Example 2: Consider a non-symmetric low-pass complex FIR log filter specified by

$$20\log_{10} D(\omega) = \begin{cases} -30 - j12\omega, & -\pi \le \omega \le -0.24\pi, \\ -j12\omega, & -0.24\pi \le \omega \le 0.3\pi, \\ -30 - j12\omega, & 0.44\pi \le \omega \le \pi, \end{cases} \quad (28)$$

which is approximated with $N = 30$ by the eigenfilter technique. The resultant amplitude response and group delay after 9 iterations are shown in Fig. 2(a) and (b), when $W_p = 5$, $W_s = 6$, $\epsilon_p = \epsilon_s = 0.02$ and $\epsilon = 0.03$ are preassigned. Notice that the peak delay error is larger in the stopband because the delay response is more sensitive when the magnitude is smaller. Fig. 2(c) and (d) present the traces of complex log errors in passband and stopband respectively, which demonstrate that the complex log errors are equiripple. The peak absolute log error in the passband and stopband are 0.04136 and 0.05109, respectively.

IV. CONCLUSION

An effective method has been proposed for the design of linear-phase and nonlinear-phase FIR log filters in this paper. The method is based on the eigenvector computation of an appropriate positive-definite matrix which is real symmetric or complex Hermitian symmetric, hence it is called by the "eigenfilter approach". To design log filters with optimal equiripple real/complex log errors in the real/complex Chebyshev sense, the iterative weighted eigenfilter approach is applied. The weighting function is chosen in such a way that the weighted error minimizes the real/complex logarithmic error, and is updated using the result of the previous iteration, which forms a powerful approach for designing FIR log filters. Another alternative least-squares approach, the weighted least squares algorithm [12], can also be used to design FIR log filters in an analogical manner, and the results are similar to those of the given examples.

REFERENCES

[1] W. K. Pratt, *Digital Image Processing,* New York: Wiley, 1978, Chapter 2.

[2] B. Yegnanarayana, "Design of ARMA digital filters by pole-zero decomposition," *IEEE Trans. Acoust., Speech, Sig. Proc.*, vol. 29, pp. 433–439, June 1981.

[3] T. Kobayashi and S. Imai, "Design of IIR digital filters with arbitrary log magnitude function by WLS technique," *IEEE Trans. Acoust., Speech, Sig. Proc.*, vol. 38, pp. 247–252, Feb. 1990.

[4] T. Kobayashi and S. Imai, "Complex Chebyshev approximation for IIR digital filters using an iterative WLS technique," *IEEE 1990 ICASSP,* pp. 1321–1324.

[5] P. P. Vaidyanathan and T. Q. Nguyen, "Eigenfilter: a new approach to least-squares FIR filter design and applications including Nyquist filters," *IEEE Trans. Circ. Syst.*, vol. 34, pp. 11–23, Jan. 1987.

[6] S. C. Pei and J. J. Shyu, "Design of FIR Hilbert transformers and differentiators by eigenfilters," *IEEE Trans. Circ., Syst.,* vol. 35, pp. 1457–1461, Nov. 1988.

[7] S. C. Pei and J. J. Shyu, "Eigenfilter design of higher order digital differentiators," *IEEE Trans. Acoust., Speech, Sig. Proc.,* vol. 37, pp. 505–511, April 1989.

[8] S. C. Pei and J. J. Shyu, "2-d FIR eigenfilters: a least-squares approach," *IEEE Trans. Circ., Syst.,* vol. 37, pp. 24–34, Jan. 1990.

[9] J. N. Franklin, *Matrix Theory.* Englewoods Cliffs, NJ: Prentice-Hall, 1968.

[10] B. Nobel and J. W. Daniel, *Applied Linear Algebra.* Englewoods Cliffs, NJ: Prentice-Hall, 1977.

[11] C. Y. Chi and Y. T. Kou, "A new self-initiated optimum WLS approximation method for the design of linear phase FIR digital filters," *Proc. of IEEE Int'l Symp. on Circ. Syst.,* pp. 168–171, June 1991.

[12] Y. C. Lim, J. H. Lee, C. K. Chen and R. H. Yang, "A weighted least squares algorithm for quasi-equiripple FIR and IIR digital filter design," *IEEE Trans. Sig. Proc.,* pp. 551–558, March 1992.

