AUTOMATIC LMS EQUALIZER ALGORITHMS WITH FAST RATE OF CONVERGENCE

by

Nazan HEKIM

Submitted to the Faculty of the
School of Engineering in Partial Fulfillment
of the Requirements for the Degree of
MASTER OF SCIENCE
in
ELECTRICAL ENGINEERING

Boğaziçi University
June 1982

This thesis has been approved by:

Doç.Dr.Yorgo Istefanapulos:
(Thesis Supervisor)

Prof.Dr.Erdal Panayırcı :

Prof.Dr.Cem Göknar :

ACKNOWLEDGEMENT

This thesis has been prepared for the partial fulfillment of the requirements of Boğaziçi University, School of Engineering for the degree of Master of Science in Electrical Engineering and been supported by Electronic Research Department of Marmara Research Institute.

I wish thank to Doç.Dr.Yorgo Istefanapulos, thesis supervisor, for his encouragement and guidance in accomplishment of this work; to Prof.Dr.Erdal Panayırcı and Prof.Dr.Cem Göknar for their useful critiques and helpful suggestions.

I am also indebted to Ms.Talin Taşçıyan for her helps during the presentation of this work.

ABSTRACT

In high speed voiceband modems, as in many other data transmission systems, linear distortion and additive noise are important degrading factors. The tapped-delay-line equalizers designed to minimize the mean-square-error cost function are commonly used to compensate these undesired effects. Among the several algorithms which minimizes the mean-square-error cost function stochastic gradient algorithm is the most popular because of its simplicity in implementation. However, for highly distorted channels stochastic gradient algorithm converges slowly and, therefore, a long training period which causes a fall in the overall performance of the system is required. Instead, more complicated algorithms with faster rate of convergence have been developed in the last years: Kalman/Godard, Fast Kalman and lattice algorithms.

In this thesis, the rate of convergence of stochastic gradient, Kalman/Godard and Fast Kalman algorithms are analyzed and their computational complexities are examined. The analysis is extended to the complex domain in order to cover equalization of quadrature-amplitude-modulated signals. Furthermore, a computer program package which simulates several telephone channels, a quadrature-amplitude-modulation system and the equalization algorithms is written. Them, the performance of the different equalization algorithms over a wide range of channels are compared.

ÖZET

Ses bandında çalışan yüksek hızlı veri iletişim sistemlerinde kalite düşürücü etkenlerin başlıcaları doğrusal bozulma ve gürültüdür. İstenmeyen bu etkenlerin giderilmesinde kullanılan ayarlanabilir bellek katsayılı dengeleyicilerin tasarımında yanılgı-karesi-ortalaması maliyet işlevinin enküçültülmesi amaçlanır. Bu amaca ulaştıran algoritmalar içerisinde uygulama basitliği açısından en yaygın olanı istatistiksel gradyan yöntemidir. Ne var ki çok bozuk kanallar için istatistiksel gradyan yönteminin yakınsama hızı çok düşüktür. Dolayısıyla dengeleyici için sistemin genel davranışı bozucu uzun bir adaptasyon süresi gerekir. Son yıllarda geliştirilen Kalman/Godard, hızlı Kalman ve kafes algoritmalarının, daha karmaşık olmalarına rağmen, yakınsama hızları çok yüksektir.

Bu tez çalışmasında, istatistiksel gradyan, Kalman/Godaard ve hızlı Kalman algoritmalarının yakınsama hızları ve hesaplama karmaşıklıkları incelendi. Analiz, dikgen-genlik-modülasyonunda dengeleme problemini kapsayacak biçimde kompleks değişkenler kullanılarak yürütüldü. Ayrıca, çeşitli telefon kanallarını, dikgen-genlik modülasyon sistemini ve sözkonusu dengeleyici algoritmalarını gerçekleştiren bir bilgisayar programları paketi hazırlandı. Son olarak her üç dengeleyici algoritma çeşitli telefon kanalları üzerinde sınandı.

TABLE OF CONTENTS

INTRODUCTION

As is well known, data transmission over the existing telephone networks requires equalization to achieve reliable performance in the presence of linear distortion and additive noise. Tapped-delay-line equalizers employing various adjusting algorithms for their taps take place in high speed voice band modems. As will be mentioned in Chapter I these equalization algorithms differ in the operational definition of equalization. However, minimization of the mean-squared-error of the transmission system has been proven to be the best equalization strategy(1-3). The algorithms in this class also show differences in the computational method used. Among the various algorithms the stochastic gradient algorithm is a wide-spread technique. There are several works investigating the performance of the stochastic gradient adaptive equalization algorithm among which Gersho's, Widrow's and Proakis's must be emphasized(1-6).

The stochastic gradient algorithm, although being very popular, has the disadvantage that it has a very slow settling time in some cases. The convergence rate of the stochostic gradient algorithm, as demonstrated by Gittlin and Ungerboeck depends on the eigenvalues of the input correlation matrix, and the number of taps of the equalizer. High distortion causes the eigenvalues of the input correlation matrix to deviate from each other and, as this happens, it becomes more and more difficult to control the convergence rate of the stochastic gradient algorithm. Therefore, a long training period which constitutes a considerable portion of the total transmission time becomes necessary.

If the dependence of the equalization performance on the eigenvalues can be eliminated, then the algorithm can be accelerated. To orthogonalize the eigenvalues the signals sent during the training period can be chosen orthogonal to each other(7). On the other hand, in the so called Kalman/ Godard algorithm estimating the inverse of the correlation matrix in each iteration results in fast convergence. The name of the algorithm is due to its first derivation by Godard with the application of Kalman filtering approach to the equalization problem(15). The Kalman/Godard algorithm uses all the past information available at each step. Thus, it is at least intuitively clear that it is expected to give better performance than the conventional stochastic gradient algorithm which uses only the current information. However, in the recursive estimation of the inverse correlation matrix, an NxN matrix where N is the number of taps must be computed and stored at each recursion. Repeatation of such a large dimensional matrix computation means a considerable complexity in both computer simulation and hardware implementation.

The Fast Kalman algorithm is mathematically equivalent to the Kalman/Godard algorithm. The difference is only in the computation of the inverse correlation matrix. The improvement is based on the fact that at each iteration the number of new samples entering and the old samples leaving the equalizer is not N but only one. Hence, Fast Kalman algorithm gives the same performance as the Kalman/Godard algorithm does while using N-dimensional vectors at most. Consequently the Fast Kalman algorithm with its acceptable computational complexity and very fast rate of convergence is the most efficient one.

In this thesis the three algorithms mentioned above are analyzed under some assumptions. More detailed mathematical treatments of the recursive least squares algorithms can be found in the literature, especially in the work of Ljung (18). The results of both analyses are almost the same. This work also includes a computer simulation to test the theoretical results. Various channels with different attenuation and group delay characteristics are simulated. The modulation scheme used is quadrature-amplitude-modulation which enables two dimensional data transmission.

Chapter I consists of the statement of the equalization problem, classification of commonly used equalization algorithms and the complex domain description of the quadrature-amplitude-modulation system.

In Chapter II the structure of the tapped-delay-line equalizer is introduced. The optimum solution for the tap coefficients can be found by setting the gradient of the mean-square-error performance function to zero with respect to the tap coefficient vector. This chapter also includes the derivation of the stochastic gradient algorithm.

Convergence of the stochastic gradient algorithm is controlled by a parameter called step-size. In Chapter III some bounds on the step-size for the tap coefficients to converge to their optimum values and for the mean-square-error to attain its minimum value are developed. Then, these bounds are elaborated to obtain the fastest convergence possible. It should be noted that in some cases to chose the best value for step-size is itself a difficult problem.

Derivation of the Kalman/Godard algorithm and its convergence analysis constitutes the subject of Chapter IV. Another derivation based on Kalman filtering approach is given in Appendix IV. It is seen that as soon as the number of iterations exceeds the number of taps the mean-square-error falls into a very small value. However the required number of multiplications in the Kalman/Godard algorithm is so large that this may be a big price for fast convergence. But the recursive matrix inversion can be modified in a simplifying manner. This leads to the Fast Kalman algorithm, which is the subject of Chapter V.

In Chapter VI the programs in the simulation package are introduced and the characteristics of the simulated transmisson system is given. This chapter also includes the results of the Stochastic Gradient, Kalman and Fast Kalman adaptive algorithms over seven different telephone channels.

Finally, the discussion about the simulation results takes place in Chapter VII. A manual for computer programs can be found in Appendix 5.

I. CHANNEL EQUALIZATION

One of the main degrading factors in data tranmsmission through the telephone lines is time dispersion caused by the deviation of the channel frequency response from the ideal characteristics of constant amplitude and linear phase. In order to provide reliable transmission these nonideal characteristics must be compensated by additional filtering. Thus, equalization constitutes an important part of the data transmission.

I.1. INTERSYMBOL INTERFERENCE AND EQUALIZATION

In Fig.I.1. a data transmission system is illustrated.

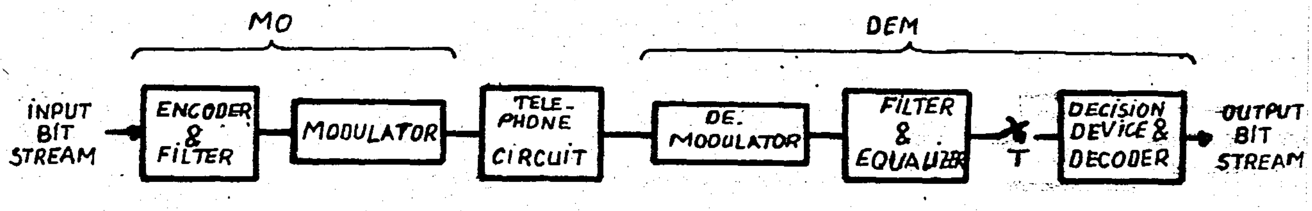

Fig (I.1) Data Transmission System

The input bits are encoded and modulated according to a certaion modulation scheme. After they are transmitted through the channel the reverse operation takes place. In efficient digital communication systems the effect of each symbol transmitted over a time dispersive channel extends beyond the time interval used to represent that symbol. The overlap of the received symbols is called intersymbol interference. Although intersymbol interference arises in all pulse modulated systems it is most easily described in the case of pulse amplitude modulation (PAM).

In Fig. I.2. a simple PAM system (except encoder/decoder section) is shown.

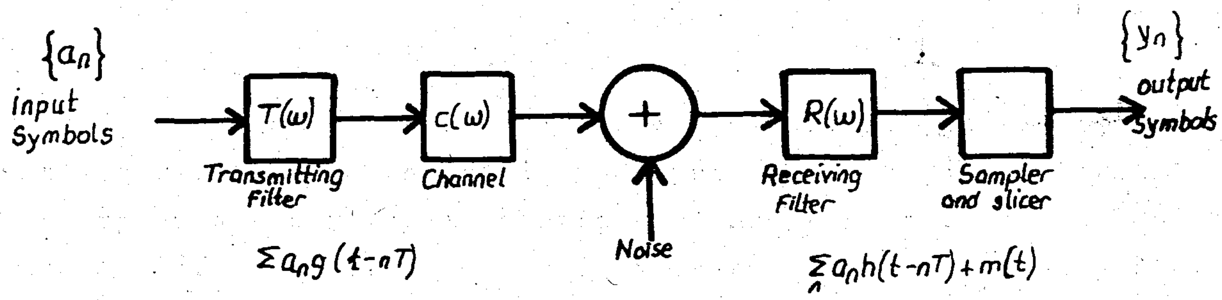

Fig (I.2) Pulse-Amplitude-Modulation System.

The transmitting filter/channel/receiving filter cascade (TCR) determines the shape of the information carrying pulses:

$$h(t) = \frac{1}{2} \int_{-\infty}^{\infty} T(\omega)C(\omega)R(\omega)e^{j\omega t}\, d\omega \qquad (I.1)$$

Then, the output is

$$y(t) = \sum_{i} a(i)h(t-iT) + \eta(t) \qquad (I.2)$$

where T is the sampling time and $\eta(t)$ is the additive noise. At $t=kT + t_o$ (to is due to the constant delay in TCR) the sampled output at the receiver corresponding to the desired output a(k) is

$$y(kT + t_o) = \sum_i a(i)h(kT + t_o - iT) + \eta(kT + t_o) \quad (I.3)$$

Let k denote $kT + t_o$, then (I.3) becomes

$$y(k) = \sum_i a(i)h(k-i) + \eta(k) \quad (I.4)$$

Note that the actual output can also be expressed as:

$$y(k) = h(0) \{ a(k) + \frac{1}{h(0)} \sum_{\substack{i \\ i \neq k}} a(i)h(k-i) + \frac{\eta(k)}{h(0)} \} \quad (I.5)$$

The gain factor h(0) can be adjusted to unity. Then, the first term is the desired output while the second term represents intersymbol interference and the last term is due to noise. Whenever the sum of these two extra terms exceeds the quantization interval an error occurs.

From (I.5) it is seen that the intersymbol interference is zero if and only if $h(k) = 0$ for all $k \neq 0$. This means that the TCR impulse response must have zero crossings at T-spaced intervals. The last requirement is called Nyquist first criterion and is also equivalent to

$$H_{eq}(\omega) = \text{constant for } |f| \leq \frac{1}{2T} \quad (I.6)$$

in the frequency domain, where

$$H(\omega) = T(\omega)C(\omega)R(\omega) \quad (I.7)$$

If the channel transfer function were known a priori, then it would be possible to completely eliminate the intersymbol interference by just realizing its inverse. One method may be to design equalizers based on average channel characteristics. However, the variations in the characteristics within a class of channels is generally large. On the other hand these characteristics may change in time. Therefore, an equalization strategy in which adaptation into the specific channel characteristics is achieved, becomes necessary.

I.2. EQUALIZATION STRATEGIES

The tapped-delay-line (TDL) equalizer, whose structure will be examined in the next chapter can be designed to follow many different strategies.

In the case of zero-forcing equalizer the aim is to force the output of the equalizer/channel combination to zero at all but one of the N T-spaced instants in the span of the equalizer. This equalizer minimizes the peak distortion. However, it neglects the noise effects and operates effectively if the peak distortion before equalization is less than 100 percent.

Least-Mean-Square (LMS) equalizers are designed to minimize the mean-square-error which is due to both intersymbol interference and noise. The LMS equalizers can also be divided into two classes as automotic and adaptive. In automatic equalization a known data sequence is sent prior to data transmission. During such a training period the equalizer's parameters are iteratively adjusted to minimize the mean square of the error between the training sequence synchronously generated at the receiver and the output of the equalizer. As long as the channel characteristics are time-invariant automotic equalization is adequate. If the channel characteristics are time varying equalizer must continuously adapt it self so as to track these changes. Adaptive LMS equalizer operates in this manner. Here, the desired response is taken to be the output of a decision device operating on the output of the equalizer. In other words, in the so called decision-directed mode of the adaptive equalizers the error is defined between the output of the equalizer and the estimate of the transmitted symbol.

Three types of equalizers mentioned above are all linear. On certain channel characteristics nonlinear decision feedback equalizers give better performance. The decision feedback equalizers consist of a forward path which is a usual TDL equalizer and a backward path along which the decisions made on the equalized signal are fed back. The purpose of the backward part is to eliminate the effects of the past symbols during the next steps.

Finally the tap spacing can be chosen less than the sampling time in linear TDL equalizers. This configuration is called fractionally spaced equalizer and has the advantage of being less sensitive to the sampling phase.

Finnally, equalization can be carried out in either baseband (after demodulation) or in passband (before demodulation). The subject of this work is restricted by the LMS automatic equalization in baseband.

I.3. QUADRATURE-AMPLITUDE-MODULATION (QAM) SYSTEMS

In quadrature-amplitude-modulation (QAM) two independent data sequences are used to separately modulate the in-phase and quadrature components of a sinusoidal carrier. Two double-sideband supressed carrier AM signals are superimposed on each other at the transmitter and separated at the receiver. In Fig. I.3. a QAM transmission system is shown:

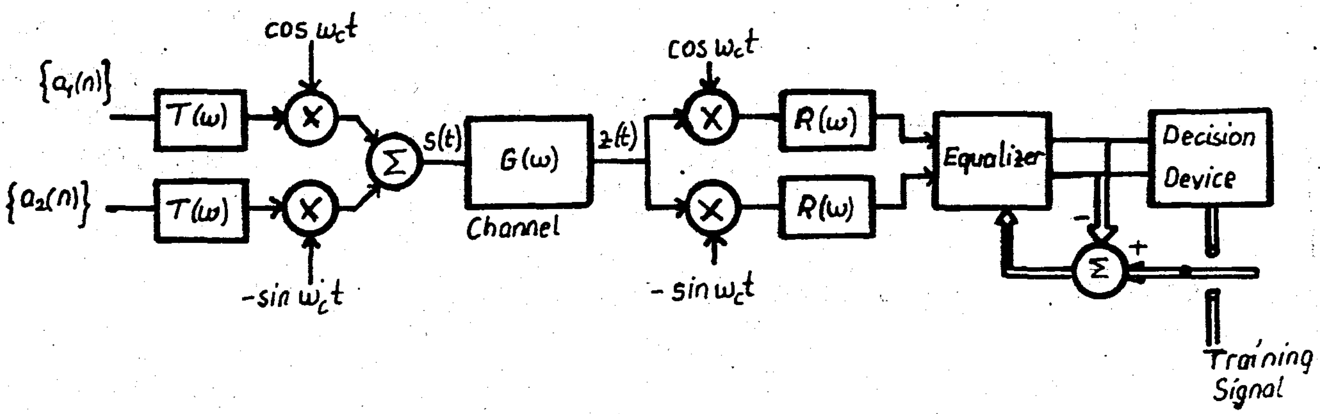

Fig. (I.3) QAM Transmission System

In the system above the transmitted waveform is

$$s(t) = \sum_n a_1(n)p(t-nT)\cos\omega_c t - \sum_n a_2(n)p(t-nT)\sin\omega_c t \tag{I.8}$$

where $\{a_1(n)\}$ and $\{a_2(n)\}$ are data sequences to be transmitted, $1/T$ is the signalling rate, f_c is the carrying frequency and $p(t)$ is the pulse shaped by the transmitting filter. This signal can also be expressed as:

$$s(t) = b_1(t)\cos\omega_c t - b_2(t)\sin\omega_c t \tag{I.9}$$

where

$$b_1(t) = \sum_n a_1(n)p(t-nT)$$

$$b_2(t) = \sum_n a_2(n)p(t-nT)$$

Let $g(t)$ be the impulse response of the channel. Then the output of the channel is

$$z(t) = \int^{\infty} g(\tau)s(t-\tau)d\tau \tag{I.10}$$

or,

$$z(t) = -\int_{-\infty}^{\infty} \check{g}(\tau)\check{s}(t-\tau)d\tau \tag{I.11}$$

where $\check{g}(t)$ and $\check{s}(t)$ are the Hilbert transforms of g(t) and s(t), respectively. It is also possible to write

$$z(t) = \frac{1}{2}\int_{-\infty}^{\infty} g(\tau)s(t-\tau)d\tau - \frac{1}{2}\int_{-\infty}^{\infty} \check{g}(\tau)\check{s}(t-\tau)d\tau \tag{I.12}$$

Since

$$\check{s}(t) = b_1(t)\sin\omega_c t + b_2(t)\cos\omega_c t \tag{I.13}$$

(I.12) takes the form

$$\begin{aligned} z(t) = &\frac{1}{2}\int_{-\infty}^{\infty} g(\tau)b_1(t-\tau)\cos\omega_c(t-\tau)d\tau \\ &- \frac{1}{2}\int_{-\infty}^{\infty} \check{g}(\tau)b_1(t-\tau)\sin\omega_c(t-\tau)d\tau \\ &- \frac{1}{2}\int_{-\infty}^{\infty} g(\tau)b_2(t-\tau)\sin\omega_c(t-\tau)d\tau \\ &- \frac{1}{2}\int_{-\infty}^{\infty} \check{g}(\tau)b_2(t-\tau)\cos\omega_c(t-\tau)d\tau \end{aligned} \tag{I.14}$$

It is convenient to rewrite (I.14) as

$$\begin{aligned} z(t) = \cos\omega_c t &\int_{-\infty}^{\infty}\Big[b_1(t-\tau)\left[g(\tau)\cos\omega_c\tau + \check{g}(\tau)\sin\omega_c\tau\right] \\ &\quad + b_2(t-\tau)\left[g(\tau)\sin\omega_c\tau - \check{g}(\tau)\cos\omega_c\tau\right]\Big]d\tau \\ - \sin\omega_c t &\int_{-\infty}^{\infty}\Big[b_1(t-\tau)\left[-g(\tau)\sin\omega_c\tau + \check{g}(\tau)\cos\omega_c\tau\right] \\ &\quad + b_2(t-\tau)\left[\check{g}(\tau)\sin\omega_c\tau + g(\tau)\cos\omega_c\tau\right]\Big]d\tau \end{aligned} \tag{I.15}$$

After the signal z(t) is demodulated and filtered to remove frequencies greater than f_c the following baseband signals are obtained:

$$\text{in-phase:} \int_{-\infty}^{\infty} b_1(t-\tau) \left[g(\tau)\cos\omega_c\tau + \check{g}(\tau)\sin\omega_c\tau\right] d\tau$$

$$+ \int_{-\infty}^{\infty} b_2(t-\tau) \left[g(\tau)\sin\omega_c\tau - \check{g}(\tau)\cos\omega_c\tau\right] d\tau \quad (I.16)$$

$$\text{quadrature:} \int_{-\infty}^{\infty} b_1(t-\tau) \left[-g(\tau)\sin\omega_c\tau + \check{g}(\tau)\cos\omega_c\tau\right] d\tau$$

$$\int_{-\infty}^{\infty} b_2(t-\tau) \left[\check{g}(\tau)\sin\omega_c\tau + g(\tau)\cos\omega_c\tau\right] d\tau \quad (I.17)$$

The two signals to be equalized are those given by (I.16) and (I.17). At this point it will be helpfull to use complex notation. Let

$$a(n) = a_1(n) + ja_2(n) \quad (I.18)$$

and

$$s(t) = s(t) + j\check{s}(t) \quad (I.18)$$

and also

$$b(t) = b_1(t) + jb_2(t) \quad (I.19)$$

It is clear from (I.8) and (I.9) that

$$b(t) = \sum_n a(n)p(t-nT) \quad (I.20)$$

Let $x_1(t)$ denote the in-phase signal and $x_2(t)$ denote the quadrature signal at the output of the receiver. The complex signal x(t) is defined by

$$x(t) = x_1(t) + jx_2(t) \quad (I.21)$$

Then, if $g_1(t)$ and $g_2(t)$ are chosen such that

$$g_1(t) = g(t)\cos\omega_c t + \check{g}(t)\sin\omega_c t \qquad (I.22)$$

$$g_2(t) = -g(t)\sin\omega_c t + \check{g}(t)\cos\omega_c t \qquad (I.23)$$

$x(t)$ becomes

$$x(t) = \tilde{g}(t) * b(t) \qquad (I.24)$$

where

$$\tilde{g}(t) = g_1(t) + jg_2(t) \qquad (I.25)$$

Thus, although the signals are real using complex notation a complex representation of the QAM system is obtained. Now the sampled complex input signal $x(t)$ can be equalized by a complex equalizer. Then the real equalizer coefficients help to eliminate the intersymbol interference in each part of the transmission system while the imaginery coefficients counteract the cross interference between two parts, caused by assymmetry in the channel characteristics, around the carrier frequency.

II. STOCHASTIC GRADIENT ALGORITHM FOR AUTOMATIC CHANNEL EQUALIZATION

II.1. TDL EQUALIZER

The tapped-delay-line (TDL) equalizer, illustrated in Fig. II.1. consists of a delay line, (2N+1)-taps and a summer. The equalizer input sequence {$\underline{x}(n)$} is delayed through the line by an amount T and each step, where 1/T is the signalling rate. Then each such delayed sample x(n-m) is multiplied by a corresponding weighting coefficient c(m). Finally, these terms are summed to form the equalizer output sequence {y(n)}.

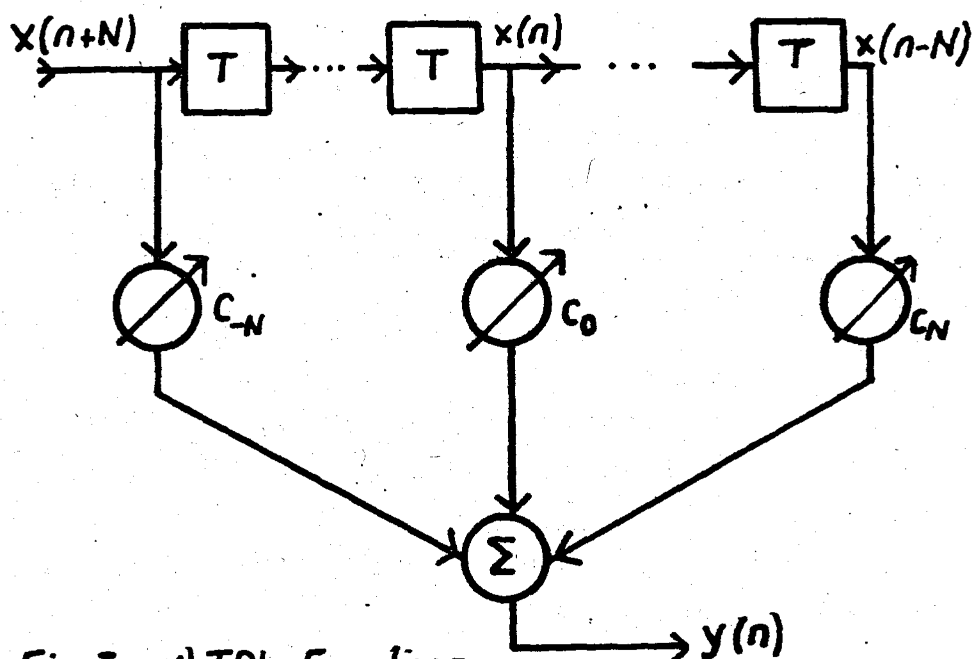

Fig I. 1) TDL Equalizer

The output of the Equalizer is

$$y(n) = \sum_{i=-N}^{N} c_i^* \, x(n-i) \qquad (*) \qquad \text{(II.1)}$$

In matrix notation y(n) can be expressed as

$$y(n) = \underline{c}^* \underline{x}(n) \qquad \text{(II.2)}$$

where $\underline{c}$ and $\underline{x}(n)$ are the following (2N+1)-vectors:

$$\underline{c} = \begin{bmatrix} c_{-N} \\ \vdots \\ c_o \\ \vdots \\ c_N \end{bmatrix} , \qquad \underline{x}(n) = \begin{bmatrix} x(n+N) \\ \vdots \\ x(n) \\ \vdots \\ x(n-N) \end{bmatrix}$$

The problem is to find the coefficient vector $\underline{c}$ of the equalizer in such a way as to minimize the mean-square-error (MSE)

$$\xi^2(n) = E\{||e(n)||^2\} \qquad \text{(II.3)}$$

where e(n) is the error between the output of the equalizer and the actual data symbol a(n), i.e.,

$$e(n) = a(n) - y(n) \qquad \text{(II.4)}$$

The input to the equalizer, or, in other words, the output of the receiver can be expressed as

$$x(n) = a(n) * h(n) + \omega(n) \qquad \text{(II.5)}$$

(*) * denotes complex-conjugate-transpose.

where $\{a(n)\}$ and $\{\omega(n)\}$ are input data and noise sequences, respectively.

The sequence $\{h(n)\}$ represents the impulse response of the transmitting filter/channel/receiving filter cascade. Throughout this work $\{a(n)\}$ and $\{\omega(n)\}$ are assumed to be stationary, ergodic processes.

Assumption 1: Input data $\{a(n)\}$ and noise $\{\omega(n)\}$ sequences are ergodic, at least wide-sense stationary processes.

II.2. OPTIMUM SOLUTION

Under the assumption of stationarity the MSE does not depend on the time index n, so (II.3) can be rewritten as

$$\xi^2 = E\{||e(n)||^2\} \tag{II.6}$$

Combining (II.2), (II.4) and (II.6) gives the MSE as

$$\xi^2 = E\{||a(n)||^2\} - 2E\{a(n)\underline{x}^*(n)\}\underline{c} + \underline{c}^*E\{\underline{x}(n)\underline{x}^*(n)\}\underline{c} \tag{II.7}$$

or

$$\xi^2 = \bar{a}^2 - 2\underline{b}^*\underline{c} + \underline{c}^*\underline{Ac} \tag{II.8}$$

where $\bar{a}^2$ stands for $E\{||a(n)||^2\}$ and, $\underline{b}^*$ and $\underline{A}$ are the following (2N+1)-vector and (2N+1)x(2N+1) matrix:

$$\underline{b}^* = E\{a(n)\underline{x}^*(n)\}$$
$$\underline{A} = E\{\underline{x}(n)\underline{x}^*(n)\}$$

In Appendix 1 these correlation matrices have been related to the impulse sequence $\{h(n)\}$.

The input correlation matrix $\underline{A}$ is Hermitian symmetric, at least positive semidefinite and Toeplitz. The last property is due to the stationarity of the input sequence $\{x(n)\}$. In Appendix 2 it is shown that $\underline{A}$ is positive definite unless the channel is identically zero.

The positive definiteness of the input correlation matrix $\underline{A}$ implies that

(i) $\underline{A}$ can be inverted
(ii) All eigenvalues of $\underline{A}$ are positive
(iii) The MSE performance function is a convex function of the real and imaginary components of the tap coefficients (See App.3).

The last property is important since it enables us to use the gradient methods in searching the minimum of the MSE performance function. In other words any minimum found by gradient methods will be the absolute minimum of the convex MSE performance function of the tap coefficients.

Therefore, the optimum solution for $\underline{c}$ can be found by taking the gradient of the MSE ξ^2 with respect to $\underline{c}$,

$$\nabla\xi^2 = -2\underline{b} + 2\underline{A}\;\underline{c} \qquad ^{(*)}\text{(II.9)}$$

and setting it to zero. Then, the optimum solution is

$$\underline{c}_{op} = \underline{A}^{-1}\underline{b} \qquad \text{(II.10)}$$

(*) Here, the MSE is a real valued function of the real and imaginary components of the elements of the tap vector $\underline{c}$ and gradient is taken with respect to real and imaginary parts of $\underline{c}$. However, the result can be expressed by combining these two gradient components in complex notatin as in Eqn (II.9). Otherwise ξ^2 is not an analytic function and complex differentiation is not possible.

This is the well-known Wiener-Hopf equation in matrix form. By putting (II.10) into (II.6), the minimum MSE is found as

$$\xi_{min} = \bar{a}^2 - \underline{b}\,\underline{c}_{op} \qquad \text{(II.11)}$$

In practice correlation matrices $\underline{b}$ and $\underline{A}$ are not known a priori and therefore stochastic gradient methods must be employed by adjusting the tap gains on the basis of measurements that can be made at the receiver.

II.3. STOCHASTIC GRADIENT ALGORITHM

The stochastic gradient algorithm, based on the steepest descent method updates the estimates for the tap vector in the direction negative to that of the gradient at that instant:

$$\underline{c}(n+1) = \underline{c}(n) - \alpha(n)\underline{\nabla}\xi^2(n) \qquad \text{(II.12)}$$

The positive scalar $\alpha(n)$ is called step-size, which will be shown later to play an important role in the behavior of the algorithm.

By definition the gradient is

$$\underline{\nabla}\xi^2(n) = \underline{\nabla}E\{e^*(n)e(n)\}$$

Interchanging the gradient and expoctation operator gives

$$\underline{\nabla}\xi^2(n) = E\{-2e^*(n)\underline{x}(n)\} \qquad \text{(II.13)}$$

The above expactation is generally not known apriori and also difficult to calculate. Insteed, an unbiased approximation to $\Delta\xi^2(n)$ may be

$$\hat{\nabla}\xi^2(n) = -2e^*(n)\underline{x}(n) \qquad \text{(II.14)}$$

Hence, the deterministic algorithm in (II.12) takes the following usable form:

$$\underline{c}(n+1) = \underline{c}(n) + 2\alpha(n)e^*(n)\underline{x}(n) \qquad \text{(II.15)}$$

III. CONVERGENCE OF THE STOCHASTIC GRADIENT ALGORITHM

In this section the following points will be examined:

1- Convergence of the tap coefficient vector to the optimum value

2- Convergence of the MSE to the minimum value

3- Quantization effects

II.1. CONVERGENCE OF THE TAP COCFFICIENT VECTOR

Let $\underline{p}(n)$ denote the error between the n th estimate $\underline{c}(n)$ for the tap vector and its optimum value:

$$\underline{p}(n) = e(n) - \underline{c}_{op} \qquad \text{(III.1)}$$

By combining (II.15) and (III.1)

$$\underline{p}(n+1) = \underline{p}(n) + 2\alpha(n)e^*(n)\underline{x}(n) \qquad \text{(III.2)}$$

is obtained. On the other hand the output error

$$e(n) = a(n) - \underline{c}^*(n)\underline{x}(n)$$

can be expressed in terms of the tap error vector:

$$e(n) = e_{op}(n) - \underline{p}^*(n)\underline{x}(n) \qquad \text{(III.3)}$$

Accordingly, (III.2) becomes

$$\underline{p}(n+1) = \underline{p}(n) + 2\alpha(n)e^*_{op}(n)\underline{x}(n) - 2\alpha(n)\underline{x}(n)\underline{x}^*(n)\underline{p}(n) \qquad \text{(III.4)}$$

Now $\underline{q}(n)$ is defined as the expected value of $\underline{p}(n)$:

$$\underline{q}(n) = E\{\underline{p}(n)\}$$

Then, the expoctation of both sides of (III.4) gives

$$\underline{q}(n+1) = \underline{q}(n) + 2\alpha(n)E\{e^*_{op}(n)\underline{x}(n)\} - 2\alpha(n)E\{\underline{x}(n)\underline{x}^*(n)\underline{p}(n)\} \qquad \text{(III.5)}$$

From linear filtering theory(25) it is known that the optimum error is orthogonal to the observation spoce, i.e.,

$$E\{e^*_{op}(n)\ \underline{x}(n)\} = \underline{0}\,(^*)$$

Thus the first term in (III.5) is zero.

The second term can be evaluated as

$$E\{\underline{x}(n)\underline{x}^*(n)\underline{p}(n)\} = E\{E\{\underline{x}(n)\underline{x}^*(n)\,|\,\underline{p}(n)\}\underline{p}(n)\}$$

where the outer expactation is over all values of $\underline{p}(n)$.

At this point an assumption that simplifies the problem is to be made:

Assumption 2: The input data vector at different instants are statistically independent of each other. In other words for any vector valued functions $\underline{f}$ and $\underline{g}$

$$E\{\underline{f}(\underline{x}(n))\underline{g}(\underline{x}(m))\} = E\{\underline{f}(\underline{x}(n))\}E\{\underline{g}(\underline{x}(m))\}$$

is valid for $n \neq m$. This assumption implies that the tap coefficient vector $\underline{c}(n)$ and the input vector $\underline{x}(n)$ are also statistically independent of each other. The last statement is based on (II.15). Note that this assumption is not strictly true since every input vector $\underline{x}(n)$ contains elements both from the previous and next ones. However it greatly simplifies the problem and will be used in the sequel.

By Assumption 2 $\underline{p}(n)$ and $\underline{x}(n)$ are statistically independent of each other Therefore,

$$E\{\underline{x}(n)\underline{x}\ (n)\underline{p}(n)\} = E\{\underline{x}(n)\underline{x}\ (n)\}E\{\underline{p}(n)\}$$

$$= \underline{Aq}(n)$$

Hence, (III.5) takes the form

$$\underline{q}(n\ 1) = \{\underline{I} - 2\alpha(n)\underline{A}\}\underline{q}(n) \qquad \text{(III.6)}$$

The system in (III.6) is stable if the eigenvalues of the matrix $\{\underline{I}-2\alpha(n)\underline{A}\}$ are less than unity in magnitude for each n. Since $\underline{A}$ is Hermition symmetric it can be expressed as

$$\underline{A} = \underline{M}^*\Lambda\underline{M} \qquad \text{(III.7)}$$

where $\underline{\Lambda}$ is the diagonal form of $\underline{A}$,

$$\underline{\Lambda} = \mathrm{diag}\{\lambda_1, \lambda_2, \ldots, \lambda_{2N+1}\}$$

and $\underline{M}$ is the unitary modal matrix of $\underline{A}$;

(*) Recall that $e^*_{op}(n) = a^*(n) - \underline{x}^*(n)\underline{c}_{op}$ Then

$$E\{\underline{x}(n)e^*_{op}\} = E\{a^*(n)\underline{x}(n) - \underline{x}(n)\underline{x}^*(n)\underline{c}_{op}\}$$
$$= \underline{b} - \underline{A}\ \underline{C}_{ap}$$
$$= \underline{b} - \underline{A}\ \underline{A}^{-1}\ \underline{b} = \underline{b} - \underline{b} = \underline{0}$$

$$\underline{M}^{-1} = \underline{M}^*$$

To make a coordinate transformation primed coordinates are defined as

$$\underline{q}'(n) = \underline{M}^*\underline{q}(n) \text{ and } \underline{q}(n) = \underline{M}\ \underline{q}'(n) \qquad \text{(III.8)}$$

Then the uncoupled form of (III.6) is

$$\underline{q}'(n+1) = \{\underline{I} - 2\alpha(n)\underline{\Lambda}\}\underline{q}'(n) \qquad \text{(III.9)}$$

and, $\underline{q}'(n)$ converges to zero as n goes to infinity provided that

$$|1-2\alpha(n)\lambda_i|<1 \quad \forall n, \quad i: 1,2,\dots,2N+1$$

or

$$0 < \alpha(n) < \frac{1}{\lambda_i} \quad \forall n, \quad i: 1,2,\dots,2N+1 \qquad \text{(III.10)}$$

Hence, the stability requirement is

$$0 < \alpha(n) < \frac{1}{\lambda_{max}} \quad \forall n \qquad \text{(III.11)}$$

where λ_{max} is the maximum eigenvalue of the input correlation matrix $\underline{A}$. If this requirement is satisfied, the solution to (III.9),

$$\underline{q}'(n) = \prod_{i=1}^{n} \{\underline{I} - 2\alpha(i)\lambda_i\}q'(0) \qquad \text{(III.12)}$$

converges to zero as n goes to infinity whatever the initial tap coefficient vector setting is. Thus, the tap vector estimates are asymptotically unbiased, i.e.,

$$\lim_{n\to\infty} E\{\underline{c}(n)\} = \underline{c}_{op}$$

As far as the stability of the algorithm is concerned the choice for the step-size $\alpha(n)$ is limited by the eigenvalues of the input correlation matrix as in (III.11). This means that the performance of the algorithm is affected by the eigenvalue spread of $\underline{A}$. To see this consider the mode $\underline{q}'_m(n)$ with the minimum eigenvalue λ_m. This modes converges with a geometric ratio.

$$r_m = 1 - 2\alpha(n)\lambda_m$$

while the mode $q'_M(n)$, with the maximum eigenvalue $_M$, the geometric convergence ratio is

$$r_m = 1 - 2\alpha(n)\lambda_M$$

When the eigenvalues are close to each other then both extreme modes converge to their optimum values at approximately the same rate. But if the eigenvalue ratio λ_M/λ_m is high, the choice for $\alpha(n)$, as imposed by (III.11) brings a slower rate for $\underline{q}'_m(n)$ than that for $q'_M(n)$, thus degrades the overall performance of the algorithm.

