A STUDY OF DISCRETE-TIME
ADAPTIVE FILTER SYSTEMS

A Thesis

Presented in Partial Fulfillment of the Requirements
for the Degree Master of Science

by

John Francis Maguire, B.S.

The Ohio State University

Approved by

C. E. Warren

Adviser
Department of Electrical Engineering

ACKNOWLEDGMENTS

The author wishes to express his deep appreciation to Professor C. E. Warren who originally suggested the subject of this thesis and who was always willing to give valuable assistance when needed.

The author is also very grateful to his family and close friends who gave him encouragement and moral support throughout his graduate studies.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES, cont'd.

LIST OF FIGURES, cont'd.

CHAPTER I

INTRODUCTION

In recent years the concept of digital or sampled-data systems has become a reality in the field of Communications. Such systems have often proven to be more efficient, less complicated, and much faster means of sending and receiving various types of signals than their continuous-time predecessors.

Several mathematical studies have been made in the past ten years concerning the feasibility of particular networks or elements of digital communications systems. One such study was performed by the Stanford University Systems Theory Laboratory. The primary objective was to describe and analyze the concept of an adaptive filter.

The term adaptive filter might be defined as a device that processes incoming signals or other data so as to produce a desired response, with the added feature that it is self-optimizing and may be adjusted in order to minimize some chosen performance or error criterion.

The purpose of this thesis is to simulate various types of adaptive filters using a digital computer and to study and analyze their performance under certain input/output requirements and for variations in design parameters.

Prior to presenting the computer programs and the experimental results, some of the background theory developed by the Stanford Systems Laboratory will be discussed.

CHAPTER II

THEORY

The term "filter" is often applied to any device or system that processes incoming signals or other data in such a way as to eliminate noise, or smooth signals, or identify each signal as belonging to a particular class, or even predict the next signal from moment to moment. The term "adaptive" used in this text refers to the ability of a system to self-optimize or self-adjust its variable parameters in order to optimize some chosen performance criterion.

The type of adaptive filter that will be discussed is a sampled-data system in which the input and output signal levels exist at only fixed instants of time, forming numerical sequences. Thus the input and output relations can be described by means of difference equations.

Two kinds of processes take place in an adaptive filter: training and operating. The training process (adaptation) is concerned with adjusting parameters (in this case variable weights or gains). The operating process consists in forming output signals by weighing the various input (and or feedback) samples, using the final optimum weight values resulting from the training process.

During the training process an additional input signal, the "desired response", must be supplied to the filter in addition to the usual input signals. It is precisely the difference between this desired signal and the actual filter output which forms an error signal used to correct or adjust the weights between successive input samples.

The performance criterion used throughout this paper will be minimum mean square error between desired and filter outputs. A least mean square algorithm based on a gradient search or "method of steepest descent" will be used as the optimizing scheme for weight adjustment.

In the case of a purely feedforward or transversal adaptive filter, the quadratic form for the mean square error "surface" as a function of the weight settings is assured and a unique minimum exists. However, the performance criterion must be modified slightly to eliminate the possibility of local minima in this same error surface when a feedback system is employed. Each case will be thoroughly discussed in this section.

Although much of the theory presented will refer to statistical properties of the input samples, it is a simple matter to adapt these ideas to deterministic signals for which means and variances have little meaning. The same filter operations are valid in both cases.

Now that a general view of adaptive filters has been presented, it is appropriate to discuss the mathematics of particular designs of these filters.

A. The Feedforward Discrete Adaptive Filter

The analysis of the feedforward adaptive filter performance will be based on a study of the system shown in Figure 2-1. Stationary input signals will be assumed.

A set of input signals is weighted and summed to form an output signal. The inputs are assumed to occur simultaneously and discretely in time.

The error signal at time L is given by:

$$E(L) = D(L) - Y(L) = D(L) - \sum_{i=1}^{N} W_i \, X(L - (i-1)) \tag{1}$$

where

$D(L)$ = the desired response at time L
$Y(L)$ = the filter output at time L
$X(L)$ = the input at time L
W_i = the i^{th} variable gain or weight

The square of the error given by (1) is:

$$E^2(L) = D^2(L) + \sum_{i=1}^{N} \sum_{j=1}^{N} W_i \, W_j \, X\,(L - (i-1)) \, X\,(L - (j-1))$$

$$-2\,D(L) \sum_{i=1}^{N} W_i \, X\,(L - (i-1)) \tag{2}$$

The expected value of this error squared (the mean-square error) is given by:

$$\overline{E^2(L)} = \overline{D^2(L)} + \sum_{i=1}^{N} \sum_{j=1}^{N} W_i \, W_j \, \phi(X_i,X_j)$$

$$-2 \sum_{i=1}^{N} W_i \, \phi(X_i,D) \tag{3}$$

where $\phi(X_i,D) \equiv \overline{D(L)\ X(L - (i-1))}$

$\phi(X_i,X_j) \equiv \overline{X(L - (i-1))\ X(L - (j-1))}$

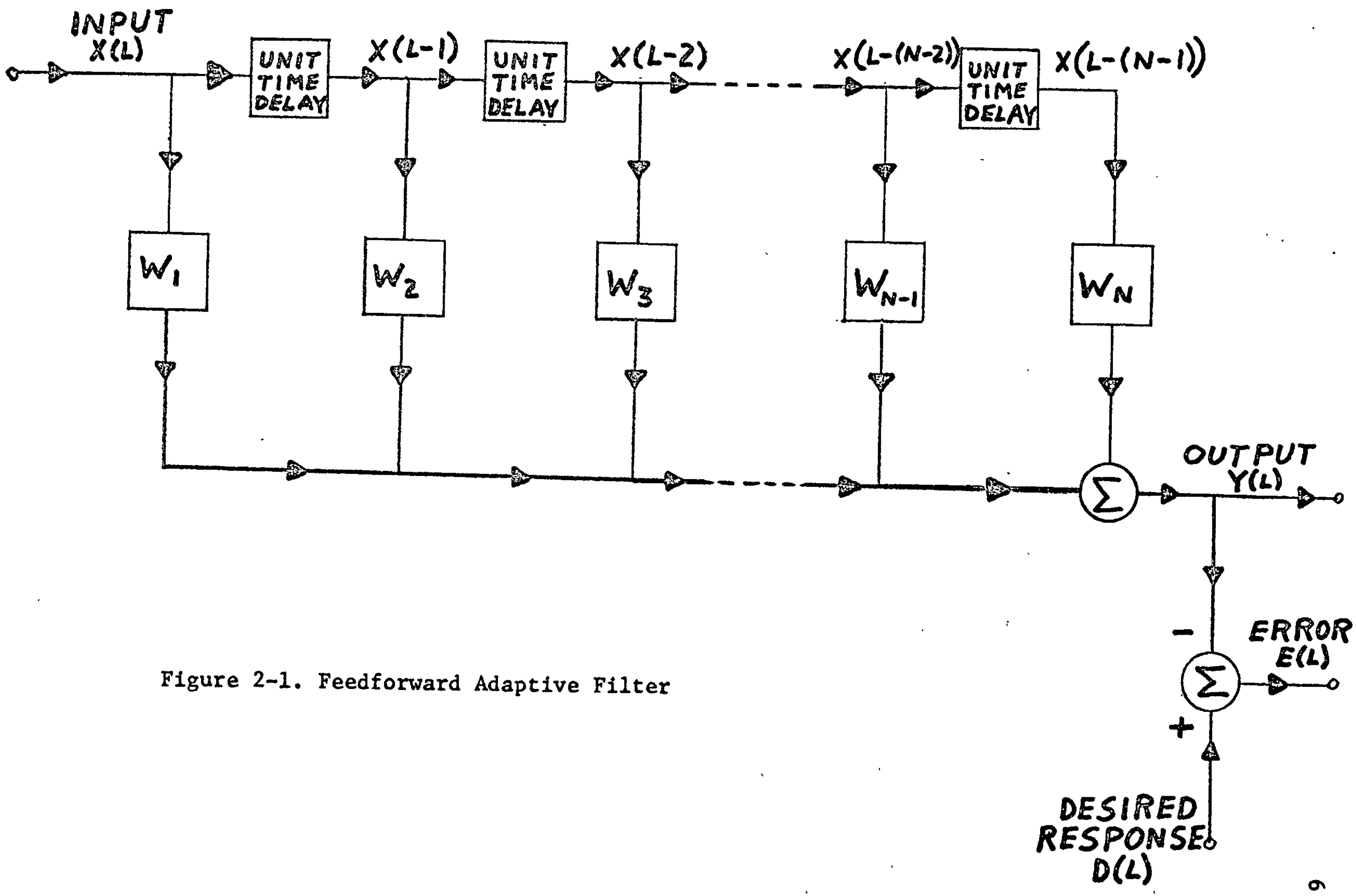

Figure 2-1. Feedforward Adaptive Filter

are the statistical correlations between input and desired response, and combinations of input samples respectively.

It may be observed that for stationary inputs, the mean-square error is a second-order or parabolic function of the weights so that a unique minimum may be found. The means that will be used to accomplish this is known as the method of steepest descent.[3,8]

The steepest descent technique uses gradients of the performance surface in seeking a minimum point on the surface. The gradient at any point on the performance surface may be obtained by differentiating the mean-square error function. The i^{th} gradient component from (3) is:

$$\frac{\partial \overline{E^2(L)}}{\partial W_i} = -2\ \phi(X_i,D) + 2 \sum_{j=1}^{N} W_j\ \phi(X_i,X_j) \tag{4}$$

To find the optimal set of weights, $\underline{W_{LMS}}$, that minimized $\overline{E^2(L)}$, set $\nabla\overline{E^2(L)} = 0$.

$$\underline{\nabla\overline{E^2(L)}} = -2\,\underline{\Phi(X,D)} + 2\,\underline{W}\,[\Phi(X,X)] \tag{5}$$

using matrix notation where

$$\underline{\Phi(X,D)} \equiv \underline{\phi(X_1,D),\ \phi(X_2,D),\ \ldots.,\ \phi(X_N,D)}$$

$$\underline{W} \equiv \underline{W_1,\ W_2,\ \ldots..,\ W_N}$$

$$[\Phi(X,X)] \equiv \begin{bmatrix} \phi(X_1,X_1) & \cdots\cdots & \phi(X_1,X_N) \\ \vdots & & \vdots \\ \phi(X_N,X_1) & \cdots\cdots & \phi(X_N,X_N) \end{bmatrix}$$

Accordingly,

$$\underline{W_{LMS}} = \underline{\Phi(X,D)}\ [\Phi(X,X)]^{-1} \tag{6}$$

An expression for the minimum mean-square error may be obtained in the form

$$\overline{E^2(L)}_{MIN} = \overline{D^2(L)} - \underline{\Phi(X,D)}\ \underline{W_{LMS}} \tag{7}$$

In seeking the minimum error by the method of steepest descent, an initial set of values is chosen for the weights. The next set weight values is obtained from the previous set by making a change in the weights in a direction opposite to the gradient vector of the error surface. If the mean-square error is reduced with each weight vector change, the process will converge on the stationary minimum regardless of the choice of initial weights.

The method of steepest descent for the feedforward adaptive filter weights may be expressed as a relation

$$\underline{W_{\substack{PRESENT\\CYCLE}}} = \underline{W_{\substack{PREVIOUS\\CYCLE}}} + k_s\ \underline{\nabla\overline{E^2}\left(\substack{PREVIOUS\\CYCLE}\right)} \tag{8}$$

where k_s is the adaptation constant of proportionality which controls the rate of change of the adaptive process.

The linearity of the gradient as a function of the weights and the quadratic form of the error surface, providing freedom from local minima, make the method of steepest descent a very desirable technique for this adaptive process.

It can be shown that the weights undergo geometric (discrete exponential) transients in relaxing toward the error surface minimum. If the unit of time is taken to be one iteration cycle, a time constant can be defined as the time constant of an exponential envelope $e^{-1/T}$ where T is the constant. If,

$$T >> 1 \text{ then } T_p \approx \frac{-1}{2k_s \lambda_p} \tag{9}$$

where T_p is the time constant of the p^{th} weight expressed in a proper coordinate system[8] and λp is the p^{th} eigenvalue of the correlation matrix $[\Phi(X,X)]$. The number of natural modes is equal to the number of weights N.

Widrow has shown that the steepest descent adaptation process is stable when all $Tp > 1/2$. Since the eigenvalues of a correlation matrix are always $\geqslant 0$, the only way that stability can be assured is for the conditions $|k_s \lambda_{MAX}| < 1$ and $k_s < 0$ to be true. A bound can then be placed on the adaptation constant k_s[8]

$$-\frac{1}{\lambda_{MAX}} < k_s < 0 \ . \tag{10}$$

In practice, the true value of the gradient vector for the mean-square error surface is seldom available. To overcome this difficulty, a Least Mean-Square (LMS) Adaptation Algorithm offers an easy procedure for implementing the method of steepest descent. This algorithm uses measured gradient estimates in place of true gradient values. These estimates may be "noisy" (contain errors) but the error can be minimized through careful application of the LMS algorithm.

In effect the algorithm employs the gradient vector approximation for time L

$$\overline{\underline{\nabla E^2(L)}} \simeq \underline{\nabla E^2(L)} = -2\,E(L)\,\underline{X(L)} \quad * \tag{11}$$

All that is needed in order to estimate the gradient is the present input-signal vector $\underline{X(L)}$ and its associated scalar error E(L). It can be shown that $\overline{\underline{(\nabla E^2(L))}} = \underline{\overline{\nabla E^2(L)}}$ and thus the gradient estimate is unbiased.[8]

The final form for the LMS algorithm for adjusting the weights can be expressed, using the gradient approximation, as

$$\underline{W\binom{\text{PRESENT}}{\text{CYCLE}}} = \underline{W\binom{\text{PREVIOUS}}{\text{CYCLE}}} - 2\,k_s\,E\binom{\text{PREVIOUS}}{\text{CYCLE}}\,\underline{X\binom{\text{PREVIOUS}}{\text{CYCLE}}} \tag{12}$$

or

$$\underline{W(L)} = \underline{W(L-1)} - 2\,k_s\,E(L-i)\,\underline{X(L-1)}$$

for the weight vector at time L.

The expression for the adaptation time constant using the LMS algorithm is the same as stated earlier in (9). The bounds on k_s which insure stability of the LMS algorithm may be expressed in a different manner than (10).[8]

$$0 > k_s > -\frac{1}{||\underline{X}||^2_{MAX}} \tag{13}$$

where $||\underline{X}||^2$ is the squared magnitude of the input vector.

---

* $\underline{X(L)}$ consists of all input samples at time L which affect the output i.e., X(L), X(L-1), ..., X(L-(N-1)).

Thus If the maximum input vector magnitude is known or can be estimated closely, the adaptation constant stability range is well defined without knowledge of the eigenvaLues of the correlation matrix. If it is not known a priori, then the magnitude can be estimated and updated as more input-signaL observations are made.

As was stated previously, the difference between the true gradient and the measured gradient estimate used in the LMS algorithm can introduce error or gradient-measurement noise. After transients in the adjustments essentialLy die out (3 to 5 time constants of slowest natural mode), there can still be random fluctuations of the weight values about their LMS-optimal values. If the input signal is deterministic, then such "noise" will not be observed since the true gradient and measured gradient are equaL.

Widrow has gone through extensive analysis to show that the amplitude of this steady state noise has a statistical relationship to the input signals expressed by[8]

$$[\Phi(V,V)] = 4\ (\overline{E^2_{MIN}})\ [\Phi(X,X)] \tag{14}$$

where

$$[\Phi(V,V)] \equiv \begin{bmatrix} \overline{V_{iL}\ V_{IL}} & \overline{V_{iL}\ V_{2L}} & \cdots \\ \vdots & & \\ \overline{V_{NL}\ V_{iL}} & \cdots & \overline{V_{NL}\ V_{NL}} \end{bmatrix}$$

$$V_{iL} = \frac{\partial \overline{E^2(L)}}{\partial W_i} - \frac{\partial E^2(L)}{\partial W_i}$$

$\overline{E^2_{MiN}}$ = minimum mean square error

He has also shown that the average excess mean-square error due to adaptation in the steady state is related to the normal mode time constants by[8]

$$\overline{(E^2(L))} - \overline{E^2_{MIN}} = \frac{1}{2}\,(\overline{E^2_{MIN}}) \sum_{p=1}^{N} \frac{1}{T_p} \tag{15}$$

summed over all modes p.

It is observed that if the adapting process is done slowly (small k_s), the T_p's will be large and theoretically the excess mean-square error can be made arbitrarily small. Slow adaptation acts as a gradient noise filter.

B. <u>The Feedback Discrete Adaptive Filter</u>

The analysis of the feedback discrete adaptive filter will be based on a study of the system shown in Figure 2-2.

A set of input signals and a set of previous output signals are weighted and summed to form the output signal. When the filter is in the normal mode after adaptation is complete, the output at time L may be expressed as a difference equation of the form

$$Y(L) = \sum_{i=1}^{N+1} A_i\, X(L - (i-1)) + \sum_{j=1}^{N} B_j\, Y(L - j) \tag{16}$$

The error criterion used for the feedback adaptive filter is slightly different from that used in the feedforward case. Mantey[4] in his study of such systems has proven that the sum-squared difference criterion in which the error at time L is given by

$$E(L) = D(L) - Y(L)\ , \tag{17}$$

where D(L) is the desired response at time L
and Y(L) is the filter output at time L given by (16),

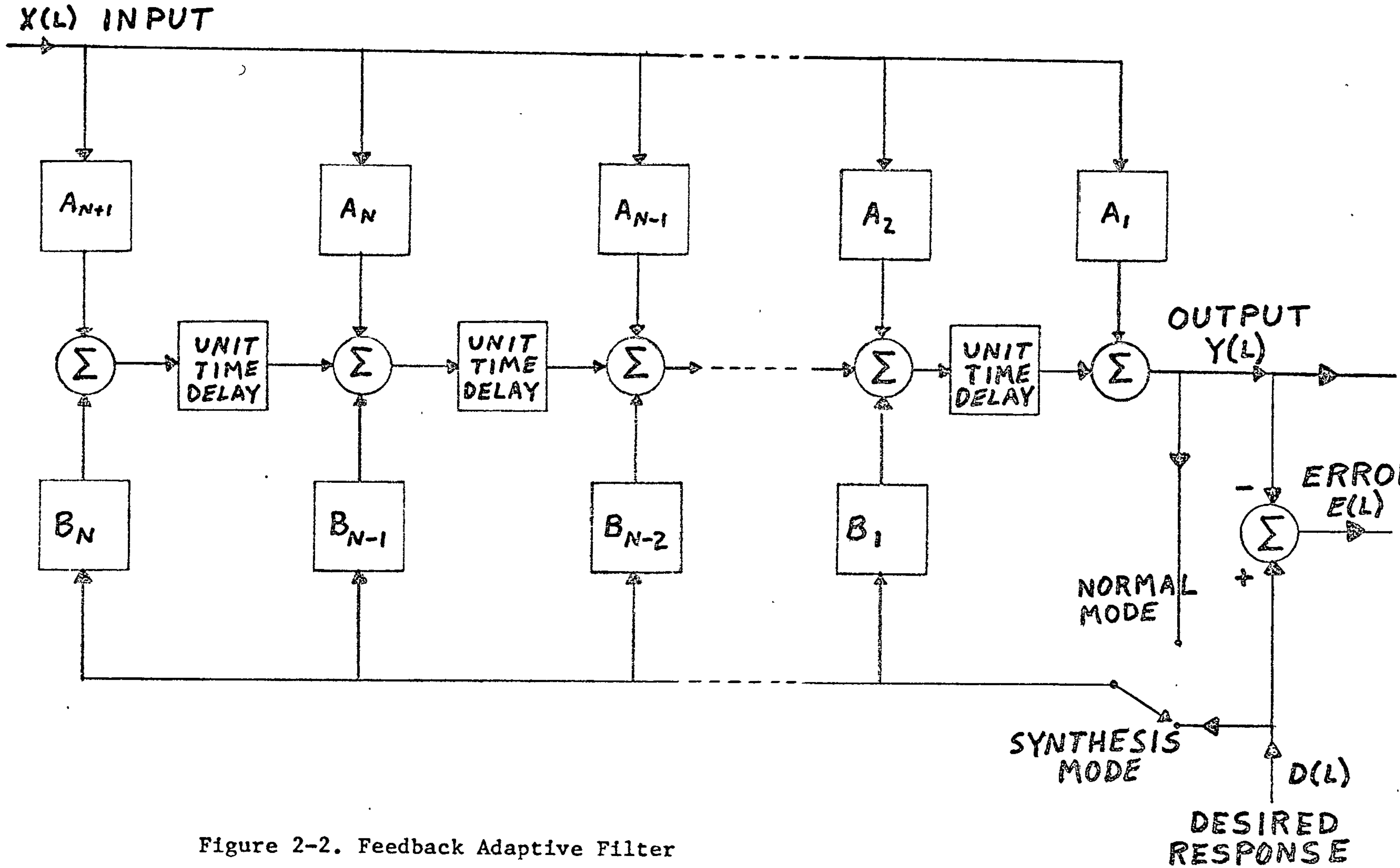

Figure 2-2. Feedback Adaptive Filter

will not have a unique set of weight values for the minimum mean-squared error. It can be shown that the error surface

$$S = \sum_{k=0}^{\infty} (E(L))^2 = \sum_{k=0}^{\infty} (D(L) - Y(L))^2 \qquad (18)$$

is quadratic in the feedforward weights A_i but not in the feedback weights B_i. So if the feedback weights are not fixed, there is a high probability that local minima will exist on the error surface S. In fact, the partial derivative of S with respect to each of the weights B_i is a function of order N + 2 in these variables. Thus the possibility of as many as N + 2 real solutions to the equation for the minima, $\frac{\partial S}{\partial B_i} = 0$, exists.[4]

Therefore, the desirable method of steepest descent for determining the optimal set of weight values could fail.

In order to circumvent this problem a modified quadratic performance criterion with a unique minimum can be established. Much of the analysis of such an error surface is based on subtle theory of Z-Transforms aud will not be presented here.[4] However, physically, this performance measure is the sum of the squares of the differences between the network output and the desired output, with the modification, that, during the synthesis or adaptation process the desired output is feedback instead of the actual filter output.

This error surface is quadratic in all of the variable weights and therefore has a single, optimal set of weight values for the minimum. In fact, if a set of values exists for which (18) vanishes, then this solution also makes the modified error surface vanish. So this criterion as a measure of "goodness" is established.[4]

Under the new criterion the error at time L is given by

$$E(L) = D(L) - \sum_{i=1}^{N+1} A_i\, X(L - (i-1)) - \sum_{j=1}^{N} B_j\, D(L-j) \tag{19}$$

The square of the error in (19) is

$$\begin{aligned} E^2(L) = D^2(L) &+ \sum_{i=1}^{N+1} \sum_{k=1}^{N+1} A_i\, A_k\, X(L - (i-1))\, X(L - (k-1)) \\ &+ \sum_{j=1}^{N} \sum_{k=1}^{N} B_j\, B_k\, D(L-j)\, D(L-k) \\ &+ 2 \sum_{i=1}^{N+1} \sum_{j=1}^{N} A_i\, B_j\, X(L - (i-1))\, D(L-j) \\ &- 2\, D(L) \sum_{i=1}^{N+1} A_i\, X(L - (i-1)) \\ &- 2\, D(L) \sum_{j=1}^{N} B_j\, D(L-j) \end{aligned} \tag{20}$$

The mean-squared error is given by

$$\begin{aligned} \overline{E^2(L)} = \overline{D^2(L)} &+ \sum_{i=1}^{N+1} \sum_{k=1}^{N+1} A_i\, A_k\, \phi(X_i, X_k) \\ &+ \sum_{j=1}^{N} \sum_{k=1}^{N} B_j\, B_k\, \phi(D_j, D_k) \\ &+ 2 \sum_{i=1}^{N+1} \sum_{j=1}^{N} A_i\, B_j\, \phi(X_i, D_j) \\ &- 2 \sum_{i=1}^{N+1} A_i\, \phi(X_i, D) - 2 \sum_{j=1}^{N} B_j\, \phi(D, D_j) \end{aligned} \tag{21}$$

where

$$\phi(X_i,X_k) \equiv \overline{X(L - (i-1)\ X(L - (k-1))}$$

$$\phi(D_j,D_k) \equiv \overline{D(L-j)\ D(L-k)}$$

$$\phi(X_i,D_j) \equiv \overline{X(L - (i-1))\ D(L-j)}$$

$$\phi(X_i,D) \equiv \overline{X(L - (i-1))\ D(L)}$$

$$\phi(D,D_j) \equiv \overline{D(L)\ D(L-j)}$$

are the statistical correlations between various combinations of input and desired output samples.

Again the method of steepest descent will be employed to seek the error surface minimum. The i^{th} gradient component for the feedforward weights from (21) is

$$\frac{\partial \overline{E^2(L)}}{\partial A_i} = 2 \sum_{k=1}^{N+1} A_k\ \phi(X_i,X_k) + 2 \sum_{j=1}^{N} B_j\ \phi(X_i,D_j) - 2\ \phi(X_i,D) \tag{22}$$

and the j^{th} gradient component for the feedback weights is given by

$$\frac{\partial \overline{E^2(L)}}{\partial B_j} = 2 \sum_{k=1}^{N} B_k\ \phi(D_j,D_k) + 2 \sum_{i=1}^{N+1} A_i\ \phi(X_i,D_j) - 2\ \phi(D,D_j) \tag{23}$$

The optimal weight vectors, $\underline{A_{LMS}}$ and $\underline{B_{LMS}}$ that minimize $\overline{E^2(L)}$ are obtained by setting the partial derivative expressions equal to zero and solving two sets of vector equations simultaneously.

$$\underline{A_{1MS}} = (\underline{\Phi(X,D)} - (\underline{\Phi(D,D)} - \underline{\Phi(X,D)}$$
$$[\Phi(X,X)]^{-1}\ [\Phi(X,D)])([\Phi(D,D)] - [\Phi(X,D)]$$
$$[\Phi(X,X)]^{-1}\ [\Phi(X,D)])^{-1}\ [\Phi(X,D)])\ [\Phi(X,X)]^{-1}$$

$$\underline{B_{1MS}} = (\underline{\Phi(D,D)} - \underline{\Phi(X,D)}\ [\Phi(X,X)]^{-1}\ [\Phi(X,D)])$$
$$([\Phi(D,D)] - [\Phi(X,D)]\ [\Phi(X,X)]^{-1}\ [\Phi(X,D)])^{-1} \qquad (24)$$

where

$$[\Phi(D,D)] \equiv \begin{bmatrix} \phi(D_1,D_i) & \cdots\cdots & \phi(D_i,D_N) \\ \vdots & & \vdots \\ \phi(D_N,D_1) & \cdots\cdots & \phi(D_N,D_N) \end{bmatrix}$$

$$[\Phi(X,D)] \equiv \begin{bmatrix} \phi(X_1,D_1) & \cdots\cdots & \phi(X_1,D_N) \\ \vdots & & \vdots \\ \phi(X_{N+1},D_1) & \cdots\cdots & \phi(X_{N+1},D_N) \end{bmatrix}$$

$$\underline{\Phi(D,D)} \equiv \underline{\phi(D_1,D),\ \phi(D_2,D), \ldots\ldots, \phi(D_N,D)}$$

$$\underline{\Phi(X,D)} \equiv \underline{\phi(X_i,D),\ \phi(X_2,D),\ \ldots\ldots\ \phi(X_{N+1},\ D)}$$

and the other correlation matrices are identical to those in the feed-forward case. Statistically stationary signals are assumed.

In order to implement the system a measured gradient estimate similar to that developed for the feedforward case is used. From (2D) the gradient estimates are obtained.

$$\frac{\partial[E^2(L)]}{\partial A_i} = -2\ E(L)\ X(L-i)$$

$$\frac{\partial[E^2(L)]}{\partial B_j} = -2\ E(L)\ D(L-j) \tag{25}$$

As in (11) the gradient vector approximation for time L, $\underline{\overline{\nabla E^2(L)}} \approx \underline{\nabla E^2(L)}$, has been used.

The final forms for the LMS algorithm for adjusting the weights can be expressed as

$$\underline{A(L)} = \underline{A(L-i)} - 2\ k_i\ E(L-i)\ \underline{X(L-1)} \tag{26}$$

for the feedforward gains at time L, and

$$\underline{B(L)} = \underline{B(L-1)} - 2\ k_2\ E(L-1)\ \underline{D(L-1)} \tag{27}$$

for the feedback gains at time L,

where k_1 = adaptation constant for the feedforward weights

k_2 = adaptation constant for the feedback weights

$\underline{X(L-1)}$ = the vector of input samples in the filter at time L-1

$\underline{D(L-1)}$ = the vector of desired response samples in the filter at time L-1.

By referring to (13) the stability bounds on the adaptation constants k_1 and k_2 can be established easily since the LMS algorithm for the feedback adaptive filter is the same as that for the feedforward case.[8]

$$0 > k_i > -\frac{1}{||\underline{X}||^2_{MAX}}$$

$$0 > k_2 > -\frac{1}{||\underline{D}||^2_{MAX}} \tag{28}$$

If a single adaptation constant k_s for adjusting all weights is desired, then a value that lies within both intervals must be chosen. A typical bound for k_s, although somewhat stricter than required, would be

$$0 > k_s > - \frac{1}{||\underset{\sqcup}{X}||^2_{MAX} + ||\underset{\sqcup}{D}||^2_{MAX}} \tag{29}$$

All weights undergo geometric transients in relaxing toward the error surface minimum. The expressions for these adaptation time constants are exactly the same as those developed for the feedforward case in (9).[8]

$$T_p \approx \frac{-1}{2\ k_1\ \lambda p} \tag{30}$$

for the feedforward weights, and

$$T_q \approx \frac{-1}{2\ k_2\ \lambda q}$$

for the feedback weights, where λp is the p^{th} eigenvalue of the correlation matrix $[\Phi(X,X)]$ and λq is the q^{th} eigenvalue of the correlation matrix $[\Phi(D,D)]$. The number of natural modes is equal to the number of weights $2N + 1$.

If a single adaptation constant k_s is used, then $k_1 = k_2 = k_s$ in (30).

It is evident that for non-deterministic signals gradient-measurement noise in the steady state will be observed because the actual and estimated mean-square error gradients are not equal. But no specific statistical quantities for the amplitude of these "noise" fluctuations about the optimal weight values has been developed for the feedback adaptive filter.

In Chapter V the experimental results will be presented. It is hoped that through analysis of these data a hypothesis concerning the merits of using feedback might be formulated. Because the output of the feedback adaptive filter is a function of previous outputs and hence of <u>all past inputs</u>, there is an advantage of more "working" knowledge for the filter in this case. The effect remains to be seen.

CHAPTER III

DISCUSSION OF TEST PROBLEMS

In order to study the adaptive filter under a wide variety of conditions, three very different problems were considered. For each case, given a known input, a certain desired output was specified. The operation of adaptive filters utilizing only feedforward weights, or only feedback weights, or a combination of the two was studied and analyzed in each of the three cases.

Computer programs were written to simulate the filter signal processing and internal adjustment or optimization schemes. These programs were constructed so that they would be easy to change to accommodate the three problems considered and the variation of filter parameters required. More will be said about this phase of the experiment in Chapter IV.

