The most vital test for a coder is, however, the subjective assessment. Although no detailed comparison of the different coders has been made, the DΣM has been found to offer an acceptable quality of speech while the DM and the modified coder offer toll quality speech at 40 kbits/s.

CONCLUSIONS

We have presented an overview of some of the different methods of realizing a companded unity bit coder where the companding is carried out on the input signal while a linear coder is used to code this compressed signal. Two different methods of extracting the control signal have been presented. One method involves the detection of the presence of digital output bits of the same polarity while the other involves the differentiation of the coder output and filtering, so as to end up with a signal that is inversely proportional to the input envelope. Whatever the companding scheme, the DΣM performs the poorest. A tradeoff between SNR and frequency response (SNR versus frequency) is possible by the use of a two-loop companded coder, which is attractive at lower bit rates, because of its better high frequency reproduction. The quality of speech in the companded DM is the same as that in the CVSD coder.

REFERENCES

[1] N. S. Jayant, "Digital coding of speech waveforms PCM, DPCM and DM quantizers." *Proc. IEEE*, vol. 62, pp. 621–642, May 1974.
[2] J. L. Flanagan, M. R. Schroeder, B. S. Atal, N. S. Jayant, R. Crochiere, and J. M. Tribolet, "Speech coding," *IEEE Trans. Commun.*, vol. COM-27, pp. 710–737, Apr. 1979.
[3] R. Steele, *Deltamodulation Systems*. London, England: Pentech, 1974.
[4] "Data sheet on MC 3417/3418—Continuously variable slope deltamodulator," Motorola Semiconductors Corp.
[5] "Data sheet on HC 55516/55532—All digital CVSD codecs," Harris Semiconductors.
[6] C. L. Song, J. Garodnick, and D. L. Schilling, "A variable step size robust adaptive deltamodulator," *IEEE Trans. Commun. Technol.*, vol. COM-19, pp. 1033–1046, Dec. 1971.
[7] N. S. Jayant, "Adaptive deltamodulation with one bit memory," *Bell Syst. Tech. J.*, vol. 49, pp. 321–342, Mar. 1970.
[8] C. V. Chakravarthy and M. N. Faruqui, "Two loop adaptive deltamodulation systems," *IEEE Trans. Commun.*, vol. COM-22, pp. 1710–1713, Oct. 1974.
[9] A. A. Cartmale and R. Steele, "Calculating the performance of syllabically companded deltasigma modulators," *Proc. Inst. Elec. Eng.*, vol. 117, pp. 1915–1921, Oct. 1970.
[10] C. V. Chakravarthy and M. N. Faruqui, "An amplitude controlled adaptive deltamodulator," *Proc. Inst. Elec. Eng.*, vol. 126, pp. 285–290, Apr. 1979.
[11] H. Stephanne and M. Villert, "An adaptive delta codec with companding outside the loop," in *Proc. IEEE Int. Conf. Commun.*, 1973, pp. 33.25–33.29.
[12] C. V. Chakravarthy, M. N. Faruqui, and J. Das, "A unity bit differential encoder with controlled input loading," in *Proc. 1st Indo-British Symp. Digital Tech.*, Indian Inst. Technol., New Delhi, 1978.
[13] J. D. Johnston and D. J. Goodman, "Multipurpose hardware for digital coding of audio signals," *IEEE Trans. Commun.*, vol. COM-26, pp. 1785–1788, Nov. 1978.
[14] R. M. Wilkinson and D. J. Goodman, "A robust adaptive quantizer," *IEEE Trans. Commun.*, vol. COM-23, pp. 1362–1365, Nov. 1975.
[15] P. K. Chatterjee, "Studies on the role of negative feedback on quantized modulation systems," Ph.D. dissertation, Indian Inst. Technol., Kharagpur, 1967.
[16] S. K. Tewkesbury and R. W. Hallock, "Oversampled, linear predictive and noise shaping coders of order > 1," *IEEE Trans. Circuits Syst.*, vol. CAS-25, pp. 436–448, July 1978.
[17] J. A. Greefkes, "A digitally companded delta codec for speech transmission," in *Rec. IEEE Int. Conf. Commun.*, 1973, pp. 7.33–7.48.

Comparison of Adaptive Linear Prediction Algorithms in ADPCM

MICHAEL L. HONIG, MEMBER, IEEE, AND DAVID G. MESSERSCHMITT, SENIOR MEMBER, IEEE

Abstract—**A comparison of adaptive differential pulse code modulation (ADPCM) speech compression systems is made using different recursive adaptive linear prediction algorithms. The particular algorithms considered are 1) a fixed predictor, 2) the adaptive least mean square (LMS) transversal predictor, 3) the LMS (gradient) lattice predictor, 4) the least squares (LS) lattice predictor, and 5) an LS lattice predictor combined with a third-order pitch inverse filter. The last configuration uses the pitch detection scheme described in [21] to recursively estimate the pitch period in the context of an adaptive predictive coder (APC). The results indicate that for the conditions simulated, the difference in system performance using the different adaptive algorithms is negligible, suggesting that the predictor having the simplest implementation is the best.**

I. INTRODUCTION

This paper reports results obtained from a series of simulations of adaptive differential pulse code modulation (ADPCM) speech compression systems using different recursive adaptive linear prediction algorithms. A block diagram of an ADPCM system is shown in Fig. 1(a). The adaptive (or fixed) linear predictor forms an estimate $\hat{y}_i$ of the current speech sample y_i, and the causal prediction residual e_i is quantized ($\tilde{e}_i$) and sent to the receiver. The received (quantized) data sample is denoted as $\tilde{y}_i$. Also shown in Fig. 1 is an adaptive quantizer, which adjusts the quantizer step size relative to the short-term prediction error power.[1]

Any fixed or adaptive linear predictor may be used in ADPCM. Because prediction alogrithms vary greatly in com-

Paper approved by the Editor for Data Communication Systems of the IEEE Communications Society for publication without oral presentation. Manuscript received June 19, 1981; revised January 21, 1982. This work was supported by the National Science Foundation under Grant ENG78-16966.

M. L. Honig was with the Department of Electrical Engineering and Computer Science, University of California, Berkeley, CA 94720. He is now with Bell Laboratories, Holmdel, NJ 07733.

D. G. Messerschmitt is with the Department of Electrical Engineering and Computer Science, and Electronics Research Laboratory, University of California, Berkeley, CA 94720.

[1] To avoid confusion we state here that throughout this paper "ADPCM" refers to a DPCM coder with an adaptive quantizer and either a fixed or adaptive predictor. Furthermore, the configuration in Fig. 1(b) will be referred to as an "adaptive predictive coder (APC)."