III.2. CONVERGENCE OF THE MSE

As the algorithm adapts toward the optimum values the error e(n) is no more stationary and consequently the MSE depends on time:

$$\xi^2(n) = E\{||e(n)||^2\} \qquad \text{(III.13)}$$

Combination of (III.3) with (III.13) gives

$$\xi^2(n) = \xi^2_{min} + E\{\underline{p}^*(n)\underline{A}\ \underline{p}(n)\} \qquad \text{(III.14)}$$

If the coordinate transformation as in (III.8) is applied to (III.14) it follows that

$$\xi^2(n) = \xi^2_{min} + E\{\underline{p}'^*(n)\underline{\Lambda}\ \underline{p}'(n)\} \qquad \text{(III.15)}$$

or

$$\xi^2(n) = \xi^2_{min} + \sum_{i=1}^{2N+1} \lambda_i E\{p_i'^*(n)p_i'(n)\}$$

From the transformed version of (III.4)

$$p_i'(n+1) = p_i'(n)+2\alpha(n)e^*_{op}(n)x_i'(n)-2\ (n)x_i'(n) \sum_{j=1}^{2N+1} x_j'\ (n)p_j'(n)$$

$$i: 1,\ 2, \ldots, 2N+1 \qquad \text{(III.16)}$$

is obtained. Since

$$\underline{x}'(n) = \underline{M}^*\underline{x}(n)$$

and $\underline{M}$ is the diagonalizing transformation it is clear that

$$E\{x_i'(n)x_j'\ (n)\} = \lambda_i\delta(i-j) \qquad \forall i,j \qquad \text{(III.17)}$$

Then the product $E\{p_i'^*(n)p_i'(n)\}$ evolves according to

$$E\{p_i'^*(n+1)p_i'(n+1)\} = T_{1,1}+T_{2,2}+T_{3,3}+2T_{1,2}-2T_{1,3}-2T_{2,3} \qquad \text{(III.18)}$$

for i: 1,2,...,2N+1, where

$$T_{1,1} = E\{p_i'^*(n)p_i'(n)\}$$

$$T_{2,2} = 4\alpha^2(n)\; E\{e^*_{op}(n)e_{op}(n)x_i'^*(n)x_i'(n)\}$$

$$T_{3,3} = 4\alpha^2(n)E\{x_i'(n)x_i'(n) \sum_{j=1}^{2N+1} \sum_{k=1}^{2N+1} x_j'^*(n)x_k'(n)p_j'(n)p_k'^*(n)\}$$

$$T_{1,2} = 2\alpha(n)E\{e^*_{op}(n)x_i'(n)p_i'^*(n)\}$$

$$T_{1,3} = 2\alpha(n)\; E\{x_i'(n)p_i'^*(n) \sum_{j=1}^{2N+1} x_j'^*(n)\; p_j'(n)\}$$

$$T_{2,3} = 4\alpha^2(n)E\{e^*_{op}(n)x_i'(n)x_i'^*(n) \sum_{j=1}^{2N+1} x_j'(n)p_j'^*(n)\}$$

Assuming $e_{op}(n)$ and $x_i(n)$ (i: 1,2,...,2N 1) are not only uncorrelated but also statistically independent $T_{2,2}$ becomes

$$T_{2,2} = 4\alpha^2(n)\; \xi^2_{min}\; \lambda_i$$

On the other hand the term $T_{1,2}$ vanishes if the orthogonality principle and Assumption 2 is applied. In $T_{3,3}$ the cross multiplications are zero in the mean, so it is reduced to a single summation and $T_{3,3}$ becomes

$$T_{3,3} = 4\alpha^2(n)\; \lambda_i \sum_{j=1}^{2N+1} \lambda_j E\{p_j'^*(n)p_j'(n)\}$$

Under the same assumptions $T_{2,3}$ also becomes zero. Finally for $T_{1,3}$

$$T_{1,3} = 2\alpha(n)\lambda_i E\{p_i'^*(n)p_i'(n)\}$$

is found. Then (III.18) can be reexpressed as

$$E\{p_i'^*(n+1)p_i'(n+1)\} = E\{p_i'^*(n)p_i'(n)\} + 4\alpha^2(n)\lambda_i \sum_{j=1}^{2N+1} \lambda_j E\{p_j'^*(n)p_j'(n)\}$$

$$-4\alpha(n)\lambda_i E\{p_i'^*(n)p_i'(n)\} + 4\alpha^2(n)\xi^2_{min}\lambda_i \qquad \text{(III.19)}$$

$i: 1,2,\ldots,2N+1$

Let $\theta_i(n)$ denote $E\{p_i'^*(n)p_i'(n)\}$. Then for $i: 1,2,\ldots,2N+1$

$$\theta_i(n+1) = \left|1-2\alpha(n)\lambda_i\right|^2 \theta_i(n) + 4\alpha^2(n)\lambda_i \sum_{\substack{j=1 \\ j\neq i}}^{2N+1} \lambda_j\theta_j(n)$$

$$+4\alpha^2(n)\xi^2_{min}\lambda_i \qquad \text{(III.20a)}$$

or

$$\theta_i(n+1) = \left|1-4\ (n)\lambda_i\right| \theta_i(n) + 4\alpha^2(n)\lambda_i \sum_{j=1}^{2N+1} \lambda_j\theta_j(n)$$

$$+4\alpha^2(n)\ \xi^2_{min}\lambda_i \qquad \text{(III.20b)}$$

The excess MSE in (III.15) can also be written as

$$\xi^2_{ex}(n) = \sum_{i=1}^{2N+1} \lambda_i\theta_i(n)$$

or

$$\xi^2_{ex}\ (n) = \underline{\lambda}^T\underline{\theta}(n) \qquad \text{(III.21)}$$

where $\underline{\theta}(n) = \begin{vmatrix} \theta_1(n) \\ \cdot \\ \cdot \\ \cdot \\ \theta_{2N+1}(n) \end{vmatrix}$

Then, from (III.20) $\underline{\theta}(n)$ is found to behave as

$$\underline{\theta}(n+1) = \underline{R}\ \underline{\theta}(n) + 4\alpha^2(n)\ \xi^2_{min}\ \underline{\lambda} \qquad \text{(III.22)}$$

where

$$\underline{R} = \begin{vmatrix} |1-2\alpha\ \lambda_1|^2 & 4\alpha^2\ \lambda_1\lambda_2 & \cdots & 4\alpha^2\ \lambda_1\lambda_{2N+1} \\ 4\alpha^2\ \lambda_1\lambda_2 & |1-2\alpha\lambda_2|^2 & \cdots & \cdot \\ \cdot & & & \cdot \\ \cdot & & & \cdot \\ |1-4\alpha^2\ \lambda_1\lambda_{2N+1} & \cdots & & |(1-2\alpha\ \lambda_{2N+1}|^2 \end{vmatrix}$$

assuming constant step-size.

The matrix $\underline{R}$ is real and symmetric, thus all eigenvalues of $\underline{R}$ are real. Therefore the transient behavior of the MSE is not oscillatory. It is known that a matrix whose elements are all positive and row sums are less than unity can have only eigenvalues with magnitude less than unity(29). For the matrix $\underline{R}$ the row sums are

$$1-4\alpha(n)\lambda_i + 4\alpha^2(n)\lambda_i \sum_{j=1}^{2N+1} \lambda_j$$

Then the system in εqn. (III.22) is stable if

$$1-4\ (n)\alpha_i+4\lambda^2(n)\lambda_i\ \text{Trace A} < 1$$

or

$$\alpha < \frac{1}{\text{Trace } \underline{A}} \qquad \text{(III.23)}$$

is satisfied. The bound in (III.23) is less than that in (III.11) which is required for the convergence of the tap coefficients Although this bound guarantees the convergence of both the tap coefficients and the MSE to the desired values it does not provide the optimum value for the fastest convergence. To obtain this optimum value for the step-size an upper bound for the excess MSE in (III.20) is to be considered. For λ_m and λ_M being the minimum and maximum eigenvalues of the matrix $\underline{A}$, respectively the excess MSE at the (n+1) th instant is bounded by

$$\xi_{ex}^2(n+1) \leq \{1 - 4\alpha(n)\lambda_m + 4\alpha^2(n)\,\lambda_M^2\,(2N+1)\}\,\xi_{ex}^2(n) + 4\alpha^2(n)\,\lambda_M^2\,(2N+1)\,\xi_{min}^2 \qquad \text{(III.24)}$$

By minimizing this bound the optimum choice for $\alpha(n)$ can be obtained. Taking the derivative of the right hand side of. (III.24) with respect to $\alpha(n)$ gives

$$-\lambda_m\xi_{ex}^2(n) + 2\lambda_M^2\,(2N+1)\,\xi_{ex}^2(n) + 2\lambda_M^2\,(2N+1)\,\xi_{min}^2\,(n)$$

Thus the optimum value for $\alpha(n)$ is obtained by setting this derivative to zero:

$$\alpha_{op}(n) = \frac{\lambda_m\xi_{ex}^2(n)}{2\lambda_M^2\,(2N+1)\,\left|\xi_{ex}^2(n) + \xi_{min}^2\right|} \qquad \text{(III.25)}$$

(III.25) shows that the optimum value of $\alpha(n)$ depends on (i) the eigenvalue ratio of the input correlation matrix, (ii) number of taps of the equalizer and (iii) it is a time-varying scalar. But this expression includes the excess MSE and therefore it is difficult to handle. In order to find a simple bound on $\alpha(n)$ let us express $\xi^2 ex(n)$ in terms of $\alpha(n)$:

$$\xi^2_{ex}(n) = \frac{2\xi^2_{min}\lambda^2_M\,(2N+1)\;\alpha_{op}(n)}{\lambda_m - 2\lambda_M{}^2(2N+1)\;\alpha_{op}(n)} \qquad \text{(III.26)}$$

Since $\xi^2_{ex}(n)$ is a positive quantity, the denominator of (III.26) must be also positive. Consequently

$$\alpha_{op}\;(n) \leq \frac{\lambda_m}{2\lambda M^2(2N+1)} \qquad \text{(III.27)}$$

is obtained. In fact as $\xi^2_{ex}(n)$ is greater than the minimum MSE ξ^2_{min} (III.25) takes the form

$$\alpha_{op}(n) = \frac{\lambda_m}{2\lambda M^2(2N+1)} \qquad \text{(III.28)}$$

Then as long as $\xi^2\;ex(n)$ is large as compared with $\xi^2 min$ the step-size $\alpha(n)$ can be held constant at its maximum value in (III.25). For the optimum step size in (III.25) the excess MSE $\xi^2_{ex}(n)$ evolves according to

$$\xi^2_{ex}(n+1) \leq \xi^2_{ex}(n) - \frac{\{\xi^2 ex(n)\lambda_m\}^2}{\left|\xi^2 ex(n) + \xi^2 min\right|\;\lambda^2_M\;(2N+1)} \qquad \text{(III.29)}$$

or,

$$\xi^2_{ex}(n+1) \leq \left|1 - 2\lambda_m\alpha(n)\right|\;\xi^2_{ex}(n) \qquad \text{(III.30)}$$

And, for the constant step-size in (3.27) the excess MSE decays at an exponential rate:

$$\xi_{ex}^2(n+1) \leq \left| 1 - \frac{\lambda_m}{\lambda_M^2 \ (2N+1)} \right| \xi_{ex}^2(n) \qquad \text{(III.31)}$$

On the other hand as $\xi_{ex}^2(n)$ approaches to ξ_{min}^2 constant $\alpha(n)$ will be no more satisfactory. By combining (III.25) and (III.30)

$$\frac{\alpha_{op}(n+1)}{\lambda_m - 2\lambda_M^2 \ (2N+1)\alpha_{op}(n+1)} \quad \frac{\left|1-2\ \alpha_{op}(n)\ \lambda_m\right|\alpha_{op}(n)}{\lambda_m - 2\lambda_M^2 \ (2N+1)\alpha_{op}(n)}$$

is found. After some manipulations it follows that

$$\alpha_{op}(n+1) \leq \alpha_{op}(n) \ \frac{1-2\lambda_m\alpha_{op}(n)}{\left|1-2\lambda_M(2N+1)\alpha_{op}(n)\right| \left|1+2\lambda_M(2N+1)\alpha_{op}(n)\right|} \qquad \text{(III.33)}$$

In order for $\xi_{ex}^2(n)$ to monotonically decrease $\alpha_{op}(n)$ must also decrease as imposed by (III.25). Then, for some $n \geq n_o$ $\alpha_{op}(n)$ so decreases that (III.33) takes the form

$$\alpha_{op}(n+1) \leq \alpha_{op}(n_o) \ \frac{1}{1 + 2\lambda_M \ (2N+1) \ \alpha_{op}(n_o)} \qquad \text{(III.34)}$$

From (III.34) it is found that

$$\alpha_{op}(n) \leq \frac{\alpha(n_o)}{1 + 2\lambda_M \ (2N+1) \ (n-n_o)\alpha(n_o)} \quad n \geq n_o \qquad \text{(III.35)}$$

Then the excess MSE $\xi_{ex}^2(n)$ also decreases in $\frac{1}{n}$ fashion.

In summary the algorithm attains the maximum rate of convergence if the step-size $\alpha(n)$ is set to the maximum value in (III.28) at the beginning and is decreased according to (III.35) after the excess MSE has been considerably decreased.

For constant α the 2-transform of (III.20) is given by

$$z\theta_i(2)-z\theta_i(0) = \left|1-2\alpha\lambda_i\right|^2 \theta_i(z) \quad 4\alpha^2\lambda_i \sum_{\substack{j=1\\ j\neq i}}^{2N\ 1} \lambda_j\theta_j(z)$$

$$+4\alpha^2 \xi^2_{min} \lambda_i \frac{z}{z-1} \quad i: 1,2,\ldots,2N+1$$

Then,

$$\theta_i(z) = \frac{z\theta_i(o)}{z-\left|1-2\alpha\lambda_i\right|^2} + 4\alpha^2\lambda_i \sum_{\substack{j=1\\ j\neq i}}^{2N+1} \lambda_j \frac{\theta_j(z)}{z-\left|1-2\alpha\lambda_i\right|^2}$$

$$+4\alpha^2 \xi^2_{min} \frac{z}{(z-1)\ \left[z-(1-2\alpha\lambda_i)^2\right]} \quad i: 1,2,\ldots 2N\ 1$$

Since $\alpha\neq 0$ and $\lambda=1$ is not an eigenvalue of the matrix R, z=1 is not a pole of any term in the equation above Hence,

$$\lim_{z\to 1} \{(z-1)\theta_i(z)\} = \frac{4\alpha^2\xi^2_{min}\ \lambda_i}{4\alpha\ \lambda_i(1-\alpha\lambda_i)}$$

$$= \xi^2_{min} \frac{\alpha}{1-\alpha\lambda_i}$$

is the steady-state value of $\theta_i(n)$. (i: 1,2,...,2N 1). Then, it follows that

$$\lim_{n\to\infty} \xi^2_{min} = \xi^2_{min} + \sum_{i=1}^{2N+1} \frac{\alpha\lambda_i}{1-\alpha\lambda_i} \xi^2_{min} \qquad \text{(III.36)}$$

Let λ_{ave} be $\frac{1}{2N+1}$ Trace $\underline{A}$, then (III.36) becomes

$$\lim_{n\to\infty} \xi^2(n) \approx \xi^2_{min} + \frac{(2N+1)\alpha\lambda_{ave}}{1-\alpha\lambda_{ave}} \xi^2_{mn} \qquad \text{(III.37)}$$

Although (III.37) has been derived for constant α it gives on approximation for the time-varying step-size. At least it indicates that the steady-state excess MSE depends on the number of taps and α. An icrease in either of them causes on increase in the excess MSE at steady-state.

III.3. EFFECTS OF DIGITAL IMPLEMENTATION

Up to that point all the quantities involved are assumed to be anolog. However, adjustable parameters are to be quantized when the equalizer is digitaly implemented.

Consider the algorithm in (II.15) as composed to its real and imaginary components:

$$\underline{c}_R(n+1) = \underline{c}_R(n) + 2\alpha(n) \left| e^*(n) \; \underline{x}(n) \right|_R$$

$$\underline{c}_I(n+1) = \underline{c}_I(n) + 2\alpha(n) \left| e^*(n) \; \underline{x}(n) \right|_I$$

where the subscripts $\underline{R}$ and $\underline{I}$ denote the real and imaginary parts, respectively. Since the variables $\underline{c}_R(n+1)$ and $\underline{c}_I(n+1)$ are to be quantized, whenever the up doing terms are less than the least significant digit (LSD) the adaptation tends to stop. In other words whenever

$$2\alpha(n) \left| e^*(n) \; \underline{x}(n) \right|_R \leq \text{LSD}$$

and (III.38)

$$2\alpha(n) \left| e^*(n) \; \underline{x}(n) \right|_I \leq \text{LSD}$$

happens, these terms are truncated consequently, the adaptation stops at an earlier stage. If the equations in (III.38) are combined

$$4\alpha^2(n)\,||e^*(n)\ \underline{x}(n)||^2 \leq 2(LSD)^2 \qquad \text{(III.39)}$$

is found. Then the number of bits and the step-sized must be chosen in such a way that the above in equality does not come into place before the algorithm approaches to the minimum MSE. It is clear from (III.39) that the minimum number of bits required for a satisfactory adaptation performance will be less when the optimum MSE is high.

Now, expected value of (III.39) is approximately

$$\xi^2(n) \approx \frac{(LSD)^2}{2\alpha^2\ E\{||\underline{x}(n)||^2} \qquad \text{(III.40)}$$

As the adaptation goes on the left hand side of (III.40) approaches to

$$\xi^2_{min}\left|1 + \frac{(2N+1)\ \alpha\lambda_{ave}}{1 - \alpha\lambda_{ave}}\right|$$

Thus, the algorithm will not be affected by quantization if the relation

$$\xi^2_{min}\left|1 + \frac{(2N+1)\ {}_{ave}}{1 - \alpha\lambda_{ave}}\right| = \frac{(LSD)^2}{2\alpha^2\ E\ \{||\underline{x}(n)||^2}$$

holds.

IV. KALMAN/GODARD ALGORITHM

In Chap. III it has been seen that the convergence rate of the stochastic gradient algorithm is highly affected by the eigenvalue spread of the input correlation matrix $\underline{A}$. One method to eliminate the effect of the eigenvalue spread may be to use different step-size for each mode. Thus, instead of the algorithm

$$\underline{c}(n+1) = \underline{c}(n) + \alpha(n)\underline{x}(n)e^*(n)$$

that one

$$\underline{c}(n+1) = \underline{c}(n) + \underline{\pi}(n)\underline{x}(n)e^*(n) \qquad \text{(IV.1)}$$

is suggested. In this case, expected value of the tap coefficient error vector evolves according to

$$\underline{q}(n+1) = \left[\underline{I} - \underline{\pi}(n)\underline{A}\right]\underline{q}(n)$$

under Assumption 2. If the matrix $\underline{\pi}(n)$ could be $\underline{A}^{-1}$, inverse of the channel correlation matrix, then the algorithm in (IV.1) would converge in a few steps. Although $\underline{A}^{-1}$ is not known a priori, it can be recursively estimated. Then these estimates can be used in updating the tap coefficient vector. This method is also equivalent to construct the algorithm as

$$\underline{A}(n)\underline{c}(n) = \underline{b}(n) \qquad \text{(IV.2)}$$

where $\underline{A}(n)$ and $\underline{b}(n)$ are the n th estimates of the correlation matrices $\underline{A}$ and $\underline{b}$.

IV.1. DERIVATION OF THE KALMAN/GODARD ALGORITHM(*)

Given the output of the TDL equalizer as

$$y(n) = \underline{c}^*(n-1)\underline{x}(n) \qquad \text{(IV.3)}$$

the problem is the same as stated in Chap. II. Let $\underline{A}(n)$ denote the n th estimate of the input correlation matrix $\underline{A}$. Since the environment is taken stationary/ergodic

$$\underline{A}(n) = \frac{1}{n}\sum_{i=1}^{n} \underline{x}(i)\underline{x}^*(i) \qquad \text{(IV.4)}$$

is an asymptotically unbiased estimate for $\underline{A}$. From (IV.4) it is seen that

$$n\underline{A}(n) = (n-1)\underline{A}(n-1) + \underline{x}(n)\underline{x}^*(n) \qquad \text{(IV.5)}$$

Similarly, the n the estimate of the correlation vector $\underline{b}$ is taken as

$$\underline{b}(n) = \frac{1}{n}\sum_{i=1}^{n} a^*(i)\underline{x}(i) \qquad \text{(IV.6)}$$

and evolves according to

$$n\underline{b}(n) = (n-1)\underline{b}(n-1) + a^*(n)\underline{x}(n) \qquad \text{(IV.7)}$$

Combining (IV.2) and (IV.7) gives

(*) Another derivation of the algorithm based on Kalman filtering is given in Appendix 4.

$$n\underline{A}(n)\underline{c}(n) = (n-1)\underline{b}(n-1) + a^*(n)\underline{x}(n) \tag{IV.8}$$

Since the error e(n) is

$$e(n) = a(n) - \underline{c}^*(n-1)\underline{x}(n) \tag{IV.9}$$

(IV.8) becomes

$$n\underline{A}(n)\underline{c}(n) = (n-1)\underline{b}(n-1)+e^*(n)\underline{x}(n)+\underline{x}(n)\underline{x}^*(n)\underline{c}(n-1) \tag{IV.10}$$

By putting $\underline{A}(n-1)\underline{c}(n-1)$ instead of $\underline{b}(n-1)$ is (IV.10)

$$n\underline{A}(n)\underline{c}(n)=(n-1)\underline{A}(n-1)\underline{c}(n-1)+e^*(n)\underline{x}(n)+\underline{x}(n)\underline{x}^*(n)\underline{c}(n-1) \tag{IV.11}$$

is obtained. On the other hand, from (IV.5), it is seen that:

$$(n-1)\underline{A}(n-1) = \underline{n}\ \underline{A}(n) - \underline{x}(n)\underline{x}^*(n) \tag{IV.12}$$

Thus,

$$n\underline{A}(n)\underline{c}(n) = \underline{n}\ \underline{A}(n)\underline{c}(n-1)-\underline{x}(n)\underline{x}^*(n)\underline{c}(n-1)+e^*(n)\underline{x}(n) + $$

$$+ \underline{x}(n)\underline{x}^*(n)\underline{c}(n-1)$$

or

$$\underline{c}(n) = \underline{c}(n-1)+\left[n\underline{A}(n)\right]^{-1}\underline{x}(n)e^*(n) \tag{IV.13}$$

is found. $\underline{D}(n)$ is defined as

$$\underline{D}(n) = \left[n\underline{A}(n)\right]^{-1} \tag{IV.14}$$

Hence, (IV.13) takes the form

$$\underline{c}(n) = \underline{c}(n-1) + \underline{D}(n)\underline{x}(n)e^*(n) \tag{IV.15}$$

(IV.5) can be reexpressed as

$$\underline{D}(n)^{-1} = \underline{D}(n-1)^{-1} + \underline{x}(n)\underline{x}^*(n)$$

and using matrix inversion lemma $\underline{D}(n)$ can be recursively computed as

$$\underline{D}(n) = \underline{D}(n-1) - \frac{\underline{D}(n-1)\underline{x}(n)\underline{x}^*(n)\underline{D}(n-1)}{1+\underline{x}^*(n)\underline{D}(n-1)\underline{x}(n)} \qquad \text{(IV.16)}$$

Now, the term $\underline{D}(n)\underline{x}(n)$, which is called Kalman gain, is

$$\underline{D}(n)\underline{x}(n) = \underline{D}(n-1)\underline{x}(n)\left[1 - \frac{\underline{x}^*(n)\underline{D}(n-1)\underline{x}(n)}{1+\underline{x}^*(n)\underline{D}(n-1)\underline{x}(n)}\right]$$

$$= \underline{D}(n-1)\underline{x}(n)/\left[1+\underline{x}^*(n)\underline{D}(n-1)\underline{x}(n)\right] \qquad \text{(IV.17)}$$

Consequently, the algorithm can be operated as

$$\underline{g}(n) = \underline{D}(n-1)\ \underline{x}(n)/\left[1+\underline{x}^*(n)\underline{D}(n-1)\underline{x}(n)\right] \qquad \text{(IV.18)}$$

$$\underline{D}(n) = \underline{D}(n-1) - \frac{D(n-1)\underline{x}(n)\underline{x}^*(n)\underline{D}(n-1)}{1+x^*(n)\underline{D}(n-1)\underline{x}(n)} \qquad \text{(IV.16)}$$

$$y(n) = \underline{c}^*(n-1)\underline{x}(n) \qquad \text{(IV.3)}$$

$$e(n) = a(n) - y(n) \qquad \text{(IV.9)}$$

$$\underline{c}(n) = \underline{c}(n-1)+\underline{g}(n)e^*(n) \qquad \text{(IV.19)}$$

However, the algorithm has the danger that $\underline{A}(n)$ can be singular at the beginning of the iteration, but this danger can be eliminated by slightly modifying the expression for $\underline{A}(n)$ as

$$\underline{A}(n) = \frac{1}{n}\sum_{i=1}^{n} \underline{x}(i)\underline{x}(i)^* + \delta\underline{I} \qquad \text{(IV.20)}$$

where δ is a small positive constant. In this case $\underline{D}(o)$ is initialized as

$$\underline{D}(o) = \frac{1}{\delta}\,\underline{I} \qquad \text{(IV.21)}$$

and nothing changes in the algorithm.

IV.2. CONVERGENCE OF THE TAP COEFFICIENT VECTOR

According to Eqn. (IV.15) the tap coefficient error vector behaves as

$$\underline{p}(n) = \underline{p}(n-1) + \underline{D}(n)\underline{x}(n)e^*(n) \qquad \text{(IV.22)}$$

It is convenient to write Eqn. (IV.22) as

$$\underline{p}(n) = \underline{p}(n-1) + \underline{D}(n)\underline{x}(n)\left[e^*_{op}(n) - x^*(n)\underline{p}(n-1)\right] \qquad \text{(IV.23)}$$

Since $e_{op}(n)$ and $\underline{x}(n)$ are orthogonal and input sample vector at different instants are independent (by Assumption 2).

$$E\{\underline{D}(n)\underline{x}(n)e^*_{op}(n)\} = \underline{0}$$

By the same reasoning

$$E\{\underline{D}(n)\underline{x}(n)\underline{x}^*(n)\underline{p}(n-1)\} = \left[\sum_{i=1}^{n} E\{\underline{x}(i)\underline{x}^*(i)\}\right]^{-1} E\{\underline{x}(n)\underline{x}^*(n)\}$$

$$.E\{\underline{p}(n-1)\}$$

$$= \left[n\underline{A}\right]^{-1} \underline{A}\ \underline{q}(n-1)$$

$$= \frac{1}{n}\,\underline{q}\ (n-1)$$

Then, it follows that

$$\underline{q}(n) = \left[1 - \frac{1}{n} \right] \underline{q}(n-1) \qquad \text{(IV.24)}$$

As n goes to infinity $\underline{q}(n)$ reaches a steady-state, i.e.,

$$\lim_{n\to\infty} \underline{q}(n) = \lim_{n\to\infty} (1 - \frac{1}{n})\underline{q}(n-1)$$

$$= \lim_{n\to\infty} \underline{q}(n-1)$$

The solution for $\underline{q}(n)$ is then

$$\underline{q}(n) = \frac{(n-1)!}{n!} \underline{q}(2)$$

$$= \frac{1}{n} \underline{q}(2)$$

Thus,

$$\lim_{n\to\infty} \underline{q}(n) = \lim_{n\to\infty} \frac{1}{n} \underline{q}(2)$$

$$= \underline{0}$$

that is, the estimates for the tap coefficients are asymptotically unbiased for all finite initial settings. But the convergence of the MSE has not been examined yet. This is the subject of the following section.

IV.3. CONVERGENCE OF THE MSE

The MSE, as seen in Chap. III consists of the minimum value and the excess MSE:

$$\xi^2(n) = \xi^2_{min} + E\{\underline{p}^*(n)\underline{A}\ \underline{p}(n)\}$$

The Hermitian form in the expoctation is

$$\underline{p}^*(n)\underline{A}\ \underline{p}(n) = \underline{p}^*(n-1)\underline{A}\ \underline{p}(n-1)+e_{op}(n)e^*_{op}(n)\underline{x}^*(n)\underline{D}(n)\underline{A}\ \underline{D}(n)\underline{x}(n)$$

$$+\ p^*(n-1)\underline{x}(n)\underline{x}^*(n)\underline{D}(n)\underline{A}\ \underline{D}(n)\underline{x}(n)\underline{x}^*(n)\underline{p}(n-1)$$

$$+\ 2\ e^*_{op}(n)\underline{x}^*(n)\underline{D}(n)\underline{A}\ \underline{p}(n-1)$$

$$-\ 2\ p^*(n-1)\underline{x}(n)\underline{x}^*(n)\underline{D}(n)A\ \underline{p}(n-1)$$

$$-\ 2\ e_{op}(n)\underline{x}^*(n)\underline{D}(n)\underline{A}\ \underline{D}(n)\underline{x}(n)\underline{x}^*(n)\underline{p}(n-1)$$

The expactation of the above expression transformed into primed coordinates is

$$E\{\underline{p}'^*(n)\underline{\Lambda}\ \underline{p}'(n)\} = T_{1,1}+T_{2,2}+T_{3,3}+2T_{1,2}-T_{1,3}-2T_{2,3}$$

where

$$T_{1,1}=E\{\underline{p}'^*(n-1)\underline{\Lambda}\ \underline{p}'(n-1)\}$$

$$T_{2,2}=E\{e_{op}(n)e^*_{op}(n)\underline{x}'^*(n)\underline{D}'(n)\underline{\Lambda}\ \underline{D}'(n)\underline{x}'(n)\}$$

$$T_{3,3}=E\{\underline{p}'^*(n-1)\underline{x}'(n)\underline{x}'^*(n)\underline{D}'(n)\underline{\Lambda}\ \underline{D}'(n)\underline{x}'(n)\underline{x}'^*(n)\underline{p}'(n-1)\}$$

$$T_{1,2}=E\{e^*_{op}(n)\underline{x}'^*(n)\underline{D}'(n)\underline{\Lambda}\ \underline{p}'(n-1)\}$$

$$T_{1,3}=E\{\underline{p}'^*(n-1)\underline{x}'(n)\underline{x}'^*(n)\underline{D}'(n)\underline{\Lambda}\ \underline{p}'(n-1)\}$$

$$T_{2,3}=E\{e_{op}(n)\underline{x}'^*(n)\underline{D}'(n)\underline{\Lambda}\ \underline{D}'(n)\underline{x}'(n)\underline{x}'^*(n)\underline{p}'(n-1)\}$$

The first term is the excess MSE at the (n-1) st instant:

$$\xi^2_{ex}(n-1) = T_{1,1} = \sum_{i=1}^{2N+1} \lambda_i E\{p_i'^*(n-1)p_i'(n-1)\}$$

For the second term, applying Assumption 2

$$T_{2,2} = E\{e^*_{op}(n)e_{op}(n)\} \quad E\{\underline{x}'^*(n)\ \frac{1}{n}\ \underline{\Lambda}^{-1}\underline{\Lambda}\ \underline{\Lambda}^{-1}\ \frac{1}{n}\ \underline{x}'(n)\}$$

$$= \frac{1}{n^2}\ \xi^2_{min} \sum_{i=1}^{2N+1} \frac{1}{\lambda_i}\ E\{x_i'^*(n)x_i'(n)\}$$

$$= \frac{1}{n^2}\ \xi^2_{min}\ (2N+1)$$

is obtained. With the same reasoning

$$T_{3,3} = E\{\underline{p}'^*(n-1)\underline{x}'(n)\underline{x}'^*(n)\underline{\Lambda}^{-1}\underline{\Lambda}\ \underline{\Lambda}^{-1}\ \underline{x}'(n)\underline{x}'^*(n)\underline{p}'(n-1)\}$$

$$= \frac{1}{n^2}\ E\{\underline{p}'^*(n-1)\underline{x}'(n)\underline{x}'^*(n)\underline{\Lambda}^{-1}\underline{x}'(n)\underline{x}'^*(n)\underline{p}'(n-1)\}$$

$$= \frac{1}{n^2}E\{ \sum_{i=1}^{2N+1} \frac{1}{\lambda_i} x_i'^*(n)x_i'(n) \sum_{j=1}^{2N+1} x_j'^*(n)p_j'(n-1) \sum_{k=1}^{2N+1} x_k'(n)$$

$$\cdot p_k'^*(n-1)\}$$

is found. Unless $j = k$, the terms $x_j'^*(n)x_k'(n)$ in expactation vanish. Thus

$$T_{3,3} = \frac{1}{n^2} \sum_{i=1}^{2N+1} \sum_{j=1}^{2N+1} \frac{1}{i}\ E\{x_i'(n)x_i'^*(n)\}E\{x_j'^*(n)x_j'(n)\}$$

$$\cdot\ E\{p_j'^*(n-1)p_j'(n-1)\}$$

$$= \frac{(2N+1)}{n^2} \sum_{i=1}^{2N+1} \lambda_i\ E\ p_i'^*(n-1)p_i'(n-1)\}$$

By means of orthogonality principle $T_{1,2}$ and $T_{2,3}$ are zero.

Finally,

$$T_{1,3} = \frac{1}{n} \sum_{i=1}^{2N+1} \lambda_i E\{p_i'^*(n-1)p_i'(n-1)\}$$

Therefore the excess MSE adapts as

$$\xi^2_{ex}(n) = \{1 - \frac{2}{n} + \frac{(2N+1)}{n^2}\}\xi^2_{ex}(n-1) + \frac{(2N+1)}{n^2} \xi^2_{min} \quad \text{(IV.25)}$$

Take the limits of both sides of this equation as n goes to infinity

$$\lim_{n\to\infty}\xi^2_{ex}(n) = \lim_{n\to\infty}\{1 - \frac{2}{n} + \frac{(2N+1)}{n^2}\} \xi^2_{ex}(n-1)+\lim_{n\to\infty} \frac{(2N+1)}{n^2} \xi^2_{min}$$

$$= \lim_{n\to\infty} \xi^2_{ex}(n-1)$$

In other words the system in (IV.25) reaches to a steady state value, which is

$$\frac{1}{n} \left[2 - \frac{2N+1}{n}\right]\xi^2_{ex}(n) = \frac{1}{n^2}(2N+1) \xi^2_{min}$$

$$\xi^2_{ex}(n) = \frac{(2N+1)}{2n - (2N+1)} \xi^2_{min} \quad \text{(IV.26)}$$

Then, the algorithm would converge to the minimum MSE in approximately 2 x (2N+1) steps.

V. FAST KALMAN EQUALIZER ALGORITHM

V.1. COMPUTATIONAL COMPLEXITY OF THE STOCHASTIC GRADIENT AND KALMAN/GODAD ALGORITHMS

The stochastic gradient algorithm requires 2M complex multiplications where M is the number of taps, i.e., M=2N+1. Although having very fast rate of convergence Kalman/Godard algorithm is computationally complex. Utilizing the Hermitian property of the matrix $\underline{D}(n)$ one needs $3/2\ M^2+3M$ complex multiplications in this case. Obviously this is a considerable increase in computational complexity. However, in the formulation of the Kalman/Godard algorithm the property that every input vector $\underline{x}(n)$ contains only one element different than those in the previous one was not used. It is that property which enables us to develop a new computation method for the Kadman/Godard algorithm, as will be done in the following section.

V.2. DERIVATION OF THE FAST KALMAN ALGORITHM

The Kalman gain in (IV.8) is

$$\underline{g}(n) = \underline{D}(n)x(n) \qquad \text{(V.1)}$$

Let $\underline{W}(n)$ denote $n\underline{A}(n)$, i.e.,

$$\underline{W}(n) = \sum_{i=1}^{n} \underline{x}(i)\underline{x}^*(i) \tag{V.2}$$

Then, $\underline{D}(n) = \underline{W}^{-1}(n)$ and (V.1) can be rewritten as

$$\underline{W}(n)g(n) = \underline{x}(n) \tag{V.3}$$

Let $\mu(n)$ be $x(n+N+1)$ and $\mu(n-M)$ be $x(n-N)$,

$$\underline{x}(n) = \begin{bmatrix} \mu(n-1) \\ \vdots \\ \mu(n-M) \end{bmatrix} : \quad \underline{x}(n+1) = \begin{bmatrix} \mu(n) \\ \vdots \\ \mu(n-M+1) \end{bmatrix}$$

The extended input vector $\underline{x}_{ex}(n)$ is defined as

$$\underline{x}_{ex}(n) = \begin{bmatrix} \mu(n) \\ \text{--} \\ \underline{x}(n) \end{bmatrix} = \begin{bmatrix} \underline{x}(n+1) \\ \text{--} \\ \mu(n-M) \end{bmatrix} \tag{V.4}$$

Similarly, the extended correlation matrix $\underline{W}_{ex}(n)$ is

$$\underline{W}_{ex}(n) = \sum_{i=1}^{n} \underline{x}_{ex}(i)\, \underline{x}^*_{ex}(n) \tag{V.5}$$

$\underline{W}_{ex}(n)$ can also be expressed in the following ways:

$$\underline{W}_{ex}(n) = \left[\begin{array}{c|c} \sum_{i=1}^{n} \mu(i)\mu(i)^* & \sum_{i=1}^{n} \mu(i)\underline{x}^*(i) \\ \hline \sum_{i=1}^{n} \mu(i)^*\underline{x}(i) & \sum_{i=1}^{n} \underline{x}(i)\underline{x}^*(i) \end{array}\right] = \left[\begin{array}{c|c} \eta(n) & \underline{z}^*(n) \\ \hline \underline{z}(n) & \underline{W}(n) \end{array}\right] \tag{V.6}$$

or

$$\underline{W}_{ex}(n) = \left[\begin{array}{c|c} \sum_{i=1}^{n} \underline{x}(i+1)\underline{x}^*(i+1) & \sum_{i=1}^{n} \mu^*(i-M)\underline{x}(i+1) \\ \hline \sum_{i=1}^{n} \mu(i-M)\underline{x}^*(i+1) & \sum_{i=1}^{n} \mu^*(i-M)\mu(i-M) \end{array}\right] = \left[\begin{array}{c|c} \underline{W}(n+1) & \underline{t}(n) \\ \hline \underline{t}^*(n) & \beta(n) \end{array}\right] \quad (V.7)$$

Now, assume that the gain vector $\underline{g}(n)$ is known. Then the problem is to find $\underline{g}(n+1)$ satisfying

$$\underline{W}(n+1)\underline{g}(n+1) = \underline{x}(n+1)$$

As a first step assume that some vector $\underline{f}(n)$ and Scalar $\Sigma(n)$ are known such that

$$\underline{W}_{ex}(n) \left[\begin{array}{c} 1 \\ \hline \underline{f}(n) \end{array}\right] = \left[\begin{array}{c} \Sigma(n) \\ \hline \underline{0} \end{array}\right] \quad (V.8)$$

and form the extended gain vector as

$$\underline{g}_{ex}(n) = \left[\begin{array}{c} -\Sigma^{-1}(n) \,|\underline{z}^*(n)\underline{g}(n) - \mu(n)| \\ \hline \underline{g}(n) - \underline{f}(n)\Sigma^{-1}(n)\,|z^*(n)\underline{g}(n) - \mu(n)| \end{array}\right] \quad (V.9)$$

Then it follows that

$$\underline{W}_{ex}(n)\underline{g}_{ex}(n) = \underline{W}_{ex}(n) \left[\begin{array}{c} 0 \\ \hline \underline{g}(n) \end{array}\right] - \underline{W}_{ex}(n) \left[\begin{array}{c} 1 \\ \hline \underline{f}(n) \end{array}\right] \Sigma^{-1}(n)\,|\underline{z}^*(n)\underline{g}(n) - \mu(n)|$$

$$= \underline{W}_{ex}(n) \begin{bmatrix} 0 \\ \hline \underline{g}(n) \end{bmatrix} - \begin{bmatrix} \Sigma(n) \\ \hline \underline{0} \end{bmatrix} \Sigma^{-1}(n) \; |\underline{z}^*(n)\underline{g}(n)-\mu(n)|$$

$$= \underline{W}_{ex}(n) \begin{bmatrix} 0 \\ \hline \underline{g}(n) \end{bmatrix} - \begin{bmatrix} \underline{z}^*(n)\underline{g}(n) - \mu(n) \\ \hline \underline{0} \end{bmatrix} \qquad (V.10)$$

On the other hand from (V.6)

$$\underline{W}_{ex}(n) \begin{bmatrix} 0 \\ \hline \underline{g}(n) \end{bmatrix} = \begin{bmatrix} \eta(n) & \underline{z}^*(n) \\ \hline \underline{z}(n) & \underline{W}(n) \end{bmatrix} \begin{bmatrix} 0 \\ \hline \underline{g}(n) \end{bmatrix}$$

$$= \begin{bmatrix} \underline{z}^*(n)\underline{g}(n) \\ \hline \underline{x}(n) \end{bmatrix} \qquad (V.11)$$

Thus, combining (V.10) and (V.11)

$$\underline{W}_{ex}(n)\underline{g}_{ex}(n) = \underline{x}_{ex}(n) \qquad (V.12)$$

is obtained.

Secondly, partion $\underline{g}_{ex}(n)$ such that

$$\underline{g}_{ex}(n) = \begin{bmatrix} \underline{h}(n) \\ \hline s(n) \end{bmatrix} \qquad (V.13),$$

and assume that there exist $\underline{k}(n)$ and $Y(n)$ such that

$$\underline{W}_{ex}(n) = \begin{bmatrix} \underline{k}(n) \\ \hline Y(n) \end{bmatrix} = \begin{bmatrix} \underline{0} \\ \hline 1 \end{bmatrix} \qquad \text{(V.14)}$$

Note that if (V.7) is used

$$\underline{W}_{ex}(n)\underline{g}_{ex}(n) = \left[\begin{array}{c|c} \underline{W}(n+1) & \underline{t}(n) \\ \hline t^*(n) & \beta(n) \end{array}\right] \begin{bmatrix} \underline{h}(n) \\ \hline s(n) \end{bmatrix}$$

$$= \begin{bmatrix} \underline{W}(n+1)\underline{h}(n) + \underline{t}(n)\ s(n) \\ \hline \underline{t}^*(n)\underline{h}(n) + \beta(n)s(n) \end{bmatrix} \qquad \text{(V.15)}$$

From Eqn.(V.12) is found a different expression for $\underline{W}_{ex}(n)$ $\underline{g}_{ex}(n)$ is

$$\underline{W}_{ex}(n)\underline{g}_{ex}(n) = \begin{bmatrix} \underline{x}(n+1) \\ \hline \mu(n-M) \end{bmatrix} \qquad \text{(V.16)}$$

Thus, equating (V.15) and (V.16)

$$\underline{W}(n+1)\underline{h}(n) + \underline{t}(n)s(n) = \underline{x}(n+1) \qquad \text{(V.17)}$$

is obtained. From (V.14) it is seen that

$$\underline{W}(n+1)\underline{k}(n) + \underline{t}(n)Y(n) = 0 \qquad \text{(V.18)}$$

Thus $\underline{t}(n) = -Y^{-1}(n)\underline{W}(n+1)\underline{k}(n)$, and putting this value into (V.17) gives

$$\underline{W}(n+1)\underline{h}(n) - \underline{W}(n+1)\underline{k}(n)Y^{-1}(n)s(n) = \underline{x}(n+1)$$

or,

$$\underline{W}(n+1)\ |\underline{h}(n) - \underline{k}(n)Y^{-1}(n)s(n)| = \underline{x}(n+1) \qquad (V.19)$$

But then by definition

$$\underline{g}(n+1) = \underline{h}(n) - \underline{k}(n)Y^{-1}(n)s(n) \qquad (V.20)$$

Then it remains to update the variables $\underline{f}(n)$, $\Sigma(n)$, $\underline{k}(n)$ and $Y(n)$.

