THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 3, March 1983
Printed in U.S.A.

Performance of a Fast Algorithm for FIR System Identification Using Least-Squares Analysis

By S. L. MARPLE, JR.,* and L. R. RABINER

(Manuscript received August 31, 1982)

A wide variety of procedures have been proposed for identifying a finite impulse response (FIR) linear system from the input and output of the system. Most recently, a fast, efficient, least-squares method was proposed by Marple, and was shown to require less computation and storage than any other known procedure for identifying moderate to large FIR systems. In this paper we measure the actual performance of the newly proposed fast system identification algorithm by using it to estimate a variety of FIR systems excited by either white noise or a speech signal. It is shown that essentially theoretically ideal performance is achieved for white noise inputs; however, for speech signals poor performance was obtained because of the lack of certain frequency bands in the excitation. A simple modification to the estimation procedure is proposed and is shown to provide substantial performance improvements. Using the spectrally modified speech signal, the performance of the fast system identification algorithm was found to be acceptable for a wide variety of applications.

I. INTRODUCTION

In previous papers,[1-4] two system identification methods, the classical least-squares analysis (LSA) and a short-time spectral analysis (SSA) procedure, had their performance compared and contrasted in the presence of high noise levels and in situations where the input signal was band-limited (nonwhite noise and speech). This earlier work found that, while the LSA method produced better performance than the SSA procedure, the computational burden of the Cholesky solution of the equations of the LSA method became prohibitive when com-

* Schlumberger Well Services, Houston, Texas.

pared to the SSA method as the system order became large (as it can be in speech processing, where the filter order can be on the order of 1000). This factor weighed in favor of the SSA method. However, the development of fast algorithms for the solution of the LSA normal equations[5-8] has greatly reduced the computational complexity. This paper evaluates the performance of the LSA procedure in the context of these fast algorithms.

The fast algorithm that we will consider for solving normal equations for the least-squares FIR system identification has computational complexity proportional to M^2, where M is the filter order, and storage that grows linearly (rather than quadratically) with M. A byproduct of the computation is an estimate of the linear prediction coefficients of the input process. The fast algorithm also has simple, built-in, numerical ill-conditioning checks. If the model order is uncertain, the fast algorithm recursively provides all optimum least-squares solutions from filter order 1 up to some user-selected maximum order, M, thereby providing a built-in search capability for finding the appropriate system order without having to start over.

The outline of this paper is as follows. In Section II we review the normal equations of the least-squares system identification and show how a fast algorithm can be derived to solve these equations. In Section III we present an evaluation of the performance of the fast algorithm for several different FIR systems with both broadband noise and speech inputs. In Section IV we review the results and compare them to those obtained previously using short-time spectral analysis methods.

II. REVIEW OF THE LEAST-SQUARES NORMAL EQUATIONS

Figure 1 shows a block diagram of the finite-impulse-response system identification model. The discrete input signal, $x(n)$, drives the unknown system to produce the discrete sequence, $y(n)$, where n is an integer index. The unknown system was modeled as an FIR filter, assumed to have an impulse response duration of $M + 1$ samples, so that the estimated impulse response, $\hat{h}(n)$, is zero for $n < 0$ and $n > M$. The order M of the FIR system is defined here as the highest index of the impulse response, $\hat{h}(M)$.

The approach used is to determine the impulse-response coefficients, $\hat{h}(n)$, and the system order M that minimize the squared error based on a finite number of measurements of the input process, $x(n)$, and the output process, $y(n)$. Denoting the linear estimate of $y(n)$ by $\hat{y}(n)$, then

$$\hat{y}(n) = \sum_{m=0}^{M} \hat{h}(m)x(n-m), \tag{1}$$

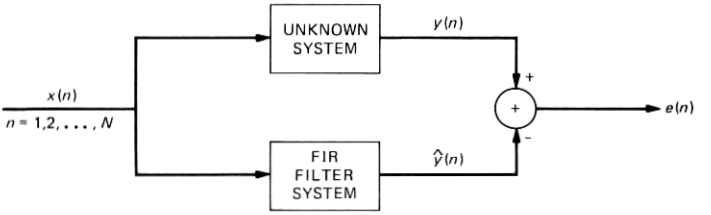

Fig. 1—Block diagram of the FIR system identification.

which is the familiar convolution expression of a finite-impulse-response filter. The estimation error, $e(n)$, is then

$$e(n) = y(n) - \hat{y}(n). \tag{2}$$

We wish to minimize the total squared error, P_M, of an Mth-order model

$$P_M = \sum_n e^2(n) \tag{3}$$

with respect to $\hat{h}(0), \cdots, \hat{h}(M)$ and based on the block of finite data sets $x(1), \cdots, x(N)$ and $y(1), \cdots, y(N)$. Note that the index range for n in eq. (3) has purposely been left unspecified. To operate only on the available data samples, the range must be selected to be $n = M + 1$ to N, which is the index range selected for this paper because it provides the best performance for relatively short data segments. However, by defining the unobserved data to be zero for $n \leq 0$, then a so-called "prewindowed" index range of $n = 1$ to N can be used [the data $x(n)$ and $y(n)$ for $n \leq 0$ is "windowed" to zero]. These two cases are illustrated in eq. (4) using a matrix structure to describe the error terms:

$$\begin{Bmatrix} 2 \\ 1 \end{Bmatrix} \left\{ \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_{M+1} \\ \vdots \\ e_N \end{bmatrix} \right\} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{M+1} \\ \vdots \\ y_N \end{bmatrix} - \begin{Bmatrix} 2 \\ 1 \end{Bmatrix} \left\{ \begin{bmatrix} x_1 & & 0 & \\ x_2 & x_1 & & \\ \vdots & \vdots & & \\ x_{M+1} & x_M & \cdots & x_1 \\ \vdots & \vdots & & \vdots \\ x_N & x_{N-1} & \cdots & x_{N-M} \end{bmatrix} \begin{bmatrix} \hat{h}_0 \\ \hat{h}_1 \\ \vdots \\ \hat{h}_M \end{bmatrix} \right\} \tag{4a}$$

or

$$E_M = Y_M - X_M \hat{H}_M \tag{4b}$$

and

$$P_M = E_M^T E_M, \tag{5}$$

where E_M, Y_M, $\hat{H}_M$ are column vectors and X_M is a rectangular Toeplitz data matrix. The T denotes matrix transpose. If X_M includes only that portion within the 1 brace, this corresponds to the index range $n = M + 1$ to N. If X_M includes that portion indicated within the 2 brace, this corresponds to the prewindowed index range $n = 1$ to N.

If P_M is now minimized by setting the derivatives with respect to $\hat{h}(0), \cdots, \hat{h}(M)$ to zero, then the resulting least-squares solution can be expressed in matrix form as

$$\Phi_M^{xx} \hat{H}_M = \Phi_M^{yx}, \tag{6}$$

where

$$\Phi_M^{xx} = X_M^T X_M = \text{an } (M+1) \times (M+1) \text{ matrix}$$

$$\Phi_M^{yx} = X_M^T Y_M = \text{an } (M+1) \text{ column vector.}$$

This is the discrete Wiener-Hopf equation. The minimum squared error is

$$\min P_M = Y_M^T Y_M - (\Phi_M^{yx})^T \hat{H}_M. \tag{7}$$

Note that this solution is applicable no matter what index range is selected. Also note that while the matrix Φ_M^{xx} is not, in general, Toeplitz, it is the product of two rectangular Toeplitz data matrices. This will prove to be a key factor in developing a fast algorithm solution for eq. (6). For the index range of interest here, individual elements of Φ_M^{xx} are

$$\phi_M^{xx}(i, j) = \sum_{n=M+1}^{N} x(n-j)x(n-i) \qquad \text{for } 0 \le i, j \le M. \tag{8}$$

The elements of Φ_M^{yx} for the unwindowed index range are

$$\phi_M^{yx}(i) = \sum_{n=M+1}^{N} y(n)x(n-i) \qquad \text{for } 0 \le i \le M. \tag{9}$$

Also, we have

$$Y_M^T Y_M = \sum_{n=M+1}^{N} y^2(n). \tag{10}$$

Note that the number of data samples must be at least twice the system order plus one, $N \ge 2M + 1$, in order for Φ_M^{xx} to be nonsingular.