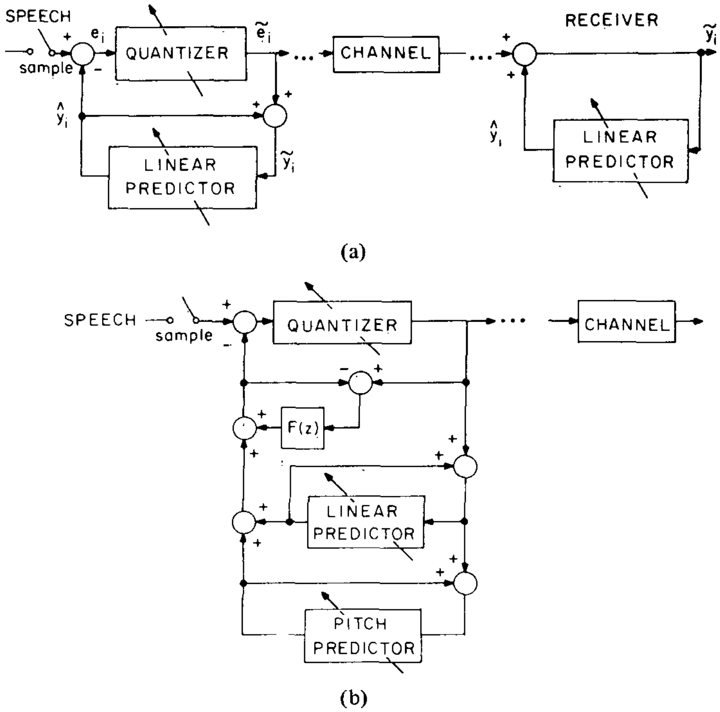


Fig. 1. (a) ADPCM system diagram. (b) Modified ADPCM system diagram.

plexity, the question arises as to whether an increase in predictor complexity yields significant improvement in overall ADPCM system performance. Because of the interaction between the adaptive predictor and the adaptive quantizer, this question is quite difficult to answer without qualification. In addition, if the predictor filter coefficients are not sent directly to the receiver, some error protection scheme must be used in conjunction with recursive algorithms to prevent channel errors from causing the receiver to mistrack the transmitter (possibly creating instability). The approach used here is to compare different adaptive linear prediction schemes assuming no channel errors and the same adaptive quantizer. In addition, for simplicity mean-square prediction error and averaged SNR criteria are used to compare the predictors. The results of informal listening tests are also reported.

Numerous papers concerned with various aspects of ADPCM systems have appeared in the literature over the last 20 years (i.e., [22]-[41]). Many of these papers deal solely with analysis and/or improvement of the adaptive quantizer ([34]-[41]). This can be attributed to the fact that the addition of an adaptive quantizer has been found to have a much greater effect on system performance than the addition of an adaptive predictor [22], [38]. Many of the remaining papers suggest methods for optimizing overall system performance and report simulation results. Recently, much attention has been given to adaptive shaping of the spectrum of the quantization noise and to channel error protection schemes [25]-[27], [33]. Since the focus here is on adaptive linear prediction, these and associated interesting issues are ignored.

In nearly all of the ADPCM simulations reported in the literature, either a block processing method (i.e., the autocorrelation method [2]) or the LMS transversal filter has been used as the adaptive linear predictor. More recently, ADPCM simulations using an adaptive LMS lattice predictor [31] and a Kalman (recursive least squares) predictor [28] have been reported. In none of these cases, however, have different adaptive linear prediction schemes been compared.

In Sections II-A and II-B the ADPCM system is described in more detail, and in Section II-C the simulation results using five different linear prediction schemes are reported.

II. SYSTEM DESCRIPTION

The first four ADPCM systems simulated are described by the block diagram in Fig. 1(a). A block diagram of the fifth ADPCM system simulated is shown in Fig. 1(b) where the quantization noise spectral shaping filter $F(z) = 1$. Design parameters for these cases are associated with either the adaptive quantizer or the adaptive predictor. The adaptive quantization scheme is discussed first since it remained the same in all the simulations.

A. The Adaptive Quantizer

The adaptive quantization scheme was taken directly from [23]. In particular, a symmetric seven-level nonuniform quantizer was used. Denoting the time-varying quantization step size as Δ_i, the input thresholds were 0, $\pm 0.5\Delta_i$, $\pm 1.5\Delta_i$, and $\pm 3.5\Delta_i$, corresponding, respectively, to outputs 0, $\pm\Delta_i$, $\pm 2\Delta_i$, and $\pm 4.5\Delta_i$. The scheme used to adapt the quantization step size Δ_i was presented by Jayant in [36]. After each sample is quantized the quantization step size Δ_i is multiplied by a constant factor depending only on the quantized value of the most recent input sample. The multipliers used for the following simulations were 0.7, 0.8, 0.9, and 2.3 associated with thresholds 0, $\pm 0.5\Delta_i$, $\pm 1.5\Delta_i$, and $\pm 3.5\Delta_i$, respectively.

In [34] and [40] it is shown that channel errors can cause the receiver quantization step size to mistrack the transmitter quantization step size indefinitely. Solutions have been proposed which remedy this problem [34], [40]; however, the resulting system performance is somewhat degraded. Although this problem is of great practical importance, it is ignored here so that the relative performance of each linear prediction algorithm under ideal transmission conditions can be observed.

B. Linear Predictors

The linear prediction schemes simulated are discussed in this section in order of increasing complexity. The simplest scheme used was a fixed coefficient linear predictor. The filter coefficients were computed using the autocorrelation method to find the minimum prediction error coefficients based upon the long-term statistics of the same input speech file used in the simulation. Since the input speech is nonstationary, however, at any given time instant the resulting fixed coefficient filter will not be the optimal predictor based upon the short-term speech statistics. The remaining predictors were adaptive, and attempted to track these short-term statistics.

The second predictor was a normalized LMS transversal algorithm [6]. Denoting the Nth-order forward prediction residual as

$$e_f(i\,|\,N) = y_i - \sum_{j=1} a_{j|N}(i) y_{i-j} \tag{1}$$

where y_i is the current speech sample and the $a_{j|N}(i)$ are the prediction coefficients at time i, the adaptation equation is

$$a_{j|N}(i+1) = a_{j|N}(i) + \frac{y_{i-j} e_f(i\,|\,N)}{D(i+1)} \tag{2}$$

and

$$D(i+1) = wD(i) + y_i^2 \tag{3}$$

where w is a constant close to unity which controls the speed of adaptation and coefficient variation after adaptation.[2]

The LMS lattice predictor was the next case considered. Denoting the nth-order backward residual as

$$e_b(i \mid n) = y_{i-n-1} - \sum_{j=1}^{n} b_{j|n} y_{i-j} \tag{4}$$

where the $b_{j|N}$ are the backward prediction coefficients, the lattice order recursions are

$$e_f(i \mid 0) = y_i, \quad e_b(i \mid 0) = y_{i-1} \tag{5a}$$

$$e_f(i \mid n) = e_f(i \mid n-1) - k_n^{(b)}(i) e_b(i \mid n-1) \tag{5b}$$

$$e_b(i \mid n) = e_b(i-1 \mid n-1) - k_n^{(f)}(i-1) e_f(i-1 \mid n-1) \tag{5c}$$

for $1 \leqslant n \leqslant N$ where $k_n^{(f)}$ and $k_n^{(b)}$ are, respectively, the "forward" and "backward" PARCOR coefficients. Numerous gradient algorithms exist to adapt the PARCOR coefficients [8]-[10], but the simulations use a particular normalized algorithm given by

$$\bar{k}_n(i+1) = w\bar{k}_n(i) + e_f(i \mid n-1) e_b(i \mid n-1) \tag{6a}$$

$$D_n(i+1) = wD_n(i) + e_f^2(i \mid n-1) \tag{6b}$$

$$k_n^{(f)}(i) = k_n^{(b)}(i) = \frac{\bar{k}_n(i)}{D_n(i)}. \tag{6c}$$

Notice that the lattice algorithm (5) takes the current sample as its input and outputs the current prediction error. In particular, looking at the receiver portion of Fig. 1(a), at time i the input to the lattice is the transmitted sample $\tilde{y}_i$ and the output of the lattice is $e_f(i \mid N) = \tilde{y}_i - \hat{y}_i$. The linear predictor, however, must output $\hat{y}_{i+1}$ given the current input $\tilde{y}_i$, and hence the lattice structure cannot be directly used in ADPCM. In order to obtain the predicted value of y_{i+1}, i.e., $\hat{y}_{i+1}$, given $\tilde{y}_i, \cdots, \tilde{y}_{i-N+1}$ using a lattice structure, note that

$$e_f(i+1 \mid N) = e_f(i+1 \mid N-1) - k_N^{(b)}(i+1) e_b(i+1 \mid N-1) \tag{7a}$$

$$= \tilde{y}_{i+1} - \sum_{j=1}^{N} k_j^{(b)}(i+1) e_b(i+1 \mid j-1). \tag{7b}$$

The $e_b(i+1 \mid j-1)$, $1 \leqslant j \leqslant N$, do not depend on y_{i+1} and, hence, can be updated in terms of lattice variables at time i. The second term on the right side of (7) is therefore the causal prediction of y_{i+1}, i.e.,

$$\hat{y}_{i+1} = \sum_{j=1}^{N} k_j^{(b)}(i+1) e_b(i+1 \mid j-1). \tag{8}$$

[2] The parameter w weights the past exponentially and can also be considered as the pole location of a one-pole window. For a discussion of different types of windowing techniques in the context of linear predictive coding see [42].