For each type of adaptive filter the number of weights or variable gains, the rate of adaptation, and the number of iteration or adjustment cycles were varied in a wide range so that their effect upon the filter response could be examined. The results will be discussed in Chapter V.

At this time it is appropriate that the three problems be defined and discussed both mathematically and practically.

A. SIN X/X Continuous Time Response

Most of the literature published on the subject of adaptive filters is concerned with continuous time responses. Both the input and desired output signals are zero for only a finite number of points.

Therefore, it seemed advantageous to formulate a problem that utilized only continuous time waveforms for the first test of the adaptive system. By analyzing these results any unforeseen difficulties in the simulation programs or in the application of the theory might be detected early.

The time functions chosen for this phase of the experiment were of the SIN X/X type. Since it was desired to make the filter response causal and since the computer simulation used subscripted variables (the subscripts must be positive integers), only the part of the waveforms for $t > 0$ could be utilized. The input sampling began at $t = 1$ and continued every integer time unit for the duration of the sampling interval.

The input signal chosen was

$$x(t) = \frac{\text{SIN } t/8}{\pi t} \; u(t)$$

and the desired output signal chosen was

$$d(t) = 3 \, \frac{\text{SIN } t/8}{\pi t} \; u(t)$$

where $u(t)$ is the unit step function. Time delays through the filter were ignored.

The corresponding Fourier Transforms for these waveforms are[5]

$$X(\omega) = \frac{1}{2\pi} (p_{1/8}(\omega) * U(\omega))$$

$$= \frac{1}{2\pi} (p_{1/8}(\omega) * (\pi\delta(\omega) + \frac{1}{j\omega}))$$

$$= \frac{1}{2} p_{1/8}(\omega), \ -1/8 \leq \omega \leq 1/8$$

$$= \frac{1}{2\pi j} \ln \left(\frac{\omega + 1/8}{\omega - 1/8}\right), \ \omega > 1/8; \ \omega < -1/8$$

where $p_{1/8}(\omega) = 1, \ -1/8 \leq \omega \leq 1/8$

$= 0$ otherwise

and $D(\omega) = 3X(\omega)$.

Since the system is linear and time-invariant the continuous time impulse response of the filter can be found from its Fourier Transform $H(\omega)$.

$$H(\omega) = D(\omega)/X(\omega) = 3 \quad -\infty < \omega < \infty$$

This is effectively an all-pass filter with linear amplifying characteristics at all frequencies. Such a filter has an impulse response of the form

$$h(t) = 3\delta(t) \quad .$$

In addition there would probably be some phase function $e^{-j\theta(\omega)}$ associated with $H(\omega)$. However, such a response can be made causal and realizable with proper choice of $\theta(\omega)$ since the Paley-Weiner Criterion is satisfied and the integral

$$\int_{-\infty}^{\infty} \frac{\ln 3}{1 + \omega^2} \, d\omega$$

converges.[5]

This particular problem really has little practical importance and was devised solely as an initial test of the simulated system using continuous time waveforms. However, an analysis of the results from this phase of the experiment provided valuable information concerning adaptation characteristics of the filter.

B. Square Pulse Distorted by RC Channel

The next problem considered was that of reconstructing a rectangular pulse that had been distorted by transmission through a channel with an RC (exponential decay) time-invariant impulse response characteristic.

In order to set up the problem assume that the square pulse f(t) shown in Figure 3-1 has been transmitted through a medium with an impulse response determined to be very similar to the time waveform h(t) shown in Figure 3-2 and have a frequency characteristic displayed in Figure 3-3. The output signal x(t) will be the convolution $f(t) * h(t)$ since the channel is linear, deterministic, and time-invariant.

$$x(t) = f(t) * h(t)$$

$$= \int_{-\infty}^{\infty} \frac{k}{RC} e^{-(t-T)/RC} u(t-T) \, [u(T) - u(T-10)] \, dT$$

$$= \frac{k}{RC} \left[\int_{0}^{t} e^{-(t-T)/RC} dT - \int_{10}^{t} e^{-(t-T)/RC} dT \right]$$

$$0 \leq t \leq \infty \qquad\qquad 10 \leq t \leq \infty$$

$$= \begin{cases} k\,(1 - e^{-t/RC}), & 0 \leq t \leq 10 \\ k\, e^{-t/RC}\,(e^{10/RC} - 1), & 10 \leq t \leq \infty \end{cases}$$

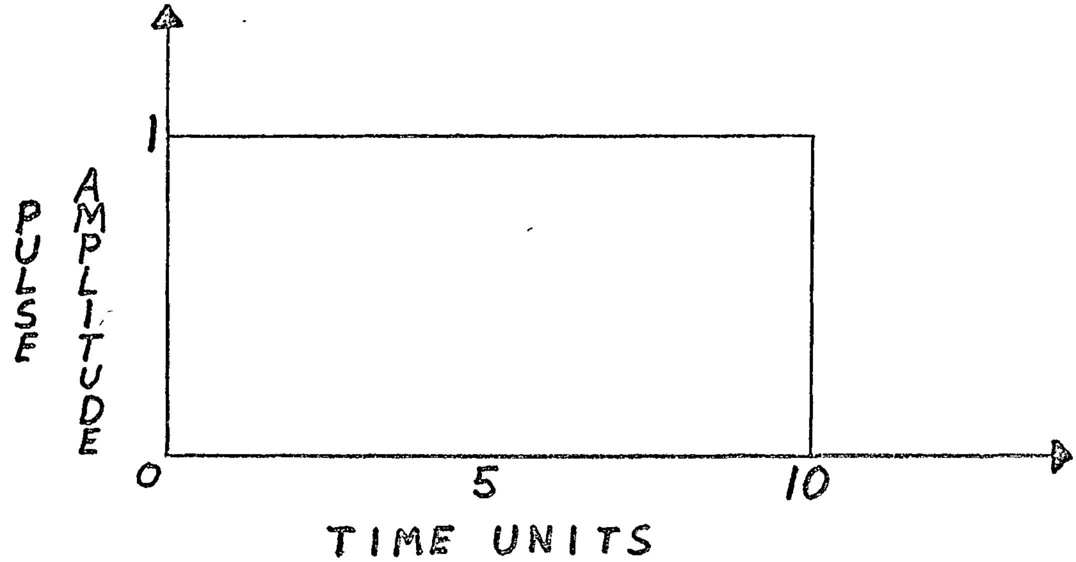

Figure 3-1. Rectangular Pulse Input to Channel

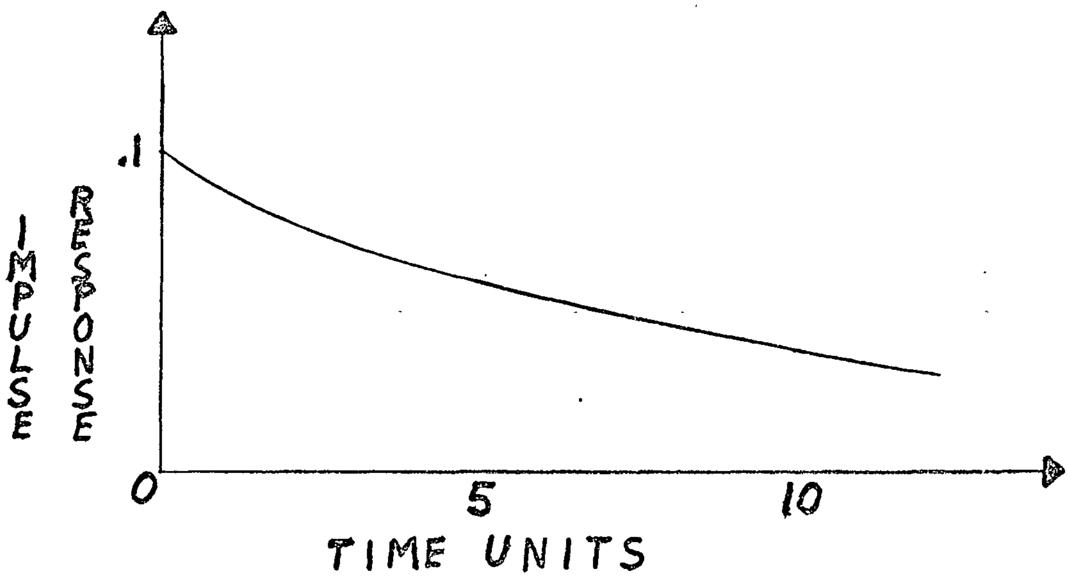

Figure 3-2. Channel Impulse Response

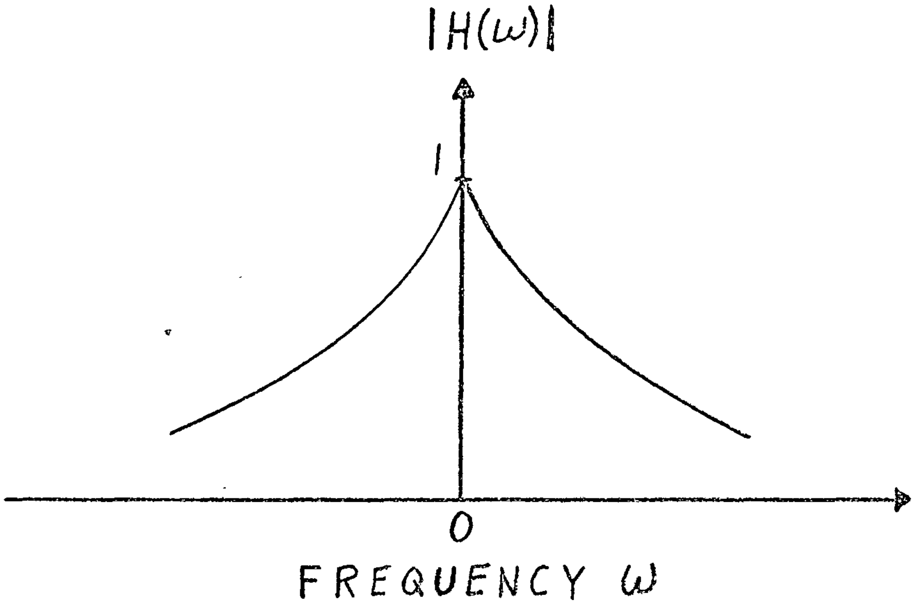

Figure 3-3. Channel Frequency Response (Magnitude)

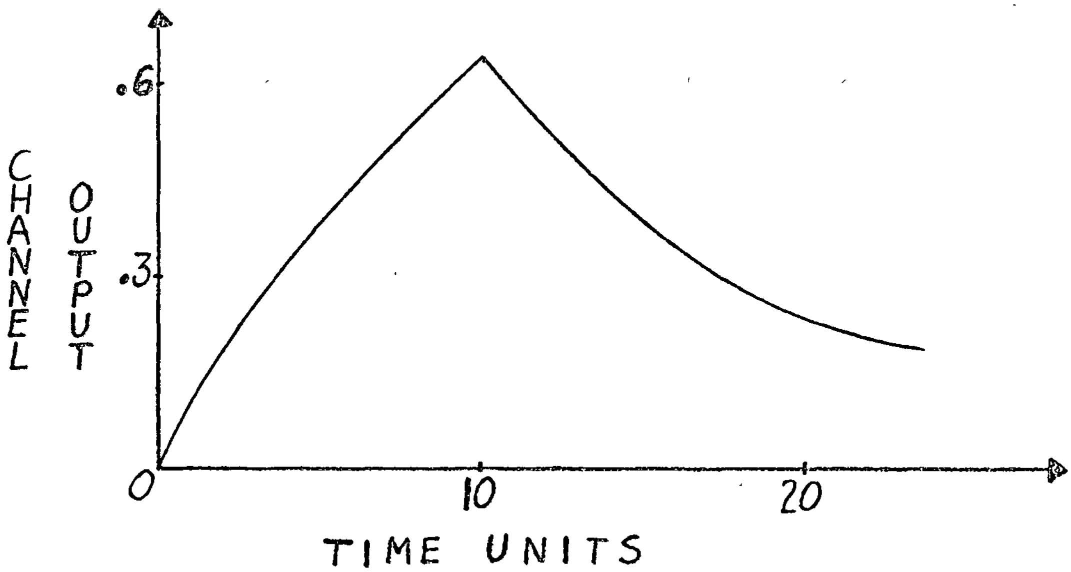

Figure 3-4. Channel Output Waveform

In this particular case k was given the value one for simplicity and the RC time constant was chosen to be 10. The reason for selecting this value for RC was that it insured enough distortion to eliminate any facsimile of the flatness of the original pulse, which would defeat the purpose of the experiment.

Figure 3-4 depicts the output waveform x(t).

This distorted signal was then applied as the input to the simulated adaptive filter system. The desired output was the original pulse. Tests were then made to determine how well the filter could adjust so that it would reconstruct the square pulse and compensate for the channel distortion. Time delays were again ignored.

The problem considered here, although somewhat narrowed in scope, has many practical applications. The need for general purpose "repeaters" to reduce the uncontrollable distortion caused by transmission media atmospheric conditions, local environment interference, and receiver front end noise has often been shown. Linear amplifiers alone will not suffice when the waveform shape has undergone great changes.

For the purposes of the simulation study a deterministic channel impulse response was assumed. In general, this problem would be handled statistically, considering the channel as representative of a stochastic process. But, as previously discussed, the filter operation is optimum under these conditions if certain statistical properties are known.

C. Square Pulse Distorted by Additive Gaussian Noise

The final problem considered in this experiment was that of reconstructing a rectangular pulse upon which additive White Gaussian Noise had been superimposed. The method used to simulate the Gaussian Process and provide the additive samples needed is discussed thoroughly in the first part of the appendix. It was assumed, for the sake of simplicity and because no test for checking serial correlation was available, that the samples were uncorrelated and hence independent.[6] The distribution had zero mean and unit variance approximately.

Figure 3-5 shows a block diagram of the simulated system. The square pulse waveform was sampled at unit time intervals and a noise sample was added at each instant to form the input signal to the filter. The desired response was the "clean" pulse. Again time delay through the filter was ignored.

The purpose of this phase of the experiment was to determine how effectively the filter could smooth out or reduce the noise distortion on the pulse. The basic difference between this problem and the previous two is that the filter input is actually a stochastic process with known statistical properties rather than a deterministic signal.

Most types of thermal noise such as that encountered at the receiver front end or in circuit elements such as resistors and gas tubes may be represented as additive broadband Gaussian Noise. In many other situations noise interferences are assumed to be Gaussian for the purpose of simplifying analysis.

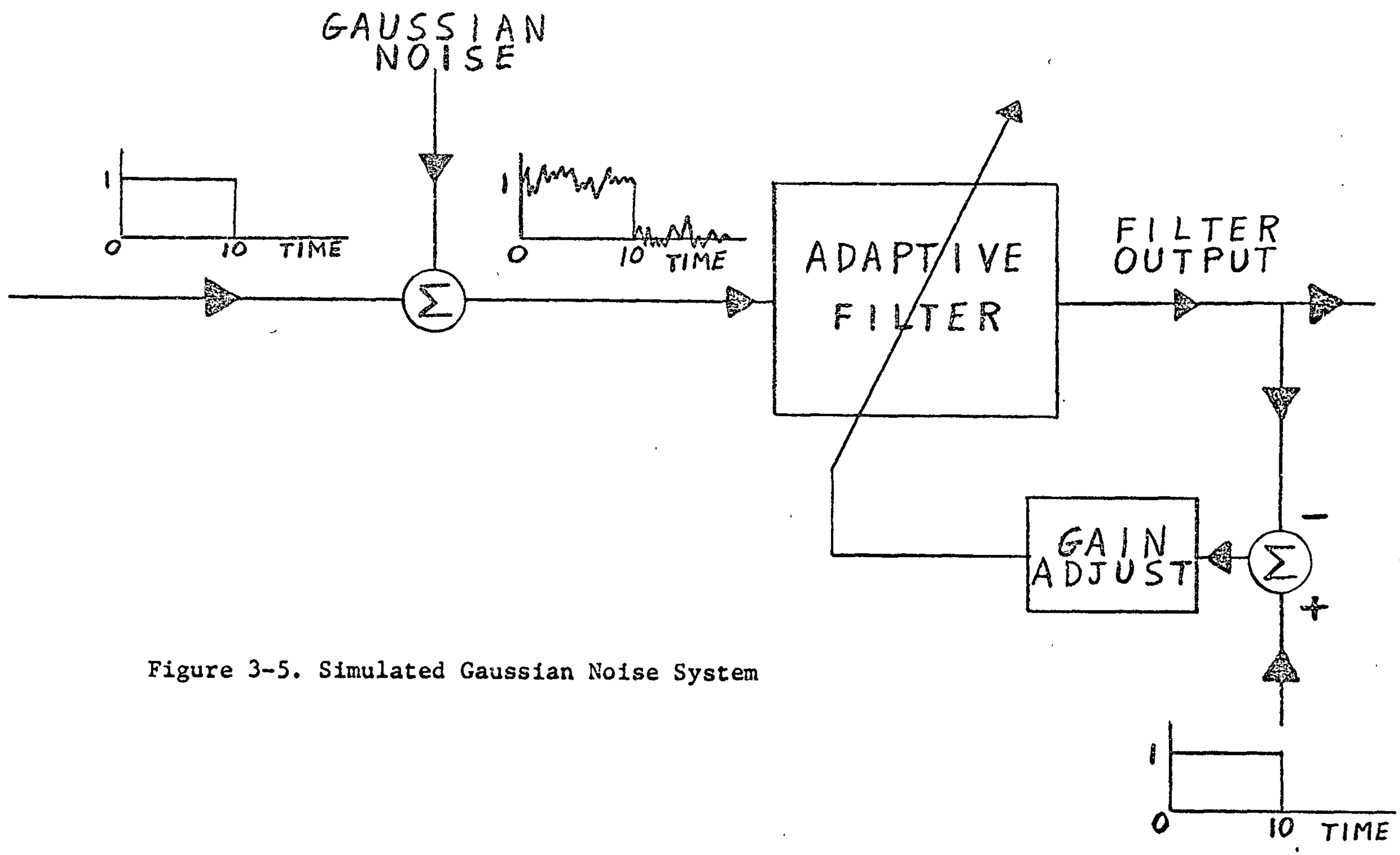

Figure 3-5. Simulated Gaussian Noise System

So this problem has many practical applications and represents one of the most common noise models used today.

CHAPTER IV

COMPUTER SIMULATION

There were two computers available for the simulation, the IBM 7094 and the Digital Equipment Corporation PDP-9. The former was used for the Gaussian Noise problem, since it possessed a random number generator in its external function library.[9] The PDP-9 was used in all other cases.

The programs were written in standard Fortran IV language.

Figures 4-1, 4-2, and 4-3 show the general flow charts describing all of the simulation programs. Figure 4-1 displays the continuous time (one-pass) sampling case in which minimum memory storage of past input samples is required during the adaptation process. Figure 4-2 depicts the repetition case in which a certain number of input samples (25 in this case) are stored during the first pass and then repeated several times at the input during adaptation. This method is especially powerful for time-limited signals or continuous signals which have most of their energy contained in a small time interval.

Figure 4-3 is common to both cases and shows the calculation of the filter output and mean-square error after the adaptation process has been completed. The same input samples were applied here as during the filter learning cycle.

The final form of the computer programs in Fortran IV language as well as a table listing the important program variables and their meaning will be presented in part B of the Appendix.

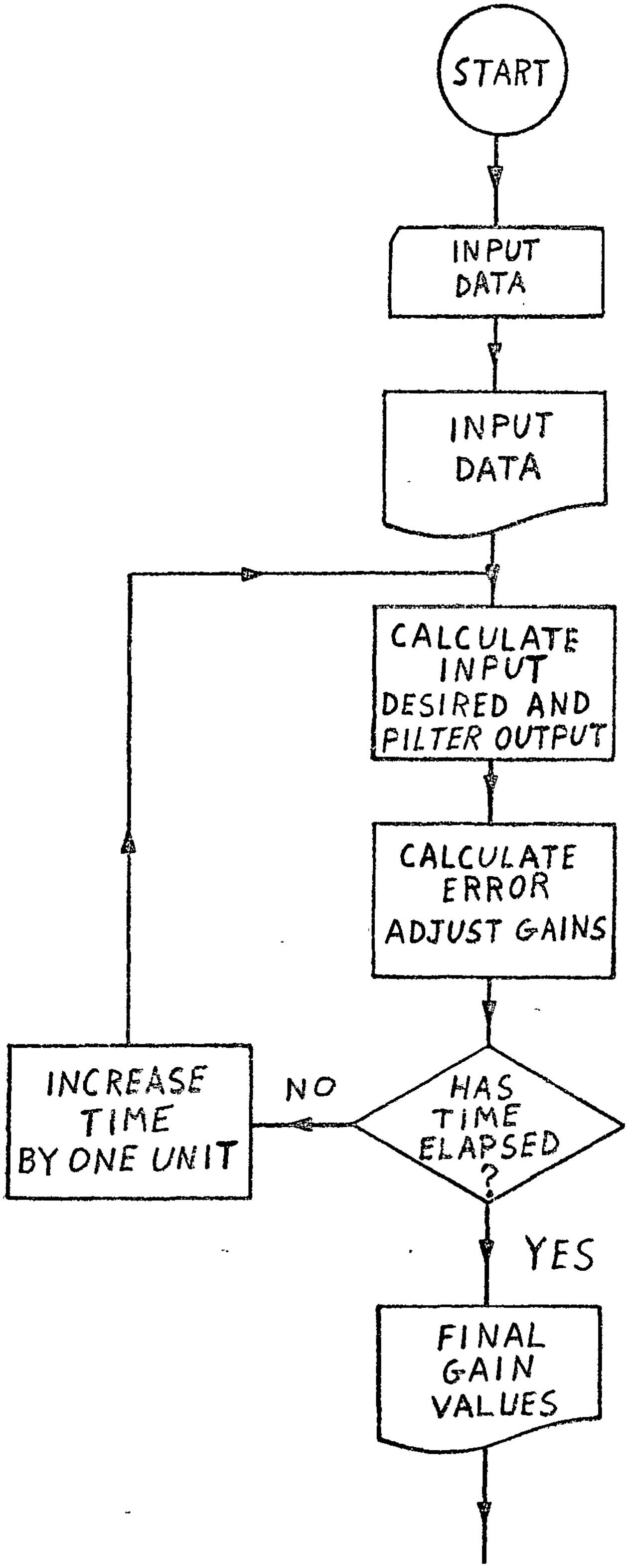

Figure 4-1. Flow Chart for Continuous-Time Sampling Program

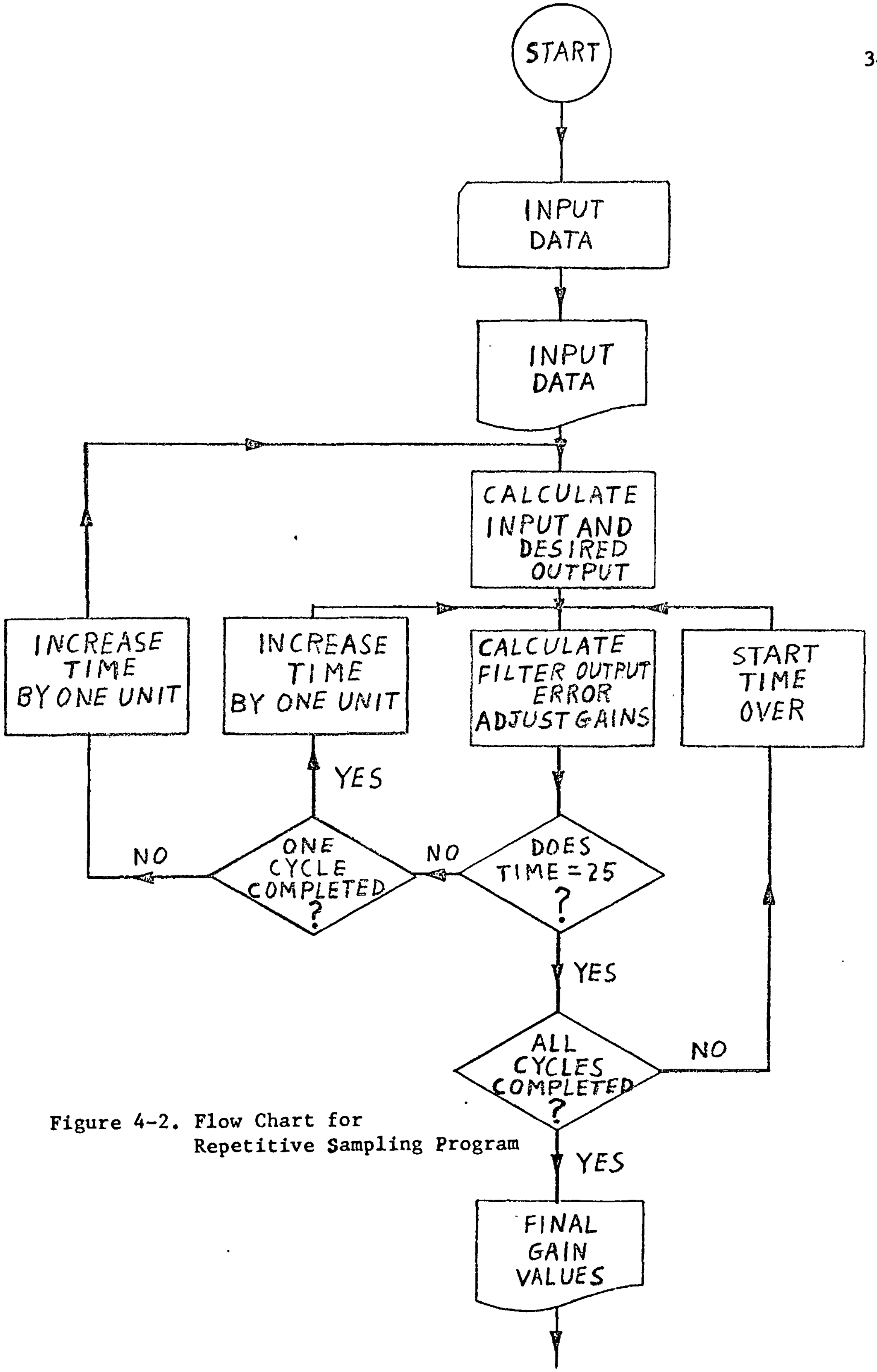

Figure 4-2. Flow Chart for Repetitive Sampling Program

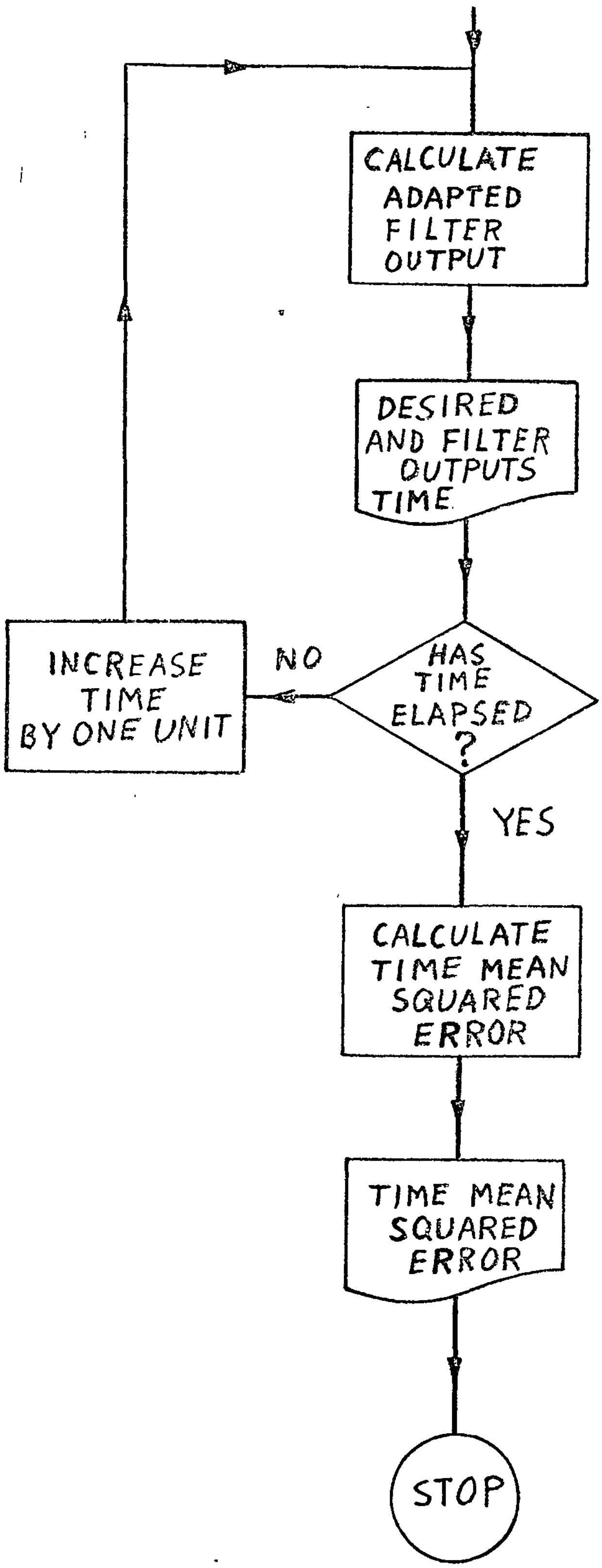

Figure 4-3. Figures 4-1 and 4-2 Continued

CHAPTER V

EXPERIMENTAL RESULTS

This section is devoted to a presentation and explanation of some of the important results obtained from the adaptive filter simulation. Each of the three problems explained in Chapter III will be considered separately.

A. $\frac{\text{SIN X}}{\text{X}}$ Continuous Time Response

All of the simulation work on this phase of the experiment was performed on the DEC PDP-9 Computer. Three parameters (the adaptation constant, the number of adaptation cycles, and the number of weights) were varied over the widest range that computer storage limits and system stability would permit. One hundred output samples from the adjusted filter were observed in every case and a time-averaged square error (T.A.S.E.) was calculated. This error can be expressed by the equation

$$\text{T.A.S.E.} = \sum_{L=1}^{100} (D(L) - Y(L))^2/100 \tag{31}$$

where D(L) = the desired response at time L

Y(L) = the adjusted filter output at time L

The initial weight values were set equal to unity before each computer run.

Table 5-1 shows the stability bounds for the adaptation constant k_s calculated from equations (13) and (29) of Chapter II. During the course of the entire experiment these equations proved quite adequate both for deterministic and statistical inputs although their accuracy was much better for the latter.

TABLE 5-1

Adaptation Constant Stability Bounds
for $\frac{\text{SIN X}}{\text{X}}$ Continuous Time Response

#Feedforward Weights	#Feedback Weights	Bounds
5	0	$0 > k_s > -133.8$
7	0	$0 > k_s > -108.5$
10	0	$0 > k_s > -76.6$
3	2	$0 > k_s > -30.5$
1	4	$0 > k_s > -17.8$
5	5	$0 > k_s > -13.4$
1	9	$0 > k_s > -9.1$

Although much of the theory developed previously was not based on deterministic signals, several of the ideas presented by these statistical expressions are verified in this case.

Figures 5-1 through 5-5 pertain to the feedforward adaptive filter and Figures 5-6 through 5-9 to the feedback system.