From (V.6) and (V.8) it is seen that

$$\underline{z}(n) + \underline{W}(n)\underline{f}(n) = \underline{0}$$

Since

$$\underline{z}(n) = \underline{z}(n-1) + \mu^*(n)\underline{x}(n)$$

it follows that

$$\underline{z}(n-1) + \mu^*(n)\underline{x}(n) + \underline{W}(n)\underline{f}(n) = \underline{0}$$

On the other hand

$$\underline{z}(n-1) = -\underline{W}(n-1)\underline{f}(n-1)$$

and thus

$$-\underline{W}(n-1)\underline{f}(n-1) + \mu^*(n)\underline{x}(n) + \underline{W}(n)\underline{f}(n) = \underline{0}$$

Using

$$\underline{W}(n-1) = \underline{W}(n) - \underline{x}(n)\underline{x}^*(n)$$

and $\underline{W}(n)\underline{g}(n) = \underline{x}(n)$

$$\underline{W}(n)\ |-\underline{f}(n-1) + \underline{g}(n)\underline{x}^*(n)\underline{f}(n-1) + \underline{g}(n)\mu^*(n) + \underline{f}(n)| = \underline{0}$$

is obtained. $\underline{W}(n)$ is nonsingular, then $\underline{f}(n)$ can be updated as

$$\underline{f}(n) = \underline{f}(n-1) - \underline{g}(n)\ |\mu^*(n) + \underline{x}^*(n)\underline{f}(n-1)| \qquad (V.21)$$

For $\Sigma(n)$, note that

$$\eta(n) + \underline{z}^*(n)\underline{f}(n) = \Sigma(n)$$

from (V.6) and (V.8). Since

$$\eta(n) = \eta(n-1) + \mu(n)\mu^*(n)$$

and

$$\eta(n-1) = \Sigma(n-1) - \underline{z}^*(n-1)\ \underline{f}\ (n-1)$$

$\Sigma(n)$ becomes,

$$\Sigma(n) = \Sigma(n-1)-\underline{z}^*(n-1)\underline{f}(n-1)+z^*(n)f(n)+\mu(n)\mu^*(n)$$

or

$$\Sigma(n) = \Sigma(n-1)-\underline{z}^*(n)\underline{f}(n-1)+\mu(n)\underline{x}^*(n)\underline{f}(n-1)+\underline{z}^*(n)\underline{f}(n)+\mu(n)\mu^*(n)$$

Using (V.21) gives

$$\Sigma(n) = \Sigma(n-1)+\mu(n)\mu^*(n)+\mu(n)\underline{x}^*(n)\underline{f}(n-1)-\underline{z}^*(n)\underline{g}(n)|\mu^*(n)+\underline{x}^*(n)\underline{f}(n-1)|$$

Thus $\Sigma(n)$ is updated as

$$\Sigma(n) = \Sigma(n-1) + |\mu(n) - \underline{z}^*(n)\underline{g}(n)|\ |\mu^*(n) + \underline{x}^*(n)\underline{f}(n-1)| \qquad (V.22)$$

The remaining variables to be updated are $\underline{k}(n)$ and $Y(n)$ in (V.14). It is clear that

$$\underline{W}_{ex}(n) \begin{bmatrix} \underline{k}(n) \\ ---- \\ Y(n) \end{bmatrix} = \underline{W}_{ex}(n-1) \begin{bmatrix} \underline{k}(n-1) \\ ------ \\ Y(n-1) \end{bmatrix}$$

Then it follows that

$$\underline{W}_{ex}(n) \begin{bmatrix} k(n) \\ ---- \\ Y(n) \end{bmatrix} = \underline{W}_{ex}(n) \begin{bmatrix} \underline{k}(n-1) \\ ------ \\ Y(n-1) \end{bmatrix} - \underline{x}_{ex}(n)\underline{x}^*_{ex}(n) \begin{bmatrix} \underline{k}(n-1) \\ ------ \\ Y(n-1) \end{bmatrix}$$

Consequently using (V.12)

$$\begin{bmatrix} \underline{k}(n) \\ ---- \\ Y(n) \end{bmatrix} = |\underline{I} - \underline{g}_{ex}(n)\ \underline{x}^*_{ex}(n)| \begin{bmatrix} \underline{k}(n-1) \\ ------ \\ Y(n-1) \end{bmatrix} \qquad (V.23)$$

is obtained.

It may be convenient to modify

$$\begin{bmatrix} \underline{k}(n) \\ ---- \\ Y(n) \end{bmatrix}$$

as

$$\begin{bmatrix} \underline{m}(n) \\ ---- \\ \cdot \end{bmatrix}$$

Where $\underline{m}(n) = Y^{-1}(n)\underline{k}(n)$. Then (V.14) becomes

$$\underline{W}_{ex}(n) = \begin{bmatrix} \underline{m}(n) \\ ---- \\ 1 \end{bmatrix} = \begin{bmatrix} \underline{0} \\ ------ \\ Y^{-1}(n) \end{bmatrix} \tag{V.24}$$

From (V.23),

$$Y(n) = Y(n-1)-s(n)\ \{\underline{x}^*(n+1)\underline{k}(n-1) + \mu^*(n-M)Y(n-1)\} \tag{V.25}$$

It is also possible to write

$$Y(n) = Y(n-1)\ |1-s(n)\underline{x}^*(n+1)\underline{m}(n-1)-s(n)\mu^*(n-M)|$$

Then, $Y^{-1}(n)$ is updated using

$$Y^{-1}(n) = Y^{-1}(n-1)\ |1-s(n)\underline{x}^*(n+1)\underline{m}(n-1)-s(n)\mu^*(n-M)|^{-1} \tag{V.26}$$

For $\underline{k}(n)$,

$$\underline{k}(n) = \underline{k}(n-1)-\underline{h}(n)\underline{x}^*(n+1)\underline{k}(n-1)-\underline{h}^*(n)\mu^*(n-M)Y(n-1) \tag{V.27}$$

is obtain from (V.23). Multiplying both sides by $Y^{-1}(n)$ gives

$$\underline{m}(n) = |\underline{k}(n-1)-\underline{h}(n)\underline{x}^*(n+1)\underline{k}(n-1)-\underline{h}^*(n)\mu^*(n-M)Y(n-1)|Y^{-1}(n) \tag{V.28}$$

Combining (V.26) and (V.28),

$$\underline{m}(n) = |\underline{k}(n-1)-\underline{h}(n)\underline{x}^*(n+1)\underline{k}(n-1)-\underline{h}^*(n)\mu^*(n-M)Y(n-1)|$$

$$\cdot\ Y^{-1}(n-1)\ |1-s(n)\underline{x}^*(n+1)\underline{m}(n-1)-s(n)\mu^*(n-M)|^{-1}$$

or

$$\underline{m}(n) = \left|\underline{m}(n-1) - \underline{h}(n)\underline{x}^*(n+1)\underline{m}(n-1) - \underline{h}^*(n)\mu^*(n-M)\right|$$

$$\cdot \left|1-s(n)\underline{x}^*(n+1)\underline{m}(n-1)-s(n)\mu^*(n-M)\right|^{-1} \qquad (V.29)$$

is obtained.

Now, at the n th instant $\underline{x}(n)$, $\underline{x}(n+1)$, $\underline{f}(n-1)$, $\Sigma(n-1)$ and $\underline{g}(n)$ are known. The first step is to determine $g_{ex}(n)$ and this requires $\underline{f}(n)$ and $\Sigma(n-1)$. Let $\varepsilon_1(n)$ be defined by

$$\varepsilon_1(n) = \mu(n) + \underline{f}^*(n-1)\underline{x}(n) \qquad (V.30)$$

Then by (V.21)

$$\underline{f}(n) = \underline{f}(n-1)-\underline{g}(n)\ \varepsilon_1{}^*(n) \qquad (V.31)$$

Also let $\varepsilon_2(n)$ defined by

$$\varepsilon_2(n) = \mu(n) + \underline{f}^*(n)\underline{x}(n) \qquad (V.32)$$

It is seen that

$$\varepsilon_2(n) = \mu(n) + \underline{f}^*(n)\underline{W}(n)\underline{g}(n)$$

And

$$\varepsilon_2(n) = \mu(n) - \underline{z}^*(n)\underline{g}(n)$$

Than, by (V.22)

$$\Sigma(n) = \Sigma(n-1) + \varepsilon_2(n)\varepsilon_1{}^*(n) \qquad (V.33)$$

Accordingly the extended gain vector becomes

$$\underline{g}_{ex}(n) = \begin{bmatrix} \Sigma^{-1}(n)\varepsilon_2(n) \\ \hline \underline{g}(n) + \underline{f}(n)\ \Sigma^{-1}(n)\varepsilon_2(n) \end{bmatrix} \qquad (V.34)$$

Partion $\underline{g}_{ex}(n)$ as

$$\underline{g}_{ex}(n) = \begin{bmatrix} \underline{h}(n) \\ \hline s(n) \end{bmatrix} \qquad (V.35)$$

To update $\underline{g}(n)$ $\underline{m}(n)$ is required. Since $\underline{m}(n-1)$ and $Y^{-1}(n-1)$ are known from the previous iteration let

$$\underline{d}(n) = \mu(n-M) + \underline{m}^*(n-1)\underline{x}(n+1) \qquad (V.36)$$

Then $\underline{m}(n-1)-\underline{h}(n)|\underline{x}^*(n+1)\underline{m}(n-1)+\mu^*(n-M)| = \underline{m}(n-1)-\underline{h}(n)d^*(n)$

and $1-s(n)|x^*(n+1)\underline{m}(n-1)-\mu^*(n-M)| = 1-s(n)d^*(n)$

Thus, from (V.29) it follows that

$$\underline{m}(n) = |\underline{m}(n-1) - \underline{h}(n)d^*(n)|\,|1-s(n)d^*(n)|^{-1} \qquad (V.37)$$

Finally using (V.20) gives

$$\underline{g}(n+1) = \underline{h}(n) - \underline{m}(n)s(n) \qquad (V.38)$$

and, of course,

$$y(n+1) = \underline{c}^*(n)\underline{x}(n) \qquad (V.39)$$
$$e(n+1) = a(n)-y(n) \qquad (V.40)$$
$$\underline{c}(n+1) = \underline{c}(n) + \underline{g}(n+1)e^*(n+1) \qquad (V.41)$$

The performance of the fast Kalman algorithm is excetly the same as Kalman/Godard algorithm since they differs only in computation. However, if equations from (V.30) to (V.38) are examined, it is seen that only 10M+3 complex multiplications are necessary. Therefore, the computational complexity of Kalman/Godard algorithm is considerably decreased. For example for a 15-taps TDL equalizer required number of complex multiplications are in the following orders:

Stochastic gradient: 30
Kalman/Godard : 385
Fast Kalman : 150

VI. SIMULATION AND RESULTS

VI.1. COMPUTER PROGRAMS

The simulation package consists of the following computer programs:

1- Polyf.Ftn: Simulates telephone channels with different attenuation and group delay characteristics(*).
2- Del.Ftn: Simulates a QAM system with specified characteristics(*).
3- Opt.Ftn: Calculates the optimum tap coefficients and the minimum MSE.
4- Help.Ftn: Calculates the eigenvalues and the trace of the input correlation matrix.
5- Dstoc.Ftn: Realizes the stochastic gradient algorithm.
6- Dkalman.Ftn: Realizes the Kalman/Godard algorithm.
7- Dfast.Ftn: Realizes the fast Kalman algorithm.

In Appendix 5 detailed explanation to use these programs is given.

(*) These two programs are supplied from the MODEM Project carried out in Electronic Research department of Marmara Research Institute.

VI.2. SIMULATED COMMUNICATION SYSTEM

A qam transmission system with a signalling rate of 1600 baud and 2 bits per symbol is simulated. The carrier frequency is placed at 1800 Hz. The transmitter consists of a raised-cosine filter with an excess bandwidth of 25 percent. The binary bits for both quadrature and in-phase components are provided from a pseudo random binary sequence. Additive noise of variance 0.001 is supplied from a gaussian noise generator. The equivalent baseband impulse response of the combined transmitter and channel is used to generate the input data for the equalizer. It is assumed that the carrier phase used for demodulation at the receiver is that of a pilot carrier transmitted through the channel.

The simulation results presented are averages over ten different training sequences and gaussian noise sequences. In the Kalman and Fast Kalman algorithms the initial error covariance matrix is set to 0.01 times the identity matrix. Since both of them give the same performance only one of the outputs is illustrated throughout the work. The initial value for all tap coefficients in all of the equalization algorithms are chosen as zero. The simulated TDL equalizer has 15 complex taps.

VI.3. RESULTS

A. Channel A

For this nearly perfect channel the three algorithms are tested with both noisy and noiseless data.

When there is no noise, the optimum tap coefficients and the minimum MSE are:

REAL TAP COEFFICIENTS,	IMAGINARY TAP COEFFICIENTS.
-0.67644266149150 32D-04	0.1101299723561884D-03
0.1238748514371620D-03	-0.4293756861847678D-03
-0.2286396954809717D-03	0.9692642596640687D-03
0.3983265033319855D-03	-0.1883136429397953D-02
-0.8354119154621916D-03	0.3463356227646182D-02
0.2668949433495415D-02	-0.7453312858927695D-02
-0.1475407491806941D-01	-0.4547521087120434D-02
0.9989985475894537D+00	0.1905480836957972D-01
0.1494450710794966D-01	-0.3721070832814822D-02
-0.2170329247547206D-02	-0.7791840404899667D-02
0.5693159458682711D-03	0.3357577515153634D-02
-0.3430506547778394D-03	-0.1808895768409438D-02
0.2396066665030081D-03	0.9228012214495684D-03
-0.1646275143091066D-03	-0.3990609982682001D-03
0.1131275109071906D-03	0.9054933849355621D-04

minimum achieveable MSE = 0.67×10^{-9} or -91.72 dB

Since the distortion is low minimum and maximum eigenvalues are identical:

TRACE=15.0150371772758799

MIN.EIG.=0.1000671523563891D+01
MAX.EIG.=0.1001082570684720D+01
RATIO=0.1000410771278236D+01

The best choice for the step-size is found to be 0.033.

From Fig. VI.A.3. it is seen that the stochastic gradient algorithm reaches to -60 dB MSE in 200 iterations, while the Fast Kalman (and Kalman) algorithm, as indicated in Fig. (VI.A.6) converges to the same value in 25 steps.

The minimum MSE for this channel is so small that the stochastic gradient algorithm is expected to be very sensitive to truncation errors. When 8-bits are used to store the tap coefficients adaptation stops at -45 dB MSE (Fig. VI.A.4). Unfortunately Kalman and Fast Kalman algorithms are also sensitive to quantization erros (Fig. VI.A.7). However with 12 bit all the algorithms gives results identical to the infinite precision case (Fig. VI.A.8, Fig. VI.A.5.).

When there exists White Gaussian Noise of variance 0.001 the optimum solutions are as follows:

REAL TAP COEFFICIENTS,	IMAGINARY TAP COEFFICIENTS.
-0.676084703465516 2D-04	0.110021087797158 9D-03
0.123782885152033 1D-03	-0.428952558550745 8D-03
-0.228432811519465 3D-03	0.968307574306776 6D-03
0.397933392601284 5D-03	-0.188127180527012 5D-02
-0.834562185671017 5D-03	0.345991582291323 6D-02
0.266625034394457 0D-02	-0.744588834692917 2D-02
-0.147393015578010 0D-01	-0.454297497159968 9D-02
0.998001545480443 1D+00	0.190357921079406 6D-01
0.149296423211614 6D-01	-0.371736433446885 2D-02
-0.216819805034466 5D-02	-0.778405166006210 2D-02
0.568762686502846 1D-03	0.335421104762531 6D-02
-0.342702794400707 4D-03	-0.180707544282246 3D-02
0.239345405615376 5D-03	0.921868748859571 7D-03
-0.164431281301285 2D-03	-0.398656119878116 8D-03
0.112983277690796 4D-03	0.904573336149504 9D-04

MINIMUM ACHIEVEABLE MSE=
0.998002209003548 4D-03   -0.300086849743308 0D 02DB

TRACE=15.0300371772758807

MIN.EIG.=0.100079978686295 6D+01
MAX.EIG.=0.100308058285648 1D+01
RATIO=0.100227897330061 8D+81

Noise addition changes only the minimum achieveable MSE. The step-size can be still set to 0.033. Then the stochastic gradient algorithm converges in 150 steps, but with a small bias as imposed by Fig. VI.A.9. However, the Fast Kalman algorithm attains a MSE of -27.5 dB in 25 iterations and converges to the exact minimum MSE in 60 iterations (Fig. VI. A.10). Since the minimum MSE is about 0.001 truncation errors do not occurs and 8 bit quantization does not changes the performance in either case (Fig. VI.A.11, Fig. VI.A.12).

Fig. (VI.A1)- Frequency Response

Fig. (VI.A2)- Impulse Response

Fig. (VI.A3)- Stochastic Gradient Algorithm Infinite Precision $\alpha=0.033$

Fig. (VI.A4)- Stochastic Gradient Algorithm No.of tap coefficient bits = 8 $\alpha=0.033$

Fig. (VI.A5)- Stochastic Gradient Algorithm No.of tap coefficient bits =12 $\alpha=0.033$

Fig. (VI.A6)- Fast Kalman Algorithm Infinite Precision

Fig. (VI.A7)- Fast Kalman Algorithm No.of top coefficient bits = 8

Fig. (VI.A8)- Fast Kalman Algorithm No.of top coefficient bits =12

Fig. (VI.A9)- Stochastic Gradient Algorithm Infinite Precision $\alpha = 0.033$, $\delta^2 = 0.001$

Fig.(VI.A10)- Fast Kalman Algorithm Infinite Precision, $\delta^2 = 0.001$

Fig.(VI.A11)- Stochastic Gradient Algorithm No.of tap Coefficient bits = 8, $\delta^2 = 0.001$ $\alpha = 0.033$

Fig.(VI.A12)- Fast Kalman Algorithm No.of tap coefficient bits = 8, $\delta^2 = 0.001$

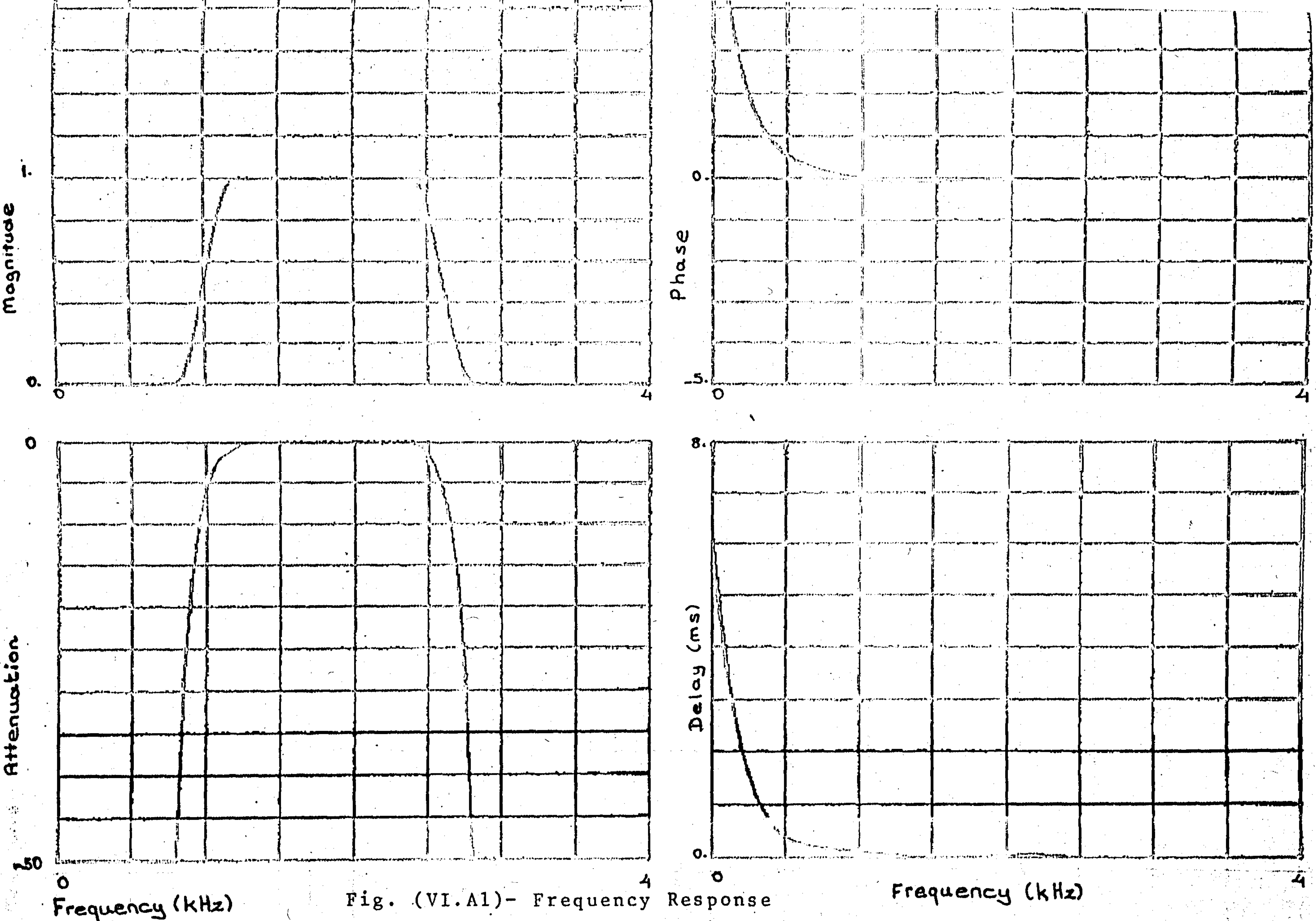

Fig. (VI.A1)- Frequency Response

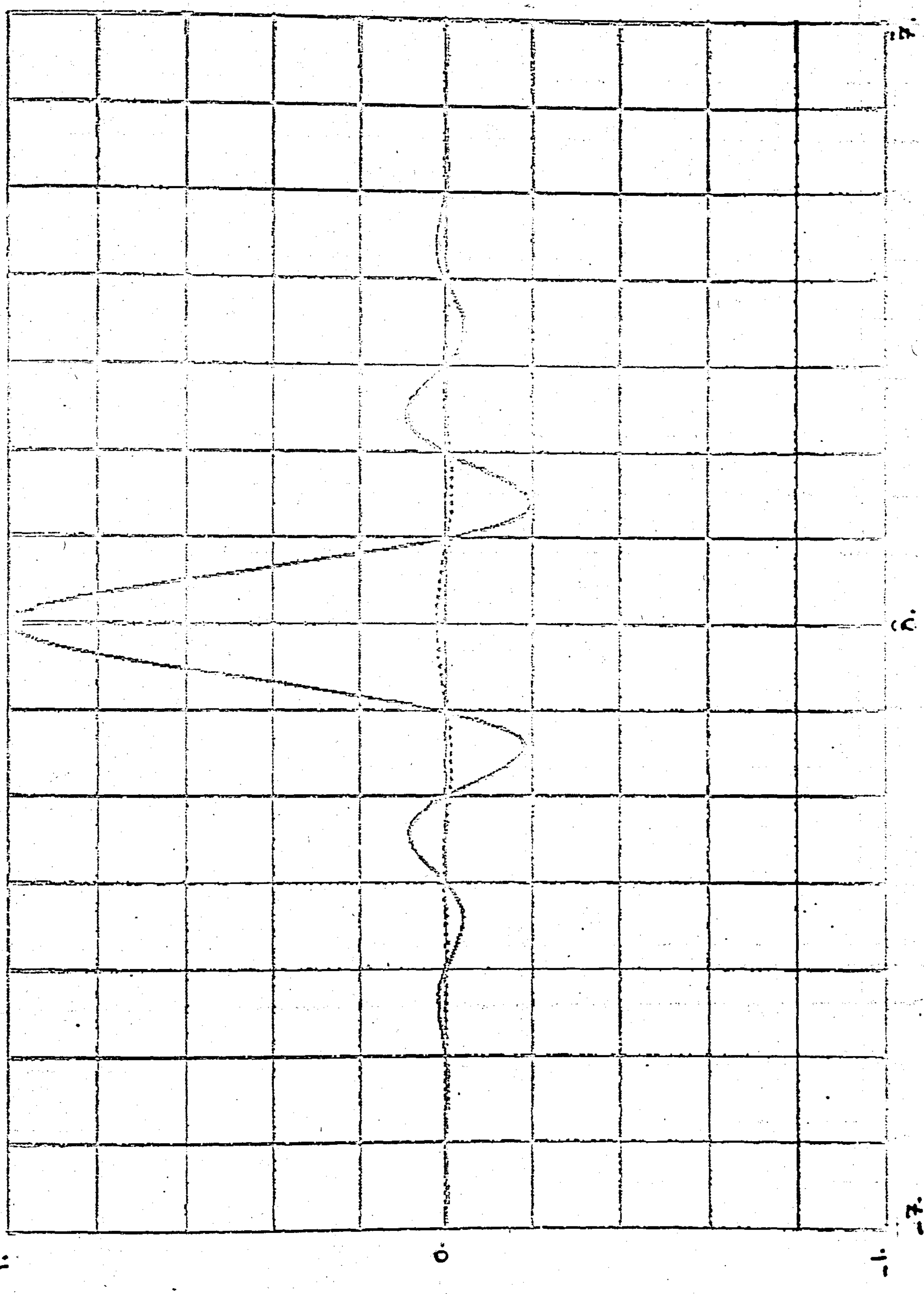

Fig. (VI.A2)- Impulse Response

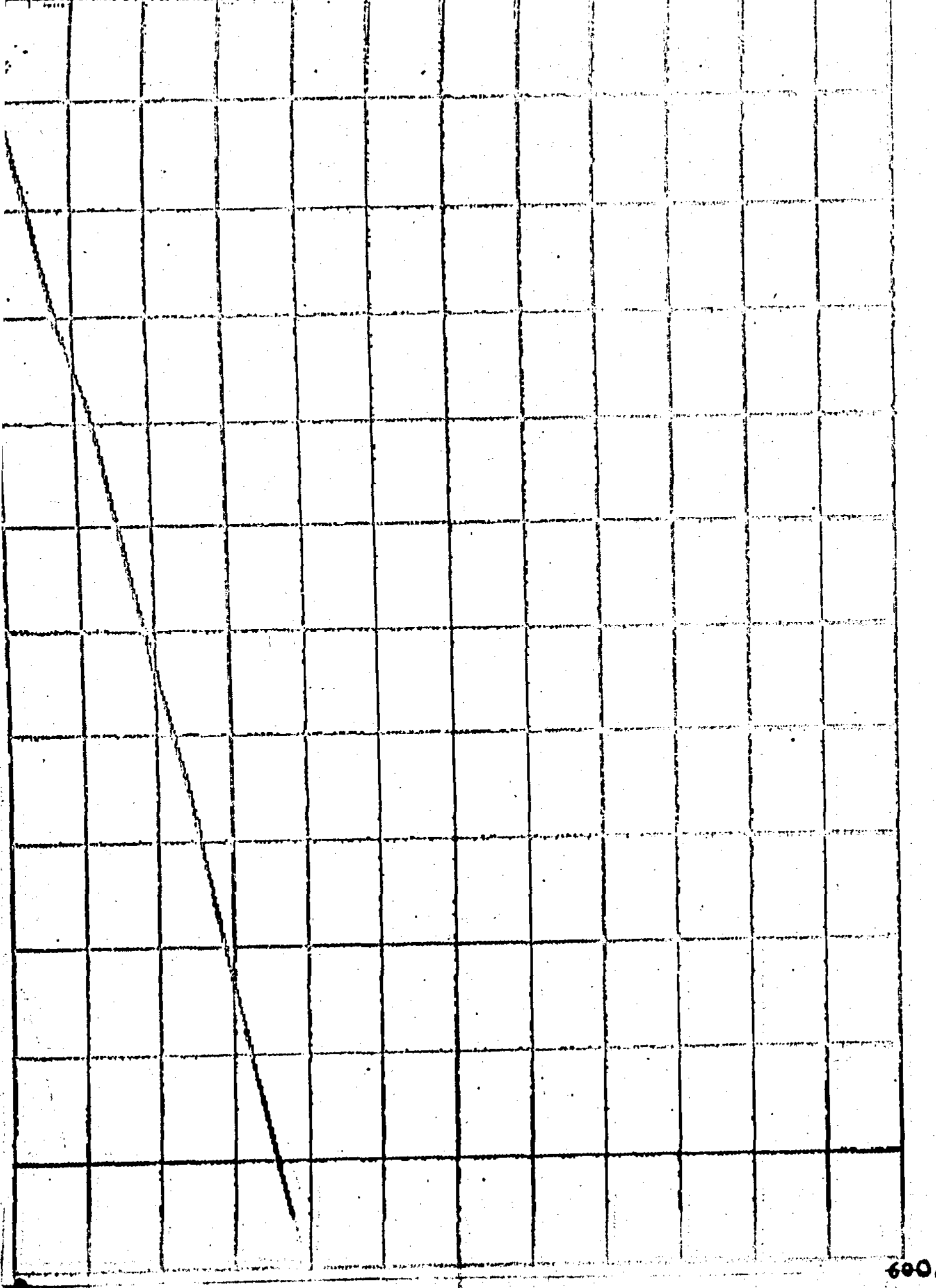

Fig. (VI.A3) Stochastic Gradient Algorithm
Infinite Precision $\alpha=0.033$

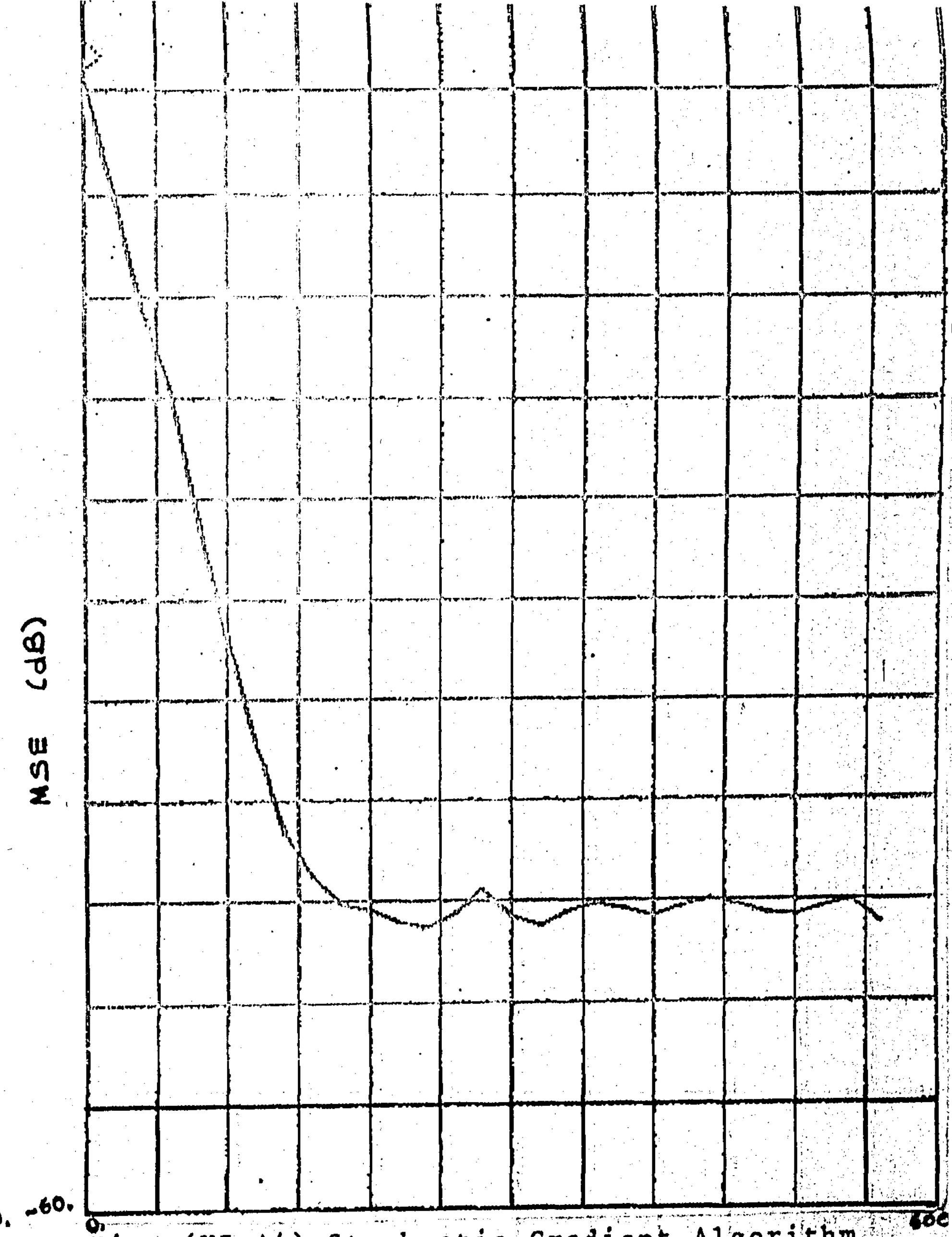

Fig. (VI.A4) Stochastic Gradient Algorithm
No. of tap coefficient bits=8
$\alpha=0.033$

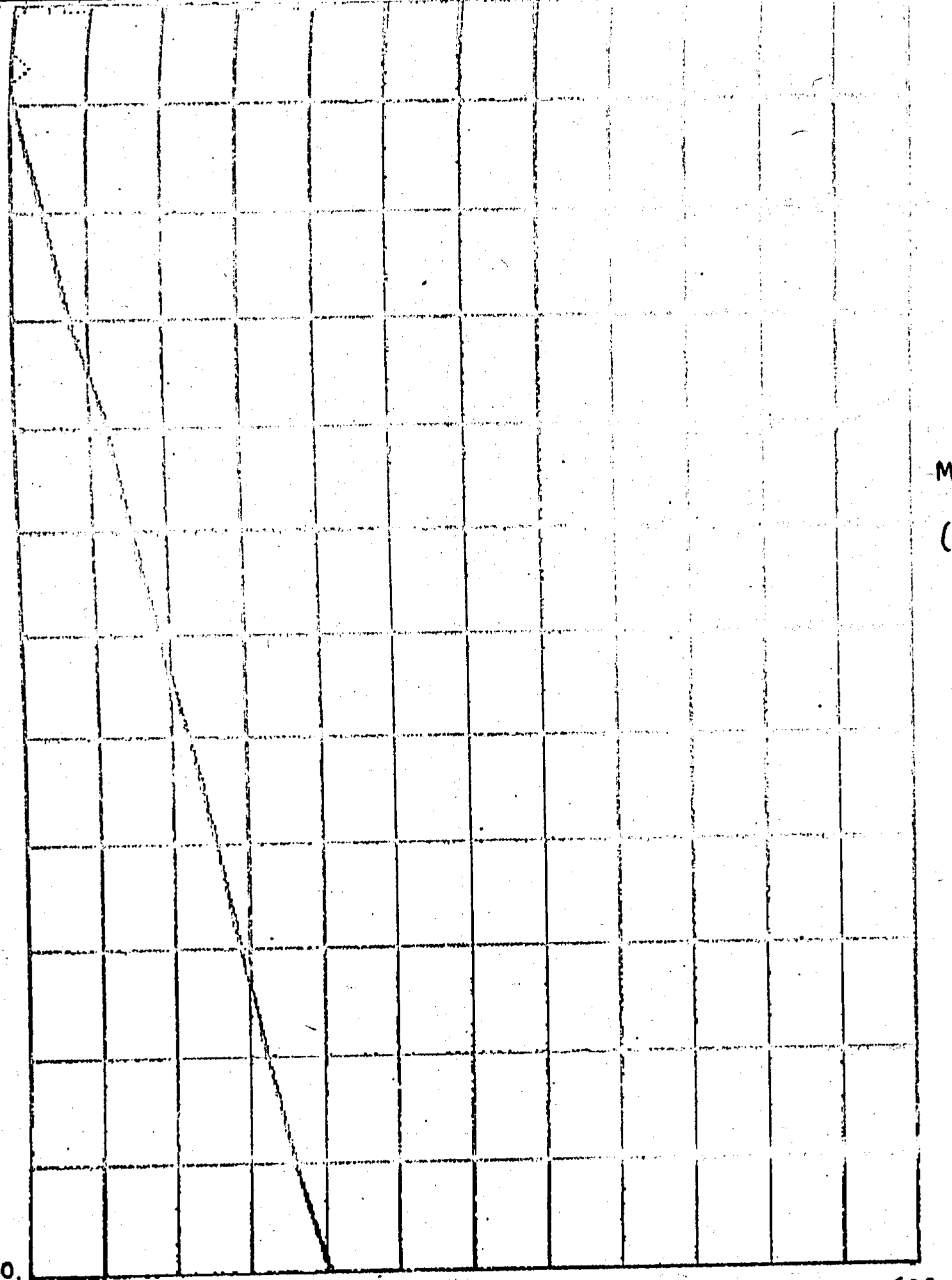

Fig. (VI.A5)-

MSE

(dB)

