TRANSFER FUNCTION CONSIDERATIONS OF AN ADAPTIVE LATTICE PREDICTOR

by

YUNG-NING WANG

B. S., National Taiwan Institute of Technology, 1977

A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Electrical Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1981

Approved by:

[illegible]
Major Professor

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGMENT

I would like to take this opportunity to thank all those who assisted me during my study. I would especially like to express my deepest thanks and appreciation to Dr. N. Ahmed as my advisor to suggest the topic of this report and who provided both guidance and encouragement, along with Dr. Floyd W. Harris and Dr. Louis M. Herman for serving on my committee. I am also grateful to the Electrical Engineering Department of Kansas State University and the Sandia Laboratories (Organization 1738), Albuquerque, New Mexico, for their financial support.

Finally, I would like to thank Mr. Dae Hee Youn for his assistance and encouragement.

1. INTRODUCTION

The ADP (Adaptive Digital Predictor) was implemented using Widrow's LMS algorithm [1] as shown in Fig. 1, in connection with an intruder detection application [2]. After the algorithm was used for a period of time, it was observed that the predictor would not only remove correlated noise, but the intruder signal as well. In other words, it gradually became a "no pass" filter.

The above "no pass" phenomenon is best conveyed by a simple experiment that was performed in a laboratory environment. The input to a 16 weight ADP (see Fig. 1) consisted of a sum of two sinusoids of frequencies 8 and 32 HZ. The sampling frequency was 128 HZ and convergence parameter ν in Fig. 1 was 0.02. If both the sinusoids were generated by repeatedly using the exact sample set of the single cycle, the ADP output was essentially zero and the transfer function of its filter portion remained unchanged indefiniteley. However, if the sinusoids were generated using the FORTRAN "SIN(A)" Function, where A was varied continuously, the transfer function was found to change slowly with time. This is depicted in Fig. 2 via gain plots. It is apparent that the gain function changes gradually from a narrow-band type with two peaks at 8 and 32 HZ to a more broadband type of function. We also note that the gains at 8 and 32 HZ remained equal to 1 throughout the transition period. As a result, the corresponding ADP output was zero. A related study [3] has shown that this behavior is due to the ADP having more weights than necessary, as was the case in the above experiment, where only 4 weights are necessary to eliminate a sum of two sinewaves. The extra weights strive to remove

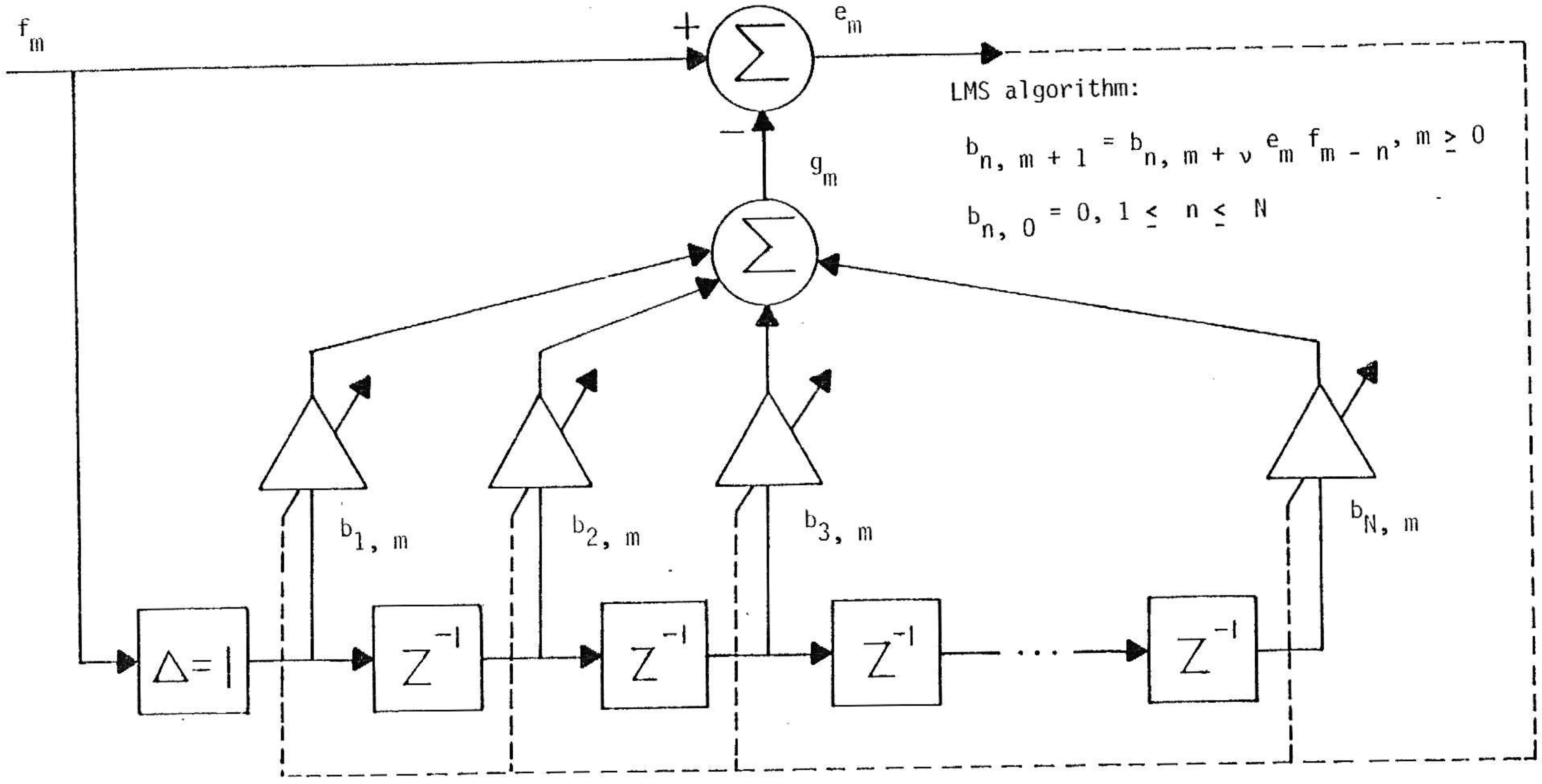


Fig.1. Tapped delav line(or transversal filter) predictor configuration. Dashed lines indicate adaptive control loop.

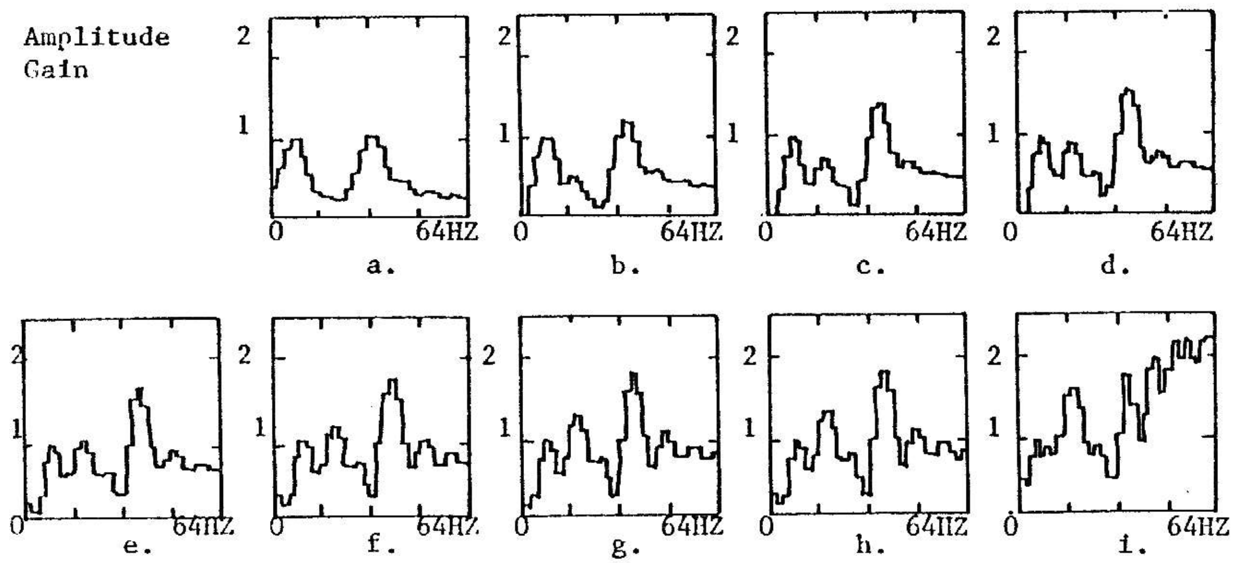


Fig.2. Absolute value of transfer function of digital filter in the adaptive predictor.Frames a. through h. give the amplitude gain every 32,000 iterations,frame i is after approximately one million iterations;input is a slightly noisy combination of 8- and 32-HZ sinusoids;(this figure was taken from[2]).

very low level components in the inputs, and hence could implement an overall input-output gain function that removes the input power over the entire passband. The predictor would then tend to become a "no pass" filter. One way of avoiding the above no-pass phenomenon is to modify the LMS algorithm. The pertinent modification is very simple, as discussed in Section 2.

The main objective of this report is to examine the behavior of the transfer function of the ADP when it is implemented in the form of a lattice (see Fig. 3) instead of the transversal filter implementation shown in Fig. 1. This is done via a computer simulation, some details related to which are given in Section 5, while Section 3 is devoted to a brief discussion of linear LMS lattice algorithm. In Section 4, the relation between lattice and transversal filters is derived. Experiment results are presented in Section 5. Some conclusions are included in Section 6.

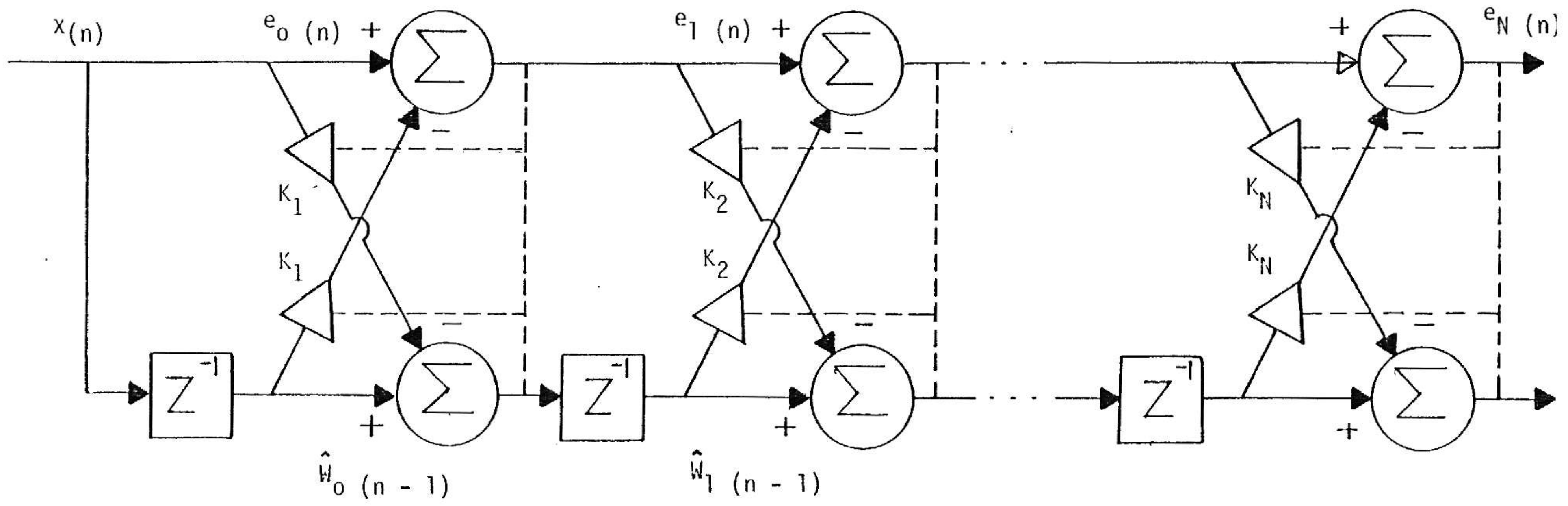


Fig.3. one-step delay lattice predictor ;

dashed lines indicate adaptive control loop

2. PREDICTION CONSIDERATIONS

In Fig. 1, suppose the predictor coefficient b_i are fixed, and Δ is the delay parameter. The optimum predictor weights are obtained by minimizing the mean-squared error (mse)

$$\overline{e}_m^{\,2} = E\,[e_m^{\,2}] = E\{f_m - g_m)^2\} \tag{1}$$

with respect to b_i, where E denotes statistical expectation. This minimization process leads to the following set of equations [4]

$$\sum_{j=1}^{N} r_{ij}\, b_j = r_{io}, \qquad i = 1, 2, 3, \ldots, N \tag{2}$$

where $r_{ij} = E\,\{f_{m-i}\, f_{m-j}\}$ denote the covariances of the input.

Equation (2) can be written as

$$R\,B = C$$

where $B^T = [b_1, b_2, b_3 \ldots b_N]$ is the weight factor, $C^T = [r_{10}, r_{20}, \ldots, r_{N0}]$ is an $(N \times 1)$ cross correlation vector

and

$$R = \begin{pmatrix} r_{11} & r_{12} & \cdots & \cdots & r_{1N} \\ \vdots & r_{22} & & & \vdots \\ \vdots & \cdots & r_{33} & \cdots & \vdots \\ \vdots & & & \ddots & \vdots \\ r_{N1} & \cdots & \cdots & \cdots & r_{NN} \end{pmatrix}$$

is an $(N \times N)$ correlation matrix.