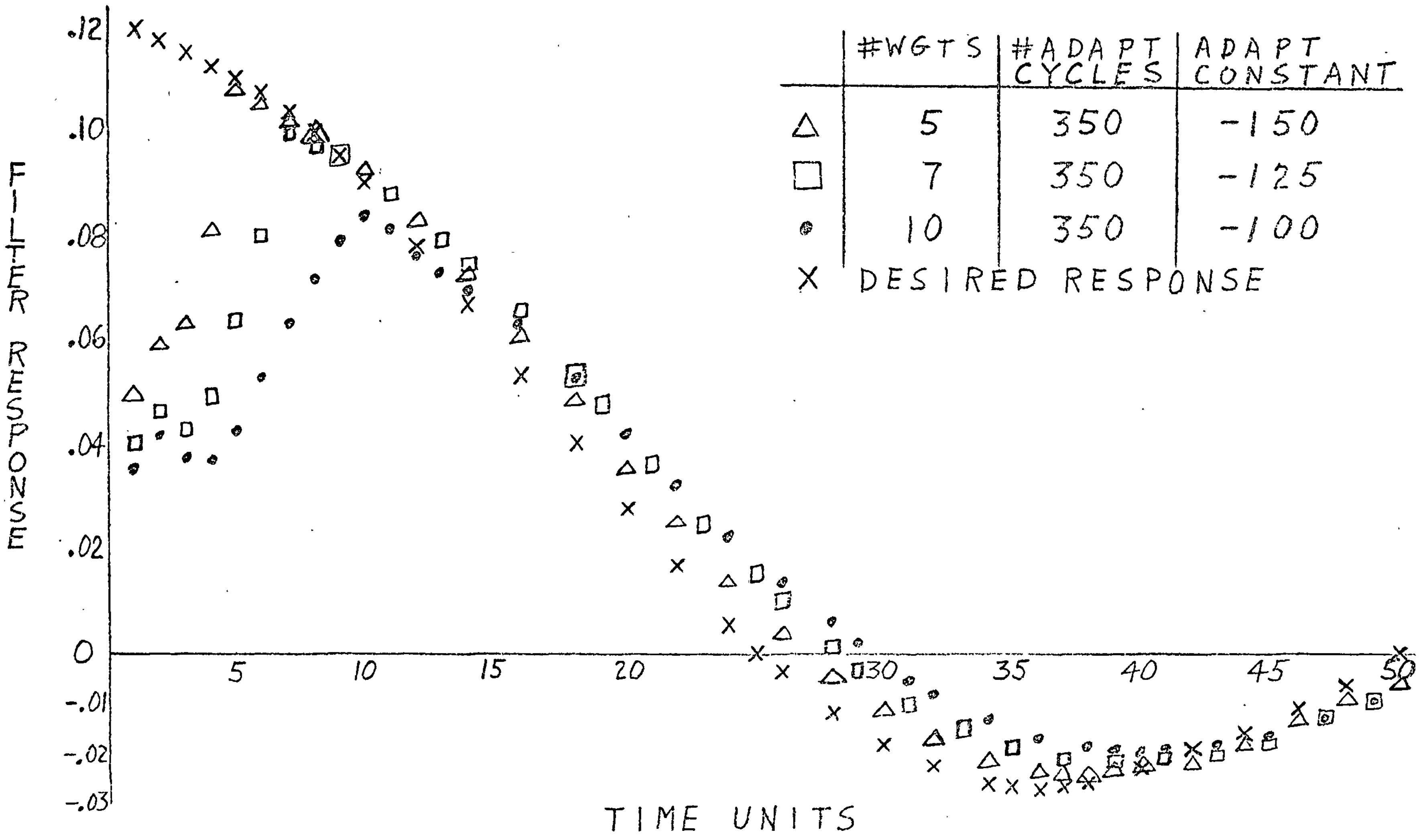

Figure 5-1. Feedforward Adapted Filter Response,Problem A.

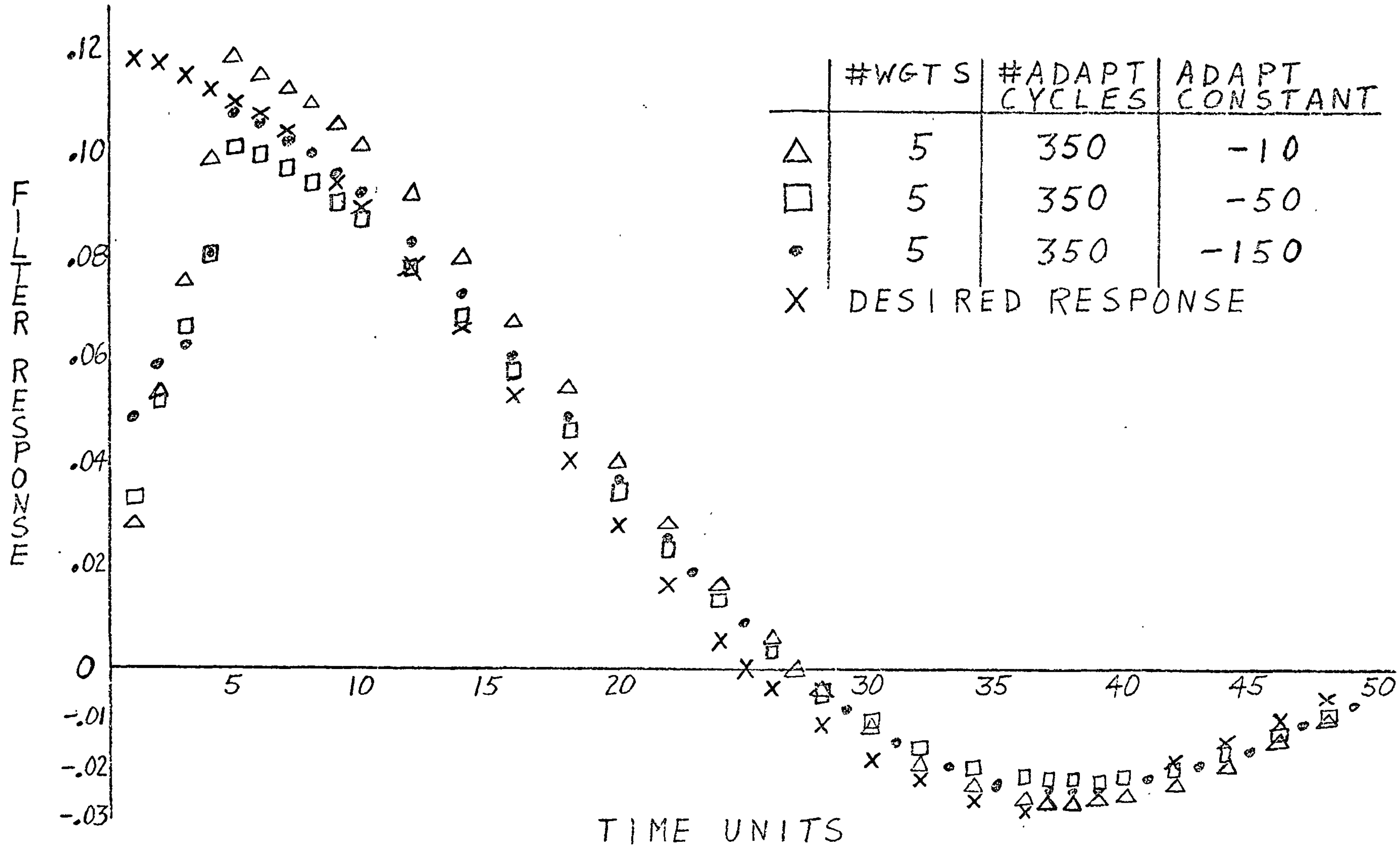

Figure 5-2. Feedforward Adapted Filter Response,Problem A.

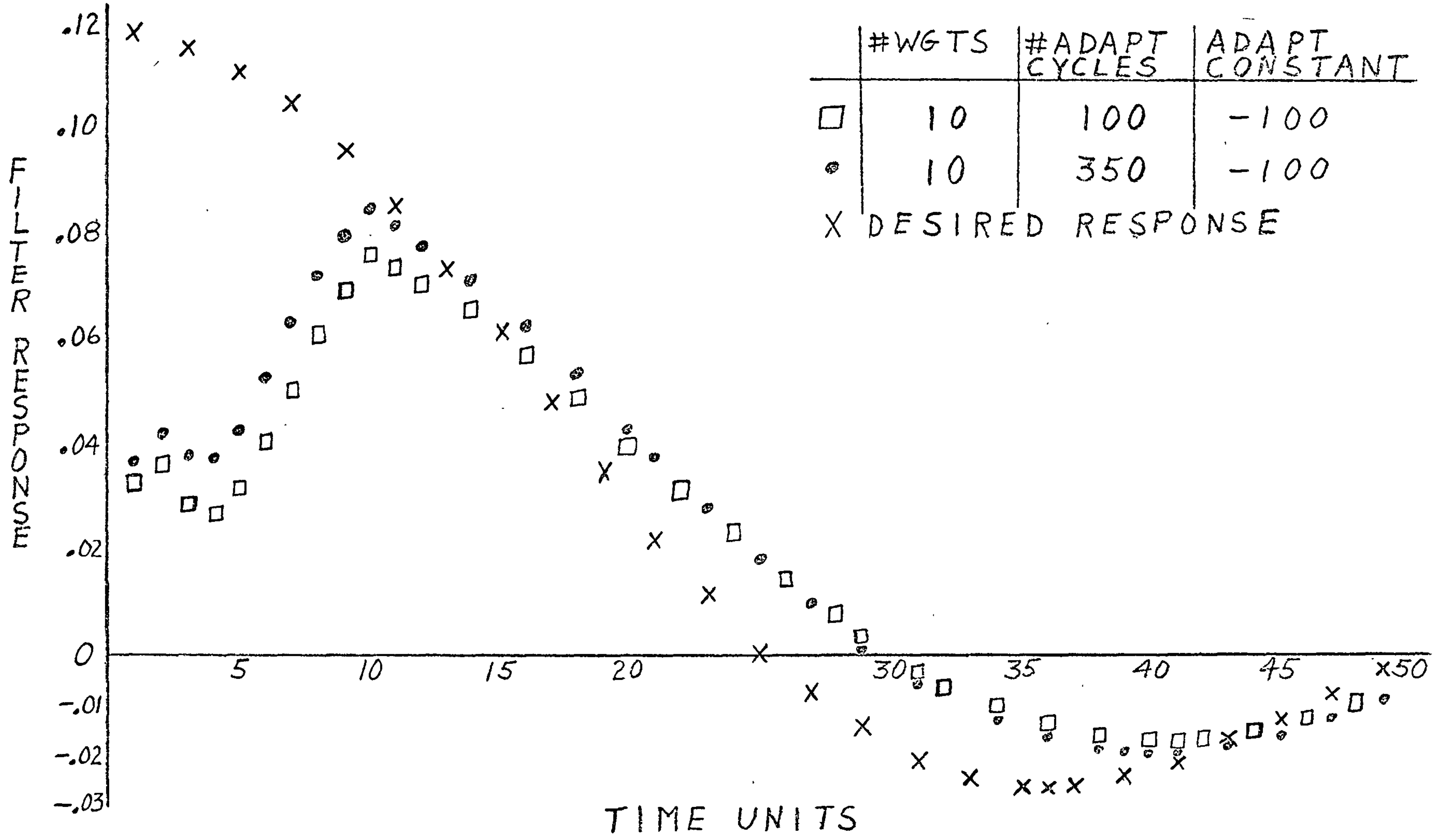

Figure 5-3. Feedforward Adapted Filter Response,Problem A.

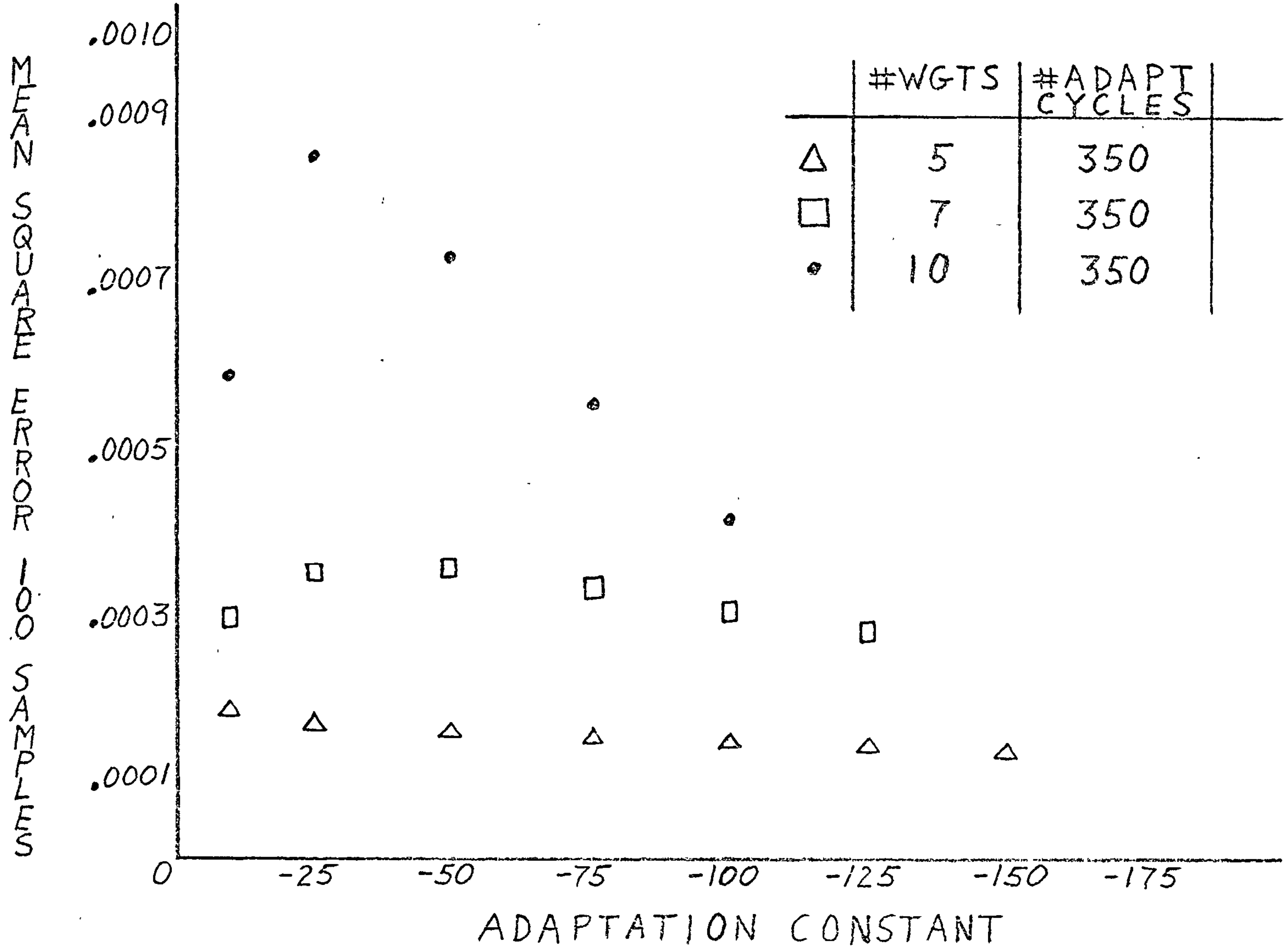

Figure 5-4. Feedforward Filter Output Mean Square Error, Problem A.

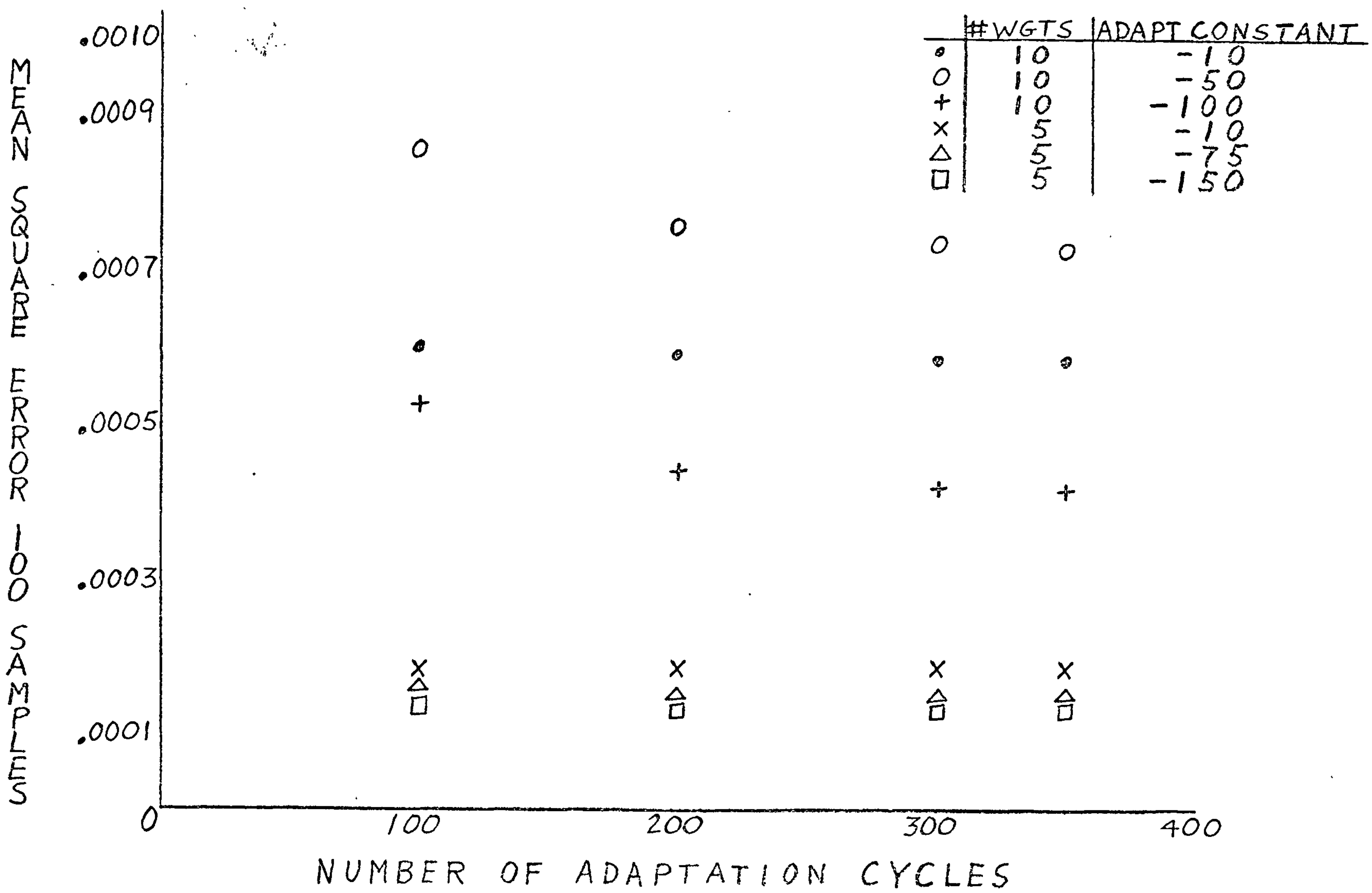

Figure 5-5. Feedforward Filter Output Mean Square Error, Problem A.

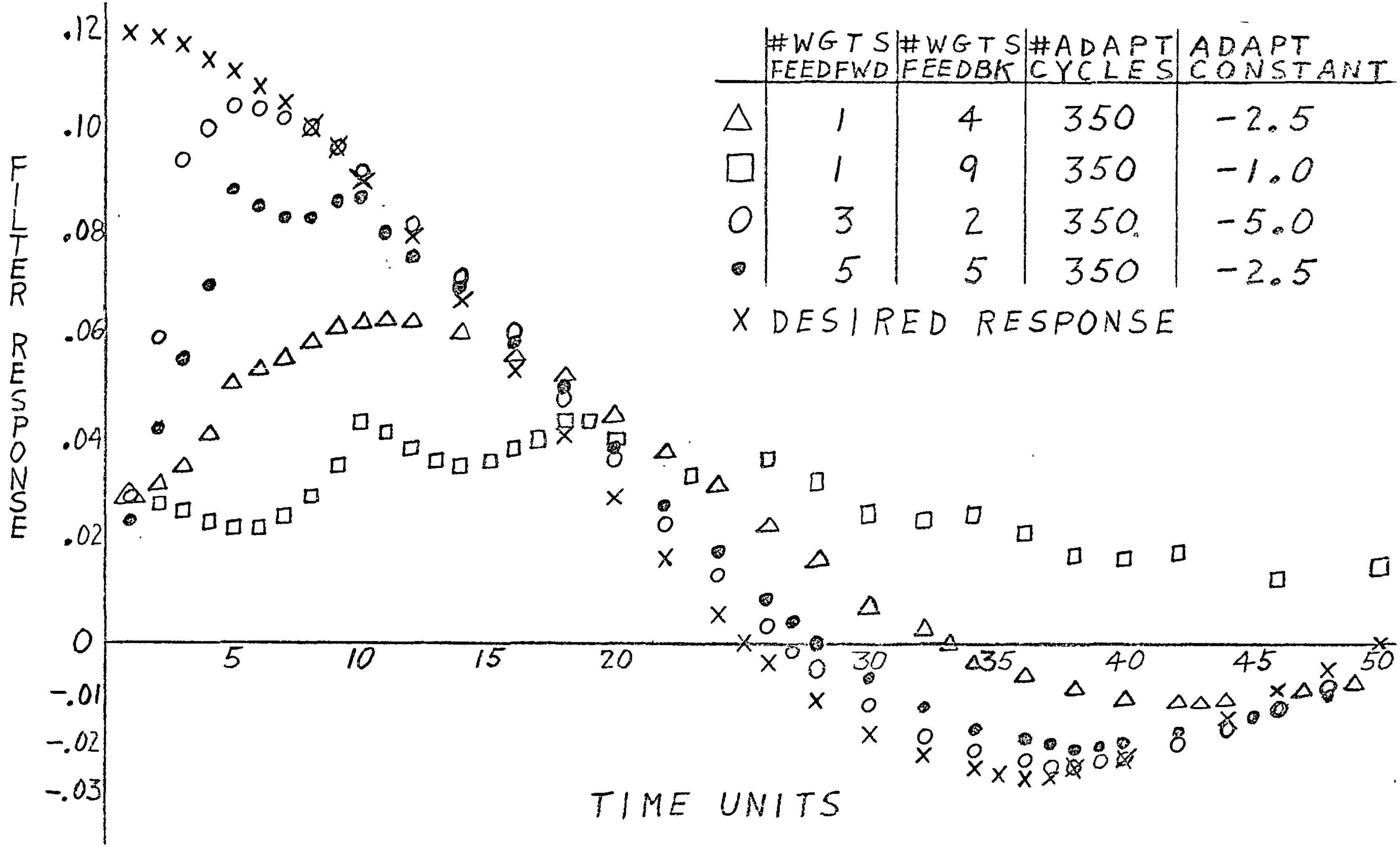

Figure 5-6. Feedback Adapted Filter Response, Problem A.

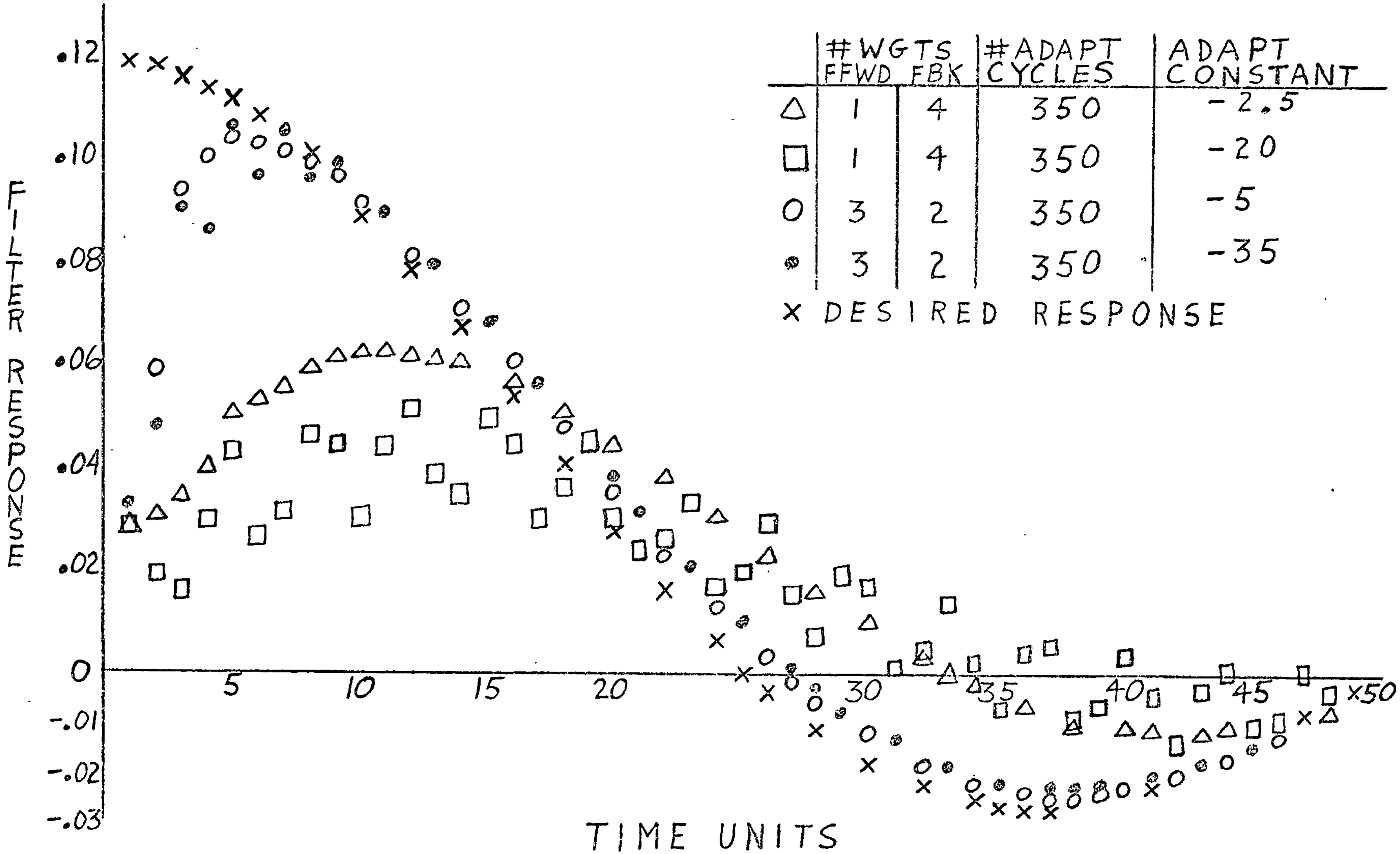

Figure 5-7. Feedback Adapted Filter Response, Problem A.

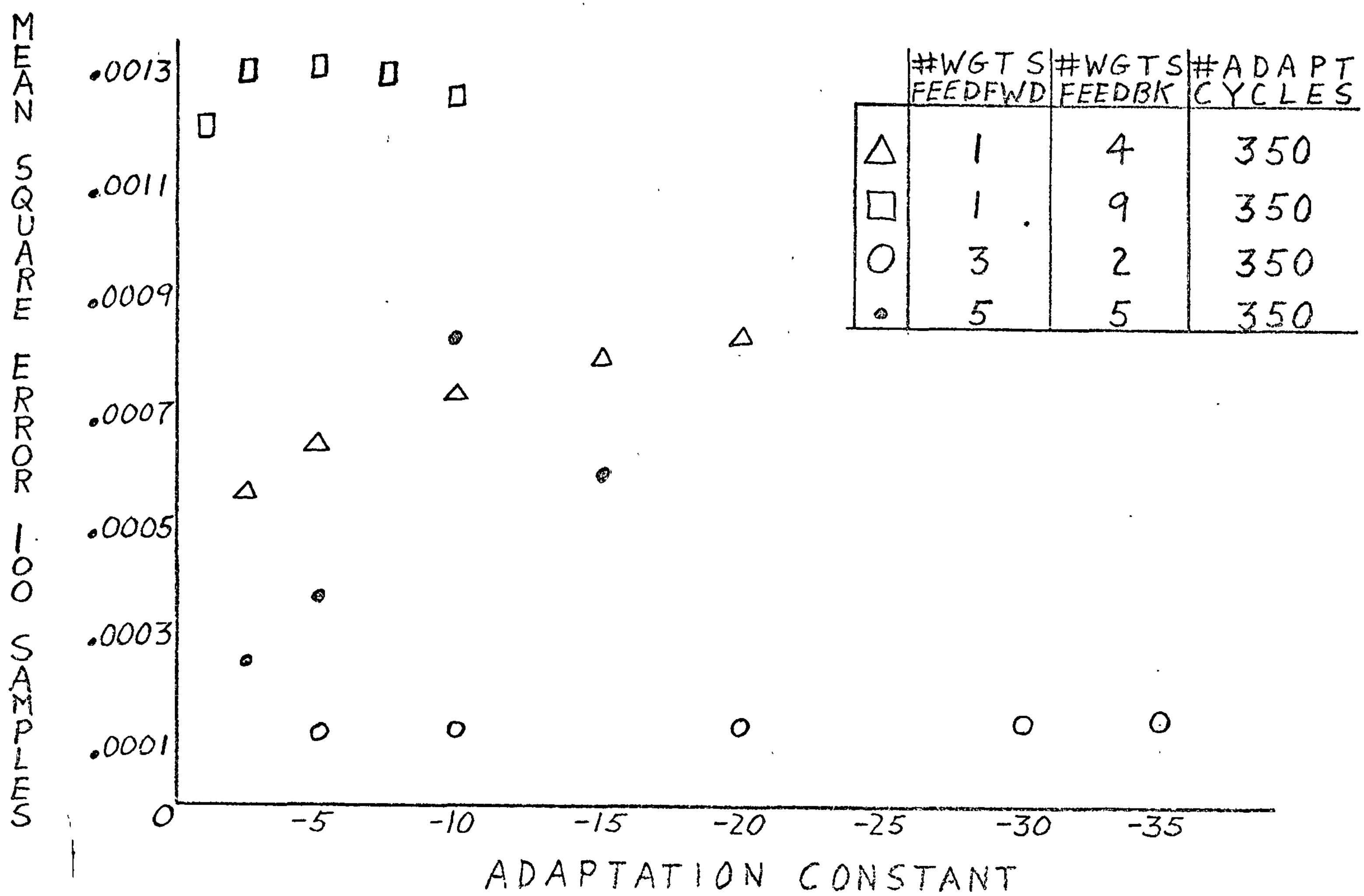

Figure 5-8. Feedback Filter Output Mean Square Error, Problem A.

