OUTPUT FILE (B#)')
      READ(5.304)KFILE(3)
C
C
      WRITE(7.307)
307   FORMAT(' ENTER BASE FILE NUMBER FOR INPUT FILE (#C)')
      READ(5.304)IFILE(4)
      WRITE(7.308)
308   FORMAT(' ENTER ABS(S/N) (##)')
      READ(5.304)IFILE(5)
      WRITE(7.309)
309   FORMAT(' ENTER AR ORDER OF NOISE  (R#) ')
      READ(5.304)JFILE(5)
      KFILE(7)=IFILE(5)
7003  WRITE(7.321)
```

```
321     FORMAT(' ENTER LAMBDA # (97=.97.00=1.00.03=1.03 ETC. (--STOP)')
        READ(5.304)KFILE(4)
        IF(KFILE(4).EQ.'--')GOTO 6999
C
306     CONTINUE
C
2001    DO 7101 I=1.2
        ECOVN(I)=0.0
        ECOVSN(I)=0.0
        DO 7101 J=1.257
        RFCOFN(J.I)=0.0
        FPEN(J.I)=0.0
        BPEN(J.I)=0.0
        RGCOF(J.I)=0.0
        SNOR(J.I)=0.0
        SOUT(J)=0.0
        RFINI(J)=0.0
        RGINI(J)=0.0
        XN(J)=0.0
7101    XSN(J)=0.0
C
        IF(AUTOL.EQ.'Y')GOTO 7002
        WRITE(7.50)
        READ(5.60)NORD
        IF(AUTOD.EQ.'Y')GOTO 28
C
        ILOOP=1
 20     FORMAT(A2)
        DO 21 I=1.7
        IPARAM(I)=KFILE(I)
 21     CONTINUE
 24     FORMAT(A2)
        WRITE(7.16)
 16     FORMAT(' HOW MANY POINTS ? (<=256) '/' ***')
        READ(5.18) NT
 18     FORMAT(I3)
7002    WRITE(7.30)
 30     FORMAT(' MEMORY FACTOR. LAMBDA '/' *.***')
        READ(5.40) WLMD
 40     FORMAT(F5.3)
        IF(AUTOL.EQ.'Y')GOTO 28
 50     FORMAT(' ENTER # TAP WEIGHTS (<=128)'/' ***')
 60     FORMAT(I2)
        WRITE(7.1025)
1025    FORMAT(' AUTO ORDER(Y/N)?')
        READ(5.301)AUTOD
        IF(AUTOD.EQ.'Y')GOTO 28
        WRITE(7.1027)
1027    FORMAT(' AUTO LAMBDA (Y/N) ?')
        READ(5.301)AUTOL
28      LNT2=NT*2
        LNT=NT*2+2
        NO=NORD+1
        NOT2=NO*2
        NTP1=NT+1
C
        CALL ASSIGN(1.IFILE.14.'OLD')
        DEFINE FILE 1(1.LNT2.U.IREC)
        CALL ASSIGN(2.JFILE.14.'OLD')
        DEFINE FILE 2(1.LNT2.U.JREC)
```

```
      CALL ASSIGN(3,KFILE,14,'NEW')
      DEFINE FILE 3(1,LNT2,U,KREC)
C
      READ (1' 1) (XSN(I),I=1,NT)
      READ (2' 1) (XN(I),I=1,NT)
      CLOSE (UNIT=1)
      CLOSE (UNIT=2)
C
C     ************* MEAN CALC OF INPUTS XSN,XN *****************
C
      AVGXN=0.0
      AVGXSN=0.0
      DO 70 I=1,NT
      AVGXN=AVGXN+XN(I)
      AVGXSN=AVGXSN+XSN(I)
 70   CONTINUE
      AVGXN=AVGXN/FLOAT(NT)
      AVGXSN=AVGXSN/FLOAT(NT)
      DO 80 I=1,NT
      XN(I)=XN(I)-AVGXN
      XSN(I)=XSN(I)-AVGXSN
 80   CONTINUE
C
C     ************* VARIANCE CALC OF INPUT SIGS ********************
C
      COVNP=0.0
      COVSNP=0.0
      DO 90 I=1,NT
      COVNP=COVNP+(XN(I)*XN(I))
      COVSNP=COVSNP+(XSN(I)*XSN(I))
 90   CONTINUE
      COVNP=COVNP/FLOAT(NT-1)
      COVSNP=COVSNP/FLOAT(NT-1)
C
      WRITE(7,91) (IFILE(I),I=1,5)
 91   FORMAT(25X,'* FOR FILE ',5A2,' *')
      WRITE(7,92) AVGXSN,COVSNP
 92   FORMAT(3X,'SIG.+NOISE - MEAN = ',E15.4,' VARIANCE = ',E15.4)
      WRITE(7,94) AVGXN,COVNP
 94   FORMAT(3X,' NOISE   -   MEAN = ',E15.4,' VARIANCE = ',E15.4)
C
      ECOVN(1)=COVNP+XN(1)*XN(1)
      ECOVSN(1)=COVSNP+XSN(1)*XSN(1)
C
      FPEN(1,1)=XN(1)/SQRT(ECOVN(1))
      BPEN(1,1)=FPEN(1,1)
      DO 100 J=1,2
      DO 100 IO=1,NO
      RFCOFN(IO,J)=0.0
      RGCOF(IO,J)=0.0
 100  CONTINUE
C
C
C     ************************************************************
C     LATTICE CALCULATIONS
C
C
 108  WRITE(7,110)
 110  FORMAT(/10X,'   *** LATTICE COMPUTATIONS IN PROGRESS ***'//)
C
```

```
 5000    DO 1000 IT=2.NT
         IND=IT-2
C
 5002    ECOVN(2)=WLMD*ECOVN(1)+XN(IT)*XN(IT)
         FPEN(1.2)=XN(IT)/SQRT(ECOVN(2))
         BPEN(1.2)=FPEN(1.2)
         ECOVSN(2)=WLMD*ECOVSN(1)+XSN(IT)*XSN(IT)
         SNOR(1.2)=XSN(IT)/SQRT(ECOVSN(2))
         IF(IT .LE. NO) MAXO=IT-1
         IF(IT .GT. NO) MAXO=NO-1
C
         WRITE(7.550) IT
 550     FORMAT('+'.27X.I3)
C
         DO 500 IO=1.MAXO
         RFCOFN(IO+1.2)=UPD1(RFCOFN(IO+1.1).FPEN(IO.2).BPEN(IO.1))
         FPEN(IO+1.2)=UPD2(FPEN(IO.2).RFCOFN(IO+1.2).BPEN(IO.1))
         BPEN(IO+1.2)=UPD2(BPEN(IO.1).RFCOFN(IO+1.2).FPEN(IO.2))
         RGCOF(IO+1.2)=UPD1(RGCOF(IO+1.1).BPEN(IO.2).SNOR(IO.2))
         SNOR(IO+1.2)=UPD2(SNOR(IO.2).RGCOF(IO+1.2).BPEN(IO.2))
 500     CONTINUE
C
         MAXOP1=MAXO+1
         SOUT(IT)=SNOR(MAXOP1.2)
C
         ECOVN(1)=ECOVN(2)
         ECOVSN(1)=ECOVSN(2)
         DO 600 IO=1.MAXOP1
         RFCOFN(IO.1)=RFCOFN(IO.2)
         FPEN(IO.1)=FPEN(IO.2)
         BPEN(IO.1)=BPEN(IO.2)
         RGCOF(IO.1)=RGCOF(IO.2)
 600     CONTINUE
 1000    CONTINUE
C
C
         SOUT(1)=SOUT(2)
         WRITE(3' 1) (SOUT(IT).IT=1.NT)
         CLOSE (UNIT=3)
C
         IF(AUTOD.EQ.'Y')GOTO 6001
         IF(AUTOL.EQ.'Y')GOTO 6001
6999     CONTINUE
         END
```

APPENDIX II