Thus, the weight vector of the optimum predictor is given by

$$B = R^{-1} C \; . \tag{4}$$

Substituting (2) in (1) we get

$$\overline{e}_m^{\,2} = r_{oo} - [b_1 r_{10} + b_2 r_{20} + \dots + b_N r_{NO}] \; . \tag{5}$$

Next, we consider the case when the predictor weights b_n change with time. Then, corresponding to (1), we have

$$g_m = b_{1,\,m} f_{m-1} + b_{2,m} f_{m-2} + \dots + b_{N,m} f_{m-N} \tag{6}$$

where $b_{n,\,m}$ is the nth weight at time m. Using the method of steepest descent, Widrow [1] has shown that the weights $b_{n,\,m}$ can be updated via the following LMS algorithm:

$$b_{n,\,m+1} = b_{n,\,m} + \nu\, e_m f_{m-n} \tag{7}$$

where e_m is the prediction error at time m; ν is a convergence parameter which must satisfy the condition [1]

$$0 < \nu < \frac{2}{\lambda_{max}}$$

where λ_{max} denotes the largest eigenvalue of R. It is known that $\overline{e}_m^{\,2}$ in (1) is quadratic function of the weights b_i, and hence the possibility of a local minimum is eliminated. However, a distributed minimum [3] is possible,and it is this type of minimum that could result in a variety of transfer functions, with the same ADP output. To illustrate, we consider the simple case when the predictor in Fig. 1 consists of two weights [3]. Then,

$$H(z) = b_1 z^{-1} + b_2 z^{-2} . \qquad (8)$$

Let the input be the alternating sequence

$$\{f_m\} = \{1, -1, 1, -1, \ldots\}. \qquad (9)$$

Then

$$r_{ij} = (-1)^{|i - j|} .$$

Substitution of (9) in (5) leads to

$$\overline{e}_m{}^2 = (1 + b_1 - b_2)^2, \text{ or } \overline{e}_{rms} = |1 + b_1 - b_2| \qquad (10)$$

From $\overline{e}_{rms}$ in (10), it is clear that there is a distributed minimum along the <u>line</u> $1 + b_1 - b_2 = 0$ rather than a single minimum [3]. As such, the corresponding least-squares solution is no longer unique, which implies that the autocorrelation matrix in (4) will not have full rank. If this relationship is substituted into the overall function

$$H_0(z) = \frac{E(z)}{F(z)}$$

we obtain

$$H_0(z) = 1 - H(z) = 1 - b_1 z^{-1} + (1 - b_1)\, z^{-2} \qquad (11)$$

[illegible]

It is apparent that H(z) has a zero at z = 1 and another somewhere on the real axis of the z-plane, depending upon the value of b_1. From the above discussion it follows that it is the surplus weight b_2 that causes $\overline{e}_m$ in (10) to have a distributed minimum, since only b_1 is sufficient to predict the sequence $\{f_m\}$ in (9). One method of avoiding the distributed minimum and hence the above problem is to modify the cost function in (1) as follows

$$C = E\,\{e_m{}^2\} + W\, B_m'\, B_m \qquad (12)$$

where $B_m' = [b_{1,m}\ b_{2,m} \cdots b_{N,m}]$ and $0 < W < 1$. Minimizing of (12) with respect to B_m, we obtain [3]

$$B_{opt.} = [WI + R]^{-1}\, P_{gf} \tag{13}$$

where I is the identity matrix and P_{gf} represents the cross correlation between f and g. Again, using the method of steepest descent, the corresponding LMS algorithm is given by [3]

$$\begin{aligned} B_{m+1} &= (1 - W)\, B_m + \nu\, e_m\, F_m \\ &= \mu\, B_{n,m} + \nu\, e_m\, F_m \end{aligned} \tag{14}$$

where $\mu = 1 - W$, and $F_m' = [f_{m-1}, f_{m-2} \cdots f_{m-N}]$.
Equation (14) is known as the modified (MLMS) algorithm [3].

To further examine the behavior of the MLMS algorithm, let us assume that for some $m \geq M$, the error term in (14) is negligible; i.e., at time $m = M$, the weight b_n has converged to some least-squares-type solution $b_{n,M}$. Then (14) simplifies to yield

$$\begin{aligned} b_{n,M+1} &= (1 - W)\, b_{n,M} \\ b_{n,M+2} &= (1 - W)^2\, b_{n,M} \\ &\vdots \\ b_{n,M+K} &= (1 - W)^K\, b_{n,M}\,. \end{aligned}$$

It is apparent that $b_{n,M+K}$ decays toward 0 with increasing values of K, since $(1 - W)$ is less than 1. This causes e_m in (14) to increase and hence the ADP begins to readapt. The adaption process causes b_n to move toward $b_{n,M}$ until e_m is negligible. Then the entire foregoing cycle repeats. The overall effect is that a distributed minimum does not occur, and a single solution is obtained. Hence the "no pass" phenomenon is avoided.

3. THE LMS LATTICE ALGORITHM

Figure 3 shows a fixed (non adaptive) lattice predictor. The corresponding forward prediction error is defined as the difference between the predicted value of the input one step into the future, and its actual value, i.e.,

$$e_\ell (n) = \chi(n) - \hat{\chi}(n)$$
$$\hat{\chi}(n) = - \sum_{\ell=1}^{N} d_{\ell,\, N} \ \chi(n - \ell), \ \ell = 1, 2, \ldots, N \tag{16}$$

where $\hat{\chi}(n)$ denotes an estimate of $\chi(n)$, $d\ell$, N is the ℓ-th forward predictor coefficient weight of an N-weight predictor. On the other hand, a backward prediction error is defined as the difference between the predicted value of the input one step into the past, and its actual value: i.e.,

$$\hat{W}_\ell (n) = \chi(n - \ell - 1) - \hat{\chi}(n - \ell - 1) \tag{17}$$

In (17), $\hat{\chi}(n - \ell - 1)$ denotes an estimate of $\chi(n - \ell - 1)$, and is expressed as

$$\hat{\chi}(n - \ell - 1) = -\sum_{\ell=1}^{N} C_{\ell,\, N} \ \chi(n - \ell) \tag{18}$$

Where $C_{\ell,\, N}$ represents the ℓ-th backward predictor weight of an N-weight predictor. A forward and a backward ADP may be combined to obtain the lattice structure shown in Fig. 3. Details of the development can be found elsewhere; e.g., see [5]. This structure is defined recursively as follows:

$$\chi(n) = e_o(n) = \hat{W}_o(n)$$
$$e_\ell(n) = e_{\ell-1}(n) - K_\ell \hat{W}_{\ell-1}(n-1) \qquad (19)$$
$$\hat{W}_\ell(n) = \hat{W}_{\ell-1}(n-1) - K_\ell e_{\ell-1}(n)$$

where K_ℓ is known as the ℓ-th lattice coefficient. Now, the total prediction error at the ℓ-th stage is given by

$$S_\ell^2(n) = e_\ell^2(n) + \hat{W}_\ell^2(n) \qquad (20)$$

Minimizing $S_\ell^2(n)$ with respect to K_ℓ we have

$$\frac{\partial^2 S_\ell(n)}{\partial K_\ell} = 0$$

which yields

$$K_\ell = \frac{2E\{e_{\ell-1}(n)\hat{W}_{\ell-1}(n-1)\}}{E\{e_{\ell-1}^2(n)\} + E\{\hat{W}_{\ell-1}^2(n-1)\}} \quad . \qquad (21)$$

If the input to the lattice predictor in Fig. 3 is nonstationary, then the lattice weights will have to be made time-varying. To this end, the method of steepest descent is used and K_ℓ are updated as follows [6, 7]:

$$K_\ell(n+1) = K_\ell(n) - \hat{\mu}\frac{\partial S_\ell^2(n)}{\partial K_\ell(n)} \qquad (22)$$

where $\hat{\mu}$ is a convergence parameter.

From (19) and (20) it follows that

$$\frac{\partial S_\ell^2 (n)}{\partial K_\ell (n)} = 2 e_\ell (n) \frac{\partial e_\ell (n)}{\partial K_\ell (n)} + 2 \hat{W}_\ell (n) \frac{\partial \hat{W}_\ell (n)}{\partial K_\ell (n)} \qquad (23)$$

where

$$\frac{\partial e_\ell (n)}{\partial K_\ell (n)} = - \hat{W}_{\ell-1} (n - 1)$$

and

$$\frac{\partial \hat{W}_\ell (n)}{\partial K_\ell (n)} = - e_{\ell-1} (n). \qquad (24)$$

Substitution of (23) and (24) in (22) leads to $K_\ell (n + 1) = K_\ell (n) + 2 \hat{\mu} [e_\ell (n) \hat{W}_{\ell-1} (n - 1) + \hat{W}_\ell (n) e_{\ell-1} (n)]$. (25)

Now, the power in the forward and backward error sequences decreases as the number of stages is increased. Thus, in order to maintain the same adaptive time constant at each stage in the lattice, the step size must be normalized by the input power at that stage [7]. To this end, we estimate the power of the ℓ-th stage using the relation

$$\sigma_\ell^2 (n) = \beta \sigma_\ell^2 (n - 1) + (1 - \beta) [e_{\ell-1}^2 (n) + \hat{W}_{\ell-1}^2 (n - 1)] \qquad (26)$$

where $0 < \beta < 1$ is a smoothing parameter. Thus (25) becomes

$$K_\ell (n + 1) = K_\ell (n) + \frac{\alpha}{\sigma_\ell^2 (n)} [e_\ell (n) \hat{W}_{\ell-1} (n - 1) + \hat{W}_\ell (n) e_{\ell-1} (n)] \qquad (27)$$

for $\ell = 1, 2, 3, \ldots, N$, where α is a constant and $\sigma_\ell^2 (n)$ is estimated via (26). The final prediction error is e_N (n) in Fig. 3, which corresponds to e_m in Fig. 1.

Comment: In practice, the algorithm in (27) has to be slightly modified to account for the case when the input to lattice is very small. Then from (26) it follows that σ_ℓ^2 (n) would be essentially zero, and hence causes the algorithm to become unstable. This undesirable condition can be avoided by not carrying out the updating in (27) if $\sigma_\ell^2 (n) < \varepsilon$ where ε is a small positive number whose value depends on the word length involved. For example, if the ALP algorithm is implemented in fixed point using a 16-bit computer, we let

$$\varepsilon = \frac{2^{-13}}{2} = .000061 .$$

Then (27) becomes

$$K_\ell (n + 1) = K_\ell (n) + \frac{\alpha W}{\sigma_\ell^2(n)} [e_\ell (n) \hat{W}_{\ell-1} (n - 1) + \hat{W}_\ell (n) e_{\ell-1} (n)] \tag{28}$$

$$\text{where} \begin{cases} W = 1 & \text{if } \sigma_\ell^2 (n) \geq \varepsilon \\ W = 0 & \text{if } \sigma_\ell^2 (n) < \varepsilon . \end{cases}$$

We refer to (28) as the LMS lattice algorithm, where σ_ℓ^2 (n) is updated using (26).

4. RELATION BETWEEN LATTICE AND TRANSVERSAL FILTER COEFFICIENTS

Given a set of lattice predictor coefficients, it is useful to obtain the corresponding transversal filter predictor coefficients. This would enable one to evaluate the overall input-output transfer function $\frac{E_N(z)}{X(z)}$ of the lattice predictor in Fig. 3. Thus, from (16) we have

$$\hat{x}(n) = -[d_{1,\,N}\, x\,(n - 1) + d_{2,\,N}\, x\,(m - 2) + \ldots +$$

$$d_{N,\,N}\, x\,(n - N)]$$

where

$$e_N\,(n) = x(n) - \hat{x}(n)\ .$$

That is

$$E_N\,(z) = x\,(z)\, D_N\,(z) \tag{29}$$

where

$$D_N\,(z) = \sum_{\ell=0}^{N} d_{\ell,\,N}\, z^{-\ell} \text{ with } d_{o,\,N} = 1\ . \tag{30}$$

Minimization of $e_N^{\,2}\,(n)$ with respect to $d_{\ell,\,N}$ leads to

$$d_N = -R_N^{\,-1}\, r_N \tag{31}$$

where

$$d_N' = [d_{1,N} \; d_{2,N} \cdots\cdots d_{N,N}] ,$$

R_N is the input autocorrelation matrix, and given by

$$R_N = \begin{pmatrix} R_{xx}(0) & R_{xx}(1) & \cdots & R_{xx}(N-1) \\ R_{xx}(1) & R_{xx}(0) & \cdots & R_{xx}(N-2) \\ \vdots & & R_{xx}(0) \ddots & \vdots \\ & & & \\ R_{xx}(N-1) & \cdots & \cdots & R_{xx}(0) \end{pmatrix}$$

and $r_N' = [R_{xx}(1) \; R_{xx}(2) \cdots R_{xx}(N)]$.