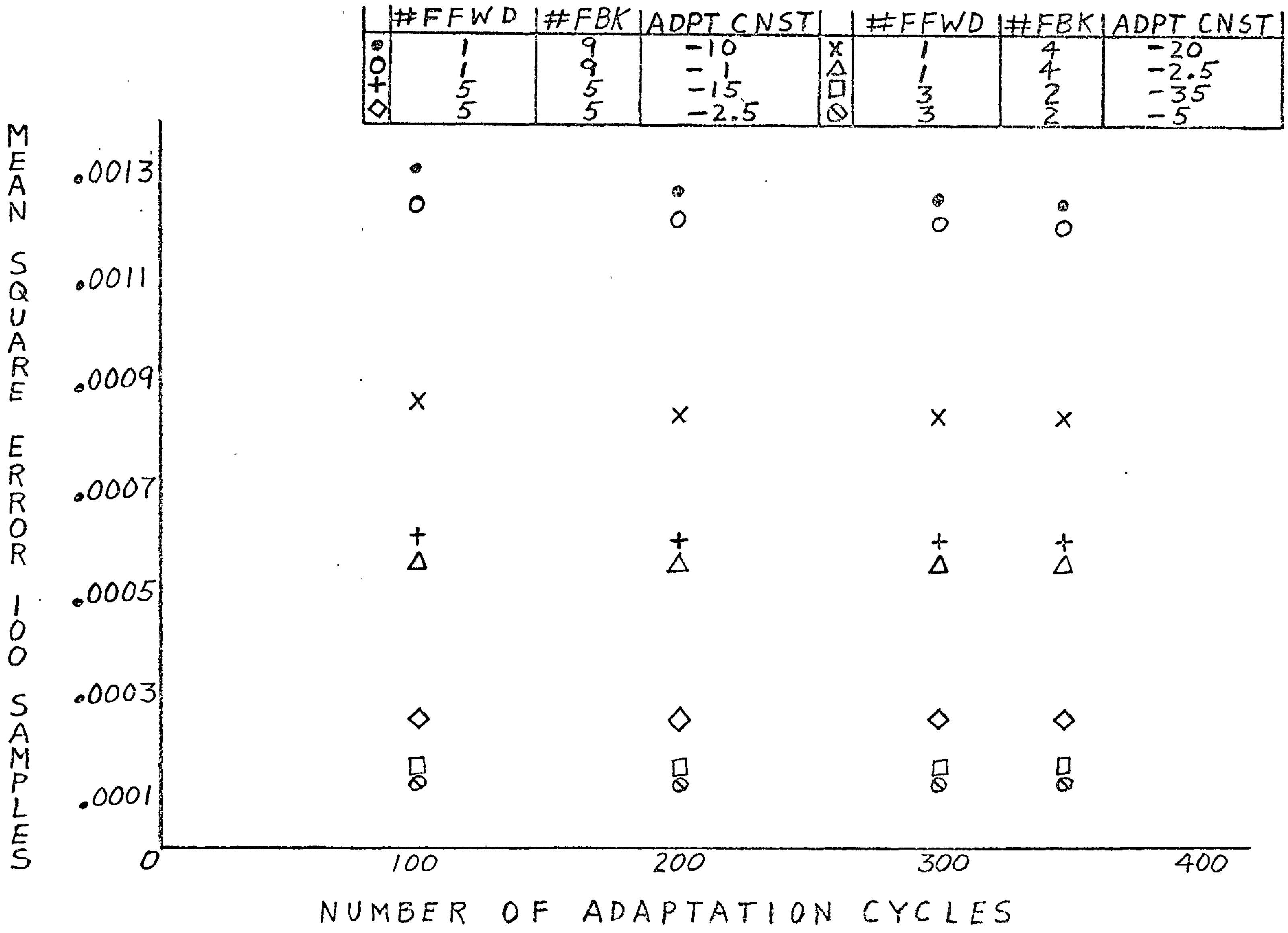

Figure 5-9. Feedbaok Filter Output Mean Square Error,Froblem A.

In Figures 5-1 and 5-6 the adaptation constant in each case corresponds to that for which the T.A.S.E. was minimum for all the experimental values considered.

Two of the important characteristics of all the response curves shown are that there is a noticeable time delay ranging from two to ten time units at the first zero-crossing and there is a "rise time" ranging from ten to fifteen time units before the filter outputs approach the desired response curve.

One explanation for these phenomena is that any filter subjected to a signal with an abrupt amplitude change will usually require a short time to adjust. A typical example is the unit step response of a linear phase filter.[5] There is a certain rise time requirement before the response approaches the final amplitude.

The zero-crossing time delay might be considered a result of this rise time. Since the peak amplitude of the filter response was not reached immediately, its time waveform can be expected to be delayed in relation to the desired response with the difference in later zero-crossing times becoming smaller and smaller.

It is interesting to observe that as the rise time becomes shorter the zero-crossings become closer as expected.

For each type of filter, as the number of weights was increased the response became poorer in shape and the mean-square error became greater. This observation was made for an equal number of adaptation cycles and the best adaptation constant in each case. One major cause for this might be the transients undergone by the weights in adjusting to the error surface minimum. For each type of filter the best adjustment rate decreased as the number of weights increased. From (9) and (30) in Chapter II the transient time constants are inversely proportional to the adaptation constant. With these time constants increasing as the number of weights increases it seems likely that the weight adjustment transients would affect the response of a filter with more weights for a longer time.

Of course, (9) and (30) were developed for statistical inputs but there is no reason to believe that a very similar relation cannot be applied to deterministic input signals.

One observation for which no explanation can be presented is that the optimum adaptation constant for the feedback filter was not the "largest" (most negative). Since gradient measurement noise is not a factor when the signals are deterministic, adapting slowly should not be an advantage. Indeed the "largest" constant was always the best for the feedforward filter, but a relatively small constant always proved superior in the feedback case.

In the final analysis the filter employing both feedforward and feedback weights seemed to perform best in approximating the desired continuous time response. Its time averaged square error was less than that for the pure feedforward and pure feedback cases given equivalent operating conditions, especially when 10 weights were used. However, the feedforward filter operation was very satisfactory for the cases examined. The pure feedback filter performed poorly in all categories.

B. Rectangular Pulse Distorted by RC Channel

The DEC PDP-9 Computer performed all of the simulation work for the RC Channel Distortion Problem. The same parameters mentioned in the continuous time example were varied over the widest ranges possible. Twenty-five output samples from the adjusted filter were observed in every case and a time-averaged square error (T.A.S.E.) was calculated. This error is given by

$$\text{T.A.S.E.} = \sum_{L=1}^{25} (D(L) - Y(L))^2/25 \qquad (32)$$

The initial weight values were set equal to unity before each computer run.

There were two key differences in simulation technique and filter operation between this test problem and the continuous time example. First, since the desired response was a time-limited waveform with all zero values beyond ten time units, it was considered highly impractical to perform continuous time, one-pass sampling in this case. Instead, the first twenty-five input and desired output samples were calculated and repeated over and over again during the adaptation process. In practice this would require the filter to have a finite memory or storage capacity available.

Secondly, the feedback adaptive filter was at a disadvantage in this case because the twenty-five repeated desired output samples consisted of several zero-values. The first ten desired response samples were equal to unity, but the last fifteen were zero. Therefore, the feedback weights were only being adjusted during two-fifths of the total adaptation time. In order to make an objective comparison of the three different types of filters in performing the pulse reconstruction, an adaptation cycle scaling process was used. The final "equalizer" in all cases was the total number of adjustments and not the total number of complete cycles. The following simple equation was used.

$$\text{SCLDCYC}\ (\text{NFWD} + (\text{NFBK} \times 2/5)) = \text{NADJ} \tag{33}$$

where SCLDCYC = number of adaptation cycles after scaling
NFWD = number of feedforward weights
NFBK = number of feedback weights
NADJ = total number of weight adjustments desired

In order to use the repeat process the number of scaled cycles was "rounded-off" to the nearest multiple of twenty-five.

Table 5-2 shows some of the values used in this phase of the experiment.

Table 5-3 shows the stability bounds for the adaptation constant k_s calculated from equations (13) and (29) of Chapter II.

Figures 5-10 through 5-14 pertain to the feedforward adaptive filter and figures 5-15 through 5-19 to the feedback system.

In Figures 5-10 and 5-15 the adaptation constant in each case corresponds to that for which the T.A.S.E. was minimum for all the experimental values considered.

TABLE 5-2

Adjustment Equalization

NFWD	NFBK	SCLDCYC (Rounded)	NADJ
3	2	125	500
3	2	250	1000
3	2	400	1500
3	2	450	1750
1	4	200	500
1	4	375	1000
1	4	575	1500
1	4	675	1750
5	5	150	1000
5	5	275	2000
5	5	425	3000
5	5	500	3500
1	9	225	1000
1	9	425	2000
1	9	650	3000
1	9	750	3500

TABLE 5-3

Adaptation Constant Stability Bounds
for Channel Distortion Problem

# Feedforward Weights	# Feedback Weights	Bounds
5	0	$0 > k_s > -.606$
7	0	$0 > k_s > -.471$
10	0	$0 > k_s > -.376$
3	2	$0 > k_s > -.325$
1	4	$0 > k_s > -.227$
5	5	$0 > k_s > -.151$
1	9	$0 > k_s > -.106$

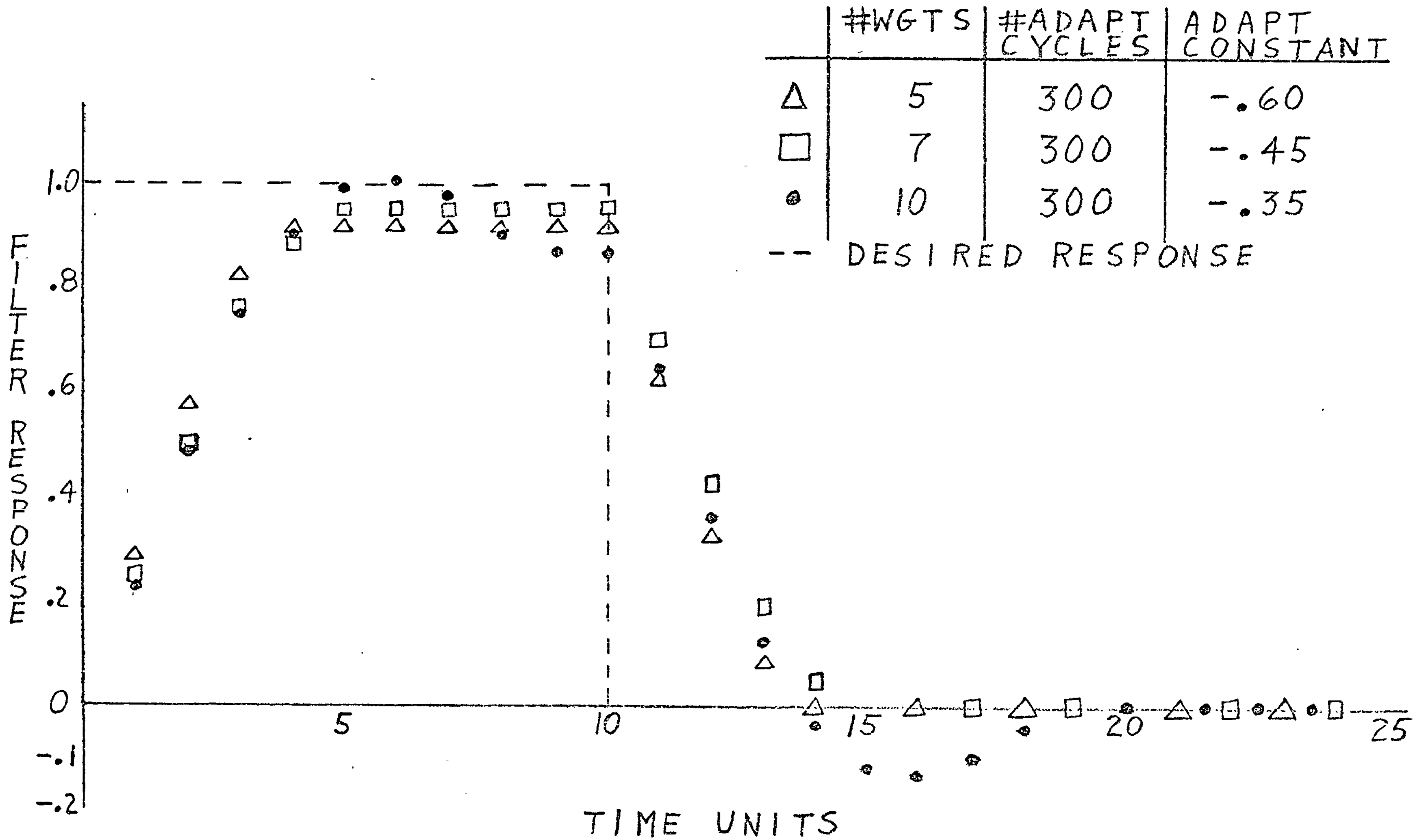

Figure 5-10. Feedforward Adapted Filter Response,Problem B.

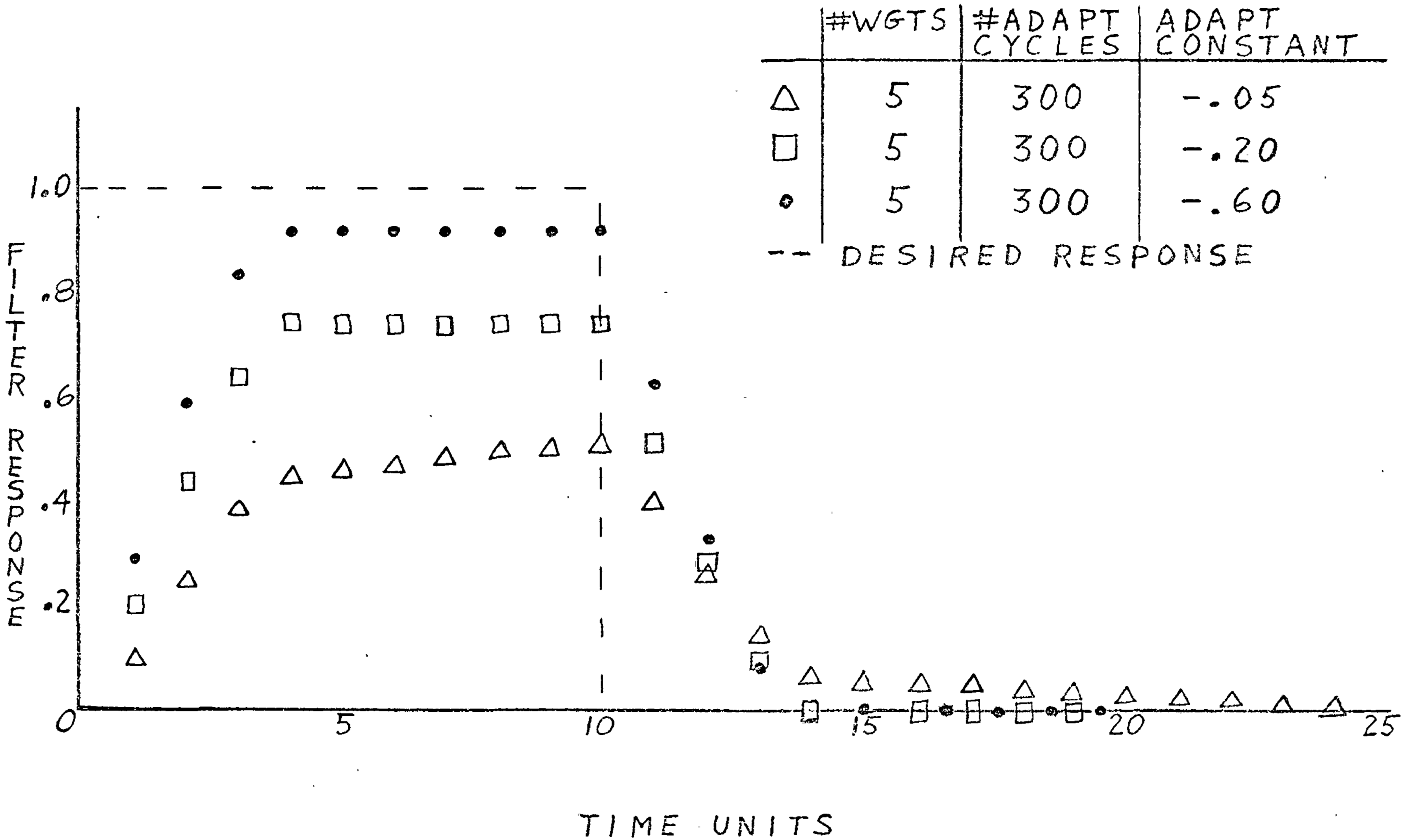

Figure 5-11. Feedforward Adapted Filter Responso,Froblem B.

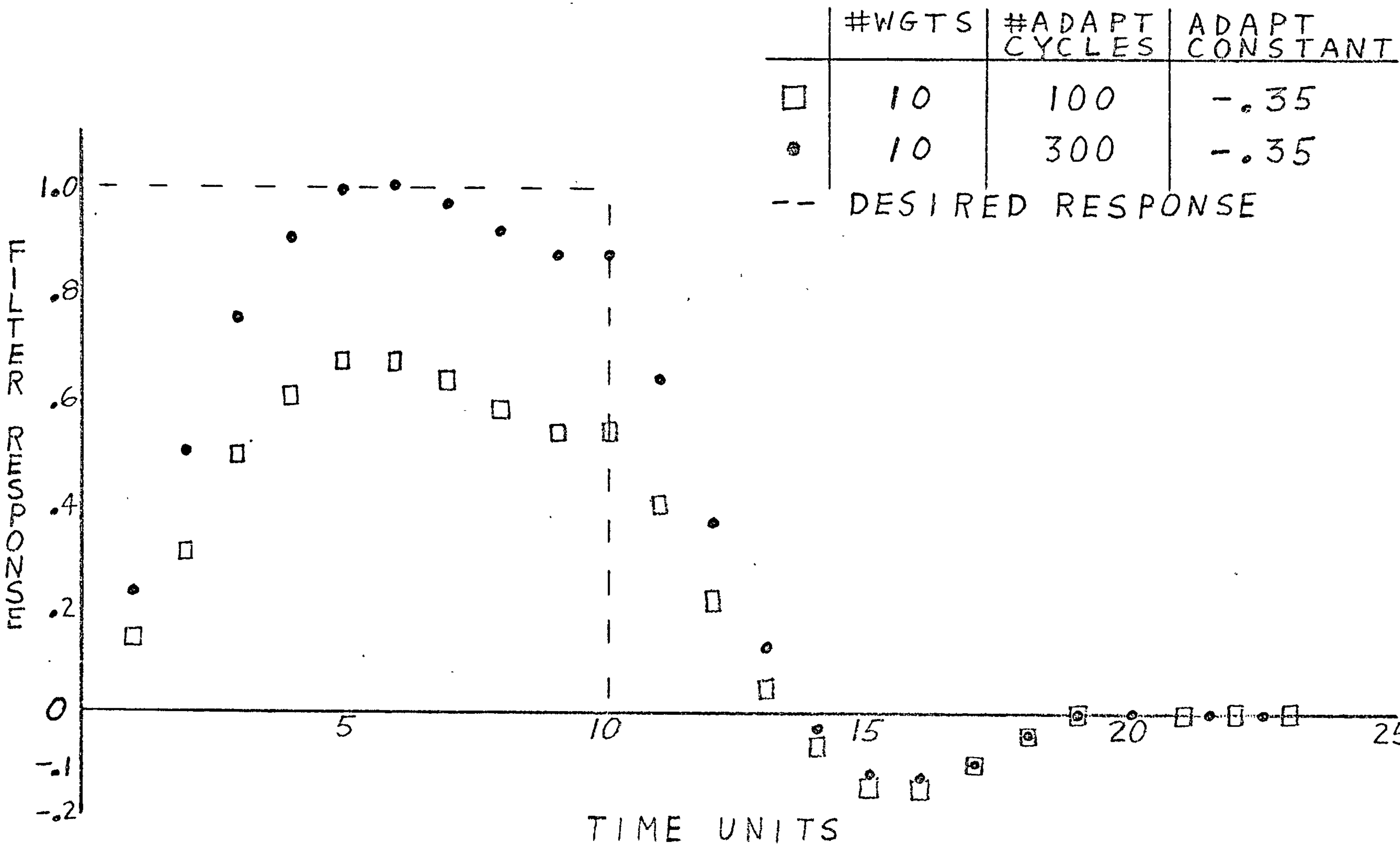

Figure 5-12. Feedforward Adapted Filter Response,Problem B.

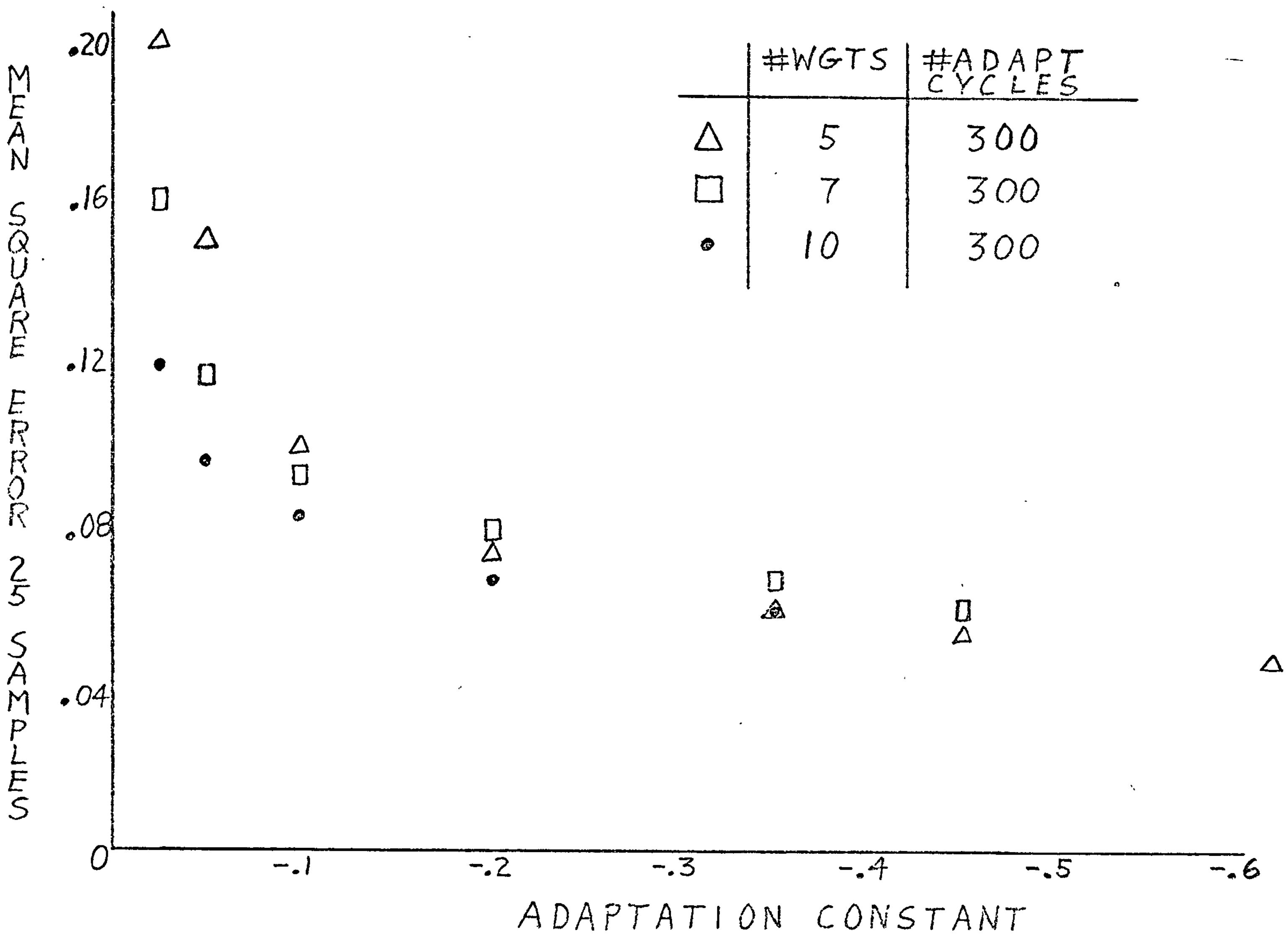

Figure 5-13. Feedforward Filter Output Mean Square Error,Problem B.

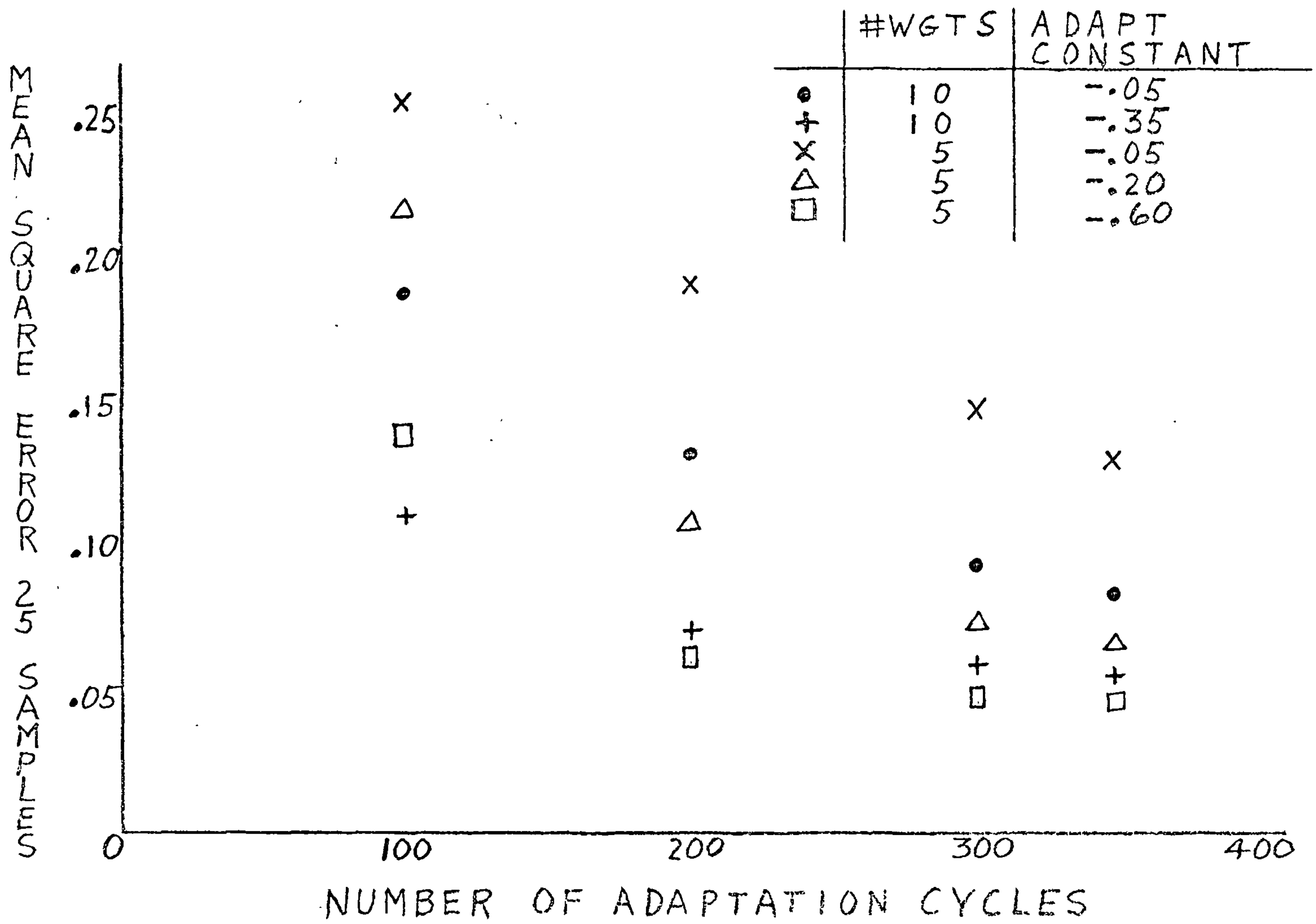

Figure 5-14. Feedforward Filter Output Mean Square Error,Problem B.

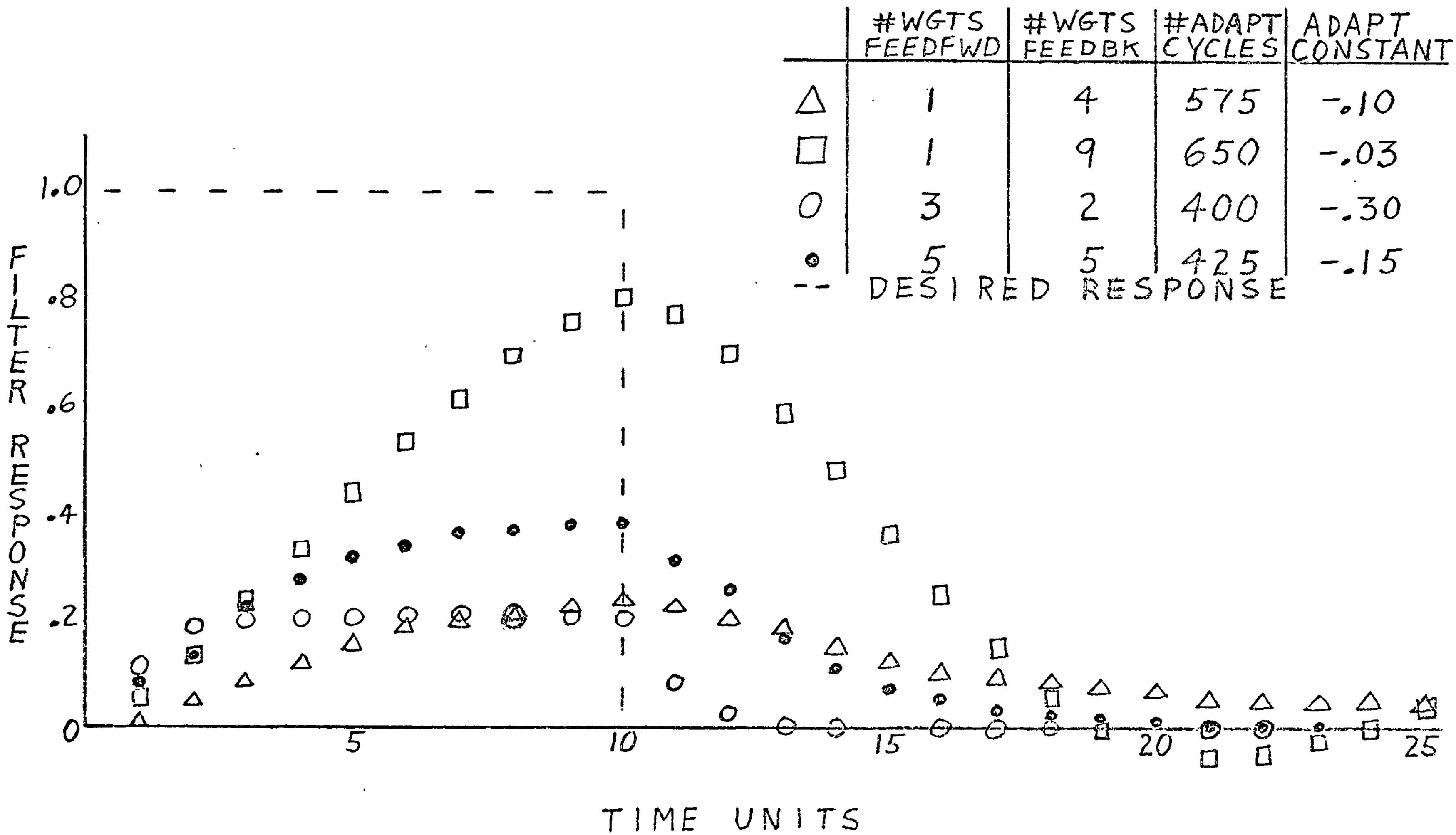

Figure 5-15. Feedback Adapted Filter Response,Problem B.