This leads to the filter structure shown in Fig. 2 which forms the prediction $\hat{y}_i$ given $\tilde{y}_{i-1}, \cdots, \tilde{y}_{i-N}$.

The LS lattice algorithm [12]-[18] recursively adapts the "forward" and "backward" PARCOR coefficients in (5) to minimize, at each time instant i, the sums

$$\sum_{j=0}^{i} w^{i-j} e_f^2(j \mid n) \tag{9}$$

and

$$\sum_{j=0}^{i} w^{i-j} e_b^2(j \mid n) \tag{10}$$

for $n = 1, \cdots, N$ where w is an exponential weighting constant analogous to the w in (3) and (6b). In addition to the order recursions (5), the LS lattice algorithm also uses the following recursions:

$$\gamma(i \mid 0) = 1, \quad R_f(0 \mid 0) = R_b(0 \mid 0) = \delta > 0 \tag{11}$$

$$R_f(i \mid 0) = wR_f(i-1 \mid 0) + y_i^2 \tag{12}$$

$$R_b(i \mid n+1) = R_b(i-1 \mid n) - \frac{\bar{k}_{n+1}^2(i-1)}{R_f(i-1 \mid n)} \tag{13a}$$

$$R_f(i \mid n+1) = R_f(i \mid n) - \frac{\bar{k}_{n+1}^2(i)}{R_b(i \mid n)} \tag{13b}$$

$$\gamma(i \mid n+1) = \gamma(i \mid n) - \frac{e_b^2(i \mid n)}{R_b(i \mid n)} \tag{14}$$

$$\bar{k}_{n+1}(i) = w\bar{k}_{n+1}(i-1) + \frac{e_f(i \mid n) e_b(i \mid n)}{\gamma(i \mid n)} \tag{15}$$

$$k_{n+1}^{(b)}(i) \equiv \frac{\bar{k}_{n+1}(i)}{R_b(i \mid n)}, \quad k_{n+1}^{(f)}(i) \equiv \frac{\bar{k}_{n+1}(i)}{R_f(i \mid n)}. \tag{16}$$

Assuming the term δ is a small positive constant, the variables $R_f(i \mid n)$ and $R_b(i \mid n)$ are approximately equal to the respective sums (9) and (10). The variable $\gamma(i \mid n)$ has been interpreted as an optimal weighting factor [14], [20] and is given by

$$\gamma(i \mid n) = 1 - \mathbf{y}_{i-1|n}^T \Phi_{i|n}^{-1} \mathbf{y}_{i-1|n} \tag{17}$$

where $\mathbf{y}_{i|n}$ is the data vector

$$\mathbf{y}_{i|n}^T = [y_i, y_{i-1}, \cdots, y_{i-n+1}] \tag{18}$$

and $\Phi_{i|n}$ is the sample covariance matrix

$$\Phi_{i|n} \equiv \sum_{j=0}^{i} w^{i-j} \mathbf{y}_{j-1|n} \mathbf{y}_{j-1|n}^T + w^i \delta I. \tag{19}$$

The causal Nth-order prediction of y_{i+1} using the LS lattice algorithm is again obtained via (8).

The LS lattice algorithm uses the same criterion and, hence, should exhibit the same performance as the "fast" Kalman or LS transversal algorithm [11] (ignoring finite wordlength effects), and hence this LS transversal algorithm was not

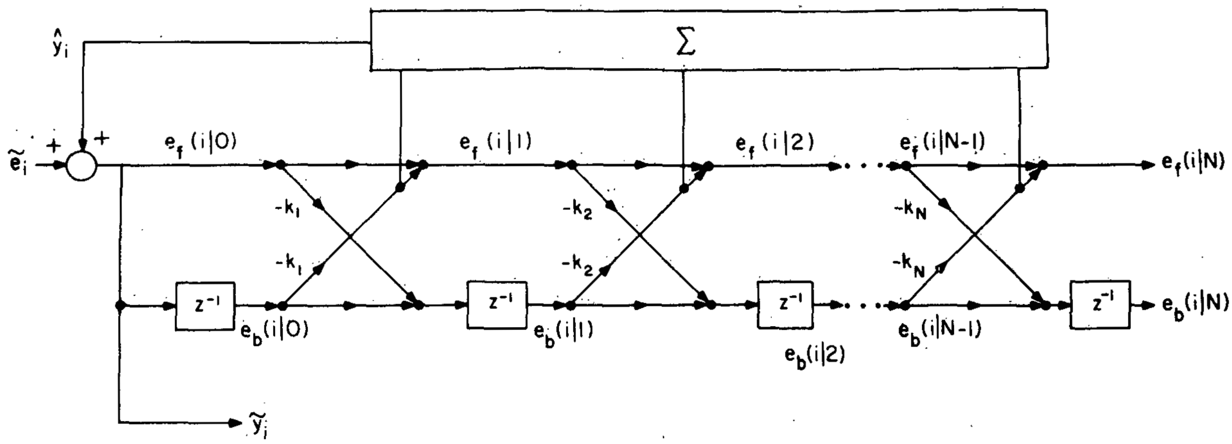


Fig. 2. Lattice predictor.

separately simulated. Although the "fast" Kalman algorithm involves somewhat less computation than the LS lattice algorithm, the lattice algorithm offers other advantages such as its order-recursive structure and a convenient stability criterion [3], [4], [9].

The square-root normalized LS lattice algorithm [16] is unsuitable for use with ADPCM since a causal prediction of y_i cannot be calculated from the normalized lattice residuals without knowing unnormalized LS lattice variables.

The last predictor simulated consisted of an LS lattice linear predictor combined with a third-order all-zero pitch inverse filter or "pitch predictor." This configuration, which is shown in Fig. 1(b), has been called an adaptive predictive coder (APC) [24]. The purpose of the pitch predictor is to further reduce the rms value of the prediction residual to be quantized by exploiting the quasi-periodicity in the prediction residual for voiced utterances. The order of the pitch predictor was taken to be 3 so that the associated transfer function is

$$C(z) = c_1 z^{-(p-1)} + c_2 z^{-p} + c_3 z^{-p+1} \tag{20}$$

where c_1, c_2, and c_3 are the time-varying pitch prediction coefficients and p is the estimated pitch period. The input to the pitch predictor in this case is the residual from the "spectral" LS lattice predictor. The signals at the outputs of the LS lattice and pitch predictors are added to form $\hat{y}_i$.

Previous pitch prediction techniques have computed the autocorrelation function over a block of input data which (hopefully) includes several pitch periods [25], [26]. The lag at which the peak value of the autocorrelation function occurs is then used as the estimated pitch period (this is the optimal estimate in the sense that the resulting mean squared prediction error is minimized).