It is also possible to perform a linear-phase FIR system identification by forcing symmetry in the filter. The linear-phase estimate is then

$$\hat{y}(n) = \hat{h}(0)x(n) + \sum_{m=1}^{M} \hat{h}(m)[x(n+m) + x(n-m)] \tag{11}$$

with error

$$\epsilon(n) = y(n) - \hat{y}(n). \tag{12}$$

Note that the total impulse-response duration of this filter is $2M + 1$. Forming the sum of squared errors over all valid error terms

$$Q_M = \sum_{n=M+1}^{N-M} \epsilon^2(n) \tag{13}$$

and minimizing leads to the normal equations

$$\Psi_M^{xx} \hat{\hat{H}}_M = \Psi_M^{yx},$$

where

$$\hat{\hat{H}}_M = \begin{bmatrix} \hat{h}(M) \\ \vdots \\ \hat{h}(1) \\ \hat{h}(0) \\ \hat{h}(1) \\ \vdots \\ \hat{h}(M) \end{bmatrix}, \qquad \Psi_M^{yx} = \begin{bmatrix} r^{yx}(M) \\ \vdots \\ r^{yx}(1) \\ r^{yx}(0) \\ r^{yx}(1) \\ \vdots \\ r^{yx}(M) \end{bmatrix},$$

$$\Psi_M^{xx} = \begin{bmatrix} r_M^{xx}(0,0) & \cdots & r_M^{xx}(0,2M) \\ \vdots & & \vdots \\ r_M^{xx}(2M,0) & \cdots & r_M^{xx}(2M,2M) \end{bmatrix}$$

$$r_M^{xx}(j,k) = \sum_{n=1}^{N-2M} [x(n+j)x(n+k) + x(n+2M-j)x(n+2M-k)]$$

$$r_M^{yx}(k) = \sum_{n=M+1}^{N-M} y(n)[x(n+k) + x(n-k)] \qquad 0 \le k \le M$$

with minimum squared error

$$Q_M = \sum_{n=M+1}^{N-M} y^2(n) - \frac{1}{2}\hat{h}(0)r_M^{yx}(0) - \sum_{m=1}^{M} \hat{h}(m)r_M^{yx}(m).$$

As before, in order for the normal equations to be nonsingular, the number of data samples must be at least twice the system order ($2M$ in this case) plus one, or $N \ge 4M + 1$.

2.1 Basis for a fast algorithm solution

In this section, a brief outline of the basis for a fast algorithm that solves eq. (6) with a number of operations proportional to M^2 is presented. For details of the full algorithm, consult Ref. 5. A fast algorithm also exists for the linear-phase FIR system identification,[8]

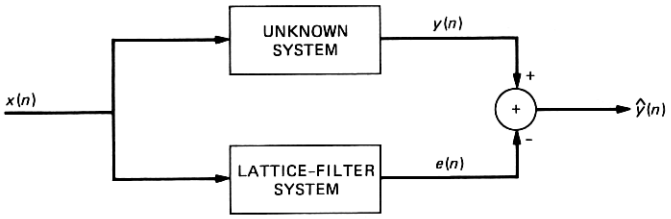

Fig. 2—Block diagram of the lattice system identification.

but will not be discussed here since the focus is on the general FIR system identification algorithm.

The fast algorithm presented here processes the available data as a block. The algorithm recursively generates all solutions from order $m = 1$ to M, where M is the maximum order. This algorithm is just one of a wider class of fast algorithms for solutions of least-squares prediction problems. If processing data on a sequential sample-by-sample basis is preferred to block processing of the data (such as for adaptive equalizer applications), then an alternative, but numerically equivalent, solution to eq. (6) is to use a lattice filter in lieu of the FIR filter. The lattice-filter output is $e(n)$ rather than $\hat{y}(n)$ (the FIR output), as shown in Fig. 2. An example of such a sequential fast algorithm for the "prewindowed" data range is presented in Ref. 6.

The key to a fast algorithm for solution of eq. (6) is the recognition that the matrix Φ_M^{xx} appears in the context of the linear prediction problem. The linear prediction filter tries to make the best estimate of the current value of $x(n)$ based on past samples of $x(n)$,

$$\tilde{x}(n) = -\sum_{m=1}^{M} a(m)x(n-m). \tag{14}$$

The prediction error $e^f(n)$ is then

$$e^f(n) = x(n) - \tilde{x}(n) = \sum_{m=0}^{M} a(m)x(n-m), \tag{15}$$

where $a(0) = 1$ and the superscript f denotes this as the "forward" linear prediction filter error. A "backward" estimate $\tilde{\tilde{x}}(n)$ can also be defined as

$$\tilde{\tilde{x}}(n-M) = -\sum_{m=1}^{M} b(m)x(n+m-M) \tag{16}$$

with "backward" prediction error

$$e^b(n) = x(n-M) - \tilde{\tilde{x}}(n-M) = \sum_{m=0}^{M} b(m)x(n+m-M),$$

where $b(0) = 1$. If we now minimize the forward squared prediction error

$$P_M^f = \sum_{n=M+1}^{N} [e^f(n)]^2$$

and the backward squared prediction error

$$P_M^b = \sum_{n=M+1}^{N} [e^b(n)]^2;$$

then we obtain the usual least-squares linear-prediction normal equations

$$\Phi_M^{xx} A_M = \begin{pmatrix} P_M^f \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \qquad \Phi_M^{xx} B_M = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ P_M^b \end{pmatrix}, \tag{17}$$

where vectors A_M and B_M are defined as

$$A_M = \begin{bmatrix} 1 \\ a(1) \\ \vdots \\ a(M) \end{bmatrix}, \qquad B_M = \begin{bmatrix} b(M) \\ \vdots \\ b(1) \\ 1 \end{bmatrix}.$$

Equation (17) is the so-called "covariance" equation of linear-prediction speech analysis. An efficient, recursive algorithm for their solution has been previously presented[7] and is incorporated as part of the FIR fast algorithm without further discussion. Equation (17) can be rewritten as

$$\Phi_M^{xx} \mathscr{A}_M = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \qquad \Phi_M^{XX} \mathscr{B}_M = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}, \tag{18}$$

where

$$\mathscr{A}_M = \begin{bmatrix} 1/P_M^f \\ a(1)/P_M^f \\ \vdots \\ a(M)/P_M^f \end{bmatrix}, \qquad \mathscr{B}_M = \begin{bmatrix} b(M)/P_M^b \\ \vdots \\ b(1)/P_M^b \\ 1/P_M^b \end{bmatrix}.$$

Thus, $\mathscr{A}_M$ and $\mathscr{B}_M$ form the first and last columns of the inverse of the Φ_M^{xx} matrix. Since the solution to eq. (12) involves the inverse $[\Phi_M^{xx}]^{-1}$,

$$\hat{H}_M = (\Phi_M^{xx})^{-1} \Phi_M^{xy}, \tag{19}$$

then we would suspect that the linear-prediction solution is an integral part of the system identification solution [eq. (19)]. This is indeed the case. In fact, only the first and last columns of the inverse (or alternatively, the vectors A_M and B_M) of Φ_M^{xx} are required to obtain a recursive solution for $\hat{H}_M$.

Three fundamental relationships govern the ability to obtain a recursive algorithm. To illustrate these relationships, it is first necessary to combine eqs. (5) and (6) into a single augmented expression as follows:

$$\bar{\Phi}_M \bar{H}_M = \bar{P}_M, \tag{20}$$

where

$$\bar{\Phi}_M = \underbrace{\left[\begin{array}{c|c} \phi_M^{yy}(0) & (\Phi_M^{yx})^T \\ \hline \Phi_M^{yx} & \Phi_M^{xx} \end{array}\right]}_{M+2} \Bigg\} M+2$$

$$\bar{H}_M = \left[\begin{array}{c} 1 \\ -\hat{H}_M \end{array}\right] \Bigg\} M+2$$

$$\bar{P}_M = \left[\begin{array}{c} P_M \\ 0 \\ \vdots \\ 0 \end{array}\right] \Bigg\} M+2, \qquad \phi_M^{yy}(0) = Y_M^T Y_M.$$

An alternative relationship for $\bar{\Phi}_M$ is

$$\bar{\Phi}_M = \sum_{n=M+1}^{N} \bar{X}_M(n) \bar{X}_M^T(n), \tag{21}$$

where $\bar{X}_M(n)$ is the vector

$$\bar{X}_M(n) = \left[\begin{array}{c} y(n) \\ x(n) \\ \vdots \\ x(n-M) \end{array}\right].$$