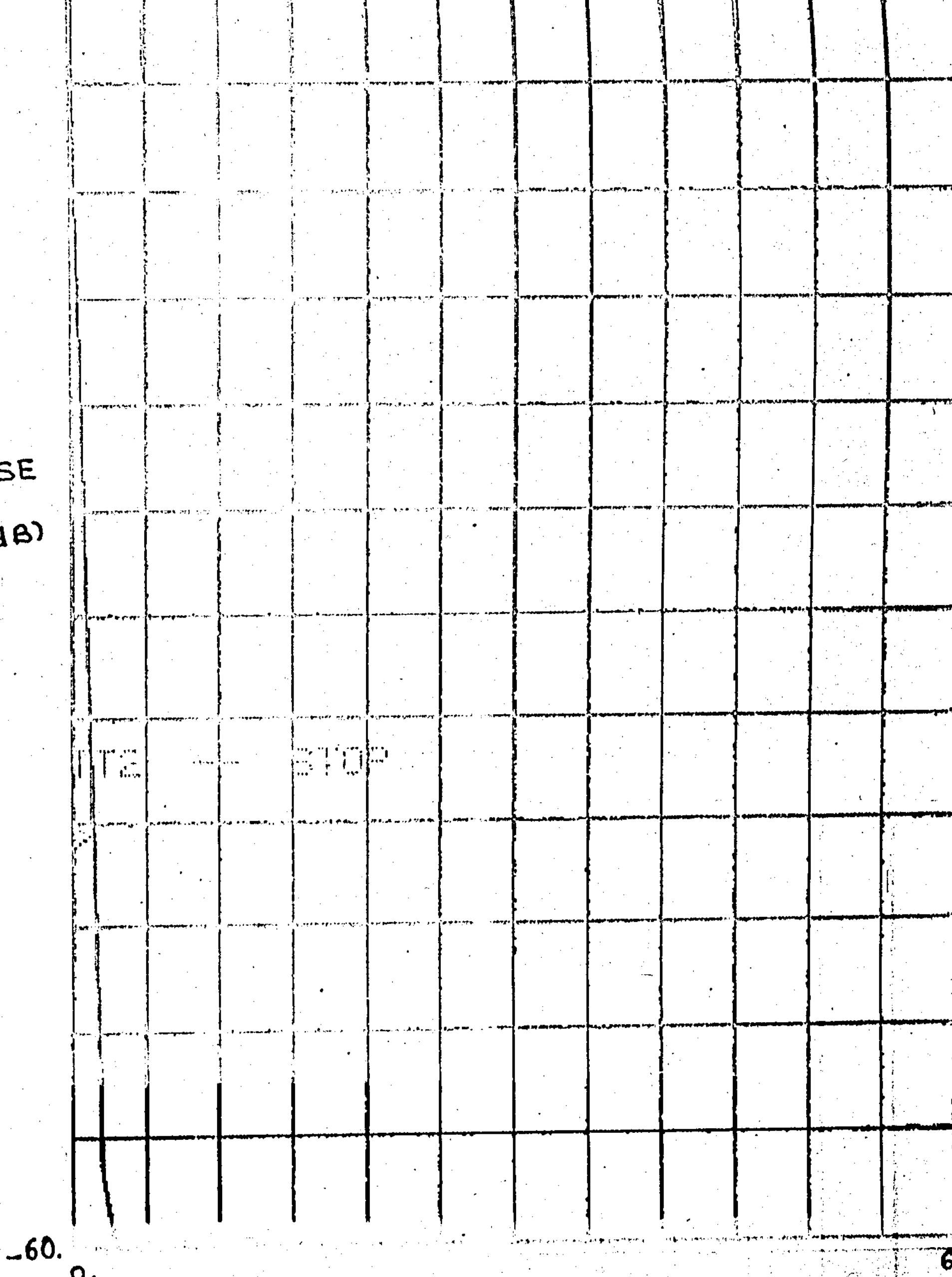

Fig. (VI.A6)-

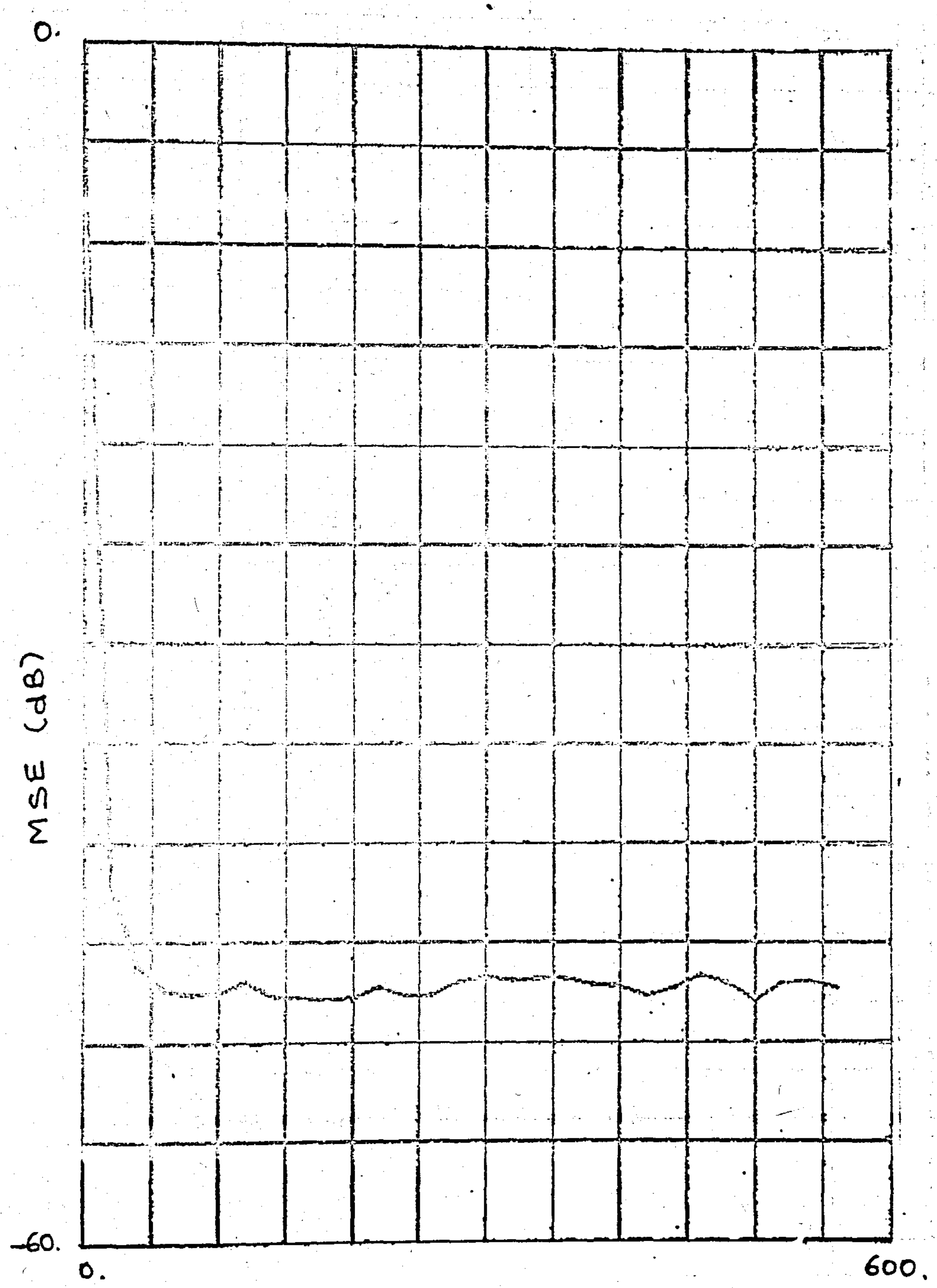

Fig.(VI.A7)

Fig.(VI.A8)

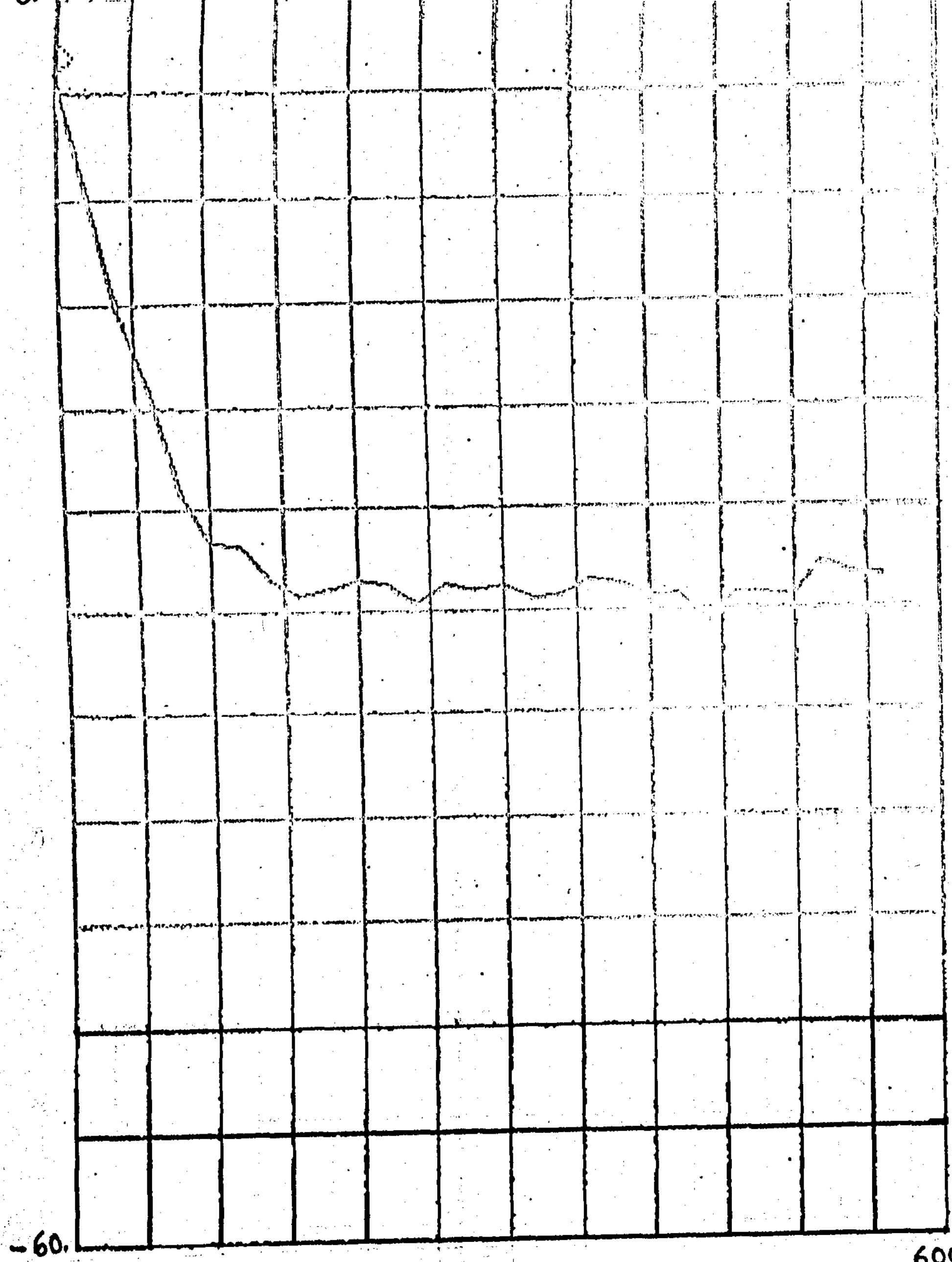

Fig.(VI.A9)-

MSE

(dB)

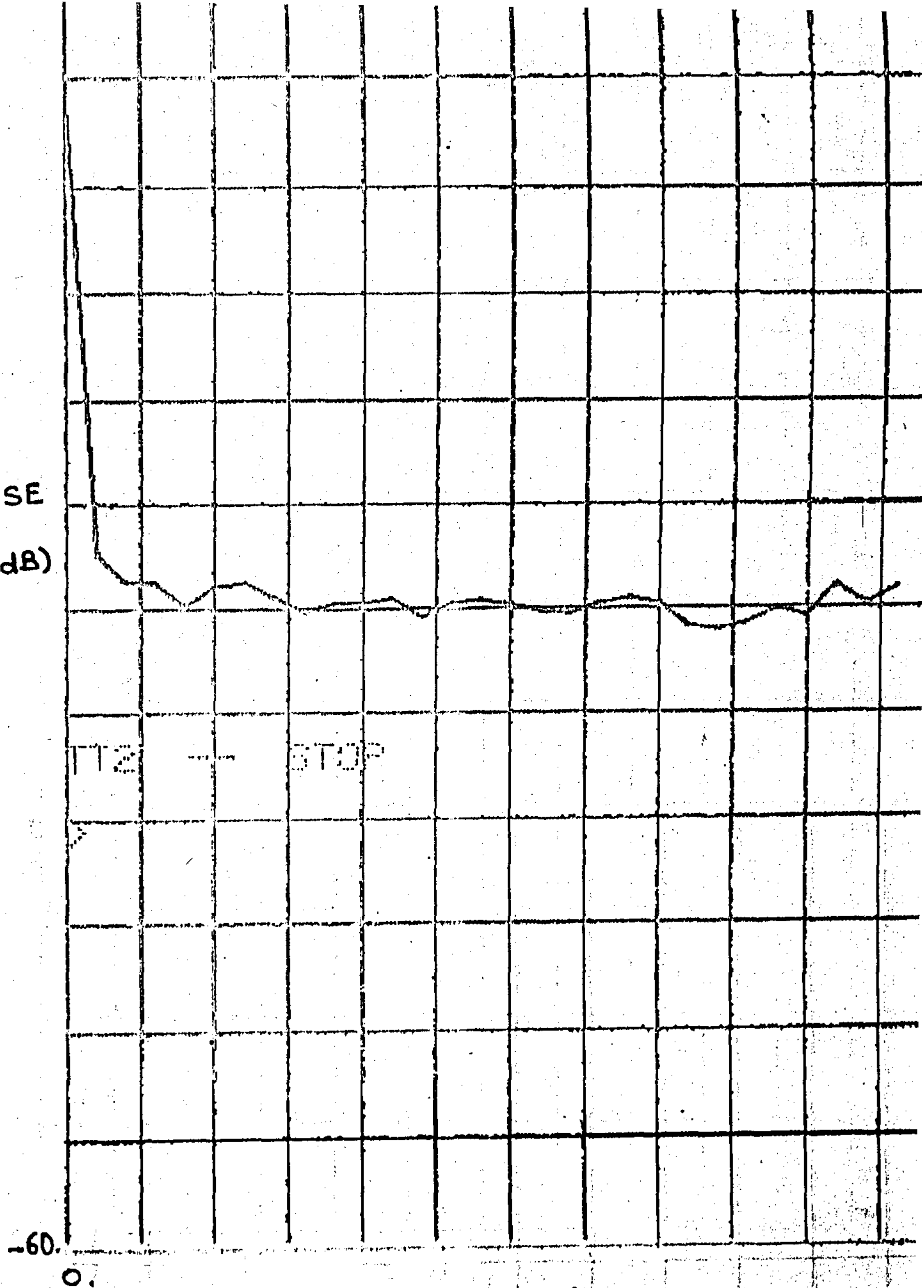

Fig.(VI.10)-

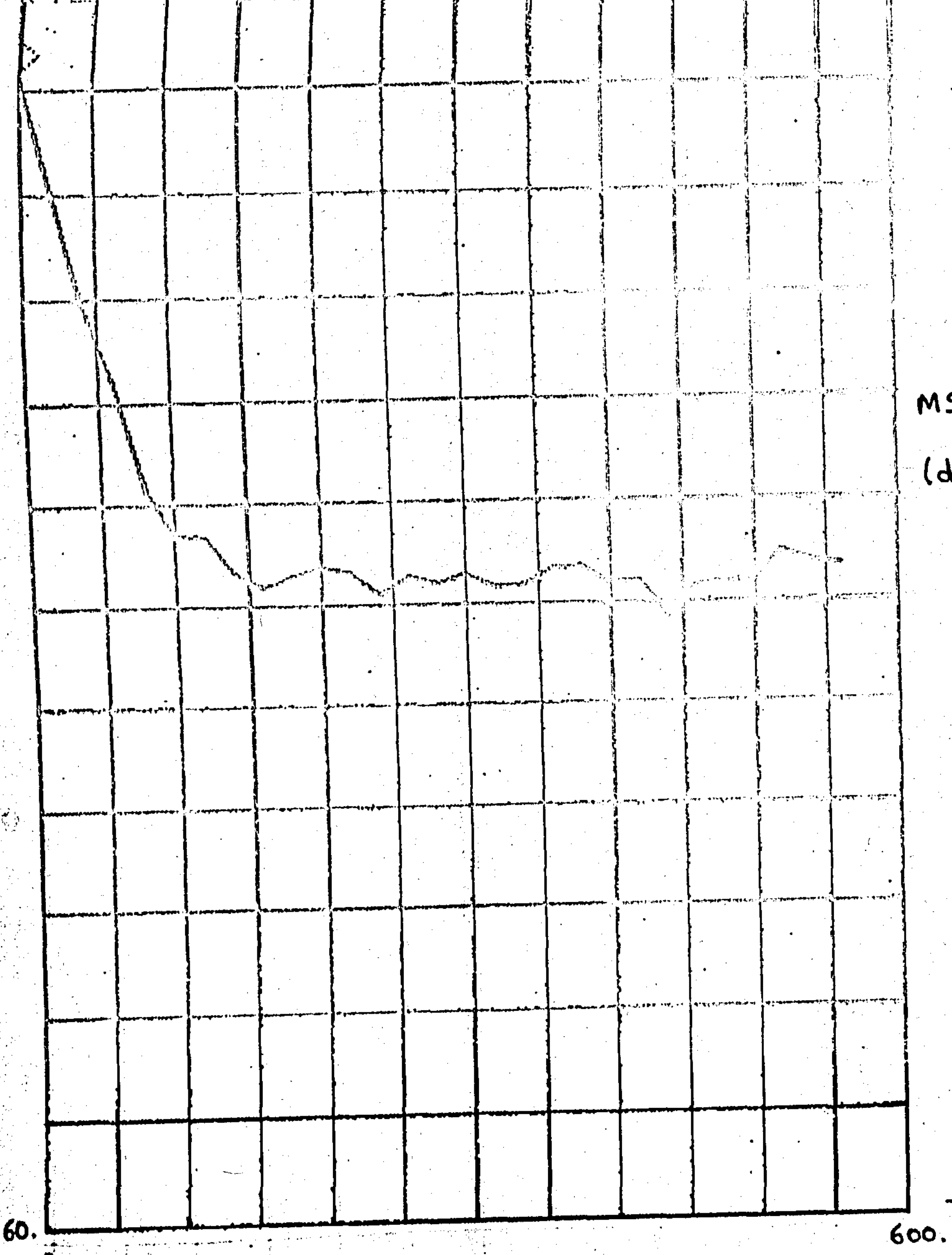

Fig.(VI.A11)

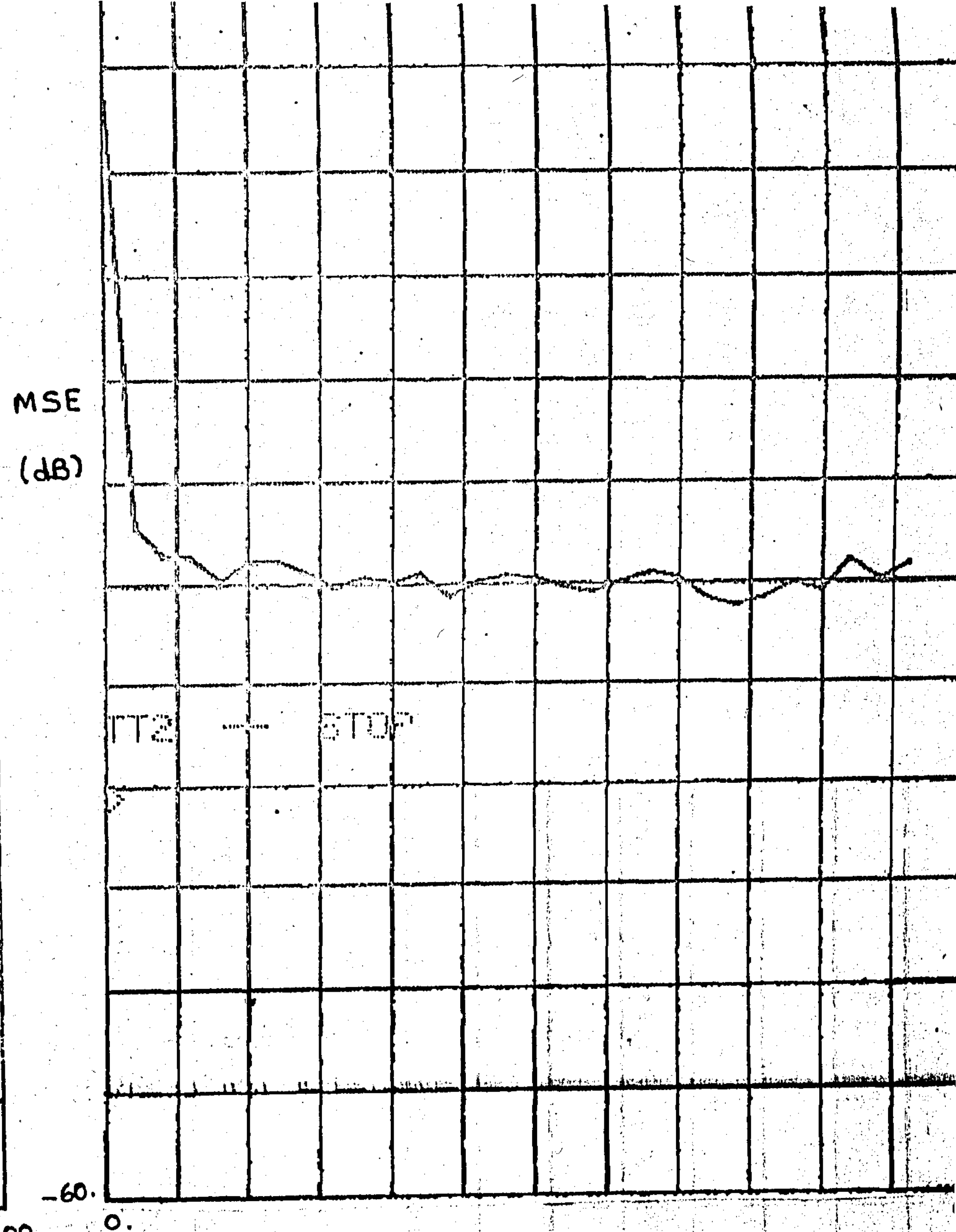

Fig.(VI.A12)

B. Channel B

Channel B has a considerable amount of phase distortion as seen from Fig. (VI.B1). In the presence of white Gaussion noise with variance 0.001 the optimum values for the tap coefficients are

REAL TAP COEFFICIENTS,	IMAGINARY TAP COEFFICIENTS.
-0.1317478670436228D-01	-0.1672873586137986D-01
0.2000786483036919D-01	0.2098693450905405D-01
-0.5451056618375947D-01	-0.2170674695771306D-01
0.1004871895560473D+00	0.3093180899305903D-02
-0.1527122724658671D+00	0.8140008836708339D-01
0.8785502168761486D-01	-0.2673254901208278D+00
0.2090951291875917D+00	0.2324757257778729D+00
0.3950745046185402D+00	0.5686060139069349D-01
0.1408563014909430D-01	-0.1076625183795648D+00
0.2755989029344683D-02	0.2053382264988014D-01
0.8142711099165590D-03	-0.1624908884499413D-01
-0.7065435480992304D-02	0.1401531861257554D-01
0.7294822889385953D-02	-0.1218286486892295D-01
-0.6358914165685750D-02	0.3717275318306364D-02
0.1168063526757725D-01	0.9697935803837195D-03

MINIMUM ACHIEVEABLE MSE =

0.1500713621621325D-02 -0.2823702175455418D 02dB
NOISE VARIANCE = 0.001000000000000

TRACE =38.0697948062263903

MIN.EIG. = 0.2150350359488008D-01
MAX.EIG. = 0.2680005184413795D-01
RATIO = 0.1246310943046460D-01

The eigenvalue ratio is still not much deviated from unity, but the best step-size is 0.013 in this case. From Figs (VI.B3) and (VI.B4) it is observed that the stochastic gradient algorithm converges in 150 iterations while the Fast Kalman in 30 iterations.

Fig. (VI.B1)- Frequency Response

Fig. (VI.B2)- Impulse Response

Fig. (VI.B3)- Stochastic Gradient Algorithm
Infinite Precision $\alpha = 0.013$, $\delta^2 = 0.001$

Fig. (VI.B4)- Fast Kalman Algorithm
Infinite Precision $\delta^2 = 0.001$

Fig. (VI.B1)- Frequency Response.

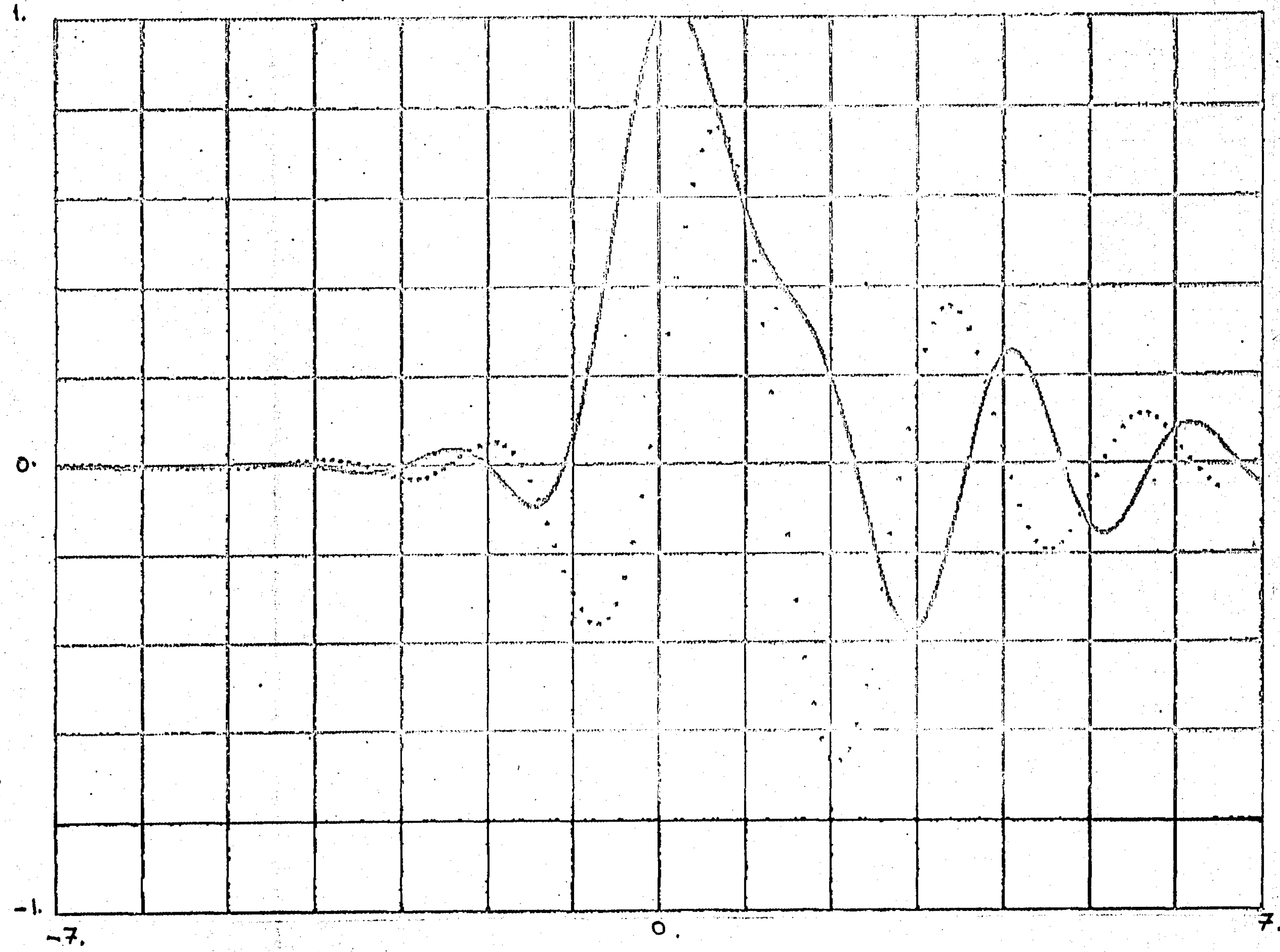

Fig. (VI.B2)- Impulse Response.

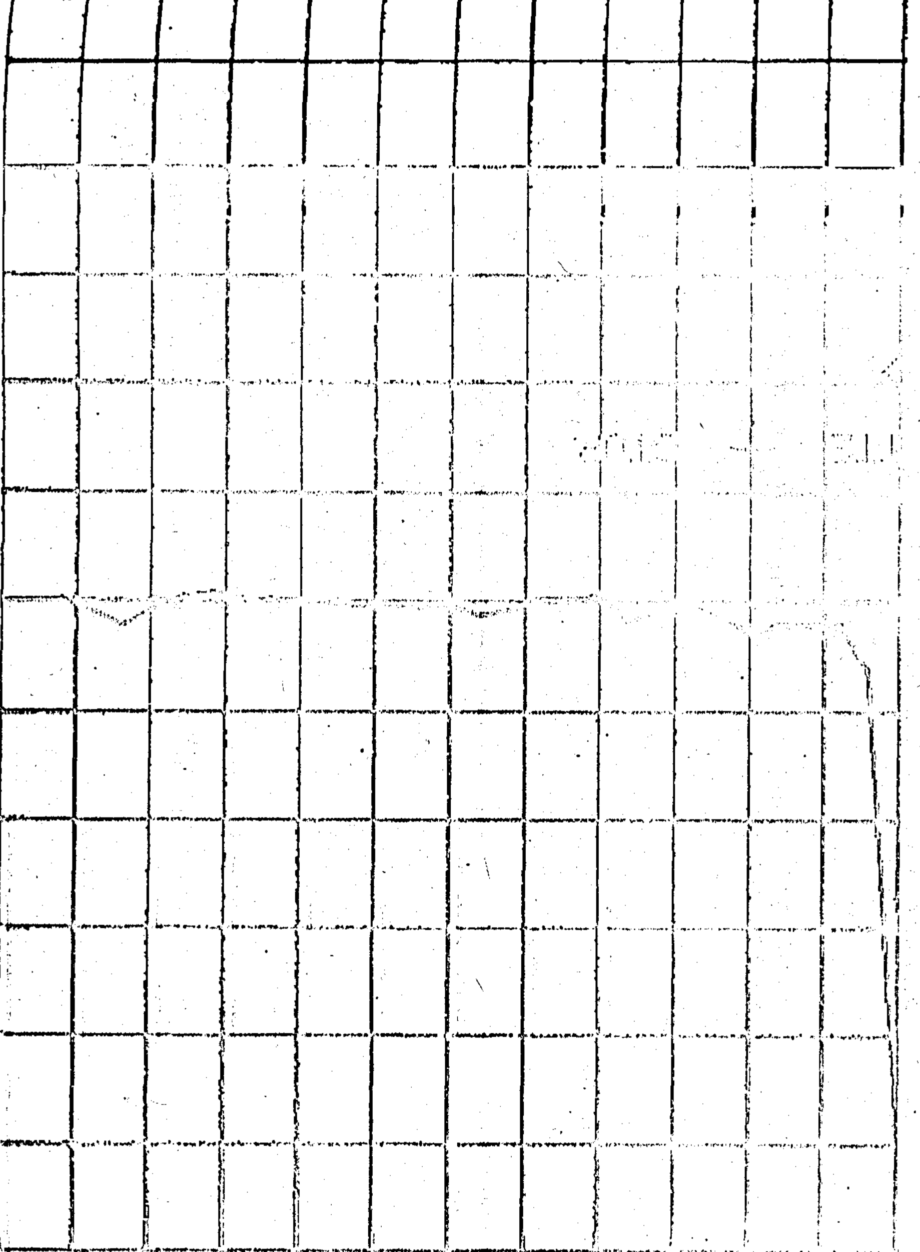

Fig. (VI.B4)

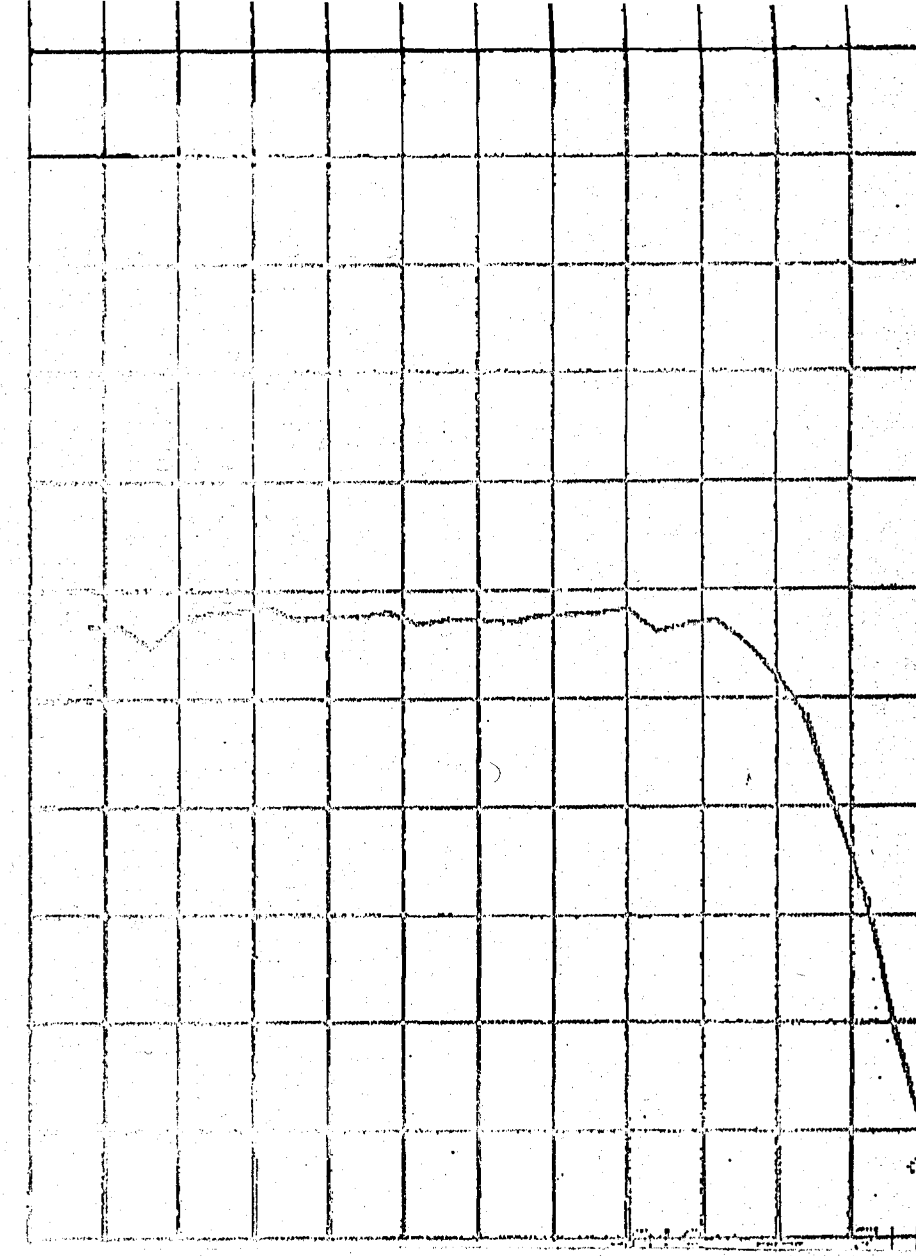

Fig. (VI.B3)

C. Channel C

In Figs. (VI.C1) and (VI.C2) frequency response and the impulse response characteristics for this considerably distorted channels are shown.

REAL TAP COEFFICIENTS,	IMAGINARY TAP COEFFICIENTS.
0.2199748614374745D-02	0.1231212200849068D-02
-0.2737524883622233D-02	-0.4433306322675832D-02
0.3856039656441366D-02	0.9535145981841520D-02
-0.5231654967249663D-02	-0.1776794804701473D-01
0.5629117711150412D-02	0.3250881944174183D-01
0.4267215549428600D-02	-0.6436080017914547D-01
-0.9346233785074677D-01	0.1294253521927656D+00
0.8860198920258066D+00	-0.1699278361537856D+00
0.1614247210021405D+00	0.1129218184058295D+00
-0.3437735943892699D-01	-0.4165782645983911D-01
0.2026614834955622D-01	0.1884477225717751D-01
-0.1312005479000683D-01	-0.9002227644476268D-02
0.8677386395320173D-02	0.2962973186576366D-02
-0.5438287840600732D-02	0.5706406693247695D-03
0.2790101719472272D-02	-0.2147427252869720D-02

MINIMUM ACHIEVEABLE MSE =

0.8991309193510100D-03   -0.3046177067541409D 02D8

NOISE VARIANCE = 0.001000000000000

TRACE = 16.9064403146612626

MIN.EIG. = 0.1028857087672275D 01
MAX.EIG. = 0.1196035313216241D 01
RATIO = 0.1162489258758179D 01

The eigenvalues of the input correlation matrix are close to unity and the minimum achieveable MSE is about -30 dB. With the best value for the step-size (0.033) the performance in Fig. (VI.C3) is obtained. As compared with the performance of the Fast Kolman algorithm in Fig. (VI.C4) the stochastic gradient converges approximately 3 times slower than the later.

Fig. (VI.C1)- Frequency Response

Fig. (VI.C2)- Impulse Response

Fig. (VI.C3)- Stochastic Gradient Algorithm
Infinite Precision
= 0.033, = 0.001

Fig. (VI.C4)- Fast Kalman Algorithm
Infinite Precision, $\delta^2 = 0.001$

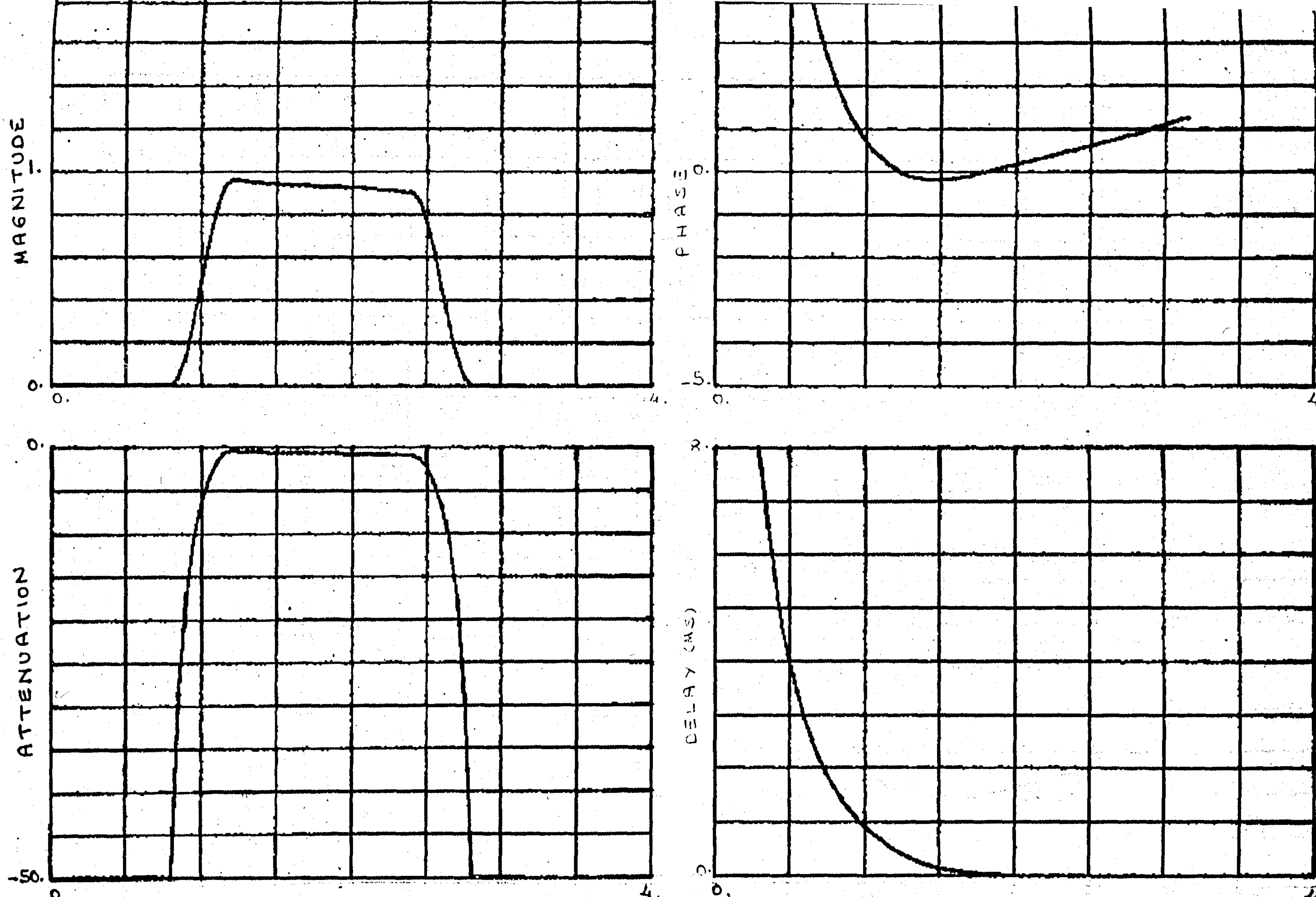

Fig. (VI.C1)

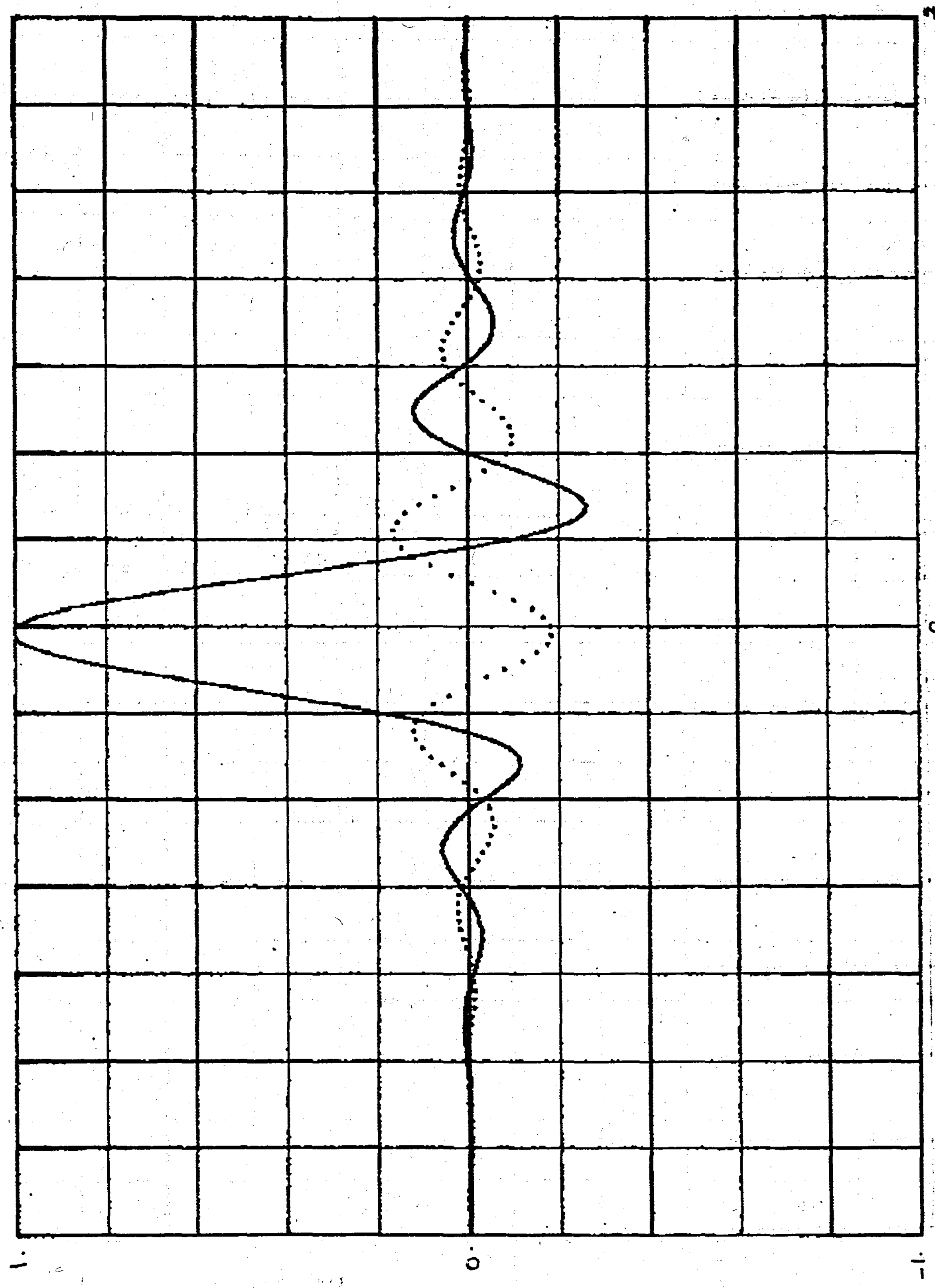

Fig. (VI.C2)