Due to the large amount of computation required to estimate autocorrelation coefficients, in [26] it was concluded that the amount of computation required to include a pitch predictor outweighs the resulting improvement in system performance. The pitch detection scheme used here was recently presented in [20] and uses LS lattice variables to decide whether or not the current input sample constitutes a pitch pulse. When used in conjunction with an LS lattice predictor, this pitch prediction scheme therefore requires little additional computational overhead. In addition, once the pitch period has been estimated, the pitch prediction coefficients can be estimated recursively, i.e., via the LMS transversal algorithm. The primary disadvantage of this scheme is that errors in the pitch period estimates cause the pitch prediction coefficients to fluctuate rather than converge to steady-state optimal values. Also, the current estimated pitch period may not equal the lag for which the short-term autocorrelation function is a maximum, so that the maximum prediction gain cannot be realized.

A brief description of the pitch detection scheme follows. (For a more detailed description see [17] or [20].) The quantity $1 - \gamma(i|n)$, where $\gamma(i|n)$ is the nth-order optimal weighting factor appearing in the LS lattice algorithm, can be interpreted as a log-likelihood variable if the input is assumed to be a zero-mean Gaussian process. Making this assumption, the log-likelihood ratio

$$\ln \left| \frac{p(y_{i-1}, \cdots, y_{i-n})}{p(y_{i-2}, \cdots, y_{i-n-1})} \right| = \nu_{i|n} - \nu_{i-1|n} \tag{21}$$

where

$$\nu_{i|n} = \ln |R_n| + y_{i-1|n}{}^T R_n{}^{-1} y_{i-1|n}, \tag{22}$$

$y_{i|n}$ is given by (18); and R_n is the $n \times n$ autocorrelation matrix. It is shown in [20] that the log-likelihood variable $\nu_{i|n}$ can be estimated from lattice variables as

$$\hat{\nu}_{i|n} \doteq \sum_{j=0}^{N-1} \ln [R_b(i|j)] + [1 - \gamma(i|n)] \tag{23}$$

where $R_b(i|j)$, $0 \leqslant j \leqslant N-1$, are the backward error covariances in (13a) and $\gamma(i|n)$ is given by (17). In [20] $\nu_{i-1|n}$ is interpreted as "a measure of the likelihood of the duration of successive data samples from a Gaussian distribution," given all of the past data samples. This implies that the sample log-likelihood ratio $\hat{\nu}_{i|n} - \hat{\nu}_{i-1|n}$ indicates how far the present sample deviates from a Gaussian distribution, given all past data values. If it is therefore assumed that speech can be modeled as a mixture of an approximately continuous Gaussian part (i.e., unvoiced speech) plus a discontinuous jump process (i.e., pitch pulses), the estimated log-likelihood ratio can be used to form a good statistic for separating the two components.

The basic pitch detection algorithm used here is as follows.

1) Compute $\hat{v}_{i|n}$.

2) If $\hat{v}_{i|n} - \hat{v}_{i-1|n}$ falls above a given threshold, store the input data value.

3) If the data value falls above an exponentially decaying window initialized at the value of the last pitch pulse (see [21]), record a new pitch pulse.

Extra steps were included to minimize the chance of errors. Accurate pitch estimates are important in this case since inaccurate estimates cause the pitch prediction coefficients to fluctuate, thereby significantly increasing the rms value of the signal to be quantized. Notice that because the computation of the variables $\gamma(i|n)$ for all i is an essential part of this algorithm, this pitch detector cannot be used in conjunction with an LMS lattice algorithm.

Once the pitch period was estimated, the LMS transversal algorithm was used, because of its simplicity, to adapt the three-tap predictor using appropriately delayed data values from the previous pitch period. If inconsistent successive pitch estimates were obtained, or the input speech was determined to be unvoiced, the pitch prediction coefficients were automatically set to zero. For the three-tap case, the response time of the filter, which depends primarily on the number of taps [7], was found to be adequate. Only the quantized data samples to be transmitted were used to compute both "spectral" and pitch predictions, and hence, in the absence of channel errors, the same computations can be reproduced in the receiver.

To illustrate the potential benefits from using this pitch prediction scheme in the context of ADPCM, Fig. 3 shows plots of (a) the input speech waveform corresponding to the sound /ah/, (b) the residual at the output of the LS lattice predictor, and (c) the quantized residual which resulted from filtering the waveform in Fig. 3(b) with a three-tap adaptive pitch predictor. The rms prediction error averaged over the 500 samples shown was reduced from 55.5 to 35.7 by incorporating the pitch predictor. The corresponding SNR increased from 11.7 dB to 17.1 dB. For this case the input is clearly periodic and the pitch period is easily identifiable so that this is comparable to the greatest improvement to be expected for different types of speech sounds.

III. SIMULATION RESULTS

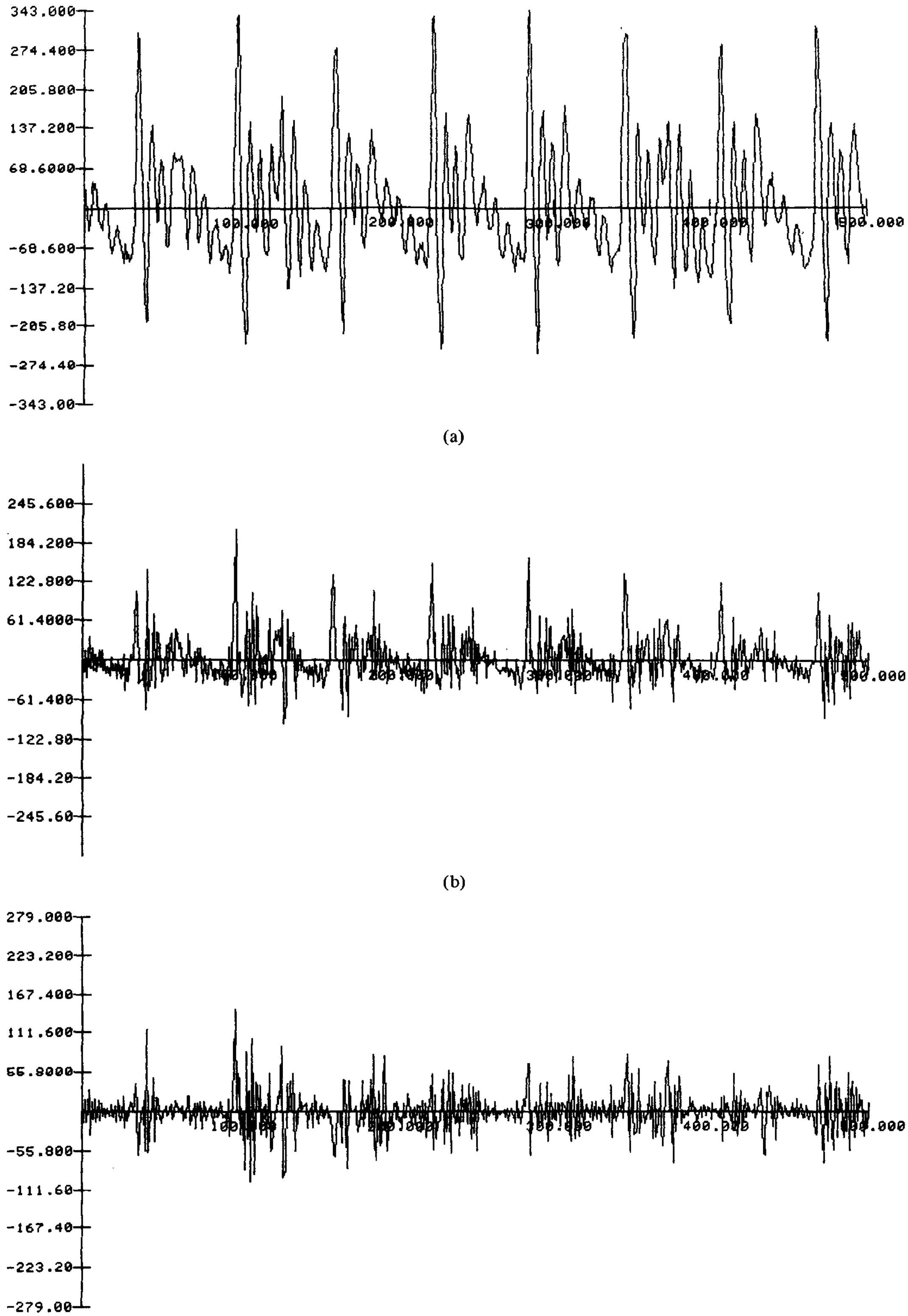
The five ADPCM systems described in the last section were simulated for two input speech files. The first file consisted of a concatenation of different vowel sounds and was selected in order to compare the steady state performance of each algorithm. Figs. 4 and 5 show, respectively, comparisons of rms predictor error and SNR averaged over blocks of 500 samples. Figs. 4(a) and 5(a) compare the LS lattice predictor plus LS pitch prediction scheme with the LS lattice predictor and Figs. 4(b) and 5(b) compare the LS lattice, LMS lattice, LMS transversal, and fixed predictors. In all cases the prediction order was five. A prediction order of ten was also tried; however, the results were similar, indicating that large prediction orders are not advantageous. (This conclusion was previously reached in [38] using block processing techniques.) The adaptation step sizes (which were experimentally chosen to give the best results) were also the same in each case ($w = 0.975$). The more complicated algorithms generally give improved performance; however, the difference in performance between the LS lattice, LMS lattice, and LMS transversal filter is hardly noticeable in this case. The pitch prediction scheme offers significant improvement (i.e., 3–5 dB increase in SNR) as long as the pitch period is estimated correctly. Unfortunately, this is quite difficult to do consistently, and hence the practical benefits of this scheme are severely limited. Informal listening tests indicated that broad-band background noise was most noticeable when the fixed predictor was used; however, no noticeable difference was detected between the other four prediction schemes.