We may obtain the three basic partitions:

$$\bar{\Phi}_M = \underbrace{\left[\begin{array}{c|c} \phi_M^{yy}(0) & (\Phi_M^{yx})^T \\ \hline \Phi_M^{yx} & \Phi_M^{xx} \end{array}\right]}_{\begin{array}{cc} 1 & M+1 \end{array}} \begin{array}{l} \}1 \\ \}M+1 \end{array} \tag{22}$$

$$\Phi_M = \underbrace{\left[\begin{array}{c|c} \bar{\Phi}'_{M-1} & W_M \\ \hline W_M^T & \phi_M^{xx}(M,M) \end{array}\right]}_{\begin{array}{cc} M+1 & 1 \end{array}} \begin{array}{l} \}M+1 \\ \}1 \end{array} \tag{23}$$

$$\bar{\Phi}'_{M-1} = \bar{\Phi}_{M-1} - \bar{X}_{M-1}(M) \bar{X}_{M-1}^T(M), \tag{24}$$

where the definition of W_M is

$$W_M = \sum_{n=M+1}^{N} x(n-M)\bar{X}_{M-1}(n).$$

All the recursive relationships in the algorithm have their roots in eqs. (22) through (24). Using these equations, it is possible to derive the following key recursive relationships for the system identification parameters

$$\bar{H}'_{M-1} = \bar{H}_{M-1} + \frac{e(M)}{(1-\delta_{M-1})}\begin{pmatrix} 0 \\ D_{M-1} \end{pmatrix} \text{ time update} \tag{25}$$

$$\bar{H}_M = \begin{pmatrix} \bar{H}'_{M-1} \\ 0 \end{pmatrix} + \frac{\alpha_M}{P_M^b}\begin{pmatrix} 0 \\ B_M \end{pmatrix} \text{ order update,} \tag{26}$$

where α_M, δ_M are scalar gain values, B_M is the vector of backward linear-prediction coefficients, and $D_M = (\Phi_M^{xx})^{-1}X_M(M+1)$ is a vector, all of which are obtained as part of the linear-prediction-algorithm solution. Equations (25) and (26) highlight the intertwined relationship of the linear-prediction fast algorithm with that of the fast algorithm for the system identification solution. One also can see how all lower-order solutions are obtained recursively along the way to the final selected order.

Counting only second-order terms, the number of multiplications required in the fast algorithm is $2NM + 12M^2$, the number of adds is $2NM + 9M^2$, the number of divides is $8M$, and a total storage of $2N + 7M + 20$ parameters is required (including input and output data sequences). Here N is the number of data values and M is the system order. Roughly NM^2 operations are required to directly solve eq. (6) by Cholesky decomposition if the fast algorithm described here is not used. The Cholesky technique also requires storage proportional to M^2, whereas the fast algorithm requires storage that only increases linearly with increasing M. Exact operation counts for the Cholesky method are given in the appendix. A simple numerical ill-conditioning check in the algorithm is to verify that the scalar variable δ_M in eq. (26) is in the range $0 \leq \delta_M < 1$, which is required by the mathematics of the solution. If it is not in this range, then the fast algorithm recursion has become ill-conditioned.

FORTRAN computer code of the general FIR system identification algorithm may be found in Ref. 5. Code for the linear-phase FIR algorithm may be found in Ref. 8.

III. PERFORMANCE EVALUATION OF THE FAST ALGORITHM

To evaluate the performance of the fast, least-squares, FIR system identification algorithm that solves eq. (6), the system of Fig. 3 was

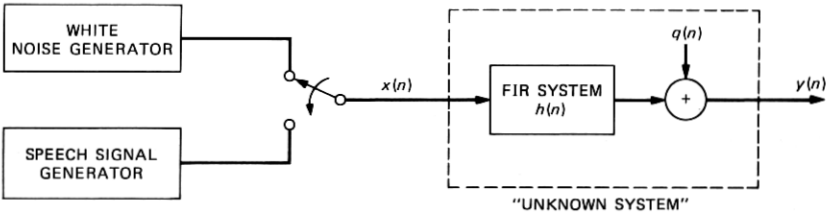

Fig. 3—Block diagram of excitation generation and signal generation for identifying an unknown system.

used to generate the required input and output signals. The signal source was either a white noise signal or a speech signal. The signal source was passed through a given Mth-order FIR system with known impulse response, $h(n)$, $n = 0, 1, \cdots, M$. An additive, random, white noise $q(n)$ was added to the output of the FIR system, at a specified signal-to-noise (s/n) ratio, giving the output signal, $y(n)$. The system was assumed to be in steady state whenever $y(n)$ and $x(n)$ were sampled for system identification purposes [i.e., no initial transients were present in $y(n)$].

The performance measure used in this study was the logarithmic misalignment error defined in Ref. 1 as

$$Q(N, \text{s/n}) = 10 \log_{10} \left[\frac{\sum_{m=0}^{M} [h(m) - \hat{h}(m)]^2}{\sum_{m=0}^{M} h^2(m)} \right], \tag{27}$$

where $\hat{h}(m)$ is a function of N and s/n, and M is the true FIR system order. Note that $h(m)$ are the true FIR coefficients and $\hat{h}(m)$ are the estimated FIR coefficients obtained from the fast algorithm. For nonwhite input signals, a weighted Q function can also be defined using the estimated linear-prediction coefficients, $a(n)$, available as part of the fast algorithm. See Ref. 2 for details.

To fully evaluate the performance of the fast algorithm, five different FIR filters were used, including:

(*i*) Filter 1—A 7-point, nonlinear-phase filter with about 10 dB of spectral deviation across the frequency range. Figure 4 shows plots of the impulse and frequency responses of this filter.

(*ii*) Filter 2—A 25-point, linear-phase, equiripple low-pass filter with about 54 dB of stopband rejection for frequencies above 0.2 times the sampling frequency. Figure 5 shows the impulse and frequency responses of this filter.

(*iii*) Filter 3—A 64-point, nonlinear-phase, reverberation filter with about 30 dB of spectral variation. This filter had an impulse response

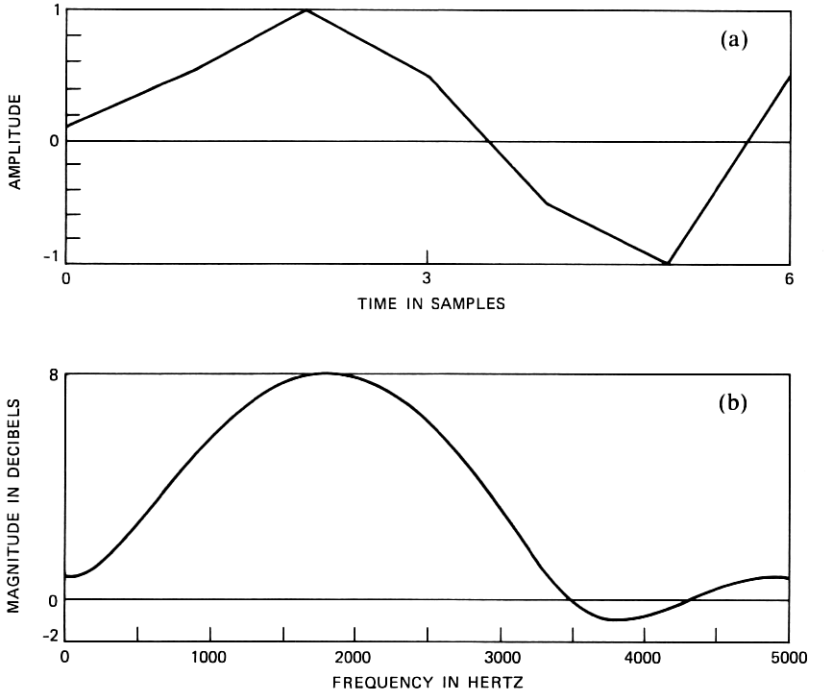

Fig. 4—(a) Impulse response and (b) frequency response of 7-point FIR filter.

that was zero except for $n = 0, 1, 3, 7, 15, 31$, and 63, at which $h(n)$ was 1.0. Figure 6 shows the impulse and frequency responses of this filter.

(*iv*) Filter 4—A 255-point, linear-phase, bandpass filter with 48 dB of rejection in both stopbands. Figure 7 shows the impulse and frequency responses of filter 4.

(*v*) Filter 5—A 256-point, nonlinear-phase, reverberation filter with impulse response that was zero except for $n = 0, 1, 3, 7, 15, 31, 63, 127, 255$, at which $h(n)$ was 1.0.

This set of five filters spans a broad range of impulse-response durations, spectral properties, and temporal properties, and it was felt that it would provide an adequate test of the fast identification algorithm.

3.1 Performance with noise excitation signals

The first set of tests used as the excitation signal the white noise source of Fig. 3. Figure 8 shows plots of the long-time average autocorrelation and power spectrum of the source. The noise spectrum is essentially flat to within ±3 dB.