Again, Eq. (17) and Eq. (18) imply that

$$W_N(z) = X(z)\, C_N(z) \tag{32}$$

where

$$C_N(z) = \sum_{\ell=1}^{N+1} C_{\ell,N}\, z^{-\ell} , \text{ with } C_{N+1,N} = 1 .$$

Minimization of $E\{W_N^2(n)\}$ with respect to the $C_{\ell,N}$ results in

$$C_N = -R_N^{-1} S_N . \tag{33}$$

where

$$C_N' = [C_{1,N} \; C_2 \cdots\cdots C_{N,N}]$$

and

$$S_N' = [R_{xx}(N) \quad R_{xx}(N-1) \text{ ------ } R_{xx}(1)] .$$

From (31) and (33) the following relations are apparent

$$r_N = M_N S_N \qquad S_N = M_N r_N$$
$$C_N = M_N d_N \qquad d_N = M_N C_N \tag{34}$$

where

$$M_N = \begin{pmatrix} 0 & & 1 \\ & \cdot^{\cdot^{\cdot}} & \\ 1 & & 0 \end{pmatrix} .$$

Equation (34) represents the set of equations

$$C_{\ell,N} = d_{N+1-\ell,N} . \tag{35}$$

Next, consider an (N + 1) weight forward predictor. Then, corresponding to (33), we obtain

$$\left(\begin{array}{c|c} R_N & u_N \\ \hline u_N' & R_{xx}(0) \end{array}\right) d_{N+1} = - \left(\begin{array}{c} r_N \\ \hline R_{xx}(N+1) \end{array}\right) \tag{36}$$

where

$$d_{N+1}' = [d_{1,N+1} \quad d_{2,N+1} \text{ ----- } d_{N+1,N+1}]$$

$$u_N' = [R_{xx}(N) \ R_{xx}(N-1) \text{ ------ } R_{xx}(1)]$$

$$r_N' = [R_{xx}(1) \quad R_{xx}(2) \text{ ------ } R_{xx}(N)]$$

Now, matrix bordering [8] enables us to write down the following solution for d_{N+1} in (36):

$$d_{N+1} = \begin{pmatrix} d_N \\ \hline 0 \end{pmatrix} - K_{N+1} \begin{pmatrix} -R_N^{-1} u_N \\ \hline 1 \end{pmatrix} \tag{37}$$

where

$$K_{N+1} = \frac{R_{xx}(N+1) + u_N' d_N}{R_{xx}(0) - u_N' R_N^{-1} u_N} ,$$

is a scalar. We observe that $u_N = S_N$ and hence (35) implies that the term in (37) can be replaced by C_N. Thus,

$$d_{N+1} = \begin{pmatrix} d_N \\ \hline 0 \end{pmatrix} - K_{N+1} \begin{pmatrix} C_N \\ \hline 1 \end{pmatrix} . \tag{38}$$

From (35) it follows (38) yields the recursive relation

$$d_{\ell, N+1} = d_{\ell, N} - K_{N+1} d_{N+1-\ell, N} , \ell = 1, 2, \ldots, N$$

and

$$d_{N+1, N+1} = -K_{N+1} . \tag{39}$$

Equation (39) is the desired relation which relates to lattice coefficients K_ℓ and transversal filter coefficients $d_{\ell,N}$. A computer program which implements the LMS lattice algorithm and also converts the lattice coefficients into corresponding transversal filter coefficients via (39) is given in Appendix B.

5. EXPERIMENTAL RESULTS

Six experiments were conducted, and in each case the input consisted of the sum of 8 Hz and 32 Hz sinusoids with equal amplitude. The sampling frequency was 128 sps.

Experiment 1

Widrow's LMS algorithm was used to implement a 16 weight predictor as indicated in Fig. 1 i.e.,

$$b_{m,\, n+1} = b_{m,\, n} + \nu e_n f_{n-m}, \quad 1 \leq m \leq 16 \qquad (40)$$

where $b_{m,\, n}$ is the m-th weight at time n

ν is the convergence parameter

e_n is the error at time n, and

f_n is input to the predictor at time n.

The overall steady-state gain function

$$|\overline{H}_0 (f)| = \frac{|E\ (e^{j2\pi fT})|}{|F\ (e^{j2\pi fT})|}\ , \quad 0 \leq f \leq 64$$

where T = 1/128 seconds was computed and plotted at various intervals, as depicted in Fig. 4. The value of ν in (40) was 0.02.

From Fig. 4 it is clear that the predictor adapts to the 8 and 32 Hz components and removes them, since nulls appear at these two frequencies.

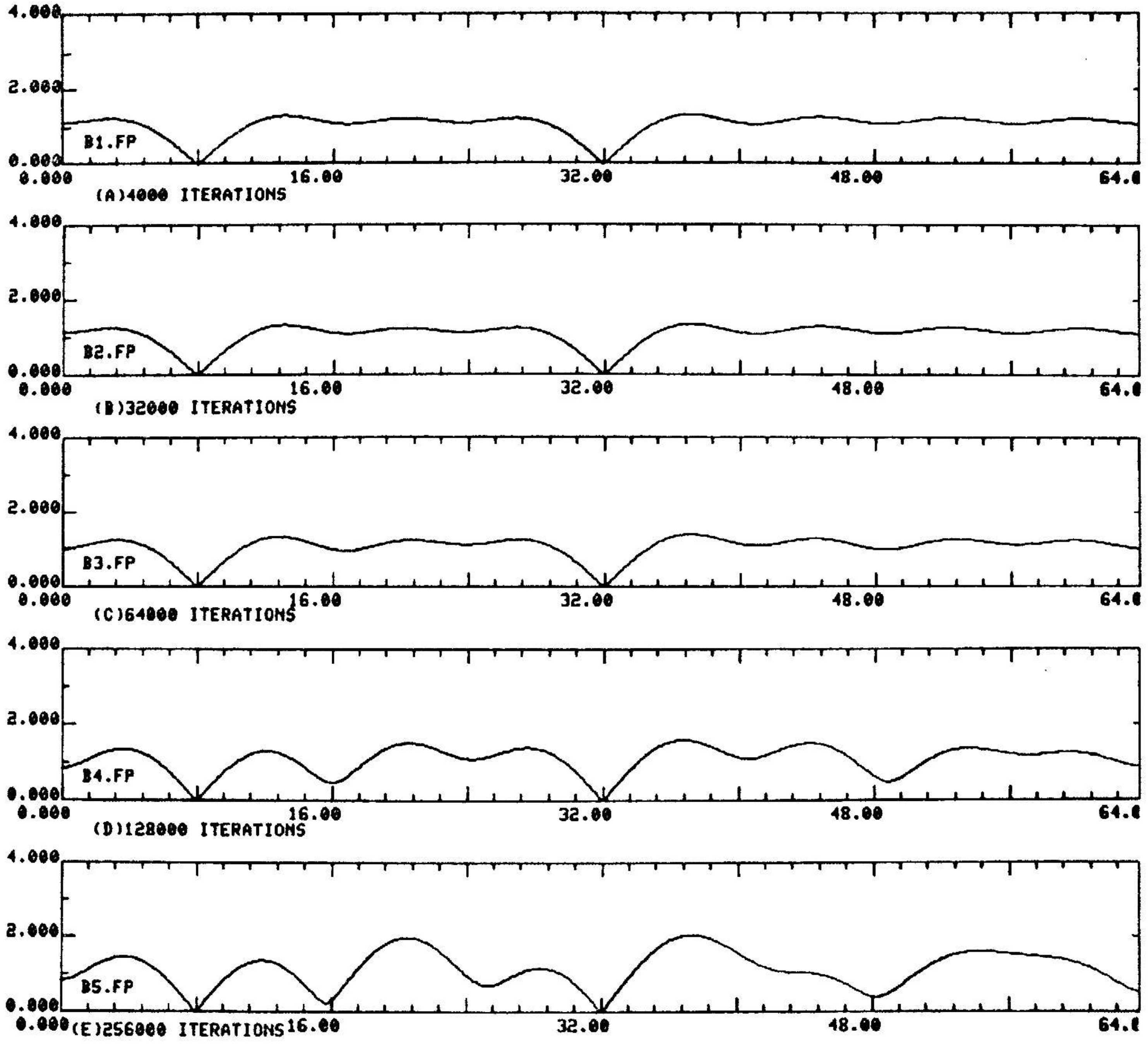


FIG.4. OVERALL GAIN FUNCTION OBTAINED VIA EXPERIMENT 1

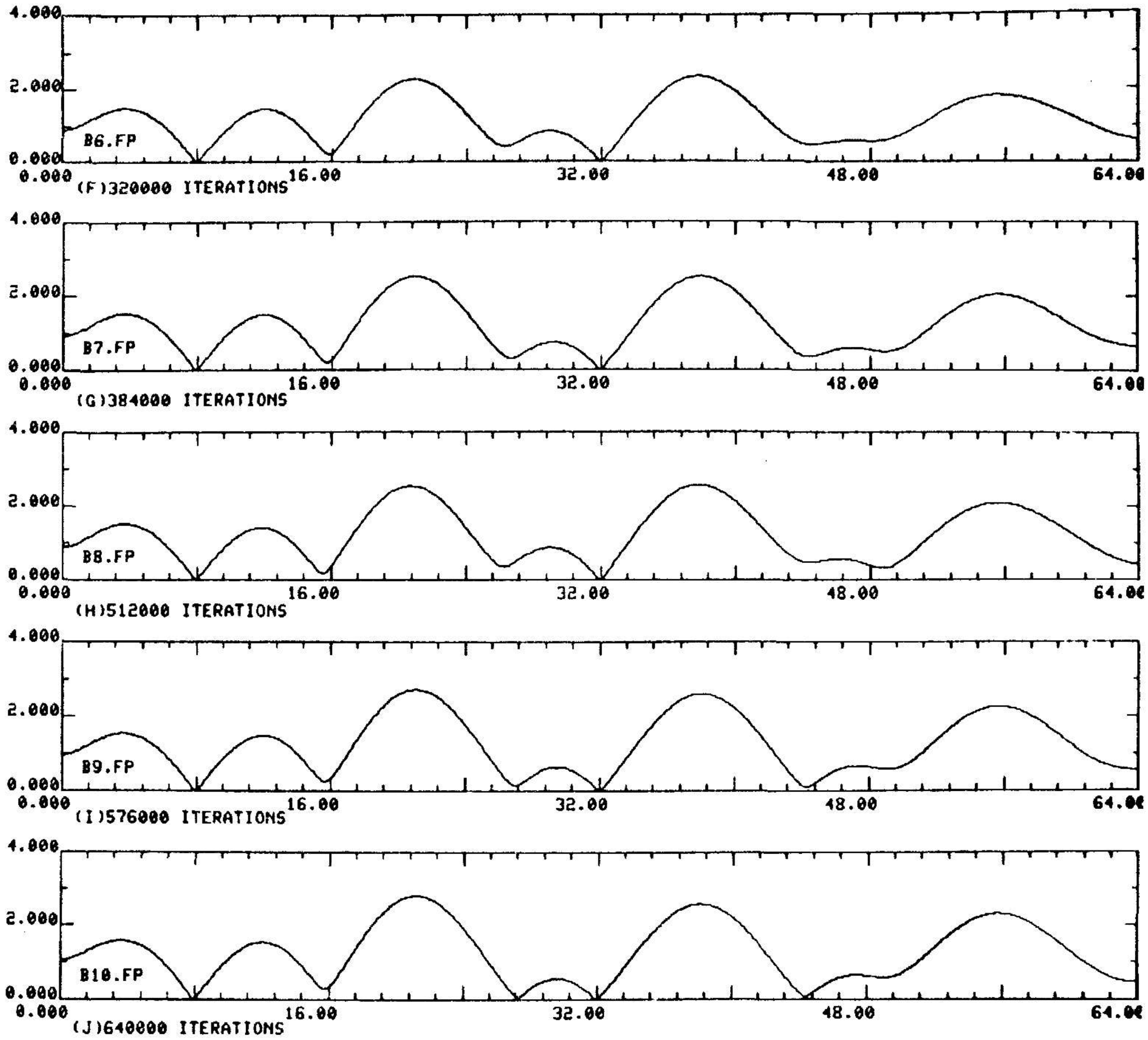


FIG.4.(CONTINUED)

Any input components at all other frequencies in the 0 - 64 HZ range will essentially be passed unattenuated. Clearly, this is basically an "ideal" overall transfer function. However, this desired shape of the overall transfer function gradually deteriorates as time goes on, as evident from Fig. 4(j). This is because the gain function in Fig. 4(j) has several more nulls than the desired ones at 8 and 32 HZ. Thus input frequencies at the additional nulls would also be eliminated if they were present. This is equivalent to the filter portion of the predictor having an approximately "flat" gain function as verified in Fig. 5. This observation agrees with that reported in [3]. More information regarding this experiment is summarized in Table 1 for the interested reader.

Experiment 2

In this experiment, Widrow's modified LMS (MLMS) algorithm was used. This algorithm is given by [3]

$$b_{m,\, n+1} = \mu\, b_{m,\, n} + \nu\, e_n\, f_{n-m} \cdot \quad 1 \leq m \leq 16 \tag{41}$$

where $0 < \mu < 1$. The value of this constant was set equal to $0.999878 \simeq 1 - 2^{13}$, which is the closest value to unity for a fixed-point implementation of the above algorithm in a 16 bit processor [3].