The second speech file used was the phrase "Peter Pan peanut butter." The plosives in this utterance cause the input file to be highly nonstationary, and hence algorithms which adapt faster should yield improved system performance. Our simulation results, however, indicate that the difference in performance between the four adaptive algorithms remained slight. Figs. 6 and 7 compare, respectively, rms predictor error and SNR averaged over blocks of 200 samples. Figs. 6(a) and 7(a) compare the LS lattice plus pitch predictor with the LS lattice, Figs. 6(b) and 7(b) compare the LS lattice predictor with a fixed optimal predictor, and Figs. 6(c) and 7(c) compare the LS lattice, LMS lattice, and LMS transversal algorithms. The LS lattice pitch prediction scheme again shows some improvement over the LS lattice predictor during vowel sounds when the pitch period can be easily estimated. Throughout most of the file, however, the pitch period could not be accurately estimated, and hence the pitch predictor was disabled and the two algorithms exhibit nearly identical performance.

Surprisingly, the LS lattice algorithm does not greatly outperform the fixed predictor in this case. Informal listening tests indicated that there was no clearly noticeable difference among the predictors.

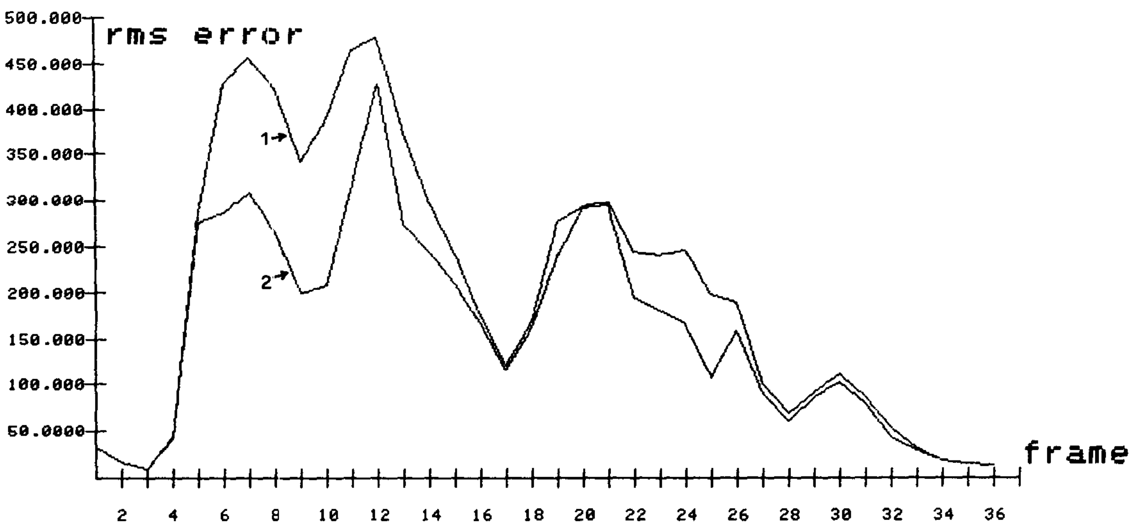
The results in this section can be explained by considering the interaction between the adaptive predictor and the adaptive quantizer. For a uniform quantizer, which is approximately the case considered here, in order to increase the SNR, the quantization step size must be reduced. When an adaptive predictor is used in conjunction with a coarse quantizer, to reduce the quantization step size and, hence, increase the SNR, the input residual energy must be significantly reduced. In cases where all the adaptive predictors are able to track changes in input statistics, the output MSE's will not differ greatly, and hence the quantization step sizes for each case will behave similarly. Because a relatively small predictor order was used, the response time of the LMS adaptive transversal filter, which depends primarily on the number of tap weights [7], is likely to be fast enough for tracking changing input statistics. The extra speed associated with the more complicated algorithms is therefore not needed in this case. Another point is that during a plosive, quantizer overload is by far the dominant source of distortion, and hence it is difficult to detect any difference in background noise for the second set of simulations. Also, this overload distortion, which is fed back to the input of the adaptive predictor, may adversely affect the filter's response time.

We add that an important distinction between the present work and studies which have used "block" processing techniques (i.e., [25] and [26]) is that in the latter case the predictor parameters are computed directly from the input speech and transmitted as side information to the receiver. Block processing techniques therefore eliminate the interaction between the quantizer and predictor which is offered as explanation for the results reported here.

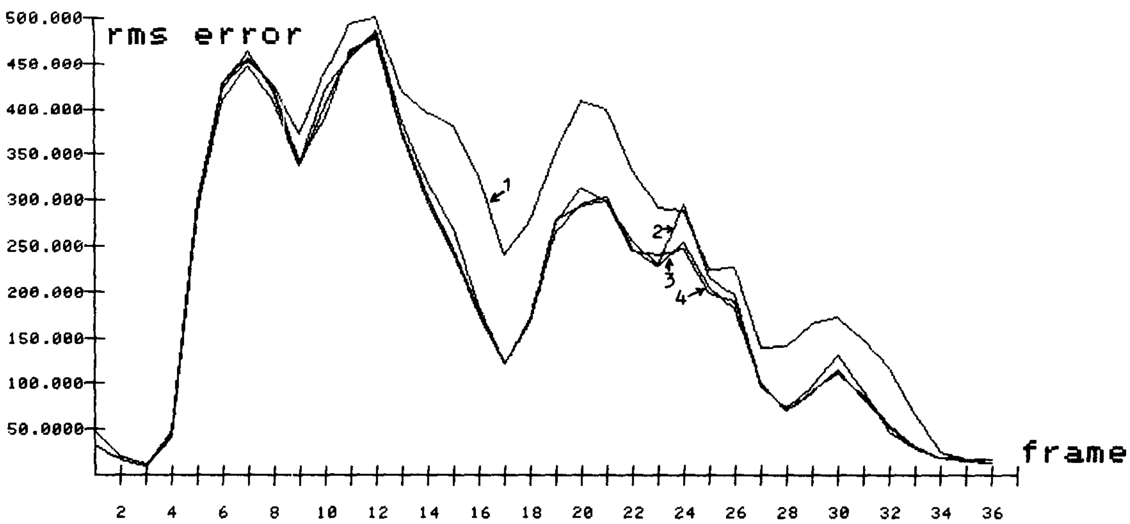


(c)

Fig. 3. (a) Speech waveform corresponding to the sound /ah/. (b) Fifth-order least squares lattice residual. (c) Residual waveform resulting from both "spectral" and pitch prediction.