The white-noise signal was used to drive the system of Fig. 3 for each of the five FIR filters discussed above. For filter 1, data lengths of N from 50 to 1950 were used, and values of s/n from −6 dB to ∞ (no

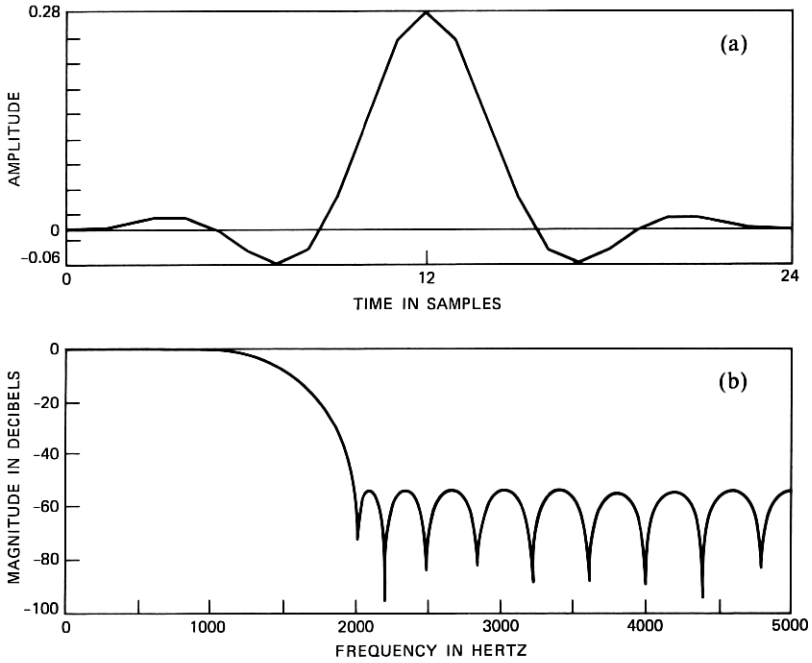

Fig. 5—(a) Impulse response and (b) frequency response of 25-point FIR low-pass filter.

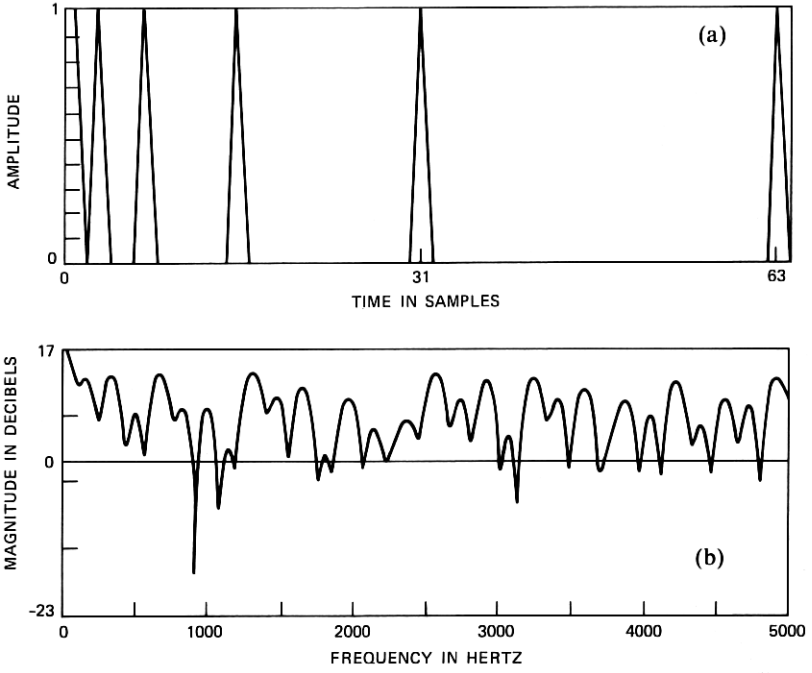

Fig. 6—(a) Impulse response and (b) frequency response of 64-point FIR echo filter.

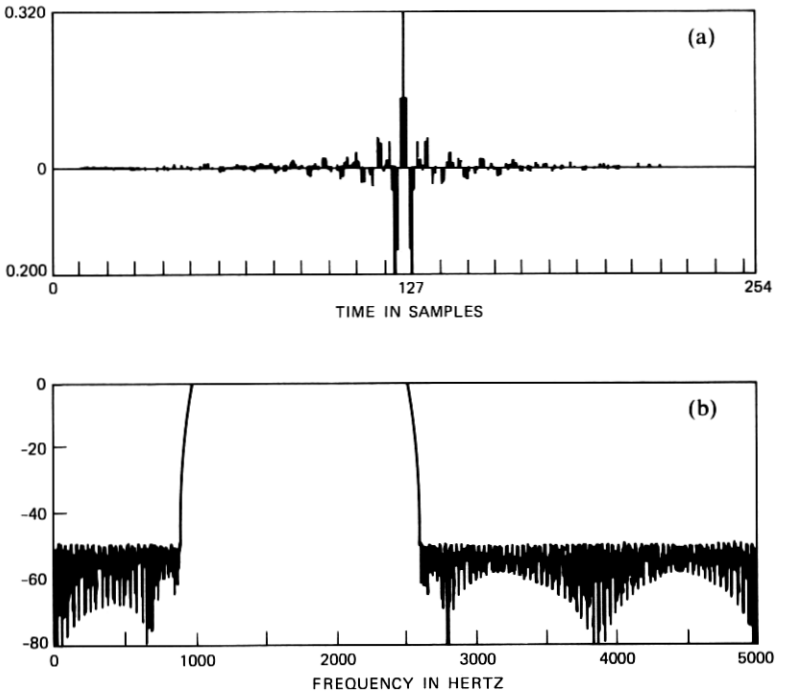

Fig. 7—(a) Impulse response and (b) frequency response of 255-point FIR bandpass filter.

additive noise) were used. For each of the other filters, values of N from the minimum possible $(2M + 1)$ to 1950 were used, along with s/n = ∞ and s/n = 30 dB.

The results for filter 1 performance scores are shown in Fig. 9, which gives a series of curves of $Q(N, \text{s/n})$ versus log N for several values of s/n. Also shown in the figure are the theoretical expected values for $Q(N, \text{s/n})$ for a white-noise input,[1] which are of the form

$$Q(N, \text{s/n})|_{\text{white input}} = 10 \log_{10}(M/N) - \text{s/n(dB)}. \tag{28}$$

It can be seen from Fig. 9 that the measured values of Q are very close to the theoretical expectations for s/n's in the range −6 to 42 dB, and for all N. For s/n = ∞, the measured values of Q (from −88 to −103 dB) reflect the obtainable single-precision accuracy of the computation.

Figures 10 and 11 show similar results for filters 3 and 4, the 64-point reverberation filter and the 255-point bandpass filter. For s/n = ∞, the algorithm had Q values of from −100 dB to −109 dB for the 64-point filter, and from −98 dB to −104 dB for the 255-point bandpass filter. The small degradation in performance is due to the higher roundoff errors for the longer impulse responses. For the case where

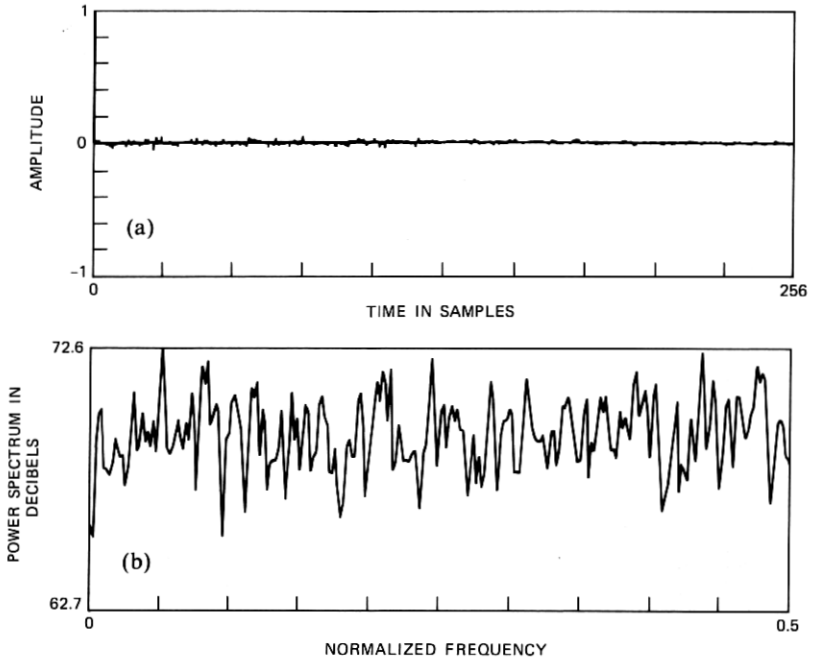

Fig. 8—Long-time average (a) autocorrelation and (b) power spectrum of white-noise source.