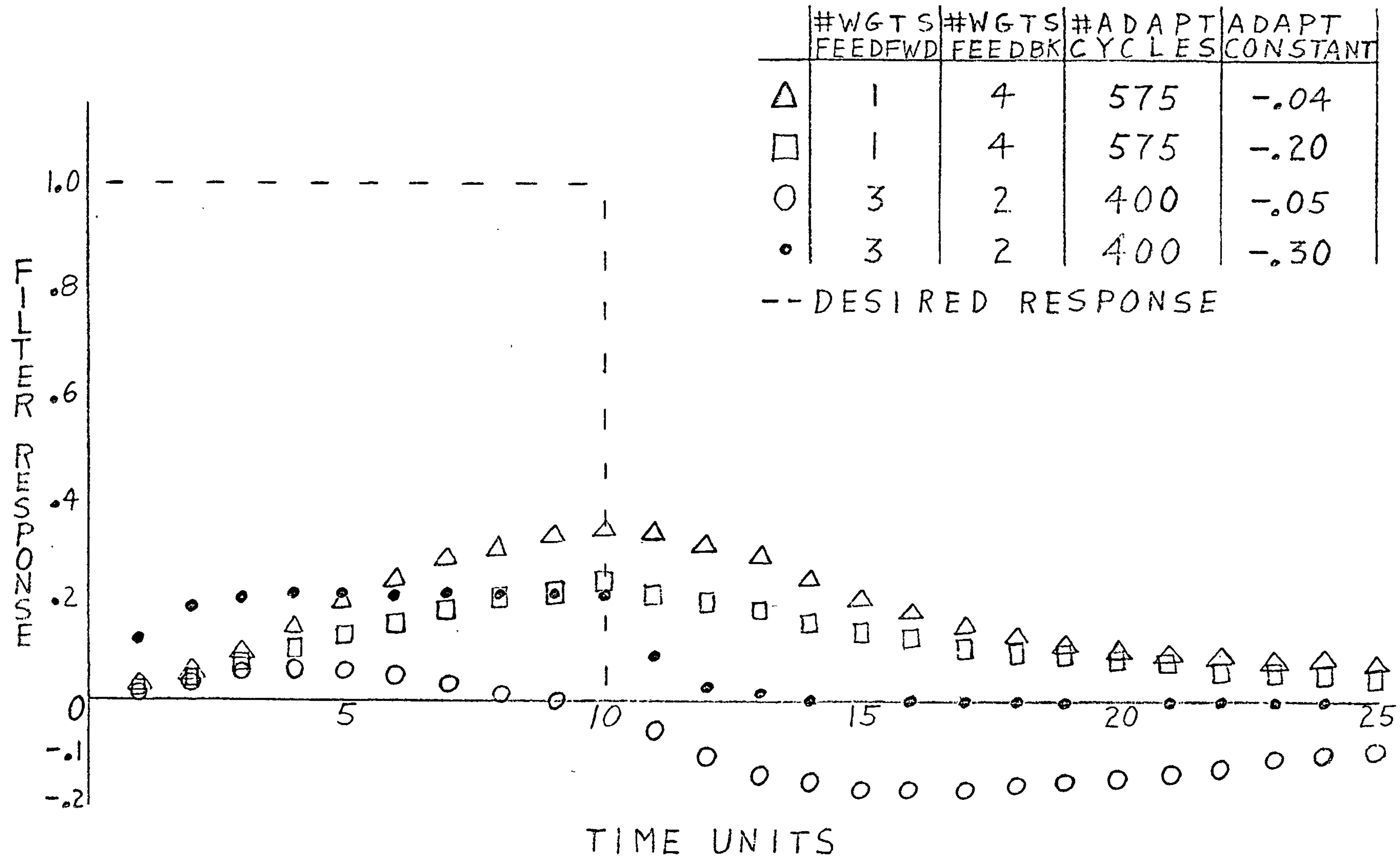

Figure 5-16. Feedback Adapted Filter Response,Problem B.

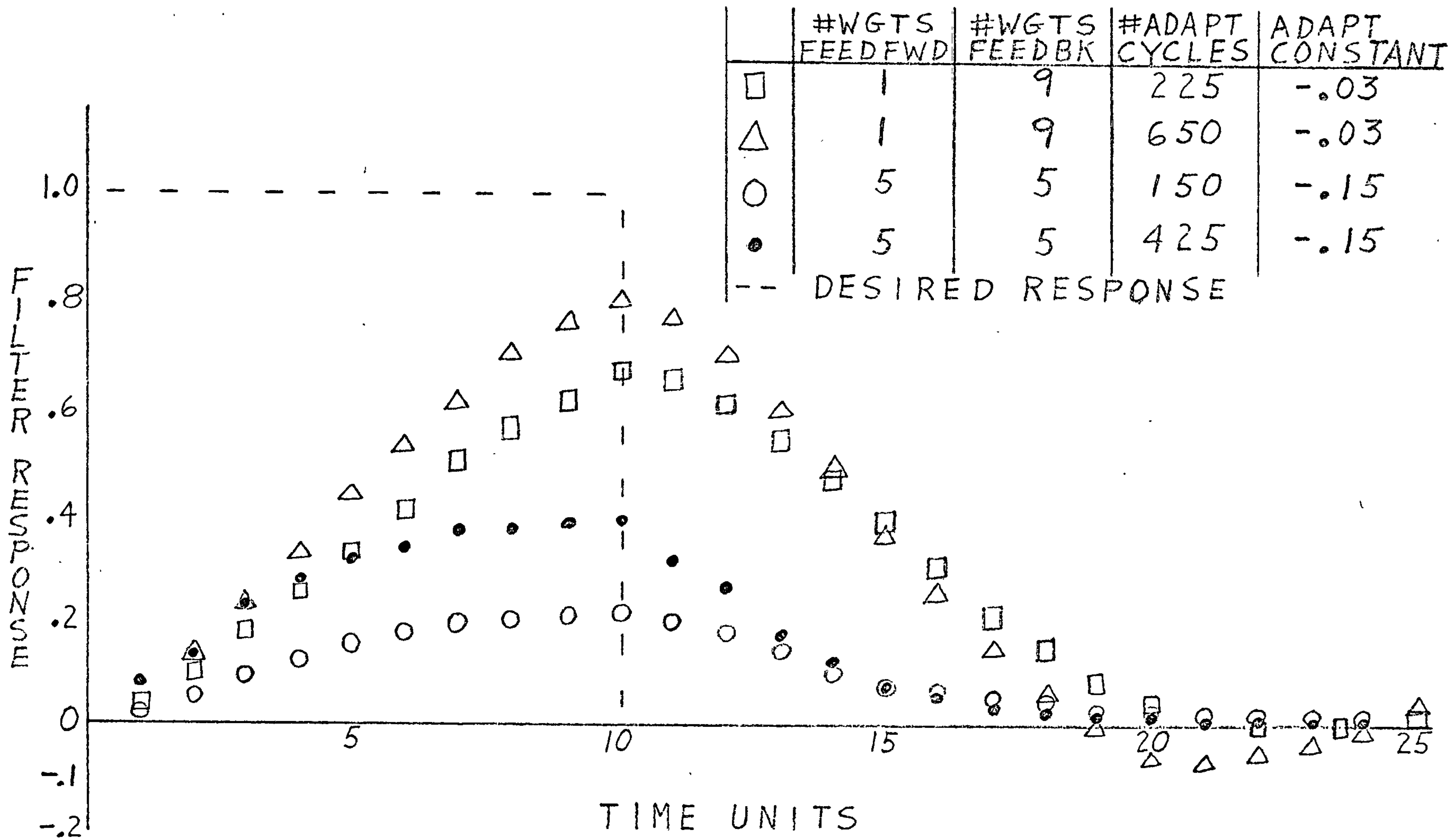

Figure 5-17. Feedback Adapted Filter Response,Problem B.

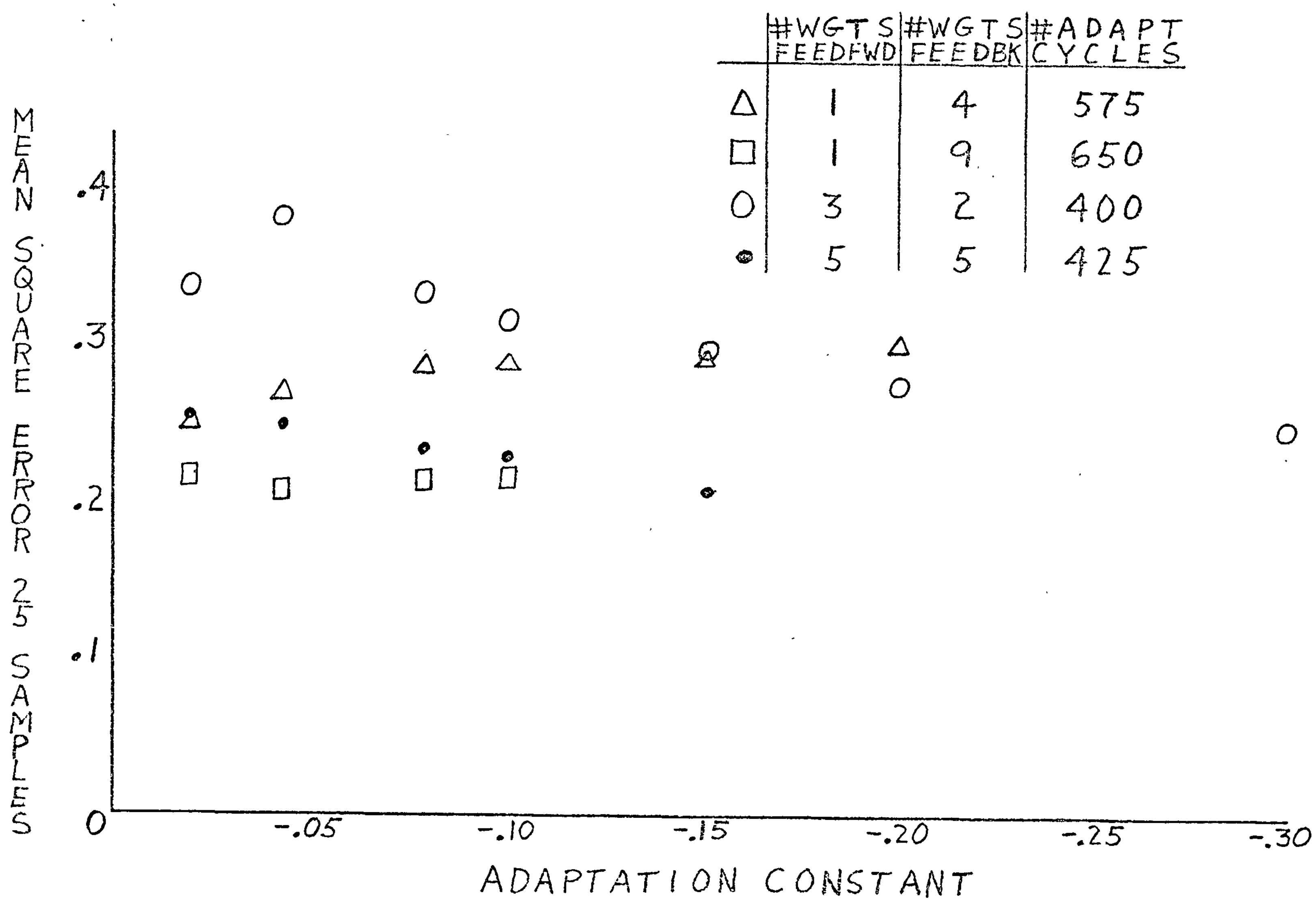

Figure 5-18. Feedback Filter Output Mean Square Error,Problem B.

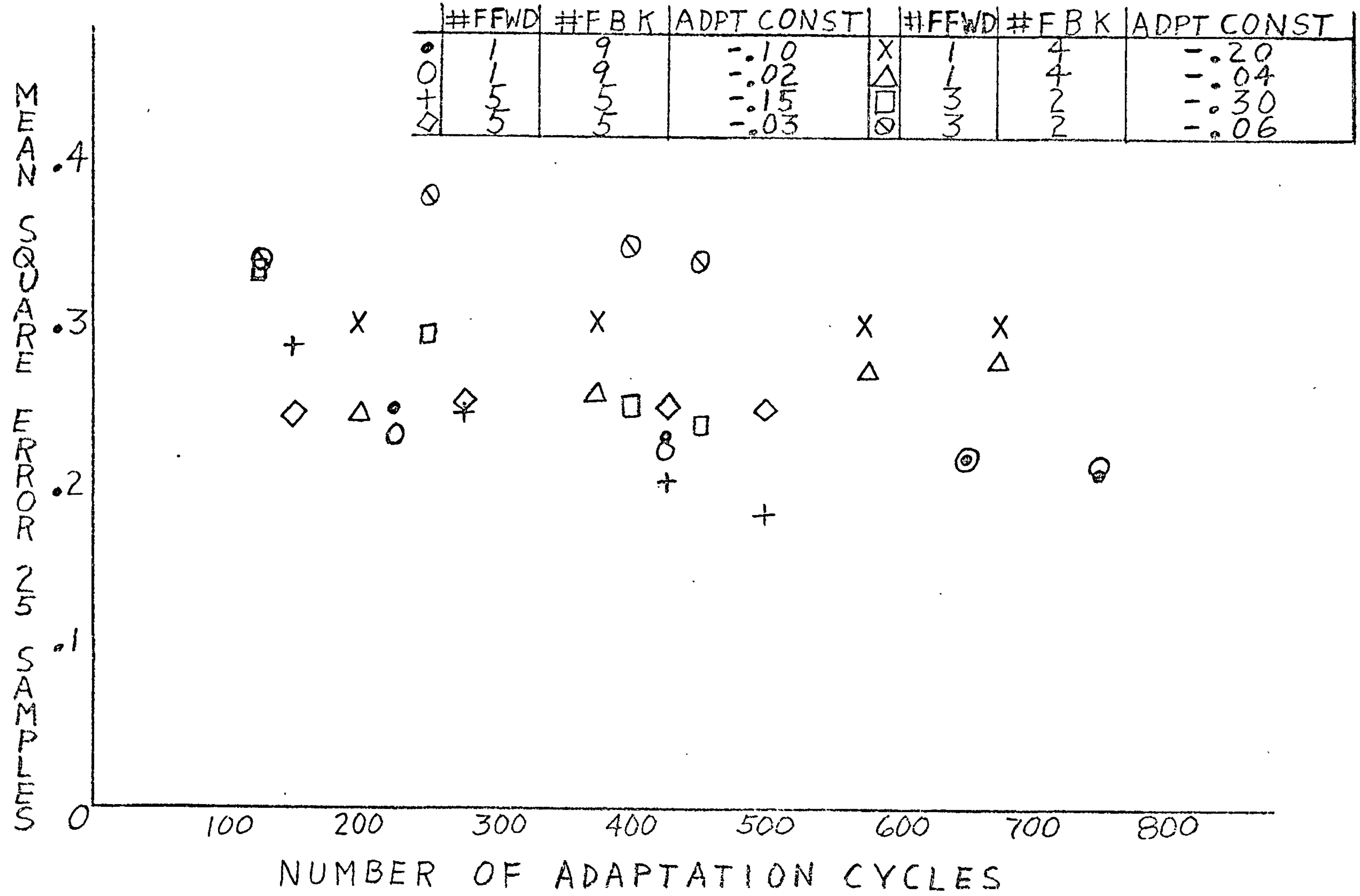

Figure 5-19. Feedback Filter Output Mean Square Error,Problem B.

The desired square pulse has two abrupt changes in amplitude. These occur at time zero and at ten time units. As discussed in part A of this chapter, the filter response exhibited rise and fall times due to its inability to adjust immediately. In the feedforward case these rise and fall times increased as the number of weights increased. Although well-defined "flat" portions for the filter response were not evident in all of the feedback cases, it appears that the time required to rise to the maximum response amplitude and fall to the zero value again also increased with the total number of weights.

For an equal number of weights the filter employing both feed-forward and feedback weights had the smallest rise and fall times. The pure feedforward filter, however, had very comparable times, especially for ten weights. The pure feedback filter had much longer rise and fall times than either of the other types.

For each type of filter, as the number of weights increased the response became generally poorer in shape for the same total number of weight adjustments and the best adaptation constant. As surmised previously, the fact that the transients undergone by the weights in adjusting to the error surface minimum have longer time constants with increasing number of weights is probably the major cause for this phenomenon.

The optimal adjustment constant for the feedforward filter was again the "largest" (most negative). As before, a relatively small constant always proved superior in the pure feedback case. However, this time the "largest" adaptation constant worked best for the filter using both feedforward and feedback weights. This is in direct contrast to the continuous time problem of part A and remains without a reasonable explanation.

The mean square error decreased as the number of adaptation cycles increased and as the adaptation constant increased in all cases in which only feedforward weights were used. This is not an unexpected result when deterministic signals are involved and the method of steepest descent is being used as the adjustment algorithm.

In general, this same observation was true for the mean square error of the filter using both feedforward and feedback weights.

No such overall remarks can be made about the error in the pure feedback case. There were no obvious trends which could be established.

There are three areas which must be considered when comparing the overall performance of the various types of adaptive filters for this particular problem. These are response shape, response amplitude, and adjustment time.

The combination feedforward and feedback adaptive filter had a slightly better response shape than that of the pure feedforward filter in most cases. However, the amplitude was much smaller than the desired response.

The feedforward filter response had an amplitude which was very close to the desired waveform in its best case.

The pure feedback filter was inferior in both categories.

But the feedforward filter required several adaptation cycles less than the combination filter to produce its reasonably good response. This is due to the time-limited nature of the desired response.

Unless a slight improvement in response shape outweighs all other considerations, the feedforward filter is desirable in this case.

C. Square Pulse Distorted by Additive White Gaussian Noise

The simulation for the Gaussian Noise problem was performed entirely on the IBM 7094 computer. The same three filter parameters previously mentioned were varied over the widest ranges possible. Twenty-five output samples from the adjusted filter were observed in every case and the time-averaged square error given by (32) was calculated.

The initial weight values were set equal to unity before each computer run.

Each input sample consisted of the value of the rectangular pulse at that time plus a random noise sample from an approximate normal distribution. The generation of this noise process is discussed in the Appendix.

The desired output was the "clean" rectangular pulse with no noise present.

Several values of signal to noise ratio (S/N) were tested for every case to determine the effect upon the filter performance. The statistical effect of multiplying each noise sample by a constant k is to scale the variance of the noise process by k^2. Increasing the variance is equivalent to reducing the S/N.

Since the desired response was time-limited the filters employing feedback were at some disadvantage. In order to make the performance comparisons for all types of filters more valid 600 adaptation cycles were used for the feedback filters and only 400 for the feedforward filter. This tended to equalize the total number of weight adjustments in all cases. The compensation was not as thorough as that used in the RC channel distortion problem but it was effective. However, one minor difficulty was encountered in this method. The computer program was written so that the input noise samples for the adjusted filter were identical for all cases in which the number of adaptation cycles was the same. Therefore, the input waveform was not identical in the feed-forward and feedback cases. Statistically this made no difference, but experimentally some problems could have been encountered in making performance comparisons for a relatively short adaptation time.

Table 5-4 shows the various signal to noise ratios used in the experiment and the approximate distribution of the normal noise process.

TABLE 5-4

Experimental Signal to Noise Ratios and Noise Distributions

S/N		Approx. Normal Noise Distribution	
Absolute	DB	Mean	Variance
400.0	26.0	0	1
100.0	20.0	0	4
44.4	16.5	0	9
25.0	14.0	0	16
16.0	12.0	0	25
11.1	10.5	0	36

The time-limited nature of the signals made it advantageous to use the method described in the RC channel problem. The first twenty-five signal values (10 unity values, and 15 zero values) were repeated at the input over and over again. However, the additive Gaussian Noise Samples were different for each time unit so that the repetition procedure did not affect the continuous time nature of the noise process.

Table 5-5 shows the stability bounds for the adaptation constant k_s calculated from equations (13) and (29) of Chapter II. Since the noise process had zero mean it was not a factor in these calculations.

TABLE 5-5

Adaptation Constant Stability Bounds
for Gaussian Noise Problem

# Feedforward Weights	# Feedback Weights	Bounds
5	0	$0 > k_s > -.00050$
10	0	$0 > k_s > -.00025$
13	0	$0 > k_s > -.00019$
3	2	$0 > k_s > -.00050$
1	4	$0 > k_s > -.00050$
5	5	$0 > k_s > -.00025$
1	9	$0 > k_s > -.00025$

Figures 5-20 through 5-27 pertain to the feedforward adaptive filter and Figures 5-28 through 5-35 to the feedback system.

In Figures 5-20, 5-21, 5-28, and 5-29 the adaptation constant in each case corresponds to that for which the time-averaged square error was minimum for all experimental values considered.

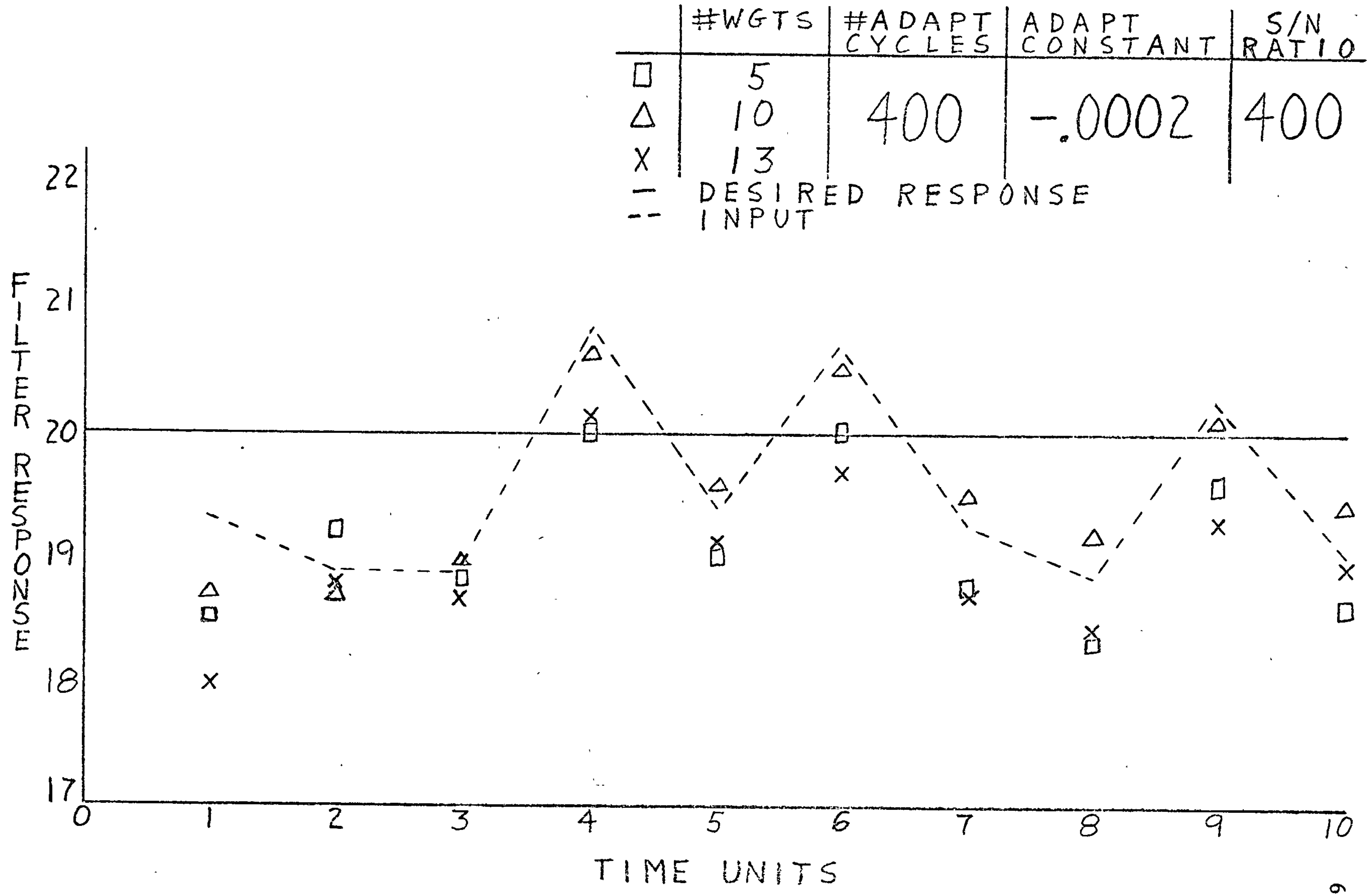

Figure 5-20. Feedforward Adapted Filter Response (High S/N), Problem C.

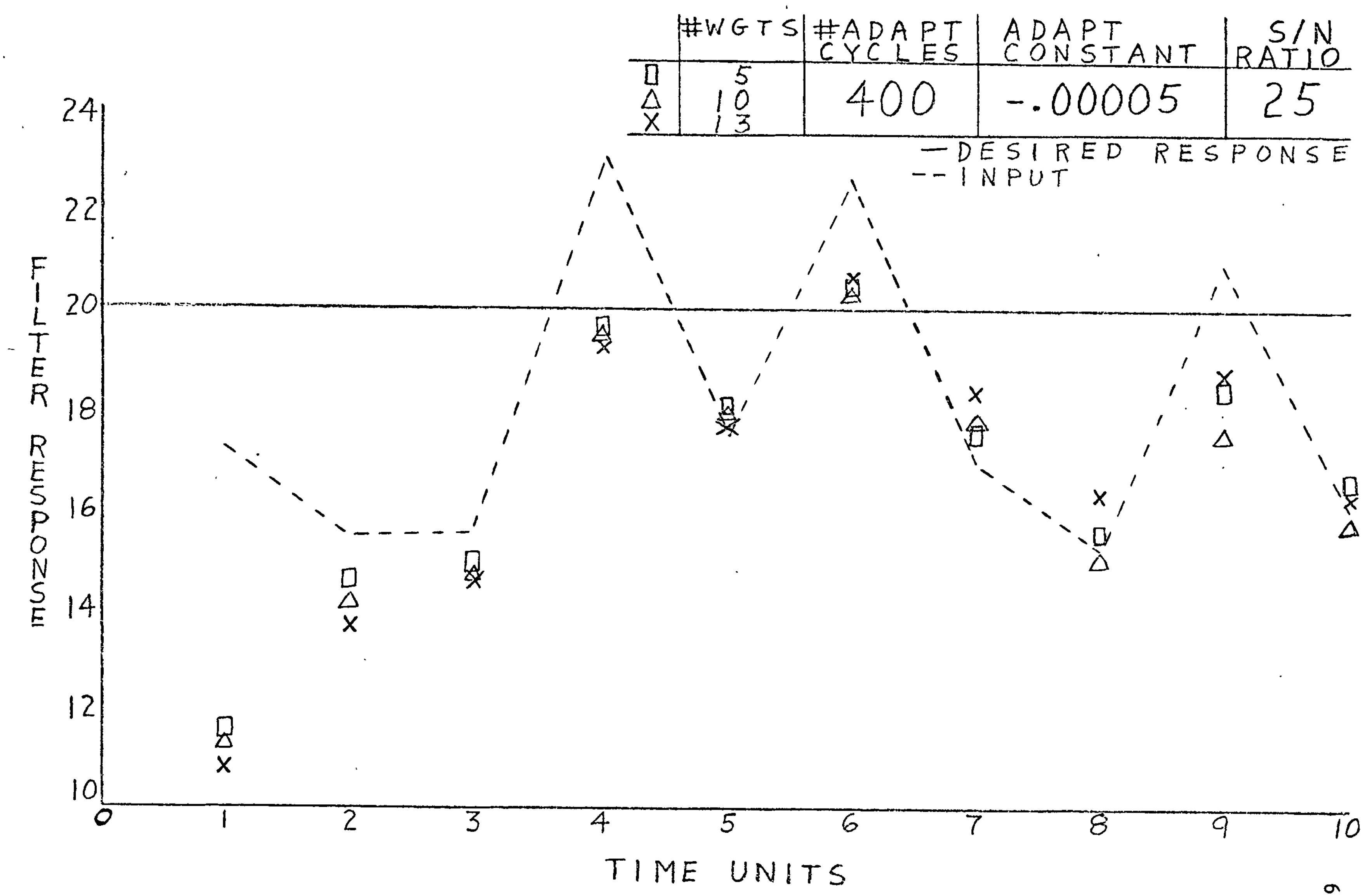

Figure 5-21. Feedforward Adapted Filter Response (Low S/N), Problem C.

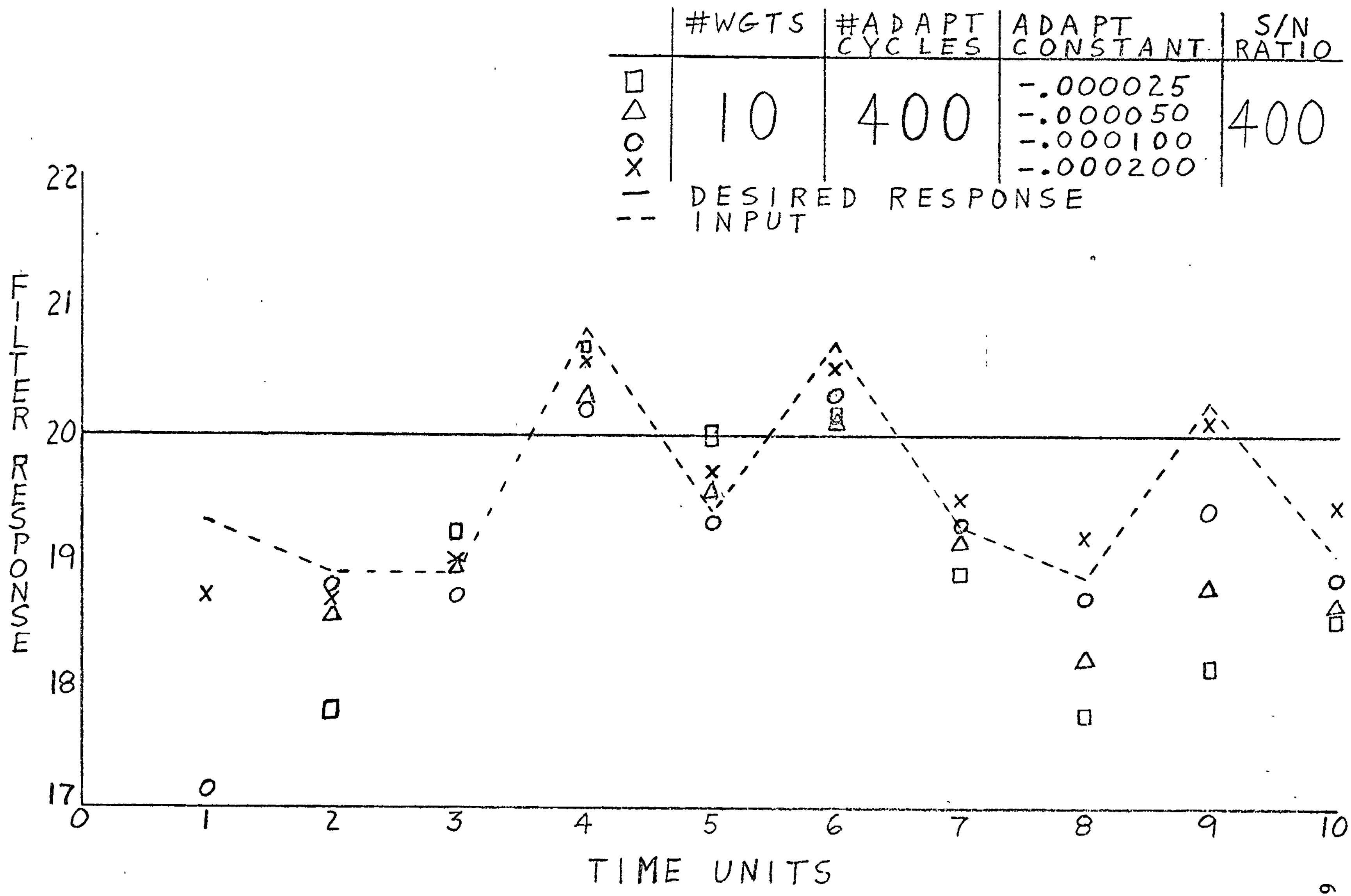

Figure 5-22. Feedforward Adapted Filter Response (High S/N),Frobiem C.