```
C       F I L E N A M E  :  S Y S A N C . F O R
C
C
C       AUTHORS : ROBERT C. SCOTT
C
C
C       NOTE: Compile with 7 units
C
C       DATE  27TH AUGUST 1985
C
C
C            ***      SYSR3A.FOR     ***
C
C     ************************************************
C     *    ADAPTIVE   NOISE   CANCELLATION   USING   *
C     *                                              *
C     *         S Y S T O L I C  A R R A Y           *
C     *      I M P L E M E N T A T I O N     I I     *
C     *                                              *
C     *           FOR   EVOKED    POTENTIALS         *
C     ************************************************
C
C
C    INPUT :- SIMULATION
C
C            PRIMARY : BS#C##___.SPN - (SIGNAL+NOISE) DATA FILE
C            REFERENCE:REFAR#___.NSE -    (NOISE) DATA FILE
C
C    OUTPUT:-
C            ESTIMATED SIGNAL :    B#####___.S##
C
C                                E.G. B3 97 30 .S 00
C                                       /    |     |
C                              0.997 =  LAMBDA.ORDER.S/N
C
C
C    INPUT : - RAW ADATAPTATION
C
C          PRIMARY:  B#REF#.EVK   SIGNAL PLUS NOISE
C          REFERENCE B#REF#.EEG   NOISE FILE
C
C
C    OUTPUT :  ESTIMATED SIGNAL
C
C           B#REF#.SNC
C
C
C        IIFILE: SPN/EVK
C        JFILE: NSE/EEG
C        KFILE: ###/SNC
C        LFILE: SIG/FIL
C
C        INPUTS: # POINTS. #WEIGHTS. LAMBDA. FILE NAMES
C              : STIMULUS POINT. RAW/SIMULATION
C
C        ********************************************************
C        FORMAT STATEMENTS
C
C
1000     FORMAT(/' INPUT RAW OR SIMULATION (R/S) ? ')
```

```
1001    FORMAT(A1)
1002    FORMAT(' ENTER DATA BASE # (B#) ')
1003    FORMAT(A2)
1004    FORMAT(' ENTER RECORD # (##) MOST SIG PAIR FIRST')
1006    FORMAT(' ENTER ABS (S/N) (##)')
1007    FORMAT(' ENTER AR ORDER OF REFERENCE SIGNAL  (R#)')
1008    FORMAT(' ANOTHER FILE (Y/N) ?')
1009    FORMAT(/' INPUT # OF POINTS IN RECORD'/' ****')
1010    FORMAT(I4)
1011    FORMAT(A5)
1012    FORMAT(' RENTER LAMBDA  FULL RANK!'/' *.***')
1013    FORMAT(' ENTER STIMULUS POINT'/' ****')
1014    FORMAT(F5.3)
1015    FORMAT(20X,' POINT # PROCESSED.... '//)
1016    FORMAT(//4X,A14,4X,' % Fit (sq)..  ',F7.2,4X,
       1 'M-Index..  ', F7.3,' dB')
1017    FORMAT(///)
1018    FORMAT(' WRITE UNITIZED AND ZERO-MEANED REF. & PRIM.',
       1' TO DISK  (Y/N) ? ')
1019    FORMAT('+',31X,I3)
1020    FORMAT(/' ENTER  LAMBDA-POST  FULL RANK! (0.901< LAM< 1.000)
       1  (RETURN for const profile)'/' *.***')
1021    FORMAT(/' ENTER  LAMBDA-PRE  FULL RANK! (0.901< LAM< 1.000)
       1 (----- TO END AUTO)'/' *.***')
1022    FORMAT(/'ENTER ORDER  FULL RANK (<= 70) (--  END AUTO)'/' **')
1023    FORMAT(I2)
1025    FORMAT(' AUTO ORDER (Y/N) ?')
1026    FORMAT(/3X,'LAMBDA-PRE',3X,F5.3,6X,'LAMBDA-POST',3X,F5.3,
       1 8X,'# TAP WEIGHTS..  ',I2/)
1027    FORMAT(I3)
1028    FORMAT(' AUTO LAMBDA (Y/N) ?')
1029    FORMAT(' RENTER ORDER  FULL RANK!'/' **')
1032    FORMAT(/' HARDCOPY %FIT (Y/N) ?')
C
C
C       ************************************************************
C       DIMENSION AND DATA STATEMENTS
C
C
        REAL ERROR(512)
        REAL UIN,UOUT,ERR
        REAL BETA(0:71),ZETA(71),UT(71)
        REAL IDATA(512),JDATA(512),LDATA(512)
        REAL LAM,LAMPR,LAMPT,HLAM,HLAMPR,HLAMPT
        INTEGER TP,TP2,TP2J,TP3,TP3J,TP3I,STIM,MLAM
        CHARACTER*1 AUTOD,AUTOL,ANS1
        CHARACTER*5 CHLAM,DHLAM
        CHARACTER*14 IIFILE,JFILE,KFILE,LFILE
C
C
C
C       ************************************************************
C       INITIALIZE  ARRAYS
C
        REAL R(72,71)
C
C
C
C       ************************************************************
C       SET FILE INPUT NAMES
```

```
C
C
      BETA(0)=1.0
      AUTOD='N'
      AUTOL='N'
      DATA IIFILE/'RK1:B#####.EVK'/
      DATA JFILE/'RK1:B#####.EEG'/
      DATA KFILE/'RK1:B#####.SNC'/
      DATA LFILE/'RK1:BS#C  .FIL'/
C
      WRITE(7,1000)
      READ(5,1001)C1
6001  IF(C1.EQ.'S')GOTO 2000
C
      WRITE(7,1002)
      READ(5,1003)IIFILE(5:6)
      WRITE(7,1004)
      READ(5,1003)IIFILE(7:8)
      WRITE(7,1004)
      READ(5,1003)IIFILE(9:10)
      JFILE(5:6)=IIFILE(5:6)
      KFILE(5:6)=IIFILE(5:6)
      JFILE(7:8)=IIFILE(7:8)
      KFILE(7:8)=IIFILE(7:8)
      LFILE(7:7)=IIFILE(6:6)
      LFILE(8:8)='C'
      JFILE(9:10)=IIFILE(9:10)
      KFILE(9:10)=IIFILE(9:10)
      GOTO 2001
C
2000  CONTINUE
      IIFILE(5:6)='BS'
      JFILE(5:6)='RE'
      JFILE(7:8)='FA'
      LFILE(5:6)=IIFILE(5:6)
      IIFILE(11:12)='.S'
      IIFILE(13:14)='PN'
      JFILE(11:12)='.N'
      JFILE(13:14)='SE'
      KFILE(11:12)='.S'
      LFILE(11:12)='.S'
      LFILE(13:14)='IG'
C
      IF(AUTOL.EQ.'Y')GOTO 7003
      IF(AUTOD.EQ.'Y')GOTO 7004
C
      WRITE(7,1002)
      READ(5,1003)KFILE(5:6)
      WRITE(7,1006)
      READ(5,1003)IIFILE(9:10)
      WRITE(7,1007)
      READ(5,1003)JFILE(9:10)
      IIFILE(8:8)='C'
      IIFILE(7:7)=KFILE(6:6)
      KFILE(13:14)=IIFILE(9:10)
      LFILE(9:10)='FP'
      LFILE(7:8)=IIFILE(7:8)
C
7004  WRITE(7,1022)
2007  READ(5,1003)KFILE(9:10)
```

```
      IF(ICHAR(KFILE(10:10)).EQ.32) THEN
      WRITE(7,1029)
      GOTO 2007
      ENDIF
      IF(KFILE(9:10).EQ.'--')GOTO 6999
      TP=((ICHAR(KFILE(9:9))-48)*10)+(ICHAR(KFILE(10:10))-48)
      TP2=TP+2
      TP3=TP+3
      IF(AUTOD.EQ.'Y')GOTO 2001
C
7003  WRITE(7,1021)
2006  READ(5,1011)CHLAM
      IF(ICHAR(CHLAM(5:)).EQ.32) THEN
            WRITE(7,1012)
            GOTO 2006
      ENDIF
      IF(CHLAM.EQ.'-----') GOTO 6999
      KFILE(7:8)=CHLAM(4:5)
C
      WRITE(7,1020)
      READ(5,1011)DHLAM
      IF(ICHAR(DHLAM(5:)).EQ.32) DHLAM=CHLAM
C
      LAM=0.0
      CALL CONVRT (CHLAM,LAM)
      LAMPR=LAM
      HLAMPR=SQRT(LAM)
      CALL CONVRT (DHLAM,LAM)
      LAMPT=LAM
      HLAMPT=SQRT(LAM)
C
2001  CONTINUE
      DO 2002 N=1,512
2002  ERROR(N)=0.0
      DO 2005 I=1,72
      DO 2005 J=1,71
2005  R(I,J)=0.0
C
C     ****************************************************************
C     INPUTS
C
C
      IF((AUTOL.EQ.'Y').OR.(AUTOD.EQ.'Y'))GOTO 6002
      WRITE(7,1009)
      READ(5,1027)NP
      NP2=2.0*NP
      WRITE(7,1013)
      READ(5,1027)STIM
      WRITE(7,1032)
      READ(5,1001)ANS1
      WRITE(7,1018)
      READ(5,1001)C2
      WRITE(7,1025)
      READ(5,1001)AUTOD
      IF(AUTOD.EQ.'Y')GOTO 6002
      WRITE(5,1028)
      READ(5,1001)AUTOL
C
C
C     ************************************************************
```