Finite-Precision Analysis of the Pipelined ADPCM Coder

Naresh R. Shanbhag and Keshab K. Parhi

***Abstract*—Roundoff error analysis of a pipelined adaptive differential pulse code modulation (ADPCM) coder is presented. The pipelined coder has been developed by employing the relaxed look-ahead technique. It is shown that the precision of the quantized prediction error and those of the predictor coefficients are critical and analytical expressions for lower bounds on these precisions are derived. In addition, an expression for the excess mean-squared prediction error due to finite-precision is also derived. Based on these expressions, a systematic method for assigning precision to various signals in the coder is developed. Simulation results with real image data confirm the results of this analysis.**

I. INTRODUCTION

Pipelined digital signal processing (DSP) algorithms have a much lower hardware overhead as compared to other high-speed techniques. They are, therefore, attractive from an integrated circuit (IC) implementation point of view. Recently, we proposed a fine-grain pipelined adaptive differential pulse code modulation (ADPCM) algorithm in [1] using the technique of *relaxed look-ahead* [2]. This technique is an approximation of the *look-ahead* technique [3], which had been proposed for pipelining recursive digital filters. The result of applying relaxed look-ahead is a pipelined algorithm with minimal hardware overhead and a negligible performance degradation. Therefore, this algorithm is very attractive from IC implementation point of view.

Finite-precision analysis is always necessary before an IC implementation is done. Roundoff error analysis of the basic LMS has been presented in [4]. There are, however, significant differences between the basic LMS predictor and the ADPCM. This is because, in addition to the adaptation feedback, the ADPCM has an inherently recursive structure. Additionally, it also contains a quantizer, which is a nonlinear element. Therefore, it is important to analyze the roundoff error of the pipelined ADPCM (or PIPADPCM).

In this paper, we present the roundoff error analysis of the PIPADPCM coder [1] and verify it through simulations on real image data [5]. In section II, we describe the relaxed look-ahead technique and PIPADPCM coder. The roundoff error analysis of the pipelined coder is presented in section III, the results of which are verified in section IV.

Manuscript received February 2, 1993; revised October 15, 1993. This paper was recommended by Associate Editor G. S. Moschytz. Research for this paper was supported by the army research office by contract number DAAL-90–G-0063.

Naresh R. Shanbhag was with the Department of Electrical Engineering, University of Minnesota. He is now with AT&T Bell Laboratories at Murray Hill, NJ 07974, USA.

Keshab K. Parhi is with the Department of Electrical Engineering, University of Minnesota, 200 Union Street S. E., Minneapolis, MN 55455, USA.

IEEE Log Number 9400250..

II. RELAXED LOOK-AHEAD

Consider the first-order recursion given by

$$x(n) = x(n-1) + a(n)u(n). \tag{2.1}$$

The computation time of (2.1) is lower bounded by a single add time. In order to reduce this lower bound, we apply an M-step look-ahead transformation [3] to (2.1), which leads to the following equation:

$$x(n) = x(n-M) + \sum_{i=0}^{M-1} a(n-i)u(n-i). \tag{2.2}$$

This transformation results in M latches being introduced into the recursive loop, which in turn can be retimed [6] to pipeline the add operation to any desired level. Note that this transformation did not alter the input-output behavior and therefore resulted in a hardware overhead (the second term in (2.2)).

Relaxed look-ahead involves the formulation of various approximations (or relaxations) to reduce the overhead due to look-ahead. The *delay relaxation* involves approximating (2.2) by

$$x(n) = x(n-M) + \sum_{i=0}^{M-1} a(n-D_1-i)u(n-D_1-i), \tag{2.3}$$

which is based on the assumption that the product $a(n)u(n)$ is more or less constant over D_1 samples. Under the same assumption, we can approximate (2.2) as

$$x(n) = x(n-M) + \frac{M}{LA}\sum_{i=0}^{LA-1} a(n-i)u(n-i), \tag{2.4}$$

where we have kept LA ($LA \leq M$) terms from the summation in (2.2). This approximation is referred to as *sum relaxation*. Note that the additional factor M/LA is necessary to make the average output profile of (2.4) identical to that of (2.1) for stationary $a(n)$ and $u(n)$.