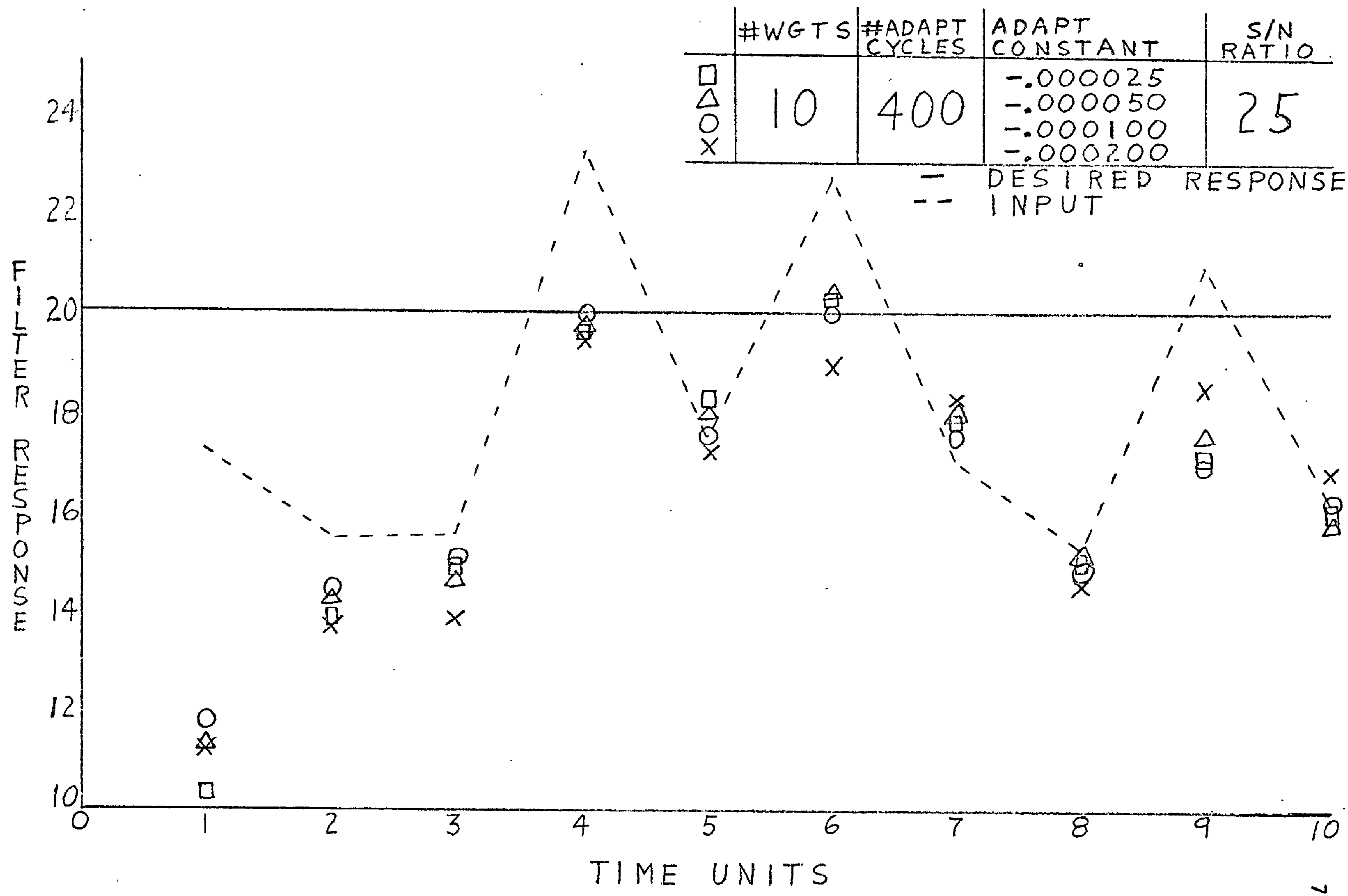

Figure 5-23. Feedforward Adapred Filter Responee (Low S/N),Problem C.

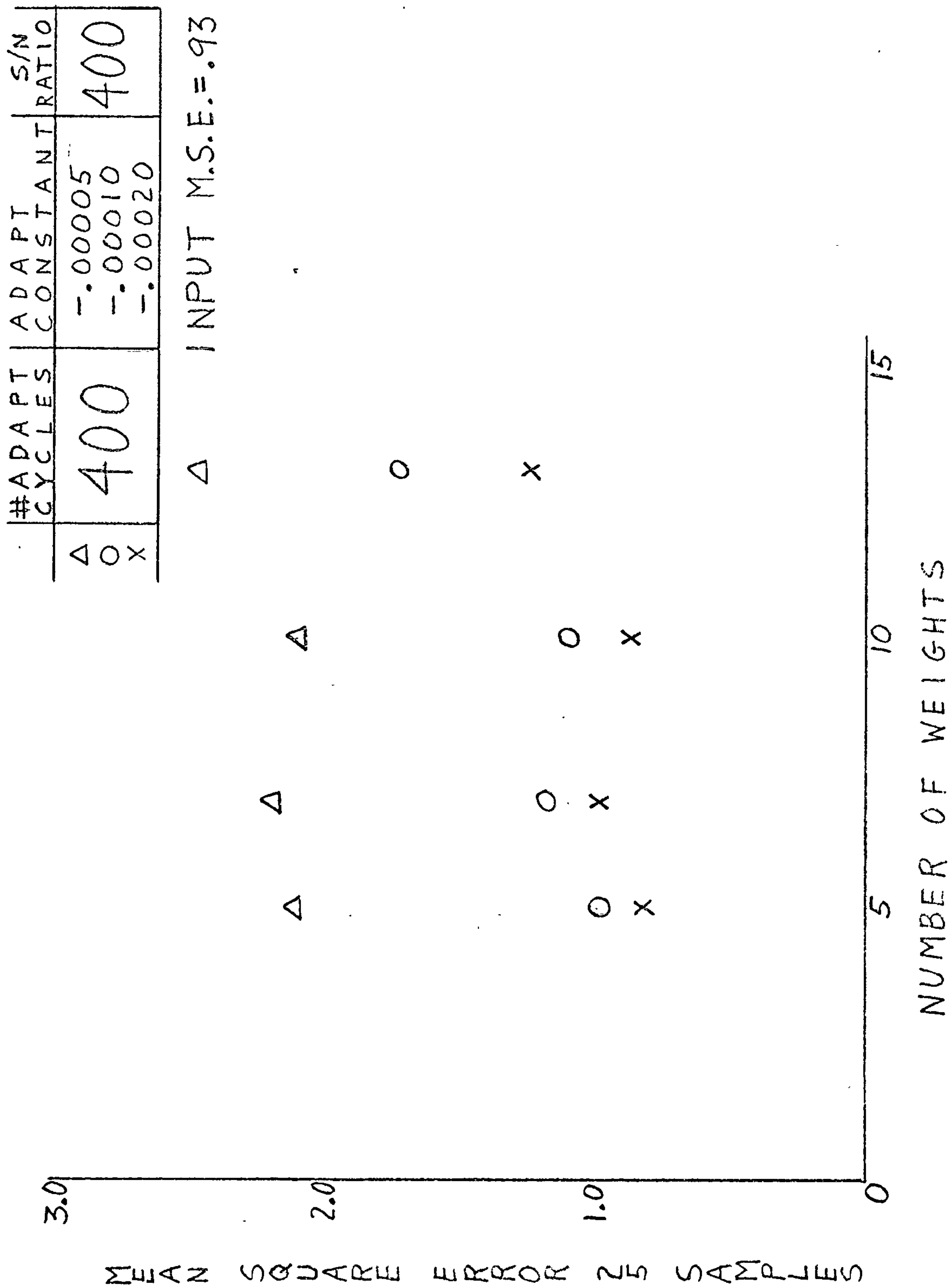

Figure 5-24. Feedforward Filter Output Mean Square Error (High S/N),Problem C.

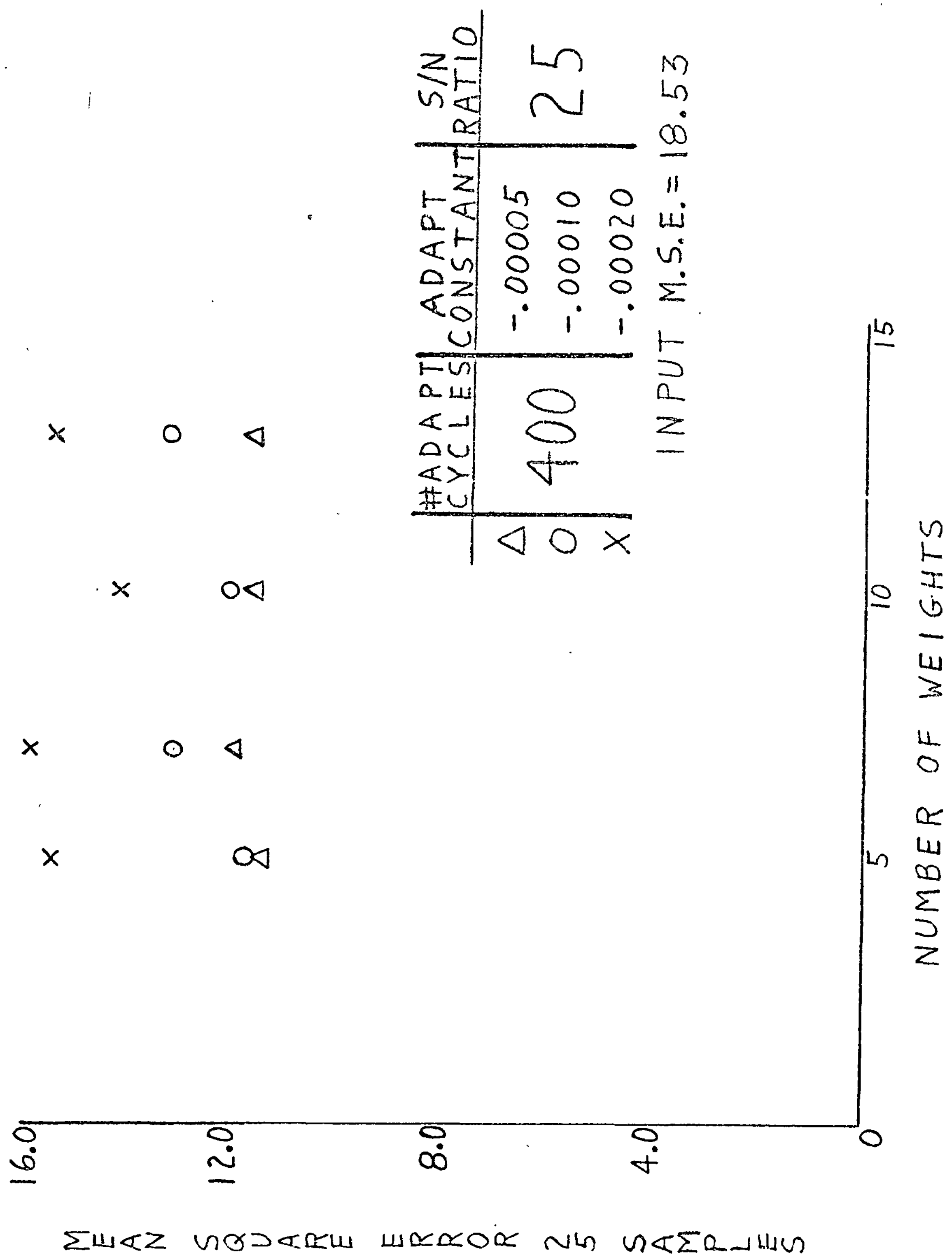

Figure 5-25. Feedforward Filter Output Mean Square Error (Low S/N),Problem C.

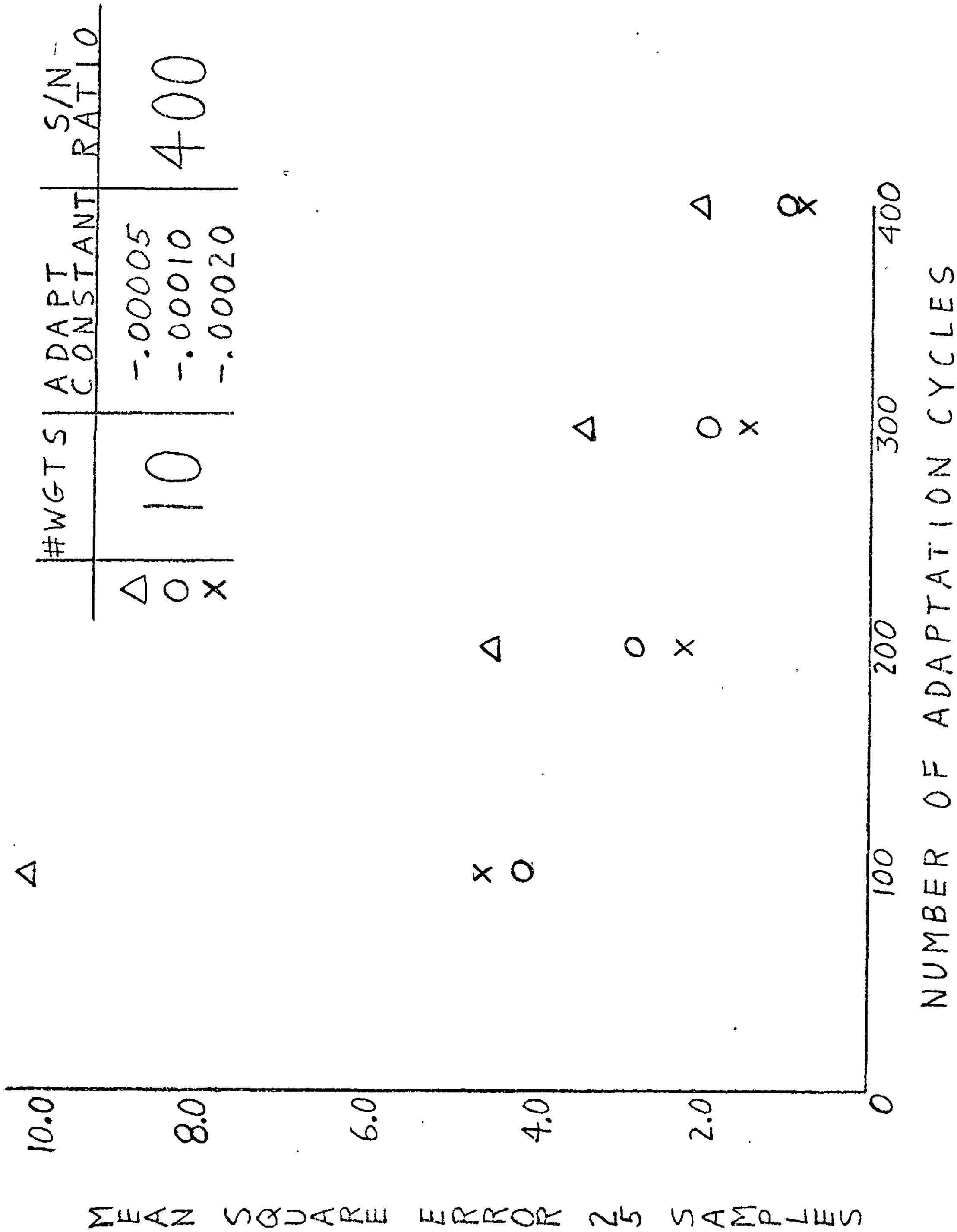

Figure 5-26. Feedforward Filter Output Mean Square Error (High S/N),Problem C.

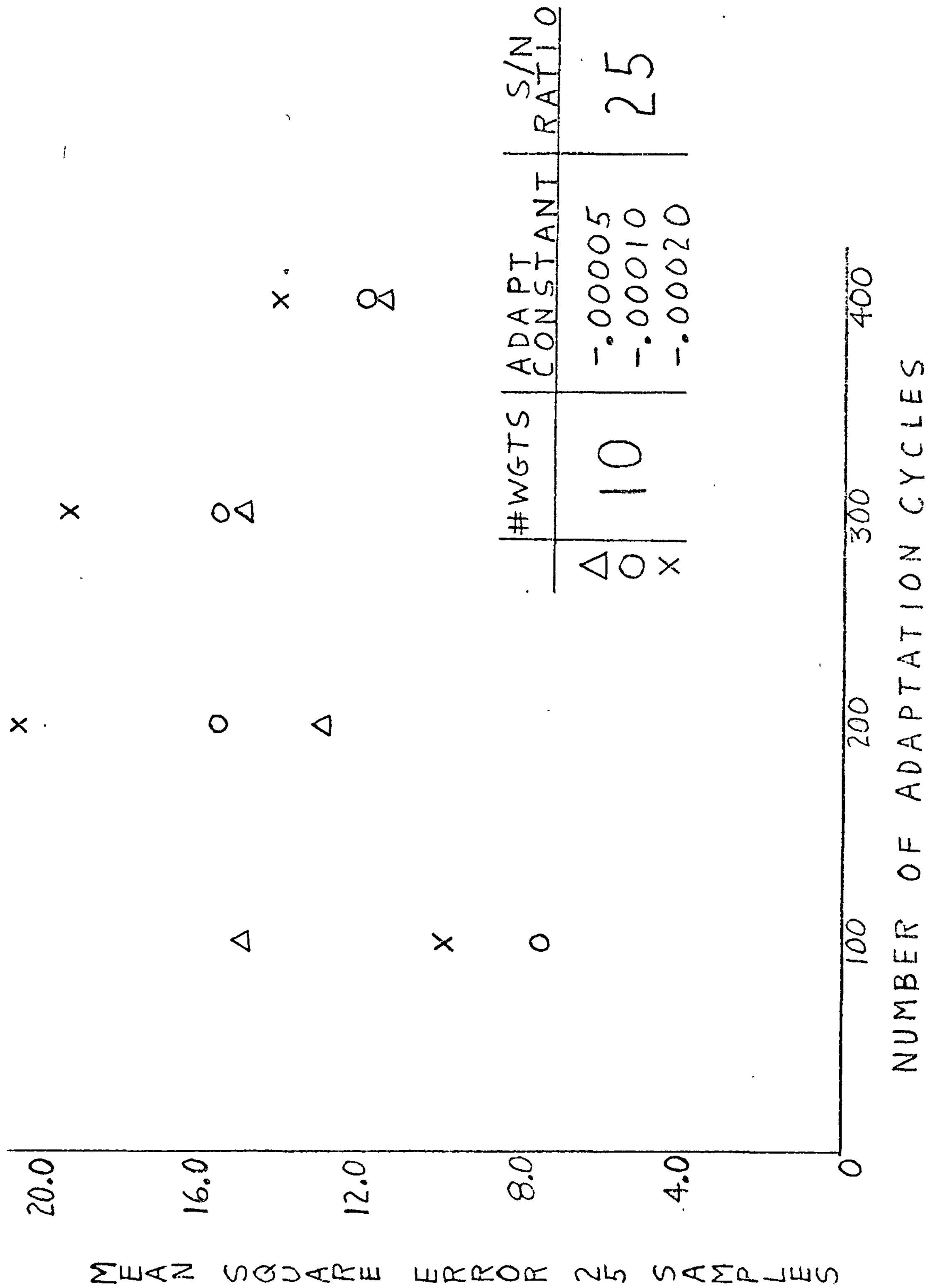

Figure 5-27. Feedforward Filter Output Mean Square Error (Low S/N),Problem C.

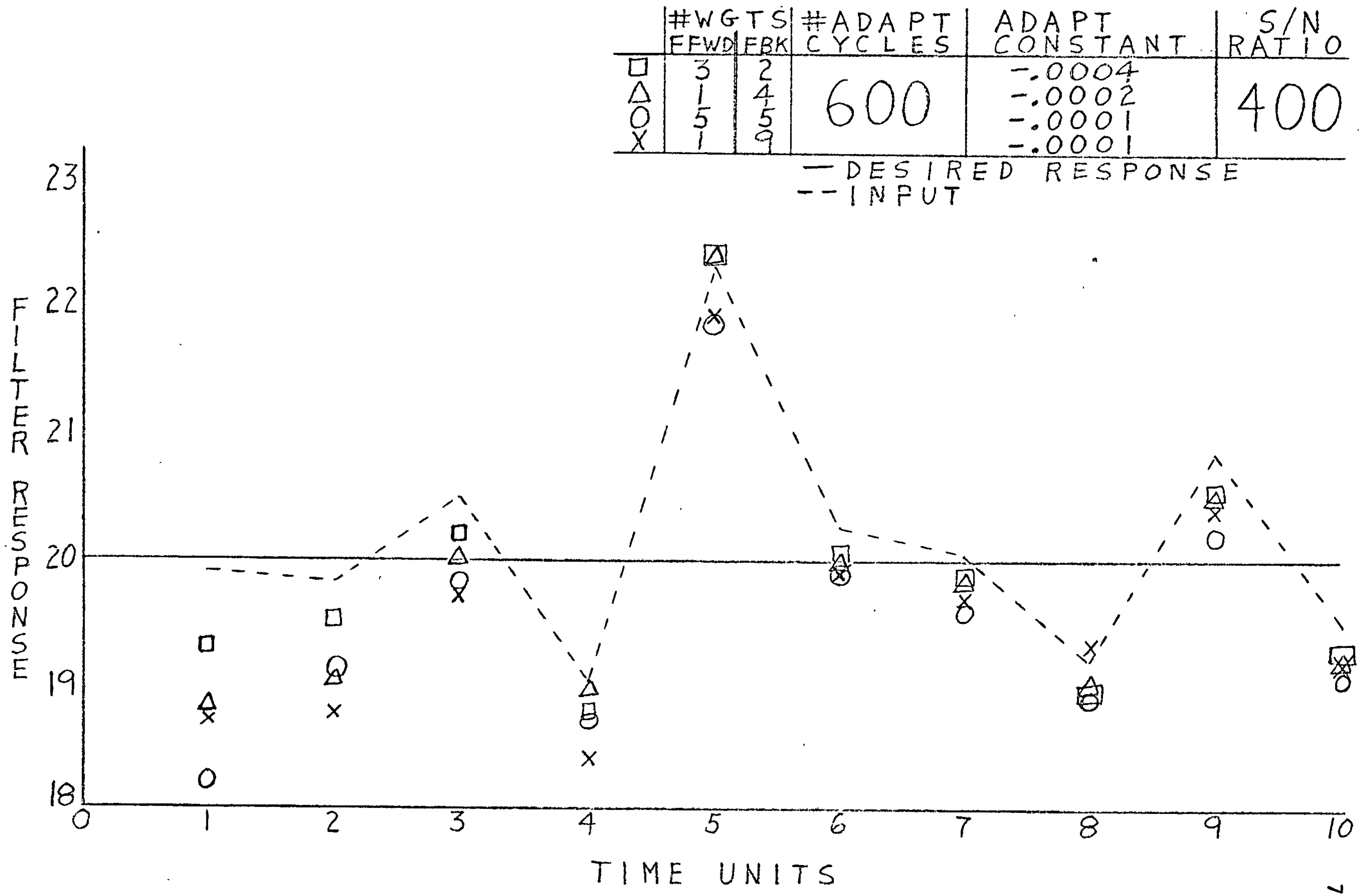

Figure 5-28. Feedback Adapted Filter Response (High S/N), Problem C.

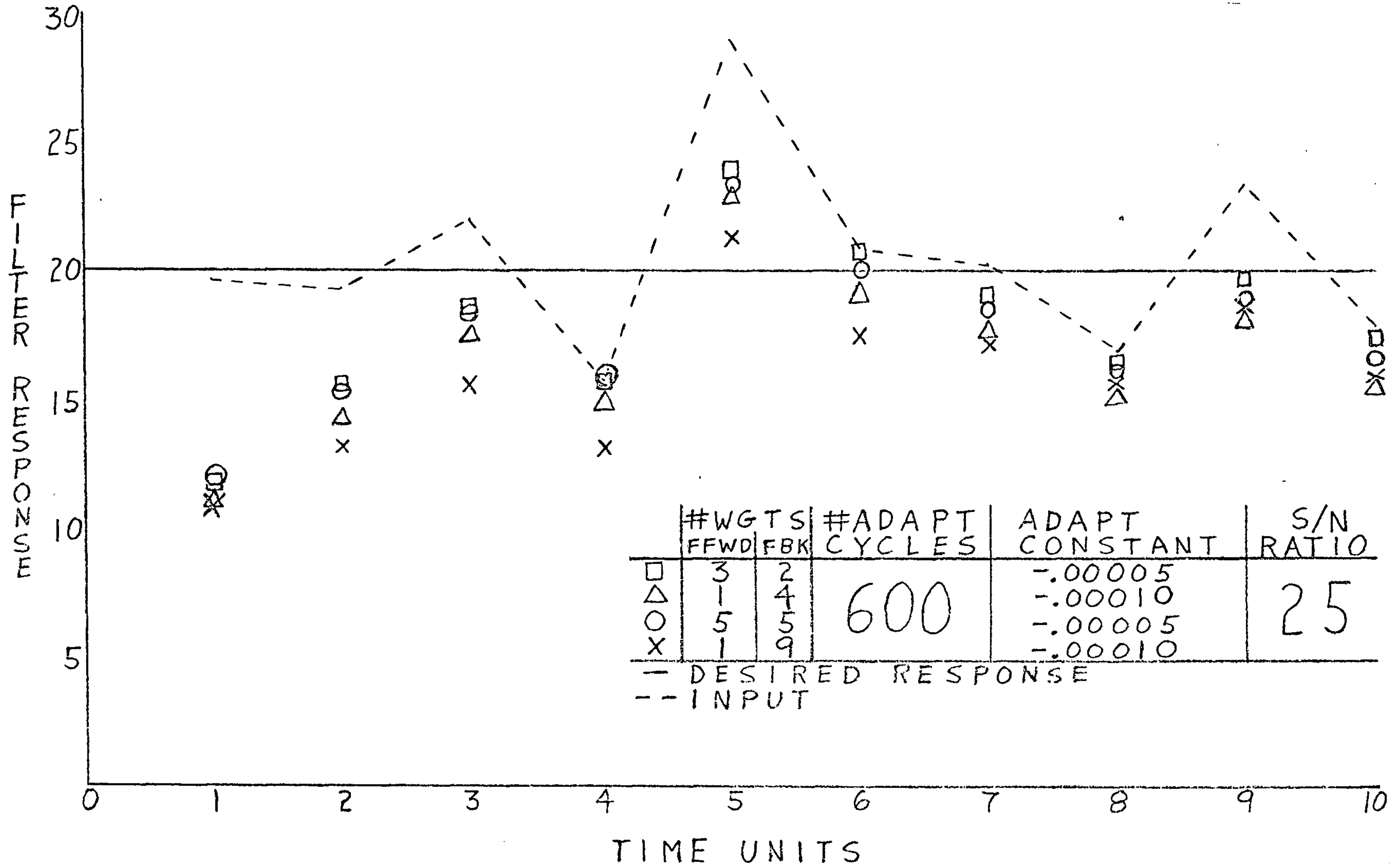

Figure 5-29. Feedbaok Adapted Filter Response (Low S/N),Froblem C.

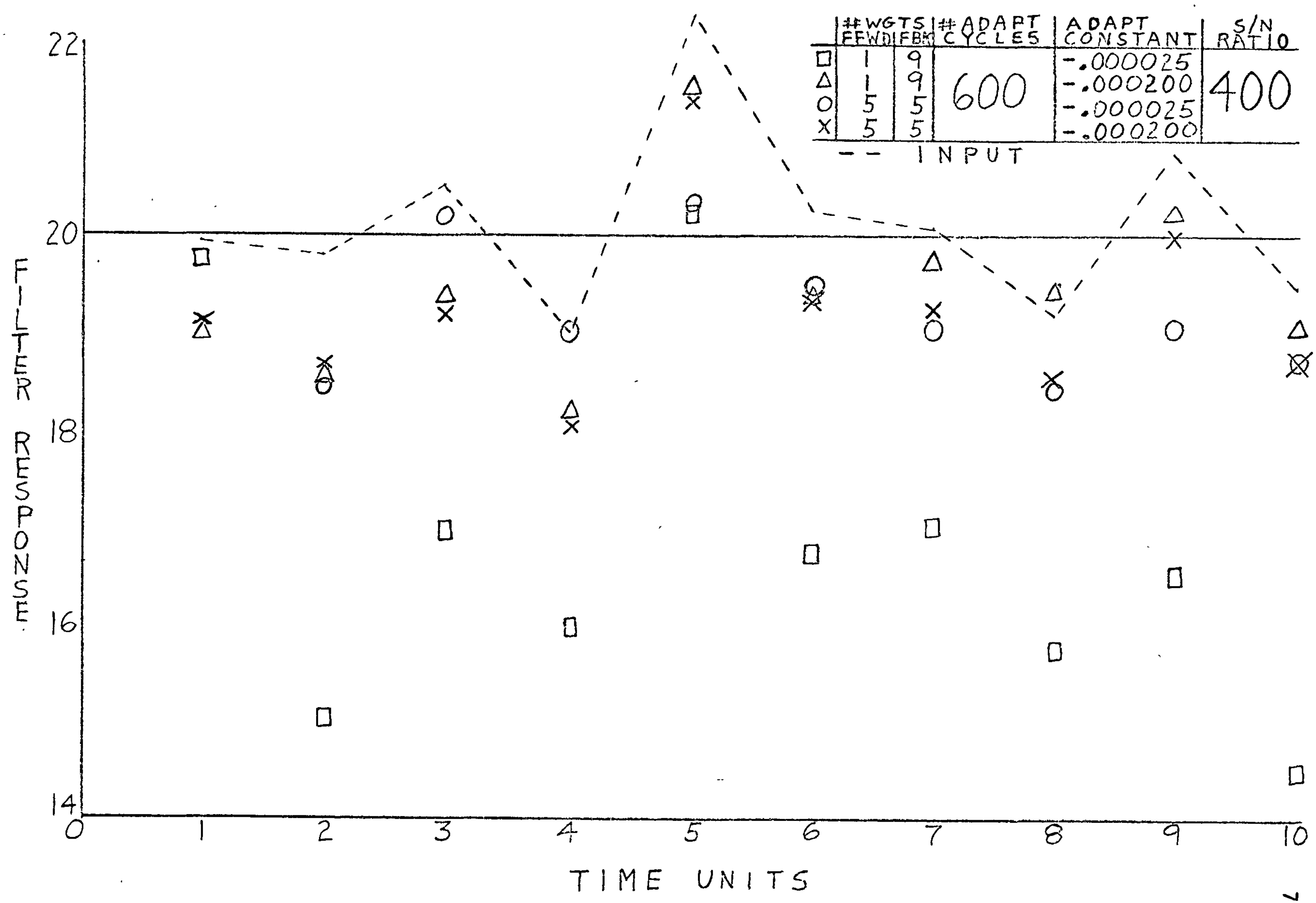

Figure 5-30. Feedback Adapted Filter Response (High S/N),Problem C.