```
C       READ FILES
C
C
6002    OPEN( UNIT=1, FILE=IIFILE, ACCESS='DIRECT', STATUS='OLD',
        1 RECL=NP, ASSOCIATEVARIABLE=N1REC)
        OPEN( UNIT=2, FILE=JFILE, ACCESS='DIRECT', STATUS='OLD',
        1 RECL=NP, ASSOCIATEVARIABLE=N2REC)
        OPEN( UNIT=3, FILE=KFILE, ACCESS='DIRECT', STATUS='NEW',
        1 RECL=NP, ASSOCIATEVARIABLE=N3REC)
        OPEN( UNIT=4, FILE=LFILE, ACCESS='DIRECT', STATUS='OLD',
        1 RECL=NP, ASSOCIATEVARIABLE=N4REC)
C
        READ(1'1)(IDATA(J),J=1,NP)
        READ(2'1)(JDATA(J),J=1,NP)
        READ(4'1)(LDATA(J),J=1,NP)
C
C
C       ************************************************************
C       ZERO MEAN AND UNIT VARIANCE ALL INPUT FILES
C
C
        CALL NORM(IDATA,512,NP,STIM,0,0)
        CALL NORM(JDATA,512,NP,STIM,0,0)
        CALL NORM(LDATA,512,NP,STIM,1,0)
C
C
C
C       ************************************************************
C       ADAPTIVE NOISE CANCELLOR SYSTOLIC II
C
C
        WRITE(7,1026)LAMPR,LAMPT,TP
        WRITE(7,1015)
        DO 3000 N=1,NP+TP
        WRITE(7,1019)N
        IF(N.LT.STIM) HLAM=HLAMPR
        IF(N.GE.STIM) HLAM=HLAMPT
        RN=FLOAT(N)
        ZETA(1)=1.0
        IF(TP.EQ.0) THEN
                ERROR(N)=IDATA(N)-JDATA(N)
                GOTO 3000
        ENDIF
C
C       **** INPUT LINE UP ****
C
        DO 3005 J=1,TP
        RJ=FLOAT(J)
        RM=RN-(RJ-1.0)*2.0
        IF(RM.LT.1.0) GOTO 3006
        IF(RM.GT.FLOAT(NP)) GOTO 3006
        UT(J)=JDATA(IFIX(RM))
        GOTO 3005
3006    UT(J)=0.0
3005    CONTINUE
        RD=FLOAT(N-TP)
        IF(RD.GT.0.5) GOTO 3007
        UT(TP+1)=0.0
        GOTO 3008
3007    UT(TP+1)=IDATA(IFIX(RD))
```

```
3008    CONTINUE
C
C       **********************
C
C       PROCESS THROUGH ARRAY
C
        DO 4000 J=TP+1,1,-1
        UIN=UT(J)
        IF(J.NE.1) THEN
C
C       ***** INTERNAL CELL *******
C
        DO 4005 I=1,J-1
        J2=TP2-J+1
        TP3I=TP3-I
        P=R(TP3I,J2)
        C=SQRT(1.0-(P*P))
        R(I,J)=HLAM*R(I,J)
        UOUT=C*UIN-P*R(I,J)
        R(I,J)=P*UIN+C*R(I,J)
        R(TP3I,TP2-J)=P
4005    UIN=UOUT
C
        ENDIF
C
C       **** BOUNDARY CELL ******
C
        R(J,J)=HLAM*R(J,J)
        TP3J=TP3-J
        TP2J=TP2-J
        IF(J.LE.TP) THEN
        IF(UIN.NE.0.0) THEN
        R(TP3J,TP2J)=UIN/(SQRT(R(J,J)**2+UIN*UIN))
        C=SQRT(1.0-(R(TP3J,TP2J))**2)
C
        ELSE
        C=1.0
        R(TP3J.TP2J)=0.0
C
C
        ENDIF
        R(J,J)=C*R(J,J)+R(TP3J,TP2J)*UIN
        BETA(J)=C*ZETA(J)
        ZETA(J)=BETA(J-1)
C
C       **** FINAL CELL ****
C
        ELSE
        ZETA(J)=BETA(J-1)
        IF(N.GT.TP) ERROR(N-TP)=UIN*ZETA(J)
C
        ENDIF
C
4000    CONTINUE
C
C
3000    CONTINUE
C
C
C       **** END ADAPTIVE PROCESS ***
```

```
C
C
CALL NORM(ERROR,512,NP,STIM,0,0)
C
C
      IF(C2.NE.'Y')GOTO 220
      WRITE(1'1)(IDATA(J),J=1,NP)
      WRITE(2'1)(JDATA(J),J=1,NP)
220   WRITE(3'1)(ERROR(J),J=1,NP)
C
C     ****************************************************************
C     CALCULATE SQUARED DIFFERENCE BETWEEN PROCESSED SIGNAL AND PURE
C
C
      CALL NORM(LDATA,512,NP,STIM,1,1)
      CALL NORM(JDATA,512,NP,STIM,1,1)
      CALL NORM(ERROR,512,NP,STIM,1,1)
      SUM=0.0
      DIFF=0.0
      RNUM=0.0
      RDEN=0.0
C
      DO 4100 I=STIM,NP
      SUM=SUM+(LDATA(I)*LDATA(I))
      DIFF=DIFF+ABS(ERROR(I)-LDATA(I))**2.0
      RDEN=RDEN+ABS(JDATA(I)-LDATA(I))**2.0
4100  CONTINUE
C
      PRCNT=((SUM-DIFF)/SUM)*100.0
      RMINDX=-10.*LOG10(DIFF/RDEN)
C
      WRITE(7,1016)KFILE,PRCNT,RMINDX
C
      IF(ANS1.EQ.'Y') WRITE(6,1016)KFILE,PRCNT,RMINDX
C
      REWIND 6
C
      CLOSE(UNIT=1)
      CLOSE(UNIT=2)
      CLOSE(UNIT=3)
      CLOSE(UNIT=4)
C
      IF((AUTOD.EQ.'Y').OR.(AUTOL.EQ.'Y'))GOTO 6001
6999  CONTINUE
      END
C
C
      SUBROUTINE CONVRT(CHAR,LAM)
      CHARACTER*5 CHAR
      REAL LAM
      LAM=0.0
      DO 2009 I=1,5
      MLAM=ICHAR(CHAR(I:I))-48
      IF(I.EQ.1) RLAM=FLOAT(MLAM)
      IF(I.GE.3) RLAM=FLOAT(MLAM)/(10**(I-2))
2009  IF(I.NE.2) LAM=LAM+RLAM
      RETURN
      END
```

APPENDIX III

```
C       FOR COLLECTION OF BSAEP DATA FROM C-4 FOR VERIFICATION
C       WRITTEN BY H. DEBRIUN AND R. SCOTT
C       SINGLE CHANNEL.RING BUFFER. ART REJECT.SCHMIDT CHECK
C
C
C
        DIMENSION IDATA(512)
        WRITE(7,1000)
1000    FORMAT(' ENTER OUTPUT FILE NAME'/'  ***:******.***'/)
        CALL ASSIGN(1,'PATDAT-DAT',-1,'NEW')
        WRITE(7,1001)
1001    FORMAT(' ENTER NO. OF STIMULI AND THRESHOLD'/' ***** *****')
        READ(5,1002)ISTIM,ITHRES
1002    FORMAT(I5,1X,I5)
        WRITE(7,1003)
1003    FORMAT(' ENTER SAMPLE RATE'/' *****')
        READ(5,1002)ISAMP
        NDIV=100000./ISAMP
        IRECO=1
        DEFINE FILE 1 (ISTIM,512,U,NREC)
        IERROR=1
        WRITE(1'IRECO) (IDATA(J),J=1,512)
        CALL EVOK1A(IDATA(1),NDIV,ITHRES,IARTP,IERROR)
        GO TO (10,20,30)IERROR
10      WRITE(1'IRECO)(IDATA(J),J=1,512)
        IRECO=IRECO+1
        DO 50 I=2,ISTIM
         CALL ENTER1
         GO TO (48,20,30)IERROR
48       WRITE(1'IRECO) (IDATA(J),J=1,512)
         IRECO=IRECO+1
50      CONTINUE
        CLOSE (UNIT=1)
        WRITE(5,1004)IARTP
1004    FORMAT(' STIMULIS ARTIFACT = ',I5/)
        GO TO 100
C
C
C
C
20      WRITE(7,1005)
1005    FORMAT(' A/D OVERUN ERROR')
        GO TO 100
C
C
C
C
30      WRITE(7,1006)
1006    FORMAT(' INSUFFICIENT PRE-STIMULIS SAMPLES')
C
C
C
100     CONTINUE
        END
```