The overall gain function $|\overline{H}_0\ (f)|$ obtained via (41) is as shown in Fig. 6(a). We observe that gain function in Fig. 6 at $n = 4000$ compares closely with the "ideal" one in Fig. 4(a), but the basic difference in this case is that the gain function retains its original shape as time goes on; i.e. frequencies other than those at 8 HZ and 32 HZ are not eliminated. This is evident from the plots related to $n = 4000$ and $n = 640{,}000$ in Fig. 6(a) and 6(j), respectively. Additional information related to this

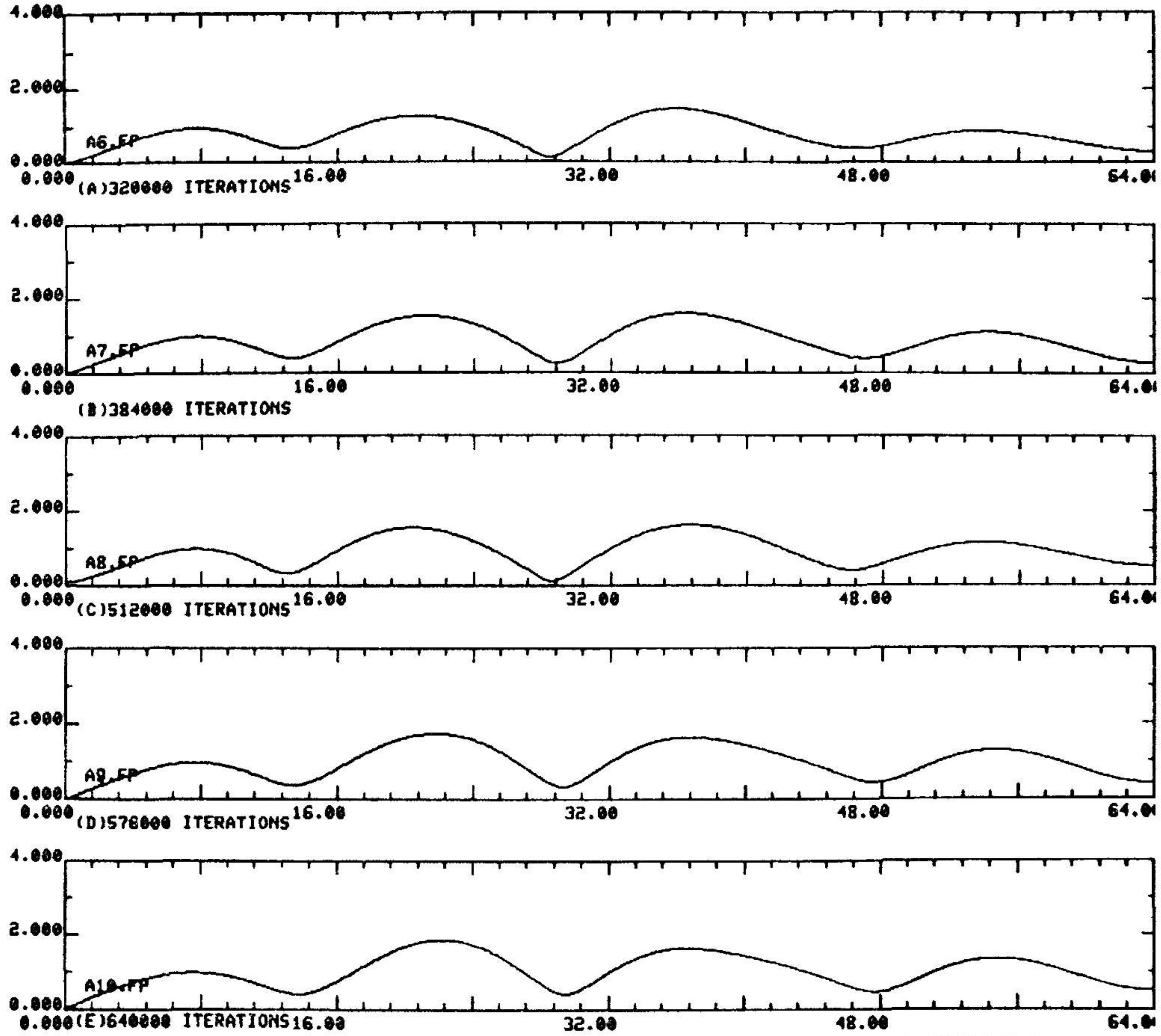


FIG.5. SOME GAIN FUNCTION PLOTS OF FILTER PORTION OF PREDICTOR CONSIDERED IN EXPERIMENT 1

TABLE 1: ADAPTIVE WIDROW FILTER (U = 1, FIXED V); EXPERIMENT 1

Iteration	Weights of Filter Portion of Predictor
4000	(1) = .17479
	(2) = -.06905
	(3) = .07417
	(4) = .17028
	(5) = -.08568
	(6) = -.28011
	(7) = -.13209
	(8) = -.00247
	(9) = -.13018
	(10) = -.20540
	(11) = -.03263
	(12) = .10641
	(13) = .03364
	(14) = -.02081
	(15) = .08305
	(16) = .15048
640000	(1) = .03545
	(2) = -.13022
	(3) = .11036
	(4) = .08408
	(5) = .05762

TABLE 1: Continued

Iteration	Weights of Filter Portion of Predictor
640000	(6) = -.20908
	(7) = .69342
	(8) = .44231
	(9) = .03518
	(10) = -.07804
	(11) = -.15665
	(12) = -.05674
	(13) = -.27485
	(14) = .02987
	(15) = .59242
	(16) = .14589

Note: All filter weights were initially set to zero.

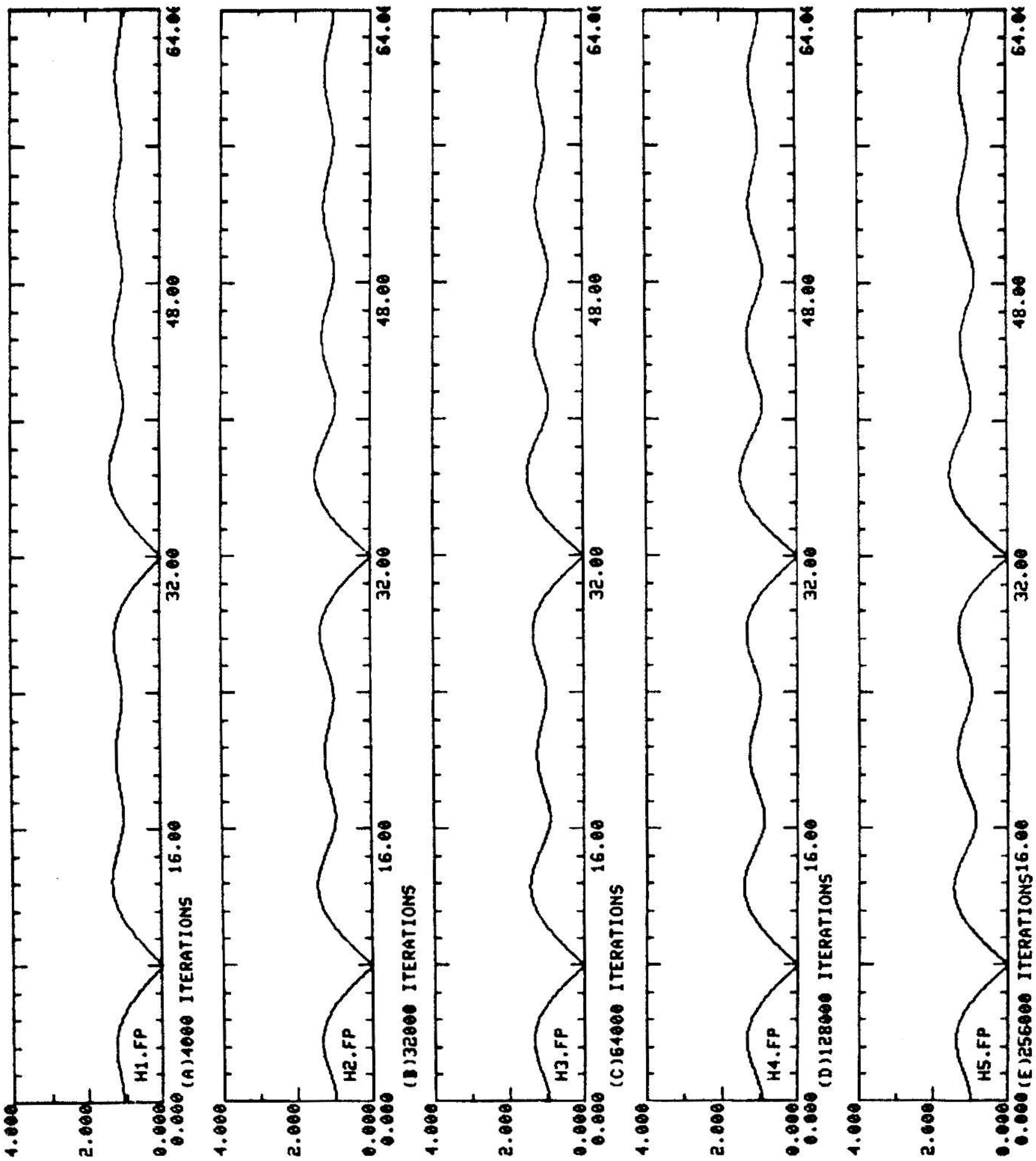


FIG.6. OVERALL GAIN FUNCTION OBTAINED VIA EXPERIMENT 2

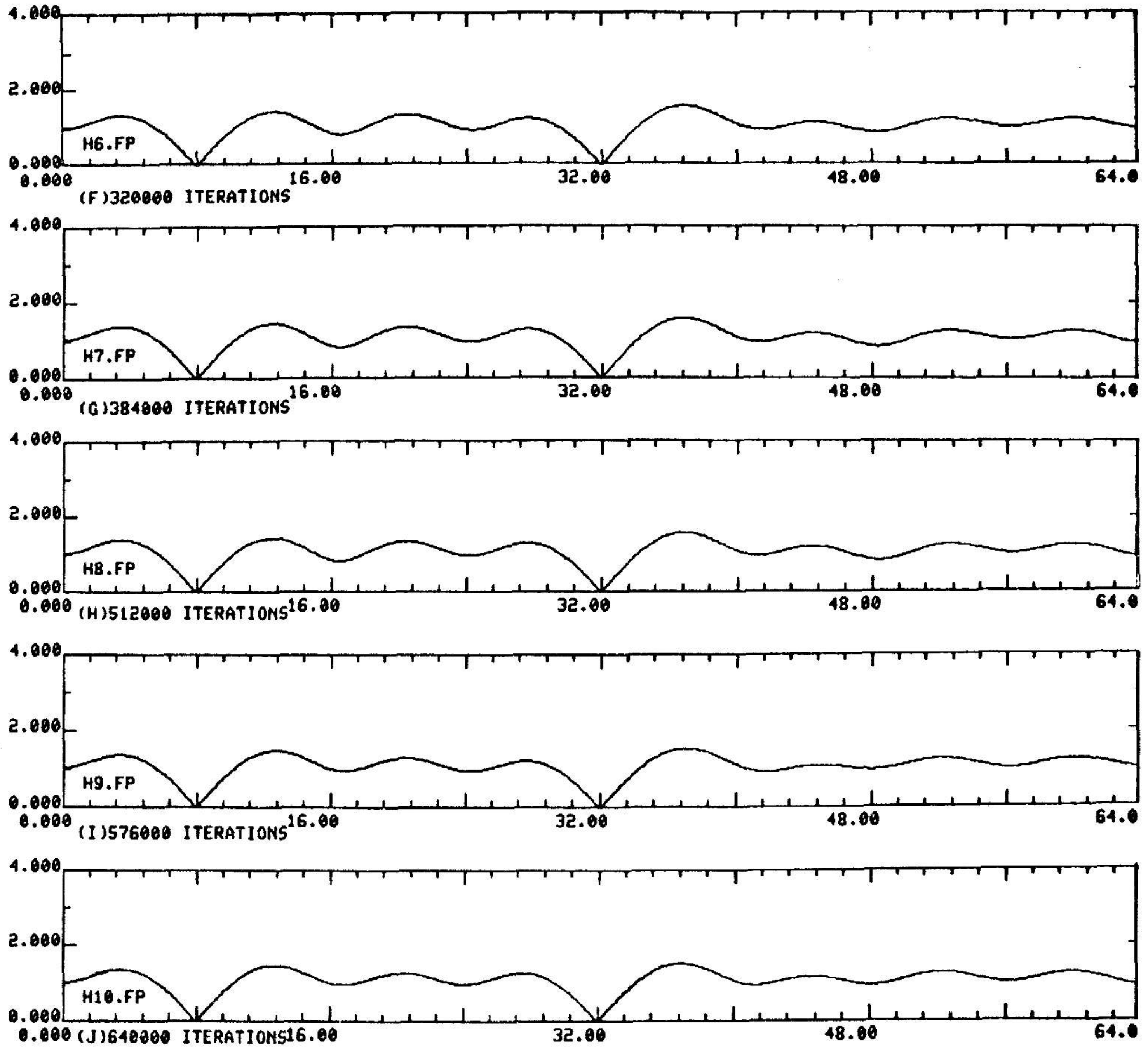


FIG.6.(CONTINUED)

experiment is summarized in Table 2.

Experiment 3

This experiment concerns the LMS lattice algorithm in (28). The values of the parameters used were as follows:

$$\alpha = 0.02 \ , \quad \beta = 0.98 \quad \text{and} \quad \varepsilon = 0.000061.$$

The overall gain functions obtained are displayed in Fig. 7. We note that the gain function has additional nulls at 16, 48 and 56 HZ in Fig.7(a) which corresponds to n = 4000. However, as *time goes on*, the overall gain function changes and these additional notches tend to disappear, e.g. see Fig. 7(j). Some additional details related to this experiment are given in Table 3.