- 76 -

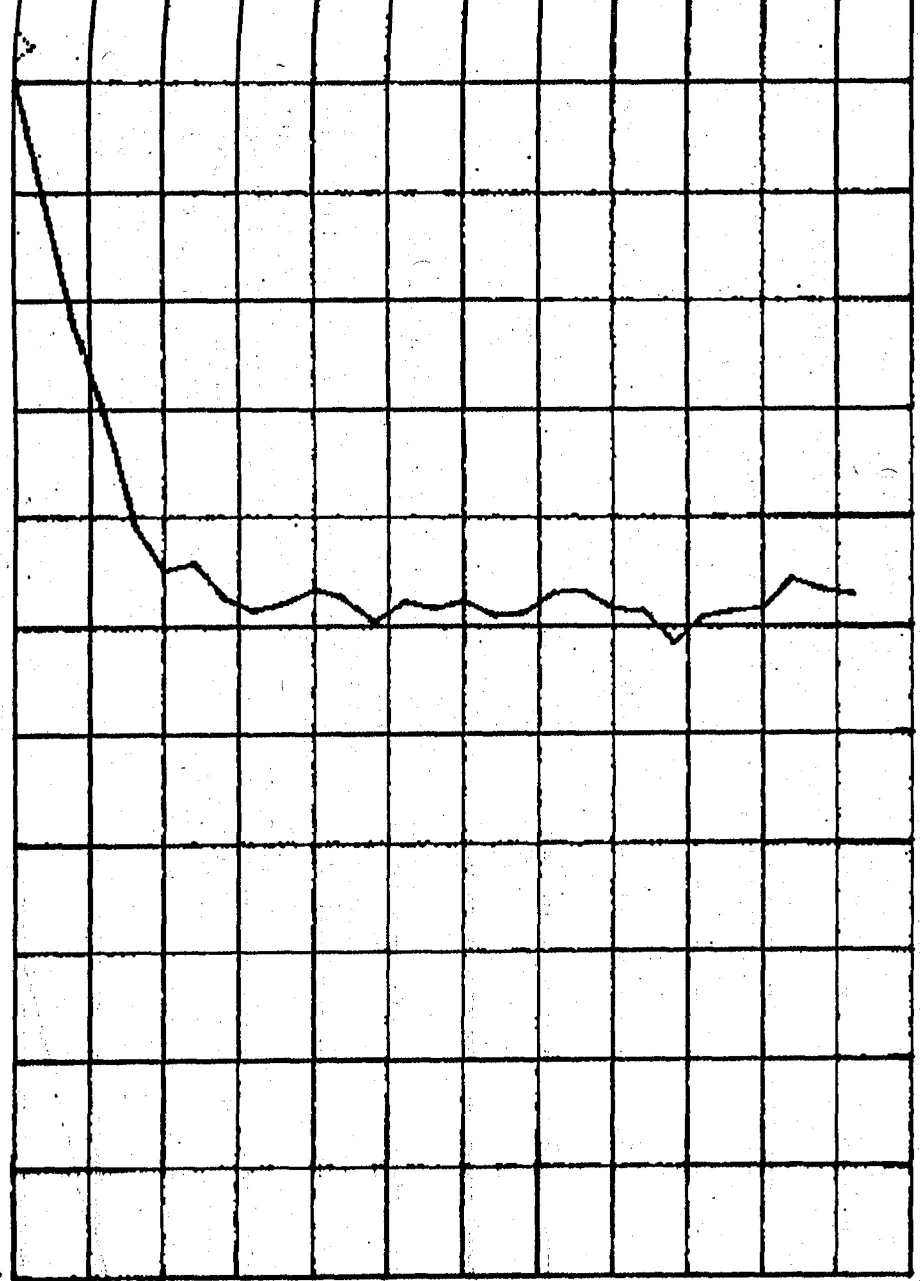

Fig. (VI.C3)

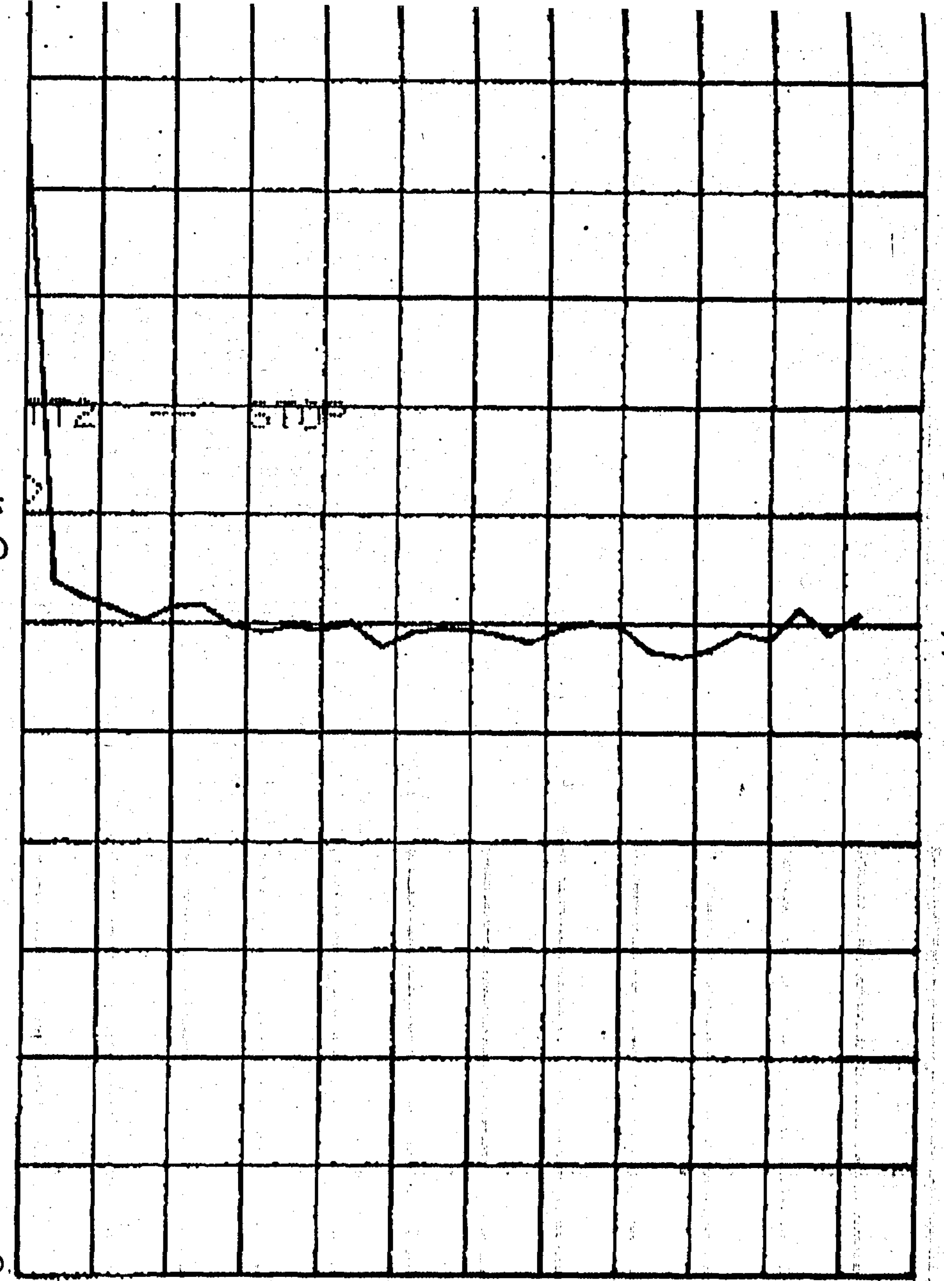

Fig. (VI.C4)

D. Channel D

Channel D whose amplitude and phase distortion is high has on eigenvalue ratio of 5.49:

REAL TAP COEFFICIENTS	IMAGINARY TAP COEFFICIENTS.
0.8716910619748832D-02	-0.3095490391568212D-01
0.1892773379269005D-01	0.4952672451245298D-01
-0.6341656059067754D-01	-0.5553047433243273D-01
0.1330021119060885D+00	0.3206105275624206D-01
-0.2132112850052661D+00	0.8082070102467632D-01
0.1256434584145515D+00	-0.3535117867062451D+00
0.4022709231404664D+00	0.2935548428124993D+00
0.2770225212435913D+00	0.1439473371295534D-01
-0.8369827414741723D-02	-0.4136821640139409D-01
0.1065130902623207D-01	0.1285909450772121D-03
-0.6724345267060802D-02	0.2704450818632886D-02
0.1388108411615949D-02	-0.1015641052715729D-02
-0.2466728074316072D-03	-0.2994237420923816D-02
0.2039274308917951D-02	0.2121001199369898D-02
0.1614386369310859D-02	-0.9358695057731815D-03

MINIMUM ACHIEVEABLE MSE=

0.2677053713430688D-02 -0.2572342914874881D+02DB

NOISE VARIANCE=

TRACE= 36.5845290130643235

MIN.EIG.= 0.8386238770123530D+00
MAX.EIG.= 0.4605974285980999D+01
RATIO= 0.5492300436746512D+01

Although the eigenvalue ratio is high, the Fast Kalman algorithm converges to the minimum MSE in almost 50 iterations without being affected by truncation errors (Fig.(VI.D3), Fig.(VI,D4)). The stochastic gradient algorithm is operated for three different step-sizes. If the step-size is taken to be inversly proportional to the number of taps, that is α=0.033, then the behavior in Fig.(VI.D5) is observed. With this step-size a considerable bias occurs. On the other hand if α is set to 0.014 which is inversly proportional to the trace of the input correlation matrix convergence rate decreases (Fig. (VI.D6).

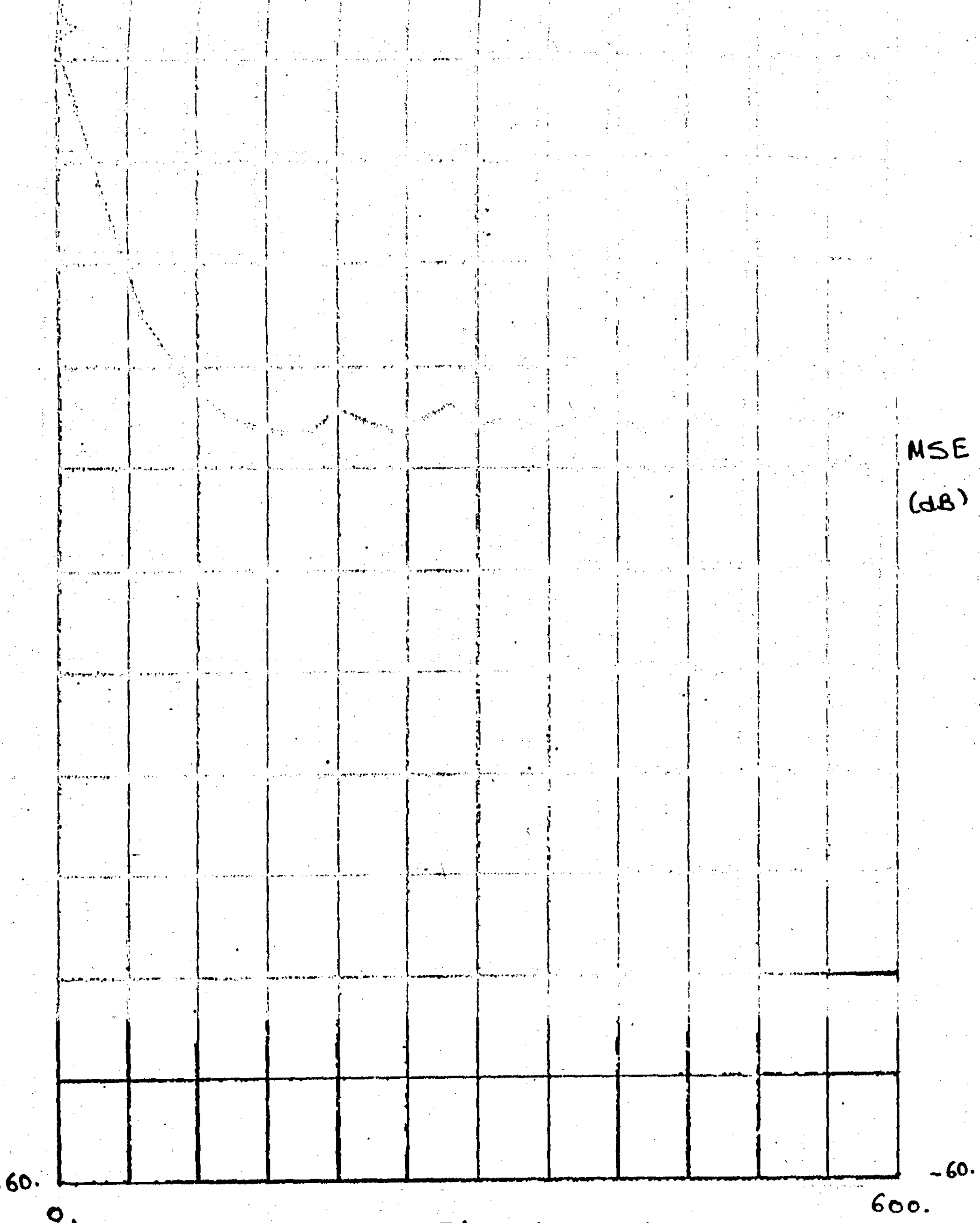

Fig. (VI.D6)

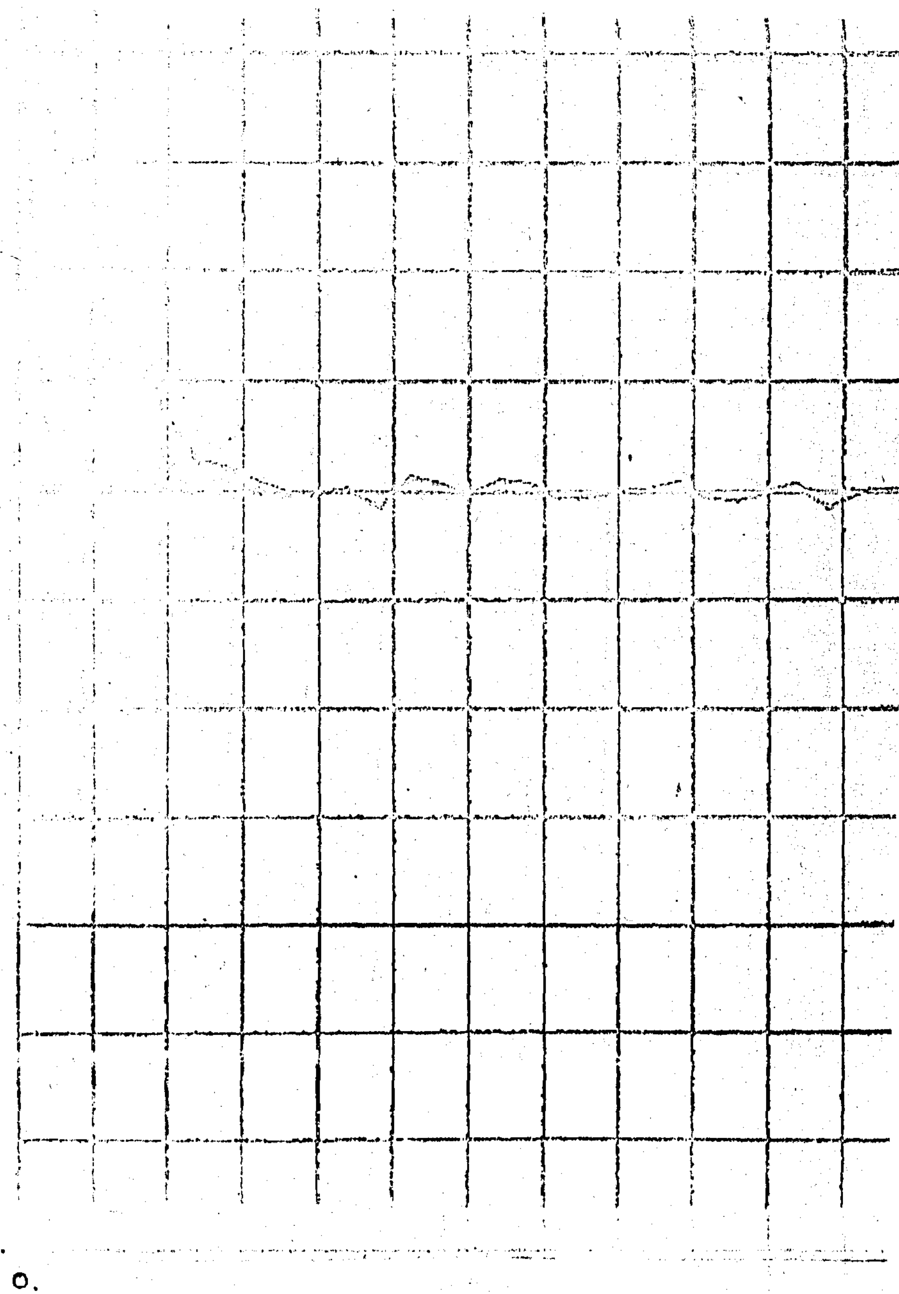

Fig. (VI.D5)

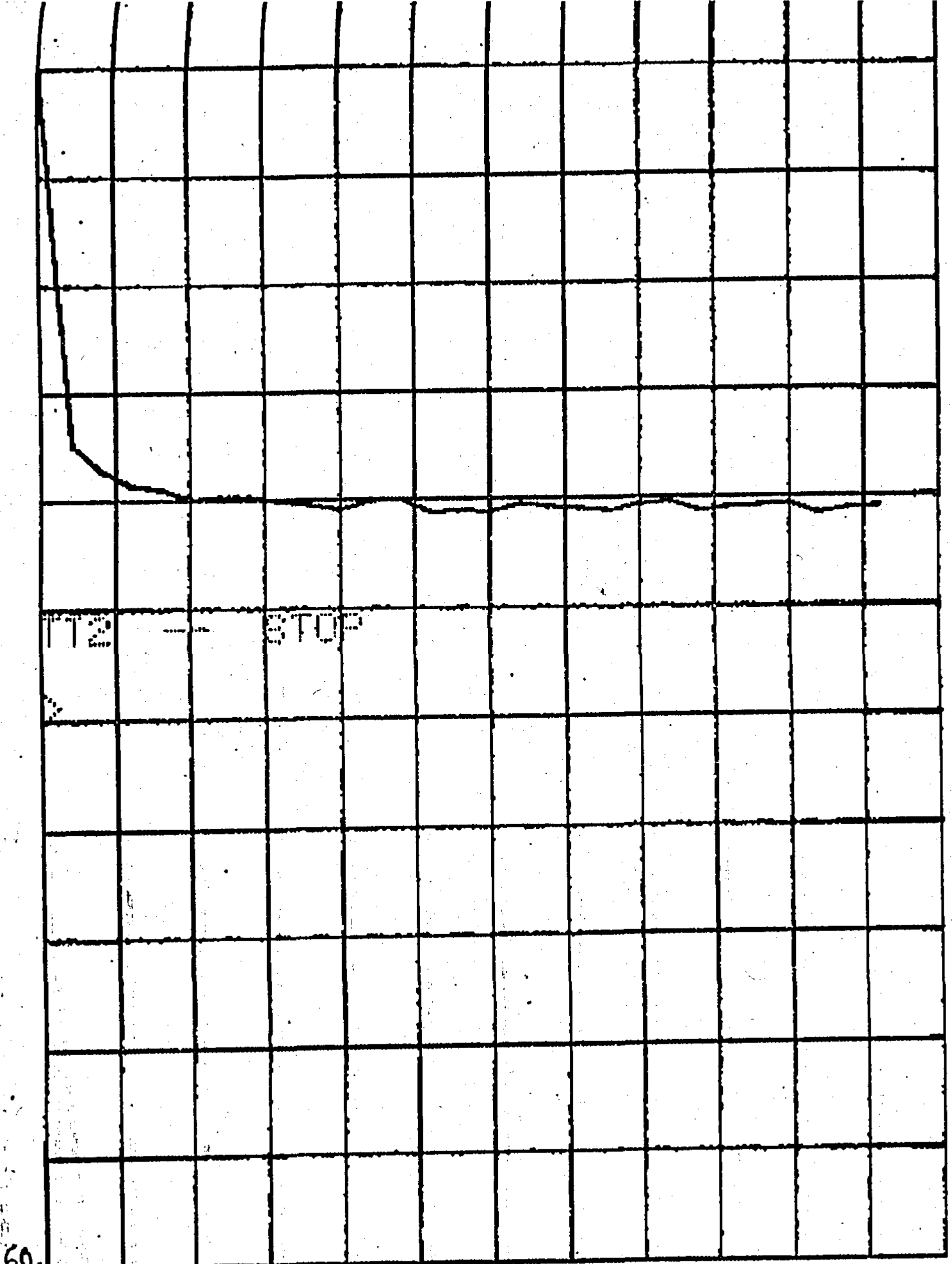

Fig. (VI.D4)

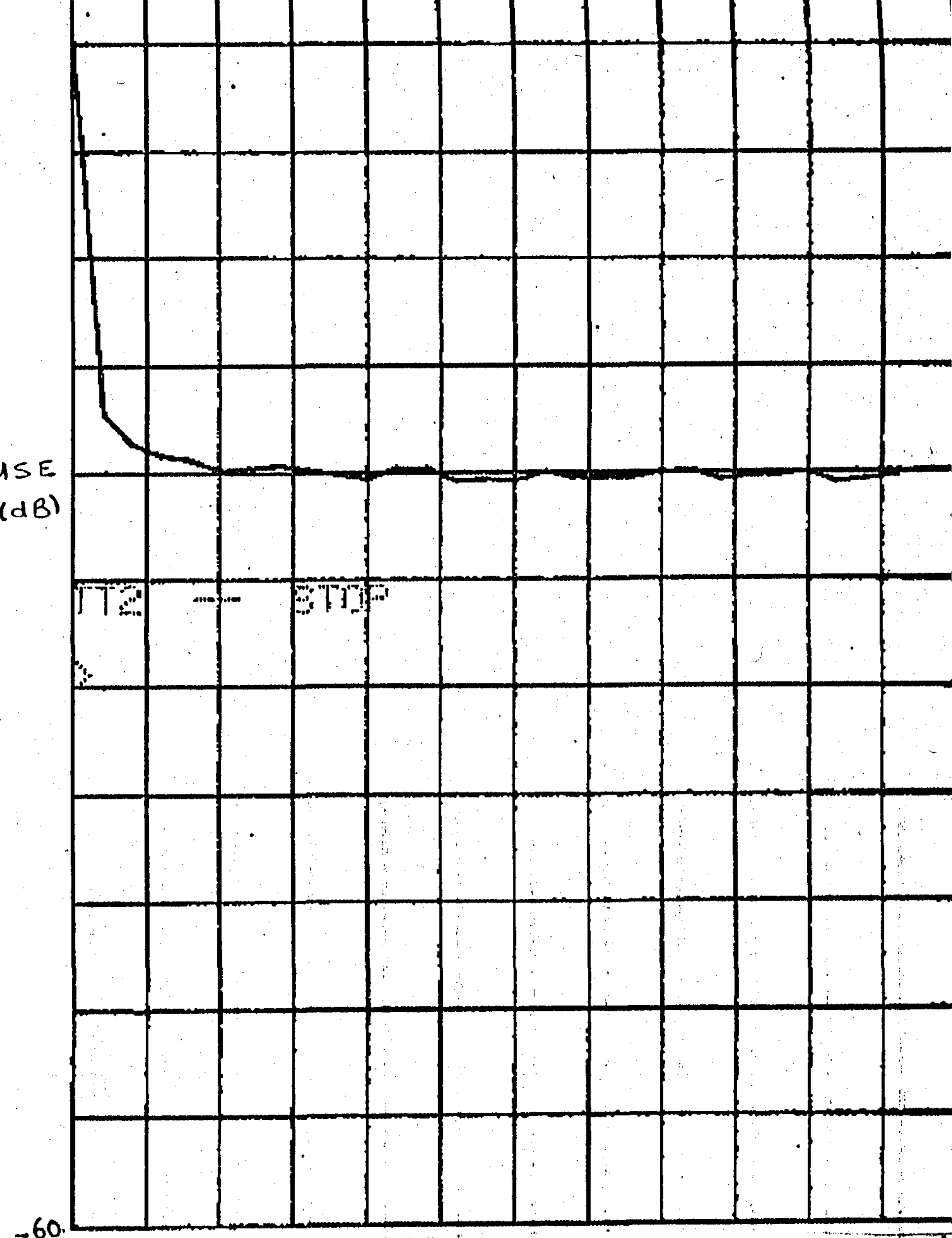

Fig. (VI.D3)

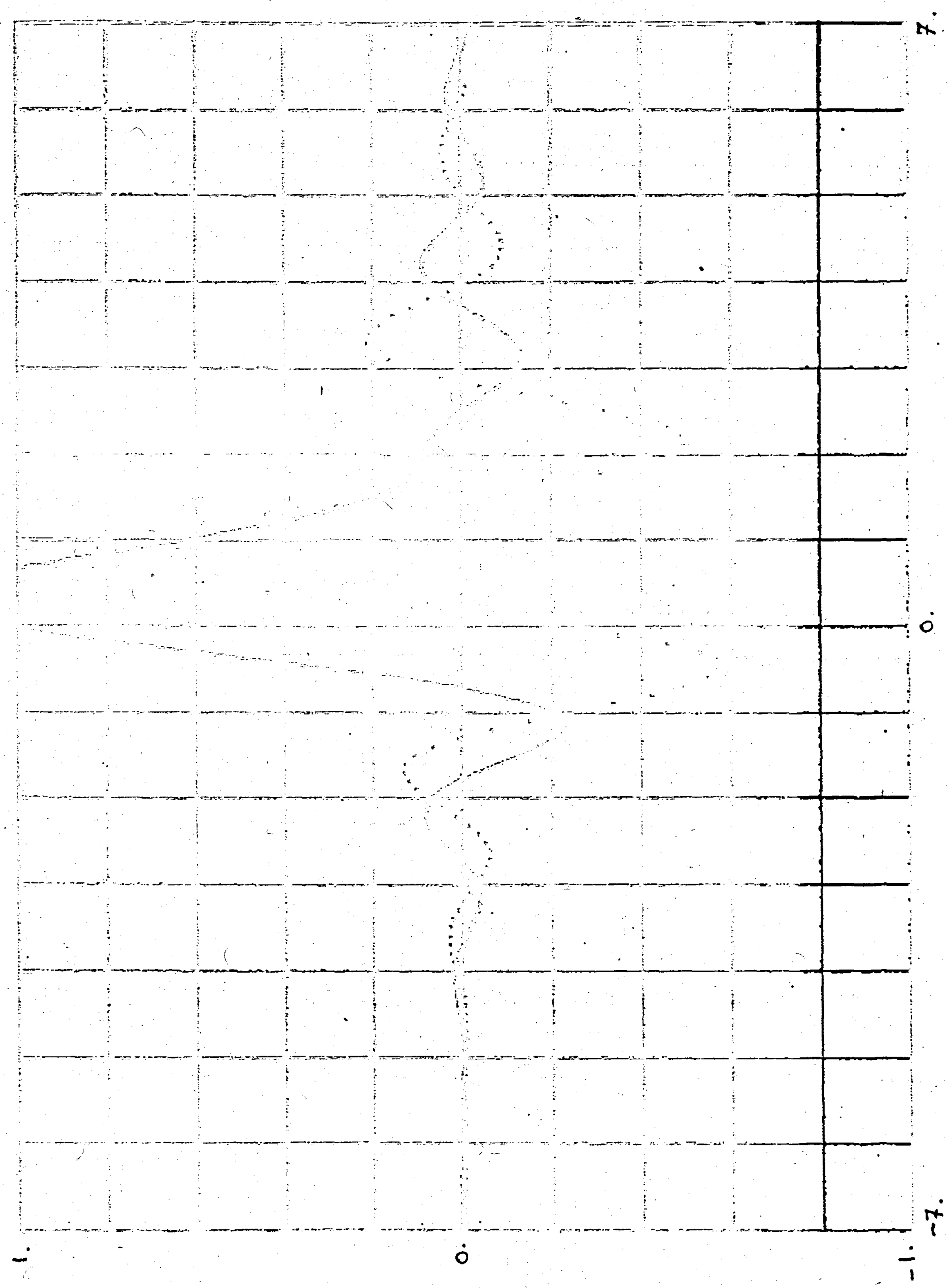

Fig. (VI.D2)

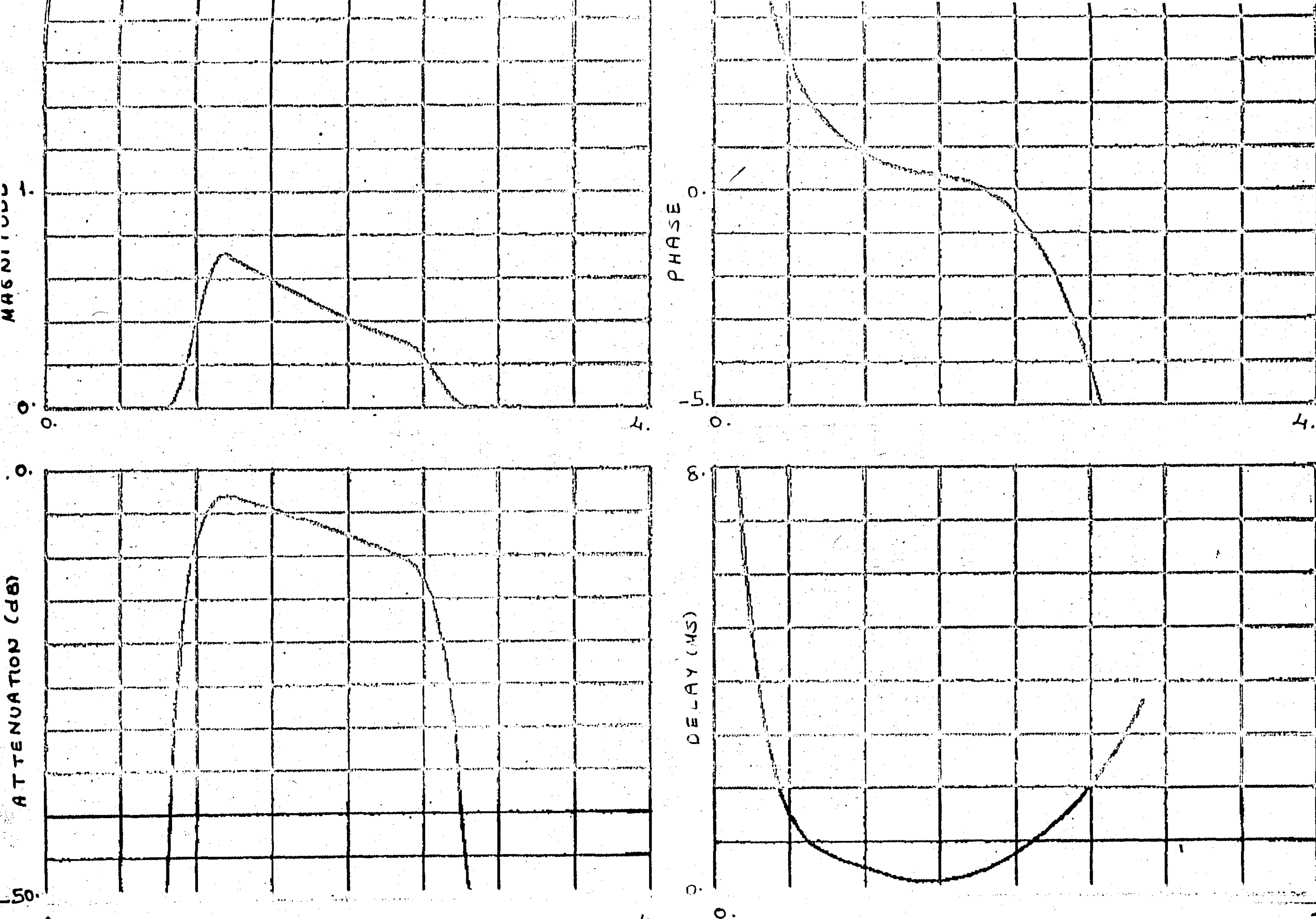
MAGNITUDE
1.
0.
0.
4.
PHASE
0.
-5.
0.
4.
ATTENUATION (dB)
0.
-50.
0.
DELAY (MS)
8.
0.
0.
4.

Fig.(VI.D1)-

E. Channel E

This channel has a nearly ideal amplitude characteristics and a parabolic group delay characteristic. In Fig. (VI.E1) and (VI.E2) the frequency response and the impulse response is seen.

REAL TAP COEFFICIENTS.	IMAGINARY TAP COEFFICIENTS.
0.5500468471424333D-02	0.1569672447388412D-02
0.9516383624429828D-02	-0.4602266745783003D-02
-0.1576618034280010D-01	0.1072220037754939D-01
0.2470477551565770D-01	-0.2578642792629241D-01
-0.2830437489280395D-01	0.6791875978822463D-01
-0.3475758521194567D-01	-0.1664770837478370D+00
0.3048338670102445D+00	0.1779206406981214D+00
0.2007265545216539D+00	-0.4253037313820738D-01
-0.2014545460346599D-01	-0.2028639372586198D-02
-0.7171373200642465D-02	-0.2896074857690417D-02
0.2948534093497227D-02	0.3261878313721546D-02
-0.8699856433437282D-04	-0.4072283076334833D-02
-0.4589004771896555D-03	0.3804017644342893D-02
0.2261684645152325D-02	-0.1200456415223284D-03

MINIMUM ACHIEVEABLE MSE=
0.2790752050254169D-03 -0.3554278747527823D+02DB
NOISE VARIANCE= 0.0010000000000000

TRACE= 73.6379536567414422

MIN.EIG.= 0.4461689570635415D+01
MAX.EIG.= 0.5081751170597266D+01
RATIO= 0.1138974617159110D+01

In Figs.(VI.E3) and (VI.E4) the performance of the stochastic gradient algorithm with different step-sizes are shown. In the first one the step-size α is chosen to be 0.033 as usual. For this value algorithm converges in approximately 300 iterations. The second experiment is with α=0.0066 which is 0.5 1/Trace $\underline{A}$. In this case the algorithm converges in 50 steps. Since the minimum MSE is low trucation highly affects the algorithm and, as show in Fig. (VI.E5) adaptation stops at-32. 5 dB MSE if 8 bits for the tap coefficients are used. However 12 bit quantization gives the same performance as in the enfinite precision (Fig. (VI.E6)). The Fast Kalman algorithm has very fast rate of convergence (approximately 50 iterations (Fig. (VI.E7)). Effects of the truncation are shown in figures (VI.E8) and (VI.E8).

Fig. (VI.E1)- Frequency Response

Fig. (VI.E2)- Impulse Response

Fig. (VI.E3)- Stochastic Gradient Algorithm Infinite Precision
$\alpha = 0.033$, $\delta^2=0.001$

Fig. (VI.E4)- S.G,A.
I.P.
$\alpha=0.0066$, $\delta^2=0.001$

Fig. (VI.E5)- S.G.A.
No.of top coefficient bits = 8
$\alpha=0.0066$, $\delta^2=0.001$

Fig. (VI.E6)- S.G.A.
N.T.C.B = 12
$\alpha=0.0066$, $\delta^2=0.001$

Fig. (VI.E7)- Fast Kalman algorithm
Inf.Pre. $\delta^2=0.001$

Fig. (VI.E8)- Fast Kalman Algorithm
No. of top coeff.bits = 8, $\delta^2=0.001$

Fig. (VI.E9)- Fast Kalman Algorithm
No.of top coefficient bits = 12, $\delta^2=0.001$

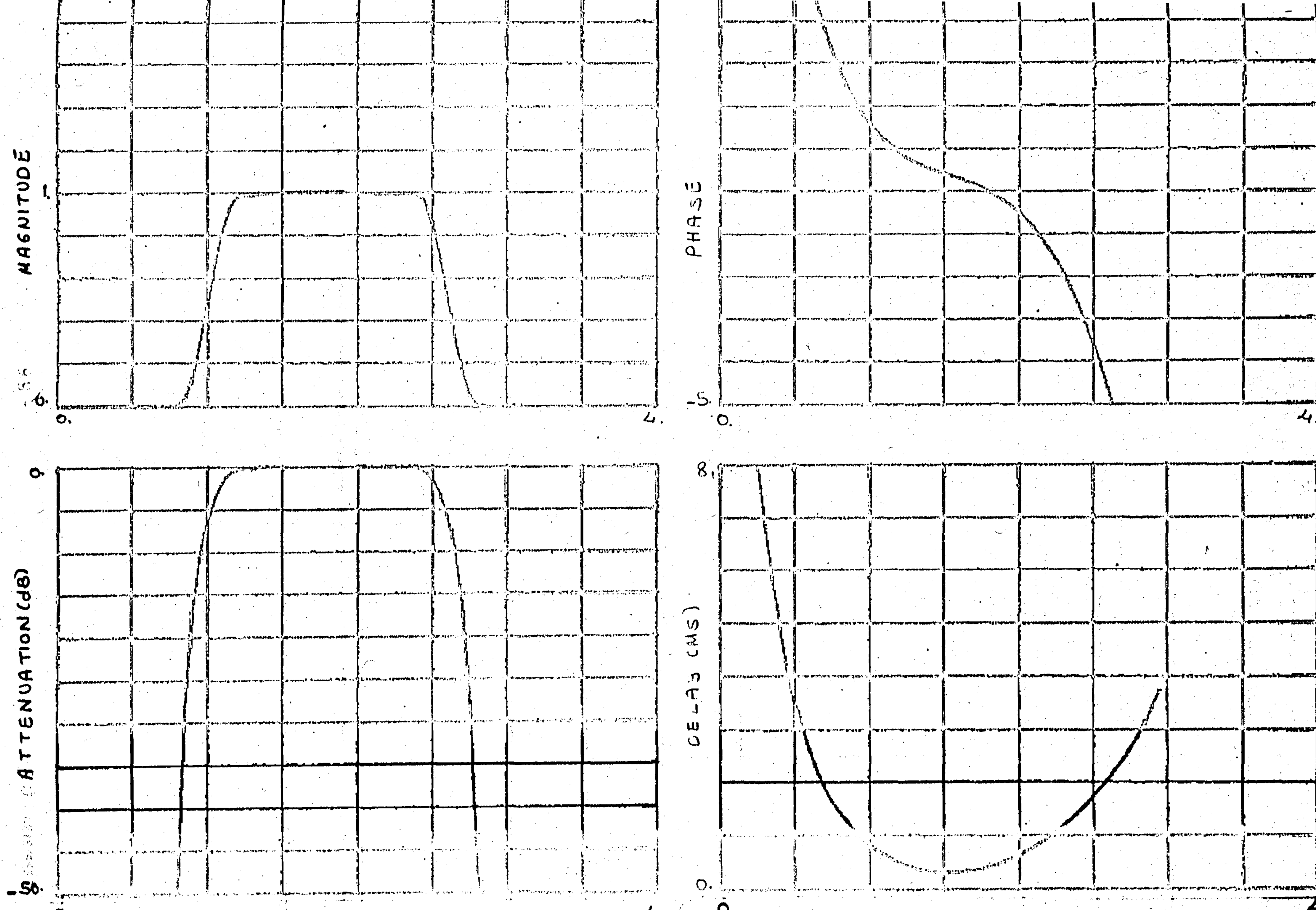
MAGNITUDE
1
.6
0.
4.
PHASE
-.5
0.
4.
ATTENUATION (dB)
0
-50
0.
4.
DELAY (MS)
8
0.
0.
4.