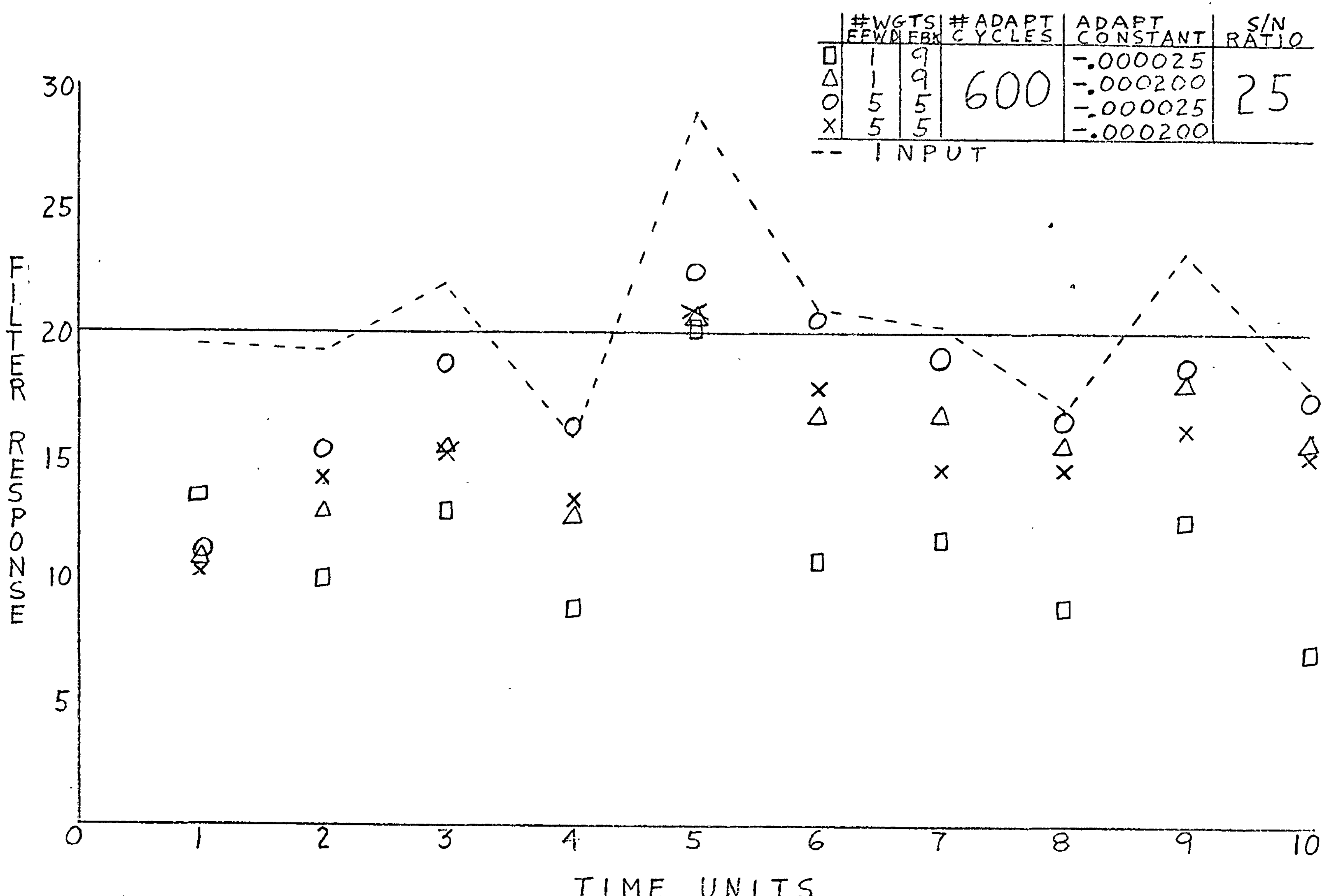

Figure 5-31. Feedbaok Adapred Filter Response (Low S/N),Problem C.

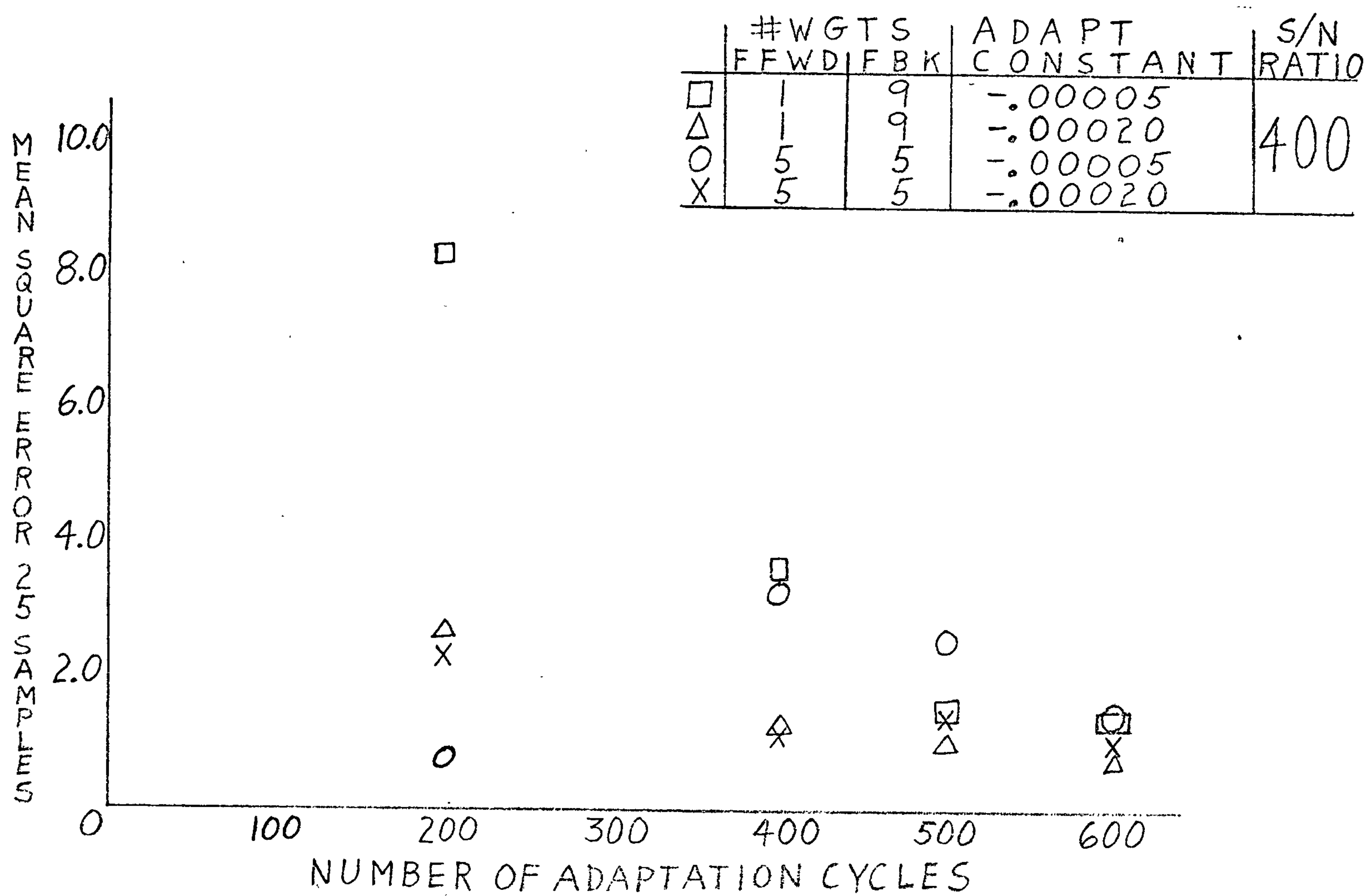

Figure 5-32. Feedbaek Filter Output Mean Square Error (High S/N),Problem C.

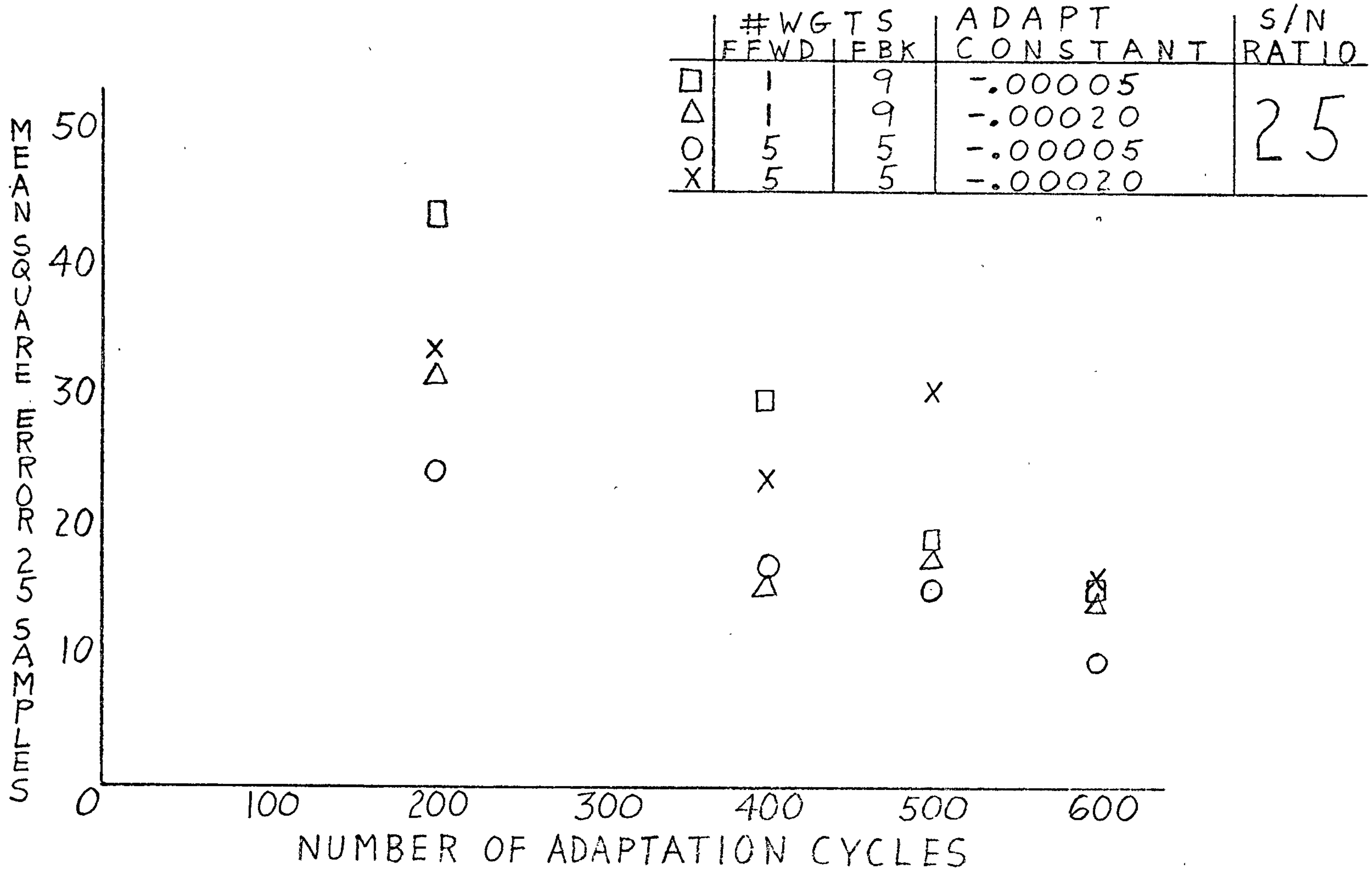

Figure 5-33. Feedback Filter Output Mean Square Error (Low S/N), Problem C.

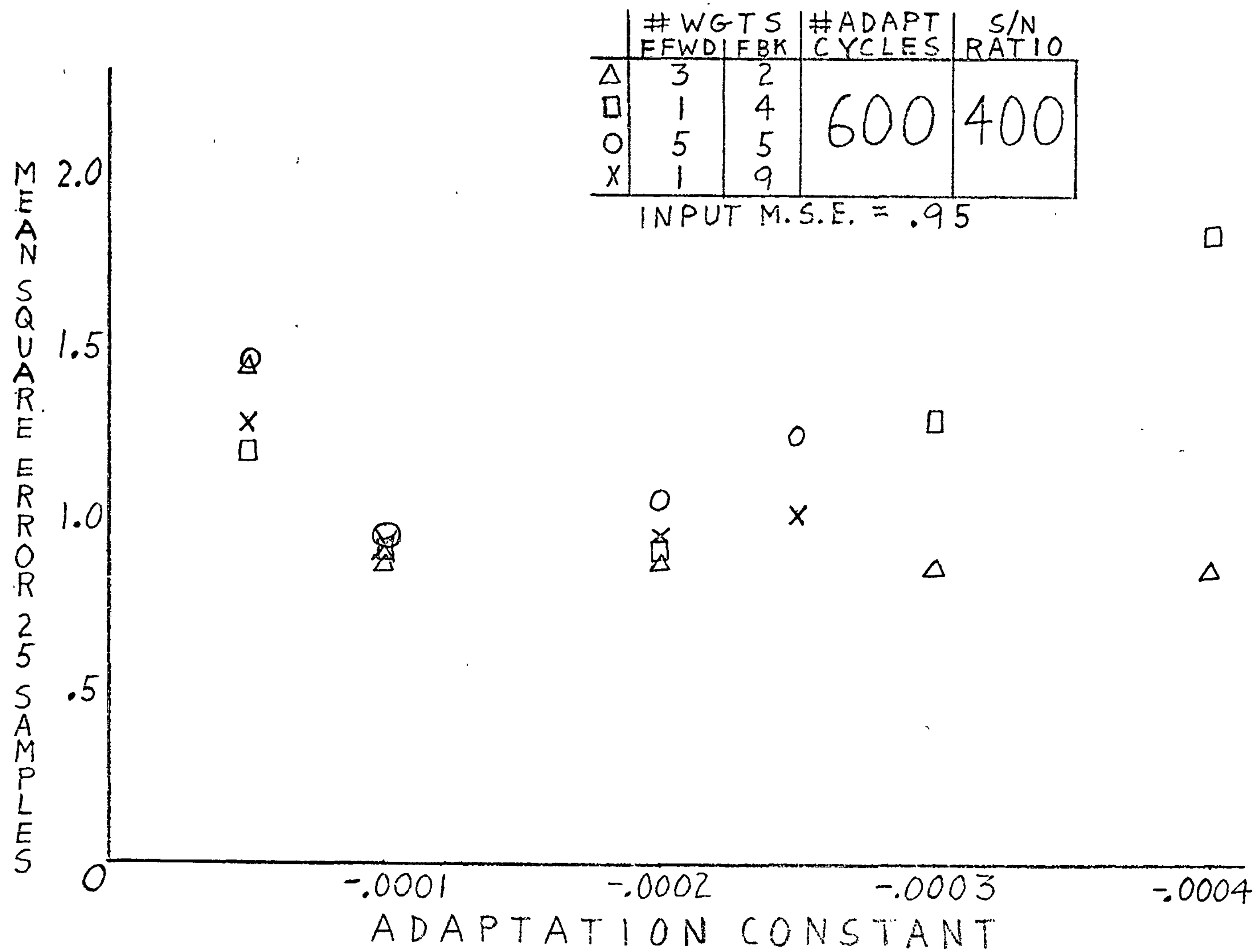

Figure 5-34. Feedback Filter Output Mean Square Error (High S/N),Froblem C.

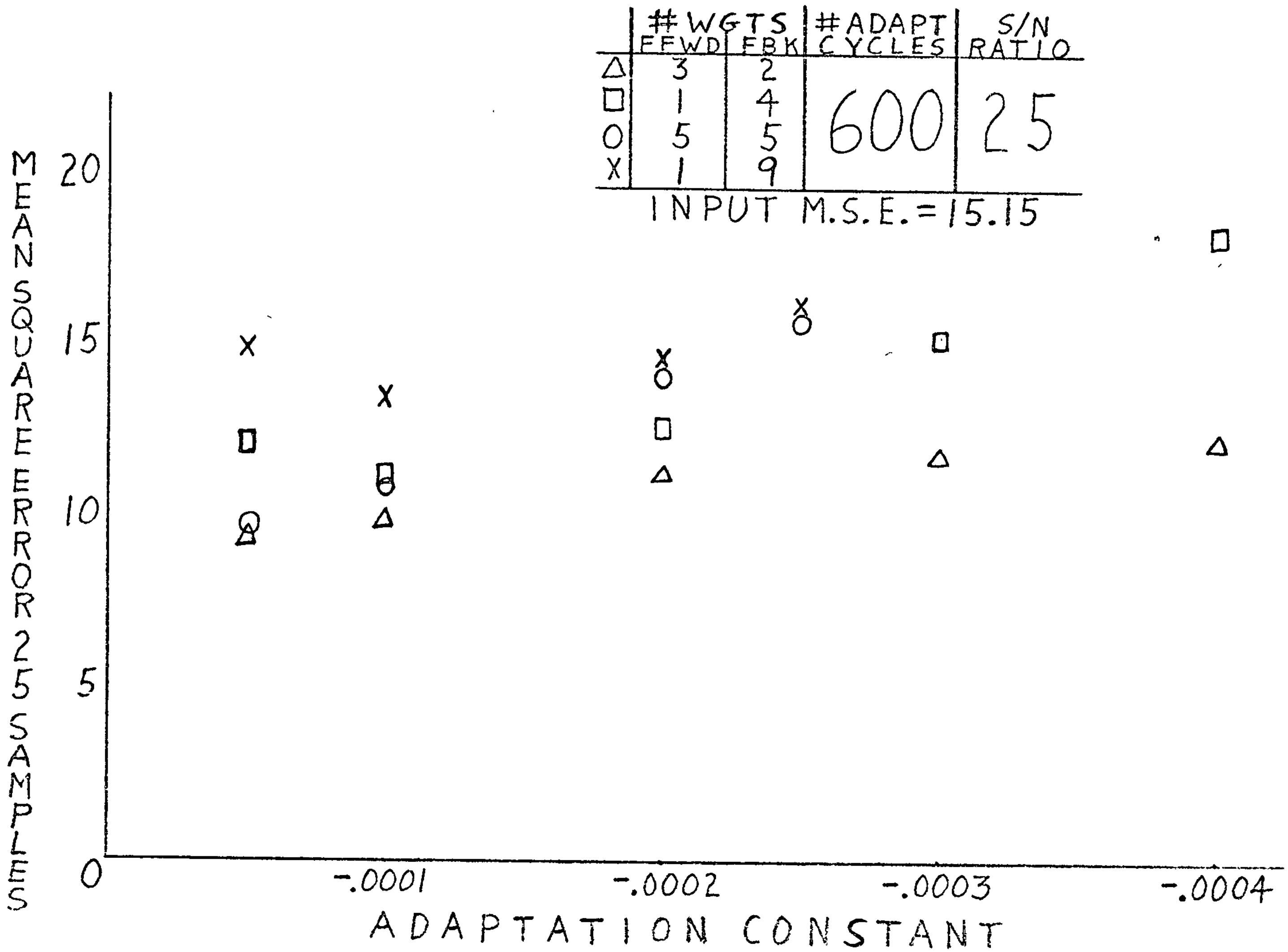

Figure 5-35. Feedbatk Filter Output Mean Square Error (Low S/N),Froblem C.

One general observation can be made for all types of filters, all S/N, and all experimental values of the filter parameters. The output waveform was much "smoother" than the input waveform in all cases. The maximum amplitude change for consecutive input samples was always greater than that for consecutive output samples. In other words, the filter was reducing the variance or noise power through the adaptation process. The smoothing effect was more noticeable as the S/N ratio was decreased.

The best adaptation constant for all filter types when the S/N was high was a relatively large (most negative) value. However, for low S/N (less than 44.4) a small constant proved superior in terms of output average square error.

This is an anticipated result. When the S/N is high the noise effect is small and the input signal possesses nearly a deterministic quality. As determined from the previous two problems, there is an advantage to adjusting at a fast rate when the gradient measurement noise effect is very small or non-existent. However, for small S/N when the input samples are composed of a large amount of noise, adjusting slowly is thought to be a remedy for reducing the steady state gradient noise fluctuations. The experimental results fortify theoretical expectations in this area.

The mean square error of the output was much lower than that of the input for small S/N but was slightly higher for large S/N. This is not alarming when it is known that the transients undergone by the weights in adjusting to the error surface minimum have time constants inversely proportional to the variance of the noise process.

Equations (19) and (30) of Chapter ii might be rewritten for <u>all</u> time constants as

$$T \approx \frac{-1}{2\ k_s\ \sigma^2} \tag{34}$$

where σ^2 is the variance of the noise process.

The eigenvalues of the input correlation matrix are all equal to the variance of the process for White Gaussian Noise.[7] As the variance increases (S/N decreases) the time constant decreases.

Therefore, the weights were probably being heavily affected by transients for high S/N cases. Calculations show that for $\sigma^2 = 1,2$ many more adaptation cycles than were used in the simulation would be needed to make transient effects negligible.

No trends could be established concerning the relations between filter performance and number of weights.

Although no filter "rise time" phenomenon could be observed <u>consistently</u>, there were many cases in which the first one or two output samples were well below the average amplitude level for the rest of the samples.

Comparison of the various types of filters showed that the combination feedforward and feedback filter consistently had slightly smaller mean square error. However, the "smoothing" effect was a little more pronounced for the feedforward filter than for either of the other types.

In both categories the pure feedback performance was worst but not significantly.

No overall rating can be placed on the various types of filters with the limited data available. But it seems that with a time-limited desired response, the smaller adaptation time requirement gives an advantage to the feedforward filter.

CHAPTER VI

CONCLUSIONS

The main objectives of this thesis--to simulate an adaptive filter system on a digital computer and to make a qualitative comparison between previously developed theory and experimental results--have been fulfilled.

In Chapter II some of the theory developed by Patrick Mantey and Bernard Widrow for the System Theory Laboratory at Stanford University was presented. In essence they showed that an adaptive filter based on a mean square error performance criterion and using a gradient search method (steepest descent) can adjust its variable weights or gains so that a minimum mean square difference between a desired response and the filter output is obtained.

In the case of feedback adaptive filters Mantey has shown that feeding back the desired response rather than the filter output during adaptation eliminates any possibility of the error surface containing local minima.

It has also been shown that the weights undergo transients in adjusting to the error surface minimum which are inversely proportional to the adaptation constant and the eigenvalues of the input correlation matrix.

The stability analysis for adaptive filters has indicated that the adaptation constant must be less than zero and greater than the negative inverse of the squared magnitude of the input sample vector.

Finally, it was shown that in the steady state the weights undergo fluctuations about their least mean square values. This phenomenon is called gradient measurement noise and results from the fact that the actual error square gradient and the estimated gradient using the Least Mean Square Algorithm are not equal. Widrow concluded that since this excess mean square error is proportional to the adaptation constant, a slow adjustment rate would be the easiest way to eliminate this problem.

In all three examples considered in this experiment the theory proved basically sound. Transients during weight adjustment were observed regularly with the effects being most severe for small adaptation constants.

The stability bounds on the adaptation constant were fairly accurate for all cases. When the adjustment rate was increased to values exceeding the maximum bound definite signs of filter instability (exceptionally large outputs) were observed.

For the Gaussian Noise Problem when the S/N was small the outputs were affected by gradient measurement noise. However, the steady state mean square error was much less when small adaptation constants were used.

The combination feedforward and feedback filter consistently had the smallest mean square error, but the feedforward filter was superior in "smoothing" the noise (reducing noise power).

The results for the continuous time response problem $(\frac{\text{SIN X}}{\text{X}})$ showed that for feedforward filtering a fast adaptation rate was best when the input signal was deterministic but a relatively slow rate proved superior in the other cases. This unexpected result could not be explained. The combination feedforward and feedback filter gave the best approximation to the desired response, although the feedforward filter performance was very satisfactory.

For the RC Channel Distortion Problem the combination filter had a slightly better response shape than the feedforward filter but fell far short of the desired response amplitude. The feedforward filter output amplitude was very close to the desired value. A fast adaptation rate proved superior in all cases except the pure feedback filter.

No particular mention has been given to the pure feedback filter because its performance in every aspect was far inferior to the other two filter types.

It is not evident to this author which type of filter (feedforward or combination) was most outstanding overall. In instances where the desired response is time-limited the shorter adjustment time requirement makes the feedforward filter most desirable. However, the fact that adaptive filters are workable systems not only on paper but also under experimental conditions has been initially substantiated by the results of this thesis.

CHAPTER VII

SUGGESTIONS FOR FUTURE RESEARCH

It was not possible to explore all of the potential uses for adaptive filters nor was it intended that this thesis be an all-inclusive analysis of adaptive filter operation. Rather it is hoped that this experimental work might generate further interest in this area.

There are several extensions to the work just completed and several new topics which might be investigated in future research on discrete-time adaptive filters.

A thorough mathematical analysis of the stability regions for <u>feedback</u> adaptive filters is necessary in order to determine why a relatively small adaptation constant proved superior to a large one for deterministic signals.

The possibility of using the repetition method for continuous time input signals when most of the energy is concentrated in a relatively small time interval could be explored.

It might prove extremely interesting to carry out the computer simulation to complete filter steady state conditions to observe gradient measurement noise closely. This would eliminate the weight adjustment transients observed so often in this work.

Other stochastic processes with different statistics might be generated in the digital computer or in an analog computer and then digitized. The adaptive filter performance for such distributions as Poisson (shot noise), Rayleigh (fading channels with random phase and amplitude), and Gamma could be investigated.[6,7]

If a method could be developed to generate non-stationary processes or those with slowly time varying statistics, then a study of the adaptive filter performance in this case would be extremely valuable. Widrow has suggested that it is in this area where the adaptive filter would have its greatest advantage over conventional designed filters.

Very little has been mentioned about the frequency response of the adaptive filter in this thesis. But it seems plausible that such filters could be designed and analyzed using Z-transform techniques. Frequency spectrum studies would be another topic worth considering.

Finally, a comparison of the performances of the adaptive filter and an optimal Weiner filter would provide important knowledge about the advantages, if any, in using this self-optimizing signal processor.

APPENDIX

A. Generation of Normally Distributed Samples

In the IBM 7094 computer a random number generator is provided as an external function RECDIS( ).[9] It is designed to supply pseudo-random floating point numbers with a uniform distribution in the interval (0,1). Such a distribution has a probability density function of the form

$$p_x(x) = 1 \quad 0 < x < 1$$
$$= 0 \quad \text{elsewhere}$$

It can be shown by use of the famous central limit theorem that the uniformly distributed random numbers can be transformed into other numbers with an approximate normal distribution having mean zero and variance one.[1,2]

> "If $X_1, X_2, \ldots, X_N$ are random samples from a distribution which has mean u and variance σ^2, then the random variable
> $$Y = \sqrt{N}\ (\bar{X}-u)/o$$
> has a limiting distribution that is normal with mean zero and variance one."[2]

Here $\bar{X} = (\sum_{1}^{N} X_i)/N$ is the sample mean.

For the case of the uniform distribution over the unit interval, $u = 1/2$ and $o^2 = 1/12$.

In this computer simulation of a Gaussian Noise Process 49 uniform samples were used to create a single normal sample. In order to check the approximate distribution of the newly generated numbers, a program was written which grouped these samples into intervals of size 1/2 centered about zero and also calculated their sample mean and variance.

Table A-1 shows the results for 400 such numbers.

TABLE A-1

Distribution of Simulated Noise Samples

Frequency of Occurrence	Interval
1	[-∞, -3.0]
0	(-3.0, -2.5]
5	(-2.5, -2.0]
22	(-2.0, -1.5]
39	(-1.5, -1.0]
65	(-1.0, -0.5]
78	(-0.5, 0]
78	(0, 0.5]
48	(0.5, 1.0]
39	(1.0, 1.5]
14	(1.5, 2.0]
7	(2.0, 2.5]
3	(2.5, 3.0]
1	(3.0, ∞]

Sample Mean = .0328
Sample Variance = 1.0211

Figure A-2 displays a comparison of the ideal normal distribution probability with that of the 400 generated normal numbers.

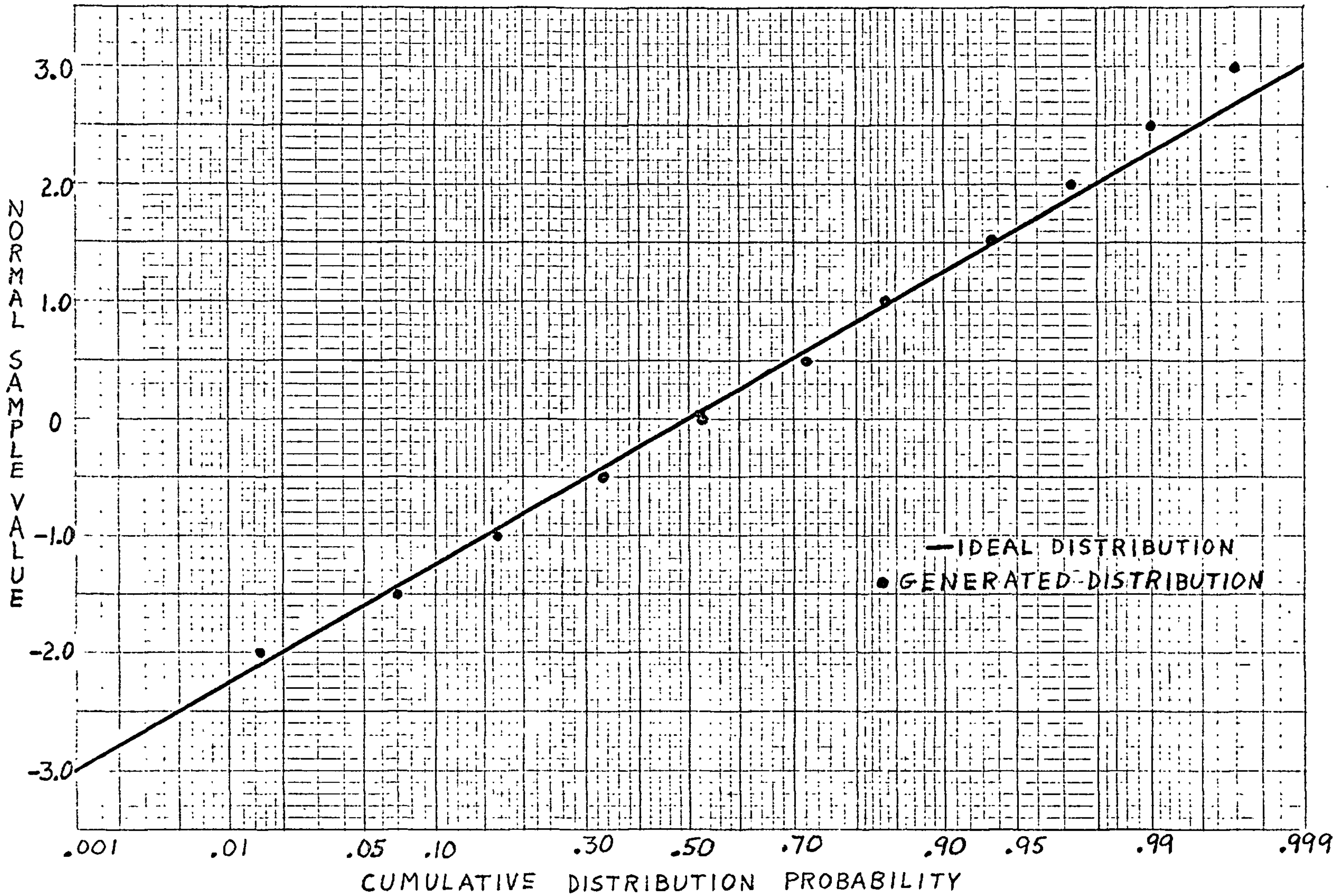

Figure A-2. Comparison of Ideal and Generated Normal Cumulative Distributions

The ideal normal distribution probability function F(x) is given by

$$F(x) = Pr\ (X \leq x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}}\ e^{-W^2/2}\ dW$$

for the unit variance, zero mean case.