Experiment 4

Here we modify the LMS lattice algorithm in (28) to obtain the following modified algorithm:

$$k_\ell \ (n+1) = \mu \ K_\ell \ (n) + \frac{\alpha \ W}{\sigma_\ell^2 \ (n)} [e_\ell \ (n) \ \hat{W}_{\ell-1} \ (n-1) + \hat{W}_\ell \ (n) \ e_{\ell-1} \ (n)] \tag{42}$$

where $\begin{cases} W = 1 & \text{if } \sigma_\ell^2 \ (n) \geq \varepsilon \\ W = 0 & \text{if } \sigma_\ell^2 \ (n) \leq \varepsilon \end{cases}$

and μ is again an additional parameter. As was the case with the MLMS algorithm, we set $\mu = 0.999878$.

TABLE 2: ADAPTIVE WIDROW FILTER (U = .999878, FIXED V); EXPERIMENT 2

Iteration	Weights of Filter Portion of Predictor
4000	(1) = .15168
	(2) = -.05648
	(3) = .06391
	(4) = .15261
	(5) = -.07098
	(6) = -.25412
	(7) = -.12545
	(8) = -.00152
	(9) = -.12438
	(10) = -.20822
	(11) = -.03852
	(12) = .11335
	(13) = .03901
	(14) = -.02687
	(15) = .09540
	(16) = .18854
640000	(1) = .10171
	(2) = -.04834
	(3) = .04983
	(4) = .12360
	(5) = -.03388

TABLE 2: Continued

Iteration	Weights of Filter Portion of Predictor
640000	(6) = -.20390
	(7) = -.15396
	(8) = .01805
	(9) = -.11230
	(10) = -.22540
	(11) = -.03958
	(12) = .12297
	(13) = .02271
	(14) = -.01364
	(15) = .12297
	(16) = .24616

Note: All filter weights were initially set to zero.

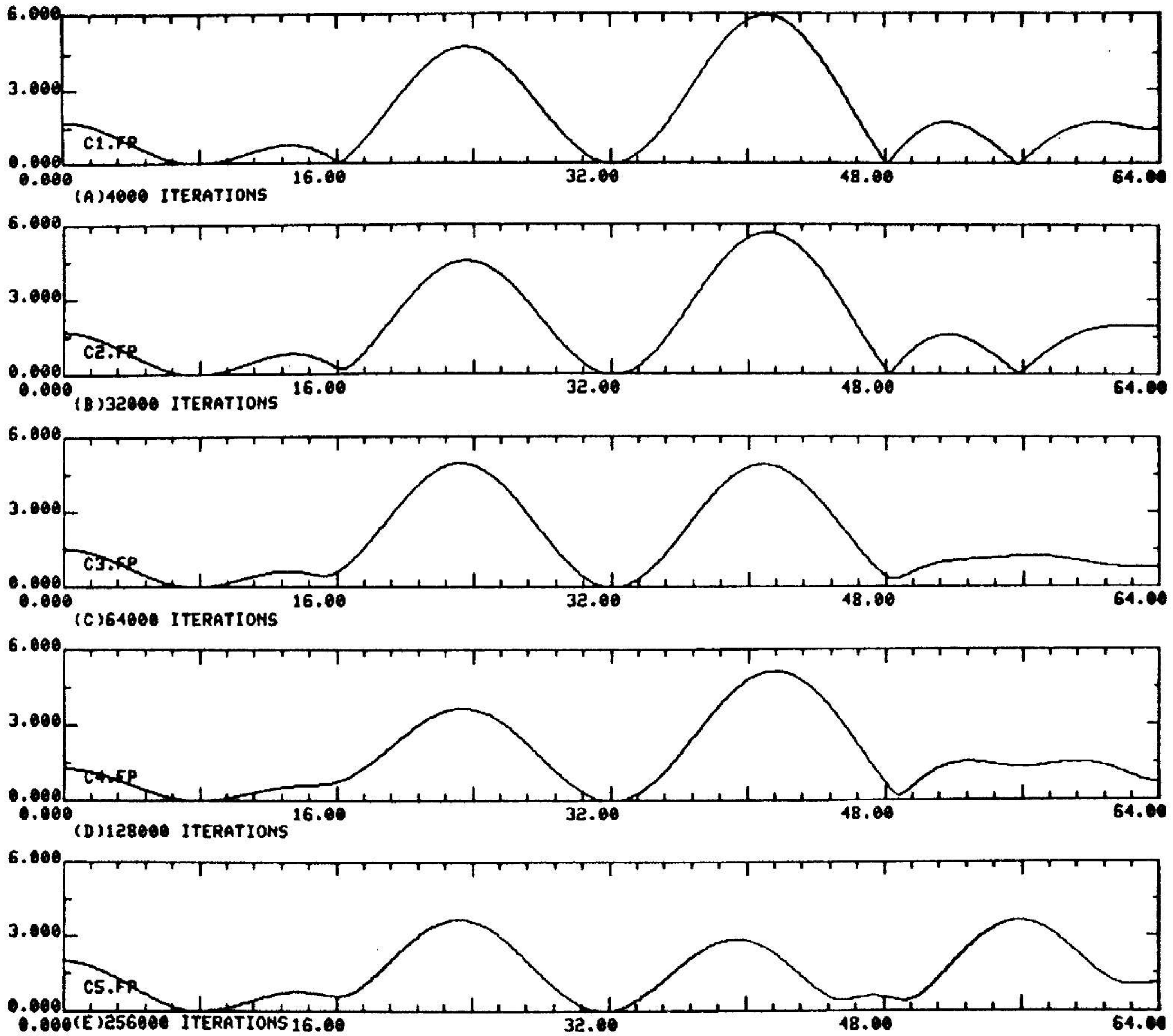
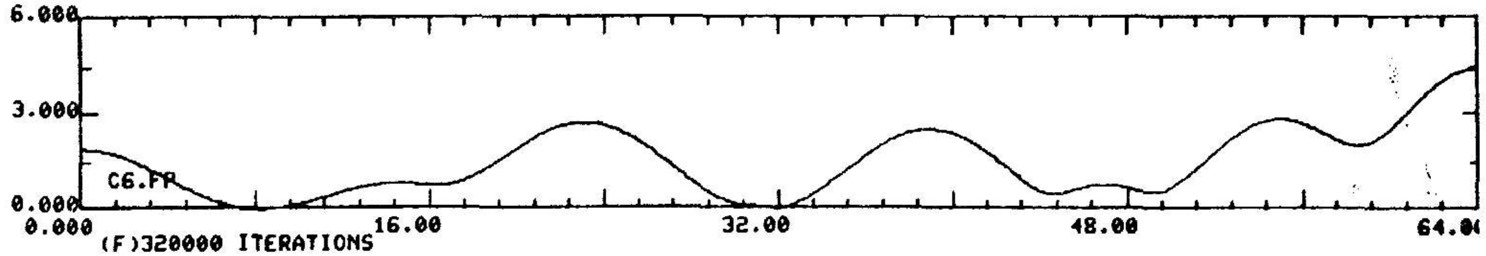
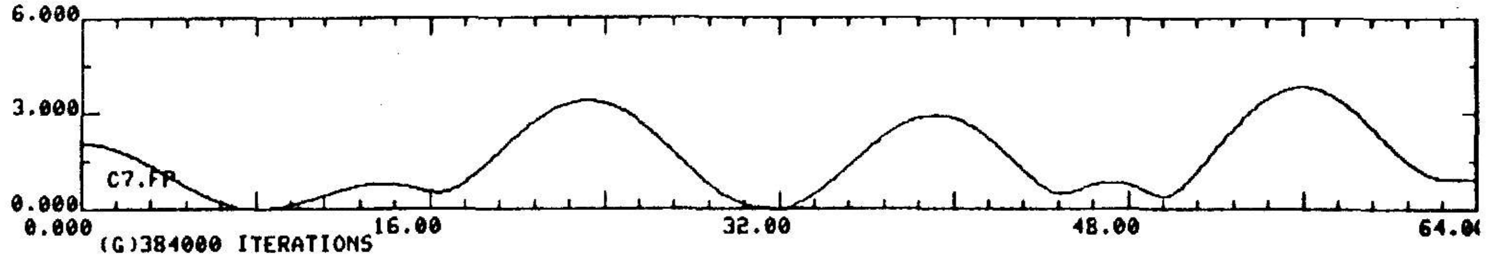


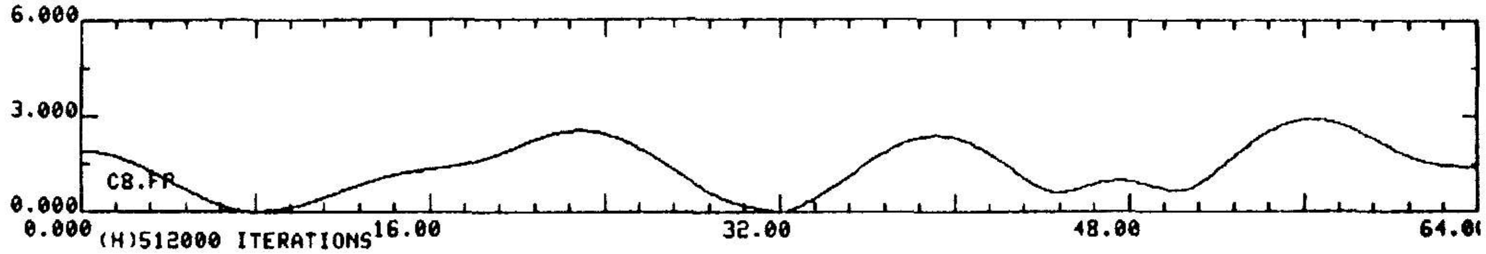
FIG.7. OVERALL GAIN FUNCTION OF LATTICE PREDICTOR OBTAINED VIA EXPERIMENT 3



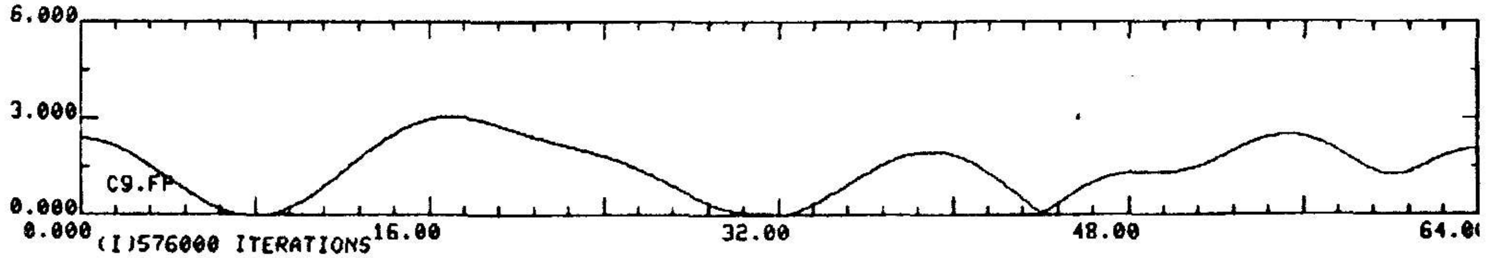
(F)320000 ITERATIONS



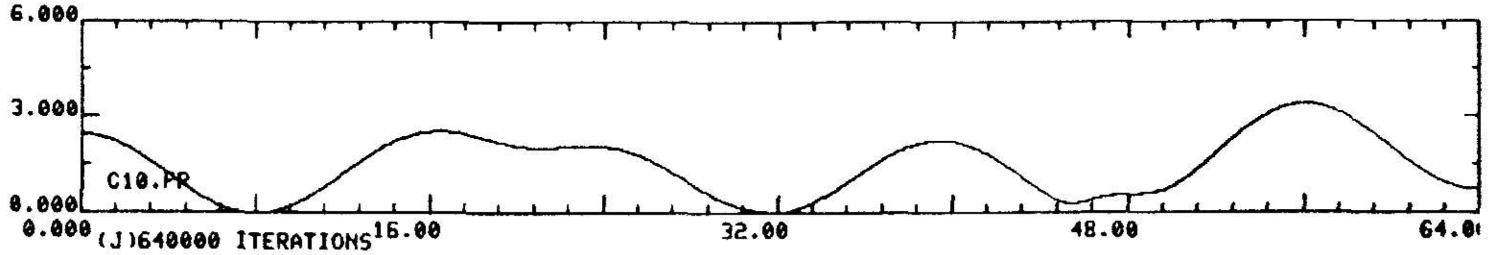
(G)384000 ITERATIONS



(H)512000 ITERATIONS



(I)576000 ITERATIONS



(J)640000 ITERATIONS

FIG.7.(CONTINUED)

TABLE 3: ADAPTIVE LATTICE FILTER; EXPERIMENT 3

Iteration	Lattice Coeff.	(Equivalent Filter) Weights	Output Error
4000	(1) = .47066	(1) = 1.0	(1) = -1.38
	(2) = -.44713	(2) = - .57751	(2) = -1.06
	(3) = .813	(3) = - .41157	(3) = - .882
	(4) = -.99764	(4) = .24111	(4) = .24
	(5) = -.68556	(5) = - .06796	(5) = .013
	(6) = -.35289	(6) = .268	(6) = - .023
	(7) = .08629	(7) = 1.21	(7) = - .018
	(8) = -.06591	(8) = - .59	(8) = - .018
	(9) = .438	(9) = .65655	(9) = - .021
	(10) = -.1	(10) = .65655	(10) = - .012
	(11) = -.553	(11) = .738	(11) = - .015
	(12) = -.857	(12) = - .21	(12) = - .005
	(13) = .593	(13) = .469	(13) = .00488
	(14) = -.51295	(14) = .26521	(14) = - .00223
	(15) = -.91	(15) = - .752	(15) = - .00404
	(16) = -.30499	(16) = .649	(16) = - .00262
		(17) = .305	(17) = - .00232
640000	(1) = .46147	(1) = 1	(1) = -1.26
	(2) = -.446	(2) = - .323	(2) = -1.21
	(3) = .812	(3) = .206	(3) = -1.32

TABLE 3: Continued

Iteration	Lattice Coeff.	(Equivalent Filter) Weights	Output Error
640000	(4) = -.99	(4) = - .46	(4) = - .087
	(5) = -.74	(5) = - .157	(5) = - .067
	(6) = -.669	(6) = .133	(6) = - .095
	(7) = .00040	(7) = .637	(7) = - .045
	(8) = .488	(8) = .825	(8) = - .045
	(9) = -.261	(9) = - .412	(9) = - .022
	(10) = .077	(10) = .449	(10) = - .026
	(11) = -.347	(11) = - .112	(11) = - .024
	(12) = -.439	(12) = .044	(12) = - .007
	(13) = -.439	(13) = - .031	(13) = - .001
	(14) = -.053	(14) = .181	(14) = .003
	(15) = -.192	(15) = .061	(15) = .003
	(16) = -.403	(16) = .03	(16) = - .003
		(17) = .403	(17) = - .003

Note: All lattice coefficients were initially set to zero.