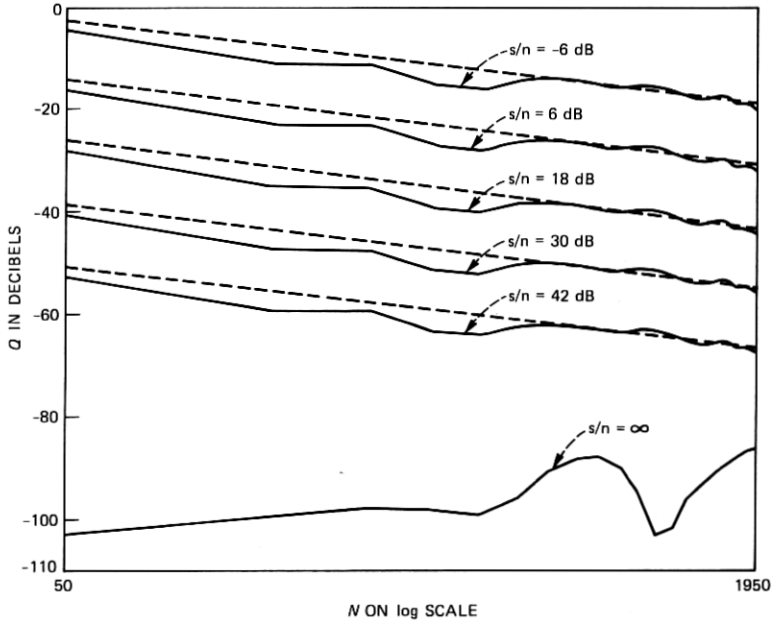

Fig. 9—Plots of Q versus N for several values of s/n for noise input and 7-point FIR filter.

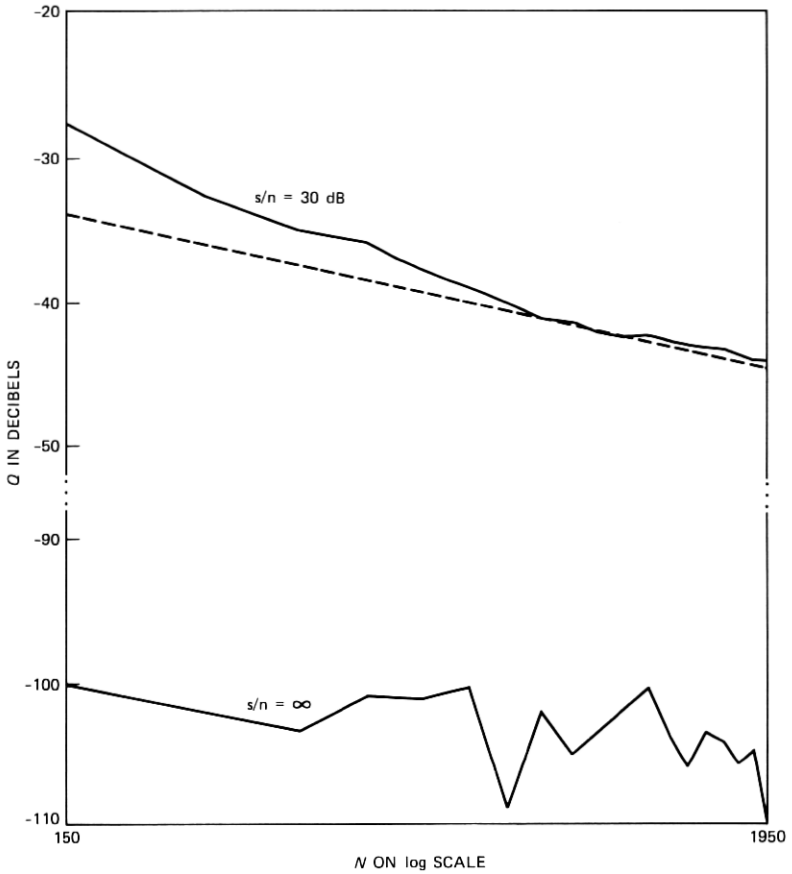

Fig. 10—Plots of Q versus N for s/n = 30 dB and s/n = ∞ for noise input and 64-point FIR echo filter.

s/n = 30 dB the Q curves of Figs. 10 and 11 show about 10-dB worse performance than the theoretical average when N is on the order of $2M + 1$ (the minimum N required to ensure nonsingularity), whereas the performance of the fast algorithm approaches the theoretical estimate as N becomes much larger than $2M + 1$.

The performance of the fast, least-squares estimation algorithm for filters 2 and 5 was essentially identical to that for filters 3 and 4.

As a final example of the noise-excited results, Fig. 12 shows a plot of $Q(N, \text{s/n})$ versus log N for filter 2 (the 25-point low-pass filter) cases, where N is varied from 30 to 70 in steps of 1 and s/n = ∞. The values of Q are very poor for $N \leq 2M$; however, once N exceeds this critical value, the values of Q fall below -75 dB, indicating excellent algorithm performance.

In summary, the results on the white Gaussian noise excitation show

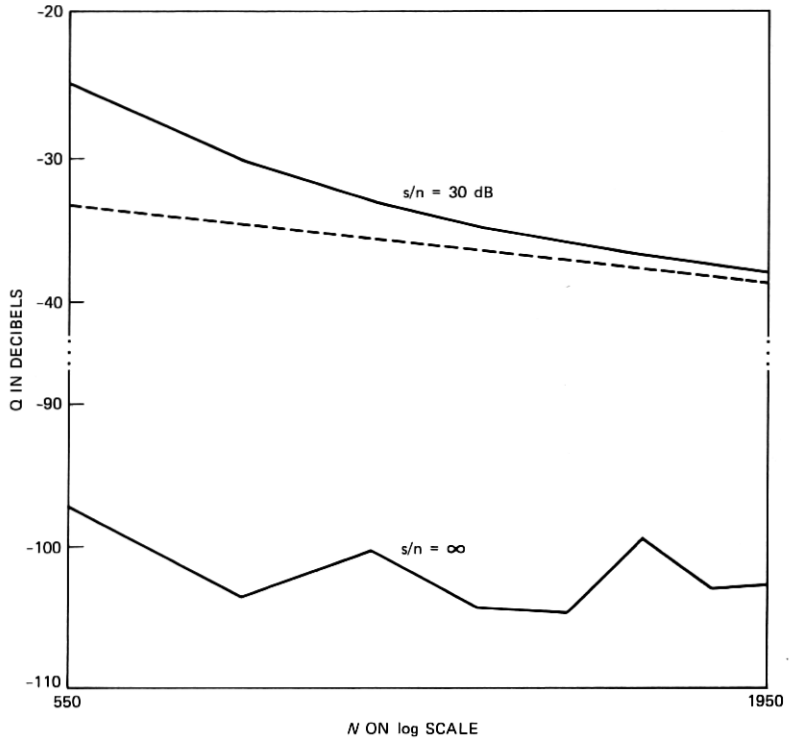

Fig. 11—Plots of Q versus N for s/n = 30 dB and s/n = ∞ for noise input and 255-point FIR bandpass filter.

that the fast, least-squares system identification algorithm performed exceedingly well for all FIR filters, signal-to-noise ratios, and data lengths (so long as $N \geq 2M + 1$).

3.2 Performance on speech excitation

The second series of tests of the performance of the fast system identification algorithm used speech samples as the excitation. Figure 13 shows plots of the long-term average autocorrelation and power spectrum of the speech signal used in our tests. It can be seen that the long-term average power spectrum exhibits a 60- to 70-dB variation in spectral magnitudes, and noticeable gaps in energy throughout the frequency range. Thus, system identification based on speech inputs and outputs is significantly more difficult than it was for white-noise inputs.

The speech excitation was used as input to each of the five FIR filters of Section III. In this section we will concentrate on results obtained using filter 2, the 25-point linear-phase low-pass filter. Results on the other filters were more or less comparable, considering the problems that were encountered.

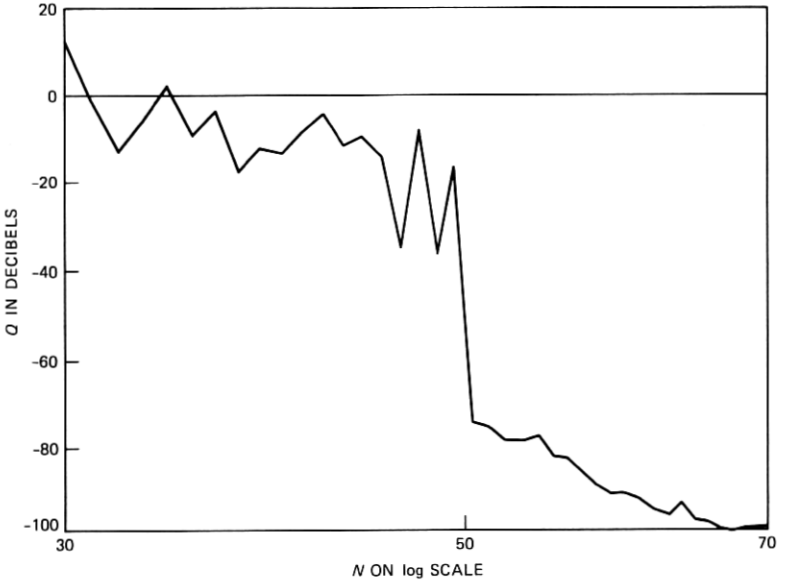

Fig. 12—Plot of Q versus N for 25-point FIR low-pass filter showing abrupt change in Q for $N = 50$.