The *delay relaxation* and the *sum relaxation* have been employed to pipeline the least mean-squared (LMS) algorithm [7]–[8] in [1]. The resulting pipelined LMS algorithm was then analyzed for convergence and also employed to develop the PIPADPCM coder. The interested reader is referred to [1] for details of this derivation. The PIPADPCM coder (see Fig. 1), with an N^{th} order predictor and operating on a $N_{\text{rows}} \times N_{\text{cols}}$ image frame, is described by

$$\hat{s}(n+M_1) = \mathbf{W}^T(n-M_2)\tilde{\mathbf{S}}\left(n - N_{\text{cols}} + M_1 + \frac{N-1}{2}\right) \tag{2.5(a)}$$

$$\mathbf{W}(n) = \mathbf{W}(n-M_2) + \mu\sum_{i=0}^{LA-1} e_q(n-i)\tilde{s}\left(n - N_{\text{cols}} + \frac{N-1}{2} - i\right) \tag{2.5(b)}$$

$$e(n) = s(n) - \hat{s}(n) \tag{2.5(c)}$$

$$e_q(n) = \mathbf{Q}[e(n)] = e(n) + q(n) \tag{2.5(d)}$$

$$\tilde{s}(n) = \hat{s}(n) + e_q(n) \tag{2.5(e)}$$

where $s(n)$ is the raster scan input formed from the image pixels, $\tilde{s}(n)$ is the reconstructed signal, $\hat{s}(n)$ is the predicted value of $s(n)$, $e(n)$ is the prediction error, $e_q(n)$ is the quantized value ($\mathbf{Q}[.]$ representing the quantization operator with R bits), $q(n)$ is the quantization error, $\mathbf{W}(n)$ is the predictor coefficient vector and μ is the adaptation step-size. M_1 and M_2 are the pipelining levels of the ADPCM and the adaptation recursions, respectively. From (2.5(a)) and (2.5(b)), we observe that the least mean-squared (LMS) algorithm [7]–[8] is being employed to update the predictor coefficients.

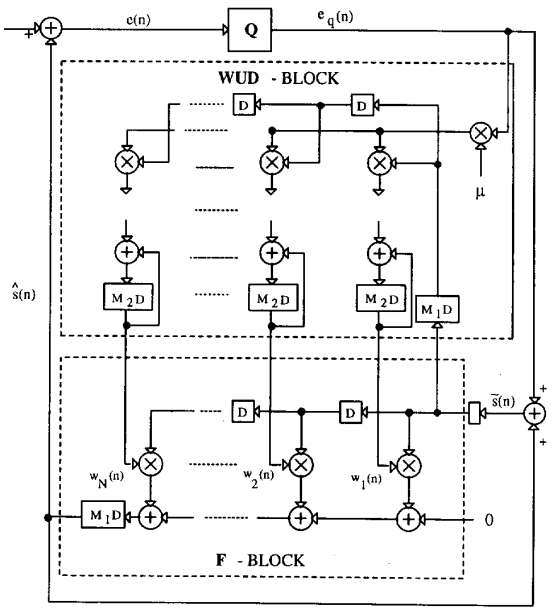


Fig. 1. The PIPADPCM coder.

III. ROUNDOFF ERROR ANALYSIS

The roundoff error analysis presented in this section proceeds along lines similar to that of the LMS algorithm in [4]. Employing the notation in [4], we denote the quantized version of a variable x by x', i.e.,

$$x' = x + \delta x \tag{3.1}$$

where δx is the quantization error. In addition, the number of bits employed in the finite-precision implementation of a variable x is denoted by $B(x)$. Therefore, the quantization error power ($\sigma^2_{\delta x}$) equals $2^{-2B(x)}/12$.

With the notation of (3.1), we can describe the finite-precision PIPADPCM (see Fig. 2) coder as follows

$$\hat{s}'(n+M_1) = \mathbf{W}'^T(n-M_2)\tilde{\mathbf{S}}'(n_1+M_1) + \delta 1(n) \tag{3.2(a)}$$

$$\mathbf{W}'(n) = \mathbf{W}'(n-M_2) + \mu\sum_{i=0}^{LA-1} e'_q(n-i)\tilde{s}'(n_1-i) + \delta 2(n) \tag{3.2(b)}$$

$$e'(n) = s'(n) - \hat{s}'(n) + \delta 3(n) \tag{3.2(c)}$$

$$e'_q(n) = \mathbf{Q}[e'(n)] = e'(n) + q'(n) = e'(n) + q(n) + \delta q(n) \tag{3.2(d)}$$

$$\tilde{s}'(n) = \hat{s}'(n) + e'_q(n) + \delta 4(n), \tag{3.2(e)}$$

where $n_1 = n - N_{\text{cols}} + \frac{N-1}{2}$ and $\delta 1(n), \delta 2(n), \delta 3(n)$ and $\delta 4(n)$ are the roundoff errors made in the computation of (3.2(a)), (3.2(b)), (3.2(c)) and (3.2(e)), respectively.