Fig. (VI.E1)

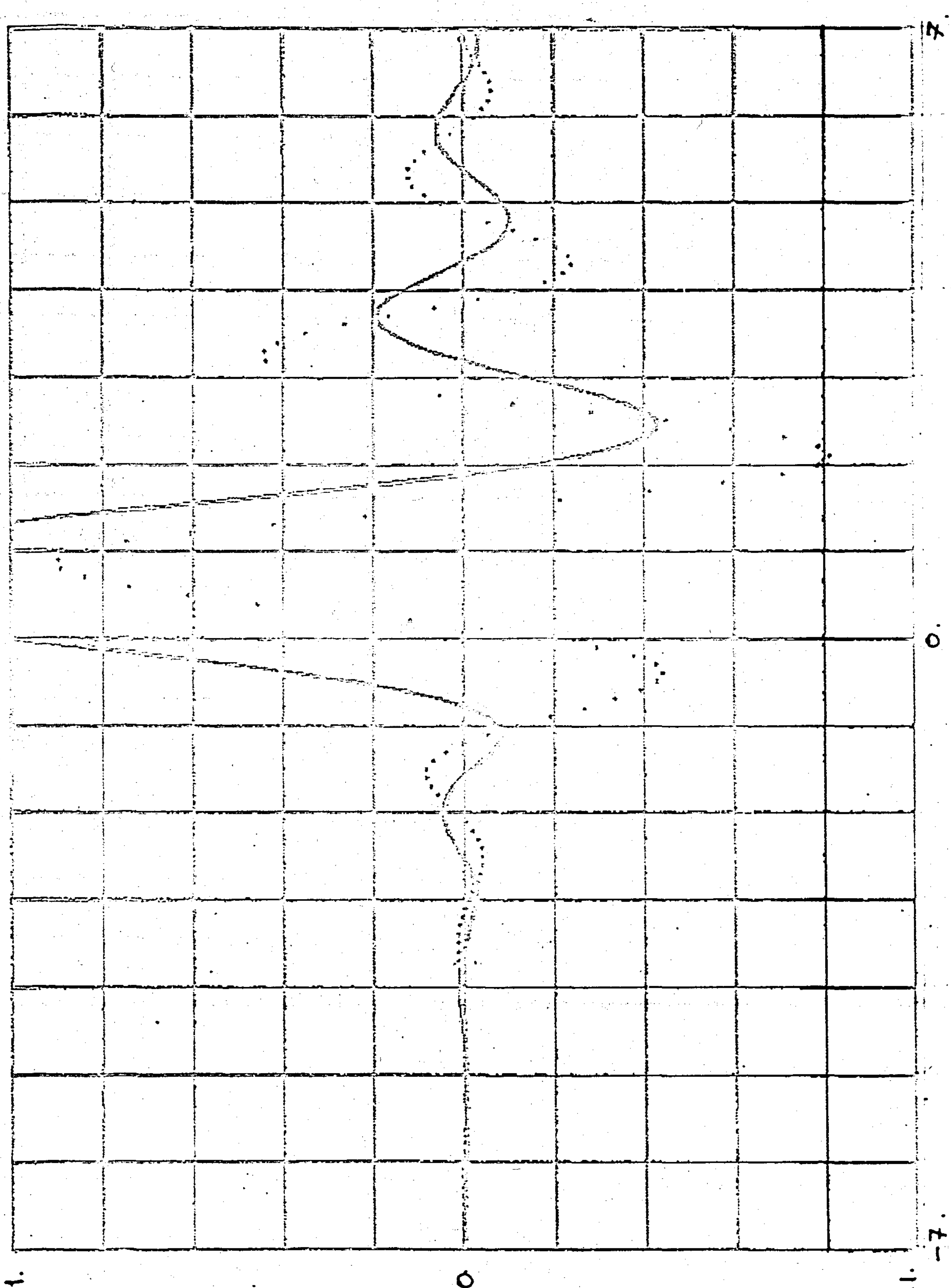

Fig. (VI.E2)

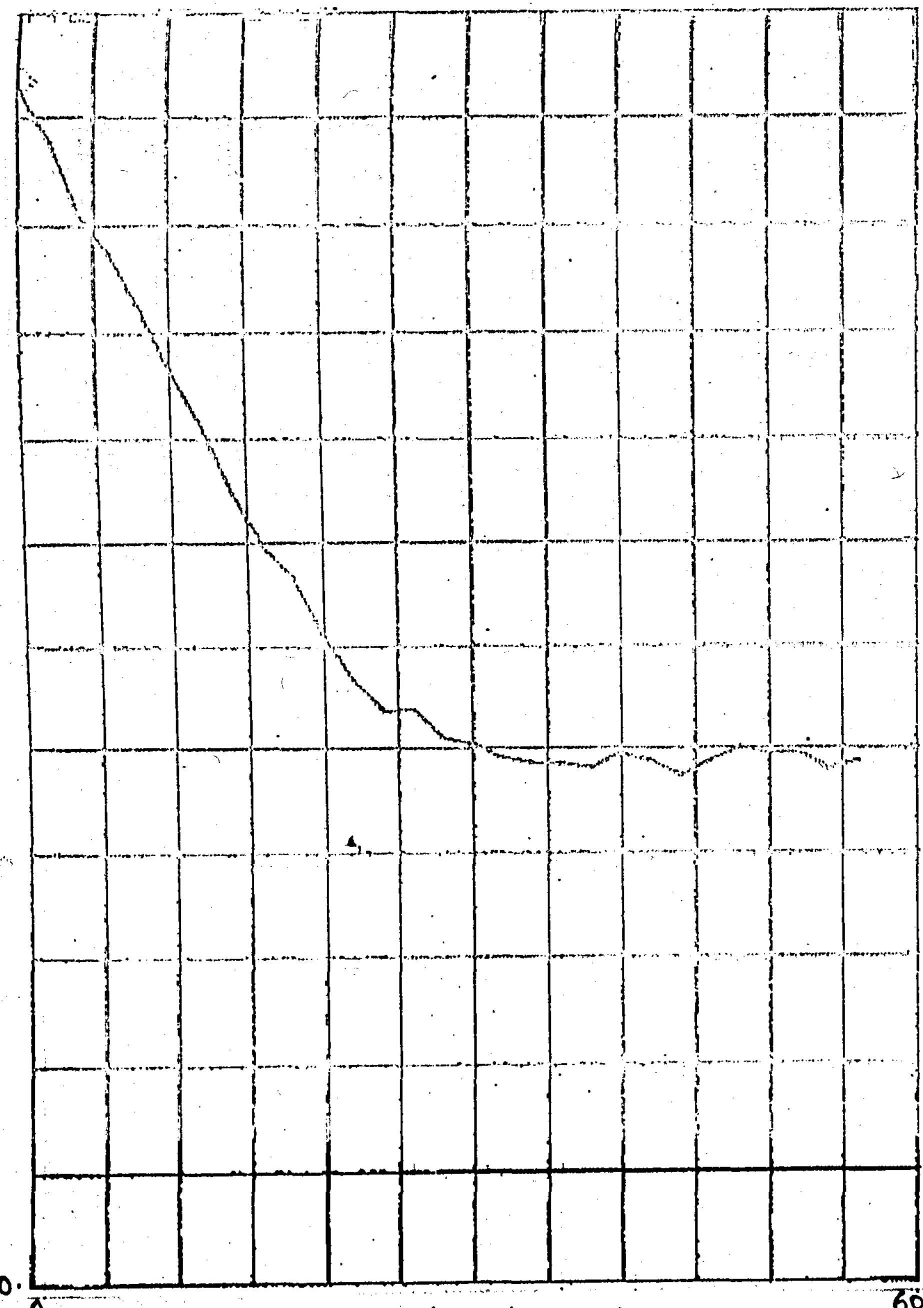

Fig. (VI.E3)

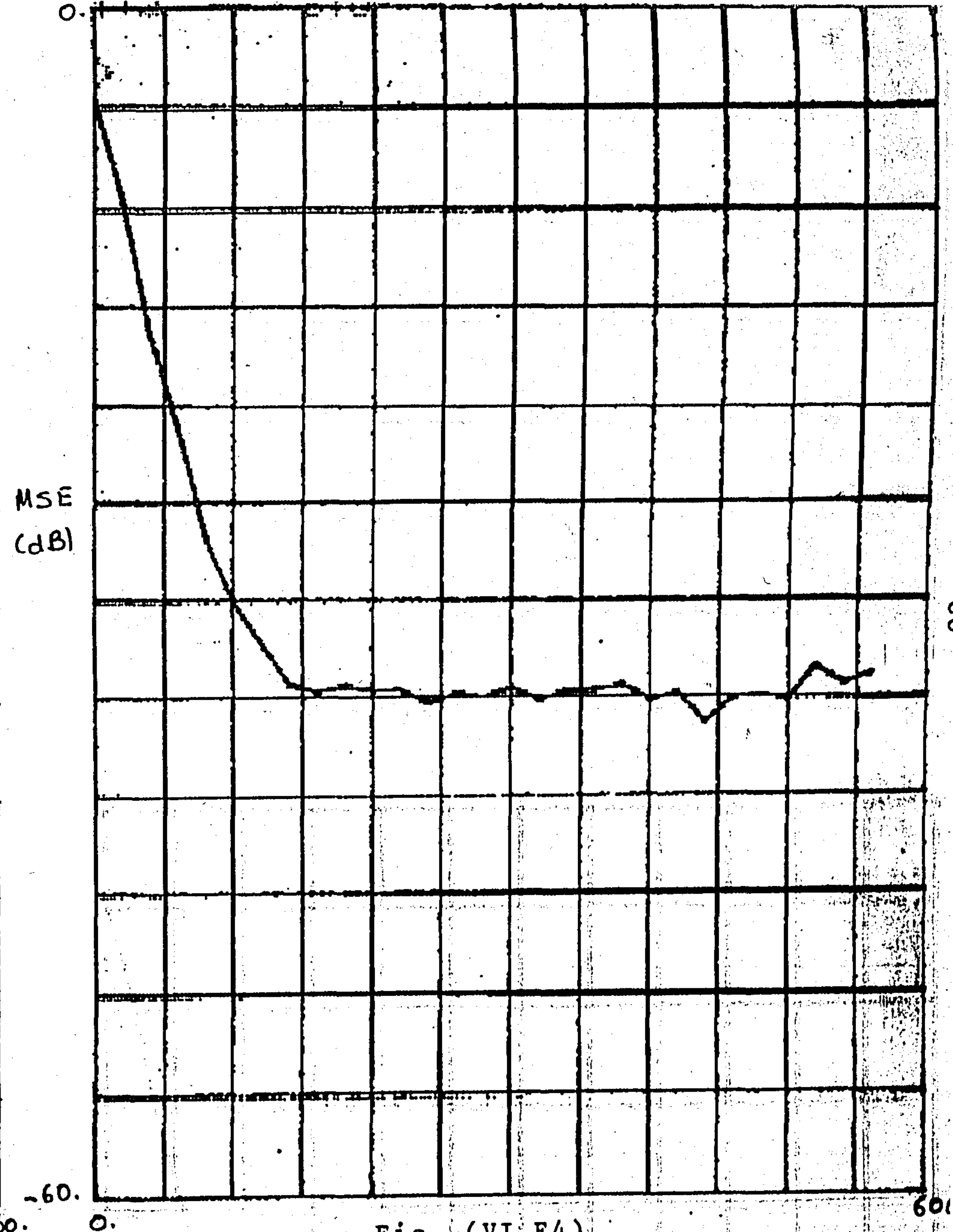

Fig. (VI.E4)

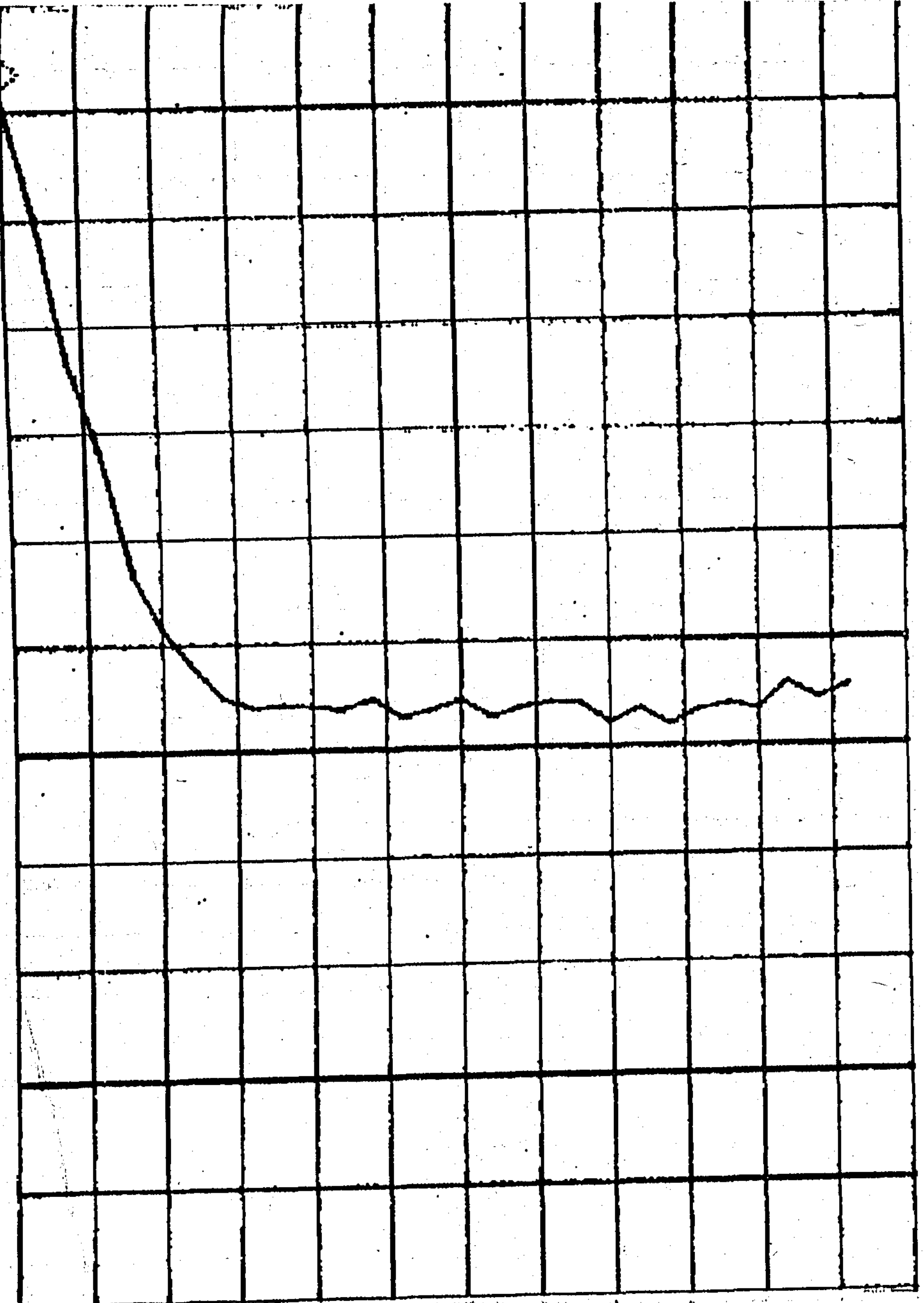

Fig. (VI.E5)

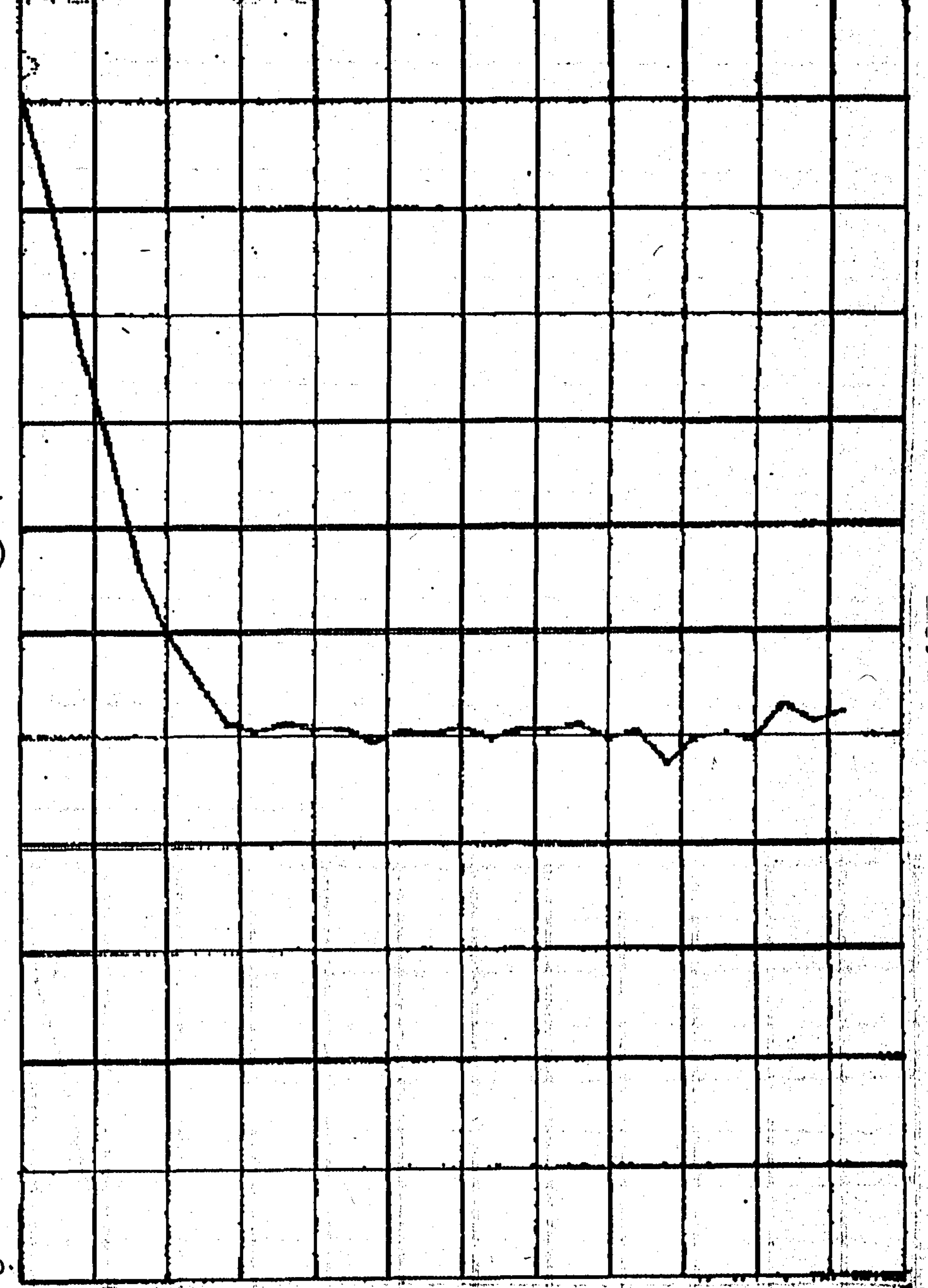

Fig. (VI.E6)

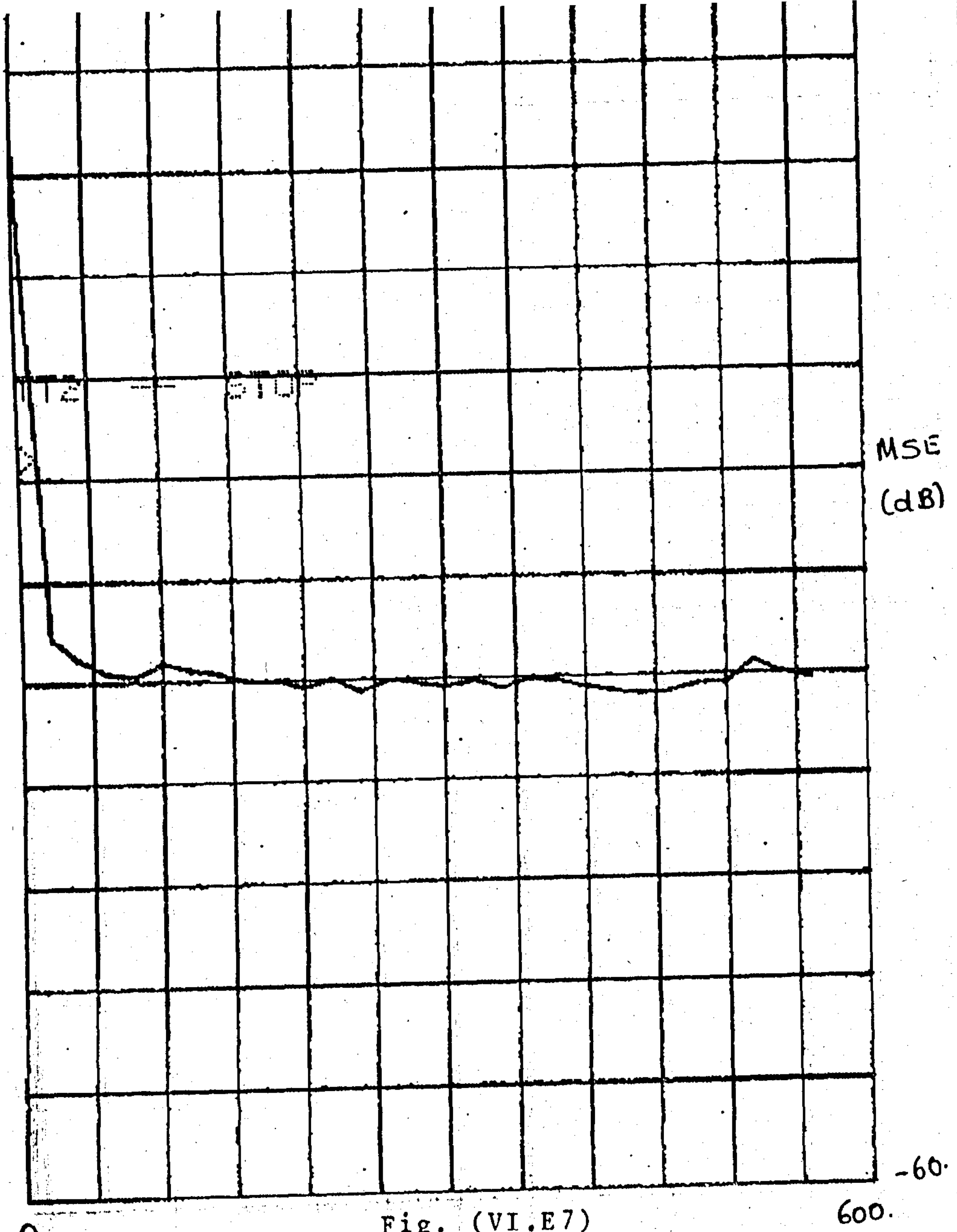

Fig. (VI.E7)

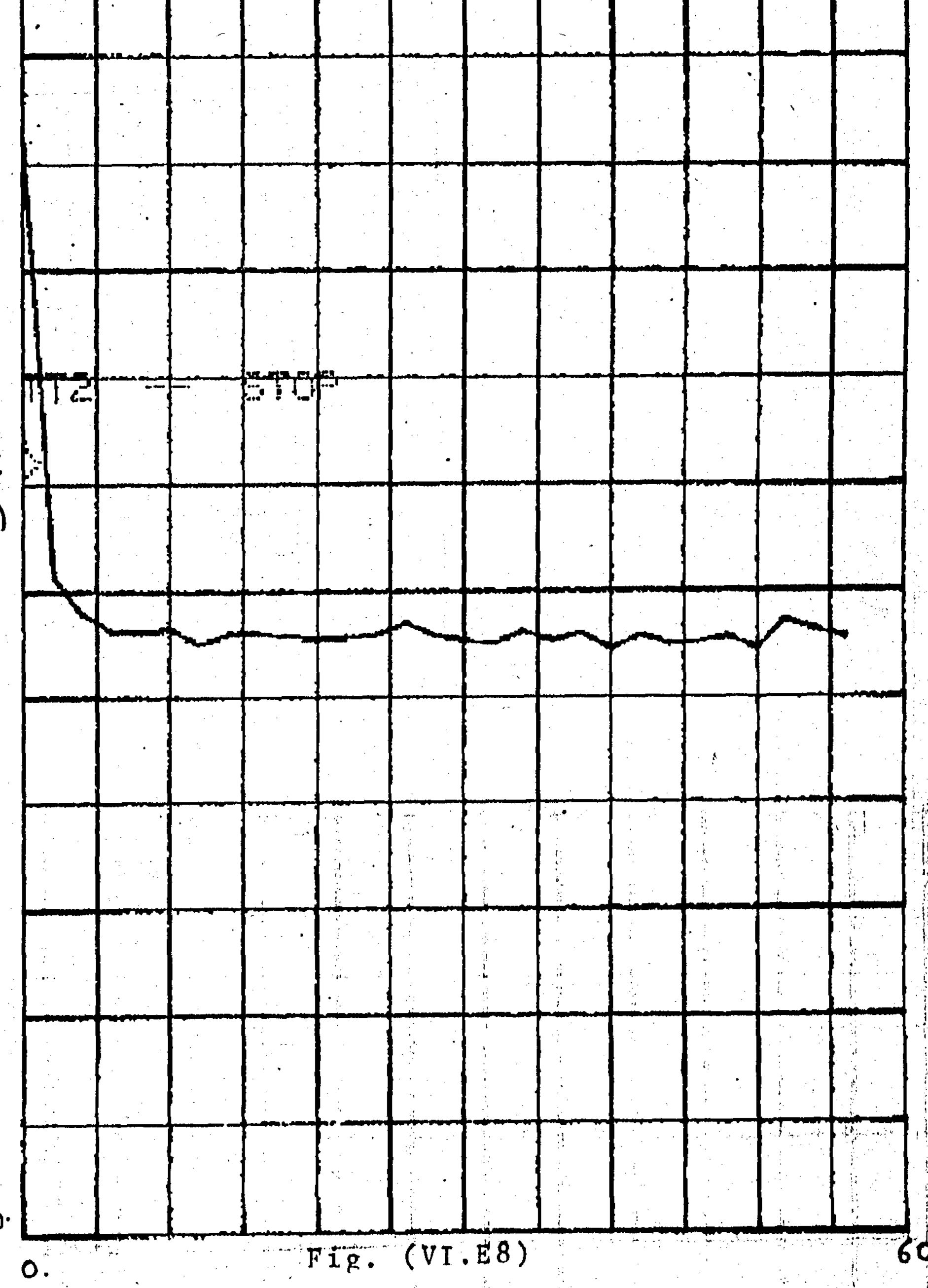

Fig. (VI.E8)

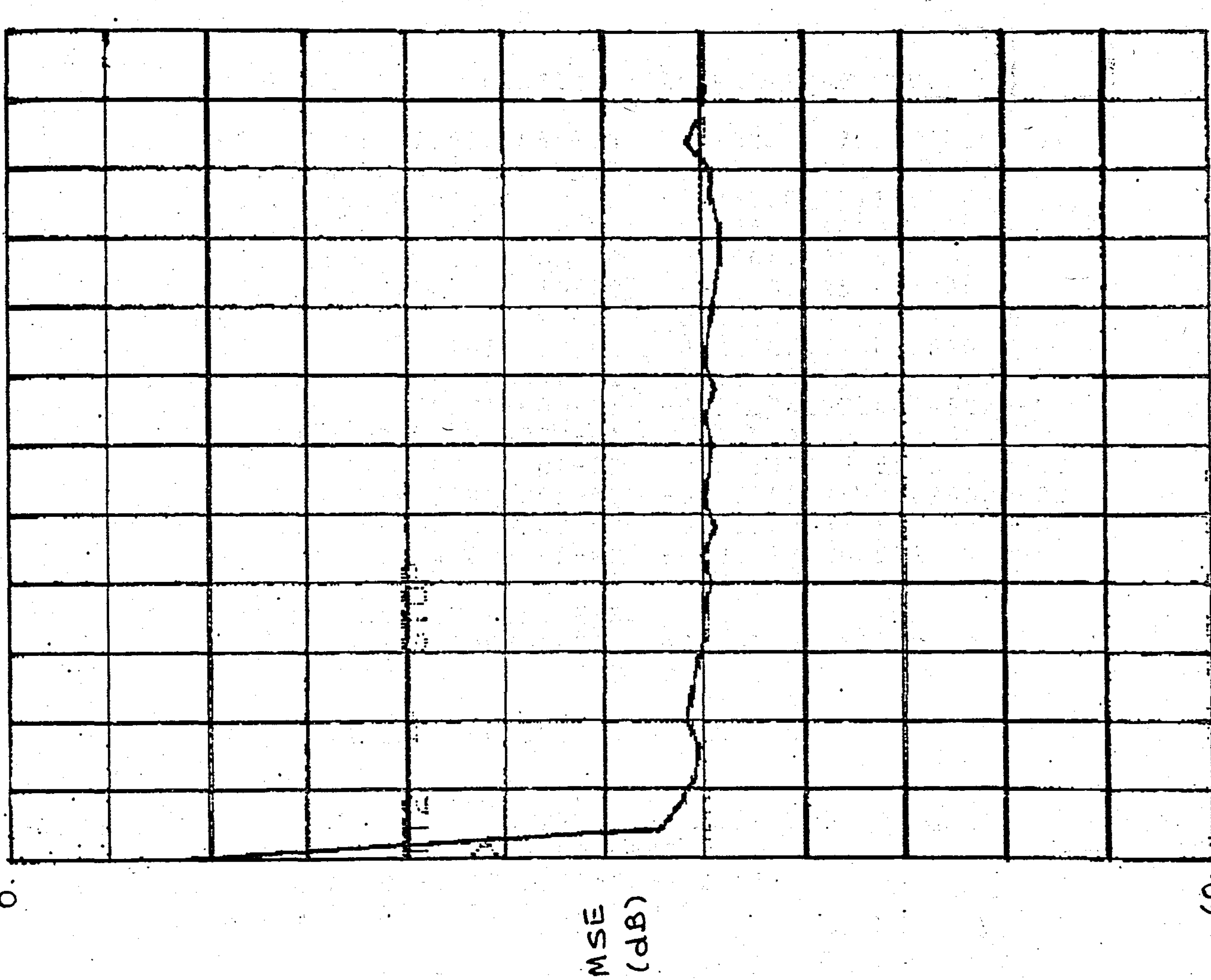

Fig. (VI.E9)

F. Channel F

Among all the channels studied on this has the maximum eigenvalue ratio (11.9) due to both high amplitude and phase distortion.

REAL TAP COEFFICIENTS,	IMAGINARY TAP COEFFICIENTS.
-0.7695480577590565D-02	0.1327085793978889D-01
-0.6574662480936086D-02	-0.2489452416636523D-01
0.3175954240556126D-01	0.2520084511469102D-01
-0.6526151791584818D-01	-0.7889414416471895D-02
0.1035302134629024D+00	-0.4004758824918638D-01
-0.1300093414700421D+00	0.1523622511711665D+00
-0.3032831664057979D-02	-0.4259968937919133D+00
0.1003063076765069D+01	0.2252695766768203D+00
0.5199881029608651D-01	0.1376676126360804D+00
-0.1230323941831160D-01	-0.7432413784449582D-01
0.1929627404654420D-01	0.2908433035521674D-01
-0.1709467940237070D-01	-0.6319908618583793D-02
0.1013738963793820D-01	-0.4785529849189438D-02
-0.2634769586924890D-02	0.7256976778185230D-02
-0.2105537324462474D-02	-0.3581318069921975D-02

MINIMUM ACHIEVEABLE MSE=
0.1527231853620983D-02   -0.2816095026398338D+02DB

NOISE VARIANCE=0.001000000000000

TRACE=21.1058086375025565

MIN.EIG.=0.2757488863108780D+00
MAX.EIG.=0.3286875561588741D+01
RATIO=0.1191981445714030D+02

The stochastic gradient algorithm, with the usual value 0.033 for α convages in 150 iterations but to a biased final MSE (VI.F3).

If the best value of α = 0.025 for this channel is used settling time falls to 225 iterations but to the desired MSE. As-expected Fast Kalman reaches to -28 dB MSE in approximately 50 iterations.

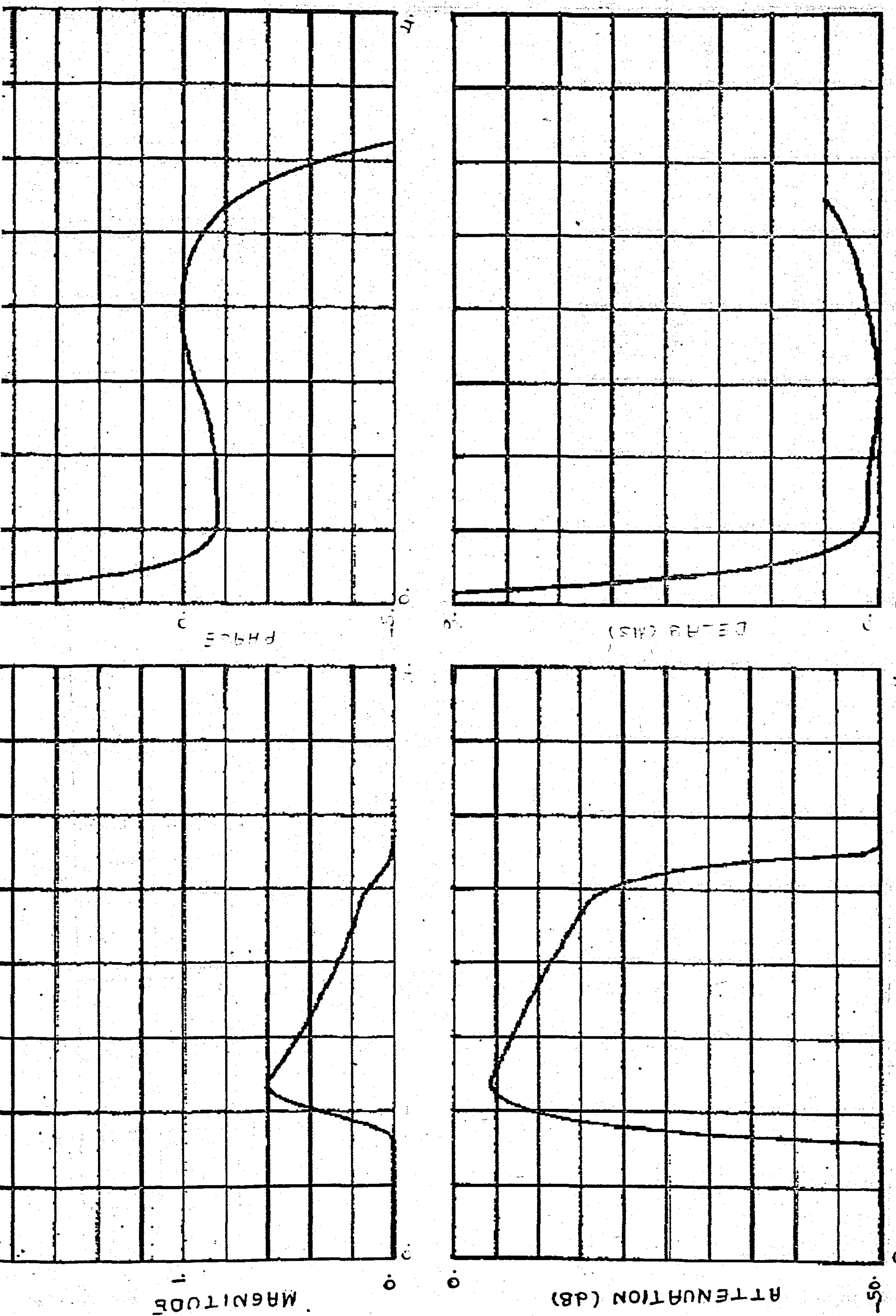

Fig. (VI.F1)

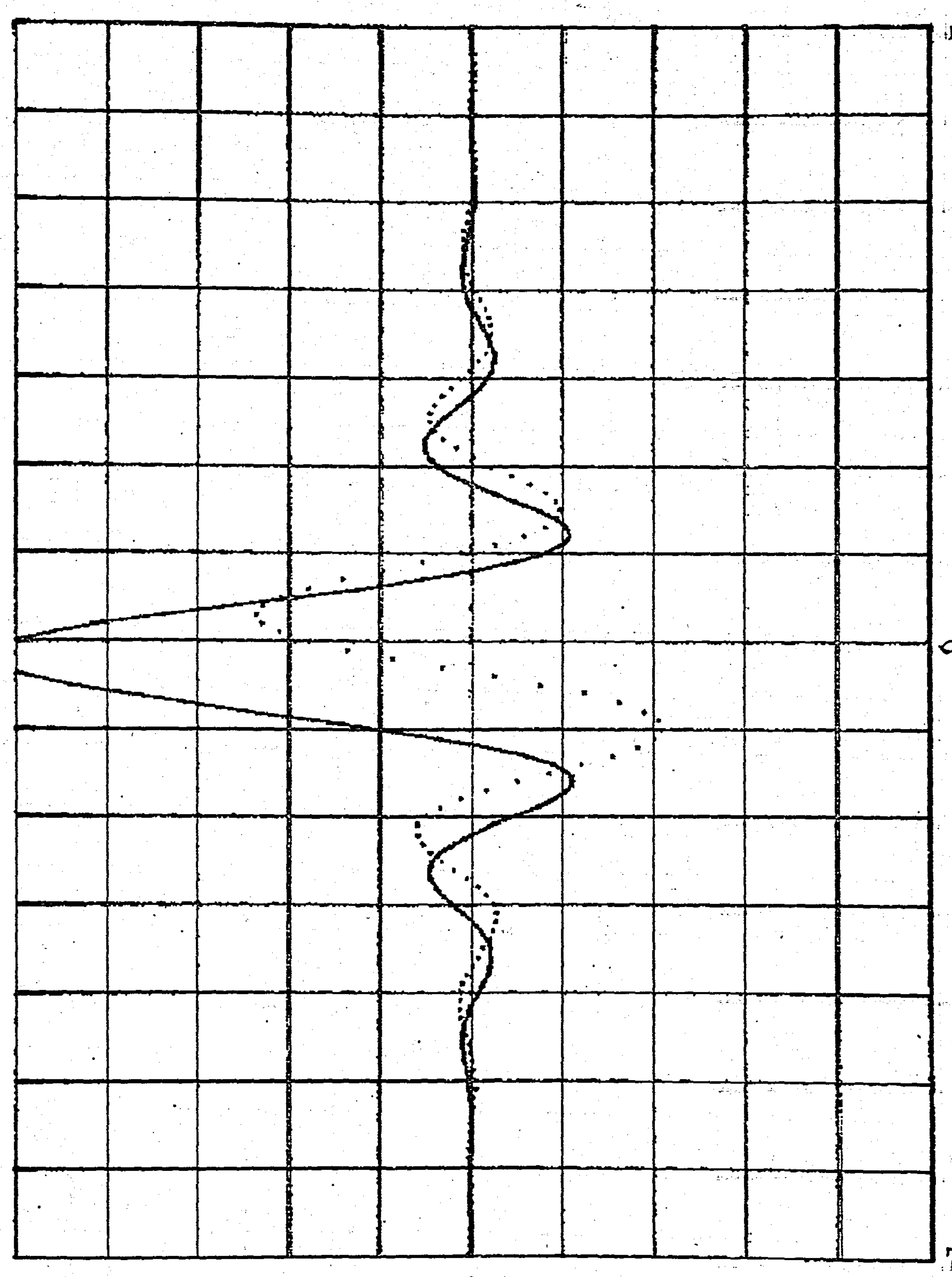

Fig. (VI.F2)

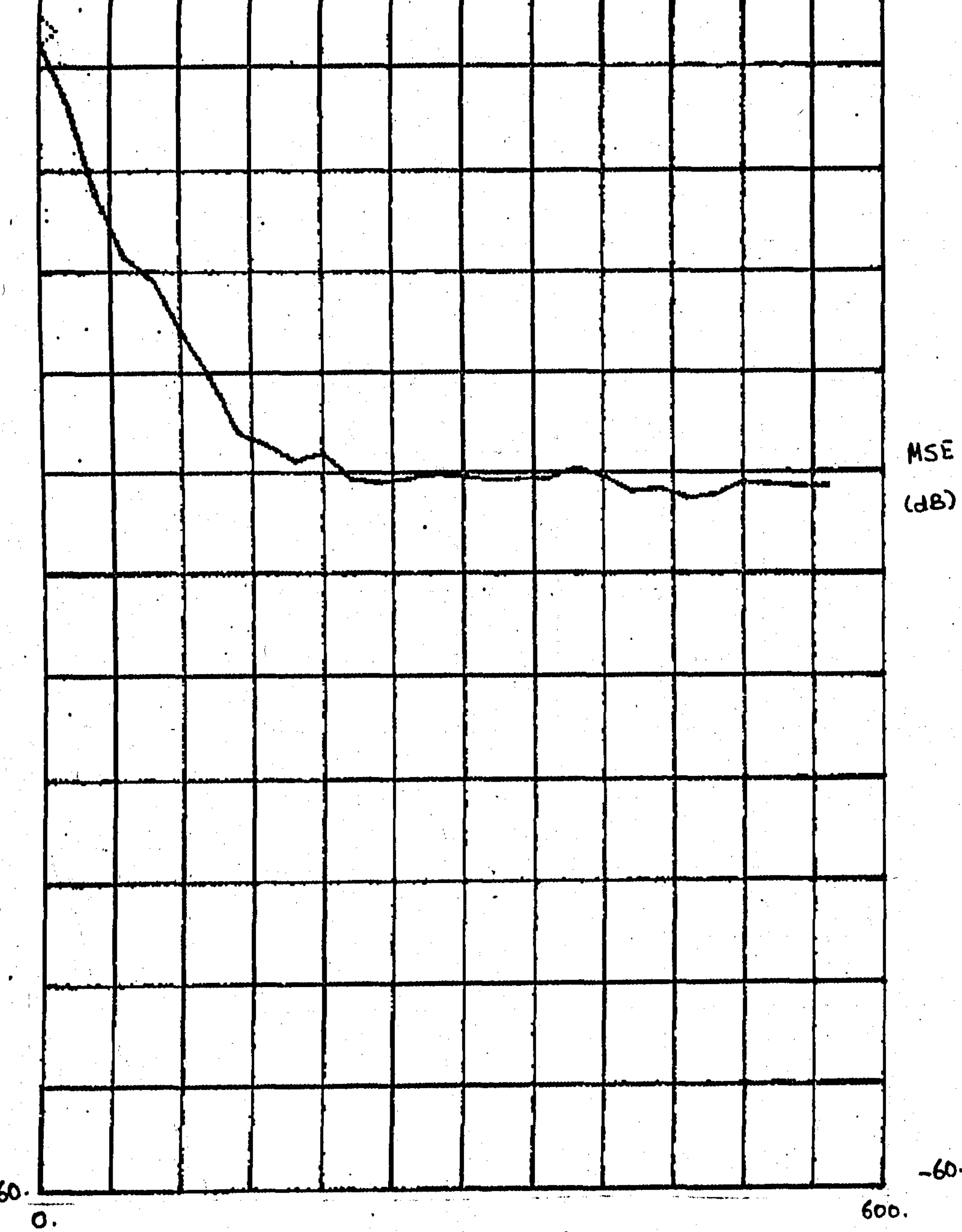

Fig. (VI.F3)