The resulting overall gain plots obtained with $\alpha = 0.02$, $\beta = 0.98$ and $\varepsilon = 0.000061$ are shown in Fig. 8. Examination of the plots in Fig. 7 and Fig. 8 shows that there is some difference in them at $n = 4000$. But these plots are essentially identical beyond $n = 64000$.

<u>Experiment 5</u>

One approach used to improve the rate of convergence of Widrow's LMS algorithm in (40) is to make the convergence parameter ν time-varying. This is accomplished by the following version of the LMS algorithm [9]:

$$b_{m,\, n+1} = b_{m,\, n} + \nu_n e_n f_{n-m}, \quad 1 \leq m \leq 16 \tag{43}$$

where

$$\nu_n = \frac{\alpha}{\sigma^2 (n)},$$

with α being a constant convergence parameter, and $\sigma^2 (n)$ being an estimate of the input variance. The variance is estimated via the relation

$$\sigma^2 (n) = \beta \ \sigma^2 (n-1) + (1-\beta) f^2_{n-m} \tag{44}$$

The basic strategy involved in this approach is that ν_n is made inversely proportional to the input signal energy. As was the case with the LMS lattice algorithm, (see (42)), the division of α by $\sigma^2 (n)$ in (43) is not carried out if $\sigma^2 (n)$ is less than some predetermined ε.

The results obtained by means of using the algorithm in (43) for $\alpha = 0.02$ and $\beta = 0.98$ are displayed in Fig. 9. The behavior of this gain function as n increases is very similar with that in Fig. 4. We observe that when $n = 640{,}000$, the gain function in Fig. 9(j) has

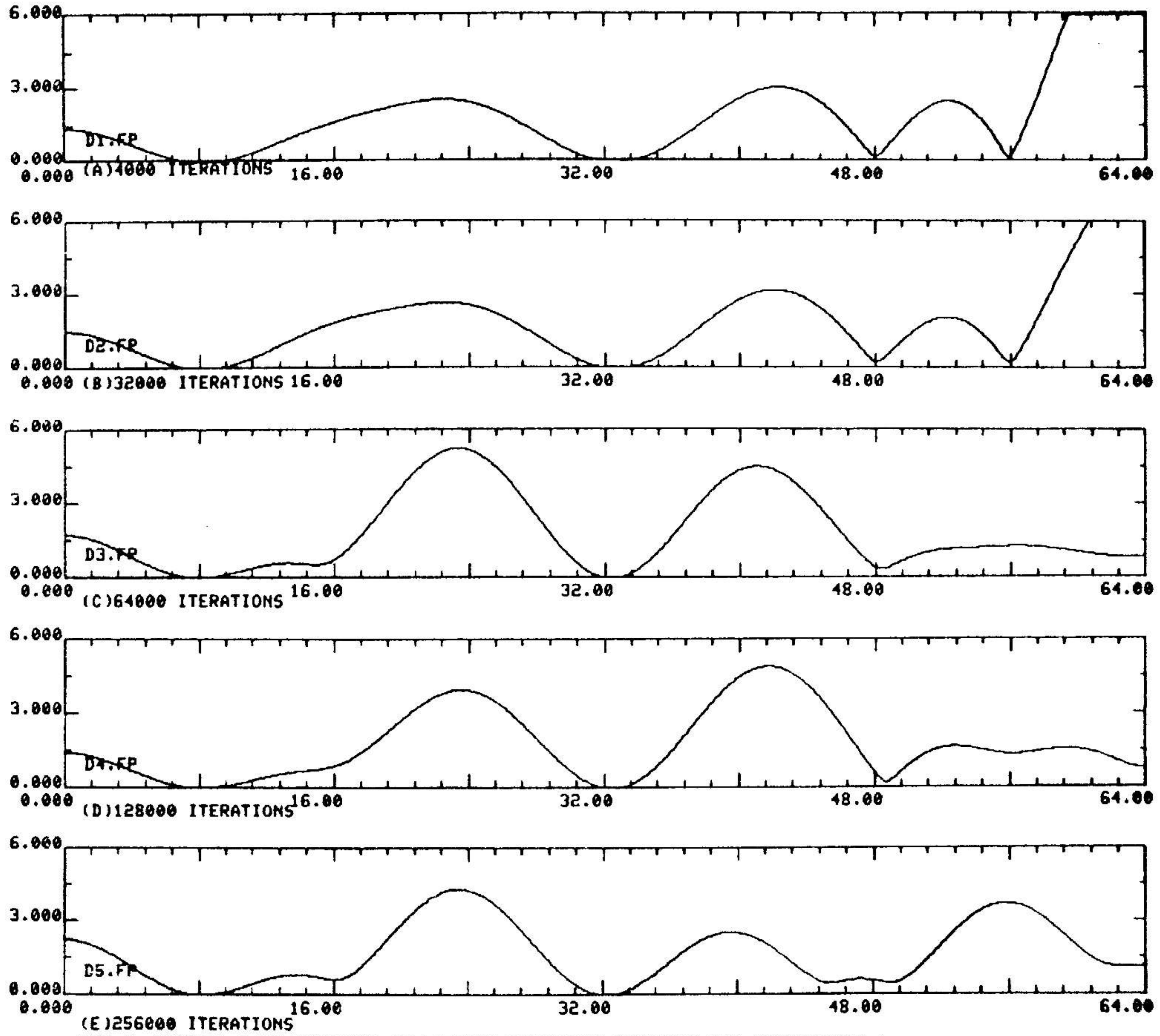


FIG.8. OVERALL GAIN FUNCTION OF LATTICE PREDICTOR OBTAINED VIA EXPERIMENT 4

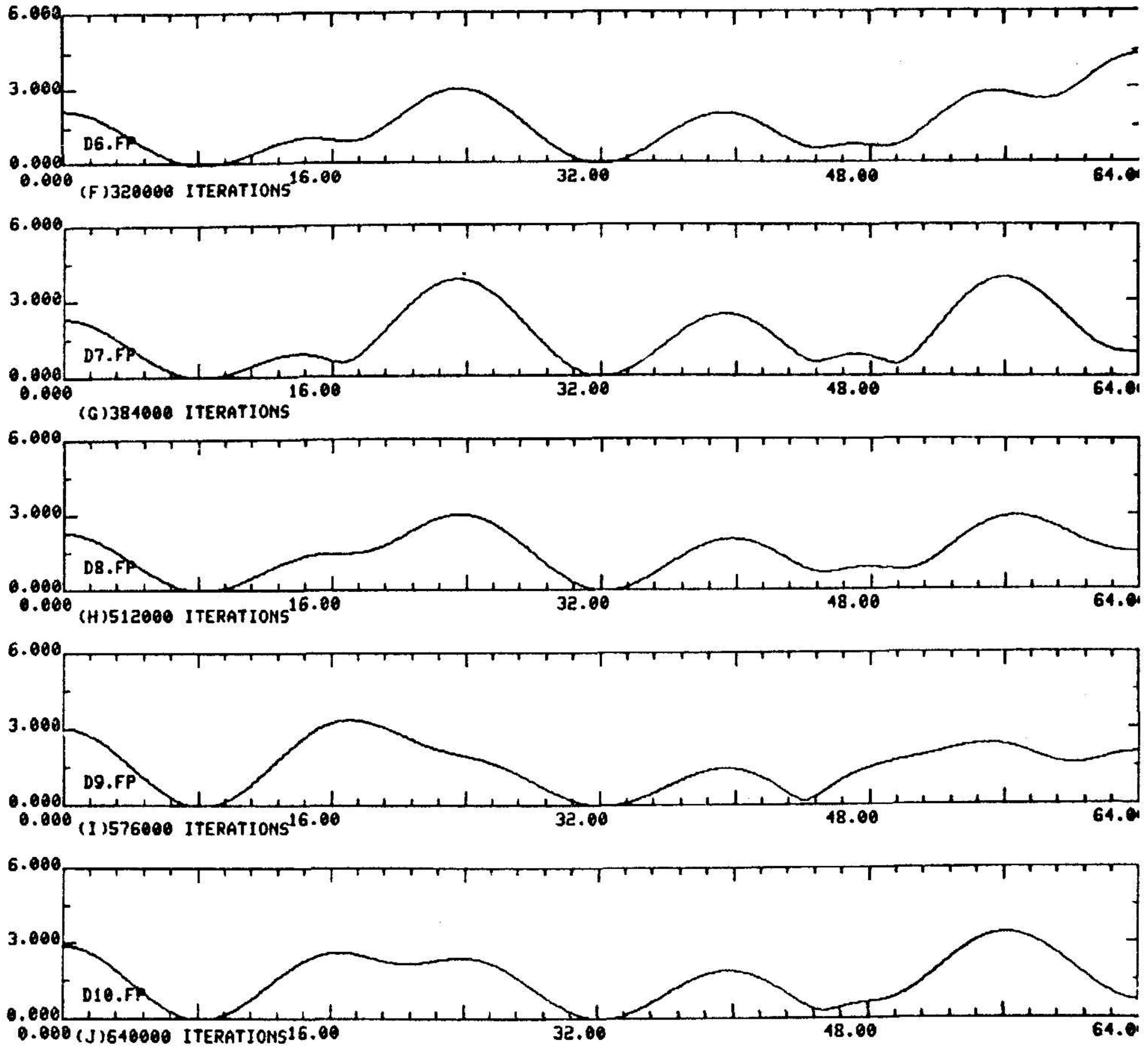


FIG.8.(CONTINUED)