(a)



(b)

Fig. 4. (a) rms predictor error obtained from using (curve 1) an LS lattice predictor and (curve 2) an LS lattice predictor plus pitch predictor for input vowel sounds. (b) rms prediction error obtained from using (curve 1) a fixed predictor, (curve 2) an LMS transversal predictor, (curve 3) an LMS lattice predictor, and (curve 4) an LS lattice predictor for input vowel sounds.

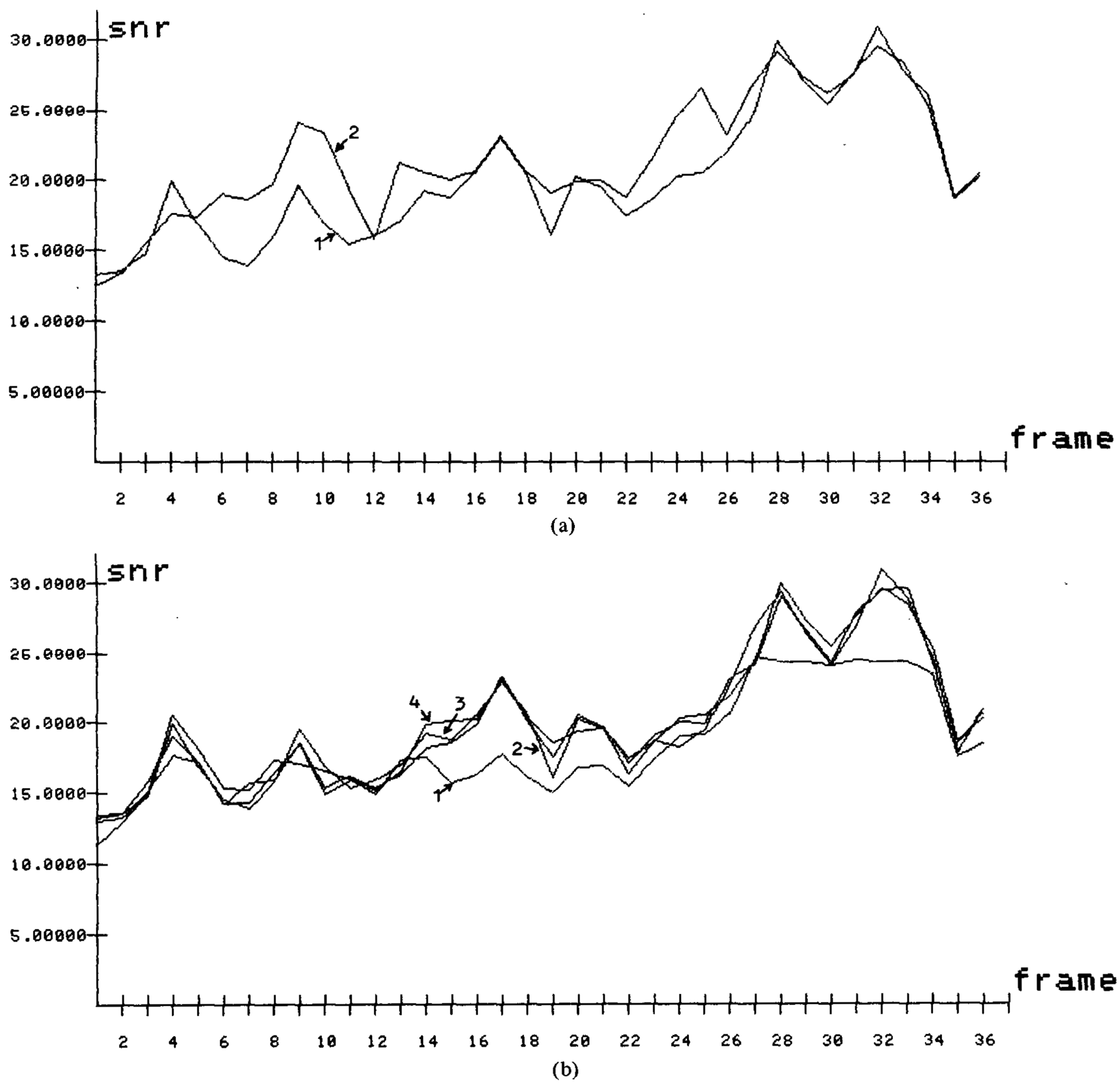


Fig. 5. (a) SNR obtained from using (curve 1) an LS lattice predictor and (curve 2) an LS lattice predictor plus pitch predictor for vowel sounds. (b) SNR obtained from using (curve 1) a fixed predictor, (curve 2) an LMS transversal predictor, (curve 3) an LMS lattice predictor, and (curve 4) an LS lattice predictor for vowel sounds.