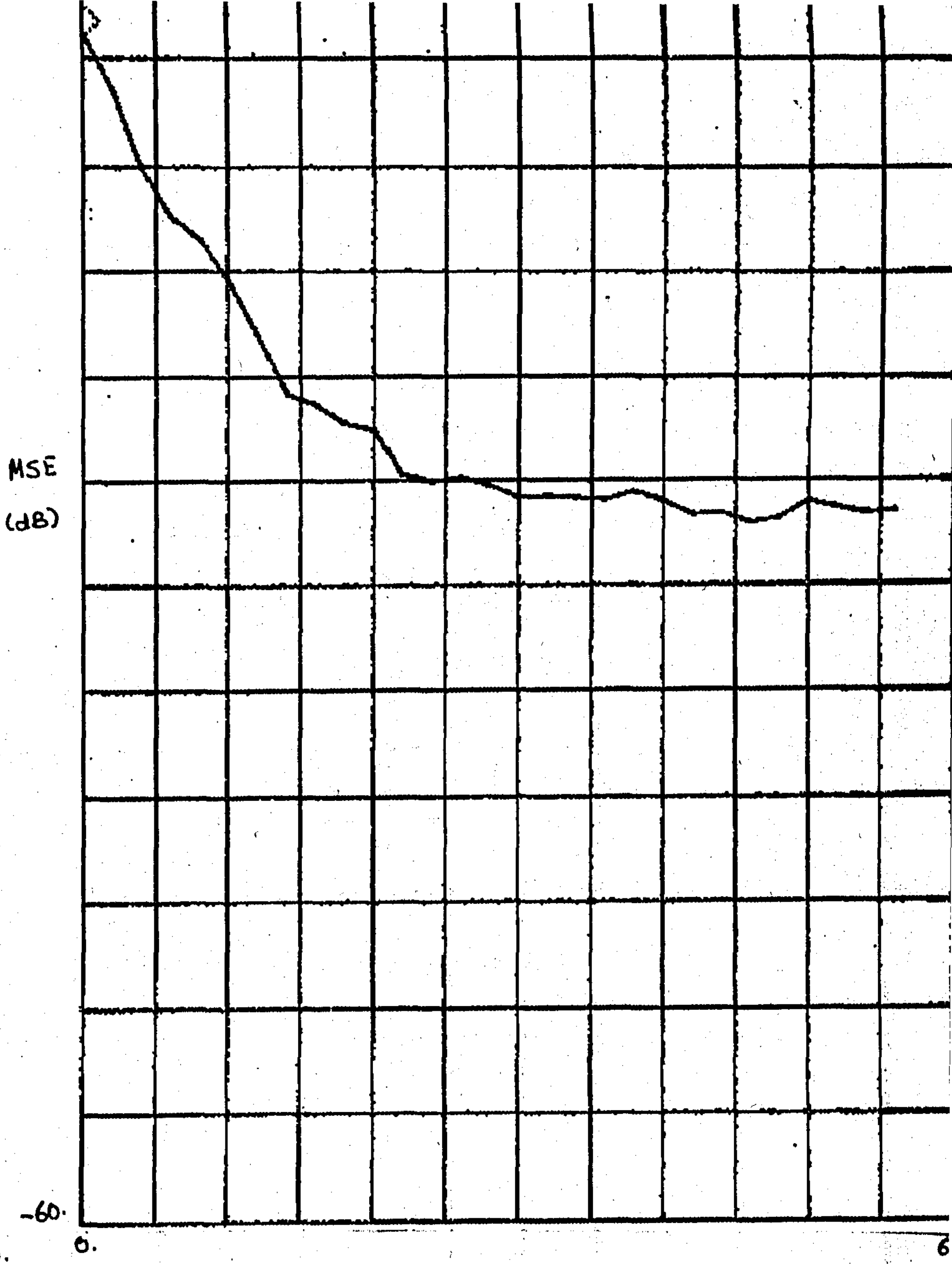

Fig. (VI.F4)

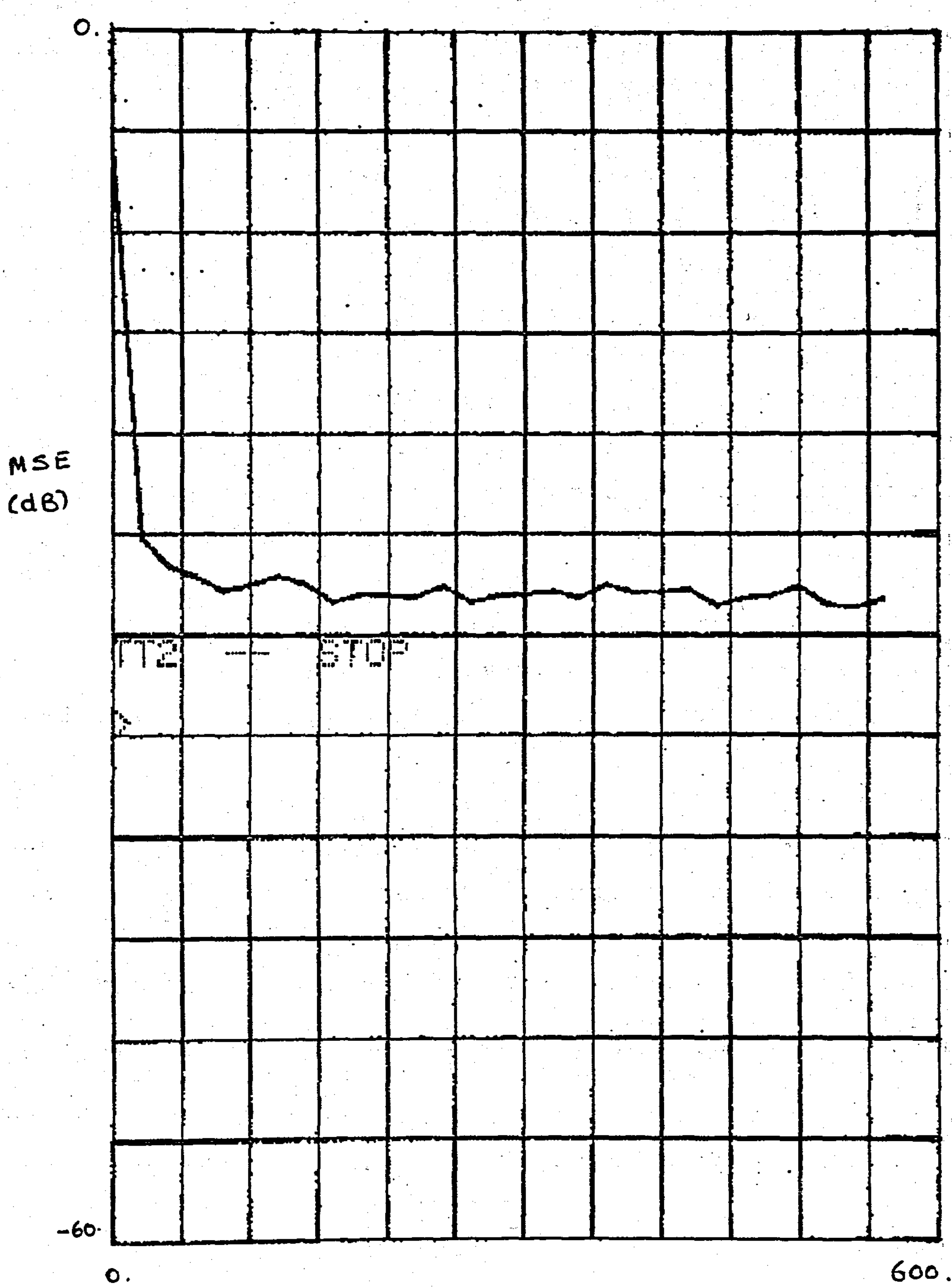

Fig. (VI.F5).

VI.4. CONCLUSION

In this work three LMS algorithms for automatic baseband equalizers are compared through analysis and simulation. The tap coefficients of the TDL automatic equalizers are adapted to minimize the MSE during a traing period and then kept constant in the rest of the data transmission. The length of this training period is an important factor and it is desired to be as short as possible. Stochastic gradient equalization algorithms, although being very simple, may require long training sequences. On the other hand Kalman/Godard algorithm whose start-up is faster is computationally complex. Fortunately, with some modification in computation a relatively simple algorithm which converges very rapidly can be obtained: Fast Kalman Algorithm.

The algorithms have been investigated using various statistical assumptions. However, the simulation results were found to be in accordance with the theoretical implications.

For all type channels the Fast Kalman algorithm gives much better performance than the stochastic gradient algorithm does. Especially for highly distorted channels this superiority increases since the step-size in the stochastic gradient algorithm can be no more related to the number of taps. Thus, the best rate of convergence of the stochastic gradient algorithm can be hardly achieved in highly distorted channels. Even in the case where the best step-size is chosen the Fast Kalman Algorithm is still much faster than the stochastic gradient algorithm. Consequently, as long as the particular application at hand requires fast convergence and the current technology is sufficient for implementation Fast Kalman algorithm can be used in practical systems.

In the simulation it has been observed that all the algorithms are sensitive to truncation errors when the minimum achieveable MSE is too small. However, if the additive noise of variance 0.001 exists, then the algorithms are not affected by the errors arising from digital implementation.

The computer programs in the package have been developed in such a way that they can be separately used. Although the equalization programs have been written for QAM transmission system it is possible to use them in other type of modulation schemes setting the imaginary components of the complex variables to zero.

Finally, I hope that this work will be a stating point for further studies in the field of channel equalization.

APPENDIX I

CORRELATION MATRICES

To derive more explicit formulas for $\underline{b}^*$ and $\underline{A}$ the following assumptions for data and noise sequences are to be made:

(i) Input symbols at different instants are uncorrelated, i.e.,

$$E\{a(n)a^*(m)\} = a^{-2}\delta(n-m) \qquad \forall n,m$$

(ii) Input data and noise sequences are uncorrelated, i.e.,

$$E\{a(n)w^*(m)\} = E\{a(n)\}E\{w^*(m)\} \quad \forall n,m$$

(iii) Noise is zero mean, i.e.,

$$E\{w(n)\} = 0 \qquad \forall n$$

(iv) Noise is white, i.e.,

$$E\{w(n)w^*(m)\} = \delta_w^2\ \delta(n-m) \quad \forall n,m$$

The correlation matrix $\underline{b}^*$ consists of elements

$$b_i^* = E\{a(n)\underline{x}^*(i)\} \qquad i{:}n\ N,\ldots,n,\ldots,n-N$$

Then, from Eqn (II.5)

$$b_i^* = E\{a(n)\left|\sum_m a(m)h(i-m) + w(i)\right|^*\}$$

is obtained. Taking the expectation into the summation gives

$$b_i^* = \sum_m E\{a(n)a^*(m)\}h^*(i-m) + E\{a(m)w^*(i)\}$$

The second term is zero as a result of (ii) and (iii) Applying (i) to the first term gives

$$b_i^* = a^{-2} \sum_m \delta(n-m)h^*(i-m)$$

$$= \overline{a^2}\ h^*(i-n)$$

Then $\underline{b}^*$ becomes

$$\underline{b}^* = a^{-2} \begin{bmatrix} h^*(N) \\ \vdots \\ h^*(0) \\ \vdots \\ h^*(-N) \end{bmatrix}$$

The input correlation matrix $\underline{A}$ has elements of the form

$$A_{ij} = E\{|\sum_m a(m)h(i-m)+w(i)||\sum_k a(k)h(j-k)+w(j)|^*\}$$

By assumptions (ii) and (iii) the cross terms are zero and it follows that

$$A_{ij} = \sum_m \sum_k E\{a(m)a^*(k)\}h(i-m)h^*(j-k)\ E\{w(i)w^*(j)\}$$

$$= a^{-2} \sum_m \sum_k \delta(m-k)h(i-m)h^*(j-k) + \delta_w^2\delta(i-j)$$

$$= a^{-2} \sum_m h(i-m)h^*(j-m) + \delta_w^2\delta(i-j)$$

APPENDIX 2

POSITIVE DEFINITENESS OF THE INPUT CORRELATION MATRIX $\underline{A}$

Let $\underline{u}$ be a (2N+1) complex -valued vector which is different then zero

$$\underline{u} \neq \underline{0}$$

The elements of $\underline{u}$ can be numbered as

$$\underline{u} = \begin{bmatrix} u_{-N} \\ u_o \\ u_N \end{bmatrix}$$

Then,

$$\underline{u}^* \ \underline{A} \ \underline{u} = \underline{u}^* E\{\underline{x}(n)\underline{x}^*(n)\} \ \underline{u}$$

$$= E \ \{||\underline{x}^*(n) \ \underline{u}||^2$$

Let Y(n) be $||\underline{x}^*(n) \ \underline{u}||$, thus the Hermitian form is

$$\underline{u}^* \ \underline{A} \ \underline{u} = E\{Y(n)^2\}$$

$$= \frac{1}{2\pi} \int_{-\pi}^{\pi} P_{YY}(w)\,dw$$

where $P_{YY}(w)$ is the power spectrum of Y(n)

The sequence Y(n) can be expressed as a convolution as

$$Y(n) = \sum_{i=-N}^{N} u(i)x^*(n-i)$$

and this implies that

$$P_{YY}(w) = P_{xx}(w)||u(w)||^2$$

where

$$U(w) = \sum_{i=-N}^{N} u(i)e^{-jwi}$$

Since U(w) is not zero then the Hermitian form vanishes if and only if $P_{xx}(w)$, the power spectrum of the equalizer input signal is identically zero. But

$$P_{xx}(w) = P_{aa}(w)||H(w)||^2$$

where $P_{aa}(w)$ is the power spectrum of the input symbols. Therefore the Hermitian form $\underline{u}^* \underline{A} \underline{u}$ is different than zero for $\underline{u} \neq 0$, or , in other word $\underline{A}$ is positive definite unless the channel impulse response is identically zero.

APPENDIX 3

CONVEXITY OF THE MSE PERFORMANCE FUNCTION OF THE TAP COEFFICIENTS

The MSE performance function of the tap coefficients is

$$\xi^2(\underline{c}) = a^{-2} - 2\,\underline{b}^*\,\underline{c} + \underline{c}^*\underline{A}\,\underline{c}$$

Let $\underline{c}_1$ and $\underline{c}_2$ be two (2N+1) complex-valued vectors and $0<\lambda<1$ be a real scalar. Then,

$$\xi^2|\lambda\underline{c}_1+(1-\lambda)\underline{c}_2| = a^{-2} - 2\,\underline{b}^*|\lambda\underline{c}_1 + (1-\lambda)\underline{c}_2|$$

$$+ |\lambda\underline{c}_1+(1-\lambda)\underline{c}_2|^*\,\underline{A}\,|\lambda\underline{c}_1+(1-\lambda)\underline{c}_2|$$

$$=\lambda|a^{-2} - 2\,\underline{b}^*\underline{c}_1 + \underline{c}_1^*\,\underline{A}\,\underline{c}_1|$$

$$+ (1-\lambda)\,|a^{-2} - 2\,\underline{b}^*\,\underline{c}_2 + \underline{c}_2^*\,\underline{A}\,\underline{c}_2|$$

$$- \lambda(1-\lambda)\,|\underline{c}_1 + \underline{c}_2|^*\,\underline{A}\,|\underline{c}_1 + \underline{c}_2|$$

That is,

$$\xi^2|\lambda\underline{c}_1 + (1-\lambda)\underline{c}_2| = \lambda\xi^2(\underline{c}_1) + (1-\lambda)\xi^2(\underline{c}_2)$$

$$- \lambda(1-\lambda)\,|\underline{c}_1+\underline{c}_2|^*\,\underline{A}\,|\underline{c}_1+\underline{c}_2|$$

Since $\underline{A}$ is positive definite and $0<\lambda<1$

$$\xi^2|\lambda\underline{c}_1+(1-\lambda)\underline{c}_2| \geq \lambda\xi^2(\underline{c}_1)+(1-\lambda)\xi^2(\underline{c}_2)$$

Therefore, $\xi^2(\underline{c})$ is a convex function of the top coefficients.

APPENDIX 4

DERIVATION OF KALMAN/GODARD ALGORITHM (II)

Given the discrete linear stochastic system

$$\underline{x}(k+1) = \Phi(k,k-1)\underline{x}(k) + \underline{W}(k) \quad (1)$$

and the observation system

$$\underline{y}(k) = \underline{M}(k)\underline{x}(k) + \underline{U}(k) \quad (2)$$

Where

Where;

$\underline{x}(k)$: n-dim, state vector

$\Phi(k,k-1)$: nxn state transition matrix

$\underline{W}(k)$: n-dim white Gaussion noise $\sim N(\underline{0},\underline{Q}(k))$

$\underline{y}(k)$: m-dim observation vector

$\underline{M}(k)$: mxn observation matrix

$\underline{U}(k)$: m-dim white Gaussion noise $\sim N(\underline{0},\underline{s}(k))$

and $\underline{x}(0)$, $\{\underline{W}(k)\}$, $\{\underline{U}(k)\}$ are assumed statistically independent.

Then the minimum variance filter for this system consists of the following equations

$$\underline{\hat{x}}(k+1,k) = \Phi(k+1,k)\underline{\hat{x}}(k,k) \quad (3)$$

$$\underline{P}(k+1,k) = \Phi(k+1,k)\underline{P}(k,k)\Phi^{*}(k+1,k)+\underline{Q}(k) \quad (4)$$

between observations, and

$$\underline{\hat{x}}(k,k) = \underline{\hat{x}}(k,k-1) + \underline{K}(k)\ |\underline{y}(k)-\underline{M}(k)\underline{\hat{x}}(k,k-1)| \quad (5)$$

$$\underline{P}(k,k) = \underline{P}(k,k-1) - \underline{K}(k)\ \underline{M}(k)\underline{P}(k,k-1) \quad (6)$$

$$\underline{K}(k) = \underline{P}(k,k-1)\underline{M}^*(k)\ |\underline{M}(k)\underline{P}(k,k-1)\underline{M}^*(k)+\underline{S}(k)|^{-1} \qquad (7)$$

at the observations, where

$$\underline{P}(k) = E\{|\underline{x}(k) - \hat{\underline{x}}(k,k)||\underline{x}(k)-\hat{\underline{x}}(k,k)|^*\} \qquad (8)$$

$$\underline{P}(k+1,k) = E\{|\underline{x}(k+1)-\hat{\underline{x}}(k+1,k)||\underline{x}(k+1)-\hat{x}(k+1,k)|^*|\underline{Y}(k)| \qquad (9)$$

$$\underline{Y}(k) = \{\underline{y}(k),\underline{y}(k-1),\ldots,\underline{y}(0)\}$$

It is also known that as long as the system is uniformly completely controllable and uniformly completely observable and $\underline{P}(0)$ is positive semidefinite the algorithm is stable.

Now, in the equalization problem, assuming a stationary environment, the vectorid quality to be estimated $\underline{c}_{op}$, and,

$$\underline{c}_{op}(k+1) = \underline{c}_{op}(k) \qquad (1')$$

Thus $\Phi(k,k-1) \rightarrow \underline{I}$ and $\underline{W}(k) \rightarrow \underline{0}$. From Eqn. (IV.3) and (IV.10)

$$e_{op}(k) = a(k) - \underline{c}^*_{op}\ \underline{x}(k)$$

or, equivalently

$$a^*(k) = \underline{x}^*(k)\ \underline{c}_{op} + e_{op}(k) \qquad (2')$$

Since $a(k)$ is known during the training period, (2') can be taken as the observation system. Then

$$\underline{M}(k) \rightarrow \underline{x}^*(k)$$

$$\underline{y}(k) \rightarrow a^*(k)$$

$$\underline{v}(k) \rightarrow e_{op}(k)$$

In order to apply the Kalman Filtering to the system described by (1') and (2') $e_{op}(n)$ musk be white Gaussian sequence. Since $e_{op}(n)$ is too small its can be assumed white Gaussian. Under this assumption

$$\underline{S}(k) \rightarrow \xi^2_{min}$$

Therefore, using D(k) insteed of P(k) in the Kalman filtering system to be consistent with the notation of Ch.IV, the filtering equations of the system (1')-(2'), corresponding to (3)-(9) are

$$\underline{c}(k+1,k) = \underline{c}(k,k)$$

$$\underline{c}(k+1) = \underline{c}(k) \qquad (3')$$

$$\underline{D}(k+1,k) = \underline{D}(k,k) \qquad (4')$$

$$\underline{C}(k) = \underline{c}(k-1) + \underline{K}(k)|a^*(k)-\underline{x}^*(k)\underline{c}(k-1)| \qquad (5')$$

$$\underline{D}(k,k) = \underline{D}(k,k-1) - \underline{K}(k)\underline{x}^*(k)\underline{D}(k,k-1) \qquad (6')$$

$$\underline{K}(k) = \underline{D}(k,k-1)\underline{x}(k)|\underline{x}^*(k)\underline{D}(k,k-1)\underline{x}(k) + \xi^2_{mn}|^{-1} \qquad (7')$$

which are the same as those in Ch.IV except that ξ^2min takes place instead of unity in the second one.

APPENDIX 5

USER'S MANUAL

This simulation package has been developed in Electronics Research Department of Marmara Research Institute. One copy is also preserved in the Computer Center of Boğaziçi University. The programs are only compatible to PDP11/45 System(8ØK) and the graphics display processor Tektronix 4Ø1Ø-1 connected to it. However, they can be adapted to any computer system by changing a few statements and the graphing programs in the package.

1.Program: POLYF.FTN

Simulates a telephone channel with specified degree of impairments of seven type.

Task: Seven type of linear distortion are defined. The attenuation and group delay polynomials related to these type of impairments are stored in data file POLY.DAT. the program reads these coefficents from the data file and the cor responding degree of impairments are supplied by the user. By multiplying the coefficients with the specified degrees forms the attenuation and group delay characteristics of the channel and stores them in data files POLY1.DAT and POLY2.DAT. Respectively.

Data Files Used: POLY.DAT.

Data Files Opened: POLY1.DAT., POLY2.DAT.

2. Program: DEL.FTN

Simulates a QAM transmission system. The user must supply the following quantities:

N= The number of decimation points in FFT

Z= The number of signalling elements spanned.

S= The number of samples required per signalling element.

Q2= Carrier frequency.

Q4= Nyquist frequency of the shaping filter (Half the signalling rate)

Q5= Roll-of factor.

Task: Reads the attenuation and group delay coefficients from files POLY1.DAT. and POLY2.DAT. generates the complex frequency response of the combined transmit ting filter (Raised-cosine filter with the specified roll-of factor) and the channel and takes the inverse Fourier transform of the frequency response seguence using a FFT technique. The impulse response samples for the quadrature and inphase components are expressed in complex notation as explained in chapter 1. These impulse response samples are then recorded into the data file FFT1.DAT. DEL.FTN also has routines to display the frequency response and the impulse response characteristics of the transmission system.

Data Files Used: POLY1.DAT,POLY2.DAT.

Data Files Opened: FFT1.DAT.

Related Subroutines: GRID.FTN.

3. Program: OPT.FTN.

Finds the optimum tap coefficients and the minimum MSE for a TDL equalizer with given number of taps.

Task: Reads the impulse response samples of the transmission system from the data file FFT1.DAT. number of taps and the variance of the white Gaussian noise must be supplied by the user. Then, the program calculates the input correlation matrix A correlation vector B. Takes the inverse of the input correlation matrix and finds the optimum tap coefficient and the minimum MSE of the equalizer.

Data Files Used: FFT1.DAT

Related Subroutine: MINV.FTN.

4. Program: HELP.FTN.

Calculates the minimum and maximum eigenvalues and the trace of the input correlation matrix.

Task: Reads the impulse response samples of the transmission system from the data file FFT1.DAT. number of taps of the TDL equalizer and the variance of the white Gaussian noise must be supplied by the user.

Data Files Used: FFT1.DAT.

Related Subroutines: EIGEN.FTN

5. Program: DSTOC.FTN.

Realizes and equalizer algorithm based on the stochastic gradient method.

Task: Reads the impulse response samples of the transmission system from the data file FFT1.DAT. Generates a pseudo random binary sequence. Encodes the binary seque nce for four point QAM to form the training analog signals. Convolves them with the impulse response sequence. As each input sequence is generated it is used to update the tap coefficients.

User must specify the number of taps. The variance of the white Gaussian noise and the step-size.

In desired, the tap coefficient can be stored as quantized.

Program also displays the MSE during the adaptation process.

Data files used: FFT1.DAT.

Related Subroutines: GRID.FTN.
GANOS.FTN.

6. Program: DKALMAN.FTN.

Realizes an equalization algorithm based on the Kalman/Godard method.

Task: The same as DSTOC.FTN. does except that with a different method.

User must specify the number of taps. The variance of the white gaussian noise and the initial value for the error covariance matrix.

Data Files Used: FFT1.DAT.

Related Subroutines: GRID.FTN.

GANOS.FTN.

8. Subroutine: GRID.FTN.

Scales the displaying area.

9. Subroutine: GANOS.FTN.

Generates Gaussian noise of desired length with specified mean and variance.

10. Subroutine: MINV.FTN (From scientific subroutines package of PDP-11 System).

Calculates the inverse of a real matrix of given dimension.

11. Subroutine: EIGEN.FTN. (From scientific subroutines package of PDP-11 System).

Calculates the eigenvalues of a real symmetric matrix of given dimension.

```
      DIMENSION A(2,10),B(2,10),C(2,10),D(2,10),E(2,10),G(6,10),H(4,10)
      DIMENSION P(11),RE(11),XI(11)
      DOUBLE PRECISION A,B,C,D,E,G,H,RE,XI
      INTEGER P
      CALL ASSIGN(3,'DK1:POLY.DAT')
      CALL ASSIGN(1,'DK1:POLY1.DAT')
      CALL ASSIGN(2,'DK1:POLY2.DAT')
      WRITE(6,100)
  100 FORMAT(10X,'READ IN THE DEGREE OF EACH OF THE IMPAIRMENTS GIVEN',/
     D,10X,'BELOW:A,B,C,D,E.',/)
      READ(6,110)(P(J),J=1,5)
  110 FORMAT(5I)
      WRITE(6,120)
  120 FORMAT(10X,'ENTER 1 FOR G1,OTHERWISE 0.',/)
      READ(6,130) P(6)
  130 FORMAT(I)
      WRITE(6,140)
  140 FORMAT(10X,'ENTER 1 FOR G2,OTHERWISE 0.',/)
      READ(6,130) P(7)
      WRITE(6,150)
  150 FORMAT(10X,'ENTER 1 FOR G3,OTHERWISE 0.',/)
      READ(6,130)P(8)
      WRITE(6,160)
  160 FORMAT(10X,'ENTER 1 FOR H1,OTHERWISE 0.',/)
      READ(6,130) P(9)
      WRITE(6,170)
  170 FORMAT(10X,'ENTER 1 FOR H2,OTHERWISE 0.',/)
      READ(6,130) P(10)
      WRITE(5,172) (P(I),I=1,10)
  172 FORMAT(/,6X,'A  B  C  D  E G1 G2 G3 H1 H2',/,5X,I2,1X,I2,1X,I2,1X,
     DI2,1X,I2,1X,I2,1X,I2,1X,I2,1X,I2,1X,I2,//)
      READ(3,180)((A(I,J),J=1,10),I=1,2)
      READ(3,180)((B(I,J),J=1,10),I=1,2)
      READ(3,180)((C(I,J),J=1,10),I=1,2)
      READ(3,180)((D(I,J),J=1,10),I=1,2)
      READ(3,180)((E(I,J),J=1,10),I=1,2)
      READ(3,180)((G(I,J),J=1,10),I=1,6)
      READ(3,180)((H(I,J),J=1,10),I=1,4)
  180 FORMAT(5D)
C MULTIPLY THE COEFFICIENTS BY THE RESPECTIVE DEGREES OF IMPAIRMENT.
      DO 30 J=1,10
      DO 10 K=1,2
      A(K,J)=P(1)*A(K,J)
      B(K,J)=P(2)*B(K,J)
      C(K,J)=P(3)*C(K,J)
      D(K,J)=P(4)*D(K,J)
      E(K,J)=P(5)*E(K,J)
      G(K,J)=P(6)*G(K,J)
   10 H(K,J)=P(9)*H(K,J)
      DO 20 K=3,4
      G(K,J)=P(7)*G(K,J)
   20 H(K,J)=P(10)*H(K,J)
      G(5,J)=P(8)*G(5,J)
   30 G(6,J)=P(8)*G(6,J)
      DO 40 J=1,10
      RE(J)=A(1,J)+B(1,J)+C(1,J)+D(1,J)+E(1,J)+G(1,J)+G(3,J)+G(5,J)+H(1,
     DJ)+H(3,J)
      XI(J)=A(2,J)+B(2,J)+C(2,J)+D(2,J)+E(2,J)+G(2,J)+G(4,J)+G(6,J)+H(2,
     DJ)+H(4,J)
   40 CONTINUE
      RE(11)=0.0
      XI(11)=0.0
      WRITE(1,190)(RE(I),I=1,11)
      WRITE(2,190)(XI(I),I=1,11)
  190 FORMAT(D)
      WRITE(5,200)((I,RE(I),XI(I)),I=1,10)
  200 FORMAT(I,2D)
```