To make the analysis tractable, we approximate (3.2(b)) as follows

$$\mathbf{W}'(n) = \mathbf{W}'(n-M_2) + \mu LA e'_q(n+M_1)\tilde{s}'(n_1+M_1) + \delta 2(n), \tag{3.3}$$

which is a reasonable approximation to make if the gradient changes slowly over $M_1 + LA - 1$ samples.

Substituting for $\hat{s}'(n)$ (from (3.2(c)) and $e'_q(n)$ (from (3.2(d)) into (3.2(e)) and after some manipulation, we obtain

$$\tilde{s}'(n) = \tilde{s}(n) + \gamma(n) \tag{3.4}$$

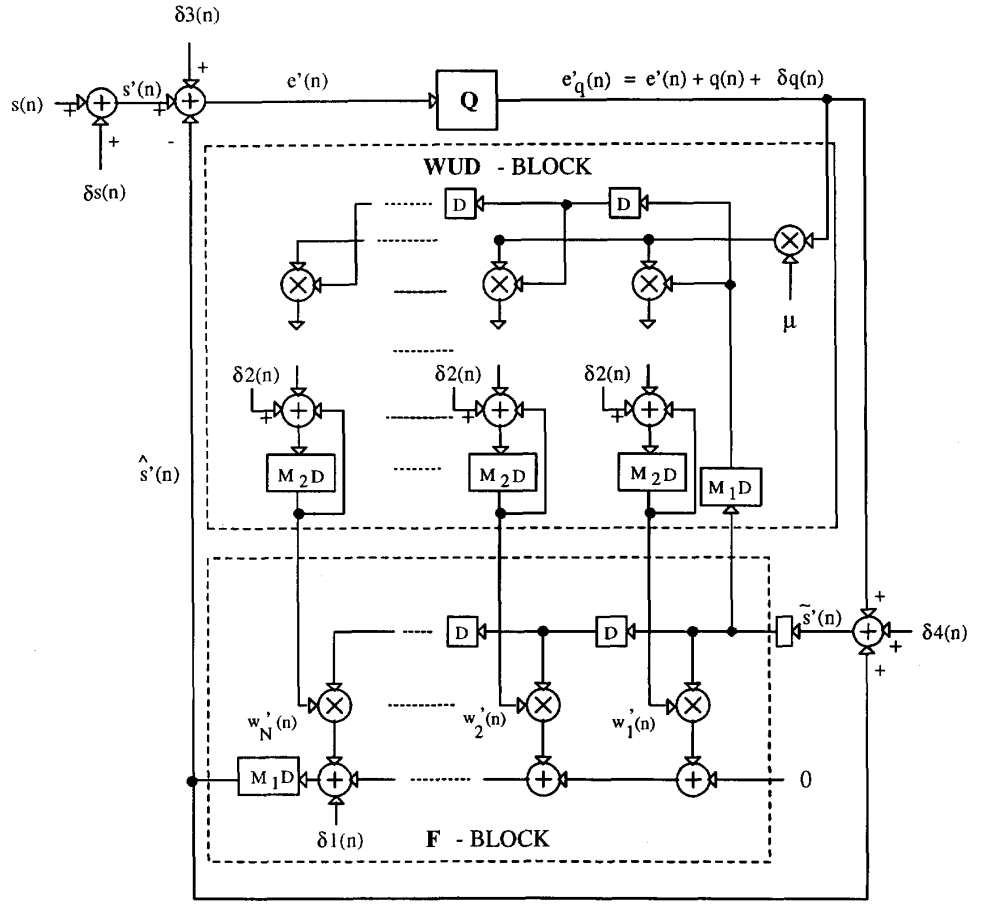


Fig. 2. The finite-precision PIPADPCM coder.