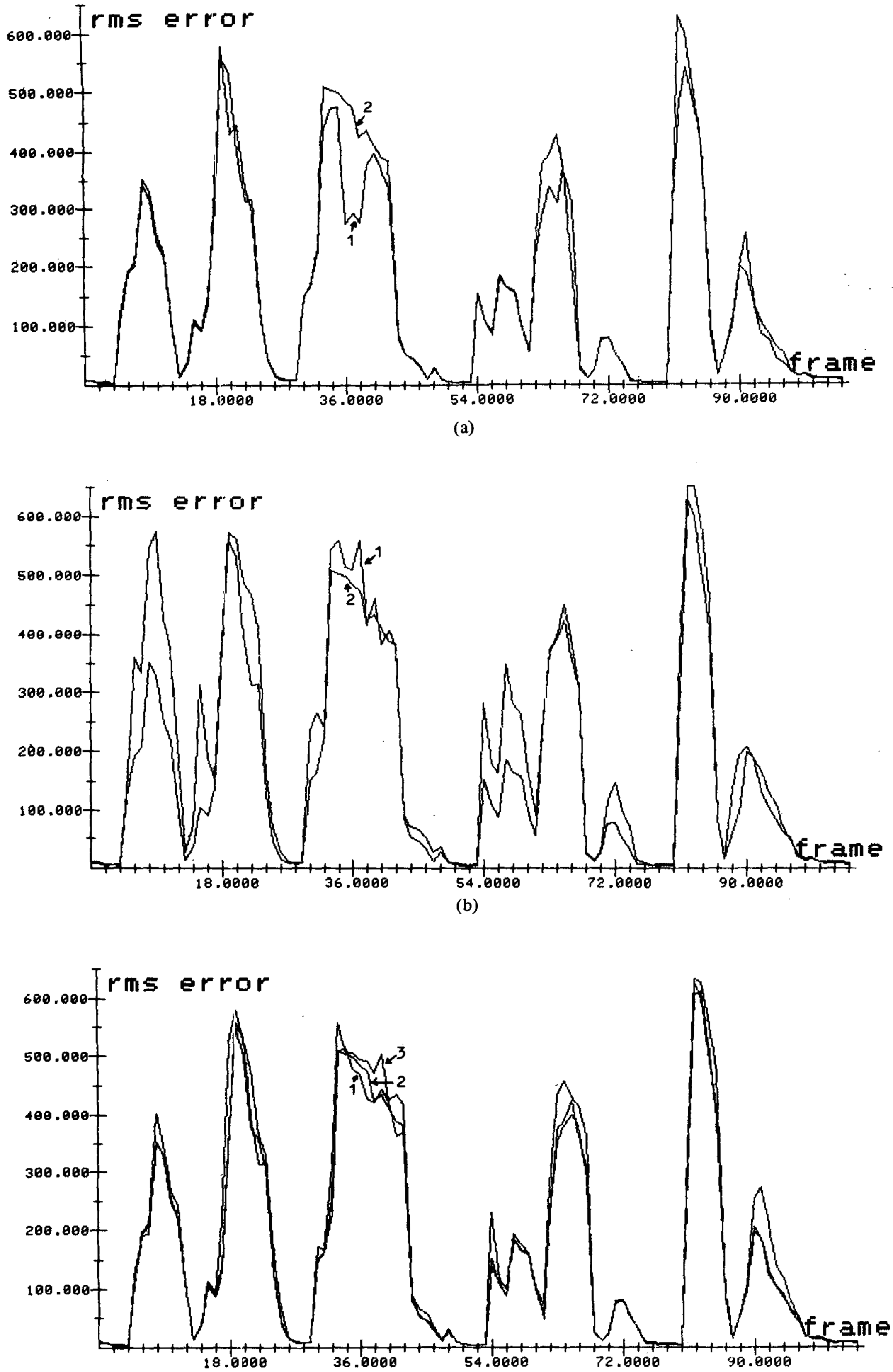


Fig. 6. (a) rms prediction error obtained from using (curve 1) an LS lattice predictor and (curve 2) an LS lattice predictor plus pitch predictor for the phrase "Peter Pan peanut butter." (b) rms prediction error obtained from using (curve 1) a fixed predictor and (curve 2) an LS lattice predictor for the phrase "Peter Pan peanut butter." (c) rms prediction error obtained from using (curve 1) an LMS transversal predictor, (curve 2) an LMS lattice predictor, and (curve 3) an LS lattice predictor for the phrase "Peter Pan peanut butter."

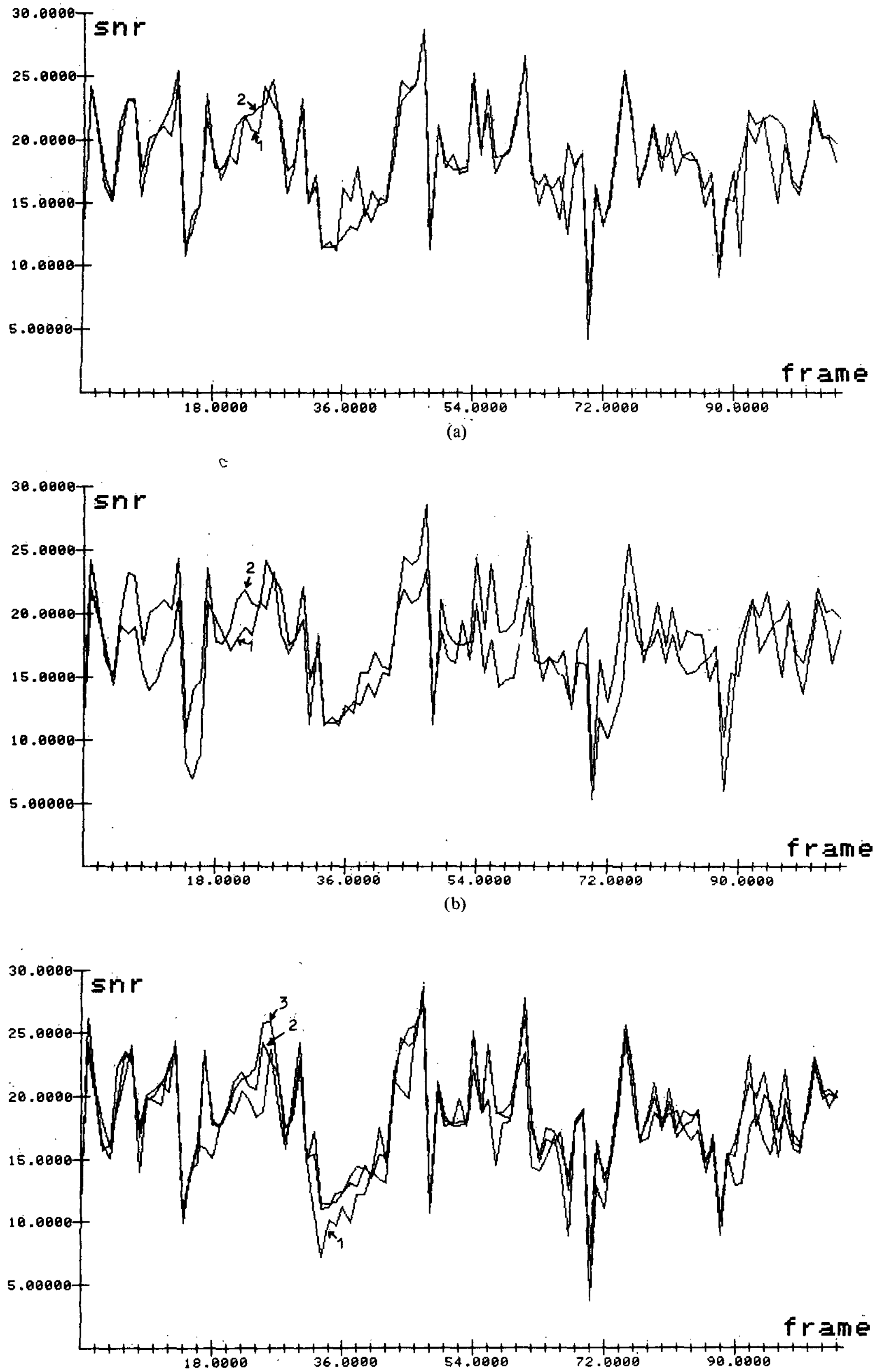


Fig. 7. (a) SNR obtained from using (curve 1) an LS lattice predictor and (curve 2) an LS lattice predictor plus pitch predictor for the phrase "Peter Pan peanut butter." (b) SNR obtained from using (curve 1) a fixed predictor and (curve 2) an LS lattice predictor for the phrase "Peter Pan peanut butter." (c) SNR obtained from using (curve 1) an LMS transversal predictor, (curve 2) an LMS lattice predictor, and (curve 3) an LS lattice predictor fot the phrase "Peter Pan peanut butter."

IV. CONCLUSIONS

The results in Section III indicate that in the context of ADPCM the extra computational burden associated with more complex adaptive linear prediction algorithms outweighs the accompanying improvement in system performance. The improvement in performance in most cases was in fact found to be negligible. The reason for this result is that the appropriate performance criterion, output MSE, is relatively insensitive to the adaptive algorithm used for small prediction orders. In addition, while the LS lattice predictor/pitch prediction algorithm works well for sounds with a clearly identifiable pitch period, the effectiveness of this scheme was compromised due to the difficulty in obtaining accurate pitch period estimates.

While the results in this paper discourage the use of the more complicated least squares algorithms in ADPCM, it should be emphasized that for other applications of adaptive linear prediction, the conclusions may well be different.

REFERENCES

[1] M. L. Honig, "Performance of FIR adaptive filters using recursive algorithms," Ph.D. dissertation, Univ. California, Berkeley, 1981.