The cumulative distribution probability expression for the 400 generated normal numbers G(N) is given by

$$G(N) = \frac{N(x)}{400}$$

where N(x) = the number of generated sample values $\leq$ x.

It is obvious that these two distributions lie in the interval [0,1] and are monotonic increasing functions in x.

The generated numbers are very nearly normal as the comparison of the distributions show.

B. Computer Programs

The actual computer programs used in the simulation are presented in the next few pages. In the READ and WRITE statements the numbers 5 and 6 appear regularly. These numbers refer to the devices which perform those operations in the IBM 7094 system. For the most part changing these calling digits and observing maximum array dimensioning would be the only requirements to use these programs on the PDP-9 or any other computer that compiles Fortran IV.

Table A-3 lists several of the important computer variables used throughout the programs and gives a brief description of their purpose. The corresponding mathematical variables from Chapter II are given in [ ] where appropriate.

TABLE A-3

Program Variables

Name of Variable		Definition
A(M)	$[A_m]$	M^{th} Feedforward Weight (Gain)
AINTL(M)		M^{th} Feedforward Weight (Initial Value)
ADRATE	$[k_s]$	Constant Controlling Adaptation Rate
B(M)	$[B_m]$	M^{th} Feedback Weight (Gain)
BINTL(M)		M^{th} Feedback Weight (Initial Value)
D(L)	[D(L)]	Desired Output at L Time Units
EMSQ		Time Averaged Mean Square Error
ERR	[E( )]	Error; Desired Output - Filter Output
ICYC		Number of Iteration or Adaptive Cycles
IREPET		Number of Repetitions of 25 Input Samples
N	[N]	Number of Weights (Gains); Feed-forward Case Only
NA	[N+1]	Number of Feedforward Weights
NB	[N]	Number of Feedback Weights
RC		RC Channel Time Constant
SMEAN		Sample Mean of Random Uniform Numbers
UNNM		A Uniform Number from Generator Routine
W(M)	$[W_m]$	M^{th} Weight (Gain); Feedforward Case Only
WTINTL(M)		M^{th} Weight; Feedforward Case Only (Initial Value)
X(L)	[X(L)]	Input Sample at L Time Units
X(L,I)	[X(L-(i-1))]	Input Sample at Time L at I^{th} Weight; Feedforward Case Only
Y(L)	[Y(L)]	Filter Output at L Time Units

In order to relate the mathematics of the adaption process developed in Chapter II with the computer simulation programs a detailed description of the important calculations will be presented.

For the feedforward adaptive filter the error at time L is given by

$$E(L) = D(L) - \sum_{i=1}^{N} W_i \, X(L-(i-1))$$

where $D(L)$ = the desired response sample at time L
W_i = the i^{th} feedforward weight or variable gain
$X(L-(i-1))$ = the input signal sample at time L-(i-1)

The Least Mean Square (LMS) Algorithm used to adjust the weights during each adaptation cycle employs the gradient search Method of Steepest Descent. This gradient search method may be expressed as a relation

$$\underline{W_{\substack{PRESENT\\CYCLE}}} = \underline{W_{\substack{PREVIOUS\\CYCLE}}} + k_s \, \underline{\overline{\nabla E^2}_{\substack{PREVIOUS\\CYCLE}}}$$

where k_s = the adaptation constant controlling the adjustment rate.

The LMS Algorithm uses a mean square error gradient estimate instead of the true error gradient and this measured estimate is given by

$$\underline{\overline{\nabla E^2(L)}} \approx \underline{\nabla E^2(L)} = -\,2\,E(L)\,\underline{X(L)}$$

where $E(L)$ = the error at time L
$\underline{X(L)}$ = the vector of input samples affecting the output at time L.

Now the LMS algorithm for adjusting the weight vector may be expressed by

$$\underbrace{W_{\substack{PRESENT\\CYCLE}}} = \underbrace{W_{\substack{PREVIOUS\\CYCLE}}} - 2\ k_s\ E_{\substack{PREVIOUS\\CYCLE}}\ \underbrace{X_{\substack{PREVIOUS\\CYCLE}}}$$

or

$$\underbrace{W(L+1)} = \underbrace{W(L)} - 2\ k_s\ E(L)\ \underbrace{X(L)}$$

for the weight vector at time L+1.

Once the adaptation process has been completed the input samples are reapplied to the adjusted filter starting at time zero. An approximate mean square error can be calculated for the normal operation process which uses time averages.

$$\overline{E^2(L)} \simeq E^2_{\text{TIME AVERAGE}}(L) = \frac{\sum_{K=1}^{L} (D(K) - Y(K))^2}{L}$$

where $Y(K) = \sum_{i=1}^{N} W_{i\ \text{ADJUSTED}}\ X(L-(i-L)$ is the adjusted filter output at time K.

For the adaptive filter employing feedback the error at time L (during adaptation) is given by

$$E(L) = D(L) - \sum_{i=L}^{N+L} A_i\ X(L-(i-L)) - \sum_{j=L}^{N} B_j\ D(L-j)$$

where $D(L)$ = the desired response sample at time L
A_i = the i^{th} feedforward weight (gain)
B_j = the j^{th} feedback weight
$X(L-(i-L))$ = the input signal sample at time L-(i-L)

The LMS Algorithm used for adjusting all weights in the feedback filter is the same as that for the feedforward filter. The gradient estimate used instead of the true error gradient for ease of implementation is given by

$$\overline{\underline{\nabla E^2(L)}} \approx \underline{\nabla E^2(L)} = -2\ E(L)\ \underline{X(L)}$$

for the feedforward weights, and

$$\overline{\underline{\nabla E^2(L)}} \approx \underline{\nabla E^2(L)} = -2\ E(L)\ \underline{D(L)}$$

for the feedback weights, where

$E(L)$ = the error at time L
$\underline{X(L)}$ = the vector of input samples affecting the output at time L
$\underline{D(L)}$ = the vector of desired response (feedback) samples affecting the output during adaptation.

Now the LMS algorithm for adjusting the feedforward and feedback weight vectors may be expressed by

$$\underline{A(L+1)} = \underline{A(L)} - 2\ k_s\ E(L)\ \underline{X(L)}$$

for the feedforward weight vector at time L + 1, and

$$\underline{B(L+1)} = \underline{B(L)} - 2\ k_s\ E(L)\ \underline{D(L)}$$

for the feedback weight vector at time L + 1.

Once the adaptation process has been completed the input samples are reapplied to the adjusted filter starting at time zero. However, the actual filter output samples are fed back instead of the desired response samples during this normal operation mode. An approximate mean square error using time averages can also be calculated for the feedback filter.

$$\overline{E^2(1)} \approx E^2_{\text{TIME AVERAGE}}(L) = \frac{\sum_{K=1}^{L} (D(K) - Y(K))^2}{L}$$

where $Y(K) = \sum_{i=1}^{N+1} A_i \, X(L-(i-1)) + \sum_{j=1}^{N} B_j \, Y(L-j)$ is the adjusted filter output at time K.

```
C          FORMING SIN X/ X TYPE RESPONSE USING
C          FEEDFORWARD ADAPTIVE FILTERING
           DIMENSION D(500).W(15),Y(500),X(500,15)
           READ(5,100)N,ICYC,ADRATE,(W(M),M=1,N)
100        FORMAT(2I5, F10.5/(7F10.5))
           WRITE(6,200)
200        FORMAT(9X,1HN,9X,4HICYC,9X,6HADRATE,31X,
          C21HINITIAL WEIGHT VALUES/)
           WRITE(6,201) N,ICYC,ADRATE,(W(M),M=1,N)
201        FORMAT(8X,I2,9X,I4,7X,F10.5,5X,5F10.5/
          C(45X,5F10.5))
           DO2 L=1,ICYC
           DO2 I=1,N
           X(L.I)=0.
2          CONTINUE
           DO4 L=1,ICYC
           IF(L.EQ.1)GOTO30
           DO 5 I=2.N
           X(L,I)=X(L-1,I-1)
5          CONTINUE
30         SUM=0.
           D(L)=3.*SIN(FLOAT(L)/8.)/(3.1416*FLOAT(L))
           X(L,1)=D(L)/3.
           DO6 I=1,N
           SUM=SUM+W(I)*X(L,I)
6          CONTINUE
           ERR=D(L)-SUM
           DO4 J=1,N
           W(J)=W(J)-2.*ADRATE*ERR*X(L,J)
4          CONTINUE
           WRITE(6,202)
202        FORMAT(//50X,19HFINAL WEIGHT VALUES)
           WRITE(6,203)(W(J),J=1,N)
203        FORMAT(/10X,10F10.5)
           WRITE(6,204)
204        FORMAT(/30X,13HFILTER OUTPUT,10X,14HDESIRED
          C OUTPUT,10X,10HTIME UNITS)
           SUMSQ=0.
           DO7 L=1,ICYC
           SUM=0.
           DO8 I=1,N
           SUM=SUM+W(I)*X(L,I)
8          CONTINUE
           ESQ=(D(L)-SUM)**2
           Y(L)=SUM
           SUMSQ=SUMSQ+ESQ
           WRITE(6,205) Y(L),D(L),L
205        FORMAT(26X,F15.6,8X,F15.6,12X,I6)
7          CONTINUE
           EMSQ=SUMSQ/FLOAT(L-1)
           WRITE(6,206)L
206        FORMAT(//30X,27HTIME AVERAGE SQUARED ERROR,,
          CI5,14HOUTPUT SAMPLES)
           WRITE(6,207) EMSQ
207        FORMAT(50X,F15.6/////)
           STOP
           END
```

```
C          FORMING SIN X/ X TYPE RESPONSE USING BOTH
C          FEEDFWD AND FEEDBACK ADAPTIVE FILTERING
           DIMENSION X(500),Y(500),D(500),B(14),A(14)
           READ(5,100)NA,NB,ICYC,ADRATE,(A(K),K=1,NA),
         C(B(K),K=1,NB)
100        FORMAT(3I5, F10.5/(5F10.5))
           WRITE(6,210)
210        FORMAT(10X,2HNA,9X,2HNB,10X,4HICYC,10X,
         C 6HADRATE,36X,21HINITIAL WEIGHT VALUES/)
           WRITE(6,209) NA,NB,ICYC,ADRATE,(A(K),K=1,NA),
         C(B(K),K=1,NB)
209        FORMAT(10X,I2,9X,I2,10X,I4, 7X,F10.5,15X,
         C5F10.5/(69X,5F10.5))
           DO4 L=1,ICYC
           SUM=0.
           D(L)=3.*SIN(FLOAT(L)/8.)/(3.1416*FLOAT(L))
           X(L)=D(L)/3.
           IF(L.EQ.1)GOTO19
           K=L-1
           DO6 I=1,NP
           SUM=SUM+B(I)*D(K)
           K=K-1
           IF(K.EQ.0)GOTO20
6          CONTINUE
20         K=L-1
           DO7 I=2,NA
           SUM=SUM+A(I)*X(K)
           K=K-1
           IF(K.EQ,0)GOTO19
7          CONTINUE
19         SUM=SUM+A(1)*X(L)
           ERR=D(L)-SUM
           DO13 J=1,NB
           II=L-J
           IF(II.EQ.0)GOTO17
           B(J)=B(J)-2.*ADRATE*ERR*D(II)
13         CONTINUE
17         DO14 J=2,NA
           II=L-J+1
           IF(II.EQ.0)GOTO18
           A(J)=A(J)-2.*ADRATE*ERR*X(II)
14         CONTINUE
18         A(1)=A(1)-2.*ADRATE*ERR*X(L)
4          CONTINUE
           WRITE(6,202)
202        FORMAT(//50X,19HFINAL WEIGHT VALUES)
           WRITE(6,203)(A(K),K=1,NA)
203        FORMAT(/10X,10F10,5)
           WRITE(6,204) (B(K),K=1,NB)
204        FORMAT(//,(10X,10F10,5))
           WRITE(6,205)
205        FORMAT(//30X,13HFILTER OUTPUT,10X,14HDESIRED
         C OUTPUT,10X,10HTIME UNITS) .
           SUMSQ=0.
           DO8 L=1,ICYC
           SUM=0.
```

```
      IF(L.EQ.1)GOTO21
      K=L-1
      DO9 I=1,NB
      SUM=SUM+B(I)*Y(K)
      K=K-1
      IF(K.EQ.0)GOTO22
9     CONTINUE
22    K=L-1
      DO10 I=2,NA
      SUM=SUM+A(I)*X(K)
      K=K-1
      IF(K.EQ.0)GOTO21
10    CONTINUE
21    SUM=SUM+A(1)*X(L)
      ESQ=(D(L)-SUM)**2
      Y(L)=SUM
      SUMSQ=SUMSQ+ESQ
      WRITE(6,206) Y(L),D(L),L
206   FORMAT(26X,F15.6,8X,F15,6,12X,I6)
8     CONTINUE
      EMSQ=SUMSQ/FLOAT(L-1)
      WRITE(6,207)L
207   FORMAT(//30X,27HTIME AVERAGE SQUARED ERROR,,
     C I5,14HOUTPUT SAMPLES)
      WRITE(6,208) EMSQ
208   FORMAT(50X,F15.6/////)
      STOP
      END
```

```
C         RECONSTRUCTING RECTANGULAR PULSE THAT HAS BEEN
C         DISTORTED BY RC CHANNEL USING FEEDFORWARD
C         ADAPTIVE FILTER
          DIMENSION D(500),Y(500),X(500,15),W(15)
          READ(5,100) N,ICYC,IREPET,ADRATE,RC
          READ(5,101) (W(M),M=1,N)
100       FORMAT(3I5,2F10.5)
101       FORMAT(7F10.5)
          WRITE(6,210)
210       FORMAT(10X,1HN,10X,4HICYC,10X,6HIREPET,10X,
         C6HADRATE,10X,13HTIME CONSTANT,19X,21HINITIAL
         C WEIGHT VALUES/)
          WRITE(6,209) N,ICYC,IREPET,ADRATE,RC,
         C(W(M),M=1,N)
209       FOPMAT(8X,I3,9X,I4,12X,I3,9X,F10.5,8X,F10.5,
         C4X,4F10.5/(80X,4F10.5))
          DD2 L=1,ICYC
          DO2 I=1,N
          X(L,I)=0.
2         CONTINUF
          DO 4 M=1,IREPET
          004 L=1,25
          SUM=0.
          IF(M.GT.1)GOTO70
          IF(L.EQ.1)GOTO30
          DO5 I=2,N
          X(L,I)=X(L-1,I-1)
5         CONTINUF
30        IF(L.LE.10)GOTO60
          D(L)=0.
          X(L,1)=EXP(-FLOAT(L)/RC)*(EXP(10.0/RC)-1.0)
          GOTO70
60        D(L)=1.0
          X(L,1)=(1.0-EXP(-FLOAT(L)/RC))
70        DO6 I=1,N
          SUM=SUM+W(I)*X(L,I)
6         CONTINUE
          ERR=D(L)-SUM
          DO4 I=1,N
          W(I)=W(I)-2.*ADRATE*ERR*X(L,I)
4         CONTINUE
          WRITE(6.202)
202       FORMAT(//50X,19HFINAL WEIGHT VALUES)
          WRITE(6,203) (W(M).M=1.N)
203       FORMAT(/10X,10F10.5)
          WRITE(6,204)
204       FORMAT(/30X.13HFILTER OUTPUT,10X,14HDESIRED
         C OUTPUT,10X,10HTIME UNITS)
          SUMSO=0.
          DO7 L=1,25
          SUM=0.
          DO8 I=1,N
          SUM=SUM+W(I)*X(L,I)
8         CONTINUE
          ESQ=(D(L)-SUM)**2
          Y(L)=SUM
```

```
      SUMSQ=SUMSQ+ESQ
      WRITE(6,205) Y(L),D(L),L
205   FORMAT(26X,F15.6,8X,F15.6,12X,I6)
7     CONTINUE
      EMSQ=SUMSQ/FLOAT(L-1)
      WRITE(6,206)L
206   FORMAT(//30X,27HTIME AVERAGE SQUARED ERROR,,
     C I5,14HOUTPUT SAMPLES)
      WRITE(6,207) EMSQ
207   FORMAT(50X,F15,6/////)
      STOP
      END
```

```
C          RECONSTRUCTING A RECTANGULAR PULSE DISTORTED
C          BY AN RC CHANNEL USING BOTH FEEOFWO
C          AND FEEDBACK ADAPTIVE FILTERING
           DIMENSION X(1000),Y(1000),D(1000).B(14),A(14)
           READ(5.100)NA.NB,ICYC,IREPET,ADRATE,RC,
         C (A(K),K=1,NA),(B(K),K=1,NB)
100        FORMAT(4I5,2F10.5/(7F10.5))
           WRITE(6,210)
210        FORMAT(10X,2HNA,9X,2HNB.10X,4HICYC,10X,
         C6HIREPET,10X,6HADRATE.10X,13HTIME CONSTANT/)
           WRITE(6,209) NA.NB,ICYC.IREPET,ADRATE,RC
209        FORMAT(9X,I3.8X,I3,10X,I4,11X,I3.10X,F10.5,
         C10X,F10.4//)
           WRITE(6,212)
212        FORMAT(50X,21HINITIAL WEIGHT VALUES/)
           WRITE(6,211)(A(K),K=1,NA),(B(K),K=1,NB)
211        FORMAT(10X,10F10.5)
           DO3 M=1,IREPET
           003 L=1,25
           SUM=0.
           IF(M.GT.1)GOTO35
           IF(L.LE.10)GOTO60
           D(L)=0.
           X(L)=EXP(-FLCAT(L)/RC)*(EXP(10.0/RC)-1.0)
           GOTO70
60         O(L)=1.0
           X(L)=(1.0-EXP(-FLOAT(L)/RC))
35         IF(L.EQ.1)GOTO19
70         K=L-1
           DO4 I=1,NB
           SUM=SUM+B(I)*D(K)
           K=K-1
           IF(K.EQ.0)GOTO20
4          CONTINUE
20         K=L-1
           005 I=2,NA
           SUM=SUM+A(I)*X(K)
           K=K-1
           IF(K.EQ.0)GOTO19
5          CONTINUE
19         SUM=SUM+A(1)*X(L)
           ERR=D(L)-SUM
           DO6 J=1,NB
           II=L-J
           IF(II.EO.0)GOTO17
           B(J)=B(J)-2.*ADRATE*ERR*D(II)
6          CONTINUE
17         DO7 J=2,NA
           II=L-J+1
           IF(II.EQ.0)GOTO18
           A(J)=A(J)-2.*ADRATE*ERR*X(II)
7          CONTINUE
18         A(1)=A(1)-2.*ADRATE*ERR*X(L)
3          CONTINUE
           WRITE(6,202)
202        FORMAT(//50X,19HFINAL WEIGHT VALUES)
```

```
      WRITE(6,203)(A(K),K=1,NA)
203   FORMAT(/10X,10F10,5)
      WRITE(6,204) (B(K),K=1,NB)
204   FORMAT(//,(10X,10F10.5))
      WRITE(6,205)
205   FORMAT(//30X,13HFILTER OUTPUT,10X,14HOESIRED
     C OUTPUT,10X,10HTIME UNITS)
      SUMSQ=0,
      008 L=1,25
      SUM=0.
      IF(L.EQ,1)GOTO21
      K=L-1
      009 I=1,NB
      SUM=SUM+B(I)*Y(K)
      K=K-1
      IF(K.EQ,0)GOTO22
9     CONTINUE
22    K=L-1
      0010 I=2,NA
      SUM=SUM+A(I)*X(K)
      K=K-1
      IF(K.EQ.0)GOTO21
10    CONTINUE
21    SUM=SUM+A(1)*X(L)
      ESQ=(D(L)-SUM)**2
      Y(L)=SUM
      SUMSQ=SUMSQ+ESQ
      WRITE(6,206) Y(L),D(L),L
206   FORMAT(26X,F15.6,8X,F15,6,12X,I6)
8     CONTINUE
      EMSQ=SUMSQ/FLOAT(L-1)
      WRITE(6,207)L
207   FORMAT(//30X,27HTIME AVERAGE SQUARED ERROR,,
     C I5,14HOUTPUT SAMPLES)
      WRITE(6,208) EMSQ
208   FORMAT(50X,F15,6/////)
      STOP
      END.
```

```
C          RECONSTRUCTION OF RECTANGULAR PULSE DISTORTED
C          BY ADDIIIVE WHITE GAUSSIAN NOISE USING
C          FEEDFORWARD ADAPTIVE FILTERING
           DIMENSION D(400),W(15).Y(400),X(400,15).WTINTL(15)
           COMMON C(400)
           CALL NORMAL
           READ(5,100)N.ICYC,IREPET,ADRATE,(WTINTL(J),J=1,N)
100        FORMAT(3I5.F10.6/(7F10.5))
           DO1NM=1,6
           DO22J=1,N
           W(J)=WTINTL(J)
22         CONTINUE
           WRITE(6,210)
210        FORMAT(10X,1HN,10X,4HICYC,10X,6HIREPET,10X,
          C6HADRATE,7X,16HNOISE MULTIPLIER,19X,
          C21HINITIAL WEIGHT VALUES/)
           WRITE(6,209)N,ICYC,IREPET,ADRATE,NM,
          C(WTINTL(J),J=1,N)
209        FORMAT(8X,I3,9X,I4,11X,I3,8X,F10.6,7X,I10,
          C7X,4F10.5/(80X,4F10.5))
           II=0
           DO2L=1,ICYC
           DO2I=1,N
           X(L,I)=0.
2          CONTINUE
           DO4M=1,IREPET
           DO4L=1,25
           II=II+1
           SUM=0.
           IF(L.EQ.1)GOTO30
           DO5I=2,N
           X(L,I)=X(L-1,I-1)
5          CONTINUE
30         IF(L.LE.10)GOTO60
           D(L)=0.
           X(L,1)=C(II)*FLOAT(NM)
           GOTO70
60         D(L)=20.0
           X(L,1)=20.0+C(II)*FLOAT(NM)
70         DO6I=1,N
           SUM=SUM+W(I)*X(L.I)
6          CONTINUE
           ERR=D(L)-SUM
           DO4J=1,N
           W(J)=W(J)-2.*ADRATE*ERR*X(L,J)
4          CONTINUE
           WRITE(6,200)
200        FORMAT(//50X,19HFINAL,WEIGHT VALUES)
           WRITE(6,203)(W(J),J=1,N)
203        FORMAT(/10X,10F10.5)
           WRITE(6,204)
204        FORMAT(/30X,13HFILTER OUTPUT.10X.14HDESIRED
          C OUTPUT,10X,10HTIME UNITS)
           SUMSQ=0.
           DO7L=1,25
           SUM=0.
```

```
          DO8I=1,N
          SUM=SUM+W(I)*X(L,I)
8         CONTINUE
          ESO=(D(L)-SUM)**2
          Y(L)=SUM
          SUMSQ=SUMSQ+ESQ
          WRITE(6,205) Y(L),D(L),L
205       FORMAT(26X,F15,6,8X,F15.6,12X,I6)
7         CONTINUE
          EMSO=SUMSQ/FLOAT(L-1)
          WRITE(6,206) L
206       FORMAT(//30X,27HTIME AVERAGE SQUARED ERROR,,
         CI5,14HOUTPUT SAMPLES)
          WRITE(6,207) EMSQ
207       FORMAT(50X,F15,6/////)
1         CONTINUE
          STOP
          END
```

```
C          RECONSTRUCTING A RECTANGULAR PULSE DISTORTED
C          BY ADDITIVE WHITE GAUSSIAN NOISE USING BOTH
C          FEEOFWO ANO FEEDBACK ADAPTIVE FILTERING
           DIMENSION X(600),Y(600),D(600),B(14),A(14)
          C,AINTL(14),BINTL(14)
           COMMON C(600)
           CALL NORMAL
           READ(5,100)MA,NB,ICYC,ADRATE,IREPET,
          C(AINTL(J),J=1,NA),(BINTL(J),J=1,NB)
100        FORMAT(4I5,F10.6/(7F10.6))
           DO1NM=1,6
           DO23J=1,NA
           A(J)=AINTL(J)
23         CONTINUE
           DO24J=1,NB
           B(J)=BINTL(J)
24         CONTINUE
           WRITE(6,210)
210        FORMAT(10X,2HNA,9X,2HNB,8X,4HICYC,10X,6HIREPET,
          C10X,6HADRATE,9X,16HNOISE MULTIPLIER)
           WRITE(6,209)NA,NB,ICYC,IREPET,ADRATE,NM
209        FORMAT(/8X,I3,8X,I3,8X,I4,11X,I3,10X,F10.6,
          C10X,I10//)
           WRITE(6,212)
212        FORMAT(50X,21HINITIAL WEIGHT VALUES/)
           WRITE(6,211)(A(K),K=1,NA),(B(K),K=1,NB)
211        FORMAT(10X,10F10.5)
           JJ=0
           DO4M=1,IREPET
           DO4L=1,25
           JJ=JJ+1
           SUM=0.
           IF(L.LE.10)GOTO60
           D(L)=0,
           X(L)=C(JJ)*FLOAT(NM)
           GOTO70
60         D(L)=20.0
           X(L)=20.0+C(JJ)*FLOAT(NM)
           IF(L.EQ.1)GOTO19
70         K=L-1
           DO5I=1,NB
           SUM=SUM+B(I)*D(K)
           K=K-1
           IF(K.EQ.0)GOTO20
5          CONTINUE
20         K=L-1
           DO6I=2,NA
           SUM=SUM+A(I)*X(K)
           K=K-1
           IF(K.EO.0)GOTO19
6          CONTINUE
19         SUM=SUM+A(1)*X(L)
           ERR=D(L)-SUM
           0011J=1,NB
           II=L-J
           IF(II.EO.0)GOTO17
```

```
          B(J)=B(J)-2.*ADRATE*ERR*D(II)
11        CONTINUE
17        DO12J=2,NA
          II=L-J+1
          IF(II.EQ.Ø)COTO18
          A(J)=A(J)-2.*ADRATE*ERR*X(II)
12        CONTINUE
18        A(1)=A(1)-2.*ADRATE*ERR*X(L)
4         CONTINUE
          WRITE(6,2Ø2)
2Ø2       FORMAT(//5ØX,19HFINAL WEIGHT VALUES)
          WRITE(6,2Ø3)(A(K),K=1,NA)
2Ø3       FORMAT(/1ØX,1ØF1Ø.5)
          WRITE(6,2Ø4)(B(K),K=1.NB)
2Ø4       FORMAT(//,(1ØX,1ØF1Ø.5))
          WRITE(6,2Ø5)
2Ø5       FORMAT(//3ØX,13HFILTER OUTPUT,1ØX,14HDESIRED
        C  OUTPUT.1ØX.1ØHTIME UNITS)
          SUMSQ=Ø.
          DO7L=1,25
          SUM=Ø.
          IF(L.EQ.1)GOTO21
          K=L-1
          DO8I=1,NB
          SUM=SUM+B(I)*Y(K)
          K=K-1
          IF(K.EO.Ø)GOTO22
8         CONTINUE
22        K=L-1
          DO1ØI=2.NA
          SUM=SUM+A(I)*X(K)
          K=K-1
          IF(K.EO.Ø)GOTO21
1Ø        CONTINUE
21        SUM=SUM+A(1)*X(L)
          ESQ=(D(L)-SUM)**2
          Y(L)=SUM
          SUMSQ=SUMSQ+ESQ
          WRITE(6,2Ø6) Y(L).D(L).L
2Ø6       FØRMAT(26X,F15.6.8X,F15.6,12X,I6)
7         CONTINUE
          EMSQ=SUMSQ/FLOAT(L-1)
          WRITE(6,2Ø7) L
2Ø7       FORMAT(//3ØX,27HTIME AVERAGE SQUARED ERROR,,
        C I5.14HOUTPUT SAMPLES)
          WRITE(6,2Ø8) EMSQ
2Ø8       FORMAT(5ØX,F15.6/////)
1         CONTINUE
          STOP
          END
```

```
      SUBROUTINE NORMAL
      COMMON C(4ØØ)
      DO3KK=1,4ØØ
      SUM1=Ø.
      DO4MM=1,49
      UNNM=RECDIS(X)
      SUM1=SUM1+UNNM
4     CONTINUE
      SMEAN=SUM1/49.Ø
      C(KK)=7.Ø*(SMEAN-.5)*SQRT(12.Ø)
3     CONTINUE
      RETURN
      END
```

REFERENCES

1. Hemmerle, W. J., Statistical Computations on a Digital Computer. Waltham, Massachusetts: Blaisdell Publishing Company, 1967, pp. 41-43.

2. Hogg, R. V., Craig, A. T., Introduction to Mathematical Statistics. New York: The MacMillan Company, 1965, pp. 196-200.

3. Kuo, F. F., Kaiser, J. F., System Analysis by Digital Computer. P. E. Fleischer, "Optimization Techniques in System Design". New York: John Wiley & Sons, Inc., 1966, pp. 175-215.

4. Mantey, P. E., "Convergent Automatic--Synthesis Procedures for Sampled-Data Networks with Feedback". Stanford Electronics Laboratory, Technical Report 6773-1, Stanford, California, October, 1964.

5. Papoulis, A., The Fourier Integral and Its Applications. New York: McGraw-Hill Book Company, Inc., 1962, Chapters 2, 3 and 6.

6. Papoulis, A., Probability, Random Variables, and Stochastic Processes. New York: McGraw-Hill Book Company, Inc., 1965, Chapter 14.

7. Van Trees, H. L., Detection, Estimation, and Modulation Theory. New York: John Wiley & Sons, Inc., 1968, Chapters 3, 4.

8. Widrow, B., "Adaptive Filters I: Fundamentals". Stanford Systems Theory Laboratory, Technical Report 6764-6, Stanford, California, December, 1966.

9. Pufft Compiler Reference Supplement, The Ohio State University Computer Center, February 1, 1967, pp. 2.005-2.006.