where $\gamma(n) = \delta s(n) + \delta q(n) + \delta 3(n) + \delta 4(n)$ represents the additional quantization noise in the reconstructed output due to finite-precision implementation. In order to minimize σ_γ^2 we have to assign sufficient precision to $e_q'(n)$. Clearly, it is necessary to have $\sigma_{\delta q}^2$ much smaller than σ_q^2. In other words

$$\sigma_{\delta q}^2 = 2^{-\alpha}\sigma_q^2 = 2^{-\alpha}\left[\frac{1}{3}\left(\frac{\Delta_{\mathbf{Q}}}{2^R - 1}\right)^2\right], \tag{3.5}$$

where $\Delta_{\mathbf{Q}}$ is the dynamic range of $\mathbf{Q}$, 2^R is the number of quantizer levels and α is a sufficiently large positive constant. Note that we have assumed a uniform quantizer in (3.5). This results in the following lower bound on $B(e_q)$

$$B(e_q) \geq -\log_2\left(\frac{\Delta_{\mathbf{Q}}}{2^R - 1}\right) + \frac{\alpha}{2} - 1. \tag{3.6}$$

To determine the precision of the predictor coefficients $\mathbf{W}(n)$, consider (3.3). For the weights to continue to adapt, it is necessary that the term being added to $\mathbf{W}'(n - M_2)$ (in (3.3)) be greater than $2^{-B(\mathbf{W})-1}$. Otherwise, due to roundoff, this term will be quantized to zero. Hence, we get the following condition

$$\mu^2 LA^2 E[e_q'^2(n)]E[\tilde{s}'^2(n_1)] + \sigma_{\delta 2}^2 \geq 2^{-2B(\mathbf{W})-2}, \tag{3.7}$$

where $E[.]$ is the expectation operator. From (3.4), we have

$$E[\tilde{s}'^2(n_1)] = \sigma_{\tilde{s}}^2 + \sigma_\gamma^2 = \sigma_s^2 + \sigma_q^2 + \sigma_\gamma^2. \tag{3.8}$$

The computation of $E[e_q'^2(n)]$ is done as follows

$$E[e_q'^2(n)] = \frac{1}{2^R}\left[\sum_{i=0}^{2^R-1} e_{qi}'^2\right], \tag{3.9}$$

where e_{qi}' $(i = 0, \ldots, 2^R - 1)$ are the quantizer codewords. Substituting (3.8) and (3.9) into (3.7) and neglecting $\sigma_{\delta 2}^2$, we obtain the following lower bound on $B(\mathbf{W})$

$$B(\mathbf{W}) \geq \frac{R}{2} - 1 - \log_2(\mu LA) - \frac{1}{2}\left[\log_2(\sigma_s^2 + \sigma_\gamma^2 + \sigma_q^2) + \log_2\left(\sum_{i=0}^{2^R-1} e_{q,i}'^2\right)\right]. \tag{3.10}$$

Finally, in Appendix A, we derive an estimate of the excess mean squared-error due to the quantization of various signals. Following the development in [4], we get the following expression for the mean-squared prediction error for the finite-precision PIPADPCM:

$$E(e'^2(n)) = E(e^2(n)) + \left(|\mathbf{W}_{\text{opt}}|^2 + \frac{1}{2}\mu LAJ_{\min}N\right)\sigma_\gamma^2 + \frac{\mu^2 LA^2\left[(|\mathbf{W}_{\text{opt}}|^2\sigma_\gamma^2 + \sigma_\beta^2)\text{tr}(\mathbf{R}) + N\sigma_\gamma^2(J_{\min} + \sigma_q^2)\right]}{2\mu LA - \mu^2 LA^2\text{tr}(\mathbf{R})}$$

Fig. 3. The original Lenna image.

$$+\frac{N\sigma_{\delta 2}^2}{2\mu LA-\mu^2 LA^2\mathrm{tr}(\mathbf{R})}+\sigma_{\delta s}^2+\sigma_{\delta 1}^2+\sigma_{\delta 3}^2, \tag{3.11}$$

where $\mathbf{W}_{\mathrm{opt}}$ is the optimal predictor coefficient values, $J_{\min}$ is the mean-squared prediction error when $\mathbf{W}=\mathbf{W}_{\mathrm{opt}}$, $\sigma_\beta^2=\sigma_{\delta s}^2+\sigma_{\delta q}^2+\sigma_{\delta 3}^2+\sigma_{\delta 1}^2$, and $\mathrm{tr}(\mathbf{R})$ denotes the trace of the autocorrelation matrix of $\tilde{s}(n)$.