```
          .TITLE EVOK1A
          .MCALL
          .GLOBL EVOK1A.ENTER1
          ADCSR=170400
          ADBUF=170402
          CLCSR=170404
          CLCBUF=170406
EVOK1A:   CLR ADCSR
          CLR CLCSR
          TST (R5)+
          MOV (R5)+,R4              ;GET DATA ADDRESS
          MOV #BUF,R0               ;RING BUFFER
          MOV R0,LIMIT1
          MOV R0,LIMIT2
          ADD #1024.,LIMIT2
          MOV @(R5)+,NDIV
          MOV @(R5)+,ITHRES
          MOV (R5)+,IARTP
          NEG NDIV
          MOV #204,@#306             ;SET UP H/W INTERUPT FOR SCHMIDT
          MOV #SCHMT1,@#304
          CLRB @#177560              ;CLEAR KEYBD
                                     ;WAIT FOR KEYSTROKE
KEYB:     TSTB @#177560
          BPL KEYB
          CLRB @#177560
          CMP @#177562,#107
          BNE KEYB
                                     ;MAIN PROGRAM
ENTER1:   CLR FLAG2
          MOV #384.,R1
          MOV #BUF,R0
          CLR CLCSR
          MOV NDIV,CLCBUF
          MOV #40,ADCSR             ;CHANNEL 1
          MOV #40405,CLCSR           ;SCHMIDT ENABLE AND CLOCK START
SRVAD0:   TSTB ADCSR
          BPL SRVAD0
          MOV ADBUF,(R0)+
          MOV #401,ADCSR
WAIT1:    TSTB ADCSR
          BPL WAIT1
          MOV #40,ADCSR
          CMP ADBUF,ITHRES
          BGE CONT1
          MOV #177777,-(R0)
          TST (R0)+
CONT1:    CMP R0,LIMIT2                 ;ENOUGH SAMPLES?
          BNE CONT2
          MOV LIMIT1,R0
CONT2:    DEC R1
          BEQ SRVAD1
          BR SRVAD0
SRVAD1:   TSTB ADCSR
          BPL SRVAD1
          MOV ADBUF,(R0)+
          MOV #401,ADCSR
WAIT3:    TSTB ADCSR
          BPL WAIT3
          MOV #40,ADCSR
```

```
          CMP ADBUF,ITHRES
          BGE CONT3
          MOV #177777,-(R0)
          TST (R0)+
CONT3:    CMP R0,LIMIT2
          BNE CONT4
          MOV LIMIT1,R0
CONT4:    TST FLAG2
          BEQ SRVAD1
          DEC R1
          BEQ COMPLT                    ;SUCCESFUL
          BR SRVAD1                     ;NEXT STIMULIS
                                        ;SCHMIDT
SCHMT1:   TST R1
          BNE ERROR2                   ;384 NOT COLLECTED BEFORE SCHMIDT
          INC FLAG2
          MOV #128.,R1
          RTI
                                        ;ERROR ROUTINES
ERROR2:   MOV #3.,@(R5)+
          CLR CLCSR
          TST (SP)+
          TST (SP)+
          CLR ADCSR
          RTS PC
                                          ;UNRING ETC.....
COMPLT:   CLR CLCSR
          CLR ADCSR
          MOV #512.,R1
          MOV R4,R2
          ADD #1024.,R2
TR1:      TST -(R0)
          CMP R0,LIMIT1
          BGE TR0
          ADD #1024.,R0
TR0:      MOV (R0),-(R2)
          DEC R1
          BEQ DON1
          BR TR1
DON1:     MOV #512.,R1
          MOV R4,R2
TSTPRE:   CMP (R2)+,#177777
          BEQ ARTPRE
          DEC R1
          BGT TSTPRE
          RTS PC
ARTPRE:   INC @IARTP
PUNCH:    TSTB @#177564
          BPL PUNCH
          MOVB #7, @#177566
          JMP ENTER1
NDIV:0
ITHRES:0
LIMIT1:0
LIMIT2:0
FLAG2:0
IARTP:0
BUF:0
          .=.+1024.
          .END
```

APPENDIX IV

```
      PROGRAM SPECTM
C
C     S P E C T M . F O R
C
C     Modified by: Robert C. Scott     21-Jan-86
C
C
C     NOTE: Compile with 10 units
C
C
C.....SPECTUM  PROGRAM TO ESTIMATE AND PLOT THE POWER
C.....SPECTRUM OF INPUT DATA FILES,BASED ON AVEREGED
C.....PERIODOGRAMS,OF OVERLAPPING WINDOWS,OF THE INPUT
C.....DATA.THE WINDOWS CAN BE WEIGHTED BY A HANNING
C.....OR RECTANGULAR DATA WINDOW.THE PROGRAM ALSO ESTIMATES
C.....THE FOLLOWING POWER SPECTRUM PARAMETERS:
C.....   MEDIAN FREQUENCY
C.....   STATISTICAL BANDWIDTH
C.....   PERCENT POWER IN THREE SELECTABLE FREQ. BANDS
C.....   HIGH-LOW RATIO;RATIO OF POWER IN HIGH BAND TO LOW BAND
C.....   TOTAL POWER IN SIGNAL
C.....
C.....THE FOLLOWING AMPLITUDE STATISTICS ARE ALSO CALCULATED:
C.....   MEAN RECTFIED EMG VALUE (MRE)
C.....   ROOT MEAN SQUARE VALUE (RMS)
C.....
C.....THE LENGTH OF THE WINDOWS CHOSEN AND THE FFT'S
C.....CALCULATED IS SELECTABLE.THIS ALLOWS VARIABLE
C.....FREQUENCY RESOLUTIONS AND STATISTICAL VARIANCES OF
C.....RESULTING SPECTRUM ESTIMATES.SUCCESSIVE RECORDS ARE
C.....READ FROM THE SPECIFIED DATA FILE AND AN AVERAGE SPECTRUM
C.....IS COMPUTED.
C.....
C.....
      DIMENSION RDAT(512),IDAT(513),SPEC(513),SPECT(513)
      DIMENSION IX(513),IXA(513)
      DIMENSION X(513)
C
      COMPLEX X ,XMN
C
      DIMENSION JWIN(3,2),AXIS(4,4)
      DIMENSION IEL(3,2),IBFREQ(3,2)
      DIMENSION ABPTOT(3),IFILE(8),IHLB(4),IIHLB(4)
C
      CHARACTER*1 ANS,ANS1,ANS3,ANS4,DEC,IZERO,TYP
      CHARACTER*14 MFILE
C
      HAMM(X)=0.54-0.46*COS(TRIG*X)
      BLHR(X)=A0-(A1*COS(TRIG*X))+(A2*COS(TRIG*X))-(A3*COS(TRIG*X))
C
      OPEN(UNIT=9,FILE='PC:FOR009.DAT',STATUS='NEW')
C
      DATA A0,A1,A2,A3/0.35875,0.48829,0.14128,0.01168/
      DATA JWIN(1,1),JWIN(1,2)/'RE','CT'/
      DATA JWIN(2,1),JWIN(2,2)/'HA','MG'/
      DATA JWIN(3,1),JWIN(3,2)/'BL','HR'/
C
      TWOPI=8.0*ATAN(1.0)
      IFILE(8)=0
      MAXM =1024
```