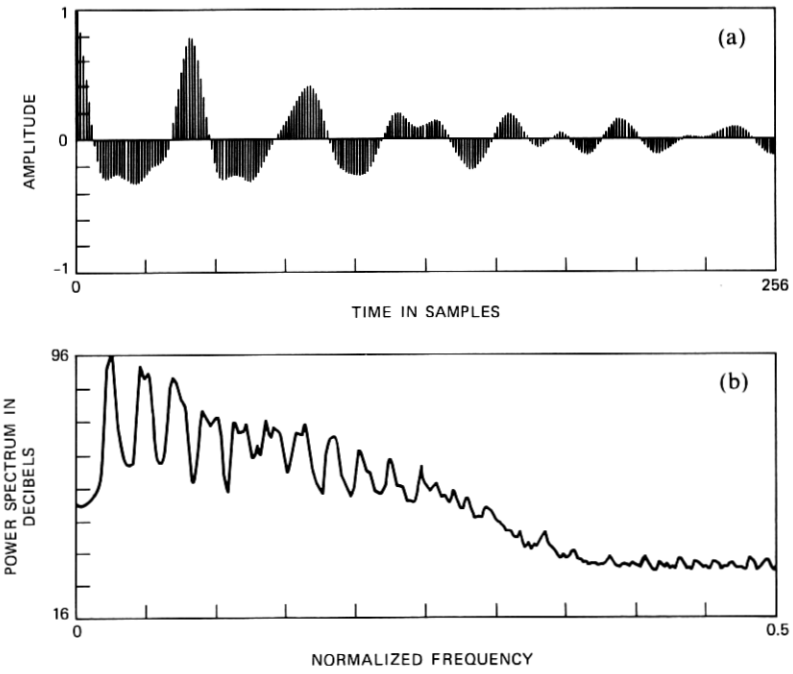

Fig. 13—Long-time average (a) autocorrelation and (b) power spectrum for speech input signal.

Figure 14 shows plots of Q versus log N for s/n = ∞ and s/n = 40 and 60 dB for the filter 2, and for values of N from 100 to 1000. [These results were obtained using the fast algorithm designed for linear-phase systems. Results using the nonlinear-phase system algorithm of eq. (6) were consistently worse than for the linear-phase algorithm, and will not be presented here.] For s/n = ∞ the values of Q range from −30 to −46 dB. These results, for the case with no additive noise,

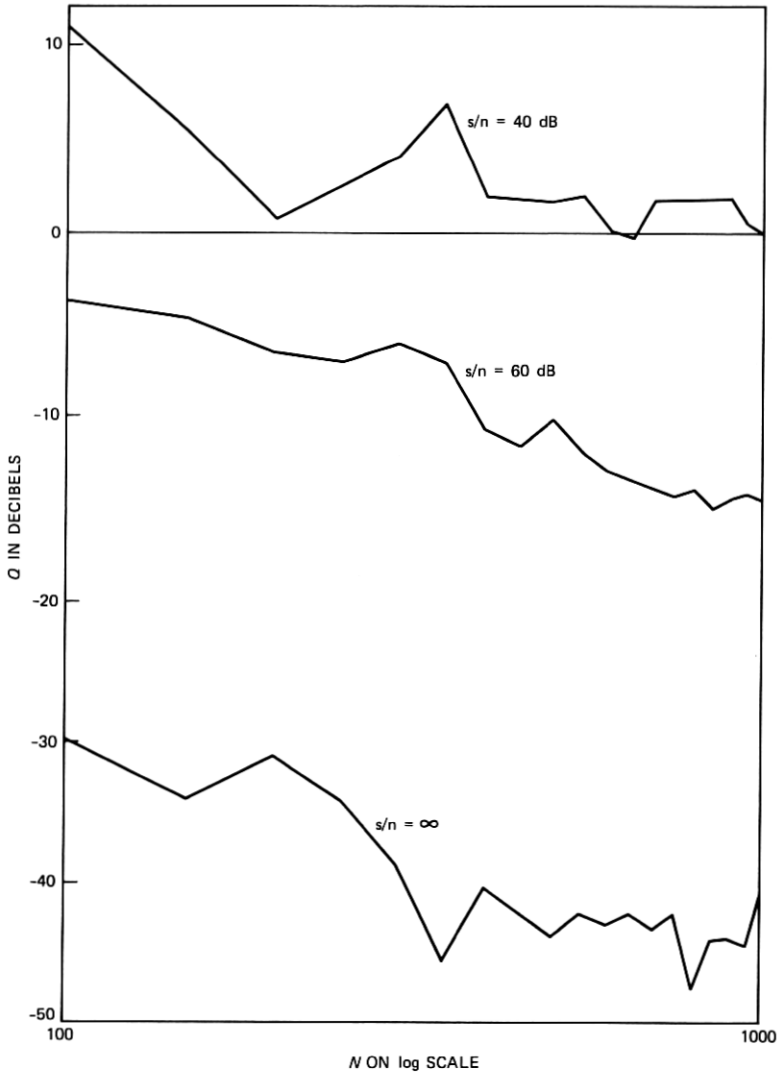

Fig. 14—Plots of Q versus N for several values of s/n for speech input and 25-point FIR low-pass filter.

are significantly worse than those for the white-noise inputs. In the s/n = 40-dB case, all values of Q were about 0 dB or higher, indicating very poor performance. For s/n = 60 dB, the results showing values of Q from −43 to −14 dB indicate marginal performance at best.

The results shown in Fig. 14 are anticipated from previous studies[1-4] and our understanding of the mechanisms of the system identification algorithm. The problem is illustrated in Fig. 15, which shows an estimated impulse response and the resulting frequency response for one set of conditions. It can be seen that the frequency-response estimate is quite good for frequencies in the passband of the low-pass filter (i.e., frequencies less than 0.2 times the sampling frequency). However, the combination of the nonwhite source and spectral gaps with the 54-dB rejection in the stopband leads to extremely poor spectral estimates in the stopband.

To eliminate the spectral gaps in the speech spectrum, a small modification was made to the system block diagram of Fig. 3. A random white noise was added to the speech signal prior to the linear filtering operation to guarantee that all frequencies were present (to some extent) at the input to the linear system. The white noise was

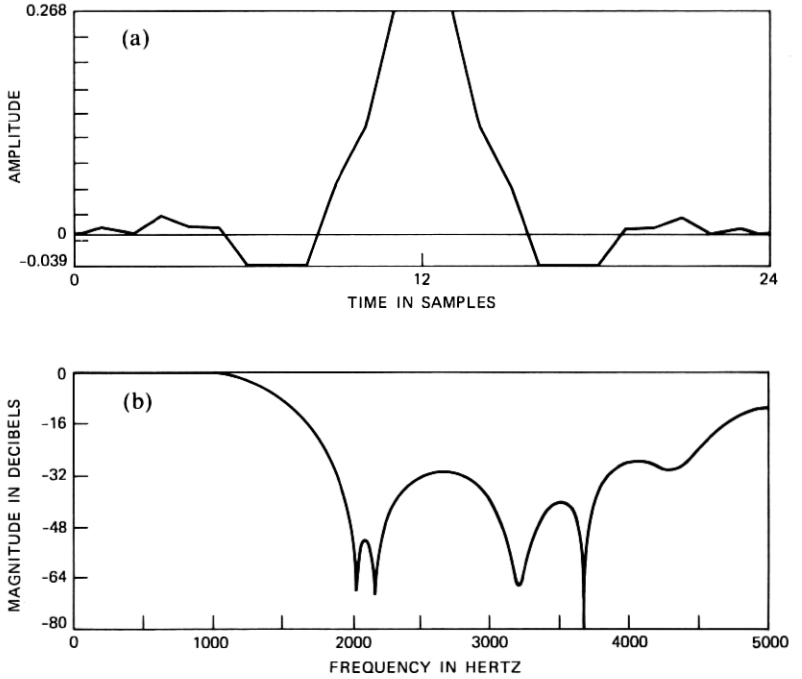

Fig. 15—(a) Plots of estimated impulse response and (b) frequency response for speech input to a 25-point FIR low-pass filter.

added to give an effective signal-to-noise ratio (at the source) of 40 dB. The speech plus noise signal was considered the new input to the system identification procedure.

Figure 16 shows the results obtained using the modified-speech signal input. Shown in this figure are plots of Q versus log N for s/n = ∞, 80, 60, and 40 dB, and for values of N from 100 to 1000. For s/n = ∞ values of Q as high as −78 dB are obtained, showing the vast improvement in system estimation. Similarly, for s/n = 40, 60, and 80 dB vast improvements in Q values are obtained, leading to useful system estimates in most cases.