The terms being added to $E(e^2(n))$ in (3.11) are the additional noise terms due to signal quantization. By satisfying (3.6) and providing sufficient precision to $s'(n)$, $\hat{s}(n)$ and $\tilde{s}(n)$, we can make the second and the last three terms in (3.11) negligible. More interesting are the third and fourth terms. As μ decreases the third term is also decreased, while the fourth term increases. Clearly, an optimal value of μ exists at which the sum of these two terms is minimized. This optimal value of μ can be shown to be equal to

$$\mu_{\mathrm{opt}}=\sqrt{\frac{N\sigma_{\delta 2}^2}{LA\left[(|\mathbf{W}_{\mathrm{opt}}|^2\sigma_\gamma^2+\sigma_\beta^2)\mathrm{tr}(\mathbf{R})+N\sigma_\gamma^2(J_{\min}+\sigma_q^2)\right]}}. \tag{3.12}$$

On the basis of (3.6), (3.10) and (3.12), we now provide a systematic method to assign precision to various signals in a finite-precision implementation of the ADPCM codec.

Step 1 Given values of $\Delta_{\mathbf{Q}}$, R and α employ (3.6) to obtain $B(e_q)$. Let $B(\tilde{s})$, $B(\hat{s})$, and $B(e)$ be equal to $B(e_q)$.

Step 2 Employ (3.12) to obtain μ_{opt}.

Step 3 Choose a value of μ close to μ_{opt} and employ (3.10) to obtain $B(\mathbf{W})$.

IV. Simulation Results

In this section, we present simulation results to verify the analysis of the previous section. For this purpose, we consider the coding of the image of 'Lenna' (see Fig. 3) with a frame size of 256×256 and with 256 gray levels scaled to lie between -1 and $+1$.

By extensive simulations, we found that $\alpha=4$ (in (3.6)) is sufficient. In Fig. 4, we plot the output signal-to-noise ratio (SNR) for different values of $B(e_q)$. The two plots are for $R=3$ and $R=5$ with $\Delta_{\mathbf{Q}}=0.25$ for both. The dashed lines in Fig. 4 represent the signal-to-noise ratios (SNR's) of the corresponding infinite-precision implementations. With the values of $R=3,5$, $\Delta_{\mathbf{Q}}=0.25$ and $\alpha=8$, (3.6) predicts $B(e_q)\geq 5.8$ and $B(e_q)\geq 7.95$ for $R=3$ and 5, respectively. This is also indicated quite accurately by the corresponding SNR plots in Fig. 4.

Next, we compare the output SNR for different values of predictor coefficient precision. The values of $R=3$ and $\Delta_{\mathbf{Q}}=0.25$ were employed. The precision of all other signals was fixed at 8 as suggested by (3.6) and Fig. 4. In Fig. 5, we plot the output SNRs for

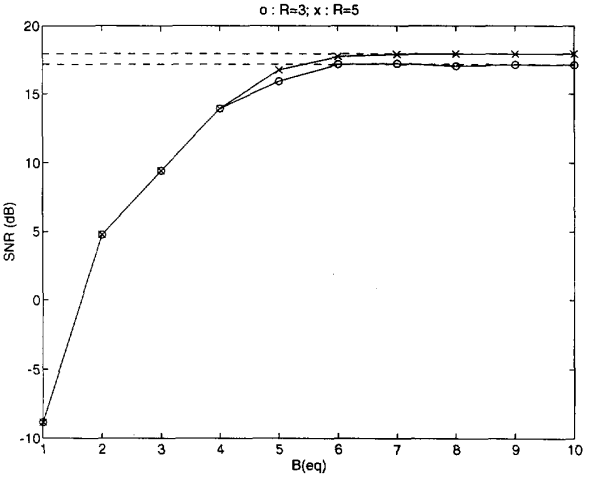


Fig. 4. SNR vs. $B(e_q)$.