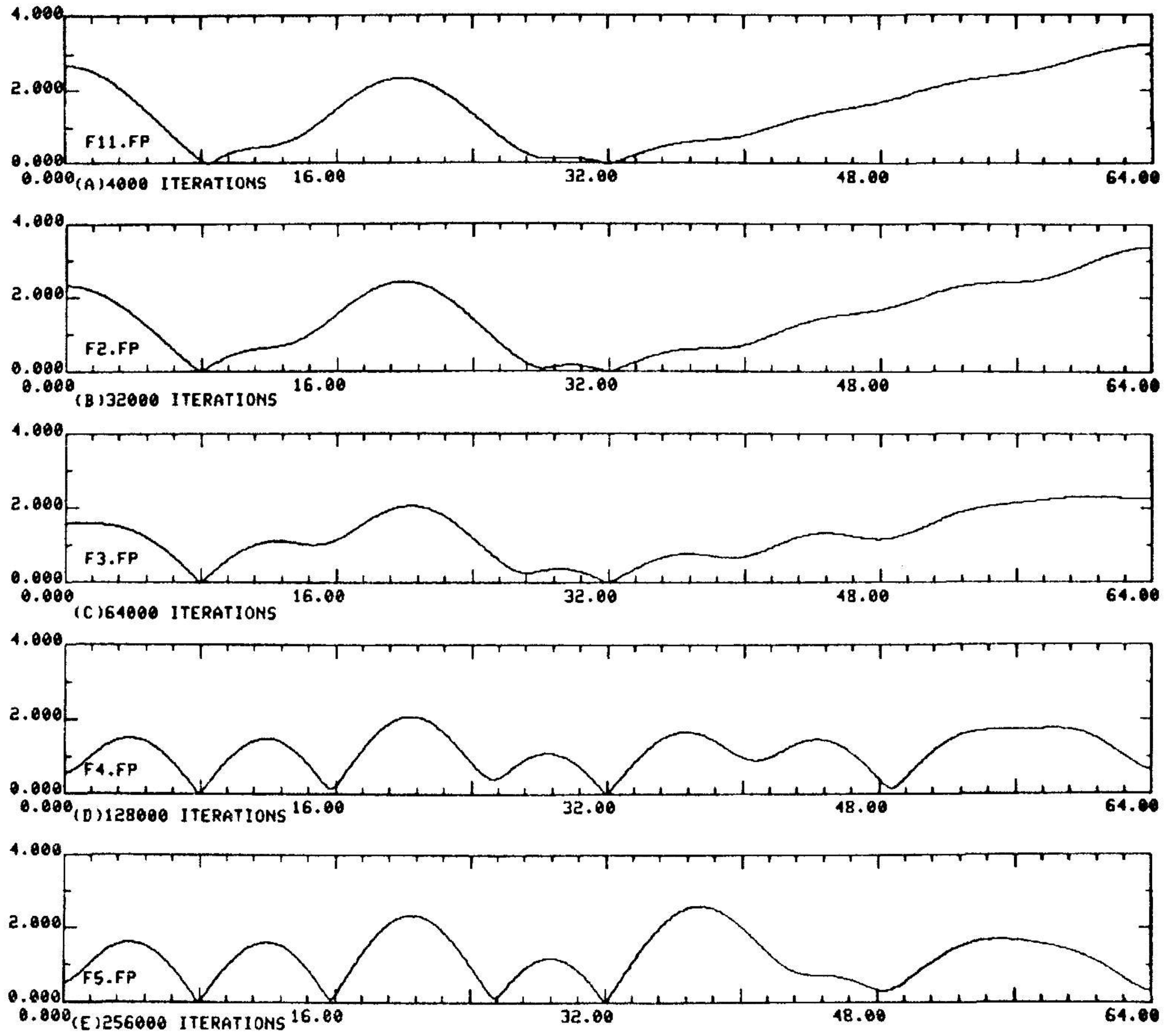


FIG.9. OVERALL GAIN FUNCTION OF PREDICTOR CONSIDERED IN EXPERIMENT 5

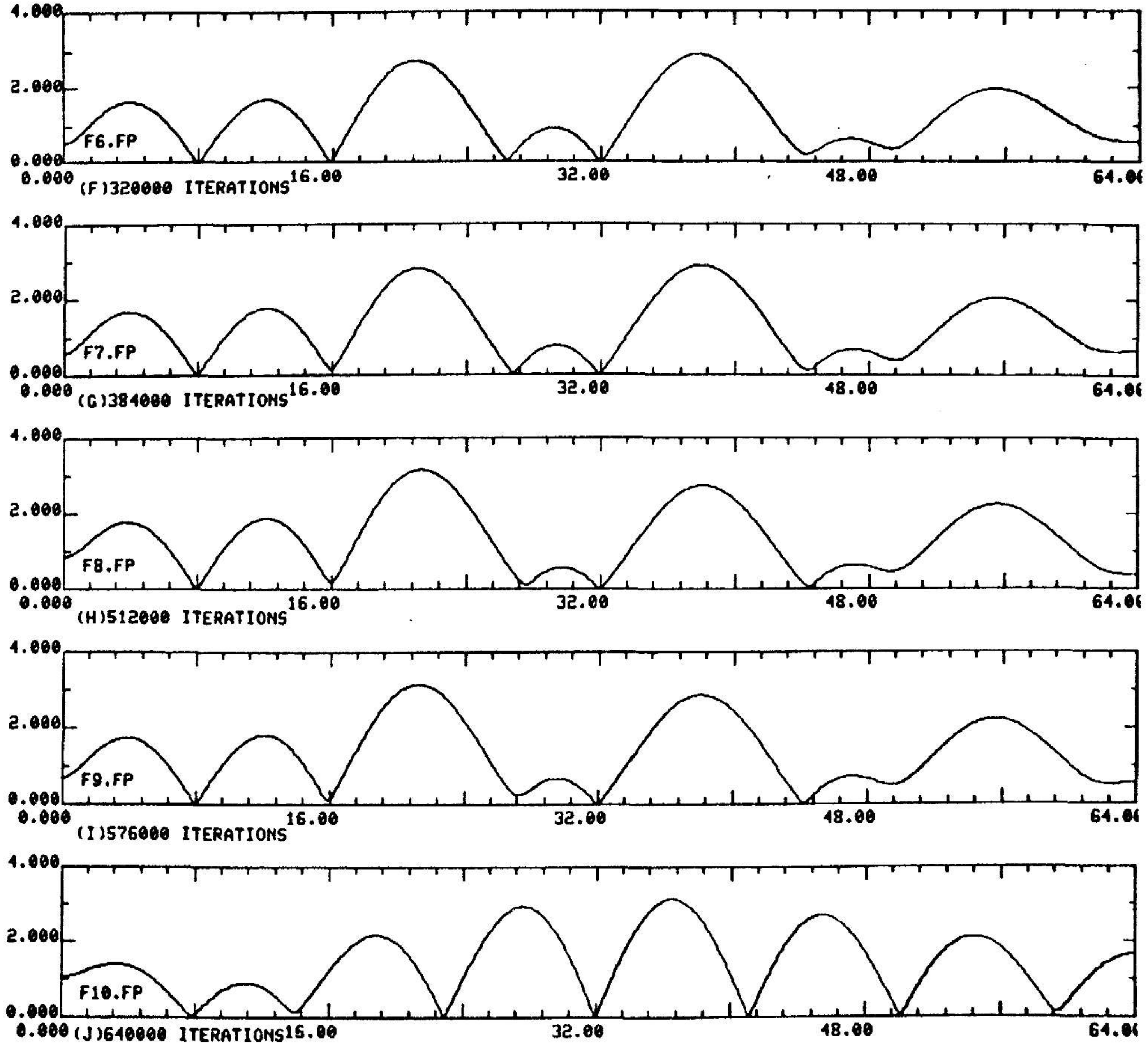


FIG.9.(CONTINUED)

several nulls in addition to those required at 8 and 32 HZ. Thus, input frequency components at which the nulls appear will be eliminated, as was the case with the conventional Widrow algorithm whose results are given in Fig. 4; in particular, compare Fig. 9(j) with Fig. 4(j).

Experiment 6

This final experiment concerns a simple modification of the algorithm in (43). This modification involves the introduction of the parameter μ, as was the case with the MLMS algorithm in (41). Thus, corresponding to (43) we have

$$b_{m,\, n+1} = \mu\, b_{m,\, n} + \nu_n\, e_n\, f_{n-m}, \quad 1 \leq m \leq 16 \tag{45}$$

where $0 < \mu < 1$ is chosen to be close to 1. Here we consider the case $\mu = 1 - 2^{-13} = 0.999878$.

The corresponding gain function for $\alpha = 0.02$ and $\beta = 0.98$ are plotted in Fig. 10, from which it is apparent that the gain function is essentially unchanged beyond $m = 32{,}000$. Further, the shape of the gain function for $m \geq 32{,}000$ is what we would desire since it has only two nulls at 8 and 32 HZ. This means that input frequency components at frequencies other than 8 and 32 HZ will not be eliminated. This result is in agreement with that obtained via the MLMS algorithm in (41). This is apparent if one compares the plots in Fig. 6 and Fig. 10 for $n \geq 32{,}000$.

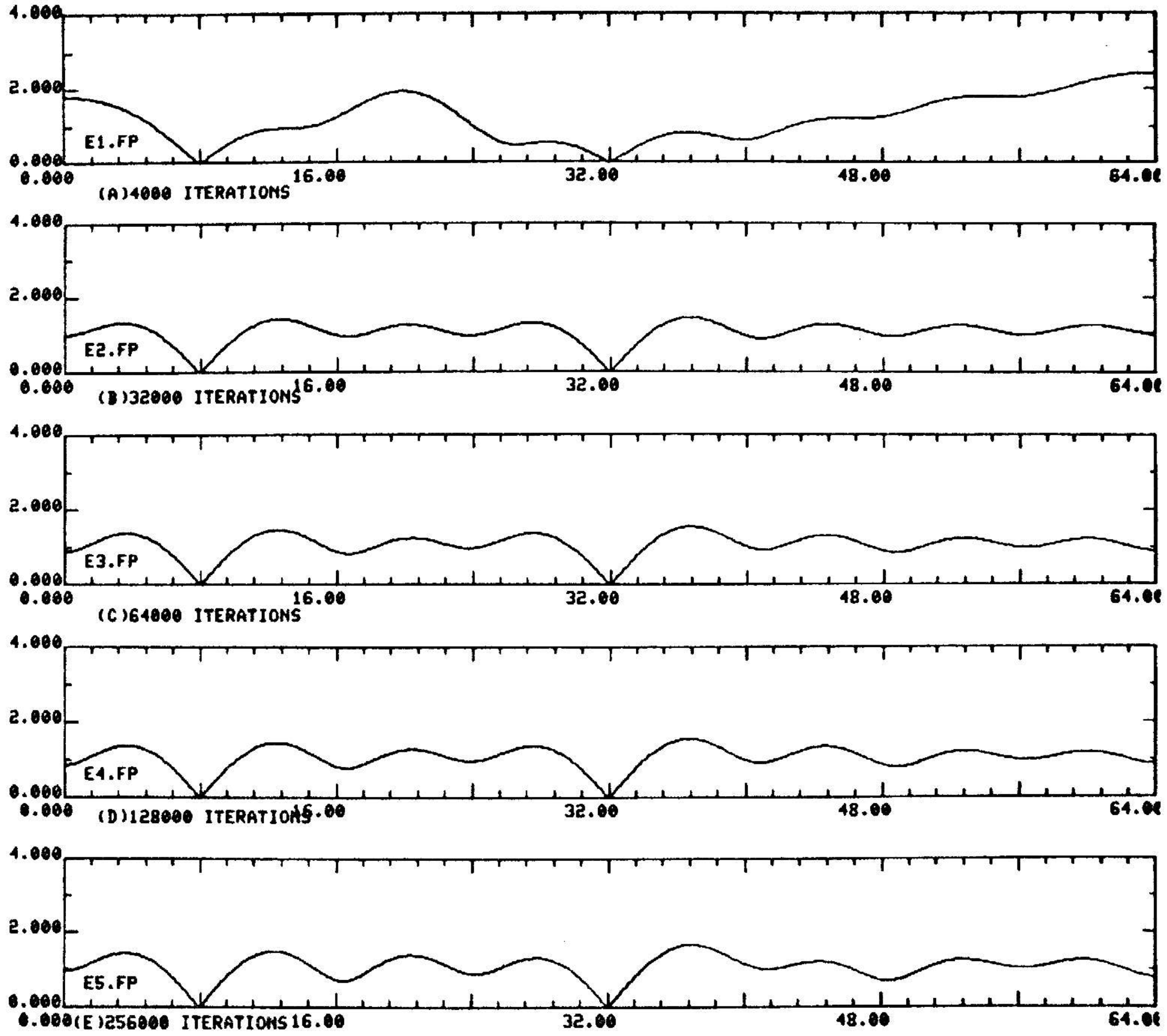


FIG.10.OVERALL GAIN FUNCTION OF PREDICTOR CONSIDERED IN EXPERIMENT 6

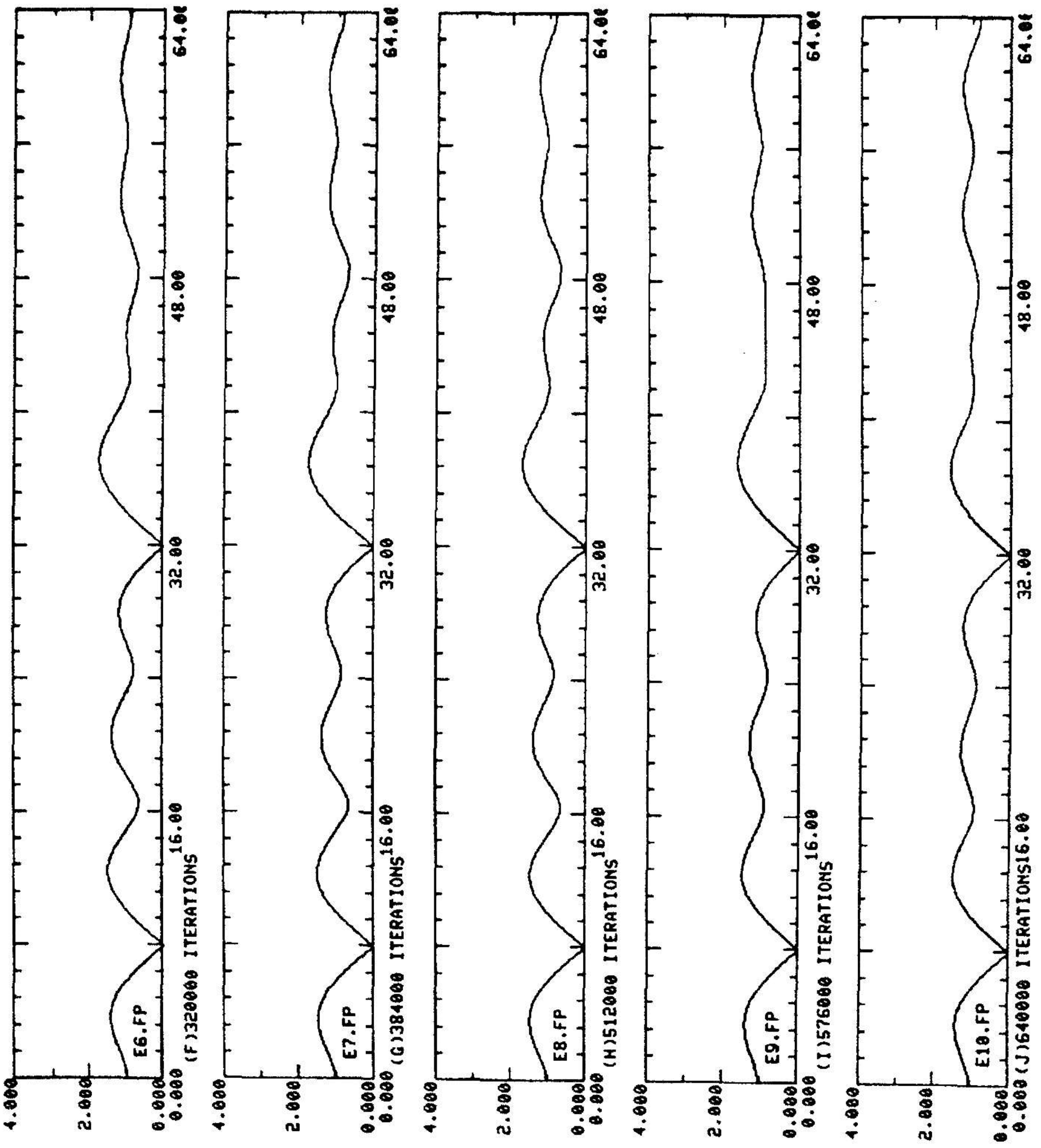


FIG.10.(CONTINUED)