In summary, the results of system identification using the fast, least-squares algorithm on speech signals indicate that without some spec-

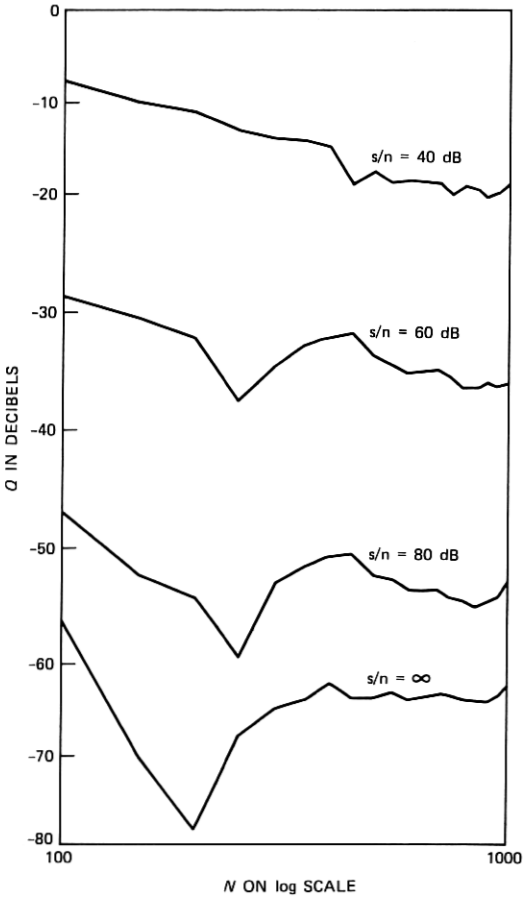

Fig. 16—Plots of Q versus N for several values of s/n for frequency-stabilized speech input to a 25-point FIR low-pass filter.

tral stabilization to guarantee some excitation at all frequencies of interest, the performance of the system identification algorithms is unacceptable in most cases of interest. However, it was shown that even a fairly trivial form of spectral stabilization produced greatly improved system estimates in all cases, and hence made the estimation procedure viable.

IV. DISCUSSION

The results presented in Section III show the following:

(*i*) For white-noise input signals the fast, nonlinear-phase, least-squares system identification procedure performed extremely well over all filter types, filter durations, and signal-to-noise ratios. For such cases the proposed algorithm appears to have significant computational and storage advantages over all other proposed methods.

(*ii*) For nonwhite input signals with spectral gaps (e.g., speech signals) the fast, linear-phase, least-squares system identification procedure was at best barely adequate (for infinite signal-to-noise ratio) and inadequate for signal-to-noise ratios in the practical range of interest (15 to 50 dB). This poor performance was shown to be due primarily to the spectral gaps in the source and a simple modification was proposed whereby a white noise was added to the speech signal to provide a stabilized spectral magnitude at all frequencies of interest.

(*iii*) When the spectral stabilization procedure was used on speech signals the fast, linear-phase, least-squares system identification procedure performed fairly well over a broad range of signal-to-noise ratios and is useful for a wide range of applications.

To appreciate just how well the fast, least-squares algorithm performed, it is worthwhile comparing the results of Section III with those obtained for alternative FIR system identification algorithms. The two main alternative procedures are the short-time spectral analysis (SSA) methods of Rabiner and Allen[1-4] and the classical "slow," least-squares analysis (LSA) solution obtained by the Cholesky method. For the SSA methods it has been found that for white-noise inputs one can approach the theoretical bounds for sufficiently long data sequences (large N). Hence the new fast, least-squares method performs significantly better than the SSA method for noise inputs, except when N becomes very large. In such cases the performances are similar, but the SSA method is still computationally more expensive than the fast, least-squares method.

In the case of speech inputs, the SSA method (which is essentially doing a spectral divide on two power spectrum estimates) runs into extremely bad sensitivity problems because of the missing frequency bands in the input signal. The spectral stabilization method proposed here does not entirely solve the problem for the SSA method because

of the large dynamic range variations in both the input and output signals. The SSA method tends to need a fairly constant spectral input level to perform at its best.

The classical least-squares method, on the other hand, performs as well as the new, fast, least-squares method. However, the computation grows as NM^2 (rather than NM) and thus the classical LSA method is greatly restricted to small values of M and N. In addition to the computational advantage, the fast least-squares algorithm has a significant storage advantage over the classical least-squares approach, growing linearly with the assumed filter order rather than quadratically. Furthermore, the classical Cholesky decomposition used for the LSA method is notoriously sensitive to numerical ill-conditioning and often yields invalid (improper) solutions for cases such as the speech signal input. Hence, except for small values of N and M, the fast, least-squares algorithm appears to be preferable to the classical LSA method.

Another important advantage of using the fast, least-squares algorithm is that it provides information about the squared prediction and identification errors [linear-prediction coefficient (LPC) and FIR] as a function of filter order. These squared-error parameters, which are computed directly in the algorithm at no additional computational cost, are important to monitor for the following reasons:

(*i*) They provide good indications of the required filter order.

(*ii*) The linear prediction and FIR system identification squared errors are monotonically decreasing functions of the filter order (at least for the nonlinear-phase algorithm).

(*iii*) When the linear prediction squared errors fall off more rapidly than the FIR system identification squared error, then the input to the filter is not spectrally rich, and therefore the solution may be ill-conditioned, and the FIR quality, Q, may be poor.

This latter case is illustrated in Fig. 17, which shows plots of FIR system identification squared error on a log scale, and LPC forward and backward squared errors (also on log scales). For this example the 25-point low-pass filter was excited by nonspectrally rich, voiced speech. The LPC prediction errors decrease rapidly for low filter orders, indicating strong spectral coloration in the input signal. In this case a very poor FIR estimate was obtained.

Figure 18 shows a similar set of squared prediction and identification error plots for the case of a white-noise excitation of the same 25-point low-pass filter. In this case the LPC squared errors are essentially flat (to within 0.6 dB) for all FIR filter orders, indicating a spectrally rich input signal. A very good estimate of the FIR filter was obtained in this case.

It should be noted that the squared error of the fast, linear-phase, system identification estimate is not guaranteed to be a monotonically

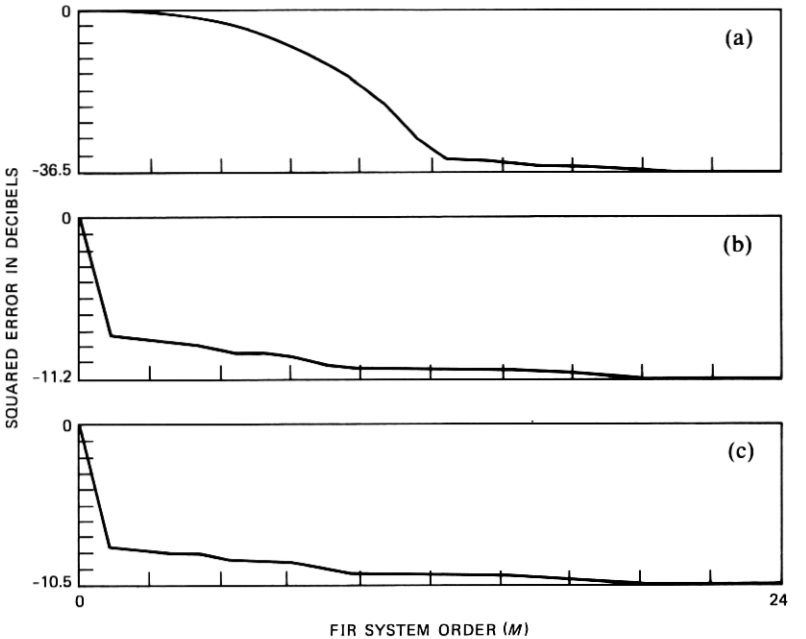

Fig. 17—Plots of squared prediction error versus FIR filter order, M, for a speech signal excitation of a 25-point low-pass filter. (a) FIR error. (b) LPC forward error. (c) LPC backward error.