DEL.FTN

```
      DOUBLE PRECISION AT,GD,Q2,Q4,Q5,H
      DIMENSION PLOT(320)
      DIMENSION SS(128),W(128),A(128),R(128),X(128),PP(1000)
      DIMENSION QQ(1000),B(128),T(128),H(2,20),AT(11),GD(11)
      CALL ASSIGN(1,'DK1:POLY1.DAT')
      CALL ASSIGN(2,'DK1:POLY2.DAT')
      CALL ASSIGN(3,'DK1:FFT1.DAT')
      PI=3.14159265358979323846264338327950288841972
      WRITE(6,1000)
 1000 FORMAT(10X,'N=NUMBER OF TIME AND FREQUENCY SAMPLES',/,10X,'Z=TOTAL
     D ELEMENT DURATION OF IMPULSE RESPONSE',/,10X,'S=NUMBER OF SAMPLES
     DPER ELEMENT INTERVAL',/,10X,'Q2=CARRIER FREQUENCY',/,10X,'Q4=NYQUI
     DST FREQUENCY (HALF THE MODULATION RATE)',/,10X,'Q5=SPECTRUM SHAPIN
     DG ROLL-OFF FACTOR',/)
      WRITE(6,1100)
 1100 FORMAT(5X,'ATTENTION...N,Z,S=INTEGERS',/,17X,'Q2,Q4,Q5=REALS',/)
      READ(6,1200)N,IZ,IS,Q2,Q4,Q5
 1200 FORMAT(3I,3D)
      WRITE(5,1225)N,IZ,IS,Q2,Q4,Q5
 1225 FORMAT(//,10X,'N=NUMBER OF TIME AND FREQUENCY SAMPLES       =',I4,/
     D,10X,'Z=TOTAL ELEMENT DURATION OF IMPULSE RESPONSE=',I4,/,10X,'S=N
     DUMBER OF SAMPLES PER ELEMENT INTERVAL    =',I4,/,10X,'Q2=CARRIER F
     DREQUENCY                         =',F7.2,/,10X,'Q5=NYQUIST FREQUENCY
     D                          =',F7.2,/,10X,'SPECTRUM SHAPING ROLL-OFF F
     DACTOR             =',F7.2,///)
      Q3=4.*Q4
      Q6=Q4*(1.+Q5)
      Q7=Q4*(1.-Q5)
      C1=Q2+Q4
      C2=Q2-Q4
      Q8=Q5*Q4
      IS=IS/2
      XN=N
      XS=IS
      XZ=IZ
      N1=N
      N2=N*2
      N3=N*3
      N4=N*4
C     READ IN ATTENUATION CONSTANTS (DB).
      READ(1,1250)(AT(I),I=1,11)
 1250 FORMAT(D)
C     READ IN GROUP DELAY COEFFICIENTS (MS).
      READ(2,1250) (GD(I),I=1,11)
      WRITE(6,1252)
 1252 FORMAT(5X,'PRINT-OUT OF ATT. AND GR.DELAY COEFF.1-0',/)
      READ(6,1254)IP
 1254 FORMAT(I)
      IF(IP.EQ.0) GO TO 45
      WRITE(5,1260)
 1260 FORMAT(//,2X,'ATTENUATION(DB) AND GROUP DELAY(MS) COEFFICIENTS',/)
      DO 4 I=1,11
    4 WRITE(5,1270)I,AT(I),GD(I)
 1270 FORMAT(I,2D)
C     SPECIFY FREQUENCY SAMPLES.
   45 Q9=Q3/N
      DO 50 J=1,N
   50 SS(J)=Q9*(J-1)
```

```
C     CALCULATE GROUP DELAY OF CHANNEL.
      DO 6 I=1,N
      A1=GD(1)
      DO 5 J=2,10
    5 A1=A1+GD(J)*SS(I)**(J-1)
      A1=A1+GD(11)*SS(I)**(10)
      NP4=N4+I
      PLOT(NP4)=A1
    6 CONTINUE
C     CALCULATE PHASE RESPONSE OF CHANNEL.
      DUM=-2.*PI/(10.**3.)
      DO 10 JJ=1,10
      J=11-JJ
      XJ=J
   10 GD(J+1)=GD(J)*DUM/XJ
C     CALCULATE AND CORRECT FOR PHASE AT CARRIER FREQUENCY.
      GD(1)=0.
      DO 20 J=1,10
   20 GD(1)=GD(1)-GD(J+1)*(Q2**J)
C     ADJUST PHASE RESPONSE TO OBTAIN NYQUIST TIMING SAMPLES.
      F1=0.
      F2=0.
      DO 30 J=2,11
      F1=F1+GD(J)*(C1**(J-1))
   30 F2=F2+GD(J)*(C2**(J-1))
      P=(F1-F2)/(2.*PI)
      IF(P.EQ.0.) GO TO 40
      PABS=ABS(P)
      P=PABS/P*(PABS-INT(PABS))
   40 GD(1)=GD(1)+P*2.*PI*Q2/(2.*Q4)
      GD(2)=GD(2)-P*2.*PI/(2.*Q4)
C     SPECIFY SHAPING FILTER.
      DO 60 I=1,N
      A(I)=0.
      D1=Q2-Q6
      D2=Q2+Q6
      D3=Q2-Q7
      D4=Q2+Q7
      IF(SS(I).LE.D1) R(I)=0.
      IF(SS(I).GE.D2) R(I)=0.
      IF((SS(I).GE.D3).AND.(SS(I).LE.D4)) R(I)=1.
      IF((SS(I).GT.D1).AND.(SS(I).LT.D3)) R(I)=1./2.*(1.+SIN(PI/2.*(SS(I
     D)-C2)/Q8))
      IF((SS(I).GT.D4).AND.(SS(I).LT.D2)) R(I)=1./2.*(1.-SIN(PI/2.*(SS(I
     D)-C2)/Q8))
   60 CONTINUE
C     SPECIFY IMPAIRMENT.
      DO 80 I=1,N
      R1=AT(1)
      A1=GD(1)
      DO 70 J=2,10
      R1=R1+AT(J)*SS(I)**(J-1)
   70 A1=A1+GD(J)*SS(I)**(J-1)
      A1=A1+GD(11)*SS(I)**(10.)
C     CONVERT DB TO GAIN AND CASCADE IMPAIRMENT AND FILTER.
      R(I)=10.**(-R1/20)*R(I)
      NP3=N3+I
      IF(R(I).LE.0.0) GO TO 75
      PLOT(NP3)=20.*ALOG10(R(I))
      GO TO 76
   75 PLOT(NP3)=-50.
   76 CONTINUE
```

```
      A(I)=A(I)+A1
      IF(SS(I).GT.4000.) A(I)=0.
      IF(SS(I).GT.4000.) R(I)=0.
   80 CONTINUE
C     PRINT OUT CHANNEL FREQUENCY RESPONSE.
      WRITE(6,1300)
 1300 FORMAT(10X,'ENTER (1.) TO OBTAIN FREQUENCY RESPONSE,OTHERWISE (0.)
     D.',/)
      READ(6,1400)    W7
 1400 FORMAT(F)
      DO 90 K=1,N
      X(K)=R(K)*SIN(A(K))
   90 W(K)=R(K)*COS(A(K))
      IF(W7.EQ.0.0) GO TO 100
      WRITE(5,101)
  101 FORMAT(//,2X,'FREQUENCY(HZ),MAGNITUDE,PHASE,ATTENUATION(DB),DELAY(
     DMS).',/)
  100 CONTINUE
      DO 102 I=1,N
      NP1=N1+I
      NP2=N2+I
      NP4=N4+I
      NP3=N3+I
      PLOT(I)=SS(I)
      PLOT(NP1)=R(I)
      PLOT(NP2)=A(I)
      IF(W7.EQ.0.) GO TO 102
      WRITE(5,103)PLOT(I),PLOT(NP1),PLOT(NP2),PLOT(NP3),PLOT(NP4)
  102 CONTINUE
  103 FORMAT(5F14.8)
      WRITE(5,104)
      WRITE(5,104)
  104 FORMAT(/,'**************************************************************',/)
      CALL INITT(950)
      IPL=1
   91 GO TO(92,93,94,95,98),IPL
   92 XMIN=0.
      XMAX=4000.
      DX=500.
      YMIN=0.
      YMAX=2.
      DY=0.2
      NP1=N1+1
      NP=N1
      CALL TWINDO(0,475,400,750)
      GO TO 96
   93 YMIN=-5.
      YMAX=+5.
      DY=1.
      NP1=N2+1
      NP=N2
      CALL TWINDO(525,1000,400,750)
      GO TO 96
   94 YMIN=-50.
      YMAX=0.
      DY=5.
      CALL TWINDO(0,475,0,350)
      NP1=N3+1
      NP=N3
      GO TO 96
   95 YMIN=0.
```

```
      YMAX=8.
      DY=1.
      CALL TWINDO(525,1000,0,350)
      NP1=N4+1
      NP=N4
   96 CALL DWINDO(XMIN,XMAX,YMIN,YMAX)
      CALL GRID(XMIN,XMAX,DX,YMIN,YMAX,DY)
      CALL MOVEA(SS(1),PLOT(NP1))
      DO 97 I=2,N
   97 CALL DRAWA(SS(I),PLOT(NP+I))
      IPL=IPL+1
      GO TO 91
   98 CONTINUE
      READ(6,1450) DR
 1450 FORMAT(F)
C     ADD ARBITRARY PHASE SHIFT TO CHANNEL.
      DO 200 IQ=1,IS
      XQ=IQ
      DUM=(XN/2.+1.-(XQ-1.)/XS)*2.*PI/Q3
      DO 110 I=1,N
  110 B(I)=A(I)-DUM*(SS(I)-Q2)
C     CALCULATE REAL AND IMAGINARY PARTS OF FREQUENCY RESPON
      DO 120 J=1,N
      X(J)=R(J)*SIN(B(J))
  120 W(J)=R(J)*COS(B(J))
C     FAST FOURIER TRANSFORM.
      IG=ALOG10(XN)/ALOG10(2.)+0.5
      P=2.*PI/XN
      DO 160 LL=1,IG
      L=LL-1
      IG1=2.**(IG-L-1)
      M=0
      L2=2**L
      DO 150 I=1,L2
      D1=M/IG1
      D2=0.
      DO 130 K=1,IG
      D3=INT(D1/2.)
      D2=2.*(D2-D3)+D1
  130 D1=D3
      D4=COS(P*D2)
      D5=SIN(P*D2)
      DO 140 J=1,IG1
      D6=W(M+IG1+1)*D4-X(M+IG1+1)*D5
      D7=W(M+IG1+1)*D5+X(M+IG1+1)*D4
      W(M+IG1+1)=W(M+1)-D6
      X(M+IG1+1)=X(M+1)-D7
      X(M+1)=X(M+1)+D7
      W(M+1)=W(M+1)+D6
  140 M=M+1
  150 M=M+IG1
  160 CONTINUE
      DO 180 IN=1,N
      I=IN-1
      D1=I
      D2=0.
      DO 170 K=1,IG
      D3=INT(D1/2.)
      D2=2.*(D2-D3)+D1
  170 D1=D3
      ID2=D2
      IF(ID2.GE.I) GO TO 180
```

```
      D3=W(I+1)
      W(I+1)=W(ID2+1)
      W(ID2+1)=D3
      D3=X(I+1)
      X(I+1)=X(ID2+1)
      X(ID2+1)=D3
  180 CONTINUE
C     CALCULATE REAL AND IMAGINARY PARTS OF IMPULSE RESPONSE.
      DO 190 IN=1,N
      I=IN-1
      U=I*2.*PI*Q2/Q3
      T(I+1)=2.*W(I+1)*COS(U)+2.*X(I+1)*SIN(U)
  190 B(I+1)=2.*W(I+1)*SIN(U)-2.*X(I+1)*COS(U)
C     INTERLEAVE IMPULSE RESPONSE SAMPLES.
      JS=IS*N-IS+IQ
      DO 200 I=IQ,JS,IS
      IK=(I-IQ+IS)/IS
      PP(I)=T(IK)
      QQ(I)=B(IK)
  200 CONTINUE
      I1=XS/2.*XN+XS-XZ*XS
      I2=XS/2.*XN+XS+XZ*XS
      CALL INITT(950)
      XMIN=-7.
      XMAX=+7.
      DX=1.
      YMIN=-1.
      YMAX=+1.
      DY=0.2
      CALL TWINDO(150,1000,50,700)
      CALL DWINDO(XMIN,XMAX,YMIN,YMAX)
      CALL GRID(XMIN,XMAX,DX,YMIN,YMAX,DY)
      IR=0
      X2=IS*2
      ID=XS/2.*XN+XS
      DO 210 I=I1,I2
      X1=I-ID
      XK1=X1/X2
      XL1=PP(I+1)/PP(ID+1)
      IF(IR.LT.1) CALL MOVEA(XK1,XL1)
      IR=2
      CALL DRAWA(XK1,XL1)
  210 CONTINUE
      IR=0
      DO 220 I=I1,I2
      X1=I-ID
      XK1=X1/X2
      XM1=QQ(I+1)/PP(ID+1)
      IF(IR.LT.1) CALL MOVEA(XK1,XM1)
      IR=2
      CALL POINTA(XK1,XM1)
  220 CONTINUE
      READ(6,1450) DR
      WRITE(6,1500)
 1500 FORMAT(10X,'DO YOU WANT A PRINT OUT OF IMPULSE RESPONSE SAMPLES.
     D/,10X,'ENTER (1.) OR (0.).',/)
      READ(6,1400) WB
C     PRINT OUT OF IMPULSE RESPONSE.
  230 DO 250 I=I1,I2
      X1=I-ID
      XK1=X1/X2
      XL1=PP(I+1)/PP(ID+1)
      XM1=QQ(I+1)/PP(ID+1)
```

```
      IF(W8.EQ.0.) GO TO 240
      WRITE(5,1700) XK1,XL1,XM1
  240 IF(INT(XK1).NE.XK1) GO TO 250
      K1P8=XK1+8.
      H(1,K1P8)=XL1
      H(2,K1P8)=XM1
  250 CONTINUE
      DO 260 J=16,20
      H(1,J)=0.
  260 H(2,J)=0.
      WRITE(5,1600)
 1600 FORMAT(10X,'THE COMPLEX IMPULSE RESPONSE SAMPLES ARE',/,12X,'T',24
     DX,'REAL',13X,'IMAGINARY',/)
 1700 FORMAT(5X,3F14.8)
      WRITE(3,1800) ((H(I,J),J=1,20),I=1,2)
 1800 FORMAT(5D)
      DO 270 J=1,20
  270 WRITE(5,1900)J,H(1,J),H(2,J)
 1900 FORMAT(10X,I5,2D)
      END FILE 1
      END FILE 2
      END FILE 3
        CALL FINITT(0,767)
      CALL EXIT
      END
```

```
C       ******************************************************
C       *                                                    *
C       *        CALCULATES THE OPTIMUM TAP COEFFICIENTS     *
C       *        AND THE MINIMUM MEAN-SQUARE-ERROR FOR       *
C       *        AN ADAPTIVE TDL EQUALIZER.                  *
C       *                                                    *
C       *        MODULATION:QAM                              *
C       *                                                    *
C       *        SUBROUTINES USED:MINV                       *
C       *                                                    *
C       *        NOTE:DIMENSION STATEMENTS IS TO BE CHANGED  *
C       *        ACCORDING TO THE DIMENSION OF THE PROBLEM.  *
C       *                                                    *
C       ******************************************************
C
        IMPLICIT DOUBLE PRECISION (A-H),(O-Z)
        DIMENSION R(15,15),XI(15,15),H(2,15),A(30,30),B(2,15)
        DIMENSION C(2,15),Y(2,30),AMIN1(30),AMIN2(30)
        CALL ASSIGN(1,'DK1:FFT1.DAT',12,IER)
        M=15
        FIR=0.
        READ(1,1)((H(I,J),J=1,20),I=1,2)
1       FORMAT(5D)
        WRITE(5,2)
2       FORMAT(10X,'NOISE VARIANCE')
        READ(5,3)SIGMA
3       FORMAT(F)
        WRITE(6,4)SIGMA
4       FORMAT(10X,'NOISE VARIANCE=',F)
C
```

```
      DO 5 J=1,M
      I1=1+MAX0(I,J)
      I2=M+MIN0(I,J)
      DO 5 N=I1,I2
      R(I,J)=R(I,J)+H(1,N-I)*H(1,N-J)+H(2,N-I)*H(2,N-J)
      XI(I,J)=XI(I,J)+H(1,N-I)*H(2,N-J)-H(2,N-I)*H(1,N-J)
5     CONTINUE
      DO 6 I=1,M
6     R(I,I)=R(I,I)+SIGMA
      DO 7 I=1,20
7     FIR=FIR+R(I,I)
      WRITE(6,18)
      WRITE(6,8)FIR
8     FORMAT(5X,'TRACE=',F,/)
C
C     INVERT MATRIX R
C
      DO 9 I=1,M
      I2=I*2
      I2M1=I*2-1
      DO 9 J=1,M
      J2=J*2
      J2M1=J*2-1
      A(I2M1,J2M1)=R(I,J)
      A(I2M1,J2)=XI(I,J)
      A(I2,J2M1)=-XI(I,J)
9     A(I2,J2)=R(I,J)
      CALL MINV(A,I2,DUM,AMIN1,AMIN2)
      DO 10 I=1,M
      I2M1=I*2-1
      DO 10 J=1,M
      J2M1=J*2-1
      J2=J*2
      R(I,J)=A(I2M1,J2M1)
10    XI(I,J)=A(I2M1,J2)
C
C     CALCULATE MATRIX B.
C
      DO 11 I=1,M
      MP1MI=M+1-I
      B(1,I)=H(1,MP1MI)
11    B(2,I)=-H(2,MP1MI)
C
C     CALCULATE  OPTIMUM TAP COEFFICIENTS.
C
      WRITE(6,18)
      WRITE(6,12)
12    FORMAT(5X,'REAL TAP COEFFICIENTS,IMAGINARY TAP COEFFICIENTS.',/
      DO 14 I=1,M
      DO 13 J=1,M
      C(1,I)=R(I,J)*B(1,J)-XI(I,J)*B(2,J)+C(1,I)
      C(2,I)=-R(I,J)*B(2,J)-XI(I,J)*B(1,J)+C(2,I)
13    CONTINUE
      WRITE(6,15) C(1,I),C(2,I)
14    CONTINUE
15    FORMAT(5X,2D)
C
C     CALCULATE MINIMUM MSE.
C
      WRITE(6,18)
      DO 16 I=1,M
16    E1=E1-B(1,I)*C(1,I)+B(2,I)*C(2,I)
      E1=1.+E1
      E2=10.*DLOG10(E1)
      WRITE(6,17) E1,E2
17    FORMAT(5X,'MINIMUM ACHIEVEABLE MSE=',/,D,5X,D,'DB')
18    FORMAT(5X,'**************************************************')
```

```
C       *       MATRIX.                                              *
C       *                                                            *
C       *       FINDS THE RATIO OF THE MAXIMUM EIGENVALUE            *
C       *       TO THE MINIMUM EIGENVALUE.                           *
C       *
C       *       MODULATION:QAM
C       *
C       *       SUBROUTINES USED:EIGEN
C       *
C       *       NOTE:DIMENSION STATEMENTS IS TO BE CHANGED
C       *       ACCORDING TO THE DIMENSION OF THE PROBLEM.
C       *
C       ****************************************************************
C
        IMPLICIT DOUBLE PRECISION  (A-H),(O-Z)
        DIMENSION R(15,15),XI(15,15),H(2,15),A(30,30)
        DIMENSION SOR(30,30),EIG(465)
        CALL ASSIGN(1,'DK1:FFT1.DAT',12,IER)
        M=15
        FIR=0.
        WRITE(5,1)
1       FORMAT(10X,'NOISE VARIANCE=')
        READ(5,2)SIGMA
2       FORMAT(F)
        WRITE(6,3)SIGMA
3       FORMAT(10X,'NOISE VARIANCE=',F)
        READ(1,4)((H(I,J),J=1,20),I=1,2)
4       FORMAT(5D)
C       CALCULATE MATRIX R.
        DO 5 I=1,M
        DO 5 J=1,M
        I1=1+MAX0(I,J)
        I2=M+MIN0(I,J)
        DO 5 N=I1,I2
        R(I,J)=R(I,J)+H(1,N-I)*H(1,N-J)+H(2,N-I)*H(2,N-J)
        XI(I,J)=XI(I,J)+H(1,N-I)*H(2,N-J)-H(2,N-I)*H(1,N-J)
5       CONTINUE
        DO 6 I=1,N
6       R(I,I)=R(I,I)+SIGMA
        DO 7 I=1,M
7       FIR=FIR+R(I,I)
        WRITE(6,17)
        WRITE(6,8)FIR
8       FORMAT(10X,'TRACE=',F,/)
        DO 9 I=1,M
        I2=I*2
        I2M1=I*2-1
        DO 9 J=1,M
        J2=J*2
        J2M1=J*2-1
        A(I2M1,J2M1)=R(I,J)
        A(I2M1,J2)=XI(I,J)
        A(I2,J2M1)=-XI(I,J)
9       A(I2,J2)=R(I,J)
        DO 10 I=1,15
        DO 10 J=1,15
        SOR(I,J)=R(I,J)
        SOR(I,J+15)=XI(I,J)
        SOR(I+15,J)=-XI(I,J)
        SOR(I+15,J+15)=R(I,J)
10      CONTINUE
        NN=2*M
        DO 11 I=1,NN
        DO 11 J=1,I
        K=I*(I-1)/2
        EIG(K+J)=SOR(J,I)
11      CONTINUE
```

```
12      EMAX=DMAX1(EMAX,EIG(I))
        DO 13 I=1,465
        IF(EIG(I).LT.0.0001)EIG(I)=100.
13      CONTINUE
        EMIN=EIG(1)
        DO 14 I=1,465
14      EMIN=DMIN1(EMIN,EIG(I))
        WRITE(6,17)
        WRITE(6,15)EMIN,EMAX
15      FORMAT(10X,'MIN.EIG.=',D,/,10X,'MAX.EIG.=',D)
        RATIO=EMAX/EMIN
        WRITE(6,16)RATIO
16      FORMAT(10X,'RATIO=',D)
17      FORMAT(10X,'*****************')
        END FILE 1
        CALL EXIT
        END


C       ***************************************************************
C       *                                                             *
C       *       STOCHASTIC GRADIENT ALGORITHM                         *
C       *       CALCULATES THE TAP COEFFICIENTS AND THE               *
C       *       MINIMUM MEAN-SQUARE-ERROR OF AN ADAPTIVE EQUALIZER.   *
C       *                                                             *
C       *       PULSE SAMPLES ARE FROM  DATA FILE FFT1.DAT            *
C       *                                                             *
        *       NOISE:WHITE GAUSSIAN WITH ZERO-MEAN                   *
C       *       MODULATION: QAM                                       *
C       *       INPUTS:                                               *
C       *       N:NO OF TAPS                                          *
C       *       H:IMPULSE RESPONSE SAMPLES                            *
C       *       ALPHA:STEP-SIZE                                       *
C       *       SIGMA:NOISE VARIANCE                                  *
C       *                                                             *
C       *       OUTPUTS:                                              *
C       *       C:TAP COEFFICIENTS                                    *
C       *                                                             *
C       ***************************************************************
        DIMENSION H(2,20),X(2,20),C(2,20),J(18),D(2,20),YP(1200),DD(2)
        DIMENSION G(2,4),GUR(600)
        DOUBLE PRECISION H
        CALL ASSIGN(3,'DK1:FFT1.DAT')
        DATA J/1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0/
        DATA G/3,-1,-1,-3,1,3,-3,1/
        WRITE(5,1)
1       FORMAT(10X,'NO OF TAPS,ALPHA=?',/)
        READ(5,2)N,ALPHA
2       FORMAT(I,F)
        WRITE(6,3)N,ALPHA
3       FORMAT(10X,'NO OF TAPS=',I2,/,10X,'ALPHA=',F10.4,/)
        WRITE(5,4)
4       FORMAT(10X,'AVE LIM.',/)
        READ(5,5)LIM
5       FORMAT(I)
        WRITE(5,30)
        READ(5,31)SEED,SIGMA
        WRITE(6,32)SIGMA
        SIGMA=SQRT(SIGMA)
        WRITE(5,6)
6       FORMAT(10X,'DO YOU INCLUDE QUANTIZATION ERRORS?')
        WRITE(5,7)
7       FORMAT(10X,'YES=1  NO=0')
        READ(5,8)IQ
8       FORMAT(I)
```

```
12    CONTINUE
      M=INT(8.+(N-1.)/2.)
C     GAUSSIAN NOISE GENERATION
      GUR(1)=SEED
      CALL GANOS(GUR,600,0.,SIGMA)
      READ(3,13)((H(I,L),L=1,20),I=1,2)
13    FORMAT(5D)
      BNN1=2**N1
C
C     IMPULSE RESPONSE
C
      DO 14 I=1,N
      X(1,I)=0.
      X(2,I)=0.
      C(1,I)=0.
      C(2,I)=0.
      D(1,I)=0.
      D(2,I)=0.
14    CONTINUE
      P=0.
      L=0
      JK=0
      CALL INITT(950)
      YMIN=-60.
      YMAX=0.
      XMIN=0.
      XMAX=600.
      DX=50.
      DY=5.
      CALL TWINDO(0,500,0,750)
      CALL DWINDO(XMIN,XMAX,YMIN,YMAX)
      CALL GRID(XMIN,XMAX,DX,YMIN,YMAX,DY)
C
C     MAIN LOOP
C
      DO 1000 K=1,600
C     PRBS GENERATION
      DO 16 I=1,2
      J(1)=1
      IF((J(6)+J(10)).NE.1)J(1)=0
      DO 15 IT=1,9
      ITT=10-IT
      J(ITT+1)=J(ITT)
15    CONTINUE
16    DD(I)=J(1)
      INX=DD(2)+DD(1)*2+1
      D(1,1)=G(1,INX)/3.
      D(2,1)=G(2,INX)/3.
      X(1,1)=0.
      X(2,1)=0.
C
C     CONVOLUTION
C
      DO 17 I=1,20
      X(1,1)=X(1,1)+H(1,I)*D(1,I)-H(2,I)*D(2,I)
17    X(2,1)=X(2,1)+H(1,I)*D(2,I)+H(2,I)*D(1,I)
C
C     NOISE ADDITION
C.
      X(1,1)=X(1,1)+(GUR(K)/1.41421356)
      X(2,1)=X(2,1)+(GUR(K)/1.41421356)
      IF(K.LT.M)GO TO 22
      Y1=0.
      Y2=0.
C     OUTPUT
      DO 20 I=1,N
      IF(IQ.EQ.0)GO TO 18
```

```
C
      A1=INT(C(1,I)*BNN1+0.5)/BNN1
      A2=INT(C(2,I)*BNN1+0.5)/BNN1
      GO TO 19
18    A1=C(1,I)
      A2=C(2,I)
19    CONTINUE
      Y1=Y1+A1*X(1,I)+A2*X(2,I)
20    Y2=Y2+A1*X(2,I)-A2*X(1,I)
C
C     ERROR
C
      E1=D(1,M)-Y1
      E2=D(2,M)-Y2
      P=P+E1*E1+E2*E2
      E1=E1*ALPHA
      E2=E2*ALPHA
C
C     EQUALIZATION
C
      DO 21 I=1,N
      C(1,I)=C(1,I)+E1*X(1,I)+E2*X(2,I)
21    C(2,I)=C(2,I)+E1*X(2,I)-E2*X(1,I)
C
C     SHIFTING
C
22    DO 23 IS=1,N
      ISS=N+1-IS
      X(1,ISS+1)=X(1,ISS)
      X(2,ISS+1)=X(2,ISS)
23    CONTINUE
      DO 24 IS=1,N
      ISS=N+1-IS
24    D(1,ISS+1)=D(1,ISS)
      D(2,ISS+1)=D(2,ISS)
      IF(K.LT.M)GO TO 25
      IF(L.LT.LIM)GO TO 26
      JK=JK+1
      XL=L
      YP(JK)=10.*ALOG10(P/XL)
      P1=YP(JK)
25    P=0.
      L=0
26    L=L+1
1000  CONTINUE
      CALL MOVEA(0.,YP(1))
      DO 27 I=1,JK
      XP=(I-1)*LIM
      CALL DRAWA(XP,YP(I))
27    CONTINUE
      WRITE(6,28)
28    FORMAT(40X,'TAP COEFFICIENTS',/)
      WRITE(6,29)(C(1,I),C(2,I),I=1,N)
29    FORMAT(40X,F12.8,10X,F12.8,/)
30    FORMAT(10X,'SEED,NOISE VARIANCE')
31    FORMAT(2F)
32    FORMAT(10X,'NOISE VARIANCE=',F)
      CALL FINITT(0.767)
      END
```

```
C     ****************************************************
C     *      KALMAN/GODARD ALGORITHM                      *
C     *                                                   *
C     *      CALCULATES THE TAP COEFFICIENTS              *
C     *      AND THE MINIMUM MEAN-SQUARE-                 *
C     *      ERROR FOR AN ADAPTIVE EQUALIZER.             *
C     *                                                   *
C     *      NOISE:WHITE GAUSSIAN WITH ZERO-MEAN          *
C     *                                                   *
C     *      MODULATION:QAM                               *
C     *                                                   *
C     *      IMPULSE RESPONSE SAMPLES ARE                 *
C     *      FROM DATA FILE FFT1.DAT.                     *
C     *                                                   *
C     *      INPUTS:                                      *
C     *      H:IMPULSE RESPONSE SAMPLES                   *
C     *      N:NO. OF TAPS                                *
C     *      SIGMA:NOISE VARIANCE                         *
C     *      ALPHA:ESTIMATED MSE                          *
C     *                                                   *
C     *      OUTPUTS:                                     *
C     *      C:TAP COEFFICIENTS                           *
C     *                                                   *
C     ****************************************************
      DIMENSION H(2,20),X(2,20),C(2,20),J(18),D(2,20),YP(1200),DD(2)
      DIMENSION G(2,4),GUR(600)
      DIMENSION GAIN(2,20),T(2,20),VAR1(20,20),VAR2(20,20)
      DOUBLE PRECISION H
      CALL ASSIGN(3,'DK1:FFT1.DAT')
      DATA J/1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0/
      DATA G/3,-1,-1,-3,1,3,-3,1/
      WRITE(5,1)
1     FORMAT(10X,'NO OF TAPS,ALPHA=?',/)
      READ(5,2)N,ALPHA
2     FORMAT(I,F)
      WRITE(6,3)N,ALPHA
3     FORMAT(10X,'NO OF TAPS=',I2,/,10X,'ALPHA=',F10.4,/)
      WRITE(5,4)
4     FORMAT(10X,'AVE LIM.',/)
      READ(5,5)LIM
5     FORMAT(I)
      WRITE(5,37)
      READ(5,38)SEED,SIGMA
      WRITE(6,39)SIGMA
      SIGMA=SQRT(SIGMA)
      WRITE(5,6)
6     FORMAT(10X,'DO YOU INCLUDE QUANTIZATION ERRORS?')
      WRITE(5,7)
7     FORMAT(10X,'YES=1  NO=0')
      READ(5,8)IQ
8     FORMAT(I)
      IF(IQ.EQ.0)GO TO 12
      WRITE(5,9)
9     FORMAT(10X,'NO OF TAP BITS?')
      READ(5,10)N1
10    FORMAT(I)
      WRITE(6,11)N1
11    FORMAT(10X,'NO.OF TAP BITS=',I2)
12    CONTINUE
      M=INT(3.+(N-1.)/2.)
C
C     GAUSSIAN NOISE GENERATION
C
      GUR(1)=SEED
      CALL GANOS(GUR,600,0.,SIGMA)
      READ(3,13)((H(I,L),L=1,20),I=1,2)
13    FORMAT(5D)
      RNN1=2**N1
```

```
      X(2,I)=0.
      C(1,I)=0.
      C(2,I)=0.
      D(1,I)=0.
      D(2,I)=0.
14    CONTINUE
      DO 15 I=1,N
      T(1,I)=0.
      T(2,I)=0.
      GAIN(1,I)=0.
      GAIN(2,I)=0.
15    CONTINUE
      DO 16 I=1,N
      DO 16 JJ=1,N
      VAR1(I,JJ)=0.
16    VAR2(I,JJ)=0.
      DO 17 I=1,N
17    VAR1(I,I)=1./ALPHA
      P=0.
      L=0
      JK=0
      CALL INITT(960)
      YMIN=-60.
      YMAX=0.
      XMIN=0.
      XMAX=600.
      DX=50.
      DY=5.
      CALL TWINDO(0,500,0,750)
      CALL DWINDO(XMIN,XMAX,YMIN,YMAX)
      CALL GRID(XMIN,XMAX,DX,YMIN,YMAX,DY)
C
C     MAIN LOOP
C
      DO 1000 K=1,600
C
C     PRBS GENERATION
C
      DO 19 I=1,2
      J(1)=1
      IF((J(6)+J(10)).NE.1)J(1)=0
      DO 18 IT=1,9
      ITT=10-IT
      J(ITT+1)=J(ITT)
18    CONTINUE
19    DD(I)=J(1)
      INX=DD(2)+DD(1)*2+1
      D(1,1)=G(1,INX)/3.
      D(2,1)=G(2,INX)/3.
      X(1,1)=0.
      X(2,1)=0.
C
C     CONVOLUTION
C
      DO 20 I=1,20
      X(1,1)=X(1,1)+H(1,I)*D(1,I)-H(2,I)*D(2,I)
20    X(2,1)=X(2,1)+H(1,I)*D(2,I)+H(2,I)*D(1,I)
C
C     NOISE ADDITION
C
      X(1,1)=X(1,1)+(GUR(K)/1.41421356)
      X(2,1)=X(2,1)+(GUR(K)/1.41421356)
      IF(K.LT.M)GO TO 29
      Y1=0.
      Y2=0.
C
C     OUTPUT
C
```

```
      GO TO 22
21    A1=C(1,I)
      A2=C(2,I)
22    CONTINUE
      Y1=Y1+A1*X(1,I)+A2*X(2,I)
23    Y2=Y2+A1*X(2,I)-A2*X(1,I)
C
C     ERROR
C
      E1=D(1,M)-Y1
      E2=D(2,M)-Y2
      P=P+E1*E1+E2*E2
C
C     EQUALIZATION
C
      DO 24 I=1,N
      T(1,I)=0.
      T(2,I)=0.
      DO 24 JJ=1,N
      T(1,I)=T(1,I)+VAR1(I,JJ)*X(1,JJ)-VAR2(I,JJ)*X(2,JJ)
24    T(2,I)=T(2,I)+VAR1(I,JJ)*X(2,JJ)+VAR2(I,JJ)*X(1,JJ)
      S1=0.
      S2=0.
      DO 25 I=1,N
      S1=S1+X(1,I)*T(1,I)+X(2,I)*T(2,I)
25    S2=S2+X(1,I)*T(2,I)-X(2,I)*T(1,I)
      S1=1.+S1
      BOM=S1*S1+S2*S2
      S1=S1/BOM
      S2=-S2/BOM
      DO 26 I=1,N
      GAIN(1,I)=T(1,I)*S1-T(2,I)*S2
26    GAIN(2,I)=T(1,I)*S2+T(2,I)*S1
      DO 27 I=1,N
      DO 27 JJ=1,N
      SS1=GAIN(1,I)*T(1,JJ)+GAIN(2,I)*T(2,JJ)
      SS2=GAIN(2,I)*T(1,JJ)-GAIN(1,I)*T(2,JJ)
      VAR1(I,JJ)=VAR1(I,JJ)-SS1
27    VAR2(I,JJ)=VAR2(I,JJ)-SS2
      DO 28 I=1,N
      C(1,I)=C(1,I)+GAIN(1,I)*E1+GAIN(2,I)*E2
28    C(2,I)=C(2,I)+GAIN(2,I)*E1-GAIN(1,I)*E2
C
C     SHIFTING
C
29    DO 30 IS=1,N
      ISS=N+1-IS
      X(1,ISS+1)=X(1,ISS)
      X(2,ISS+1)=X(2,ISS)
30    CONTINUE
      DO 31 IS=1,N
      ISS=N+1-IS
31    D(1,ISS+1)=D(1,ISS)
      D(2,ISS+1)=D(2,ISS)
      IF(K.LT.M)GO TO 32
      IF(L.LT.LIM)GO TO 33
      JK=JK+1
      XL=L
      YP(JK)=10.*ALOG10(P/XL)
      P1=YP(JK)
32    P=0.
      L=0
33    L=L+1
1000  CONTINUE
      CALL MOVEA(0.,YP(1))
      DO 34 I=1,JK
      XP=(I-1)*LIM
```

```
37      FORMAT(10X,'SEED,NOISE VARIANCE')
38      FORMAT(2F)
39      FORMAT(10X,'NOISE VARIANCE=',F)
        CALL FINITT(0.767)
        END

C       ****************************************************
C       *       FAST KALMAN ALGORITHM                      *
C       *       CALCULATES THE TAP COEFFICIENTS AND        *
C       *       MEAN-SQUARE-ERROR FOR AN ADAPTIVE          *
C       *       EQUALIZER.                                 *
C       *                                                  *
C       *       NOISE:WHITE GAUSSIAN WITH ZERO MEAN        *
C       *                                                  *
C       *       IMPULSE RESPONSE SAMPLES ARE FROM DATA     *
C       *       FILE FFT1.DAT                              *
C       *                                                  *
C       *       MODULATION QAM                             *
C       *                                                  *
C       *       INPUTS:                                    *
C       *       H:IMPULSE RESPONSE SAMPLES                 *
C       *       N:NO.OF TAPS                               *
C       *       SIGMA:NOISE VARIANCE                       *
C       *       ALPHA:ESTIMATED MSE                        *
C       *                                                  *
C       *       OUTPUTS:                                   *
C       *       C:TAP COEFFICIENTS                         *
C       *                                                  *
C       ****************************************************
        DIMENSION H(2,20),X(2,20),C(2,20),J(18),D(2,20),YP(1200),DD(2)
        DIMENSION G(2,4),GUR(600)
        DIMENSION GAIN(2,20),F(2,20),B(2,20),GEX(2,20),XOLD(2,20)
        DOUBLE PRECISION H
        CALL ASSIGN(3,'DK1:FFT1.DAT')
        DATA J/1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0/
        DATA G/3,-1,-1,-3,1,3,-3,1/
        WRITE(5,1)
1       FORMAT(10X,'NO OF TAPS,ALPHA=?',/)
        READ(5,2)N,ALPHA
2       FORMAT(I,F)
        WRITE(6,3)N,ALPHA
3       FORMAT(10X,'NO OF TAPS=',I2,/,10X,'ALPHA=',F10.4,/)
        WRITE(5,4)
4       FORMAT(10X,'AVE LIM.',/)
        READ(5,5)LIM
5       FORMAT(I)
        WRITE(5,39)
        READ(5,40)SEED,SIGMA
        WRITE(6,41)SIGMA
        SIGMA=SQRT(SIGMA)
        WRITE(5,6)
6       FORMAT(10X,'DO YOU INCLUDE QUANTIZATION ERRORS?')
        WRITE(5,7)
7       FORMAT(10X,'YES=1  NO=0')
        READ(5,8)IQ
8       FORMAT(I)
        IF(IQ.EQ.0)GO TO 12
        WRITE(5,9)
9       FORMAT(10X,'NO OF TAP BITS?')
        READ(5,10)N1
10      FORMAT(I)
        WRITE(6,11)N1
11      FORMAT(10X,'NO.OF TAP BITS=',I2)
12      CONTINUE
        M=INT(8.+(N-1.)/2.)
C
```

```
      BNN1=2**N1
      DO 14 I=1,N
      X(1,I)=0.
      X(2,I)=0.
      C(1,I)=0.
      C(2,I)=0.
      D(1,I)=0.
      D(2,I)=0.
 14   CONTINUE
      DO 15 I=1,N
      F(1,I)=0.
      F(2,I)=0.
      GAIN(1,I)=0.
      GAIN(2,I)=0.
      B(1,I)=0.
      B(2,I)=0.
      GEX(1,I)=0.
      GEX(2,I)=0.
      XOLD(1,I)=0.
 15   XOLD(2,I)=0.
      RES1=ALPHA
      RES2=0.
      P=0.
      L=0
      JK=0
      CALL INITT(950)
      YMIN=-60.
      YMAX=0.
      XMIN=0.
      XMAX=600.
      DX=50.
      DY=5.
      CALL TWINDO(0,500,0,750)
      CALL DWINDO(XMIN,XMAX,YMIN,YMAX)
      CALL GRID(XMIN,XMAX,DX,YMIN,YMAX,DY)
C
C     MAIN LOOP
C
      DO 1000 K=1,600
C
C     PRBS GENERATION
C
      DO 17 I=1,2
      J(1)=1
      IF((J(6)+J(10)).NE.1)J(1)=0
      DO 16 IT=1,9
      ITT=10-IT
      J(ITT+1)=J(ITT)
 16   CONTINUE
 17   DD(I)=J(1)
      INX=DD(2)+DD(1)*2+1
      D(1,1)=G(1,INX)/3.
      D(2,1)=G(2,INX)/3.
      X(1,1)=0.
      X(2,1)=0.
C
C     CONVOLUTION
C
      DO 18 I=1,20
      X(1,1)=X(1,1)+H(1,I)*D(1,I)-H(2,I)*D(2,I)
 18   X(2,1)=X(2,1)+H(1,I)*D(2,I)+H(2,I)*D(1,I)
C
C     NOISE ADDITION
C
      X(1,1)=X(1,1)+(GUR(K)/1.41421356)
      X(2,1)=X(2,1)+(GUR(K)/1.41421356)
      FOR1=X(1,1)
```

```
      DO 20 I=1,N
      F(1,I)=F(1,I)+GAIN(1,I)*FOR1+GAIN(2,I)*FOR2
20    F(2,I)=F(2,I)+GAIN(2,I)*FOR1-GAIN(1,I)*FOR2
      PFOR1=1.
      PFOR2=0.
      DO 21 I=1,N
      PFOR1=PFOR1-GAIN(1,I)*XOLD(1,I)-GAIN(2,I)*XOLD(2,I)
21    PFOR2=PFOR2-GAIN(1,I)*XOLD(2,I)+GAIN(2,I)*XOLD(1,I)
      DUM1=PFOR1
      DUM2=PFOR2
      PFOR1=DUM1*FOR1-DUM2*FOR2
      PFOR2=DUM1*FOR2+DUM2*FOR1
      RES1=RES1+PFOR1*FOR1+PFOR2*FOR2
      RES2=RES2+PFOR2*FOR1-PFOR1*FOR2
      S1=RES1*RES1+RES2*RES2
      SS1=RES1/S1
      SS2=-RES2/S1
      GEX(1,1)=PFOR1*SS1-PFOR2*SS2
      GEX(2,1)=PFOR1*SS2+PFOR2*SS1
      DO 22 I=2,N+1
      GEX(1,I)=GAIN(1,I-1)-F(1,I-1)*GEX(1,1)+F(2,I-1)*GEX(2,1)
22    GEX(2,I)=GAIN(2,I-1)-F(1,I-1)*GEX(2,1)-F(2,I-1)*GEX(1,1)
      BAC1=XOLD(1,N)
      BAC2=XOLD(2,N)
      DO 23 I=1,N
      BAC1=BAC1-B(1,I)*X(1,I)-B(2,I)*X(2,I)
23    BAC2=BAC2-B(1,I)*X(2,I)+B(2,I)*X(1,I)
      V1=1.-GEX(1,N+1)*BAC1-GEX(2,N+1)*BAC2
      V2=-GEX(2,N+1)*BAC1+GEX(1,N+1)*BAC2
      VV=V1*V1+V2*V2
      V1=V1/VV
      V2=-V2/VV
      DO 24 I=1,N
      B(1,I)=B(1,I)+GEX(1,I)*BAC1+GEX(2,I)*BAC2
      B(2,I)=B(2,I)+GEX(2,I)*BAC1-GEX(1,I)*BAC2
      U1=B(1,I)*V1-B(2,I)*V2
      U2=B(1,I)*V2+B(2,I)*V1
      B(1,I)=U1
      B(2,I)=U2
24    CONTINUE
      DO 25 I=1,N
      GAIN(1,I)=GEX(1,I)+B(1,I)*GEX(1,N+1)-B(2,I)*GEX(2,N+1)
25    GAIN(2,I)=GEX(2,I)+B(1,I)*GEX(2,N+1)+B(2,I)*GEX(1,N+1)
      Y1=0.
      Y2=0.
C
C     OUTPUT
C
      DO 28 I=1,N
      IF(IQ.EQ.0)GO TO 26
C
C     QUANTIZATION OF TAP VALUES
C
      A1=INT(C(1,I)*BNN1+0.5)/BNN1
      A2=INT(C(2,I)*BNN1+0.5)/BNN1
      GO TO 27
26    A1=C(1,I)
      A2=C(2,I)
27    CONTINUE
      Y1=Y1+A1*X(1,I)+A2*X(2,I)
28    Y2=Y2+A1*X(2,I)-A2*X(1,I)
C
C     ERROR
C
      E1=D(1,M)-Y1
      E2=D(2,M)-Y2
      P=P+E1*E1+E2*E2
C
```

```
C        EQUALIZATION
C
         DO 29 I=1,N
         C(1,I)=C(1,I)+E1*GAIN(1,I)+E2*GAIN(2,I)
29       C(2,I)=C(2,I)+E1*GAIN(2,I)-E2*GAIN(1,I)
C
C        SHIFTING
C
         DO 30 I=1,N
         XOLD(1,I)=X(1,I)
30       XOLD(2,I)=X(2,I)
31       DO 32 IS=1,N
         ISS=N+1-IS
         X(1,ISS+1)=X(1,ISS)
         X(2,ISS+1)=X(2,ISS)
32       CONTINUE
         DO 33 IS=1,N
         ISS=N+1-IS
         D(1,ISS+1)=D(1,ISS)
         D(2,ISS+1)=D(2,ISS)
33       CONTINUE
         IF(L.LT.LIM)GO TO 35
         JK=JK+1
         XL=L
         YP(JK)=10.*ALOG10(P/XL)
         P1=YP(JK)
34       P=0.
         L=0
35       L=L+1
1000     CONTINUE
         CALL MOVEA(0.,YP(1))
         DO 36 I=1,JK
         XP=(I-1)*LIM
         CALL DRAWA(XP,YP(I))
36       CONTINUE
         WRITE(6,37)
37       FORMAT(40X,'TAP COEFFICIENTS',/)
         WRITE(6,38)(C(1,I),C(2,I),I=1,N)
38       FORMAT(40X,F12.8,10X,F12.8,/)
39       FORMAT(10X,'SEED,NOISE VARIANCE')
40       FORMAT(2F)
41       FORMAT(10X,'NOISE VARIANCE=',F)
         CALL FINITT(0,767)
         END
```

```
      SUBROUTINE GRID(XMIN,XMAX,DX,YMIN,YMAX,DY)
      Y=YMIN
      IY=ABS(YMAX-YMIN)/DY+1.
      DO 1 I=1,IY
      CALL MOVEA(XMIN,Y)
      CALL DRAWA(XMAX,Y)
    1 Y=Y+DY
      X=XMIN
      IX=ABS(XMAX-XMIN)/DX+1.
      DO 2 I=1,IX
      CALL MOVEA(X,YMIN)
      CALL DRAWA(X,YMAX)
    2 X=X+DX
      RETURN
      END
```

```
C      ********************************************
C       GAUSSIAN NOISE GENERATOR
C
C
C
       SUBROUTINE GANOS(X,N,XQRT,SIGMA)
       DIMENSION X(1)
       IU=X(1)
       JU=-X(1)
       DO 1 I=1,10
 1     XX=RAN(IU,JU)
       DO 2 I=1,N
 2     X(I)=RAN(IU,JU)
       PIKI=2.*3.14159
       XA=SIGMA*SQRT(2.)
       DO 3 I=1,N,2
       XR=XA*SQRT(-ALOG(1.-X(I)))
       QM=X(I+1)*PIKI
       X(I)=XR*SIN(QM)+XQRT
 3     X(I+1)=XR*COS(QM)+XQRT
       M=N/2
       IF(N.EQ.2*M)GO TO 4
       X(N)=(X(1)+X(2))/2.
 4     RETURN
       END
```

REFERENCES

1- R.W.Lucky, "Techniques For Adaptive Equalization of Digital Communication Systems", BSTJ, Vol 45, Feb.1966, pp.255-86.

2- R.W.Lucky, "Automatic Equalization For Digital Communication", BSTJ, Vol.44, Apr.1965, pp.547-88.

3- A.Gersho, "Adaptive Equalization of Highly Dispersive Channels For Data Transmission", BSTJ, Vol.48, Jan.1969, pp.55-70.

4- J.G.Proakis and J.H.Miller, "An Adaptive Receiver For Digital Signalling Through Channels With Intersymbol Interference", IEEE Trans. Inform Theory, Vol.IT-15, July 1969. pp.484-97.

5- J.M.McCool and B.Windrow, "Principles and Applications of Adaptive Filters: A Tutorial Review", Standford University, Rep. IEEE Sept.1976.

6- B.Widrow,J.M.McCool, M.G.Larimore, C.R.Johnson, "Stationary and Nonstationary Learning Characteristics of the LMS Adaptive Filter", Proc.IEEE, Vol.64, Aug.1976, pp.1151-1162.

7- R.W.Chang, "A New Equalizer Structure For Fast Start-up Digital Communication", BSTJ, Vol.50, July-August 1971, pp.1969-2013.

8- G.Ungerboeck, "Theory on the Speed of Convergence in Adaptive Equalizers For Digital Communication", IBM, IBM Res-Develop., Vol.16, Nov.1972, pp.546-55.

9- R.D.Gittlin and F.R.Mogee, "Self-Orthogonalizing Adaptive Equalization Algorithms", IEEE Trans. on Communications Vol.COM-25, July 1977, pp.666-72.

10- R.D.Gittlin, J.E.Mazo and M.G.Taylor, "On the Design of Gradient Algorithms For Digitally Implemented Adjustment Filters", IEEE Trans. on Circuit Theory, Vol. CT-20, March 1973, pp.125-36.

11- R.D.Gittlin, and S.B.Weinstein, "On the Required Tap Weight Precision For Digitally Implemented. Mean Squared Equalizers", BSTJ, Vol.58, Feb.1979, pp.301-21.

12- D.D.Falconer and L.Ljung, "Application of Fast Kalman Estimation to Adaptive Equalization", IEEE Trans. on Communications, Vol.COM-26, Oct. 1978, pp.1439-46.

13- M.S.Mueller, "Least-Squares Algorithms For Adaptive Equalizers", B.S.T.J., Vol.60, Oct.1981, pp.1905-25.

14- L.Ljung, M.Morf and D.D.Falconer, "WastCalculation of Gain Matrices For Recursive Estimation Schemes", Int.J.Control, Vol.27, Jan.1978, pp.1-19.

15- D.Godard, "Channel Equalization Using a Kalman Filter For Fast Data Transmission", IBM Res. Develop., May 1974, pp.267-73.

16- R.D.Gittlin and S.B.Weinstein, "Fractionally-Spaced Equalization: An Improved Transversal Equalizer", BSTJ, Vol. 60, Feb. 1981, pp.275-96.

17- S.Qureshi, "Adaptive Equalization", IEEE Comm.Mag., March 1982, pp.9-16.

18- L.Ljung, "Analysis of Recursive Stochastic Algorithms", IEEE Trans. Automat. Contr., Vol.AC-22, Aug.1977, pp.551-75.

19- D.C.Forden, "Stochastic Approximation With Correlated Data", IEEE Trans.on Inform Theory. Vol.IT-27, Jan. 1981, pp.105-13.

20- P.Butter and A.Cantoni, "Noniterative Automatic Equalization", IEEE Trans. on Communications, Vol.COM-23, June 1975, pp.621-33.

21- E.H.Satorius and S.T.Alexander, "Channel Equalization Using Adaptive Lattice Algorithms", IEEE Trans. on Communications. Vol.COM-27, June 1979, pp.899-905.

22- L.Frank, Signal Theory, Prentice-Hall, New Jersey 1969.

23- J.M.Mendel, Discrete Techniques of Parameter Estimation, Marcel Dekker, New York, 1973.

24- A.P.Sage and J.L.Melsa, System Identification, Academic Press, New York 1971.

25- A.H.Jazwinski, Stochastic Processes and Filtering Theory, Academic Press, New York, 1970.

26- R.W.Lucky, J.Salz and E.J.Weldon, Principles of Data Communication, McGraw-Hi-l, New York, 1968.

27- J.Istefonopulos, State Variables and Linear Control Systems, Boğaziçi University Publications, İstanbul 1981.

28- A.V.Oppenheim and R.W.Schafer, Digital Signal Processing, Prentice-Hall, New Jersey, 1975.

29- F.J.Gantmacher, The Theory of Matrices, Chelsea Publishing Co., New York, 1964.