6. DISCUSSION OF RESULTS AND CONCLUSIONS

The simulation results related to Experiments 3 and 4 demonstrate that the LMS lattice algorithm as defined in (28) does not have the undesirable property that its overall gain function starts to develop an increasing number of nulls as time goes on. We attribute this property of the LMS lattice algorithm to the lattice structure itself, in that essentially all the prediction (decorrelation) error is confined to the first four stages when the input is a sum of two sine waves. For example, using the output (forward prediction) errors in Table 3 at iteration 4000, it can be verified that about 99.94% of the squared error occurs in the first four of the sixteen stages of the lattice. This suggests that the corresponding prediction error surface is essentially dependent on the first four stages. Since only four stages are necessary for exact prediction of the sum of two sine waves, the error surface should be quadratic with a single minimum, and remains that way as time goes on; i.e., a distributed minimum is avoided. It is conjectured that the surplus lattice coefficients merely tend to change this quadratic error surface a little, and move it around in a local region. For example, using the output error values in Table 3, one can verify that the total squared error is about 24.7% more at iteration 640,000, compared to that at iteration 4000. However, 99.6% of the squared error still occurs in the first four stages. This suggests that the quadratic error surface has been moved upwards relative to its location at 4000 iterations, and its shape is now slightly different. Thus we conclude that a distributed minimum is avoided, and hence the "no-pass" phenomenon does not occur when the LMS lattice algorithm in (28) is used for intrusion detection purposes [3].

Again, the results obtained via Experiments 1, 2, 5 and 6 demonstrate that it is necessary to include the parameter μ in the two versions of the Widrow algorithm defined in (41) and (43), respectively. Inclusion of this parameter avoids the "no-pass" phenomenon which is undesirable in applications such as instrusion detection [3].

REFERENCES

1. B. Widrow et al., Adaptive Noise Cancelling: Principles and Applications, Proc. IEEE, vol. 63, No. 12, pp. 1692-1712, Dec. 1975.

2. N. Ahmed et al., On An Intrusion-Detection Approach via Adaptive Prediction, IEEE. Trans. Aerospace Elec. System, vol. AES-15, No. 3, May 1979.

3. N. Ahmed, G. R. Elliott, and S. D. Stearns, Long-Term Instability Problems in Adaptive Noise Cancellers, Sandia Lab., Albuquerque, N. M., Tech. Rep. SAND., 78-1032, Aug. 1978.

4. A. Papoulis, Signal Analysis, New York: McGraw-Hill, 1977.

5. N. Ahmed and R. J. Fogler, On an Adpative Lattice Predictor and a Related Applications, IEEE Circuits and Systems Magazine, Vol. 1, No. 4, 1979, pp. 17-23.

6. L. J. Griffiths, A Continuously-Adaptive Filter Implemented as a Lattice Structure, in Proc. ICASSP, Hartford, CT., May, 1977, pp. 683-686.

7. L. J. Griffiths, An Adpative Lattice Structure for Noise-Cancelling Applications, Proc. ICASSP, Tulsa, OK., AP. 1978, pp. 87-90.

8. V. N. Faddeva, Computational Methods of Linear Algebra, Dover Publi-

9. N. Ahmed, J. P. Claassen, and R. C. Beckmann, Intrusion Detection Using Adaptive Digital Signal Processing, SANDIA Lab., Albuquerque, NM, Tech. Rep. SAND. 79-1585 C, Oct. 1979.

```
C                 APPENDIX A
C       ********************************************
C       GENERAL ADAPTIVE WIDROW PREDICTOR TO
C       TEST SYSTEM USING SINE INPUTS
C
C       DEPARTMENT OF ELECTRICAL ENG.           KSU
C
C       DATE                 PROGRAMMER
C       ----                ------------
C       FEB.7,1981          WANG YUNG-NING
C
C
C       ********************************************
C
C
        REAL ITER
        DIMENSION REF(129) ,DES(129),W(129)
C
90      CONTINUE
C
C       OPEN I/O FILE
C
        ACCEPT'# OF ITERATION          :       ',ITER
        TYPE
        ACCEPT'# OF TEST NUMBER         :       ',NB
        TYPE
        ACCEPT'# OF WEIGHTS             :       ',NW
        TYPE
        ACCEPT'TEST SINEWAVE FREQ1      :       ',FREQ1
        ACCEPT'TEST SINEWAVE FREQ2      :       ',FREQ2
        ACCEPT'SAMPLING FRRQ            :       ',SAM
        TYPE
        ACCEPT'VALUE OF V               :       ',V
        TYPE
        ACCEPT'VALUE OF U               :       ',U
C
        FREQ1=2.0*3.141592*FREQ1/SAM
        FREQ2=2.0*3.141592*FREQ2/SAM
C
C
C       INITIALIZATION
C
        NW1=NW-1
        DO 11 K=1,129
        REF(K)=0.0
        DES(K)=0.0
        W(K)=0.0
11      CONTINUE
        A=1.0
        J=2
        LA=0.0
C
C
C       MAIN LOOP
C
C
        DO 100 I=1,ITER
C
C       READ DESIRED OUTPUT
C
        DES(1)=SIN(FREQ1*A)+SIN(FREQ2*A)
        NWK=NW
        DO 30 K=1,NW1
        NWK=NWK-1
        REF(NWK+1)=REF(NWK)
```

```
30      CONTINUE
C
C       READ REFERENCE INPUT
C
        REF(1)=SIN(FREQ1*(A-1))+SIN(FREQ2*(A-1))
        A=A+1
C
C       COMPUTE ESTIMATED VALUE
C
        GM=0.0
        DO 40 K=1,NW
        GM=GM+(W(K)*REF(K))
40      CONTINUE
        EM=DES(1)-GM
        DO 51 K=1,NW
        W(K)= W(K)+(V*EM*REF(K))
51      CONTINUE
        J=J+1
        IF(J.NE.NB) GO TO 100
        LA=LA+1
        TYPE LA
        WRITE(10,50)LA
50      FORMAT( '     ',I3)
C
C       DUMP VALUE OF FIR WEIGHTS W(M)
C
        DO 102 M=1,NW
        WRITE(10,400)M,W(M)
400     FORMAT( 'K( ',I2,')= ',F15.5)
102     CONTINUE
        J=0
100     CONTINUE
C
C       CLOSE FILE
C
        CALL QUERY(' RE-EXECUTE ?(YES/NO) ',LL)
        IF (LL.EQ.1) GO TO 90
        STOP
        END
```

```
C               APPENDIX B
C       *********************************************
C
C
C       LEAST MEAN SQUARE ADAPTIVE LATTICE
C       PREDICTOR TO TEST SYSTEM USING SINE INPUTS.
C
C
C       DEPARTMENT OF ELECTRICAL ENG.             KSU
C
C       DATE                                PROGRAMMER
C       -----                               -----------
C       FEB.7,1981                          WANG YUNG-NING
C
C
C       *********************************************
C
C
        REAL W(51),E(51),W1(51)
        REAL  V(50),O(20),DP(20),K(50),ITER
C
38      CONTINUE
C
                TYPE
C
        ACCEPT '# OF ITERATION       : ',ITER
        ACCEPT '# OF TEST NUMBER      :',NB
        ACCEPT '# OF STAGE          : ',N
        ACCEPT 'SAMPLING FREQ       : ',SAM
        ACCEPT 'TEST SINEWAVE FREQ-1:',FREQ1
        ACCEPT 'TEST SINEWAVE FREQ-2:',FREQ2
        TYPE
        ACCEPT 'CONV. PARAMETER      : ',ALPHA
        ACCEPT 'SMOO. PARAMETER      : ',BETA
        ACCEPT 'EPSILON        : ',EPSP
        ACCEPT 'U              : ',U
        TYPE
C
C
        FREQ1=2.0*3.141592*FREQ1/SAM
        FREQ2=2.0*3.141592*FREQ2/SAM
        K(1)=(COS(FREQ1)+COS(FREQ2))/2.0
        SQ=K(1)*K(1)
        K(2)=(COS(2.0*FREQ1)+COS(2.0*FREQ2)-SQ*2.0)/(2.0-K(1))
        TYPE '*** OPTIMUM K(1) AND K(2) ***'
        TYPE K(1),K(2)
C
C
        BETA1=1.0-BETA
        N1=N+1
C
C       INITIALIZE THE ARRAYS.
C
        DO 5 L=1,N
        E(L)=0.0
        W(L)=0.0
        W1(L)=0.0
        K(L)=0.0
        V(L)=0.0
5       CONTINUE
C
        E(N1)=0.0
        W(N1)=0.0
        W1(N1)=0.0
        A=0.0
```

```
C
C        MAIN LOOP.
C
         J1=0
         LA=0.0
         DO 100 I=1,ITER
C
C
C
C                 READ REF INPUT.
C
           E(1)=SIN(FREQ1*A)+SIN(FREQ2*A)
         A=A+1
         W(1)=E(1)
         L1=2
C
C        STAGE LOOP.
C
         DO 200 J=1,N
C
         E(L1)=E(J)-K(J)*W1(J)
         W(L1)=W1(J)-K(J)*E(J)
         V(J)=BETA*V(J)+BETA1*(E(J)*E(J)+W1(J)*W1(J))
         TV=1.0
         IF (V(J).LT.EPSP) TV=0.0
         K(J)=U*K(J)+TV*ALPHA*(E(L1)*W1(J)+E(J)*W(L1))/V(J)
         W1(J)=W(J)
         L1=L1+1
C
200      CONTINUE
C
C        COMPUTE D(I)(COEFF.OF FIR)
C        FROM K(I)(LATTICE COEFF.)
C
         NKM1=N-1
         D(1)= -K(1)
         DP(1)=D(1)
         IF(NKM1.EQ.0)GO TO 150
         DO 15 L=1,NKM1
         DO 17 M=1,L
         D(M)=DP(M)-K(L+1)*DP(L+1-M)
17       CONTINUE
         D(L+1)=-K(L+1)
         IP1=L+1
         DO 19 JJ=1,IP1
         DP(JJ)=D(JJ)
19       CONTINUE
15       CONTINUE
150      CONTINUE
         J1=J1+1
         IF(J1.NE.NB)GO TO 100
         LA=LA+1
         TYPE LA
         WRITE(12,50)LA
50       FORMAT('     ',I3)
C
C        DUMP VALUE OF LATTICE WEIGHT K(I),
C        FIR WEIGHT D(I) AND
C        PREDICTOR OUTPUT
C
         DO 102 I1=1,N
         WRITE(12,400)I1,K(I1)
400      FORMAT(' K(',I2,')=',F15.5)
102      CONTINUE
         DO 103 I2=1,N1
         WRITE(12,401)I2,E(I2)
[illegible]
```

```
401     FORMAT(' E(',I2,')=',F15.5)
103     CONTINUE
        KK=0.0
        T=1.0
        WRITE(12,21)KK,T
        DO 23 II=1,N
21      FORMAT('  D(',I2,')=',F15.5)
        WRITE (12,21)II,D(II)
23      CONTINUE
        J1=0
100     CONTINUE
C
C
        CALL QUERY ('<12><7>RE-EXECUTE (Y/N) ? ',NEX)
        IF (NEX.EQ.1) GO TO 88
C
        STOP ** NORMALLY TERMINATED **
        END
```

TRANSFER FUNCTION CONSIDERATIONS OF AN ADAPTIVE LATTICE PREDICTOR

by

YUNG-NING WANG

B. S., National Taiwan Institute of Technology, 1977

AN ABSTRACT OF A MASTER'S REPORT

submitted in partial fulfillment of the

requirement for the degree

MASTER OF SCIENCE

Department of Electrical Engineering

KANSAS STATE UNIVERSITY

Manhattan, Kansas

1981

ABSTRACT

In this report, we study the long-term behavior of the transfer function of an adaptive lattice predictor, and compare it with that Widrow's LMS (least mean squares) adaptive predictor. This study and related comparison is carried out via a computer simulation.