[2] J. Makhoul, "Linear prediction: A tutorial review," *Proc. IEEE*, vol. 63, pp. 561–580, Apr. 1975.

[3] A. H. Gray and J. D. Markel, *Linear Prediction of Speech.* Berlin, Germany: Springer-Verlag, 1976.

[4] L. R. Rabiner and R. W. Schafer, *Digital Processing of Speech Signals.* Englewood Cliffs, NJ: Prentice-Hall, 1978.

[5] B. Widrow, "Adaptive filters I: Fundamentals," Stanford Electron. Lab., Stanford, CA, Rep. SEL-66-126 (Tech. Rep. 6764-6), Dec. 1966.

[6] ——, "Adaptive filters," in *Aspects of Network and System Theory*, R. Kalman and N. De Claris, Eds. New York: Holt, Rinehart, and Winston, 1971, pp. 563–587.

[7] G. Ungerboeck, "Theory on the speed of convergence in adaptive equalizers for digital communication," *IBM J. Res. Develop.*, pp. 546–555, Nov. 1972.

[8] L. J. Griffiths, "A continuously-adaptive filter implemented as a lattice structure," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Hartford, CT, pp. 683–686, May 1977.

[9] J. Makhoul, "A class of all-zero lattice digital filters: Properties and applications," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-26, pp. 304–314, Aug. 1978.

[10] E. H. Satorius and S. T. Alexander, "Channel equalization using adaptive lattice algorithms," *IEEE Trans. Commun.*, vol. COM-27, pp. 899–905, June 1979.

[11] D. D. Falconer and L. Ljung, "Application of fast Kalman estimation to adaptive equalization," *IEEE Trans. Commun.*, vol. COM-26, pp. 1439–1446, Oct. 1978.

[12] E. H. Satorius and M. J. Shensa, "On the application of recursive least squares methods to adaptive processing," presented at the Int. Workshop Appl. Adaptive Contr., Yale Univ., New Haven, CT, Aug. 1979.

[13] M. Morf, D. T. Lee, J. R. Nickolls, and A. Vieira, "A classification of algorithms for ARMA models and ladder realizations," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Hartford, CT, Apr. 1977, pp. 13–19.

[14] M. Morf and D. Lee, "Recursive least squares ladder forms for fast parameter tracking," in *Proc. IEEE Conf. Decision, Contr.*, San Diego, CA, Jan. 12, 1979, pp. 1326–1367.

[15] M. Morf, A. Vieira, and D. T. Lee, "Ladder forms for identification and speech processing," in *Proc. IEEE Conf. Decision, Contr.*, New Orleans, LA, Dec. 1977, pp. 1074–1078.

[16] D. T. L. Lee and M. Morf, "Recursive square-root ladder estimation algorithms," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Denver, CO, Apr. 1980.

[17] M. Morf and D. T. L. Lee, "Fast algorithms for speech modeling," Inform. Syst. Lab., Stanford Univ., Stanford, CA, Tech. Rep. M308-1, Dec. 1978.

[18] E. Satorius and J. Pack, "Application of least squares lattice algorithms to adaptive equalization," *IEEE Trans. Commun.*, vol. COM-29, pp. 136–142, Feb. 1981.

[19] M. L. Honig and D. G. Messerschmitt, "Convergence properties of an adaptive digital lattice filter," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-29, no. 3, pp. 642–653, 1981.

[20] D. T. L. Lee and M. Morf, "A novel innovations based time-domain pitch detector," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Denver, CO, 1980.

[21] B. Gold and L. Rabiner, "Parallel processing techniques for estimating pitch periods of speech in the time domain," *J. Acoust. Soc. Amer.*, vol. 46, pp. 442–448, Aug. 1969.

[22] N. S. Jayant, "Digital coding of speech waveforms: PCM, DPCM, and DM quantizers," *Proc. IEEE*, vol. 62, pp. 611–632, May 1974.

[23] D. L. Cohn and J. L. Melsa, "The residual encoder–An improved ADPCM system for speech digitization," *IEEE Trans. Commun.*, vol. COM-23, pp. 935–941, Sept. 1975.

[24] B. S. Atal and M. R. Schroeder, "Adaptive predictive coding of speech signals," *Bell Syst. Tech. J.*, vol. 27, pp. 1973–1986, Oct. 1970.

[25] ——, "Predictive coding of speech signals and subjective error criteria," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-27, pp. 247–254, June 1979.

[26] J. Makhoul and M. Berouti, "Adaptive noise spectral shaping and entropy coding in predictive coding of speech," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-27, pp. 63–73, Feb. 1979.

[27] R. Viswanathan *et al.*, "Speech-quality optimization of 16 kb/s adaptive predictive coders," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Denver, CO, 1980, pp. 520–525.

[28] J. D. Gibson, V. P. Berglund, and L. C. Sauter, "Kalman backward adaptive predictor coefficient identification in ADPCM with PCQ," *IEEE Trans. Commun.*, vol. COM-28, pp. 361–371, Mar. 1980.

[29] J. D. Gibson and L. C. Sauter, "Experimental comparison of forward and backward adaptive prediction in DPCM," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Denver, CO, 1980, pp. 508–511.

[30] J. D. Gibson and R. C. Reininger, "Forward and backward prediction in ADPCM over nonideal channels," in *Proc. IEEE Int. Conf. Commun.*, Seattle, WA, 1980, pp. 42.5.1–42.5.4.

[31] A. Le Guyader and A. Pissard, "Codage differentiel adaptatif de la parole pour le reseau telephonique," presented at the 7ieme Colloque sur le Traitment du Signal et SES Applications, Nice, France, May 28–June 2, 1979.

[32] T. J. Aprille and Y.-L. Kuo, "Two ADPCM algorithms with widely separated error recovery times," *IEEE Trans. Commun.*, vol. COM-27, pp. 876–883, June 1979.

[33] E. Singer, "Techniques for improving the robustness of an adaptive predictive coder in the presence of channel errors," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Denver, CO, 1980, pp. 530–534.

[34] D. Mitra and B. Gotz, "An adaptive PCM system designed for noisy channels and digital implementations," *Bell Syst. Tech. J.*, vol. 57, pp. 2727–2763, Sept. 1978,

[35] P. Cummiskey, N. S. Jayant, and J. L. Flanagan, "Adaptive quantization in differential PCM coding of speech," *Bell Syst. Tech. J.*, vol. 52, pp. 1105–1118, Sept. 1973.

[36] N. S. Jayant, "Adaptive quantization with a one-word memory," *Bell Syst. Tech. J.*, vol. 52, pp. 1119–1144, Sept. 1973.

[37] D. L. Cohn and J. L. Melsa, "The relationship between an adaptive quantizer and a variance estimator," *IEEE Trans. Inform. Theory*, vol. IT-21, pp. 669–671, Nov. 1975.

[38] P. Noll, "A comparative study of various quantization schemes for speech encoding," *Bell Syst. Tech. J.*, vol. 54, pp. 1597–1614, Nov. 1975.

[39] D. J. Goodman and A. Gersho, "Theory of an adaptive quantizer," *IEEE Trans. Commun.*, vol. COM-22, pp. 1037–1045, Aug. 1974.

[40] D. J. Goodman and R. M. Wilkinson, "A robust adaptive quantizer," *IEEE Trans. Commun.*, vol. COM-23, pp. 1362–1365, Nov. 1975.

[41] D. Mitra, "New results from a mathematical study of an adaptive quantizer," *Bell Syst. Tech. J.*, vol. 54, pp. 335–368, Feb. 1975.

[42] J. I. Makhoul and I. K. Cosell, "Adaptive lattice analysis of speech," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-29, pp. 654–659, June 1981.