```
      LHM =MAXM/2+1
      IFLAG=0
      IFLAG2=0
      IFLAG3=0
      REDO=0.0
4046  WRITE(7,101)
      NSECTT=0
101   FORMAT(' WHAT IS DATA FILE NAME?'/' ***:******.***')
      READ(5,201)(IFILE(J),J=1,7)
201   FORMAT(7A2)
      IF(REDO.EQ.1.0)GOTO 4047
343   WRITE(7,331)
331   FORMAT(' ENTER DATA TYPE (R OR I)')
      READ(5,332)TYP
332   FORMAT(A1)
3306  WRITE(7,99)
 99   FORMAT(' NO. OF SAMPLES PER RECORD? AND NO. OF RECORDS?'
     1/' ***** *****')
      READ(5,100) N,NTOT
      N2=2*N
100   FORMAT(I5,1X,I5)
      WRITE(7,102)
102   FORMAT(' WHAT IS THE SAMPLE FREQ AND BAND. WIDTH?'/
     +  1X,'*****.**',1X,'*****.**')
      READ(5,202)SAMP,BAND
202   FORMAT(F8.2,1X,F8.2)
      WRITE(7,777)
777   FORMAT( ' DO YOU HAVE A ZERO MEAN SIGNAL?'$)
      READ(5,332) IZERO
C
C
C     ****************************************************************
C     READ IN ANALYSIS PARAMETERS M,IWIN,L
C
      WRITE(7,9999)
9999  FORMAT(' FFT LENGTH ='
      1,5X,' MUST BE A POWER OF 2'/' ****')
 4    READ(5,9997) M
      IF (M.GT.MAXM) WRITE (7,9998)
9998  FORMAT(' M TOO LARGE -- REENTER VALUE ')
      IF(M.GT.MAXM) GO TO 4
9997  FORMAT (I4)
      WRITE(7,9996)
9996  FORMAT(' WINDOW TYPE 1=RECTANGULAR, 2=HAMMING, 3=BLACKMAN
      1-HARRIS '/' *')
      READ(5,9995) IWIN
9995  FORMAT(I1)
5     WRITE(7,9994)
9994  FORMAT(' WINDOW LENGTH = ',
      1/,' ( MUST BE LESS THAN 1000 AND FFT LENGTH, '
      1/,' EVEN AND BE AN INTEGRAL DIVISOR OF REC LENGTH )'/' ****')
      READ(5,9997) L
      IF (L.GT.M) GO TO 5
      LTST=(N/L)*L
      IF(LTST.NE.N) GO TO 5
      IF(L.GT.1000) GO TO 5
4047  WRITE(6,4000)(IFILE(J),J=3,7)
4000  FORMAT(5X,'FILE:',5A2,/)
      WRITE(6,4100)
4100  FORMAT(2X,'REC',4X,'MRE',4X,'RMS',4X,'FC',4X,
```

```
      1'SB',4X,'H/L',6X,'L',6X,'M',6X,'H',8X,'PWR'/)
      SCAL=(17./19.05)*1023.
      DELTAF=SAMP/(1.0*M)
      INUM=(BAND/DELTAF)+1
C
C     ************************************************************
C     NSECT = THE TOTAL NUMBER OF ANALYSIS SECTIONS
C     NP = THE TOTAL NUMBER OF SAMPLES ACTUALLY USED
C     OVERLAP OF 2 TO 1 IS USED ON ADJACENT ANALYSIS SECTIONS
C     NP = N IF(N-L/2)/(L/2) = AN INTEGER
C
C
352   MHLF1 =IFIX((FLOAT(M)/2.0)+1)
      LRS=IFIX(FLOAT(L)/4.0)
      LRS1=2*LRS
      NSECT=IFIX((FLOAT(N)-FLOAT(LRS1))/FLOAT(LRS1))
      IF(TYP.EQ.'R')LRS2=IFIX(2.0*FLOAT(LRS1))
      IF(REDO.EQ.1.)GOTO 4048
      RTOT=2.0*FLOAT(NTOT)*FLOAT(N)/FLOAT(L)
      IF(RTOT.LT.32001.) GOTO 351
      L=IFIX(2.0*FLOAT(L))
      GOTO 352
351   NTOT=IFIX(RTOT)
304   WRITE(7,112)
112   FORMAT(' HOW MANY RECORDS TO BE INCLUDED IN
      1THE SPECTRUM CALCULATION?'/' **')
      READ(5,111)NREC
111   FORMAT(I2)
3307  WRITE(7,106)
106   FORMAT(' WHAT IS THE STARTING REC. NO.?'/' **')
      READ(5,206)IREC
206   FORMAT(I3)
4048  WRITE (7,9899)
9899  FORMAT(' COMPUTING SPECTRUM COEFFICENTS PLEASE WAIT!')
      NSECTT=0
C
C
C     ************************************************************
C     GENERATE WINDOW WEIGHT FOR NORMALIZING SPECTRUM
C
      FL1=FLOAT(L-1)
      TRIG=TWOPI/FL1
      U=0.0
      DO 50 I=1,L
            FI=FLOAT(I-1)
            IF(IWIN.EQ.1.)WD=1.
            IF(IWIN.EQ.2.)WD=HAMM(FI)
            IF(IWIN.EQ.3.)WD=BLHR(FI)
50    U=U+WD*WD
C
      AVGRMS=0.
      AVGMRE=0.
      IREC2=IREC+NREC-1
      IREC1=(NSECT+1)*(IREC-1)+1
      KL=(NREC+1)/NSECT
      NT=NREC+KL
      IF(NREC.EQ.1)NT=1
      DO 70 I=1,MHLF1
         SPECT(I)=0.
 70   CONTINUE
```

```
      DO 200 KJ=1,NT
         KK=NSECT+1
         IF(KJ.NE.NT) GO TO 71
         KK=MOD(NREC+1,NSECT)
         KK=KK+2
71       IF(NT.EQ.1)KK=NSECT+1
      IF(TYP.EQ.'R')GOTO 341
      IF(TYP.EQ.'I')GOTO 342
      GOTO 343
C
C
C     ****************************************************************
C     REAL DATA CONVERSION
C
341   CALL ASSIGN (2,IFILE,14,'RDO')
      DEFINE FILE 2(1,N2,U,M1REC)
      READ(2'IREC1)(RDAT(J),J=1,N)
      CLOSE(UNIT=2)
      RMAX=0.0
      DO 346 J=1,N
346   RMAX=AMAX1(ABS(RDAT(J)),RMAX)
      FSCAL=30000./RMAX
      DO 347 J=1,N
347   IDAT(J)=IFIX(RDAT(J)*FSCAL)
      CALL ASSIGN (3,'DK0:TEMP.DAT',14,'NEW')
      DEFINE FILE 3(1,N,U,M2REC)
      WRITE(3'1)(IDAT(J),J=1,N)
      CLOSE(UNIT=3)
C
342   IF(TYP.EQ.'R')CALL ASSIGN(1,'DK0:TEMP.DAT',14,'OLD')
      IF(TYP.EQ.'I')CALL ASSIGN(1,IFILE,14,'RDO')
      DEFINE FILE 1(NTOT,LRS1,U,JREC)
         DO 205 I=1,KK
            KK1=(I-1)*LRS1+1
            KK2=I*LRS1
            READ(1'IREC1)(IDAT(J),J=KK1,KK2)
            IREC1=IREC1+1
205      CONTINUE
345      IREC1=IREC1-1
         IF(KK.EQ.NSECT+1) GO TO 7
         N=KK*LRS1
         RLP=FLOAT(L)/2.0
         RSECT=(FLOAT(N)-RLP)/RLP
         NSECT=IFIX(RSECT)
   7     CONTINUE
      CLOSE(UNIT=1)
C
C
C     ****************************************************************
C     READ IN DESIRED DATA.
C
        CALL INMAX(IDAT,2000,N,JMAX)
        RMULT=30000./FLOAT(JMAX)
        DO 10 J=1,N
        IDAT(J)=IFIX(FLOAT(IDAT(J))*RMULT)
        IXA(J)=IDAT(J)
10      CONTINUE
        XSUM=0.
        XCOR=409.6*RMULT
C
```

```
C
C       ****************************************************************
C        CALCULATE DATA MEAN.
C
          DO 20 J=1,N
          XSUM=XSUM+IXA(J)
20        CONTINUE
          XMEAN=XSUM/N
C
C
C       ****************************************************************
C       ONLY FOR CALCULATION OF MRE AND RMS
C
          IF(IZERO .EQ. 'N') GO TO 31
          DO 30 J=1,N
          IXA(J)=IXA(J)-XMEAN
30        CONTINUE
C
C
C       ****************************************************************
C       MAKE SIGNAL ZERO MEAN IF DESIRED
C       SET XMN FOR LATER PROCESSING
C
          XMN=CMPLX(XMEAN,XMEAN)
          GO TO 32
C
C        * SET XMN FOR LATER PROCESSING. *
C
31        XMN=CMPLX(0.0,0.0)
C
C
C       ****************************************************************
C        CALCULATE MRE AND RMS.
C
  32      RMRE=0.
          RMS=0.
          DO 40 J=1,N
          RMRE=RMRE+IABS(IXA(J))
          TEMP=IXA(J)
          RMS=RMS+TEMP**2
40        CONTINUE
          RMRE=RMRE/FLOAT(N)/XCOR
          RMS=SQRT(RMS/FLOAT(N))/XCOR
          AVGRMS=AVGRMS+RMS
          AVGMRE=AVGMRE+RMRE
C
C
C       ****************************************************************
C       LOOP TO ACCUMULATE SPECTRA 2 AT A TIME
C
          SS=1.
C
C
C       ****************************************************************
C        READ L/2 SAMPLES TO INITIALIZE BUFFER
C
          L1 =IFIX(FLOAT(L)/2.0)
          NRD=L1
          L2=L1
          CALL GETX(IXA,L2,IDAT,NRD,SS)
```