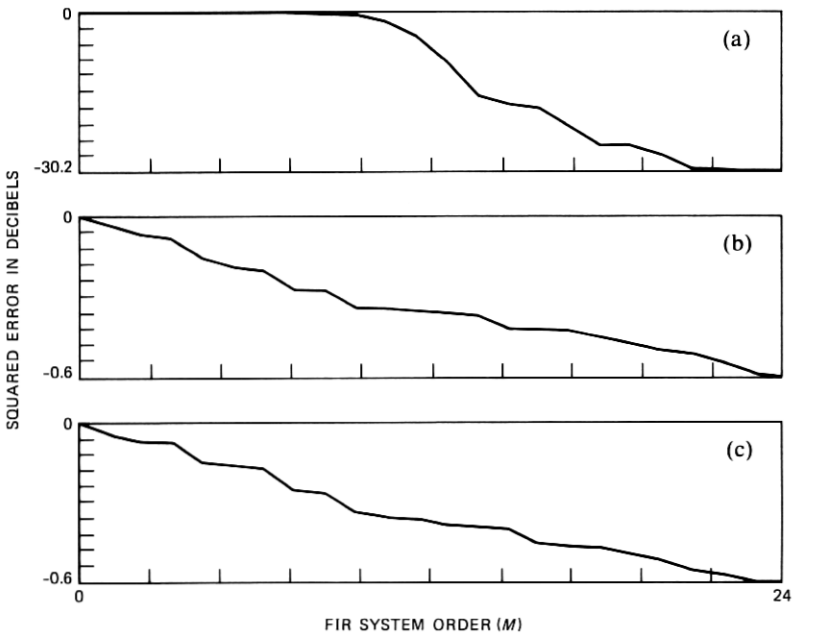

Fig. 18—Plots of squared prediction error versus FIR filter order, M, for a noise-signal excitation of a 25-point low-pass filter. (a) FIR error. (b) LPC forward error. (c) LPC backward error.

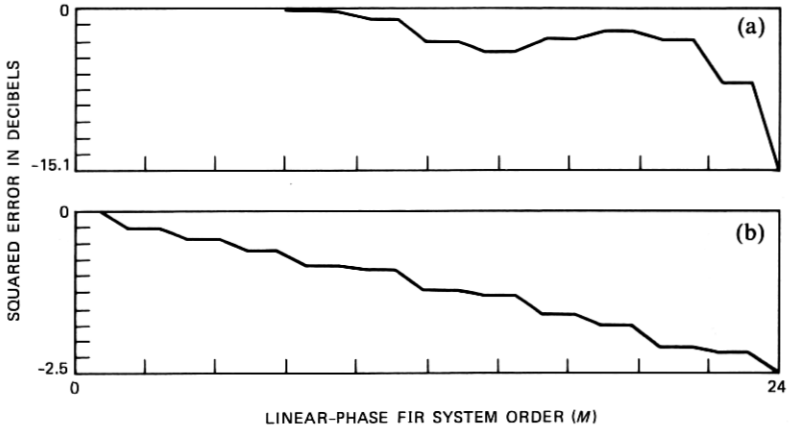

Fig. 19—Plots of squared prediction error versus FIR filter order, M, for the linear-phase algorithm with noise excitation of a 25-point low-pass filter. (a) Linear-phase FIR error. (b) LPC forward and backward error.

decreasing function of filter order. This point is illustrated in Fig. 19, which shows the normalized squared error versus filter order for a noise-signal input to the 25-point low-pass filter. The reader will note that for M in the neighborhood of 19, an increase in squared error occurs. The non-monotonicity of the squared error in the linear-phase algorithm is due to the fact that the algorithm is solving a linear *smoothing* problem (taking future and past samples) rather than a linear *prediction* problem (taking only past samples). Also, separate forward and backward linear prediction squared errors are *not* obtained in the linear-phase, fast algorithm; only a combined forward plus backward linear prediction squared error is obtained.

One final point is worth reiterating. In all cases where the FIR system to be estimated is known to be a linear-phase system, the linear-phase, fast, least-squares method is preferable to the general (nonlinear-phase) procedure in that it requires less computation and yields more accurate results.

V. SUMMARY

We have shown that the fast, least-squares FIR system identification algorithm originally proposed by Marple performs essentially perfectly for white-noise inputs for a wide range of FIR system responses and signal-to-noise ratios. For input signals whose spectrum is highly colored (e.g., speech signals) it was shown that the simple expedient of adding a low-level white noise to the input of the linear system provided a high degree of spectral stabilization and enabled the fast, least-squares algorithm to work well over a wide variety of conditions.

REFERENCES

1. L. R. Rabiner, R. E. Crochiere, and J. B. Allen, "FIR System Modeling and Identification in the Presence of Noise and with Band-Limited Inputs," IEEE Trans. Acoust., Speech, and Signal Processing, *ASSP-26* (August 1978), pp. 319–33.
2. J. B. Allen and L. R. Rabiner, "Unbiased Spectral Estimation and System Identification Using Short-Time Spectral Analysis Methods," B.S.T.J., *58* (October 1979), pp. 1743–63.
3. L. R. Rabiner and J. B. Allen, "On the Implementation of a Short-Time Spectral Analysis Method for System Identification," IEEE Trans. Acoust., Speech, and Signal Processing, *ASSP-28* (February 1980), pp. 69–78.
4. L. R. Rabiner and J. B. Allen, "Short-Time Fourier Analysis Techniques for FIR System Identification and Power Spectrum Estimation," IEEE Trans. Acoust., Speech, and Signal Processing, *ASSP-27* (April 1979), pp. 182–92.
5. S. L. Marple, Jr., "Efficient Least Squares FIR System Identification," IEEE Trans. Acoust., Speech, and Signal Processing, *ASSP-29* (February 1981), pp. 62–73.
6. E. Shichor, "Fast Recursive Estimation Using the Lattice Structure," B.S.T.J., *61* (January 1982), pp. 97–115.
7. M. Morf, B. Dickinson, T. Kailath, and A. Vieira, "Efficient Solution of Covariance Equations for Linear Prediction," IEEE Trans. Acoust., Speech, and Signal Processing, *ASSP-25* (October 1977), pp. 429–33.
8. S. L. Marple, Jr., "Fast Algorithms for Linear Prediction and System Identification Filters with Linear Phase," IEEE Trans. Acoust., Speech, and Signal Processing, *ASSP-30*, No. 6 (December 1982), pp. 942–53.

APPENDIX

Count of Operations to Solve Classical LSA Normal Equations Using Cholesky Decomposition

Solve $\Phi\alpha = \Psi$ for vector α.

Elements of matrix Φ:

$$\phi_{ij} = \sum_{t=M+1}^{N} x(t-j)x(t-i) \qquad \begin{matrix} 0 \le i \le M \\ 0 \le j \le i. \end{matrix}$$

Elements of vector Ψ:

$$\psi_i = \sum_{t=M+1}^{N} y(t)x(t-i) \qquad 0 \le i \le M.$$

In Step 1 we form elements of Φ and Ψ:

+	$\frac{1}{2}NM^2 + \frac{5}{2}NM + 2N - \frac{1}{2}M^3 - 3M^2 - \frac{9}{2}M - 2$
×	$\frac{1}{2}NM^2 + \frac{5}{2}NM + 2N - \frac{1}{2}M^3 - \frac{5}{2}M^2 - 2M$
Storage	$\frac{1}{2}M^2 + \frac{5}{2}M + 2$ (does not include x and y data).

In Step 2 we let $\Phi = VDV^T$. Solve for V and D:

+	$\frac{1}{6}M^3 + \frac{1}{2}M^2 + \frac{1}{3}M$
×	$\frac{1}{6}M^3 + \frac{1}{2}M^2 - \frac{2}{3}M$
Storage	M.

In Step 3 we solve for vector Y in $VY = \Psi$:

+	$\frac{1}{2}M^2 + \frac{1}{2}M$
×	$\frac{1}{2}M^2 + \frac{1}{2}M$
Storage	Store in place.

In Step 4 we solve for α in $DV^T\alpha = Y$

+	$\frac{1}{2}M^2 - \frac{1}{2}M$
×	$\frac{1}{2}M^2 + \frac{1}{2}M + 1$
Storage	Store in place.

In Step 5 we compute residual squared error $E_m = \sum_{t=M+1}^{N} y^2(t) - \alpha^T\Psi$

+	$N + 1$
×	$N + M + 3$
Storage	2.

As a result our total computations are:

Adds	$\frac{1}{2}NM^2 + \frac{5}{2}NM + 3N - \frac{1}{3}M^3 - \frac{3}{2}M^2 - \frac{25}{6}M - 1$
Multiplies	$\frac{1}{2}NM^2 + \frac{5}{2}NM + 3N - \frac{1}{3}M^3 - M^2 - \frac{2}{3}M + 4$
Storage	$\frac{1}{2}M^2 + \frac{7}{2}M + 4$ (not including x and y data).

The fast, least-squares algorithm requires

Adds	$2NM + 9M^2$
Multiplies	$2NM + 12M^2$(+8M divides)
Storage	$7M + 20$ (not including x and y data),

which only counts terms of squared powers.

Running some various numbers for filter order M and data lengths N shows that the Cholesky method is more efficient than the "fast" algorithm when $N < 25$ and $M < 10$. If the Cholesky decomposition must be applied many times to determine the correct order to select, then the fast algorithm is more efficient since all lower solutions are obtained recursively.