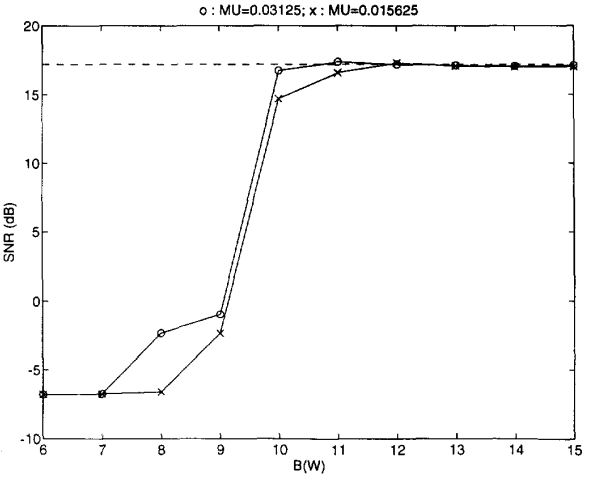
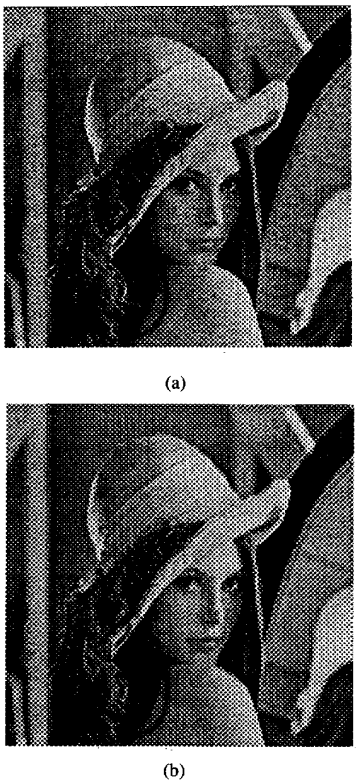


Fig. 5. SNR vs. $B(\mathbf{W})$.

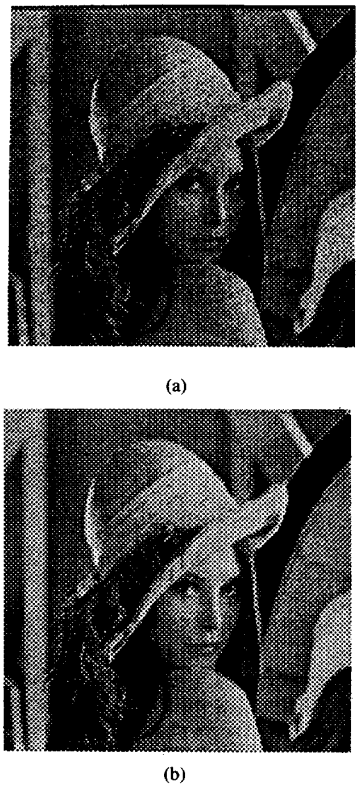
$\mu=2^{-5}$ and $\mu=2^{-6}$. Again, the dashed lines represent the SNR of the corresponding infinite-precision implementation. For $\mu=2^{-5}$ and 2^{-6}, (3.10) predicts a lower bound of 10 and 11, respectively. This is verified experimentally in Fig. 5. The importance of the predictor coefficient precision is evident.

We now compare perceptually the reconstructed outputs of the finite-precision and the infinite-precision PIPADPCM coders. As suggested by (3.6) and (3.10), we fix the precision of predictor coefficients $B(\mathbf{W})=15$ and that of all other signals at 8. The values of $\mu=2^{-6}$, $\Delta_{\mathbf{Q}}=0.25$ and $R=3$ were chosen for all the simulations in this experiment. In Figs. 6 (a)–(b), we show the reconstructed outputs of the PIPADPCM with $M_1=0$, $M_2=1$ and $LA=1$. These values of M_1 and M_2 result in PIPADPCM being equivalent to a serial ADPCM. Perceptually, we do not find any difference in the quality of the two images. This was also found to be true in the case where values of $M_1=40$, $M_2=2$ and $LA=2$ were considered. These values of M_1, M_2 and LA result in a speed-up of approximately 40. Again, the reconstructed outputs (see Figs. 7 (a)–(b)) of the finite and infinite precision PIPADPCM algorithms are perceptually identical. Thus, we verify that the performance



(a) (b)

Fig. 6. Reconstructed output of PIPADPCM with no speed-up: (a) infinite-precision, and (b) finite-precision implementation.



(a) (b)

Fig. 7. Reconstructed output of PIPADPCM with speed-up of 40: (a) infinite-precision, and (b) finite-precision implementation.

of the finite-precision PIPADPCM is close to its infinite-precision counterpart if the signal precisions are obtained from (3.6) and (3.10).

Note also that the PIPADPCM outputs in Fig. 7 are similar to the corresponding ones in Fig. 6. This indicates that the increase in the speed has been achieved without any loss of performance.