```
            SS=SS+FLOAT(NRD)
            IMN=L1+1
            KMX=IFIX((FLOAT(NSECT)+1.0)/2.0)
            NSECTP=IFIX(FLOAT(KMX)*2.0)
            NRD=L
            DO 191 I=1,MHLF1
               SPEC(I)=0.
 191        CONTINUE
            DO 190 K=1,KMX
C
C
C       ***************************************************************
C       MOVE DOWN UPPER HALF OF IXA BUFFER
C
            DO 80 I=1,L1
              J=L1+I
              X(I)=CMPLX(FLOAT(IXA(J)),0.)
 80         CONTINUE
            IF(K.NE.KMX .OR.NSECTP.EQ.NSECT) GO TO 95
            DO 90 I=IMN,NRD
              IXA(I)= 0.0
 90         CONTINUE
            NRD=IFIX(FLOAT(L)/2.0)
 95         L2 = 0
            CALL GETX(IXA,L2,IDAT,NRD,SS)
            DO 110 I=1,L1
              J= I+L1
              X(J)=CMPLX(FLOAT(IXA(I)),FLOAT(IXA(J)))-XMN
              X(I)=CMPLX(REAL(X(I)),FLOAT(IXA(I)))-XMN
 110        CONTINUE
            IF(K.NE.KMX.OR.NSECTP.EQ.NSECT) GO TO 130

C
C
C       ************************************************
C       AN ODD NUMBER OF SECTIONS -- ZERO OUT THE SECOND PART
C       ON LAST TIME THROUGH IF HAVE AN ODD NUMBER OF SECTIONS.
C
            DO 120 I=1,L
              X(I)= CMPLX(REAL(X(I)),0.)
 120        CONTINUE
 130        CONTINUE
            SS=SS+FLOAT(NRD)
            FL1 = FLOAT(L-1)
            DO 140 I=1,L
            FI = FLOAT(I-1)
                 IF(IWIN.EQ.1.)TWIND=1.
                 IF(IWIN.EQ.2.)TWIND=HAMM(FI)
                 IF(IWIN.EQ.3.)TWIND=BLHR(FI)
 140        X(I)=X(I)*TWIND
C
            IF (L.EQ.M) GO TO 170
            LP1=L+1
            DO 160 I=LP1,M
              X(I)=(0.,0.)
 160        CONTINUE
 170        CONTINUE
            DO 171 I=1,M
               X(I)=X(I)/XCOR
 171        CONTINUE
```

```
            CALL FFT(X,M,0)
            DO 180 I=2,MHLF1
              J=M+2-I
              SPEC(I)=SPEC(I)+REAL(X(I)*CONJG(X(I))+X(J)*CONJG(X(J)))
 180        CONTINUE
            SPEC(1)=SPEC(1)+REAL(X(1)*CONJG(X(1)))*2
 190      CONTINUE
          DO 195 I=1,MHLF1
             SPECT(I)=SPECT(I)+SPEC(I)
 195      CONTINUE
          NSECTT=NSECTT+NSECT
 200    CONTINUE
C
C
C       ****************************************************************
C       NORMALIZE SPECTRAL ESTIMATE
C
        FNORM = 2. * U *FLOAT(NSECTT)
        DO 210 I=1,MHLF1
                SPECT(I) = SPECT(I)/FNORM
 210    CONTINUE
        AVGRMS=AVGRMS/FLOAT(NT)
        AVGMRE=AVGMRE/FLOAT(NT)
        NP=NREC*N
C
C
C       ************************************************************
C       Save Spectrum Coefficients ?
C
        MHLF2=2*MHLF1
        WRITE(7,3752)
3752    FORMAT(/' SAVE SPECTRUM ? [Y/N] '$)
        READ(5,3753)ANS
        IF(ANS.EQ.'Y') THEN
                WRITE(7,3754)
                READ(5,3755)MFILE
                CALL ASSIGN (3,MFILE,14,'NEW')
                DEFINE FILE 3(1,MHLF2,U,N4REC)
                WRITE(3'1)(SPECT(I),I=1,MHLF1)
                WRITE(7,3756)MFILE(5:),MHLF1
                CLOSE(UNIT=3)
C
        ENDIF
C
3753    FORMAT(A1)
3754    FORMAT('ENTER FILENAME FOR SPECTRUM COEFFS'/' ***:******.***')
3755    FORMAT(A14)
3756    FORMAT(/' FILE.. ',A10,' WITH',I5,' POINTS IS NOW SAVED'/)
C
C
C       ****************************************************************
C       SELECT THE FREQUENCY RANGES FOR THE POWER STATISTICS
C
        IF(REDO.EQ.1.0)GOTO 4049
        IF(IFLAG3.EQ.1) GO TO 3305
550     WRITE(7,103)
103     FORMAT(' WHAT ARE FREQ. BANDS (3) (HZ.)?'/1X,'*****-*****')
        DO 300 I=1,3
300     READ(5,203)(IBFREQ(I,J),J=1,2)
203     FORMAT(I5,1X,I5)
```

```
      WRITE(7,104)
104   FORMAT(' WHAT ARE FREQ. BANDS FOR H/L RATIO (HZ.)?'/
     + 1X,'LOW BAND',2X,'HIGH BAND'/1X,'***** *****  *****  *****')
      READ(5,204)(IIHLB(J),J=1,4)
204   FORMAT(I5,2X,I5,2X,I5,2X,I5)
4049  DO 306 I=1,3
      DO 305 J=1,2
305   IEL(I,J)=((IBFREQ(I,J)*1.)/DELTAF)+1
306   CONTINUE
      IMULT=1
      FL=0
      FH=BAND
      NN1=1
      IF(IFLAG.EQ.0)GO TO 326
      IFLAG=0
 325  CONTINUE
      WRITE(7,7500)
7500  FORMAT(' WHAT IS THE AMP. MULT?'/' *****')
      READ(5,7501)IMULT
7501  FORMAT(I5)
      WRITE(7,5999)
5999  FORMAT(' SELECT FREQ. BAND TO BE PLOTTED.
     1TYPE RETURN FOR FULL BAND.'/' *****.**  *****.**')
      READ(5,5998)FL,FH
5998  FORMAT(F8.2,2X,F8.2)
      IF(FL.EQ.0) GO TO 500
      NN1=IFIX(FL/DELTAF+1.0)
      GO TO 501
 500  NN1=1
 501  IF(FH.EQ.0) GO TO 502
      NN2=FH/DELTAF+1
      INUM=(NN2-NN1)+1
      IF(IFLAG2.EQ.0) BAND=FH-FL
      GO TO 503
 502  FL=0
      FH=BAND
 503  CONTINUE
 326  CONTINUE
      IF(IFLAG2.EQ.1) GO TO 551
      IFLAG2=0
C
C
C     ***************************************************************
C     CALCULATE TOTAL POWER AND STATISTICAL BANDWIDTH
C
3305  CONTINUE
      PTOT=0
      PTOT2=0
      NN3=NN1
      IF(NN1.EQ.1) NN3=2
      DO 311 J=NN3,INUM
      PTOT=PTOT+SPECT(J)*(1/(1.0*FLOAT(M)))
      PTOT2=PTOT2+SPECT(J)**2*(1/(1.0*FLOAT(M)))
311   CONTINUE
      IF(NN1.NE.1) GO TO 700
      PTOT=PTOT*2+SPECT(1)*(1/(1.0*FLOAT(M)))
      PTOT2=PTOT2*4+(SPECT(1)**2*(1/(1.0*FLOAT(M))))
      GO TO 701
700   PTOT=PTOT*2
      PTOT2=PTOT2*4
```