APPENDIX A : DERIVATION OF THE EXPRESSION FOR THE MSE

In the derivation below, we neglect second and higher order error terms. Also, we assume that all quantization errors are uncorrelated and independent of the signals themselves.

Substituting for $\hat{s}'(n)$ from (3.2(a)) into (3.2(c)), we obtain

$$e'(n) = s(n) + \delta s(n) - \mathbf{W}'^{T}(n-M)\tilde{\mathbf{S}}'(n_1) - \delta 1(n-M_1) + \delta 3(n), \tag{A.1}$$

where $M = M_1 + M_2$. Next, we substitute for $\mathbf{W}'(n)$ and $\tilde{s}'(n)$ in terms of their infinite-precision variables (see (3.1)) to obtain

$$e'(n) = e(n) - \mathbf{W}^{T}(n-M)\Gamma(n_1) - \delta\mathbf{W}^{T}(n-M)\tilde{\mathbf{S}}(n_1) + \delta s(n) - \delta 1(n-M_1) + \delta 3(n), \tag{A.2}$$

where $\Gamma(n_1) = [\gamma(n_1)\gamma(n_1-1)\cdots\gamma(n_1-N+1)]^T$ and $\gamma(n)$ (see (3.4)) is the quantization error in $\tilde{s}(n)$. Squaring and taking the expected value of (A.2) we get

$$E[e'^2(n)] = E[e^2(n)] + E[\mathbf{W}^{T}(n-M)\mathbf{W}(n-M]\sigma_\gamma^2 + E]\delta\mathbf{W}^{T}(n-M)\tilde{\mathbf{S}}(n_1)\tilde{\mathbf{S}}^{T}(n_1)\delta\mathbf{W}(n-M)] + \sigma_{\delta s}^2 + \sigma_{\delta 1}^2 + \sigma_{\delta 3}^2. \tag{A.3}$$

In (A.3), the first term is the MSE of the infinite-precision PIPADPCM and the rest are the excess MSE due to the finite-precision effects. We now derive analytic expressions for the second and the third terms in (A.3).

The second term in (A.3) can be written as

$$E[\mathbf{W}^{T}(n-M)\mathbf{W}(n-M)]\sigma_\gamma^2 = E[(\mathbf{V}(n-M)+\mathbf{W}_{\text{opt}})^{T}(\mathbf{V}(n-M)+\mathbf{W}_{\text{opt}})]\sigma_\gamma^2 \tag{A.4}$$

where $\mathbf{V}(n)$ is the weight error vector. Employing the analysis in [8, Appendix D(B)], it is easy to show that

$$E[\mathbf{V}(n)^{T}\mathbf{V}(n)] = \frac{1}{2}\mu L A J_{\min} N. \tag{A.5}$$

Thus, we can write (A.4) as

$$E[\mathbf{W}^{T}(n-M)\mathbf{W}(n-M)]\sigma_\gamma^2 = (E[\mathbf{V}(n-M)^{T}\mathbf{V}(n-M)] + |\mathbf{W}_{\text{opt}}|^2)\sigma_\gamma^2 = \left(\frac{1}{2}\mu L A J_{\min} N + |\mathbf{W}_{\text{opt}}|^2\right)\sigma_\gamma^2 \tag{A.6}$$

To obtain an analytic expression for the third term in (A.3) we rewrite it as follows

$$E[\delta\mathbf{W}^{T}(n-M)\tilde{\mathbf{S}}(n_1)\tilde{\mathbf{S}}^{T}(n_1)\delta\mathbf{W}(n-M)] = \text{tr}[E[\delta\mathbf{W}(n-M)\delta\mathbf{W}^{T}(n-M)]\mathbf{R}], \tag{A.7}$$

where tr[.] represents the trace of a matrix.

Next, we substitute for $e'(n)$ from (A.2) into (3.2(d)) to get

$$\begin{aligned} e'_q(n) &= e(n) - \mathbf{W}^{T}(n-M)\Gamma(n_1) - \delta\mathbf{W}^{T}(n-M)\tilde{\mathbf{S}}(n_1) + \delta s(n) - \delta 1(n-M_1) + \delta 3(n) + q(n) + \delta q(n) \\ &= e_q(n) - \mathbf{W}^{T}(n-M)\Gamma(n_1) - \delta\mathbf{W}^{T}(n-M)\tilde{\mathbf{S}}(n_1) + \beta(n), \end{aligned} \tag{A.8}$$