```
701     SB=PTOT**2/PTOT2*SAMP
C
C
C       ************************************************************
C       CALCULATE PERCENT POWER IN SELECTED FREQ.BANDS.
C
        DO 312 J=1,3
        ABPTOT(J) =0.
        BPTOT=0.
        IF(IEL(J,1).LE.1.AND.IEL(J,2).LE.1.) GO TO 312
        IST=IEL(J,1)
        IET=IEL(J,2)
        IBTOT=IET-IST+1
        DO 313 JJ=IST,IET
313     BPTOT=BPTOT+SPECT(JJ)*(1/(1.0*M))
        BPTOT=BPTOT*2
        IF(IST.EQ.1)BPTOT=BPTOT-SPECT(1)*(1/(1.0*M))
        ABPTOT(J)=BPTOT/PTOT*100.
312     CONTINUE
C
C
C       ************************************************************************
C       CALCULATE MEDIAN FREQ.
C
        SUM=0.0
        DO 315 I=NN3,INUM
315     SUM=SUM+2*SPECT(I)*(1/(1.0*M))*(I-1)*(1/(1.0*M))
        FMED=SUM/PTOT*SAMP
C
C
C       ************************************************************
C       CALCULATE HIGH/LOW RATIO
C
        DO 316 J=1,4
316     IHLB(J)=(IIHLB(J)*1.)/DELTAF+1
        RATIO =0.
        HBPTOT=0.
        IF(IHLB(1).LE.1.AND.IHLB(2).LE.1.) GO TO 42
        BBPTOT=0.
        IL1=IHLB(1)-1
        IL2=IHLB(2)-IHLB(1)+1
        IH1=IHLB(3)-1
        IH2=IHLB(4)-IHLB(3)+1
        DO 317 I=1,IL2
317     BBPTOT=BBPTOT+SPECT(IL1+I)*(1/(1.0*M))
        BBPTOT=BBPTOT*2
        IF(IL1.EQ.0)BBPTOT=BBPTOT-SPECT(1)*(1/(1.0*M))
        DO 318 I=1,IH2
318     HBPTOT=HBPTOT+SPECT(IH1+I)*(1/(1.0*M))
        HBPTOT=HBPTOT*2
        IF(IH1.EQ.0)HBPTOT=HBPTOT-SPECT(1)*(1/(1.0*M))
        RATIO=HBPTOT/BBPTOT
        WRITE(6,4101)IREC,AVGMRE,AVGRMS,FMED,SB,RATIO,
        1(ABPTOT(J),J=1,3),PTOT
4101    FORMAT(1X,I3,1X,F7.3,1X,F7.3,1X,F6.1,1X,
        1F6.1,1X,F5.3,3(1X,F6.1),4X,E9.3,/)
        REWIND 6
 551    CONTINUE
        DO 350 I=2,MHLF1
          SPECT(I)=SPECT(I)*2
```

```
 350     CONTINUE
         SCALD=(6.0/19.05)*1023
         CALL REMAX(SPECT,513,MHLF1,XMAX)
         SCAL2=SCALD/XMAX
C
C
C        ****************************************************************
C        PLOT POWER SPECTRUM
C
         IX(1)=0
         XTEMP=0.
         IF(INUM*DELTAF.GT.2000) GO TO 308
         DO 307 I=2,INUM
         XTEMP=XTEMP+DELTAF
C
C        MULT. BY 15 TO AVOID INT.TRUNC.
C
         IX(I)=XTEMP*15.
 307     CONTINUE
         GO TO 309
 308     DO 310 I=2,INUM
            XTEMP=XTEMP+DELTAF
            IX(I)=XTEMP
 310     CONTINUE
 309     SMULT=SCAL/IX(INUM)
         IHASH=IX(INUM)/50
C
         AXIS(1,1)=0.
         AXIS(1,2)=17.
         AXIS(1,3)=0.0
         AXIS(1,4)=0.
         XO=1.
         YO=1.7
         IERAS=1
         CALL AXPLOT(AXIS,4,4,1,SMULT,XO,YO,IHASH,IERAS)
         IERAS=0
         CALL SCALE(IX,1024,1,INUM,XO,SMULT)
         DO 314 J=1,INUM
         K=NN1+J-1
         IDAT(J)=IFIX(SPECT(K)*SCAL2*IMULT)
 314     CONTINUE
         CALL SCALE(IDAT,2000,1,INUM,YO,1.)
         CALL PLOTEK(IX(1),IDAT(1),INUM,1,0,0)
  42     CALL PLOTEK(0,780,1,1,0,0)
         CALL HOME
         WRITE(9,9989) (JWIN(IWIN,J),J=1,2)
9989     FORMAT(' WINDOW TYPE = ',2A2)
         WRITE(9,9988) M,NP,L,SAMP
9988     FORMAT(' M =',I5,5X,'NP =',I5,5X,' L =',I5,
     +   5X,' SAMPLING FREQUENCY =',F8.2)
         WRITE(9,400)(IFILE(J),J=3,7),IREC,IREC2,IMULT
400      FORMAT(5X,'FILE: ',5A2,5X,'RECORDS USED ',I3,'-'
     1,I3,5X,' MULT =',I5)
         WRITE(9,403)((IBFREQ(I,J),J=1,2),I=1,3)
403      FORMAT(/,5X,'FREQ. BANDS:',6X,3(I5,' - ',I5,5X))
         WRITE(9,404)(ABPTOT(J),J=1,3)
404      FORMAT(5X,'% POWER IN BAND:',6X,3(F6.2,12X))
         WRITE(9,405)BAND,PTOT
405      FORMAT(/,5X,'TOTAL POWER IN ',F8.2,' HZ. BAND= ',E9.3)
         WRITE(9,406)FMED,SB
```

```
406     FORMAT(5X,'MEDIAN FREQ.   ',F7.1,' HZ.',
        15X,'STATISTICAL BANDWIDTH ',F7.1,' HZ')
        WRITE(9,407)RATIO,(IIHLB(J),J=1,4)
407     FORMAT(5X,'THE H/L RATIO IS',F7.3,'FOR BANDS',2(I5,'-',I5,2X))
        WRITE(9,408)AVGMRE
408     FORMAT(5X,'THE MRE IS=',F7.3,' MV')
        WRITE(9,409)AVGRMS
409     FORMAT(5X,'THE RMS EEG IS=',F7.3,' MV')
        WRITE(9,410)FL,FH
 410    FORMAT(/////////////////,F8.2,58X,F8.2)
        REWIND 9
C
C
C       ************************************************************
C       RESET SPECTRAL COEFFICENTS
C
        DO 560 J=2,MHLF1
           SPECT(J)=SPECT(J)/2
 560    CONTINUE
        WRITE(7,7502)
7502    FORMAT(' DO YOU WANT TO CHANGE POWER STAT RANGES? [Y/N] ')
        READ(5,3753)ANS
        WRITE(7,7503)
7503    FORMAT(' DO YOU WANT TO CHANGE AMP. MULT. OR
        1 PLOTTED FREQ. BAND? [Y/N] ')
        READ(5,3753)ANS3
        IF(ANS.EQ.'N') IFLAG2=1
        IF(ANS.EQ.'Y') IFLAG2=0
        IF(ANS3.EQ.'Y') IFLAG=1
        IF(ANS.EQ.'Y') GO TO 550
        IF(ANS3.EQ.'Y')GO TO 325
        WRITE(7,107)
107     FORMAT(' ANOTHER DISPLAY [Y/N]')
        READ(5,3753)DEC
        IF(DEC.NE.'Y') GO TO 3304
        IFLAG=0
        IFLAG2=0
        IFLAG3=0
        WRITE(7,3000)
3000    FORMAT(' DO YOU WANT ANY CHANGES? [Y/N]')
        READ(5,3753)ANS4
        IF(ANS4.EQ.'N')IFLAG3=1
        IF(ANS4.EQ.'Y') CLOSE(UNIT=1)
        IF(IFLAG3.EQ.1)GOTO 3307
        GO TO 3306
3304    IF(TYP.EQ.'R')CLOSE(UNIT=1,DISP='DELETE')
        WRITE(7,4031)
4031    FORMAT(' ANOTHER FILE ? [Y/N]')
        READ(5,4032)C2
4032    FORMAT(A1)
        IF(C2.EQ.'N')GOTO 4045
        REDO=1.0
        GOTO 4046
4045    STOP
        END
C
C
C SUBROUTINE TO LOAD WORKING VECTOR WITH DESIRED POINTS.
C
        SUBROUTINE GETX(IX,L2,IDAT,NRD,SS)
```

```
      DIMENSION IX(1),IDAT(1)
        DO 10 I=1,NRD
          IX(I+L2) = IDAT(I+SS-1)
10      CONTINUE
      RETURN
      END
```

REFERENCES

AUNON, J.I., C.D. McGILLEM, "Techniques for Processing Single Evoked Potentials", Proc. San Diego Biomedical Symposium, San Diego, 1975.

AUNON, J.I., C.D. McGILLEM and D.G. CHILDERS, "Signal Processing in Evoked Potential Research : Averaging and Modelling", Critical Reviews in Bioengineering, vol.5, no.4, p. 327, July 1981.

BERGER, H., "Uber des Elekrenephalogram des Menschen", Arch Psychiatr Nerven, vol.87, pp. 527-570, 1929.

BHUWAN, P.G., O.N. MARKAND, P.F. BUSTION, "Brainstem Auditory Evoked Responses in Hereditary Motorsensory Neuropathy: Site of Origin of Wave II", Neurology, vol.32, pp. 1017-1019, 1982.

BOSTON, J.R., "Spectra of Auditory Brainstem Responses and Spontaneous EEG", IEEE Trans. on Biomedical Engineering, vol. BME-28, no.4, April 1981

CHILDERS, D.G., "Evoked responses: Electrogenesis, Models, Methodology, and Wavefront reconstruction and Tracking Analysis", Proceedings of the IEEE, vol.65, no.5, pp. 611-626, May 1977.

CHIAPPA, K.H., Evoked Potentials in Clinical Medicine, Raven Press, 1983.

COHEN, B.A., and A. SANCES Jr., "Stationarity of the Human Electroencephalogram". Med. & Biol. Eng. & Comp., vol.15, pp. 513-518, Sept. 1977.

COPPOLA, R., R. TABOR, and M.S. BUCHSBAUM, "Signal to Noise Ratio and Response Variability Measurements in Single Trial Evoked Potentials", Electroeceph. and Clin. Neuro., vol. 44, pp. 211-222, 1978.

DAWSON, G.D., "Cerebral Responses to Electrical Stimulation of Peripheral Nerve in Man", J. Neurol. vol.10, pp. 134-140, 1947.

DAWSON, G.D., " A Summation technique for detection of small evoked potentials", Electroencephalography and Clinical Neurophysiology, vol.6, pp. 65-84, 1954.

de WEERD, J.P.C. and J.I. KAP, "A Posteriori Time-Varying Filtering of Averaged Evoked Potentials", Biol. Cybernetics, vol.41, pp. 223-234, 1981.

DOYLE, D., "A Proposed Methodology for the Evaluation of the Weiner Filtering Method of Evoked Potential Estimation", Electroencephalography and Clinical Neurophysiology, vol.43, pp. 749-751, 1977.

FERRARA, E.R. Jr. and B. WIDROW, "Multichannel Adaptive Filtering for Signal Enhancement", IEEE Transactions on Acoustics, Speech, and Signal Processing, vol.ASSP-29, no.3, June 1981.

FRIDMAN, J., E.R. JOHN, M. BERGEBON, J.B. KAISER, and H.W. BAIRD. "Application of Digital Filtering and Automatic Peak Detection to Brain Stem Auditory Evoked Potential", Electroencephalography and Clinical Neurophysiology, vol.53, pp. 405-416, 1982.

GENTLEMAN, W.M. and H.T. KUNG, "Matrix Triangularization by Systolic Arrays", Proc. SPIE, vol. 298, Real Time Signal Processing IV, pp. 298-303, 1981.

GIVENS, W., "Computation of Plane Unitary Rotations Transforming a General Matrix to Triangular Form", J. Soc. Industrial and Applied Mathematics, vol.6, no.7, pp. 26-50, March 1958.

GREINER, Y., "Time-Dependent Modelling of Non-Stationary Signals", IEEE Transactions on Acoustics, Speech, and Signal Processing, vol.ASSP-31, no.4, August 1983.

HARRISON, W.A., J.S. LIM and E. SINGER, "A New Application of Adaptive Noise Cancellation", IEEE Transactions on Acoustics, Speech, and Signal Processing, vol.ASSP-34, no.1, June 1986.

HAYKIN, S., Adaptive Filter Theory, New Jersey, Prentice Hall, 1986.

HUNT, E., "Mathematical Model of the Event Related Potential", Psychophysiology, vol.22, no.4, pp. 395-402, 1985.

KAVEH, M., S. BRUZZONE and F. TORRES, "A New Method for The Estimation of Averaged Evoked Responses", IEEE Transactions on Man, Systems and Cybernetics, vol.8, no.5, pp. 414-417, 1978.

KILOH, L.G., A.J. McCOMAS, J.W. OSSELTON, and A.R.M. UPTON, Clinical Electroencephalography, London, Butterworths, 1981.

KUNG, H.T., "Why Systolic Architecture?", IEEE Computer, vol. 15, pp. 37-46, 1982.

KUNG, H.T. and C.E. LEISERSON, "Systolic Arrays", Society for Industrial and Applied Mathematics, pp. 256-282, 1979.

LANDAU, I.D., "A Feedback System Approach to Adaptive Filtering", IEEE Trans. Information Theory, vol. IT-30, Special Issue on Linear Adaptive Filtering, pp. 251-262, 1984.

MADHAVAN,G.P."Adaptive Filtering and Pattern Recognition of Evoked Potentials", Doctoral Thesis, 1985.

MADHAVAN, G., H. deBRUIN and A.R.M. UPTON, "Evoked Potential Processing & Pattern Recognition", 6th Conference IEEE (EMBS): Frontiers of Engineering and Computing in Health Care, Los Angeles, Sept. 1984.

McGILLEM, C.D. , J.I.AUNON and C.A. POMALAZA, "Improved Waveform Estimation Procedures for Event-Related Potentials", IEEE Transactions on Biomedical Engineering, vol.32, no.6, p. 321, June 1985.

McWHIRTER, J.G., "Recursive Least Squares Minimization using Systolic Arrays", Proc. SPIE, vol. 431, Real Time Signal Processing VI, pp. 105-112, 1985.

NASHI, H., "A Maximum Likelihood Method to Estimate Evoked EEG Potentials", Doctoral Thesis, McGill University , 1985.

NASHI, H., "A Maximum Likelihood Method to Estimate Evoked EEG Potentials", IEEE Transactions in Biomedical Engineering, vol. BME-33 pp. 1087-1095, Dec. 1986.

RAUNER, H., W. WOLF and U. APPEL, "New Perspectives to Noise Reduction in Evoked Potential Processing", Signal Processing II, Elsevier Publishers, 1983.

ROGERS, L. J., "Ipsilateral and Contralateral Correlations Between EEG and EP Principal Components", Electroencephalography and Clinical Neurophysiology, vol.50, pp. 441-448, 1980.

SORENSON, H.W., "Least Squares Estimation: from Gauss to Kalman", IEEE Spectrum, pp. 63-68, July 1970.

SPEISER, J.M. and H.S. WHITEHOUSE, "A Review of Signal Processing with Systolic Arrays", Proc. SPIE, vol. 431, Real Time Signal Processing VI, pp. 2-6, 1983.

WALTER, D.O., "A posteriori "Weiner Filtering" of Averaged Evoked Potentials", Electroencephalography and Clinical Neurophysiolgy, suppl. 27, pp.61-70, 1969

WEBSTER, J.G. (ed.), Medical Instrumentation, Boston Houghton Mifflin, 1978.

WIDROW, B., J.R. GLOVER Jr., J.M. McCOOL, J. KAVNITZ, C.S. WILLIAMS, R.H. HEARN, J.R. ZEIDLER, E. DONG, and R.C. GOODLIN, "Adaptive Noise Cancelling: Principles and Applications", Proc. IEEE, vol. 63, no. 12, pp. 1692-1716, Dec. 1975.

WINSKI, R. and N.M. ALLINSON, "Adaptive Processing of Brain Evoked Potentials", IEEE Colloquium on Adaptive Processing and Biomedical Applications, London, 1984.

WOODY, C.D., "Characteristics of an Adaptive Filter for the Analysis of Variable Latency Neuroelectric Signals", Med. and Biol. Eng., vol. 5, pp. 539-553, 1967.

YU, Kai-Bor and C.D. McGILLEM, "Optimum Filters for Estimating Evoked Potential Waveforms", IEEE Transactions on Biomedical Engineering, Vol. 30, no. 11, p.730, Nov. 1983.